VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

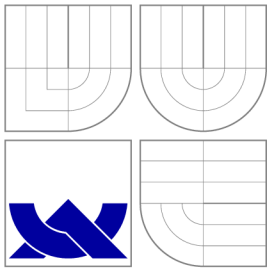
MONITOROVÁNÍ PROVOZU RADIUS POMOCÍ IPFIX

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

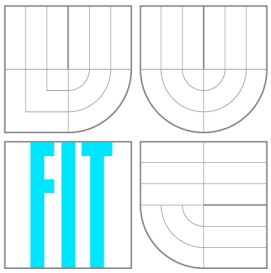
AUTOR PRÁCE
AUTHOR

PAVEL VYSKOČIL

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVÁNÍ PROVOZU RADIUS POMOCÍ IPFIX

RADIUS MONITORING USING IPFIX

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL VYSKOČIL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2014

Abstrakt

Tato bakalářská práce se zabývá monitorováním RADIUS provozu v počítačové síti za pomocí použití technologie IPFIX. Na základě získaných znalostí o RADIUS provozu a možnostech IPFIX protokolu vznikl vstupní plugin pro FlowMon sondu od společnosti INVEA-TECH. Implementovaný plugin během testování prokázal schopnost detekovat a zpracovat RADIUS komunikaci v počítačové síti.

Abstract

This barchelor thesis is focused on monitoring RADIUS traffic in the computer network based on IPFIX technology. A new input plugin for the FlowMon probe from the INVEA-TECH company was created using the acquiered knowledge about the RADIUS traffic and the possibilities of the IPFIX protocol. During the tests, the implemented plugin showed the ability to detect and process RADIUS comunication in the LAN network.

Klíčová slova

RADIUS, IPFIX, FlowMon, monitorování sítě

Keywords

RADIUS, IPFIX, FlowMon, network monitoring

Citace

Pavel Vyskočil: Monitorování provozu Radius pomocí IPFIX, bakalářská práce, Brno, FIT VUT v Brně, 2014

Monitorování provozu Radius pomocí IPFIX

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Matouška, Ph.D.

.....
Pavel Vyskočil
21. května 2014

Poděkování

Na tomto místě bych rád poděkoval panu Ing. Petru Matouškovi, Ph.D. za odborné rady a vedení při zpracování této bakalářské práce. Dále také děkuji Ing. Petru Špringlovi a Mgr. Martinu Elichovi ze společnosti INVEA-TECH a.s. za spolupráci a cenné informace.

Obsah

Kapitola 1

Úvod

Téměř všechny větší firmy jsou závislé na spolehlivosti a zabezpečení svých počítačových sítí. Každý sebemenší výpadek či narušení bezpečnosti sítě může vést k velkým finančním ztrátám nebo nespokojenosti (v horším případě i ztrátě) zákazníků. Díky současným technologiím pro monitorování provozu v síti lze tyto vlivy úspěšně eliminovat. Jedním z nejrozšířenějších řešení je použití technologie NetFlow/IPFIX [4] u zařízení umožňující měření a monitorování sítě na základě IP toku. Zvýšení bezpečnosti a přehledu nad tím, kdo a jakým způsobem síť používá, lze dosáhnout pomocí protokolu RADIUS [14].

Předmětem a hlavním cílem této práce bylo navrhnout a vytvořit rozšíření pro sondu FlowMon poskytnutou společností INVEA-TECH, díky kterému bude schopna detekovat provoz RADIUS a z něho získávat užitečné informace o uživatelích z počítačové sítě, ve které je monitorovací zařízení zapojeno. Vytvořený plugin má dále za úkol rozšířit záznamy o toku IPFIX, ze kterých se vytvářejí podrobné statistiky. Doplňujícím cílem této práce bylo vytvořit vizualizační nástroje pro zobrazení nasbíraných dat týkajících se komunikace RADIUS.

Úvodní kapitoly jsou svým charakterem teoretickým základem pro praktickou část práce. Druhá kapitola seznámí čtenáře se samotným protokolem RADIUS, jeho vlastnostmi, s důvody, které vedly k jeho vzniku a také nastíní jeho budoucnost v oblasti počítačových sítí. Následující kapitola popisuje technologii NetFlow a exportní formát IPFIX. Jakým způsobem jsem postupoval při návrhu a implementaci pluginu sondy FlowMon popisuje kapitola 4. V ní lze dále naleznout informace týkající se zařízení FlowMon. Kapitolou 5 již začíná popis praktické části práce. Ta je věnována způsobům jakým jsem vytvořený plugin testoval. Zároveň nabízí zhodnocení výsledků testů a návrhy na další rozšíření funkcionality implementovaného pluginu. Poslední kapitola je věnována závěrečnému zhodnocení práce a dosažených výsledků.

Praktická část práce spočívala v implementaci rozšiřujícího pluginu do sondy FlowMon, po jehož spuštění je veškerý provoz RADIUS v síti monitorován a jsou o něm uchovávány potřebné informace umožňující další analýzu provozu. Dále bylo nutné upravit konfigurační soubory dodaných nástrojů tak, aby akceptovaly nově vytvořený plugin. V poslední části jsem navrhl a vytvořil webové rozhraní umožňující vizualizaci nasbíraných dat a vypracoval z nich jednoduché statistiky, které jsou uživatelům prostřednictvím tohoto rozhraní zobrazovány.

Kapitola 2

RADIUS

Bylo by velmi obtížné monitorovat provoz RADIUS bez znalosti vlastního obsahu pojmu *RADIUS*. Co tedy RADIUS je, k čemu slouží, jak funguje, čeho využívá a proč je v reálném světě rozšířený? Odpovědi na tyto otázky obsahuje právě tato kapitola.

Dříve, než-li se přesunu k samotnému protokolu, zmíním se o některých oblastech, které s protokolem RADIUS úzce souvisí. Ve spojení s protokolem RADIUS se budeme setkávat s pojmy *klient* a *server*, a proto nejdříve ujasním jaké role budou tyto pojmy představovat.
Klient - v tradičním slova smyslu jde o aplikaci, která vytváří požadavky na použití zdrojů jiných počítačových stanic. Klientem může být počítačová stanice, která transformuje požadavky uživatele, který žádá například o přístup k internetu.
Server - proces, který reaguje na klientovy požadavky, ve kterých žádá o zdroje. Může jím být třeba síťový server, který dokáže plnit funkci autentizace, autorizace a účtování.

2.1 Autentizace, autorizace a účtování

Pojmy autentizace, autorizace a účtování jsou v oblasti počítačové bezpečnosti a sítí známé spíše pod zkratkou AAA vzniklou z anglických slov *authentication, authorization and accounting protocol*. Jde o protokol umožňující kontrolu nad tím, k jakým službám mají kteří uživatelé přístup nebo také kolik prozatím využili prostředků. Jedním ze síťových protokolů, který tyto funkce poskytuje, je RADIUS. Mohlo by se zdát, že AAA model vznikl dříve než RADIUS. Není tomu tak. RADIUS byl pouze prvním z protokolů, který modelu AAA odpovídal, avšak tomuto modelu vděčí za to, že je v praxi hojně využíván.
A proč vůbec AAA architektura vznikla?

Dříve neexistoval žádný standard, který by říkal, jakým způsobem se má autentizace provádět, ale přesto se nějaký způsob autentizace požadoval. Z toho důvodu každý fyzický stroj, který autentizaci vyžadoval, potřeboval nějaký autentizační mechanismus, který závisel na konkrétním zařízení. Je však zřejmé, že s každým přidáním nového mechanismu neúměrně narůstá náročnost na správu a zdroje (za předpokladu, že každý fyzický stroj používá jiný mechanismus).
Díky pomoci IETF[1] vznikla skupina *AAA Working Group* s cílem vytvořit architekturu, která by byla řešením výše zmíněných problémů. Výsledkem jejich práce byla architektura *AAA* [8].

[1] Internet Engineering Task Force.

Autentizace

Autentizace je proces ověření identity uživatele (nebo přístroje). Nejčastější formou autentizace je použití jedinečného uživatelského jména (nebo identifikačního čísla) společně s heslem. Nicméně tento způsob autentizace nepatří k těm nejspolehlivějším. Mezi vhodnější mechanismy, jak ověřit uživatelovu identitu, patří například použití jednorázových hesel nebo digitálních certifikátů. Úspěšnou autentizací je tedy vytvořen vztah mezi dvěma unikátními stanicemi, ve kterém si vzájemně důvěřují [8].

Autorizace

Proces autorizace následuje po úspěšném autentizačním procesu. Jde o sady pravidel nebo rozhodovacích šablon, na základě kterých jsou ověřenému uživateli přidělena práva definující způsob, jakým může s poskytnutými službami nakládat, jaké služby může využívat a jaké operace může v systému provádět.
Autorizace může být založená na množinách omezení, které pro uživatele plynou, např. časová rozmezí, kdy může uživatel danou službu využívat, nebo omezení na počet přihlášených zařízení jednoho uživatele.

Účtování

Díky účtování jsme schopni monitorovat využití služeb uživatelem. Může jít o sledování doby, po kterou službu využíval, množství využitého systémového času nebo množství přijatých/odeslaných dat v rámci sezení. Takto získané informace mohou být použity pro různé účely, např. k řízení, fakturaci, využití zdrojů nebo také k plánování. Účtování však může sloužit i k obyčejnému sběru informací o „typu“ uživatele či o jeho identitě (například v závislosti na službách, které uživatel používá), o časovém rozpětí využívání takovýchto služeb apod.

Nyní již k samotnému protokolu RADIUS.

2.2 Protokol RADIUS

Remote Authentication Dial In User Service (*RADIUS*) je síťový protokol sloužící pro centralizované ověřování, autentizaci, autorizaci a účtování (AAA) uživatelů, kteří se přihlašují do sítě, aby mohli využívat služeb, které tato síť poskytuje.

Vzhledem k faktu, že hlavním cílem této práce je implementovat plugin sondy FlowMon, který má být schopen úspěšně detekovat veškerý provoz RADIUS v počítačové síti, je nutné čtenáře seznámit se základní charakteristikou protokolu RADIUS. Kapitola také pojednává o historii, postupu při autentizaci a autorizaci uživatele.

2.2.1 Historie

Jak již bylo řečeno, protokol RADIUS vznikl dříve než vlastní architektura AAA, jejíž principy využívá. První zmínka o protokolu RADIUS byla v RFI[2] od Merit Network roku 1991 pro jejich síť NSFnet. Na tuto skutečnost zareagovala firma Livingston Enterprises a ve spolupráci s Merit Network začal vznikat protokol RADIUS pro jejich PortMaster série Network Access Servers [20]. První standard RADIUS RFC (2058) byl vydán v lednu roku

[2]Request for Information.

1997. K tomu došlo za pomoci IETF, a to díky potřebě standardizovat protokol, který by umožňoval autentizaci, autorizaci a účtování (což jsou základní prvky protokolu RADIUS). Původní standard se roku 2000 dočkal své aktualizace. Jde o RADIUS RFC (2865) [14], který je dosud aktuální. Současně s ním vznikal i standard zabývající se pouze účtováním. Jeho aktuální verzí je RADIUS Accounting RFC (2866) [13] z roku 2000. Společně s těmito standardy vzniklo více dokumentů RFC, které protokol RADIUS rozšiřují a upřesňují (např. RADIUS a IPv6 – RFC 3162 [1]).

2.2.2 Specifikace protokolu RADIUS

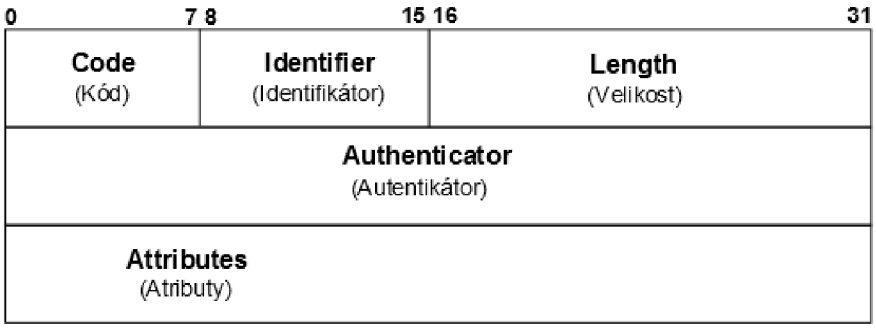
Jde o klient-server model, kde roli klienta zastává *Network Access Server (NAS)*. Tento klient je zodpovědný za předání informací o uživateli konkrétnímu serveru RADIUS a současně musí umět reagovat na odpovědi, které jsou serverem zaslány zpět, případně musí být schopen zobrazit koncovému uživateli informace, které server v odpovědi zaslal.

Serverová část zodpovídá za příjem žádostí na spojení od klienta, autentizaci uživatele a rovněž za zaslání všech nezbytných informací pro poskytnutí služby klientovi, o kterou uživatel žádal. Server RADIUS může vystupovat také jako proxy klient pro jiný server RADIUS.

Komunikace mezi klientem a serverem probíhá pomocí protokolu UDP *(User Datagram Protocol)*. Oficiálně přidělená čísla portů UDP protokolu RADIUS jsou `1812` pro autentizaci uživatele a `1813` pro účtování. Tyto porty nahradily původní porty UDP `1645` a `1646`. Důvodem změny byl konflikt se službami „datametrics“, nicméně z historického hlediska a také kvůli zpětné kompatibilitě se zařízeními, která stále komunikují na starých portech, některé ze serverů RADIUS kontrolují provoz na obou variantách portů UDP.

2.2.3 Formát paketu

Paket RADIUS je zapouzdřen v datové části paketu UDP, přičemž hodnota položky cílový port je nastavena na 1812. V odpovědi jsou hodnoty položek zdrojový a cílový port vzájemně prohozeny.



Obrázek 2.1: Formát RADIUS paketu

Kód

Položka Kód (`Code`) nám jednoznačně určuje, o jaký typ paketu RADIUS se jedná (jde o nezápornou celočíselnou hodnotu). Velikost tohoto pole je jeden oktet (tj. 8 bitů). V případě přijetí paketu s nevalidní položkou `Code`, je paket zahozen a dále se nezpracovává (ve statistikách se pak mohou objevit i tyto pakety s tím, že je u nich poznámka o chybě).
Výčet hodnot, kterých může položka `Code` nabývat, je následující:

- *1 – Access-Request*
- *2 – Access-Accept*
- *3 – Access-Reject*
- *4 – Accounting-Request*
- *5 – Accounting-Response*
- *11 – Access-Challenge*
- *12 – Status-Server* (experimentální)
- *13 – Status-Client* (experimentální)
- *255 – Reserved*

V této práci se budeme zabývat pouze kódy 1, 2, 3, 4, 5 a 11.

Identifikátor

Toto pole následuje po položce `Code`. Jeho velikost je rovněž jeden oktet a pomáhá při identifikaci požadavků a odpovědí. Server RADIUS je schopný detekovat duplicitní žádosti, pokud je zdrojová IP adresa, zdrojový port UDP a identifikátor stejný (v krátkém čase).

Délka

Další v pořadí je položka Délka (`Length`). Její velikost jsou dva oktety (16 bitů). Hodnota této položky určuje délku paketu RADIUS. V ní je započínána i velikost položek `Kód`, `Identifikátor`, `Délka`, `Autentizátor` a `Atributy`. Rozmezí velikosti je 20 až 4096 bitů. Pakety o menší než minimální velikosti jsou zahozeny.

Autentizátor

Autentizátor (`Authenticator`) je položka o velikosti 16 oktetů (tedy 128 bitů). Nejvýznamnější oktet je odeslán jako první. Tato hodnota je použita k ověření odpovědi od serveru RADIUS a je také použita v algoritmu pro skrytí hesla.

Rozlišovány jsou dvě hodnoty. Pro požadavek a pro odpověď. Hodnota pro požadavek (`Request-Authenticator`) je použita v paketech `Authentication-Request` a `Accounting-Request`. Ve skutečnosti jde o náhodně vygenerované číslo o velikosti 16 oktetů, což znesnadňuje případné útoky. RADIUS sice nevytváří žádná opatření proti odposlechu komunikace a zachytávání paketů, nicméně zcela náhodně generované hodnoty ve spojení se silným heslem zaručují, že případné útoky či odposlechy budou nesnadné.
Hodnota pro odpověd (`Response Authenticator`) se používá v paketech `Access-Accept`,

`Access-Reject` a `Access-Challenge`. Hodnota se počítá pomocí jednosměrné hashovací funkce MD5 vygenerované z hodnot `Code, Identifier, Length` a `Request Authenticator` získaných z hlavičky paketu požadavku a atributů pro odpověď společně se sdíleným tajným klíčem. Výsledný vztah je následující:
`ResponseAuth = MD5(Code+ID+Length+RequestAuth+Attributes+Secret)`[3]

2.2.4 Typy paketů

V této části práce blíže představím typy paketů zmíněné v předchozí podkapitole týkající se fází autentizace a autorizace z procesu AAA. Poslední fázi procesu AAA, účtování, je věnována samostatná sekce (2.3) této práce.

Access-Request

Jde o paket s žádostí o přístup k některé ze síťových služeb, který zasílá klient serveru RADIUS. Aby se klient mohl pokusit o úspěšnou autentizaci uživatele, musí být položka `Kód` v hlavičce paketu RADIUS nastavena na hodnotu `1`. Po obdržení platného Access-Request paketu zašle server odpovídající odpověď na základě toho, zda je autentizace uživatele úspěšná či nikoliv. Tělo paketu Access-Request by mělo obsahovat atribut `User-Name` a musí zahrnovat atribut `NAS-IP-Adress` (nebo `NAS-Identifier` případně oba) a také `User-Password` (alternativou je `CHAP-Password` nebo `State`). `User-Password` je skryto pomocí metody založené na *RSA Message Digest Algorithm MD5* [15]. Počet atributů, které musí v požadavku klient zaslat, závisí na konkrétní službě, o kterou žádá.

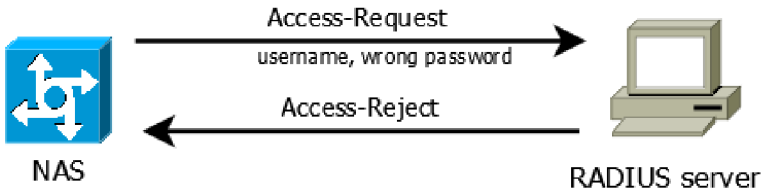
Access-Accept

Access-Accept, neboli žádost přijata, je paket zasílaný serverem RADIUS klientovi v případě úspěšné autentizace a úspěšného přidělení zdrojů, o které klient žádal. Odpověď obsahuje i nezbytné konfigurační údaje pro začátek používání služby. Že jde o úspěšné schválení požadavku, zjistí klient z hodnoty položky kód v hlavičce paketu, která musí obsahovat číselnou hodnotu `2`. Jak již bylo zmíněno výše, při přijetí je odpověď s požadavkem spárována pomocí pole `Identifier` z hlavičky paketu. Atributy, které jsou obsaženy v odpovědi, závisí na službě, o kterou uživatel žádal. Počet atributů v těle paketu může být roven nule, ale může jich být též několik (*nikdy* se však v odpovědi nenachází atribut `User-Password`). Často bývají v těle paketu odpovědi informace popisující službu, o kterou klient žádal.

Access-Reject

Opakem odpovědi Access-Accept je *Access-Reject*, tedy žádost zamítnuta. Tuto odpověď klient získá v případě, že pokus o autorizaci či přidělení žádaných zdrojů neuspěl. Zamítnutí požadavku může klient obdržet i v návaznosti na systémové politiky nebo ve spojení s právy konkrétního uživatele v rámci žádané služby.
Access-reject může také sloužit pro ukončení již běžícího sezení a to v závislosti na přečerpání limitů poskytovaných služeb. Ne všechny systémy, které RADIUS používají, umožňují popsané chování. Označení paketu se zamítnutým přístupem ke službám je určeno hodnotou kódu v hlavičce paketu. Jde o hodnotu s číslem 3. Klient často dostává ve spojení s tímto typem paketu textovou zprávu o stavu/chybě, kterou interpretuje uživateli. Tuto zprávu klient získá z atributu `Reply-Message`.

[3] Znak `+` značí konkatenaci.



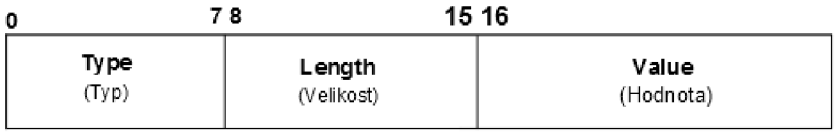
Obrázek 2.2: Ukázka klient/server komunikace (zvoleno chybné heslo)

Access-Challenge

V případě, že server vyžaduje nějaké doplňující informace nebo naopak obdrží informace, které jsou v konfliktu, nebo se jen pokusí snížit riziko podvodného přihlášení. Pak zasílá klientovi v návaznosti na jeho požadavek odpověď `Access-Challenge`, čím uživatele vyzývá k dalšímu kroku. Na tento stav klient reaguje vytvořením nového požadavku, ve kterém se pokusí předat doplňující informace serveru. Paket tohoto typu, ve kterém server žádá dodatečné informace, se může několikrát opakovat. Tento druh výzvy však není podporován všemi klienty a uvedená situace je řešena zasláním `Access-Reject` paketu s upřesňující informací. Hodnota položky kód v hlavičce paketu je `11` a počet atributů zaslaných v těle je závislý na typu požadavku.

2.2.5 Atributy a jejich hodnoty

Doposud jsem hovořil zejména o hlavičce paketu RADIUS. Nyní se již mohu zaměřit na jádro celého paketu – na atributy. Atributy neboli *attribute-value pairs* (AVPs) jsou klíčem ke správnému použití modelu AAA a mohou také sloužit ke konfiguračním účelům. Umožňují sdílet informace mezi klienty, proxy servery a servery RADIUS.



Obrázek 2.3: Formát těla paketu RADIUS: atributu (*attribute-value pairs*)

Postupně uvedu a popíši nejčastěji se vyskytující atributy z běžného provozu. Každý atribut (AVP) se skládá z typu, velikosti a hodnoty. `Typ (Type)` je celočíselná kladná hodnota, kterou lze nalézt v prvním oktetu, a jednoznačně určuje o jaký atribut se jedná. Následující oktet těla paketu zabírá položka `velikost (Length)`, jejíž hodnota udává velikost těla (jde o sumu velikostí položek `typ`, `délka`, `hodnota`). Poslední částí těla paketu je položka nesoucí hodnotu samotného atributu (`value`). Velikost a formát hodnoty se odvíjí v závislosti na konkrétním atributu.

User-Name

Uživatelské jméno jednoznačně označuje uživatele, který žádá o nějakou službu, a proto musí být ověřen. Uživatelské jméno, pokud je k dispozici, by mělo být v požadavku vždy

zadáno. Může se objevit i v odpovědi s úspěšnou autentizací. Typ č. 1 určuje, že jde o atribut uživatelské jméno, a hodnota atributu je ve tvaru textového řetězce.

User-Password

Jde o atribut obsahující informaci o heslu, které si uživatel pro přístup k žádané službě zvolil nebo které mu bylo přiděleno. Heslo je nezbytnou součástí procesu ověření pravosti uživatele a protože jde o citlivou informaci, je heslo šifrováno a musí splňovat základní vlastnosti (minimální délka hesla, speciální znaky, atd.). Přítomností tohoto atributu je rovněž řečeno, že při ověřování bude použit mechanismus PAP [12] namísto CHAP [17]. Typ č. 2, jde o textovou položku, která je pomocí bezpečnostních mechanismů šifrována. Jde o povinný atribut v paketu s požadavkem. Výjimkou je pouze situace, kdy je namísto tohoto atributu použit atribut `CHAP-Password`.

CHAP-Password

Obsahuje-li paket tento atribut, pak se uživatelova totožnost bude ověřovat pomocí mechanismu *PPP Challenge-Handshake Authentication Protocol (CHAP)* [17]. Zároveň s tímto atributem nesmí být v paketu obsažen atribut `User-Password`. Struktura AVP je pro tento atribut lehce odlišná. Oproti trojici `Typ, Velikost, Hodnota` přibývá nově pole `CHAP Ident` o velikosti jednoho oktetu nesoucí hodnotu CHAP identifikátoru. Nová položka je umístěna mezi položkami `Velikost` a `Hodnota`.

NAS-IP-Address

Tento atribut reprezentuje IP adresu zařízení NAS, které žádá jménem počítače klienta o nějakou službu. Setkáme se s ním pouze v požadavku. V případě, když není využito tohoto atributu, je nutné v požadavku uvést NAS-Identifier (použití jednoho z nich je povinné). Jednoznačná identifikace číslem 4. Jeho hodnota je reprezentována jako IP adresa.

NAS-Port

Podobně jako `NAS-IP-Address` reprezentuje IP adresu zařízení NAS, tak tento atribut reprezentuje číslo portu zařízení NAS (nejde však o port ve smyslu portu TCP/UDP). Musí být použit v paketu s požadavkem na službu. Pokud tomu tak není, tak musíme do požadavku začlenit atribut `NAS-Port-Type` (mohou však být v paketu oba dva). Atribut identifikován číslem 5. Jde o číselnou hodnotu datového typu `integer`.

Service-Type

Jak již napovídá název, jde o atribut, který určuje o jaký typ služby uživatel ve svém požadavku žádá, případně jaký typ služby mu má být poskytnut. Nejde o povinný atribut a může se objevit jak v požadavku, tak v úspěšné odpovědi. Jde o typ s číslem 6 a položka *value* (hodnota) může nabývat čísel v rozmezí 1-11, přičemž každé z nich má svoji vypovídací hodnotu. Nabývané hodnoty jsou například:

- *1 Login* – uživatel může být připojen ke službě
- *2 Framed* – pro připojení se použije tzv. framed protokol jako je PPP nebo SLIP

- *3 Callback Login* – současné spojení je ukončeno a následně zavolá uživatele zpět. Obdoba Login.
- *4 Callback Framed* – podobně jako v předchozím případě, odpojení a následné připojení, tentokrát ale pomocí framed protokolu
- *5 Outbound* – uživatel může využít odchozí služby/zařízení
- *6 Administrative* – navýšení oprávnění uživatele pro administraci

Úplný výčet hodnot, kterých může tato položka nabývat, společně s jejich popisem lze naleznout v RFC 2865 [14].

Reply-Message

Atribut byl již zmíněn v souvislosti s typy paketů RADIUS. Je nositelem textu, který by měl klient zobrazit uživateli. Objevuje se v paketech `Access-Accept` jako normální zpráva, v `Access-Reject` jako chybová zpráva, případně v `Access-Challenge` jako dialogová zpráva. Číselný kód pro `Reply-Message` je 18 a hodnota je, jak již bylo zmíněno, textová.

Called-Station-ID

Hodnota tohoto atributu pomáhá zařízení NAS sdělit, jaké číslo uživatel vytočil, aby získal přístup k jeho službě. Při použití vytáčeného čísla jakožto identifikace služby (DNIS[4]), může NAS tuto informaci využít k ověření umístění. Atribut může sloužit také pro identifikaci proxy serverů RADIUS při předávání žádosti cílovému serveru RADIUS. Typ: 30, hodnota: textový řetězec.

Calling-Station-ID

Lze na něj nahlížet jako na telefonní číslo volajícího. Opět jde o atribut, který usnadňuje identifikaci uživatele a to pomocí automatické identifikace čísla (ANI[5]). Typ: 31, hodnota: textový řetězec.

NAS-Identifier

Jedná se o identifikaci zařízení NAS, které zodpovídá za vytváření paketů s požadavky. Hodnotou tohoto atributu je často tzv. `Fully Qualified Domain Name (FQDN)`, což je označení pro plně specifikované doménové jméno počítače. FQDN se často používá pro zredukování opakujících se identifikátorů NAS, což vede k lepší orientaci mezi nimi. Paket s požadavkem na službu musí obsahovat buď tento atribut, anebo atribut `NAS-IP-Address`. Typ: 32, hodnota: textový řetězec.

Proxy-State

Již bylo zmíněno, že server RADIUS může plnit roli proxy serveru. Právě proto máme k dispozici tento atribut. Vždy, když potřebuje takový proxy server uložit informaci o požadavku (např. doménové jméno), je použit tento atribut. Zásady pro použití tohoto atributu jsou upřesněny v příslušném RFC. Typ: 33, hodnota: textový řetězec.

[4] Dialed Number Identification.
[5] Automatic Number Identification.

NAS-Port-Type

Výše jsem popisoval atribut s názvem `NAS-Port`. Sám název `NAS-Port-Type` napovídá, jak se tento atribut oproti `NAS-Port` liší. Hodnoty, kterých tato položka nabývá, jsou výčtem celočíselných hodnot datového typu `integer` a upřesňují způsob, jakým mezi sebou klient a server komunikují. Typ: 61, hodnota: celočíselná hodnota reprezentující daný typ (rozmezí 0-19). Jako příklad uvedu pouze některé typy:

- *0 Asynchronous* – nejběžněji používaný
- *1 Synchronous*
- *2 ISDN Synchronous*
- *9 X.75*
- *17 Cable*
- *19 Wireless – IEEE 802.11*

Základní sada atributů byla rozšířena, protože technologie RADIUS zareagovala na příchod IPv6. Některé z doplněných atributů nyní krátce představím (čerpáno z RFC 3162 [1] a RFC 6911 [6]).

- *NAS-IPv6-Address* – Obsahuje IPv6 adresu NAS. Objevuje se pouze v síti IPv6 v paketu `Access-Request`. Typ nabývá jednoznačné hodnoty `95`.
- *Login-IPv6-Host* – Atribut označující systém, se kterým se uživatel pokouší spojit (v případě, že paket obsahuje atribut `Login-Service`). Není povinný a může se vyskytnout jak v `Access-Accept`, tak `Access-Request` paketu (zde může sloužit jako nápověda pro NAS, která říká, k jakému hostiteli by se uživatel raději připojil). Položka Typ nese hodnotu `98`.
- *DNS-Server-IPv6-Address* – Tento atribut obsahuje IPv6 adresu DNS serveru[6]. V paketu se může objevit vícekrát nebo ani jednou. Může být součástí `Access-Accept` i `Access-Request` paketu (znovu jde pouze o doporučení – konkrétně od NAS pro server RADIUS).

Podrobný popis všech atributů je k dispozici v příslušných specifikacích RFC.

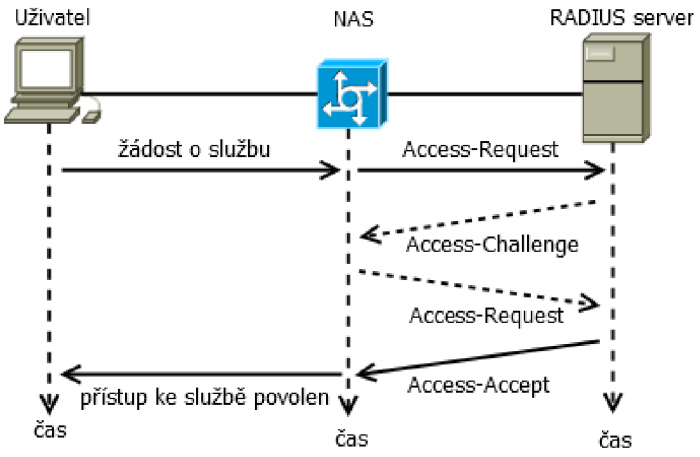
2.2.6 Postup autentizace a autorizace

Typický scénář, jak postup autentizace probíhá, je následující. Nejprve uživatel zažádá o přístup k určitým sdíleným informacím či službám. Nečiní tak však osobně, ale prostřednictvím klienta (nejčastěji NAS). Posléze klient zformuluje požadavek (`Access-Request`) a zašle jej serveru RADIUS, od kterého očekává odpověď (`Access-Accept`, `Access-Reject`, případně `Access-Challenge`). Paket `Access-Request` bude obsahovat minimálně tyto položky: uživatelské jméno (`User-Name`), šifrované heslo (`User-Password`), IP adresu zařízení (`NAS-IP-Address`) a port (`NAS-Port`).

Ve chvíli, kdy server přijme od klienta požadavek na přístup (paket `Access-Request`), začne vyhledávat ve své databázi uživatele se zaslaným uživatelským jménem. Pokud tohoto uživatele nenalezne, je klientovi neprodleně zaslána zpráva o zamítnutí přístupu ke

[6] DHCPv6 umožňuje konfiguraci hostitele s IPv6 adresou DNS serveru [6].

službě (`Access-Reject`). Existuje i možnost, že je uživateli nabídnut profil návštěvníka, pak může některé ze služeb využívat. Nicméně toto chování se odvíjí od bezpečnostních politik daného systému, do kterého chce uživatel získat přístup. Zpráva o zamítnutém přístupu (`Access-Reject`) může být doplněna o textovou zprávu s důvodem, proč byla žádost zamítnuta. Jestliže server RADIUS uživatele v databázi nalezne, přistoupí se k dalšímu kroku, a to ke kroku ověření uživatelovy identity. Server RADIUS podporuje několik metod, jak ověřit uživatelovu totožnost. Patří mezi ně například PAP, CHAP či EAP[7]. Pokud vše odpovídá a vše proběhlo úspěšně, zasílá server RADIUS odpověď s povolením přístupu (`Access-Accept`), která obsahuje nezbytné informace pro začátek používání žádané služby.



Obrázek 2.4: Úspěšná autentizace a autorizace uživatele

2.3 Účtování RADIUS

Účtování (accounting) je rozšíření standardního protokolu RADIUS umožňující sběr informací týkajících se používání prostředků, o které uživatel žádal. Mezi typické použití patří distribuování internetového spojení koncovým uživatelům. Klíčové vlastnosti účtování RADIUS jsou:

- využití *klient/server* modelu
- zabezpečená komunikace mezi zařízeními
- rozšiřitelnost protokolu

Klient (v našem případě NAS) odpovídá za doručení informací o účtování určenému účtovacímu serveru RADIUS. Server požadavky přijímá a odpovídá zpět klientům. Stejně jako v případě serveru RADIUS pro autentizaci a autorizaci může i server RADIUS, zodpovědný za účtování, vystupovat jako proxy klient pro jiné účtovací servery [13]. Komunikace mezi klientem a serverem je zabezpečena pomocí sdíleného hesla (*shared secret*), které se po síti nezasílá. Poslední vlastností je rozšiřitelnost protokolu. Ta spočívá v možnosti přidání nové hodnoty atributů bez toho, aniž by byla porušena struktura protokolu. Proces účtování může začít až po úspěšném ověření uživatele a přidělení zdrojů.

[7] Extensible Authentication Protocol.

2.3.1 Princip činnosti

Veškerá komunikace ze strany klienta spočívá v zasílání paketu `Accounting-Request`[8]. Po správné konfiguraci klienta pro používání účtování RADIUS, vytvoří klient paket `Accounting Start`, což je zvláštní případ paketu `Accounting-Request`, a zašle jej účtovacímu RADIUS serveru (dále jen server). Tento jej následně příjme a pošle klientovi potvrzení o přijetí (`Accounting-Response`). Účtovací proces běží do doby, dokud server neobdrží požadavek na ukončení služby. Jde o paket `Accounting Stop`. Pro komunikaci mezi klientem a serverem je použit protokol UDP a komunikace probíhá na portu UDP `1813`.

2.3.2 Formát paketu

Paket účtování RADIUS má stejnou strukturu jako paket pro autorizaci a autentizaci, který je znázorněn obrázkem č. 2.4. Protokol účtování RADIUS používá pro komunikaci dva typy paketu:

- `Accounting-Request`
- `Accounting-Response`

Accounting-Request

Jde o paket, který zasílá klient serveru. Položka `Code` nese hodnotu `4`. Server rozpozná, že jde o paket s požadavkem na účtování, a potvrdí klientovi jeho přijetí. Paket může obsahovat atributy jak standardního protokolu RADIUS, tak protokolu účtování RADIUS. Výjimku tvoří atributy `User-Password`, `CHAP-Password`, `Reply-Message` a `State`. Tyto atributy paket pro účtování nesmí obsahovat.

Accounting-Response

Paket `Accounting-Response` je zasílaný serverem klientovi ve chvíli přijetí paketu `Accounting-Request`. Jeho účelem je pouze informovat klienta o správném doručení požadavku (při začátku účtování a na jeho konci). Pouze zřídka obsahuje nějaké atributy.

2.3.3 Atributy účtovacích paketů

V souvislosti s účtováním popíši pouze některé z atributů, o které je protokol RADIUS prostřednictvím účtování RADIUS rozšířen. Kompletní popis všech atributů týkajících se účtování lze nalézt v RFC 2866 [13].

Acct-Status-Type

Atribut je obsažen v paketu `Accounting-Request` (typ č. `40`). Ohlašuje začátek či konec účtování služeb konkrétnímu uživateli. Hodnoty, kterých může tento atribut nabývat, jsou:

- *1 – Start*
- *2 – Stop*
- *3 – Interim–Update*

[8] Požadavek na účtování.

- *7 – Accounting–On*
- *8 – Accounting–Off*
- *9–15 – Rezervováno*

Acct-Authentic

Tento atribut je reprezentovaný číslem `45`. Jde o volitelný atribut říkající, jakou metodou byla uživatelova identita ověřena. Rozlišujeme tyto způsoby:

- *1 – RADIUS* – Ověření proběhlo pomocí RADIUS protokolu.
- *2 – Local* – NAS sám ověřil uživatelovu identitu.
- *3 – Remote* – Uživatel byl ověřen pomocí vzdáleného autentizačního protokolu.

Acct-Session-Time

Jde o volitelný atribut, který je reprezentovaný číselnou hodnotou `46`, a může se objevit v `Accounting-Request` paketu, ale pouze v případě, že je přítomen také atribut `Acc-Status-Type` nastavený na hodnotu `Stop`. Tento atribut udává dobu (v sekundách), po kterou byl uživatel ke službě připojen.

Acct-Link-Count

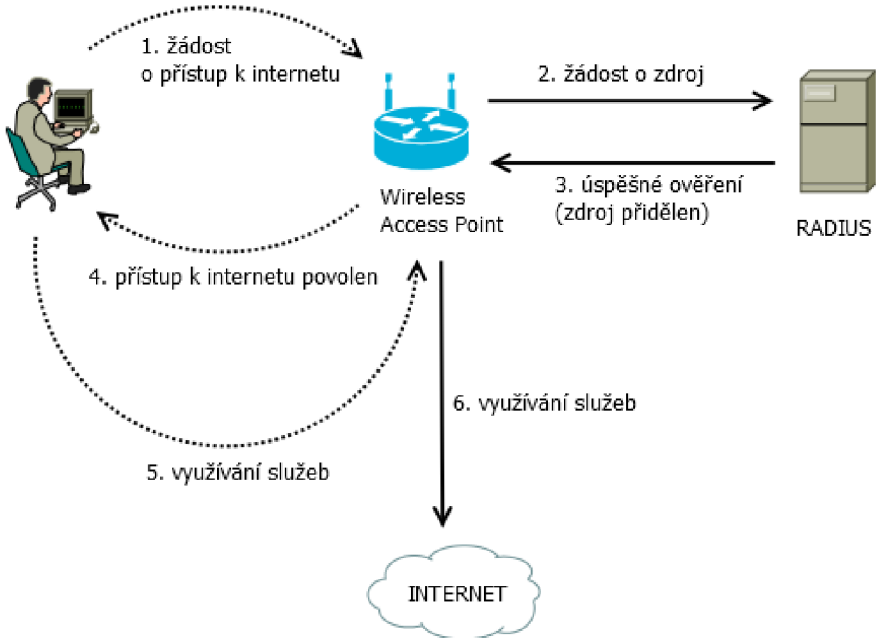
Tento atribut pak udává počet aktuálních spojení v případě vícenásobného připojení ke službě. Jedná se o volitelný atribut identifikovaný `typem` č. 51 obsažený v `Accounting-Request` paketu. Zpravidla se používá k usnadnění činnosti účtovacího serveru.

2.3.4 Příklad použití protokolu RADIUS

Mezi nejčastější typy služeb patří zajisté poskytování připojení k internetu. Může se jednat například o DSL nebo bezdrátové připojení k síti s přístupem na internet. Uživatel, který se chce k uvedené službě dostat, je při pokusu o připojení vyzván k zadání informací jako jsou uživatelské jméno a heslo. Další činnost je již v režii síťových prvků, klientů NAS a serverů RADIUS. Podaří-li se serveru RADIUS proces ověření totožnosti, získá uživatel přístup k internetu. Tento typ služby bývá zpravidla zpoplatněn. S poplatky je spojeno i zajištění kvality poskytované služby. Informace týkající se účtování může poskytovatel služby (správce sítě) snadno zjistit právě díky účtování RADIUS. Případů, kdy se můžeme s ověřováním pomocí protokolu RADIUS setkat, je mnoho a koncovým uživatelům by měla být tato komunikace skryta.

2.4 RADIUS a bezpečnost

Útoky na počítačové sítě jsou běžnou součástí dnešního světa. Útočníci se často chtějí dostat k informacím, ke kterým má přístup jen určitá skupina lidí, u níž je zaručeno, že informace nezneužije. RADIUS je protokolem pro ověření totožnosti uživatelů, a z tohoto důvodu neunikl pozornosti útočníkům. Útoky na RADIUS směřují z bezdrátové sítě. Mezi používané útoky patří *útok na identifikační údaje hrubou silou slovníkové útoky na sdílené tajemství* nebo také *podvržení paketů*. Tato oblast však není náplní mé práce, proto se jí nebudu více zabývat.



Obrázek 2.5: Zjednodušená demonstrace využití RADIUS

2.5 Budoucnost protokolu RADIUS

Lze očekávat, že postupem času vznikne snaha nahradit RADIUS protokolem Diameter. Jde o protokol, který začal vznikat krátce po přepracování protokolu RADIUS. Cílem bylo vytvořit „čistou“ verzi protokolu RADIUS a pojmenovat ji RADIUS v2. Tento název IETF nepovolila (RADIUS v1 nebyl odsouhlasen). Nakonec pro nový protokol zvoleno jméno Diameter[9]. Protokol Diameter je popsán v RFC 6733 [2].
Budoucnost protokolu RADIUS je ale ovlivněna vývojem nových rozšíření, na kterých se stále pracuje. RADIUS je tedy stále živý a přinejmenším v blízké budoucnosti bude stále patřit mezi nejvíce využívaný protokol umožňující autorizaci, autentizaci a účtování.

2.6 Shrnutí

Tato kapitola popisuje základní vlastnosti protokolu RADIUS, důvody jeho vzniku a také nastiňuje jeho budoucnost. Část kapitoly byla věnována vysvětlení modelu AAA a okrajově byla zmíněna i bezpečnost protokolu. Značný důraz byl kladen na popis atributů RADIUS, ze kterých jsou získávány potřebné informace, jež rozšiřují záznamy IPFIX o tocích. Nutno podotknout, že atributy, které zmiňuji, jsou pouze výběrem z velké skupiny atributů nabízených tímto protokolem. Tato část práce tvoří společně s následující kapitolou *„základní kameny“* pro správný návrh pluginu pro FlowMon sondu, díky kterému bude možno detekovat provoz RADIUS na síti.

[9]Diameter (tj. průměr) je oproti RADIUSu (tj. poloměr) dvakrát větší [20]. Autoři se nepřímo snaží naznačit, že je i dvakrát lepší.

Kapitola 3

Monitorování sítě pomocí IPFIX

Monitorování počítačových sítí na základě IP toků je v dnešní době zcela běžné. Jedná se o nezbytnou součást každé větší počítačové sítě a umožňuje síťovým administrátorům provádět podrobnou analýzu provozu na síti v reálném čase. Nejznámější technologií pro monitorování sítě na základě IP toku je bezesporu technologie NetFlow.

Technologie FlowMon, pro jejíž sondu je plugin umožňující detekci komunikace RADIUS v rámci této bakalářské práce vytvořen, nabízí kompletní řešení pro monitorování sítí na základě IP toků právě díky technologii NetFlow/IPFIX [9]. Dříve než přistoupím k samotnému protokolu IPFIX, zmíním se krátce o technologii NetFlow, která s IPFIX úzce souvisí.

3.1 Úvod do technologie NetFlow/IPFIX

Protokol NetFlow byl vyvinut společností *Cisco Systems* a původně šlo pouze o rozšíření k jejich směrovacím zařízením. Termín NetFlow obecně označuje celou oblast sběru, monitorování a vyhodnocování dat v počítačové síti [7]. Tok (flow) je základním prvkem NetFlow.
Obecná definice toku: Tok je definován jako jednosměrná posloupnost paketů majících společnou vlastnost a procházejících bodem pozorování za určitý časový interval. Všechny pakety patřící do jednoho toku mají společné vlastnosti odvozené z obsahu paketu [3].
V případě NetFlow se jedná o následující vlastnosti:

- Zdrojová a cílová IP adresa.
- Zdrojový a cílový port.
- Typ protokolu L3 (TCP, UDP, ICMP, IGMP).
- Název rozhraní (interface).
- Typ služby (Type od Service - ToS)

Mezi základní prvky systému NetFlow patří:

- **Exportér** - softwarové (sonda) nebo hardwarové (router) zařízení zajišťující sběr informací o tocích. Po ukončení toku zasílají kolektoru záznam NetFlow.
- **Kolektor** - zařízení pro sběr dat z exportéru (může jich být i více), která se ukládají na disk

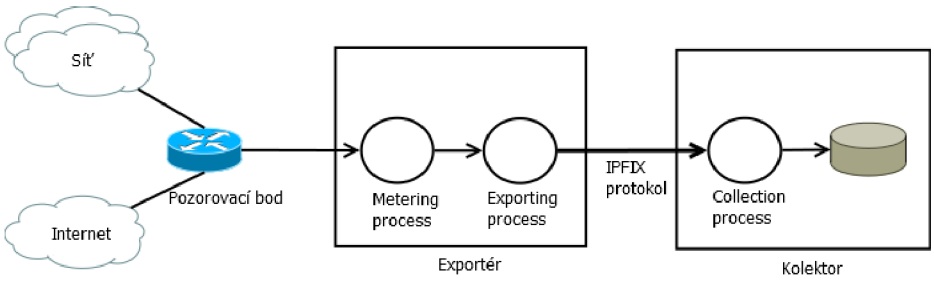
- **Komunikační protokol NetFlow** - protokol pro odesílaní dat mezi exportérem a kolektorem
- **Nástroje pro zobrazení nasbíraných dat** - např. zobrazení statistik, grafická reprezentace využití sítě apod.

V současnosti existuje několik verzí protokolu NetFlow. První používanou verzí byla NetFlow verze 5. NetFlow verze 9 je prozatím poslední inovace Cisco IOS NetFlow. Tato verze přináší mnoho novinek a vylepšení, jednou z nich je i její flexibilita (možnost přidávání nových položek bez změny exportního formátu). Z Cisco NetFlow v9 vychází i protokol IPFIX, jehož formát je použit v technologii FlowMon.

Organizace IETF založila pracovní skupinu s cílem vytvořit univerzální standard pro export IP toku s důrazem na zlepšení interoperability v oblasti měření provozu sítě na úrovni toků. Výsledkem jejich snažení byl standard RFC 5101 - protokol IPFIX. IPFIX neboli *IP Flow Information eXport* je jednosměrný, transportně nezávislý protokol s flexibilní reprezentací dat a s informačním modelem zahrnujícím většinu informací z vrstev L3 (transportní vrstva) a L4 (síťová vrstva) pro správu sítě [18]. Standard IPFIX říká, jakým způsobem a v jakém formátu jsou informace o IP tocích přenášeny z exportéru na kolektor. Tok hraje tedy i v případě IPFIX klíčovou roli. To však není vzhledem k faktu, že IPFIX vychází z Cisco NetFlow v9, nijak překvapivé. Do jednoho toku patří pakety IPFIX shodující se v následující „pětici" – `zdrojová IP adresa, cílová IP adresa, zdrojový port, cílový port` a `typ protokolu L3` (i tu je zřejmá shoda s NetFlow v9).

3.2 Architektura IPFIX

Proces měření (*Metering Process*) vytváří záznamy o toku (Flow Record) z paketů, které se nachází v pozorovacím bodě (*Observation Point*). Proces exportu (*Exporting Process*) zasílá procesu sběru (*Collecting process*) IP toky získané z procesu měření za pomocí protokolu IPFIX. Celý proces ilustruje obrázek č. 3.1.

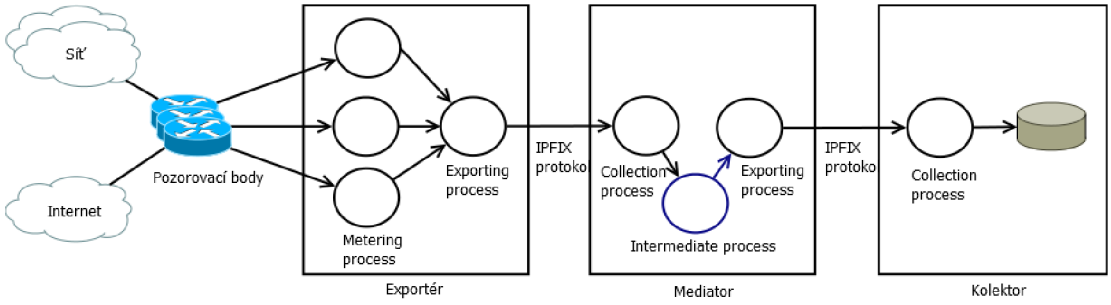


Obrázek 3.1: Jednoduchá architektura

Proces měření a exportu zastupuje prvek exportér, proces sběru je zastoupen kolektorem. Jde o stejné prvky jako v případě NetFlow v9, avšak ke komunikaci mezi nimi je použit protokol IPFIX. Mezi exportérem a kolektorem platí vztah 1:N, tedy exportér může zaslat informace o IP toku několika kolektorům. Výše popsanou trojici později rozšířil meziproces (*Intermediate Process*) umožnující modifikaci dat jako je např. anonymizace, agregace či korelace toků. Tuto část zastupuje tzv. *mediator*[1] a jeho použití znázorňuje obrázek č. 3.2.

Více informací týkajících se architektury pro IPFIX lze nalézt v dokumentu RFC 5470 [16].

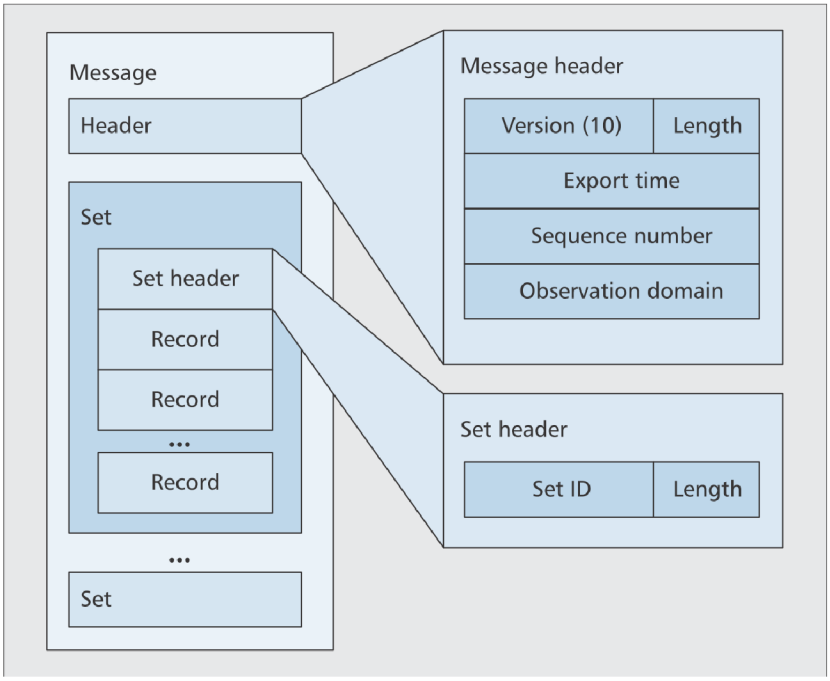
[1]Mediator zajišťuje sběr, transformaci a opětovné zaslání IPFIX zpráv (více viz. RFC 7119 [5] a RFC 6183 [11]).



Obrázek 3.2: Architekrura s použitím prvku Mediator

3.3 Formát protokolu IPFIX

Informace zasílané z exportéru na kolektor pomocí protokolu IPFIX jsou obsaženy ve zprávách IPFIX. Zpráva, která je zapouzdřená v transportní vrstvě, je základem tohoto protokolu. Skládá se z *hlavičky* (*message header*) a *množiny dat* (*set*). Může se jednat o sadu šablon (*template set*) sloužící pro záznamy, nebo datovou sadu (*data set*), která obsahuje datové záznamy. Nutno podotknout, že každá ze zmíněných sad má svoji vlastní hlavičku. Zmíněné sady mohou být ve zprávě obsaženy vícekrát, avšak nemusí být přítomny ani jednou [4]. Formát IPFIX zprávy je zachycen obrázkem č. 3.3. Podrobný popis je dostupný



Obrázek 3.3: Formát IPFIX zprávy [18]

v RFC 7011 [4].

3.4 Budoucnost protokolu IPFIX

IPFIX je poměrně nový protokol a již od jeho vzniku je hojně využíván. Protokol se stal velmi oblíbeným a lze předpokládat další růst jeho používání. Tomu nasvědčuje i stále rostoucí množství produktů dostupných na trhu, které tuto technologii využívají či podporují. IPFIX Working Group i skupina IETF i nadále pracuje na rozšíření použitelnosti protokolu.

3.5 Shrnutí

Cílem této práce ani kapitoly není podrobný popis technologie IPFIX či metod pro monitorování síťového provozu na úrovni IP toků. Účelem této kapitoly je přiblížit čtenáři technologii, která je použita v implementaci pluginu pro sondu FlowMon, a tím nastínit princip vytváření záznamů o IP tocích a způsob ukládání informací, o něž jsou záznamy rozšířeny. Příklad zprávy IPFIX s konkrétními hodnotami získané z komunikace RADIUS je uveden v následující kapitole. Mimo jiné v ní lze nalézt informace týkající se postupu při návrhu a implementaci pluginu.

Kapitola 4

Návrh a implementace pluginu

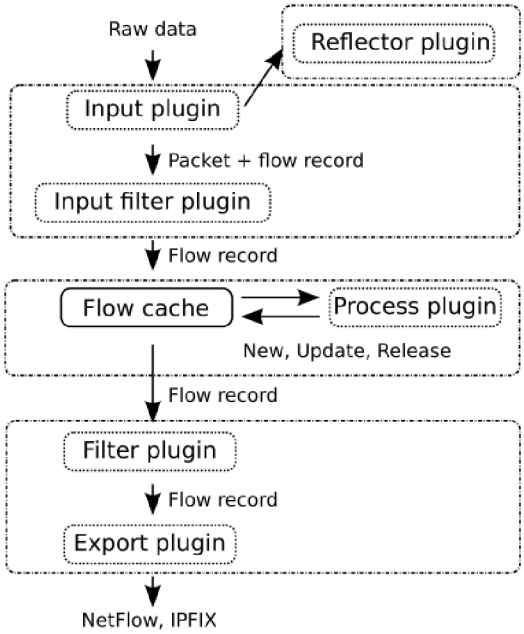
Dalším z hlavních předpokladů správného návrhu a následné implementace pluginu je dobré seznámení se s vlastní sondou FlowMon, které se věnuje úvodní část této kapitoly. Postupně zde popíši vlastnosti a architekturu sondy FlowMon a dále pak postup při návrhu a implementaci pluginu pro monitorování provozu RADIUS na bázi IP toků.

4.1 Popis a architektura sondy FlowMon

Sonda FlowMon patří do portfolia produktů FlowMon společnosti INVEA-TECH nabízející kompletní řešení pro monitorování sítí na základě IP toků (za pomocí technologie NetFlow/IPFIX). Celé portfolio se skládá z výkonných autonomních sond FlowMon, kolektorů FlowMon a rozšiřujících pluginů FlowMon.

Sonda FlowMon (exportér) je zařízení monitorující provoz v počítačové síti. Z něj vytváří statistiky (v podobě IP toků), které zasílá kolektoru FlowMon k uložení a další analýze. Díky podpoře flexibilních formátů NetFlow verze 9 a IPFIX umožňuje sonda vybrat informace, které se mají monitorovat a exportovat.
Klíčové vlastnosti sondy jsou [10]:

- Výkonná autonomní NetFlow v5/v9, IPFIX sonda
- Podpora pro 10 Mb/s až 100 Gb/s Ethernet
- Zpracování dat bez ztráty paketů
- Podpora pro IPv4, IPv6, MAC, VLAN a MPLS
- HTTP a VoIP analýza, detekce aplikací (NBAR2)
- Dostupná jako fyzické nebo virtuální zařízení
- Jednoduchá správa přes webové rozhraní
- Integrovaný kolektor pro zobrazení a analýzu dat
- Plně kompatibilní s NetFlow kolektory třetích stran
- Zpracování až 16.000.000 toků současně



Obrázek 4.1: Architektura exportéru (převzato z [19])

Z obrázku 4.1 popisující architekturu FlowMon exportéru je patrné, že je složen ze tří částí – *vstupní* (input), *paměti pro tok* (flow cache) a *exportní* (export) části [19].

Vstupní část zpracovává data ze vstupu a vytváří z nich záznamy o toku (flow record) pro každý z paketů. *Paměť pro tok* obsahuje o každém z toků jeden záznam. Pro nově příchozí pakety se vytváří nové záznamy o toku, ale to pouze v případě, že nepatří do již existujícího toku. Pak je záznam aktualizován. Další činností paměti pro tok je periodická kontrola aktivních a neaktivních časovačů v záznamech toků. Je-li časovač neaktivní (dojde k vypršení časovače), pak je záznam předán exportní části, která vytvoří z předaných informací zprávu NetFlow nebo v našem případě IPFIX, kterou „exportuje“ na kolektor.

Typy pluginů

Sonda FlowMon „běží“ nad distribucí operačního systému Linux CentOS. Exportér je program implementovaný v jazyce C, jehož činnost lze ovlivňovat pluginy rozšiřujícími jeho funkcionalitu a vlastnosti. Tvorbu pluginů podporuje společnost INVEA-TECH svým komunitním programem. Základní typy pluginů, které lze pro sondu FlowMon vyvíjet jsou:

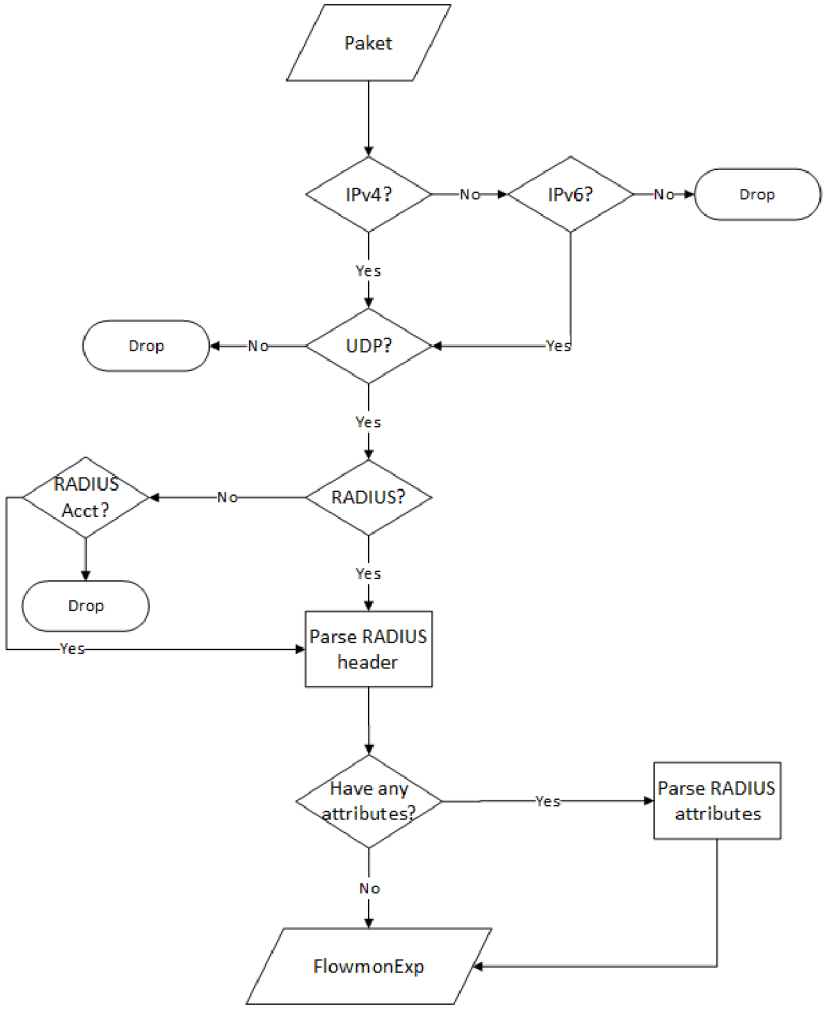
- **Vstupní plugin:** Zachycuje pakety ze specifikovaného síťového rozhraní či ze souborů typu *PCAP*. Zachycené pakety zde lze procházet a získávat z nich informace, které rozšiřují základní záznam toku.
- **Procesní plugin:** Umožňuje pracovat se záznamy toku. Jednotlivé záznamy lze sdružovat, případně dodatečně rozšiřovat. Pomocí něj je možné detekovat různé útoky v reálném čase.

- **Filtrovací plugin:** Slouží pro filtrování expirovaných toků. Na základě definovaných pravidel je tok předán k exportu či zahozen.
- **Exportní plugin:** Jedná se o plugin vytvářející záznamy IPFIX z předaných dat. Tyto záznamy se zasílají na kolektor, případně mohou být tisknuty na obrazovku.

Pro tvorbu pluginů jsou k dispozici hlavičkové soubory s deklarací základních funkcí. Každý typ pluginu má své specifické konstrukce umožňující jeho další vývoj.

4.2 Návrh systému monitorující provoz RADIUS

Detekce a zpracování komunikace RADIUS na úrovni IP toku v počítačové síti je hlavním cílem bakalářské práce. Tato sekce popisuje postup při návrhu systému, který má být schopen RADIUS úspěšně detekovat, získat užitečné informace a o ně poté rozšířit záznam IPFIX.



Obrázek 4.2: Návrh na zpracování příchozích paketů vstupním pluginem

Diagram č. 4.2 demonstruje kroky při zpracovávání nově příchozího paketu. Plugin má být schopen zpracovávat pakety typu IPv4 i IPv6. Princip zpracování paketu RADIUS je pro

IPv4 i IPv6 stejný. Takřka všechny atributy, které se mohou objevit ve verzi 4, mohou být obsaženy i v paketu verze 6. Předtím, než bude příchozí paket dále zpracováván, musí dojít k ověření, že se skutečně jedná o protokol RADIUS (případně účtování RADIUS). Zpracované informace jsou dále předány exportnímu pluginu FlowMon IPFIX. Veškerý provoz je zpracováván vstupním (input) pluginem. Při jeho implementaci jsem vycházel z výše zmíněného návrhu. Jiné typy pluginů není třeba vytvářet.

4.3 Popis implementace pluginu

Jak již bylo řečeno, zpracování příchozích paketů RADIUS protokolu se odehrává ve vstupním pluginu pro FlowMon sondu. Tato část bakalářské práce seznamuje s postupem při jeho vývoji. Před začátkem implementace vlastního pluginu bylo třeba zjistit, jaké zabudované funkce FlowMon exportér nabízí. Každý z pluginů musí dodržovat základní strukturu. Pro vstupní plugin s variantou `PLUGIN_INPUT_GET_FLOW` je struktura následující:

- `PLUGIN_INPUT_DESC` – Zde se definuje velikost, o kterou budu záznam o toku rozšiřovat. Zároveň by tato funkce měla obsahovat základní popis pluginu společně s nápovědou k jeho používání.
- `PLUGIN_INPUT_INIT` – Tato funkce má na starosti následující operace: zpracování parametrů při spuštění z příkazové řádky, inicializaci pluginu společně s nastavením getterů[1] a alokaci privátní struktury.
- `PLUGIN_INPUT_GET_FLOW` – Vyčítání a zpracování dat do záznamu o toku. Výstupní záznam o toku je prozatím „fragment“.
- `PLUGIN_INPUT_SHUTDOWN` – Tato část není povinná. Lze zde definovat operace, které mají být při ukončení pluginu provedeny.

Společně s těmito položkami je třeba definovat *privátní struktury*, které plugin bude při své činnosti využívat, definovat *gettery* a také navrhnout *strukturu rozšiřující vlastní záznam o toku*. Nyní se již budu věnovat vlastní implementaci.

Prvním krokem při implementaci byl návrh struktury s položkami, o které se budou záznamy o toku rozšiřovat. Původní návrh obsahoval veškeré informace získané z hlavičky RADIUS a většinu atributů, které protokol poskytuje. Nicméně mnoho hodnot má v RADIUSu nízkou vypovídací hodnotu, což byl také jeden z důvodů značného zredukování atributů, o které ve výsledku základní záznam o toku rozšiřuji. Jiné z atributů sice nesou zajímavé informace, avšak v běžném použití se s nimi v paketech RADIUS příliš nesetkáme. Obrázek 4.3 zachycuje část struktury s položkami získanými z paketu RADIUS. Tato struktura je poté použita k rozšíření záznamů IPFIX. Téměř všechny atributy, jejichž hodnoty uchovávám, již byly popsány v kapitole pojednávající o protokolu RADIUS. Plugin umožňuje zpracovávat reálný provoz jak na určeném síťovém rozhraní, tak ze specifikovaného souboru typu PCAP. Tato vlastnost je implementována v části `plugin_input_init`. Mimo zpracování parametrů příkazové řádky má tato funkce na starost inicializaci getterů. Dochází zde také k alokaci potřebného paměťového prostoru privátní struktury, která v tomto případě obsahuje pouze položky s případným souborem PCAP, hodnotu offsetu dat RADIUS a příznak určující, zda se bude zpracovávat síťový tok z rozhraní nebo ze souboru.

[1] Gettery slouží pro definici množiny položek, s kterou pak pracuje export.

```
typedef  struct {
   uint8_t code;
   uint8_t identif;
   uint16_t length;
   uint8_t validity_map;
   struct {
     unsigned char username[STR_DATA_LEN];
     uint32_t nas_ip_addr;
     uint32_t nas_port;
     unsigned char service_type[STR_DATA_LEN];
     unsigned char framed_proto[STR_DATA_LEN];
     uint32_t framed_ip_addr;
     uint32_t framed_mtu;
     uint32_t login_ip_host;
     ...
   }avp;
   struct {
     ipv6_addr_t nas_ipv6_addr;
     ipv6_addr_t login_ipv6_host;
     ipv6_addr_t dns_server_ipv6addr;
   }avpIPv6;
}radius_record_t;
```

Obrázek 4.3: Ukázka struktury záznamu

Po úspěšné inicializační části se běh programu přesouvá k funkci `plugin_input_get_flow`. Zde probíhá vlastní zpracování dat a jsou také nastaveny vlastnosti flow záznamu. Výsledkem je přidání rozšířeného záznamu o toku do flow cache FlowMon exportéru (v podobě MD5 hash). Touto funkcí tedy začíná vlastní zpracování odchycených dat.

Získaná data jsou předána k analýze funkci `parse_eth`. Pomocí ní se analyzuje hlavička Ethernet a doplňuje se záznam toku o informace získané z této části. Vlastní analýza probíhá dle schématu č. 4.2. Na základě získaných informací z hlavičky paketu ověřím, o jakou verzi IP protokolu jde (IPv4 či IPv6). Zda jde skutečně o komunikaci RADIUS kontroluji za pomoci typu protokolu a hodnoty zdrojového a cílového portu. Musí se jednat o komunikaci `UDP` na portu `1812` (pro autorizaci a autentizaci) nebo na portu `1813` (pro účtování). Původní porty pro RADIUS kontrolovány nejsou[2]. Jakmile dojde k ověření těchto položek, lze předpokládat, že se skutečně jedná o komunikaci RADIUS, a zpracování je předáno další funkci – `parse_radius_hdr`.

Jak již napovídá samotné pojmenovaní, jedná se o funkci, která má za úkol zpracovat informace obsažené v hlavičce RADIUS a rozšířit o tyto informace záznam o toku. To však pouze v případě, že je číselná hodnota položky `code` v hlavičce paketu `1-5`, případně `11`. Žádné jiné kódy nejsou podporovány. V okamžiku, kdy paket touto kontrolou úspěšně projde, proběhne kontrola, zda jsou přítomny některé z atributů. V kladném případě je volána funkce, která podporované atributy dále zpracuje. V případě opačném, jsou uloženy pouze položky získané z hlavičky paketu. O zpracování atributů paketů RADIUS se v tomto pluginu stará funkce `parse_radius`. Zde postupně proběhne detekce na přítomnost atributů

[2]Podpora starých portů (1645 a 1646) u serverů RADIUS je v dnešní době spíše z historického hlediska.

týkajících se autorizace, autentizace a účtování, a to i pro IP verze 6. Většina z atributů je pro verzi 4 i 6 totožných, a z tohoto důvodu probíhá detekce pouze v jedné funkci.

Některé z atributů mohou být v zprávě RADIUS obsaženy více než jednou. U těchto atributů je implementováno tzv. *počítadlo výskytů* daného atributu, které má informativní charakter. Pokud se některý z těchto atributů vyskytne vícekrát, je počítadlo atributu inkrementováno. Vlastní hodnota atributu však zůstává nezměněna (exportuje se hodnota obsažená v prvním z výskytů). Dále uvádím výčet atributů, které plugin detekuje a zpracovává:

- *User-Name*
- *NAS-IP-Address*
- *NAS-Port*
- *Framed-Protocol*
- *Framed-IP-Address*
- *Framed-MTU*
- *Login-IP-Host*
- *Reply-Message*
- *Vendor-Specific*
- *Called-Station-ID*
- *Calling-Station-ID*
- *NAS-Identifier*
- *Proxy-State*
- *NAS-Port-Type*
- *Acct-Status-Type*
- *Acct-Session-ID*
- *Acct-Authentic*
- *Acct-Session-Time*
- *Acct-Link-Count*
- *NAS-IPv6-Address*
- *Login-IPv6-Host*
- *DNS-Server-IPv6-Address*

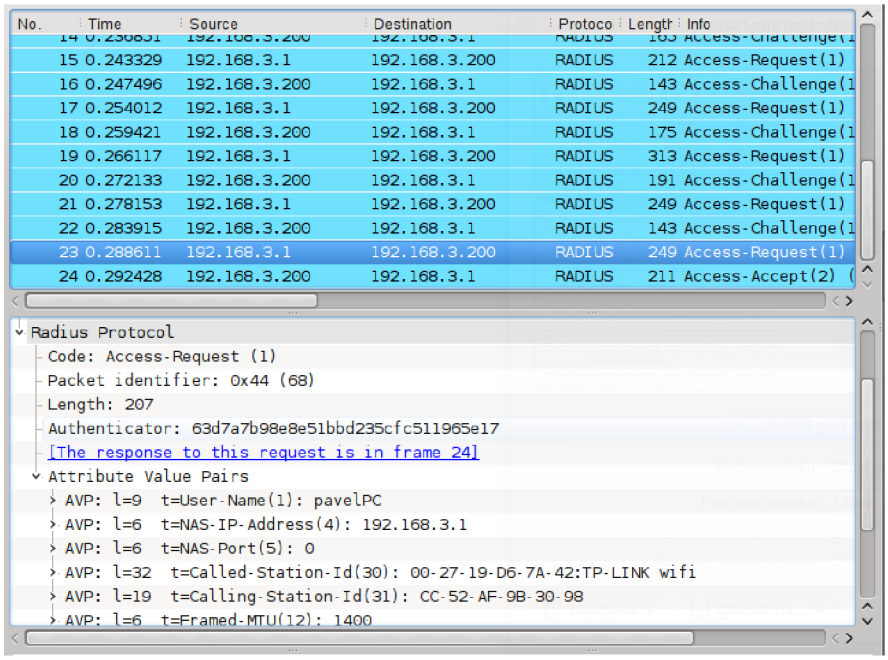
Identifikace atributů probíhá na základě číselné hodnoty položky `Type` v těle paketu. Získané hodnoty jsou ukládány v číselném formátu, ve formátu adres typu IPv4 či IPv6 nebo jako textový formát o maximální délce 32 znaků. Mnoho z atributů je v paketu RADIUS reprezentováno číselnou hodnotou. Ta má však většinou nízkou vypovídací schopnost. Z tohoto důvodu jsou takové atributy reprezentovány jejich textovou formou. Jako příklad lze použít hodnotu získanou z atributu `NAS-Port-Type`, kde namísto číselné hodnoty `19` bude uložena a exportována jeho textová reprezentace, tedy `Wireless - IEEE 802.11`. Po zpracování posledního z atributů je rozšiřování záznamu IPFIX u konce.

Tímto způsobem vznikal vstupní plugin *input-radius* monitorující provoz RADIUS v počítačové síti. Plugin se skládá ze dvou souborů: `input-radius.c` a `input-radius.h` (pro hlavičkový soubor). Hlavičkový soubor obsahuje zejména definovaná makra, pomocné struktury a v neposlední řadě textovou reprezentaci některých z atributů.

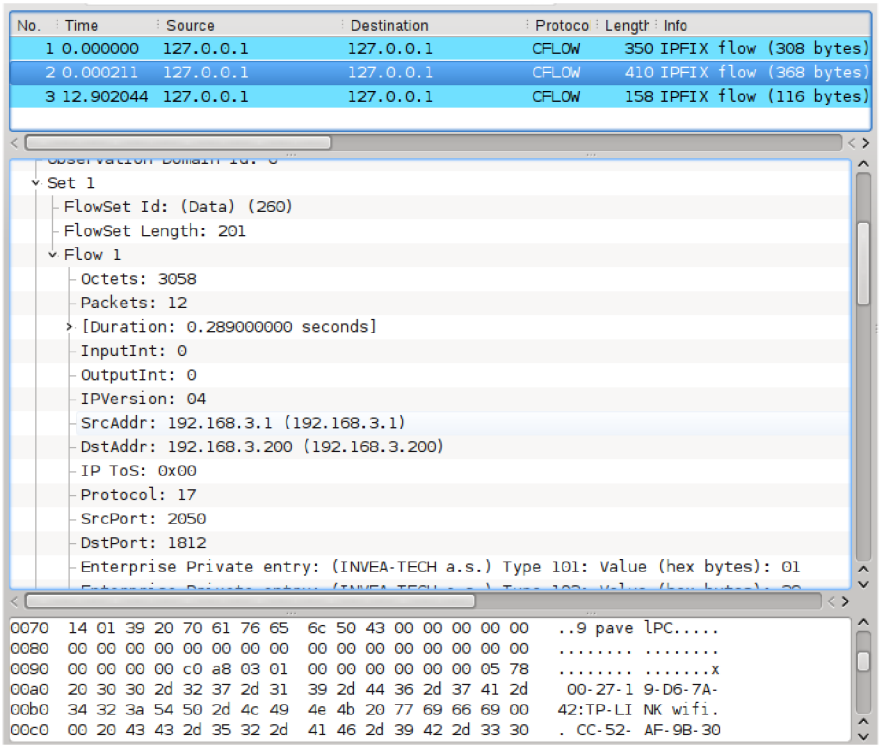
Implementovaný plugin však není schopen fungovat samostatně. Ke své činnosti potřebuje exportní plugin – `flowmon-export-ipfix` dodaný firmou INVEA-TECH. Ten na základě nastavení exportuje modifikované záznamy z FlowMon cache na kolektor. Obrázek č. 4.5 zachycuje rozšířený záznam IPFIX o data z obrázku č. 4.4. Oba výstupy jsou pořízeny programem *Wireshark*.

Nasbíraná data lze zobrazit přímo v příkazové řádce jako text za pomocí nástroje `fbitdump`[3]. Pro správné zobrazení informací, o které jsou záznamy IPFIX rozšířeny, bylo

[3]Fbitdump – jedná se o nástroj pro efektivní práci s IPFIX záznamy. Podrobnější informace o něm lze naleznout na http://is.muni.cz/th/255519/fi_m/thesis.pdf .



Obrázek 4.4: Ukázka komunikace RADIUS



Obrázek 4.5: Ukázka rozšířené zprávy IPFIX

nutné upravit konfigurační soubor zmíněného nástroje. Prázdné položky jsou ve výpisu reprezentovány pomlčkou.

Obrázek č. 4.6 demonstruje výpis dvou záznamů za pomocí nástroje fbitdump. Orientace mezi jednotlivými záznamy je kvůli velkému množství vypisovaných položek nepřehledná, a proto je k dispozici grafická nástavba. Jde o plugin do webového rozhraní kolektoru umožňující přehlednou vizualizaci výsledků.

```
        Date flow start  Duration      Proto         Src IPv4:sPort ->         Dst IPv4:dPort Packets Flows    Bytes
  Code Identif.             Username          NAS IP NAS Port         Service Type Framed Protocol        Framed IP Fr
amed MTU    Login IP Host:Count Vendor specific           Called SID          Calling SID         NAS Identif.     Prox
y state:Count        NAS port type
2014-04-28 09:01:37.363     0.405        UDP      192.168.3.1:2050  ->    192.168.3.200:1812        7     1     1678
     1      100    host/satellite-pc      192.168.3.1        0                   -           -                -     140
0                -:  -         - 00-27-19-D6-7A-42:TP-LINK wifi    00-14-A5-A0-C4-64                    -
-:  -      WIRELESS_802_11
2014-04-21 20:31:08.161     0.347        UDP      192.168.3.1:2050  ->    192.168.3.200:1812       12     1     3058
     1      194             pavelPC      192.168.3.1        0                   -           -                -     140
0                -:  -         - 00-27-19-D6-7A-42:TP-LINK wifi    CC-52-AF-9B-30-98                    -
-:  -      WIRELESS_802_11
```

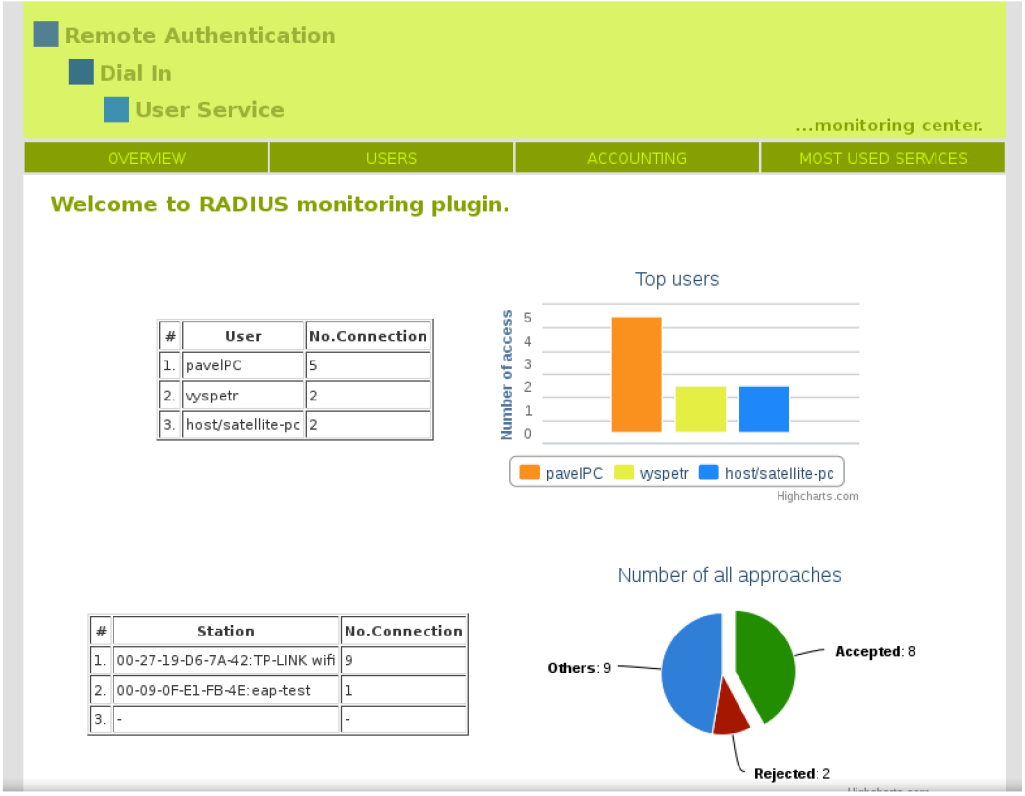
Obrázek 4.6: Výpis rozšířených IPFIX záznamů

4.4 Prostředí pro vizualizaci nasbíraných dat

Kolektor FlowMon umožňuje vizualizovat síťový provoz za pomocí webového rozhraní. Toto rozhraní lze rozšiřovat o další doplňky, pomocí nichž lze zobrazit podrobné statistiky, analýzy, hlášení anomálií, čímž se zvětšuje přehled nad komunikací v síti. Základní verze pouze provoz RADIUS detekuje, ale podrobné informace o komunikaci zobrazit nelze. Z tohoto důvodu bylo nutné vytvořit vizualizační plugin, který poskytne informace o provozu RADIUS v síti a umožní zobrazení statistik o provozu. Za implementační jazyk byl zvolen jazyk PHP s využitím *CodeIgniter framework* a volně dostupnou knihovnou pro tvorbu grafů *HighCharts*[4].

Úvodní stránka nabízí obecný přehled o komunikaci RADIUS v síti. Mezi informace, které lze na této stránce naleznout, patří statistiky o tom, kteří uživatelé a která zařízení žádali o autentizaci a autorizaci server nejčastěji a dále pak počet zamítnutých a povolených přístupů v závislosti na počtu všech požadavků na server. Stránka je znázorněna na obrázku č. 4.7. Hlavní nabídka poskytuje mimo obecného přehledu stránky s informacemi o uživatelích, účtování či přehledu služeb, o které uživatelé nejčastěji žádali. Rovněž nabízí přehled požadavků a odpovědí v čase, a to jak formou tabulek, tak i formou grafů. Jednotlivé záznamy se dají filtrovat pomocí uživatelského jména, IP adresy či ID přihlašovaného zařízení. Stránky týkající se účtování a služeb mají stejný charakter.

[4]HighCharts je knihovnou umožňující tvorbu grafů v programovacím jazyce *JavaScript*. Bezplatnou verzi lze použít pouze pro nekomerční účely.



Obrázek 4.7: Ukázka uvítací stránky webového rozhraní

4.5 Shrnutí

V této kapitole jsem ukázal, jak jsem postupoval při návrhu a implementaci pluginu. Kapitola obsahuje základní informace týkající se vlastní sondy FlowMon, která byla formou virtuálního zařízení poskytnuta, a také přehled činností, jež bylo nutné před vlastním započetím implementace realizovat. Byl také zmíněn princip při výběru atributů protokolu RADIUS, které mají užitečný informativní charakter. Jde o kapitolu klíčovou, popisující činnost v praktické části této bakalářské práce.

Kapitola 5

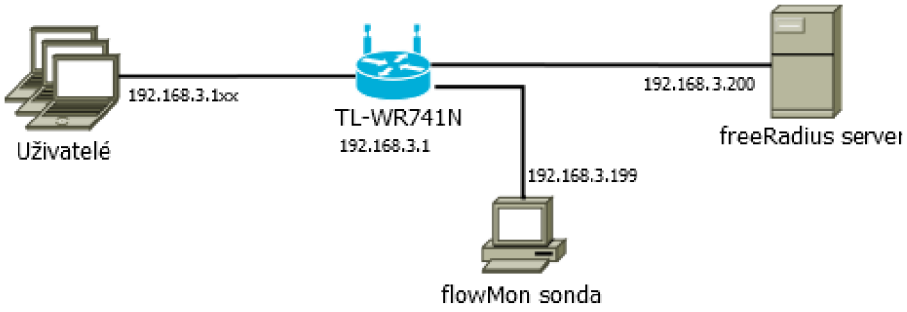
Testování

Testování tvoří důležitou část vývoje každého počítačového programu. Při vývoji vstupního pluginu pro sondu FlowMon tomu nebylo jinak. V této kapitole postupně demonstruji, jakým způsobem a nad jakými daty byl vytvořený plugin testován, a rovněž zhodnotím dosažené výsledky. Veškeré testování bylo prováděno za pomocí poskytnuté (virtuální) sondy FlowMon.

Za účelem testování vznikla za pomoci *freeRADIUS*[1] serveru domácí počítačová síť umožňující autentizaci a autorizaci uživatelů, kteří žádají o internetové připojení. Mimo sítě domácí byla vytvořena další počítačová sít a to v prostředí Cisco laboratoře Fakulty informačních technologií VUT v Brně. Testována byla také schopnost zpracovávat data jak ze souboru typu PCAP, tak data získaná z provozu na fyzickém rozhraní.

5.1 Testování v prostředí domácí sítě

Testování v domácí síti bylo umožněno skutečností, že bezdrátový směrovač *TP-Link TL-WR741N*, který je v této síti použit, dovoluje více variant zabezpečení přístupu. Jednou z nich je použití vzdáleného ověření uživatelů pomocí serveru RADIUS při bezdrátovém přístupu. Topologie sítě je znázorněna obrázkem č. 5.1.



Obrázek 5.1: Zapojení domácí sítě

Na bezdrátovém směrovači bylo nutné nastavit IP adresu (192.168.3.200) a port (1812) serveru RADIUS. Další nastavení pak probíhalo na straně serveru RADIUS. V konfiguračních souborech serveru RADIUS bylo třeba zadefinovat nové uživatele, kteří budou oprá-

[1] FreeRADIUS – jde o formu serveru RADIUS poskytující autorizaci, autentizaci a účtování uživatelů (dostupné na http://freeradius.org).

něni k využívání poskytované služby (v tomto případě se jedná o přístup k internetovému připojení).

Při tomto testu žádali různí uživatelé o přístup k internetovému připojení. Před akceptací byly serverem RADIUS požadovány dodatečné informace o uživateli. Poslední odpověď zaslaná serverem obsahuje informaci o povolení či zamítnutí přístupu. Vstupní plugin při tomto testu detekoval veškerou komunikaci a rozšířil záznamy IPFIX daných IP toků. Komunikaci, která na síti během testování proběhla, demonstruje tabulka 5.1.

Čas	**Celkem paketů**	**Požadavků**	**Odpovědí**	**Povolení přístupu**
22:11 7. 4. 2014	24	12	12	Ano
22:11 7. 4. 2014	26	13	13	Ano
23:09 7. 4. 2014	24	12	12	Ano
23:11 7. 4. 2014	24	12	12	Ano
05:53 8. 4. 2014	24	12	12	Ano
08:11 8. 4. 2014	24	12	12	Ano
09:21 8. 4. 2014	14	7	7	Ne
09:22 8. 4. 2014	14	7	7	Ne

Tabulka 5.1: Přehled provozu RADIUS na síti během testování

Každý řádek tabulky postupně udává: čas, kdy klient NAS vytvořil první požadavek na server RADIUS, celkový počet paketů v komunikaci klient-server (požadavky i odpovědi), počet paketu `Access-Request` (pomocí nich server RADIUS vyhodnotí, zda přístup uživateli povolí či odepře), počet paketů odpovědí serveru RADIUS (`Access-Access`, `Access-Reject` nebo `Access-Challenge`). Poslední sloupec udává, zda byl přístup uživateli povolen či nikoliv. Tabulka 5.2 reprezentuje zjednodušený výstup programu fbitdump.

Čas záznamu	**Počet paketů v rámci toku**	**Uživatelské jméno**	**Identifikátor stanice**
22:11 7. 4. 2014	12	pavelPC	CC-52-AF-9B-30-98
22:11 7. 4. 2014	13	pavelPC	CC-52-AF-9B-30-98
23:09 7. 4. 2014	12	pavelPC	CC-52-AF-9B-30-98
23:11 7. 4. 2014	12	vyspetr	28-CC-01-03-0E-7E
05:53 8. 4. 2014	12	pavelPC	28-CC-01-03-0E-7E
08:11 8. 4. 2014	12	vyspetr	00-14-A5-A0-C4-64
09:21 8. 4. 2014	7	host/satellite-pc	00-14-A5-A0-C4-64
09:22 8. 4. 2014	7	host/satellite-pc	00-14-A5-A0-C4-64

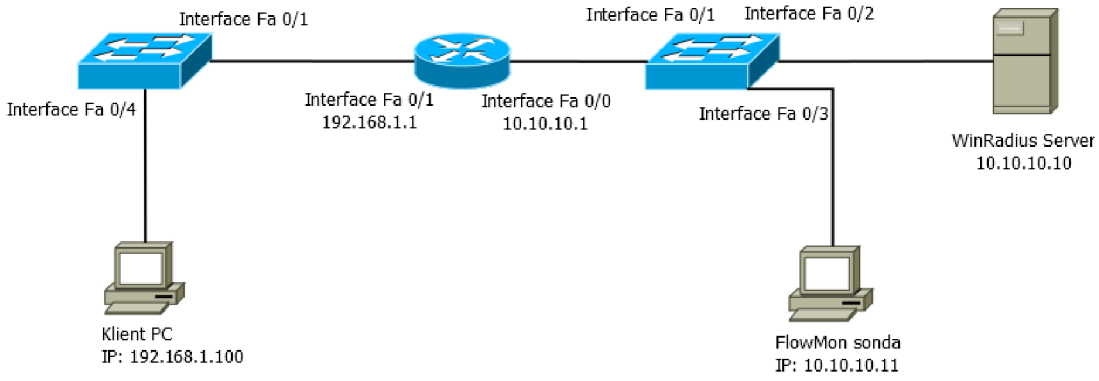
Tabulka 5.2: Ukázka získaných informací z požadavků na spojení

První sloupec tabulky reprezentuje čas, kdy byl pluginem vytvořen rozšířený záznam IPFIX. Dále udává počet zachycených paketů `Access-Request`, které byly serveru zaslány, než proběhlo ověření uživatelovi totožnosti. Následující dva sloupce zobrazují vybrané atributy RADIUS získané z komunikace ve směru klient-server.

Provoz z této počítačové sítě je zaznamenán na přiloženém DVD (v souborech typu PCAP). Kompletní výstup provedeného testu je uveden v příloze.

5.2 Laboratorní prostředí

Další testování vstupního pluginu probíhalo na jednoduché laboratorní síti s využitím zařízení společnosti *Cisco*. Roli serveru RADIUS zde zastoupil *WinRadius 3.00*[2]. Zapojení sítě je zachyceno obrázkem č. 5.2.



Obrázek 5.2: Zapojení sítě v Cisco laboratoři

Ověření uživatelů za pomocí RADIUS serveru v navržené síti proběhlo ve chvíli, kdy se některý z uživatelů pokusil o přístup ke konfiguraci směrovače. V tomto okamžiku byl vyzván k zadání uživatelského jména a hesla. Po ověření správnosti přihlašovacích údajů z RADIUS databáze byl přístup k nastavení směrovače povolen či odepřen. Část proběhlé RADIUS komunikace demonstruje následující tabulka.

Čas	Celkem paketů	Požadavků	Odpovědí	Povolení přístupu
16:10 12. 5. 2014	2	1	1	Ano
16:30 12. 5. 2014	2	1	1	Ano
16:43 12. 5. 2014	2	1	1	Ano
16:45 12. 5. 2014	2	1	1	Ano
17:00 12. 5. 2014	2	1	1	Ano
17:22 12. 5. 2014	2	1	1	Ano
17:49 12. 5. 2014	2	1	1	Ne
17:50 12. 5. 2014	2	1	1	Ano

Tabulka 5.3: Přehled provozu RADIUS na síti během testování

Vstupní plugin rozšířil IPFIX záznamy z proběhlé komunikace. K ověření správnosti zachycené komunikace může sloužit tabulka č. 5.4 demonstrující údaje o tocích rozšířených o některé z údajů obsažených v komunikaci RADIUS. Popis tabulek je analogický jako v případě tabulek č. 5.1 a 5.2.

5.3 Příklad komunikace RADIUS reálného provozu

V úvodu této kapitoly jsem zmínil, že byla testována i schopnost zpracovávat provoz ze souborů typu PCAP. Pro testování funkčnosti jsem využil webový portál *pcapr.net*, který

[2]Jde o volně dostupný nástroj simulující činnost serveru RADIUS v prostředí Windows (dostupný na http://winradius.soft112.com).

Čas záznamu	Počet paketů v rámci toku	Uživatelské jméno	Typ NAS portu
16:10 12. 5. 2014	1	radiusTest	Async
16:30 12. 5. 2014	1	usr1	Async
16:43 12. 5. 2014	1	usr1	Async
16:45 12. 5. 2014	1	usr1	Async
17:00 12. 5. 2014	1	usr1	Async
17:22 12. 5. 2014	1	usrPC1	VIRTUAL
17:49 12. 5. 2014	1	user	Async
17:50 12. 5. 2014	1	admin	Async

Tabulka 5.4: Ukázka získaných informací z požadavků na spojení

registrovaným uživatelům poskytuje velké množství různorodé komunikace v počítačové síti. Pro potřeby této bakalářské práce byly vybrány soubory PCAP, ve kterých je zachycena komunikace RADIUS při ověřování a účtování uživatelů. Prvním z testovaných souborů byl `EAP-TLS.pcap`. RADIUS komunikaci z tohoto souboru demonstruje obrázek č. 5.3.

```
  3 0.001385  192.168.1.99  192.168.1.112  RADIUS    192 Access-Request(1) (id=83, l=150)
  4 0.003902  192.168.1.112 192.168.1.99   RADIUS    106 Access-Challenge(11) (id=83, l=64)
  5 0.013031  192.168.1.99  192.168.1.112  RADIUS    306 Access-Request(1) (id=84, l=264)
  6 0.014623  192.168.1.112 192.168.1.99   RADIUS   1132 Access-Challenge(11) (id=84, l=1090)
  7 0.024533  192.168.1.99  192.168.1.112  RADIUS    207 Access-Request(1) (id=85, l=165)
  8 0.025473  192.168.1.112 192.168.1.99   RADIUS    981 Access-Challenge(11) (id=85, l=939)
  9 0.044477  192.168.1.99  192.168.1.112  IPv4     1514 Fragmented IP protocol (proto=UDP 17,
 10 0.044611  192.168.1.99  192.168.1.112  RADIUS    223 Access-Request(1) (id=86, l=1661)
 11 0.051505  192.168.1.112 192.168.1.99   RADIUS    106 Access-Challenge(11) (id=86, l=64)
 12 0.061210  192.168.1.99  192.168.1.112  RADIUS    781 Access-Request(1) (id=87, l=739)
 13 0.071228  192.168.1.112 192.168.1.99   RADIUS    169 Access-Challenge(11) (id=87, l=127)
 14 0.082837  192.168.1.99  192.168.1.112  RADIUS    207 Access-Request(1) (id=88, l=165)
 15 0.083925  192.168.1.112 192.168.1.99   RADIUS    208 Access-Accept(2) (id=88, l=166)
 16 0.960420  192.168.1.99  192.168.1.112  RADIUS    188 Accounting-Request(4) (id=89, l=146)
```

Obrázek 5.3: RADIUS komunikace v souboru EAP-TLS.pcap

V této komunikaci se uživatel pokoušel o přístup ke službě. Během ní si o něm server RADIUS vyžádal několik dodatečných informací. Po úspěšném ověření jeho totožnosti byl serveru zaslán požadavek na započetí účtovacího procesu.

Obrázek č. 5.4 znázorňuje upravený výpis informací po použití nástroje fbitdump, získaných vstupním pluginem.

Druhý z testovaných souborů nese název `radius_nas.pcap`. Dle zdrojové i cílové IP adresy a atributů paketu RADIUS lze usoudit, že jde pouze o testovací komunikaci, při které se ověřuje správné nastavení RADIUS serveru. Jedná se nejspíše o uměle vytvořenou komunikaci nástrojem *radclient* či jeho alternativou. Tabulka 5.5 znázorňuje zmíněný provoz.

No.	Time	Source	Destination	Protocol	Len.	Info
1	0.000000	127.0.0.1	127.0.0.1	RADIUS	90	Access-Request (id=1, l=48)
2	0.000045	127.0.0.1	127.0.0.1	RADIUS	68	Access-Accept (id=1, l=24)

Tabulka 5.5: Výpis provozu ze souboru radius_nas.pcap

Obrázek 5.5 demonstruje všechny položky, o které byly záznamy o toku vstupním pluginem rozšířeny. Opět jde o upravený výpis zobrazený nástrojem fbitdump.

```
-------------------------------------------------------------------------------------------------------
Date flow start: 2004-12-31 17:00:17.717  |  Duration: 0.081  |  Protocol: UDP  |  Type: Access-Request
-------------------------------------------------------------------------------------------------------
Source IP:Port    ->  Destination IP:Port  Packets  Flows    Bytes     Code    Identifier
192.168.1.99:1041 -> 192.168.1.112:1812      5       1       1693       1          83

Username  NAS IP   NAS Port  Framed MTU      Called SID               Calling SID       NAS port type
jtan      0.0.0.0    0         1400      00-09-0F-E1-FB-4E:eap-test  50-63-13-C1-A1-94  WIRELESS_802_11

-------------------------------------------------------------------------------------------------------
Date flow start: 2004-12-31 17:00:17.799  |  Duration: 0.000 |  Protocol: UDP  |  Type: Access-Accept
-------------------------------------------------------------------------------------------------------
Source IP:Port     -> Destination IP:Port Packets Flows  Bytes  Code  Identifier  Username Vendor spec.
192.168.1.112:1812 -> 192.168.1.99:1041     1      1    208     2       88          jtan       311

-------------------------------------------------------------------------------------------------------
Date flow start: 2004-12-31 17:00:17.719 | Duration: 0.068 |  Protocol: UDP  |  Type: Access-Challenged
-------------------------------------------------------------------------------------------------------
Source IP:Port     -> Destination IP:Port Packets Flows  Bytes  Code  Identifier
192.168.1.112:1812 -> 192.168.1.99:1041     5       1     2494   11       83

-------------------------------------------------------------------------------------------------------
Date flow start: 2004-12-31 17:00:18.676  |  Duration: 0.000 | Protocol: UDP | Type: Accounting-Request
-------------------------------------------------------------------------------------------------------
Source IP:Port    ->  Destination IP:Port Packets Flows  Bytes  Code  Identifier  Username  Status type
192.168.1.99:1040 ->  192.168.1.112:1813    1     1       188    4       89        jtan        1

Authentic. NAS IP    NAS Port        Called SID                Calling SID         NAS port type
    1      0.0.0.0     0       00-09-0F-E1-FB-4E:eap-test    50-63-13-C1-A1-94     WIRELESS_802_11
-------------------------------------------------------------------------------------------------------
```

Obrázek 5.4: Informace získané vstupním pluginem z komunikace v souboru EAP-TLS.pcap

```
-------------------------------------------------------------------------------------------------------
Date flow start: 2008-11-26 21:07:03.903  |  Duration: 0.000  |  Protocol: UDP  |  Type: Access-Request
-------------------------------------------------------------------------------------------------------
Source IP:Port  ->  Destination IP:Port  Packets  Flows  Bytes   Code  Identifier   Username   NAS IP
127.0.0.1:45298 ->  127.0.0.1:1812         1        1     90      1         1         mu      127.0.0.1

-------------------------------------------------------------------------------------------------------
Date flow start: 2008-11-26 21:07:03.909  |  Duration: 0.000 |  Protocol: UDP  |  Type: Access-Accept
-------------------------------------------------------------------------------------------------------
Source IP:Port  ->  Destination IP:Port  Packets  Flows  Bytes  Code  Identifier
127.0.0.1:1812  ->  127.0.0.1:45298      1      1     68     2       1
-------------------------------------------------------------------------------------------------------
```

Obrázek 5.5: Výsledek zpracování komunikace souboru radius_nas.pcap vstupním pluginem

5.4 Dosažené výsledky při testování

Z výsledků jednotlivých testů lze učinit závěr, že vytvořený plugin je schopen správně detekovat provoz RADIUS na síti. Z komunikace na portu 1812 sbírá data z paketů, které jsou zaslány pro autentizaci a autorizaci uživatele. Z komunikace směřující na port 1813 získá informace týkající se účtování uživatelů. O tyto informace pak plugin rozšíří záznamy IPFIX, které jsou zaslány na kolektor.

5.5 Možná rozšíření

Vytvořený plugin je možné rozšířit o další funkcionalitu, která by umožňovala monitorování komunikace RADIUS více do hloubky. V případě potřeby lze implementaci rozšířit o detekci provozu na starých portech RADIUS (1645 a 1646). Dále by bylo možné přidat detekci a zpracování zbývajících atributů protokolu RADIUS, ale jak jsem již zmínil, v běžné komunikaci se příliš nevyskytují nebo mají nízkou vypovídací úroveň.

Webové rozhraní by bylo možné rozšířit o další statistiky, které by zobrazovaly dodatečné informace o uživatelích. Dalším potenciálním rozšířením by mohlo být vytvoření „katalogu uživatelů“, který by shromažďoval informace o jednotlivých uživatelích a zobrazoval statistiky, jak využívali služeb a počítačové sítě.

5.6 Shrnutí

V této kapitole jsem uvedl přehled provedených testů ověřujících správnost implementace a návrhu pluginu. Kontrola správnosti implementace spočívala v porovnání vstupní komunikace s výstupy pluginu a také v kontrole rozšířených záznamu IPFIX zasílaných exportérem na kolektor. Formou obrázků zde byly demonstrovány sítě, ve kterých byla poskytnutá sonda FlowMon nasazena. U každého z testů jsou uvedeny informace o vstupních datech a o úspěšnosti detekce RADIUS provozu. Závěr kapitoly je věnován zhodnocení dosažených výsledků při monitorování RADIUS provozu v síti a případným rozšířením vstupního pluginu a vizualizačního nástroje.
Veškerá komunikace popsaná v této kapitole je dostupná v podobě PCAP souborů na přiloženém DVD.

Kapitola 6

Závěr

Tato bakalářská práce si kladla za cíl navrhnout a posléze implementovat nové rozšíření (plugin) pro sondu FlowMon společnosti INVEA-TECH, která díky tomuto rozšíření bude schopna detekovat provoz RADIUS v počítačové síti a rozšířit o získané informace IPFIX záznamy o toku. Výsledkem práce je vytvoření vstupního pluginu pro sondu a vizualizačního nástroje pro zobrazení rozšířených IPFIX záznamů na kolektoru.

V úvodu teoretické části práce jsem představil protokol RADIUS a možnosti, které protokol nabízí, a také technologii NetFlow/IPFIX. Prostudoval jsem také vývojové prostředí FlowMon sondy a zhodnotil možnosti pro tvorbu pluginů. Po získání všech potřebných informací jsem navrhl způsob, jakým by bylo možné zpracovat RADIUS komunikaci, uchovat užitečné informace získané z provozu a rozšířit o ně IPFIX záznam.

V praktické části práce jsem implementoval vstupní plugin pro exportér sondy FlowMon podle dříve vytvořeného návrhu. Vytvořený plugin je schopen detekovat provoz RADIUS v síti a zpracovat vybrané položky paketu RADIUS. Komunikaci RADIUS lze tímto rozšířením monitorovat v sítích s komunikací IPv4 i IPv6. O získané položky rozšiřuji záznam IPFIX. Po uzavření toku jsou modifikované záznamy exportovány na kolektor exportním pluginem. Společně se vstupním pluginem jsem implementoval rozšíření pro vizualizaci nasbíraných dat na kolektoru. Funkčnost řešení byla otestována na reálném provozu v domácí síti. Mezi další testovací prostředí patřila síť vytvořená v Cisco laboratoři na FIT VUT v Brně. Provedené testy prokázaly schopnost pluginu detekovat provoz RADIUS a správný způsob rozšíření záznamu IPFIX.

Vstupní plugin pro exportér i vizualizační nástroj pro zobrazení dat je možné rozšířit o další funkčnost. Návrhy na možné rozšíření jsem zmínil v kapitole 5.

Díky této práci jsem získal přehled nad možnostmi monitorování sítě technologií NetFlow/IPFIX a způsobu ověření a účtování uživatelů za pomoci protokolu RADIUS. Dále jsem se seznámil s možnostmi vývoje rozšiřujících pluginů pro zařízení FlowMon a činností exportního pluginu *flowmon-ipfix*, který je dodán společně se sondou. RADIUS je stále jedním z nejvíce rozšířených protokolů AAA a lze předpokládat, že tento typ zabezpečení v nejbližší době nevymizí. Monitorováním tohoto provozu správce sítě získá kompletní přehled nad uživateli, kteří se přihlásili ke službám, a také přehled o účtování uživatelů. Z provozu lze také detekovat možné pokusy o prolomení hesel při přístupu ke službě.

Věřím, že se mnou vytvořený plugin, včetně vizualizačního nástroje, společnosti INVEA-TECH osvědčí tak, že jej nabídne i svým zákazníkům. Současně bych rád ve spolupráci s touto firmou do budoucna svoji práci dále rozšířil.

Literatura

[1] Aboba, B.; Zorn, G.; Mitton, D.: RADIUS and IPv6. RFC 3162, srpen 2001.

[2] Calhoun, P.; Loughney, J.; Guttman, E.; aj.: Diameter Base Protocol. RFC 6733, říjen 2012.

[3] Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954, říjen 2004.

[4] Claise, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011, září 2013.

[5] Claise, B.; Kobayashi, A.; Trammell, B.: Operation of the IP Flow Information Export (IPFIX) Protocol on IPFIX Mediators. RFC 7119, únor 2014.

[6] Dec, W. E.; Sarikaya, B.; Zorn, G. E.; aj.: RADIUS Attributes for IPv6 Access Networks. RFC 6911, duben 2013.

[7] Elich, M.: *Rozšíření NetFlow kolektoru NfSen o detekci síťových anomálií [online].* Diplomová práce, Masarykova univerzita, Fakulta informatiky, 2009.

[8] Hassell, J.: *RADIUS.* O´Reilly, 2003, ISBN 0–596–00322–6.

[9] INVEA-TECH: FlowMon. [online], [cit. 2014-02-10].
URL http://www.invea.cz/products/flowmon

[10] INVEA-TECH: FlowMon sondy. [online], [cit. 2014-02-11].
URL http://www.invea.cz/produkty-sluzby/flowmon/flowmon-sondy

[11] Kobayashi, A.; Claise, B.; Muenz, G.; aj.: IP Flow Information Export (IPFIX) Mediation: Framework. RFC 6183, duben 2011.

[12] Lloyd, B.; Simpson, W.: PPP Authentication Protocols. RFC 1334, říjen 1992.

[13] Rigney, C.: RADIUS Accounting. RFC 2866, červen 2000.

[14] Rigney, C.; Rubens, A.; Simpson, W.; aj.: Remote Authentication Dial In User Service (RADIUS). RFC 2865, červen 2000.

[15] Rivest, R.: The MD5 Message-Digest Algorithm. RFC 1321, duben 1992.

[16] Sadasivan, G.; Brownlee, N.; Claise, B.; aj.: Architecture for IP Flow Information Export. RFC 5470, březen 2009.

[17] Simpson, W.: PPP Challenge Handshake Authentication Protocol (CHAP). RFC 1994, srpen 1996.

[18] Trammell, B.; Boschi, E.: An introduction to IP flow information export (IPFIX). *Communications Magazine, IEEE*, ročník 49, č. 4, April 2011: s. 89–95, ISSN 0163–6804, doi:10.1109/MCOM.2011.5741152.

[19] Velan, P.: FlowMon Exporter 3.05.x Documentation. 2013, [cit. 2014-02-25]. URL https://www.invea.cz/trac/community/raw-attachment/wiki/WikiStart/flowmonexp-doc.pdf

[20] Vollbrecht, J.: The Beginnings and History of RADIUS. 2009-04-15, [cit. 2013-12-21]. URL http://www.interlinknetworks.com/app_notes/HistoryofRADIUS.pdf

Dodatek A

Obsah DVD

- `README.txt` ..... manuál k použití
- `BP_Pavel_Vyskocil_2014.pdf` ..... text práce ve formátu PDF
- `Radius_plugin` ..... zdrojové soubory pluginu
  - `input` ..... zdrojové soubory vstupního pluginu
  - `vizualizer` ..... zdrojové soubory pluginu pro vizualizaci
  - `config` ..... upravené konfigurační soubory
- `tex` ..... zdrojové soubory textové části bakalářské práce
  - `img` ..... obrázky použité v textové části bakalářské práce
- `tests` ..... soubory s testovanou komunikací RADIUS
  - `home_network` ..... provoz z domácí sítě
  - `lab_network` ..... provoz z Cisco laboratoře
  - `pcapr_net` ..... provoz získaný z portálu pcapr.net

Dodatek B

Manuál

Plugin je nutno překládat a spouštět na virtuální nebo hardwarové sondě FlowMon ve verzi 3.4.2 (a vyšší) s nainstalovaným rozšířením IPFIXCol. Sonda musí obsahovat exportní plugin *flowmon-export-ipfix*, který je nezbytný pro činnost vstupního pluginu. Pro správnou činnost nově vytvořeného vstupního pluginu je třeba nahradit soubory fbitdump.xml (*/usr/share/fbitdump/*), ipfix-elements.xml (*/etc/ipfixcol/*) a ipfix-template-file.txt (*/etc/-flowmon/*) soubory ze složky *Radius_plugin/config* z přiloženého DVD. Toto rovněž obsahuje v práci zmíněný vizualizační nástroj. Obsah adresáře *Radius_plugin/vizualizer* je nutné nahrát do adresáře */var/www/html/community* na sondě. Toto jsou nezbytné kroky před prvním spuštěním pluginu.

Překlad vstupního pluginu se provede zadáním příkazu `make`, kde přiložený Makefile zajistí překlad zdrojového souboru.
Kroky při spuštění pluginu:

1. Spustit nástroj IPFIXCol příkazem `ipfixcol`.
2. Spustit exportér flowmon se vstupním pluginem input-radius a exportním pluginem ipfixx.

Spuštění pluginu ze složky Radius_plugin, pro detekci provozu RADIUS na síťovém rozhraní:

```
sudo flowmonexp -X /home/flowmon/ipfix_export/flowmon-export-ipfix.so -X
./input/input_radius.so -I input-radius:pcap_if=nazev_rozhrani -E
ipfixx:host=localhost,port=4739
```

Pro detekci provozu ze souboru typu PCAP je nutno spustit jako:

```
input-radius:pcap_file=soubor.pcap
```

Záznamy IPFIX lze zobrazit pomocí nástroje fbitdump následovně:

- Požadavky na server RADIUS:
  `fbitdump -R /data/ipfixcol -o radius-request ‘‘%dstport = 1812’’`
- Odpovědi serveru RADIUS:
  `fbitdump -R /data/ipfixcol -o radius-response ‘‘%srcport = 1812’’`
- Požadavky na účtování RADIUS:
  `fbitdump -R /data/ipfixcol -o radius-accounting ‘‘%dstport = 1813’’`

Při nasazení v síti IPv6 je třeba spouštět jako *radiusv6-request*, *radius6-response* a *radius6-accounting*. Vizualizační nástroj lze spustit přes webové rozhraní monitorovacího centra záložkou *Community*.

Dodatek C

Ukázka výstupů z testování

C.1 Domácí síť

```
=================================================================================================
Type: Access-Request  |  Protocol: UDP   | Flows: 8 | Packets: 87
Source IP:Port -> Destination IP:Port :  192.168.3.1:2050 -> 192.168.3.200:1812
Called SID: 00-27-19-D6-7A-42:TP-LINK wifi  |  NAS port type:  WIRELESS_802_11
=================================================================================================
Date flow start         Duration   Packets  Flows  Bytes  Code   Identifier   Username      NAS IP
NAS Port  Framed MTU     Calling SID
-------------------------------------------------------------------------------------------------
2014-04-07 22:11:36.965   0.348      12       1     3058   1       194         pavelPC      192.168.3.1
0           1400       CC-52-AF-9B-30-98
-------------------------------------------------------------------------------------------------
2014-04-07 22:11:38.823   9.492      13       1     3371   1       206         pavelPC      192.168.3.1
0           1400       CC-52-AF-9B-30-98
-------------------------------------------------------------------------------------------------
2014-04-07 23:09:25.990   3.982      12       1     3232   1       218         vyspetr      192.168.3.1
0           1400       28-CC-01-03-0E-7E
-------------------------------------------------------------------------------------------------
2014-04-07 23:11:15.265   0.401      12       1     3058   1       254         pavelPC      192.168.3.1
0           1400       CC-52-AF-9B-30-98
-------------------------------------------------------------------------------------------------
2014-04-08 05:53:50.588  0.181       12       1     3232   1        69         pavelPC      192.168.3.1
0           1400       28-CC-01-03-0E-7E
-------------------------------------------------------------------------------------------------
2014-04-08 08:11:48.540  6.524       12       1     3056   1        81         vyspetr      192.168.3.1
0           1400       00-14-A5-A0-C4-64
-------------------------------------------------------------------------------------------------
2014-04-08 09:21:53.003  2.875        7       1     1678   1        93    host/satellite-pc 192.168.3.1
0           1400       00-14-A5-A0-C4-64
-------------------------------------------------------------------------------------------------
2014-04-08 09:22:05.082  0.406        7       1     1678   1       100    host/satellite-pc 192.168.3.1
0           1400       00-14-A5-A0-C4-64
=================================================================================================

=================================================================================================
Type: Access-Accept  |  Protocol: UDP   | Flows: 6 | Packets: 6
Source IP:Port -> Destination IP:Port :  192.168.3.200:1812 -> 192.168.3.1:2050
=================================================================================================
Date flow start         Duration   Packets  Flows  Bytes  Code   Identifier   Username  Vendor specific
-------------------------------------------------------------------------------------------------
2014-04-07 22:11:37.382   0.000       1       1      211    2       205       pavelPC        311
-------------------------------------------------------------------------------------------------
2014-04-07 22:11:49.033   0.000       1       1      211    2       217       pavelPC        311
-------------------------------------------------------------------------------------------------
2014-04-07 23:09:30.004   0.000       1       1      211    2       229       vyspetr        311
-------------------------------------------------------------------------------------------------
2014-04-07 23:11:15.670   0.000       1       1      211    2       254       pavelPC        311
-------------------------------------------------------------------------------------------------
```

```
2014-04-08 05:53:51.063   0.000       1       1      211    2        80       pavelPC        311
-------------------------------------------------------------------------------------------------------
2014-04-08 08:11:55.068   0.000       1       1      211    2        92       vyspetr        311
=======================================================================================================

=======================================================================================================
Type: Access-Reject  |  Protocol: UDP   | Flows: 2 | Packets: 2
Source IP:Port -> Destination IP:Port :  192.168.3.200:1812 -> 192.168.3.1:2050
=======================================================================================================
Date flow start         Duration   Packets  Flows  Bytes  Code   Identifier   Username  Vendor specific
-------------------------------------------------------------------------------------------------------
2014-04-08 09:21:56.871   0.000       1       1       86    3        99          -            -
-------------------------------------------------------------------------------------------------------
2014-04-08 09:22:06.486   0.000       1       1       86    3       106          -            -
=======================================================================================================

=======================================================================================================
Type: Access-Challenge  |  Protocol: UDP   | Flows: 9 | Packets: 79
Source IP:Port -> Destination IP:Port :  192.168.3.200:1812 -> 192.168.3.1:2050
=======================================================================================================
Date flow start         Duration   Packets  Flows  Bytes  Code   Identifier
-------------------------------------------------------------------------------------------------------
2014-04-07 22:11:36.988   0.318       11     1     4625    11       194
-------------------------------------------------------------------------------------------------------
2014-04-07 22:11:38.835   2.405        9     1     4291    11       206
-------------------------------------------------------------------------------------------------------
2014-04-07 22:11:47.469   0.840        3     1      525    11       215
-------------------------------------------------------------------------------------------------------
2014-04-07 23:09:27.843   2.122       11     1     4843    11       218
-------------------------------------------------------------------------------------------------------
2014-04-07 23:11:15.539   0.121       11     1     4625    11       254
-------------------------------------------------------------------------------------------------------
2014-04-08 05:53:50.592   0.171       11     1     4843    11        69
-------------------------------------------------------------------------------------------------------
2014-04-08 08:11:48.543   6.123       11     1     4618    11        81
-------------------------------------------------------------------------------------------------------
2014-04-08 09:21:53.007   0.546        6     1     3801    11        93
-------------------------------------------------------------------------------------------------------
2014-04-08 09:22:05.380   0.080        6     1     3801    11       100
=======================================================================================================
```

C.2 Cisco laboratoř

```
=======================================================================================================
Type: Access-Request  |  Protocol: UDP   | Flows: 15 | Packets: 15
Source IP:Port -> Destination IP:Port :  10.10.10.1:1645  -> 10.10.10.10:1812   |  NAS IP: 10.10.10.1
=======================================================================================================
Date flow start         Duration  Packets  Flows  Bytes  Code  Identifier  Username  NAS Port  NAS Port
                                                                                                 Type
-------------------------------------------------------------------------------------------------------
2014-05-12 16:10:43.986  0.000        1       1     116    1        3     radiusTest     0        Async
-------------------------------------------------------------------------------------------------------
2014-05-12 16:30:37.485  0.000        1       1     110    1        4        usr1        0        Async
-------------------------------------------------------------------------------------------------------
2014-05-12 16:31:20.235  0.000        1       1     110    1        5        usr1        0        Async
-------------------------------------------------------------------------------------------------------
2014-05-12 16:43:09.721  0.000        1       1     110    1        6        usr1        0        Async
-------------------------------------------------------------------------------------------------------
2014-05-12 16:45:12.102  0.000        1       1     110    1        7        usr1        0        Async
-------------------------------------------------------------------------------------------------------
2014-05-12 16:47:14.086  0.000        1       1     110    1        8        usr1        0        Async
-------------------------------------------------------------------------------------------------------
2014-05-12 16:48:11.917  0.000        1       1     110    1        9        usr1        0        Async
-------------------------------------------------------------------------------------------------------
2014-05-12 17:00:11.592  0.000        1       1     110    1       10        usr1        0        Async
-------------------------------------------------------------------------------------------------------
2014-05-12 17:19:08.963  0.000        1       1     111    1       11       admin        0        Async
```

```
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:22:14.493  0.000        1       1     114    1       12       usrPC1     514     VIRTUAL
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:23:35.461  0.000        1       1     113    1       13       admin      514     VIRTUAL
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:29:01.619  0.000        1       1     112    1       14       usrPC1       0       Async
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:38:15.941  0.000        1       1     111    1       15       admin        0       Async
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:49:50.511  0.000        1       1     110    1       16        user        0       Async
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:50:08.999  0.000        1       1     111    1       17       admin        0       Async
=============================================================================================================

=============================================================================================================
Type: Access-Accept  |  Protocol: UDP   | Flows: 14 | Packets: 14
Source IP:Port -> Destination IP:Port :  10.10.10.10:1812 -> 10.10.10.1:1645
=============================================================================================================
Date flow start         Duration   Packets  Flows  Bytes  Code   Identifier
-------------------------------------------------------------------------------------------------------------
2014-05-12 16:10:43.986     0.000    1        1      68     2         3
-------------------------------------------------------------------------------------------------------------
2014-05-12 16:30:37.486     0.000    1        1      68     2         4
-------------------------------------------------------------------------------------------------------------
2014-05-12 16:31:20.236     0.000    1        1      68     2         5
-------------------------------------------------------------------------------------------------------------
2014-05-12 16:43:09.723     0.000    1        1      68     2         6
-------------------------------------------------------------------------------------------------------------
2014-05-12 16:45:12.103     0.000    1        1      68     2         7
-------------------------------------------------------------------------------------------------------------
2014-05-12 16:47:14.087     0.000    1        1      68     2         8
-------------------------------------------------------------------------------------------------------------
2014-05-12 16:48:11.917     0.000    1        1      68     2         9
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:00:11.593     0.000    1        1      68     2        10
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:19:08.965     0.000    1        1      68     2        11
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:22:14.494     0.000    1        1      68     2        12
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:23:35.462     0.000    1        1      68     2        13
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:29:01.621     0.000    1        1      68     2        14
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:38:15.942     0.000    1        1      68     2        15
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:50:08.999     0.000    1        1      68     2        17
=============================================================================================================

=============================================================================================================
Type: Access-Reject  |  Protocol: UDP   | Flows: 1 | Packets: 1
Source IP:Port -> Destination IP:Port :  10.10.10.10:1812 -> 10.10.10.1:1645
=============================================================================================================
Date flow start         Duration   Packets  Flows  Bytes  Code   Identifier
-------------------------------------------------------------------------------------------------------------
2014-05-12 17:49:50.513    0.000     1        1      62     3        16
=============================================================================================================
```