TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

DISERTAČNÍ PRÁCE

Autentizační mechanismy v distribuovaném prostředí a jejich aplikace

Autor: Mgr. Jindřich Jelínek

Vedoucí práce: doc. RNDr. Pavel Satrapa, Ph.D.

Technická univerzita v Liberci, Studentská 2, 461 17 Liberec Tel.: 485 353 763 http://www.fm.tul.cz

ANOTACE

Disertační práce „Autentizační mechanismy v distribuovaném prostředí a jejich aplikace“ se zaměřuje na problematiku distribuovaných sítí založených na autentizačním protokolu RADIUS, jakou je například v současnosti velmi rozšířená síť *eduroam*. V práci jsou rozebrány dva hlavní směry inovace protokolu RADIUS.

V první části se práce zabývá vylepšením protokolu RADIUS za účelem zvýšení spolehlivosti činnosti distribuované počítačové sítě, která využívá k autentizaci uživatelů stávající implementaci protokolu RADIUS. Byl navržen nový algoritmus činnosti protokolu, který využívá k autentizaci uživatelů dostupné informace ze všech autentizačních serverů systému. Činnost nového algoritmu je simulována s využitím barevných Petriho sítí.

Druhá část práce se zaměřuje na inovaci protokolu RADIUS z hlediska bezpečnosti sítě na systémové úrovni. Práce navrhuje nový přístup k využití protokolu RADIUS Accounting pro předávání bezpečnostních informací o klientech. Zlepšení je založeno na novém systému výměny informací mezi RADIUS servery a spoluprací s nasazenými Intrusion Detection Systémy.

Výsledky této práce mohou přispět k celkovému zvýšení spolehlivosti a bezpečnosti distribuovaných sítí založených na protokolu RADIUS, protože navrhují řešení nevyřešených situací, které mohou nastat.

Klíčová slova

RADIUS protokol, *eduroam*, distribuovaná počítačová síť, IDS, autentizační mechanismus, Petriho sítě, CPN Tools

ABSTRACT

PhD thesis “Authentication mechanisms in a distributed environments and their applications“ focuses on issues of distributed networks based on the RADIUS authentication protocol, such as the currently widespread *eduroam* network. Two main directions of innovation of the RADIUS protocol are discussed in this work.

The first part of the thesis deals with improving the RADIUS protocol to increase reliability in the distributed computer networks using existing implementation of the RADIUS protocol for authentication of users. Operations of the new algorithm are simulated by colored Petri nets.

The second part focuses on innovation of the RADIUS protocol in terms of network security at the system level. We propose a new approach to use the RADIUS Accounting protocol for transmission of a security information about clients. The improvement is based on a new system of exchanging information between RADIUS servers and cooperating Intrusion Detection Systems.

The results of this work may contribute to the overall reliability and security of distributed networks based on the RADIUS protocol because it proposes solutions of unsolved situations which may occur.

Key words

RADIUS Protocol, Eduroam, Distributed Computer Network, IDS, Authentication Mechanism, Petri Nets, CPN Tools

PODĚKOVÁNÍ A PROHLÁŠENÍ

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně a použil jen těch zdrojů, které cituji a uvádím v přiloženém seznamu zdrojů.

V Ústí nad Labem, dne 20. 4. 2013 Jindřich Jelínek

Děkuji všem, kteří pomohli ke vzniku této práce. Zejména školiteli docentu Pavlu Satrapovi a doktoru Jiřímu Fišerovi za jejich nedocenitelné rady a inspirace.
Dále děkuji svým nejbližším, že mně poskytli časový prostor pro dokončení této práce a zároveň mě motivovali, abych tento prostor nepromrhal.

OBSAH

1 ÚVOD

Rostoucí portfolio, kvalita a dostupnost služeb internetu vedou jeho uživatele ke snaze získat dostatečně rychlé připojení na libovolném místě. Kromě jiných možností lze zaznamenat rozvoj tzv. **distribuovaných sítí**. Jejich základní charakteristikou je, že se vyskytují na velkém území, jsou heterogenní, rozptýlené a mají různé "provozovatele" i technická řešení. Jsou obvykle tvořeny jednotlivými přístupovými sítěmi (často bezdrátovými) a jsou vzájemně propojeny prostřednictvím nějaké transportní sítě do celku s určitou mírou spojitosti. V takové distribuované síti lze odlišit například podle „provozovatelů" dílčích částí sítě jednotlivé oblasti.

Příslušná oblast („doména", resp. jednoznačný identifikátor) sítě se označuje jako **realm.** Tyto sítě primárně slouží pro poskytování připojení k internetu a příp. intranetových služeb domovské sítě příslušnému uživateli, i když se fyzicky nachází na geograficky vzdáleném místě od této domovské sítě. Charakteristikou těchto vzájemně propojených sítí je obrovské množství potenciálních uživatelů, kteří mohou migrovat mezi jednotlivými dílčími oblastmi této distribuované sítě. Další charakteristickou vlastností obdobných sítí je **federativní uspořádání** připojených institucí[i]. Jednotlivé instituce zapojené v dané federaci rozdělujeme na **poskytovatele identit** a **poskytovatele služeb**. Poskytovatel identity je vždy vůči danému uživateli jen jeden. Síť poskytovatele identity uživatele pak označujeme jako *domovskou síť* tohoto uživatele. Vůči danému uživateli jsou sítě ostatních institucí, než síť poskytovatele identity (domovská síť), v roli poskytovatele služeb.

V souvislosti s tím, co bylo uvedeno, dochází při využívání uvedeného typu sítí k přesunům stále většího množství dat, která jsou nějakým způsobem pro uživatele těchto sítí bezpečnostně citlivá. Za data citlivá z hlediska bezpečnosti můžeme označit taková data, jejichž odposlech, modifikace či znehodnocení při přenosu sítí by vytvořilo pro příslušného uživatele či skupinu uživatelů sítě

[i] Federativním uspořádáním se rozumí fakt, že instituce zapojené do takové federace mají vlastní nezávislou správu své sítě, přičemž v souvislosti s provozováním společné distribuované sítě spolu navzájem spolupracují na základě nějakých společně definovaných pravidel [4].

hrozbu. Hrozbou může být například neoprávněné použití (zneužití) citlivých dat, změna jejich významu na pragmatické úrovni či zabránění jejich doručení.

Vzhledem k velkému množství migrujících uživatelů distribuované sítě můžeme definovat nová bezpečnostní rizika vycházející z poměrně volného uspořádání vzájemně propojených institucí a z chování uživatelů, které se může v sítích dílčích institucí lišit. Objevuje se tak otázka, jak se mají dílčí části distribuované sítě chovat k uživatelům, kteří jednají v některých realmech nestandardně z hlediska bezpečnosti, přičemž toto nestandardní jednání lze vyvodit z nějakého bezpečnostního systému příslušného provozovatele sítě.

V úvodu bylo naznačeno, že uživatelé očekávají od počítačových sítí spolehlivé služby. Nicméně při používání služeb distribuované sítě se může v některých případech stát, že dojde k výpadku domovského serveru příslušného realmu, resp. k výpadku spojení s tímto serverem. Tím může dojít k odepření služeb uživateli. Vzniká otázka, zda není možné v tomto případě nějak nahradit chybějící informace od domovského serveru informacemi z jiného zdroje, když je používaná síť distribuovaná a požadovaná autentizační informace tedy může být v jiném uzlu. Na druhou stranu ale přihlašovací informace uživatele, který žádá nějaký autentizační server o autentizaci, mohou být uloženy v jiných autentizačních serverech a jejich informace nemusí být vzájemně konzistentní. Jedná se tedy o problém z hlediska spolehlivosti sítě, který bude vyžadovat nalezení vhodného algoritmu řešení, komunikačních pravidel mezi servery a dohodu systému na nějakém autentizačním rozhodnutí.

Nastíněná problematika povede do oblasti autentizačních protokolů, rozhodování v distribuovaných systémech a síťových bezpečnostních systémů.

Vzhledem k výše nastíněným problémům bude tato práce zaměřena na návrh nových přístupů a součástí autentizace a účtování v distribuovaných sítích s federativní autentizací tak, aby mohly být zdokonaleny některé postupy při řešení bezpečnostních incidentů a při poskytování služeb za zhoršených podmínek, tedy spolehlivější a bezpečnější služby.

2 PŘEHLED O SOUČASNÉM STAVU PROBLEMATIKY

2.1 Definice základních pojmů

2.1.1 Dependabilita

V obecné řeči se často setkáváme s pojmem **spolehlivost**, který nelze považovat za dostatečný, protože nezahrnuje dostatečně široké spektrum selhání, kterých se problematika řešená v této disertační práci bezprostředně týká. Pojem spolehlivost je podle [1] pouze jeden z atributů zahrnutých v širším termínu dependabilita.

Termín **dependabilita systému** je poměrně obecně v [1], str. 123 definován jako *„…schopnost systému vyhnout se takovým selháním, jejichž četnost výskytu, resp. závažnost, by byla větší než úroveň přípustná pro uživatele.“*

Dependabilitu systému lze dále definovat a případně kvantifikovat primárními a sekundárními atributy. Mezi primární atributy podle [1] patří:

1. Dostupnost (*availability*), což je pohotovost k provedení služby;
2. Spolehlivost (*reliability*), tedy kontinuita v poskytování služby;
3. Zabezpečení (*safety*) z hlediska vyhýbání se katastrofickým následkům;
4. Důvěrnost (*confidentality*) – absence prozrazení důvěrných informací;
5. Integrita (*integrity*), tedy vyhýbání se nepovoleným změnám systému;
6. Udržovatelnost (*maintability*) je schopnost podstupovat změny a údržbu.

Mezi sekundární atributy pak patří atributy, které jsou založeny na kombinaci primárních atributů či na vztahu dependability systému k určitým konkrétním hrozbám. Mezi sekundární atributy patří například *nepopíratelnost* (dostupnost a integrita nějakého prvku systému), *robustnost* (odolnost systému proti chybným vstupním datům) a také ve vztahu k této práci důležitá *bezpečnost* (*security*). **Bezpečnost** můžeme podle [1] definovat jako “*absence neoprávněného přístupu ke stavu systému, resp. neoprávněné ošetření jeho stavu*“.

Z výše uvedeného odstavce lze vyvodit, že zásahem do činnosti stávajících distribuovaných sítí můžeme zvyšovat (resp. snižovat) dependabilitu systému jako celku. Tato práce se zaměří na některé atributy dependability distribuovaných sítí s federativní autentizací, konkrétně na **dostupnost**, **spolehlivost** a **bezpečnost**.

2.1.2 Selhání systému

Z hlediska dependabilty rozlišujeme podle [1] řetězec:

závada ⇨ chyba ⇨ selhání

Závadou (fault) rozumíme předpokládanou příčinu chyby, přičemž se může jednat například o pokus útočníka napadnout systém, technickou chybu v hardwaru, chybně napsaný program apod. Závada může být jednoduchá nebo složená z několika událostí a může jít také o řetězec několika závad. Ne všechny závady mohou způsobovat chybu – některé závady mohou být v latentním stavu a nemusí se nikdy projevit, nebo naopak mohou být aktivovány jinou závadou v již zmíněném řetězci závad.

Za **chybu** (error) označujeme podle [1] „*část celkového stavu systému, která může vést k následujícímu selhání služby*.“

I v tomto případě je možné, že dojde k řetězení chyb a ovlivňování částí systému (komponent) jednotlivými chybami tak, až dojde k selhání systému, resp. služby jako celku.

Selhání systému pak vznikne v okamžiku, kdy chyba nebo chyby systému (resp. jednotlivých komponent) způsobí, že úroveň dependability systému se sníží (systém de facto přestane být podle výše uvedené definice dependabilní).

2.1.3 Bezpečnostní selhání a bezpečnostní incident

Specifickým případem je pak tzv. **bezpečnostní selhání**. To může nastat v případě, že chyba nebo chyby v systému narušují stanovené **bezpečnostní politiky** systému.

Bezpečnostní politiky definujeme jako *souhrn pravidel systému, která specifikují pro daný systém významné bezpečnostní cíle (např. důvěrnost dat, integrita dat, nepopiratelnost autorství dat apod.)*

Jako jednotlivý **bezpečnostní incident** lze označit událost, která způsobila snížení úrovně dependability systému v souvislosti s činností uživatele nebo skupiny uživatelů ve vztahu k bezpečnostní politice daného systému. Bezpečnostní incidenty mohou být neúmyslné nebo úmyslné a mohou podle závažnosti incidentu vést ke vzniku bezpečnostní závady, chyby nebo selhání.

Bezpečnostní politika daného systému by měla mj. definovat jakým způsobem ošetřovat bezpečnostní incidenty.

2.2 Architektury autentizačních systémů

Současné počítačové sítě poskytují celou řadu služeb a funkcí a je zřejmé, že ne všichni uživatelé mají mít volný přístup k těmto službám. Proto bylo nutné vytvořit nástroje, které umožňují omezit přístup k některým službám jen pro některé uživatele. Jedná se o tzv. autentizační, autorizační a účtovací nástroje (služby). Tyto služby jsou zajišťovány **autentizačními protokoly**.

Autentizační protokoly slouží pro zajištění služeb v rámci tzv. **technologie AAA** (Authentization, Authorization, Accounting). Tato metoda poskytuje v současnosti základní úroveň obecného zabezpečení počítačových sítí.

Technologie AAA se opírá o tři základní pilíře:

- Autentizace = ověření totožnosti (identity) subjektu (uživatele, systému, procesu…),
- Autorizace = přidělení přístupových práv autentizovanému subjektu ke zdrojům systému podle výsledku autentizace či zařazení do skupiny apod.,
- Účtování = evidence činností autorizovaného subjektu v systému.

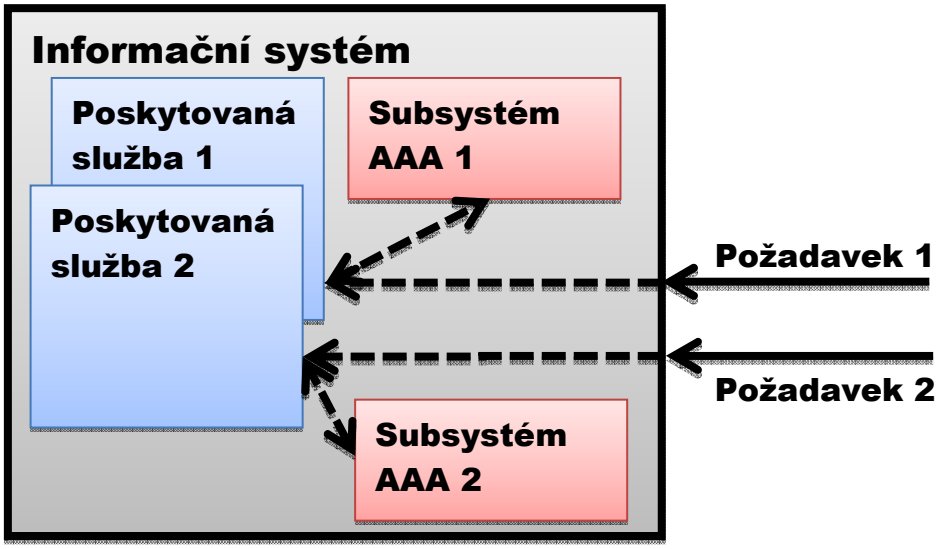
Ne všechny autentizační protokoly však nabízejí všechny výše uvedené nástroje a také kvalita provedení těchto nástrojů může být různá. Rozdíly jsou jednak mezi jednotlivými protokoly, jednak i v rámci jednoho protokolu je úroveň příslušného nástroje dána konkrétní konfigurací.

Autentizační systémy prošly během let postupným vývojem, takže lze podle [2] definovat tři základní architektury autentizačních systémů:

- lokální autentizační systémy,
- centralizované autentizační systémy,
- federativní autentizační systémy.

2.2.1 Lokální autentizační systémy

V tomto případě je autentizační subsystém součástí systému poskytujícího službu. Potřebné autentizační a další údaje o uživatelích jsou umístěny přímo v daném informačním systému. Jedná se o historické řešení, které je výhodné především tam, kde se neplánuje nasazení více služeb. Toto řešení je také výhodné pro svou nezávislost na ostatních systémech či na spolehlivosti systému poskytujícího autentizační službu. Dnes se využívá například tam, kde si vystačíme jen s nižší úrovní zabezpečení (resp. nevadí nám, že uživatelé budou nejspíše využívat jedno heslo pro mnoho systémů s lokální autentizací). Typickým příkladem jsou systémy webových služeb, například redakční systémy webových stránek (často s prefabrikovaným autentizačním subsystémem).



obr. 1 Lokální autentizační systém

Mezi **výhody** lokálních autentizačních systémů můžeme zařadit například:

- jednoduchost implementace do konkrétní aplikace,

- nezávislost činnosti aplikace na činnosti nějakého vnějšího autentizačního systému.

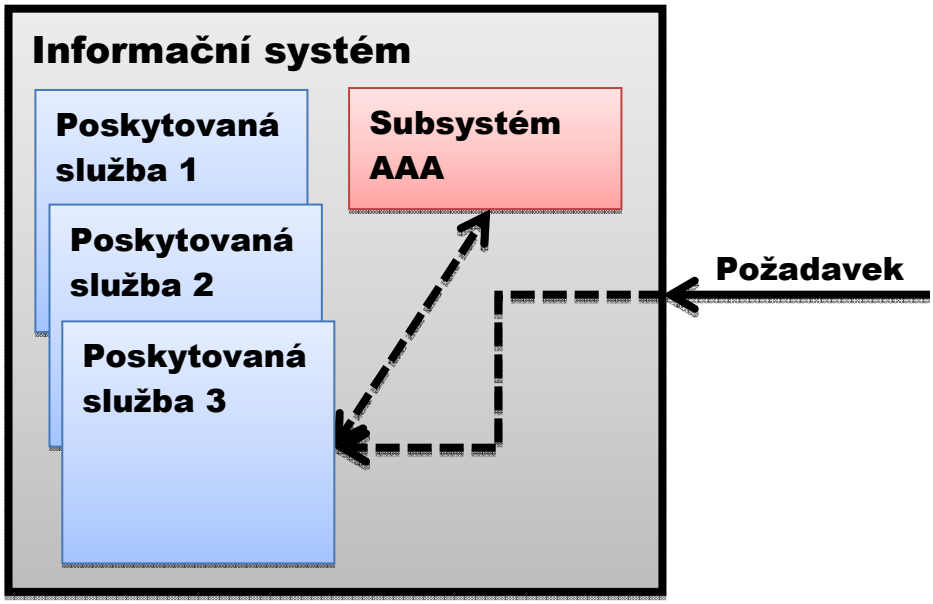
Naopak mezi **nevýhody** lokálních autentizačních systémů patří zejména:

- nižší bezpečnost systému,
- složitá správa autentizačních údajů uživatelů, pokud je nasazeno více podobných systémů v dané instituci,
- obtížná možnost integrace s jinými systémy,
- pravděpodobně vyšší vnitřní složitost informačního systému s lokálním autentizačním subsystémem.

2.2.2 Centralizované autentizační systémy

Informačnímu systému, kde je autentizační subsystém oddělen od služeb, které jej využívají, říkáme centralizovaný autentizační systém[ii], viz obr. 2.

Takový systém umožňuje nasadit pro autentizaci uživatelů vhodnou autentizační metodu[iii], která je pro uživatele jednoduchá a transparentní a umožňuje jim přístup ke všem službám daného IS. Přitom tento systém zajišťuje poměrně vysokou úroveň bezpečnosti všech služeb.



obr. 2 Centralizovaný autentizační systém

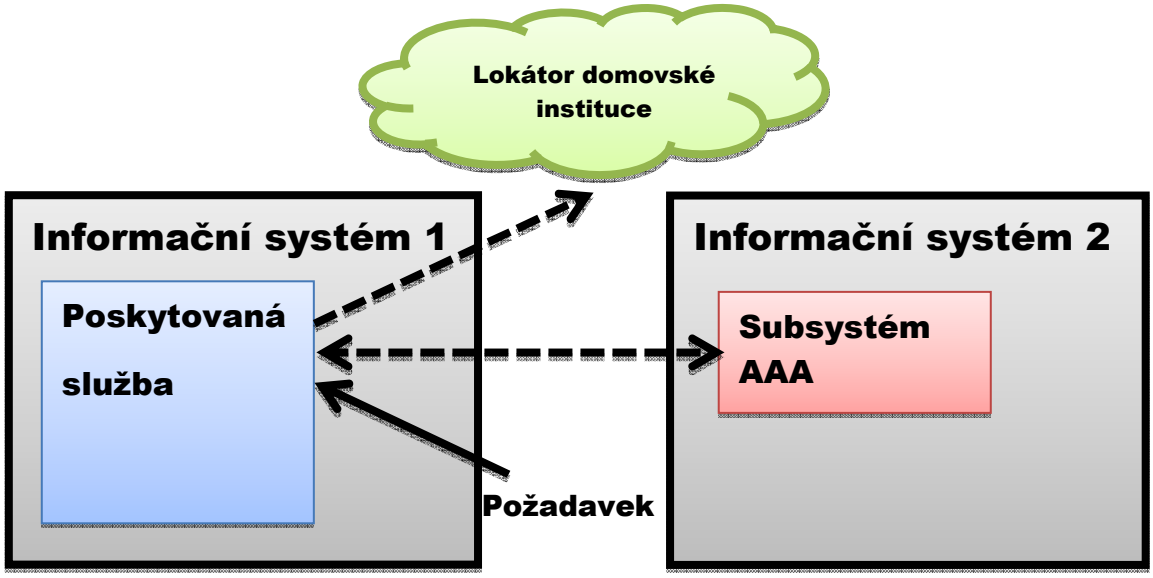
[ii] Mnoho centralizovaných AAA systémů je založeno na adresářovém protokolu LDAP (viz např. [46]), zmiňme například Microsoft Active Directory nebo Apache Directory Server.
[iii] Autentizační metody rozdělujeme: znalostí, vlastnictvím, vlastností a jejich kombinace, viz např. [46].

Mezi hlavní **výhody** centralizovaného autentizačního systému patří:

- centralizovaná správa autentizačních a dalších informací,
- možnost vyššího zabezpečení autentizačního subsystému,
- zjednodušení údržby subsystému,
- možnost nasadit propracovanější autentizační metodu.

Nevýhodou může být složitá implementace a následná správa autentizačního subsystému a bezpečnostní riziko při ztrátě nebo poškození autentizačních informací.

Typickým příkladem centralizovaných autentizačních systémů jsou dnešní běžně používané firemní informační systémy (například SAP).



obr. 3 Federativní autentizační systém

2.2.3 Federativní autentizační systémy

Jedná se o progresivní systém založený na federativním uspořádání institucí (viz obr. 3), které ve vzájemné kooperaci nabízejí uživatelům určitou službu nebo služby a sdílejí přitom vzájemně autentizační informace těchto uživatelů. Instituce sdružené ve federaci uplatňují společné postupy při autentizaci uživatelů a uznávání autentizačních výsledků.

Obrázek popisuje, jak dochází k poskytování služeb. Uživatel požaduje službu od určitého informačního systému, který nedisponuje jeho autentizačními údaji, ale je připraven za určitých podmínek službu poskytnout. Tento systém se

obrátí na domovský informační systém uživatele. Předá mu autentizační údaje vložené uživatelem a vyčká na jeho autentizační rozhodnutí. Na základě tohoto rozhodnutí, pak může přistoupit k poskytování služeb tomuto uživateli.

Aby taková architektura mohla fungovat, byla zavedena podslužba **lokátor domovské instituce**. Tato podslužba zajistí poskytovateli služeb na základě autentizačních údajů uživatele určení síťové adresy autentizačního subsystému, který provede vlastní autentizaci. Podslužba pracuje na podobné bázi jako DNS [3].

Výhody federativního autentizačního systému jsou:

- transparentnost a komfort systému pro uživatele,
- minimalizace duplicit autentizačních údajů uživatelů ve federaci,
- snížení náročnosti zajištění služeb pro uživatele odjinud.

Mezi **nevýhody** patří již značná složitost autentizačního systému, nutnost koordinace mezi subjekty a také de facto částečná potřeba společné infrastruktury (např. technické zajištění lokátoru domovské instituce).

2.3 Distribuovaná sít *eduroam*

Reálně fungujícím příkladem distribuované sítě s federativní autentizací je akademická bezdrátová síť ***eduroam***, kterou využijeme jako vhodné modelové prostředí pro nastíněné problémy a jejich řešení.

Motivací pro vznik sítě ***eduroam*** bylo umožnit uživateli jedné sítě přístup k internetu a případně ke službám své domovské sítě, i když k ní není fyzicky připojen. Organizátorem sítě ***eduroam*** v České republice je CESNET[1] a participují na ní další připojené organizace.

Ze systémového hlediska jsou distribuční sítí ***eduroam***u veřejné sítě. V současnosti je to v ČR především síť CESNET2, nicméně systém byl navržen tak, aby mohl pracovat nad libovolnou (i nedůvěryhodnou) sítí. Bezpečnostní roli v rámci ***eduroam*** zajišťuje autentizační a autorizační infrastruktura (AAI). Zásadní pozici v tomto hraje systém vzájemně propojených RADIUS[2] serverů jednotlivých organizací (členů federace) založený na autentizačním protokolu RADIUS. Významnou úlohu mají také další autentizační protokoly, které

zajišťují autentizaci uživatele při komunikaci s přístupovým bodem bezdrátové sítě (viz dále). Systém distribuované bezpečnosti je postaven především na protokolech IEEE 802.1X[3] a RADIUS, viz kapitola 4.2.

Důležitým dokumentem, popisujícím činnost sítě ***eduroam***, je takzvaná **Roamingová politika** [4].

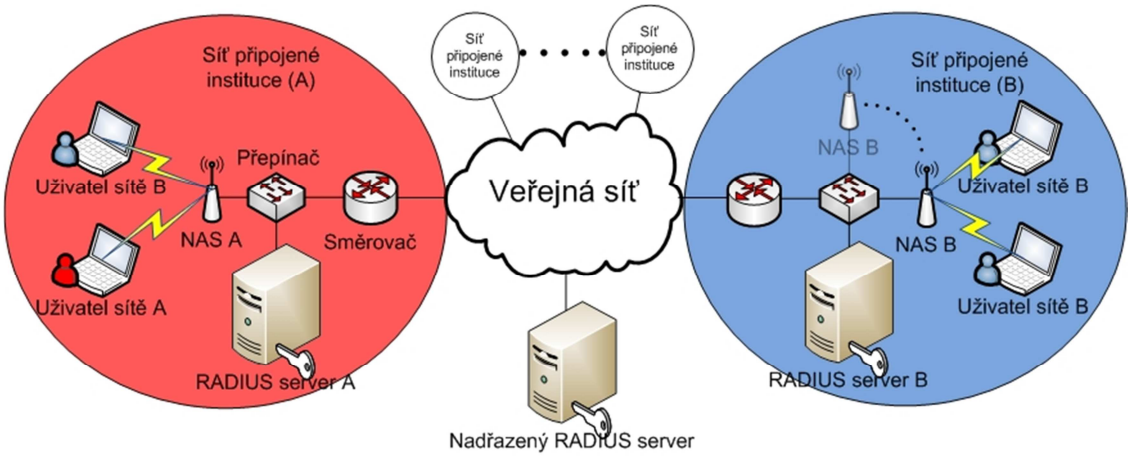
2.4 Roamingová politika v síti *eduroam*

Roamingová politika [4] je dokument, který upravuje vztahy mezi jednotlivými členskými institucemi české ***eduroam*** federace. Tento dokument definuje jednotlivé role institucí v ***eduroam*** federaci, upravuje připojení a odpojení členských institucí, řešení bezpečnostních incidentů, pravidla pro uchovávání auditních informací apod.

Základní role v ***eduroam*** federaci podle [4] jsou:

- správce ***eduroam*** federace,
- poskytovatel identity,
- poskytovatel zdrojů,
- uživatel.

Jednotlivé role mají definovaná práva a povinnosti. Zjednodušené schema sítě ***eduroam*** je patrné z obr. 4.



obr. 4 Schéma distribuované sítě *eduroam*

Správce federace je odpovědný za koordinaci dění ve federaci, připojování a odpojování členů federace, dodržování pravidel, připojení k mezinárodní

federaci ***eduroam***, správu nadřazených RADIUS serverů a jejich fungování. Dále provádí logování informací o autentizačním provozu a technickou podporu pro členy federace.

Poskytovatel identity je zobrazen na obr. 4 jako instituce A nebo B a jeho úkolem je vést databázi uživatelů sítě ***eduroam***, které eviduje ve své organizaci, a provádět jejich autentizaci ve vztahu ke všem členům federace. Dále provozuje své autentizační servery a zajišťuje jejich napojení na národní autentizační servery a zabezpečenou činnost. Samozřejmou součástí činností poskytovatele identity je logování informací o autentizaci, řešení bezpečnostních incidentů apod.

Poskytovatel zdrojů je instituce, která v rámci federace nabízí své služby. Jedná se především o poskytnutí konektivity, připojení k intranetovým aplikacím apod. Služby jsou poskytnuty pro daného uživatele pouze v případě jeho kladné autentizace u poskytovatele identity, jinak je přístup ke službám zamítnut. A to bez rozdílu toho, zda uživatel zadal nesprávné autentizační údaje, nebo došlo k technickým problémům na straně poskytovatele identity. Samozřejmou součástí činností poskytovatele zdrojů je logování informací o autentizačních informacích, řešení bezpečnostních incidentů apod.

Uživatel je na obr. 4 zobrazen jak pro domovskou síť A (červený), tak pro domovskou síť B (modrý). Uživatel musí mít právní vztah k poskytovateli identity a musí se řídit povinnostmi definovanými Roamingovou politikou a dalšími pravidly poskytovatele indentity a poskytovatelů zdrojů. Služby, které jsou poskytovány sítí ***eduroam***, jsou bezplatné, a tedy na ně není právní nárok, ale jsou poskytovány jako „best effort“. Přesný rozsah služeb pro uživatele definují pravidla příslušného poskytovatele služeb.

Z uvedeného je zřejmé, že jednotlivé logovací informace o autentizaci, o činnosti uživatelů a podobně jsou v systému ukládány distribuovaně bez vzájemného vztahu. Z Roamingové politiky dále plyne, že jejich ukládací doba je 6 měsíců a že si je jednotlivé subjekty poskytují ad hoc při řešení bezpečnostních incidentů. Stejně tak při technických problémech v síti nelze použít dostupné informace umístěné v jiném místě, než v domovské síti.

Z obr. 4 jsou patrné vztahy mezi jednotlivými subjekty, a tedy lze dovodit jednotlivé datové toky mezi nimi. Jedná se především o informace mezi klientským zařízením a NAS[4], mezi NAS a RADIUS serverem a mezi jednotlivými RADIUS servery. Zde se vyskytují kromě uživatelských dat jednak autentizační informace a jednak logovací záznamy. Tyto datové toky a vztahy mezi součástmi systému jsou definovány protokoly, které budou rozebrány v následujících kapitolách.

3 CÍLE DISERTAČNÍ PRÁCE

Mezi jednotlivými uzly v sítích s federativní autentizací (jako např. ***eduroam***) existuje, jak bylo popsáno výše, komunikace na bázi protokolu RADIUS, přičemž tento protokol se také využívá pro částečné logování činností uživatelů a prvků sítě. Na systémové úrovni se v případě takové sítě jedná o distribuovanou databázi, která poskytuje svým uživatelům služby. V podkapitole 2.4 bylo uvedeno, že na tyto služby není nárok, a navíc je zřejmé, že v předloženém systému není zaveden žádný mechanismus pro podporu spolehlivosti, tedy například při výpadku autentizačního serveru není služba dostupná. To odpovídá deklarované filozofii sítě na úrovni „best effort“, což může být do budoucna omezujícím faktorem pro další rozvoj sítí s federativní autentizací.

Problémy mohou vznikat také při pokusech uživatelů zneužít k nepovoleným účelům či jinak napadnout síť poskytovatele služeb nebo jeho síť využít k útokům na jiné sítě. O takovém chování uživatele se poskytovatel identity tohoto uživatele nemusí vůbec dozvědět, natož aby mohl účinně proti němu zasáhnout.

Obecným cílem této práce je **nalezení mechanismů pro zvýšení spolehlivosti a bezpečnosti protokolu RADIUS** v distribuovaných sítích s federativní autentizací a využít přitom prostředí modelové sítě ***eduroam***.

V souvislosti s tím bude třeba **navrhnout úpravu protokolů a/nebo algoritmů zúčastněných prvků**, případně interních bezpečnostních dokumentů provozovatelů distribuovaných sítí (Roamingové politiky) v souvislosti s výše uvedenými mechanismy pro podporu spolehlivosti a bezpečnosti distribuovaných sítí s federativní autentizací, včetně případného rozšíření vazby mezi poskytovateli zdrojů a identit a se zohledněním ochrany soukromí uživatelů daných sítí.

3.1 Problematika spolehlivosti protokolu RADIUS

Jak už bylo naznačeno výše, výpadky sítě mezi autentizačními servery, mohou způsobit dočasné omezení služeb. Tomu by se dalo vyhnout, pokud by bylo možné využít při autentizaci uživatele z důvodu výpadku jeho domovské sítě autentizační informace z jiných autentizačních serverů, pokud jsou tyto informace na jiném autentizačním serveru k dispozici. Aktuální RADIUS protokol to však neumožňuje.

S využitím informací z více zdrojů ovšem může nastat problém šíření odlišných autentizačních informací o daném uživateli, čímž nastává problém, jak věrohodně rozhodnout o autentizaci uživatele v síti daného poskytovatele služeb.

Dílčím cílem práce je najít vhodný algoritmus pro řešení problému s výpadkem domovského autentizačního RADIUS serveru, který zajistí autentizaci uživatele z jiného autentizačního serveru po dohodě mezi uzly a navrhnout inovaci protokolu RADIUS v tomto směru. Předložený algoritmus by měl být využitelný i v síti s velkým množstvím uzlů a měl by být schopen činnosti v asynchronním distribuovaném systému[iv].

V této souvislosti je třeba najít vhodný simulační nástroj a ověřit činnost nalezeného algoritmu ve spolupráci s protokolem RADIUS v simulovaném prostředí.

3.2 Problematika bezpečnosti protokolu RADIUS

Je zřejmé, že uživatelé mohou své chování v různých sítích měnit, například jinou než domovskou síť mohou zneužívat k útokům na vnitřní zdroje této sítě nebo na externí sítě. Pokud k tomu dojde, odhaleného útočníka lze nejspíše manuálně vyřadit z databáze poskytovatele identit v součinnosti s tamějším administrátorem, ale systémové opatření na úrovni protokolu RADIUS dosud neexistuje. Poskytovatel identity se, jak už bylo uvedeno, teoreticky nemusí o problémovém uživateli vůbec dozvědět. Z hlediska bezpečnosti je třeba rozšířit

[iv] Viz subkapitola 4.3.1

protokol RADIUS o architekturu pro jistou formu sledování činnosti a šíření informací o chování jednotlivých uživatelů v dílčích sítích jednotlivých poskytovatelů služeb a identit.

Informace o uživatelích v systému už nyní v určité kvalitě shromažďují jednotlivé zainteresované RADIUS servery v rámci svých účtovacích informací. Je třeba analyzovat, zda a jak je možné využít existující accounting informace k případnému omezení služeb u uživatelů s „podezřelým chováním“. Zaměříme se na definici, co je „podezřelé chování“ uživatelů. To je možné definovat na základě dostupných accounting informací, lze zřejmě také navrhnout rozšíření accounting záznamů nebo využití logovacích informací z jiných zdrojů, např. z dostupných *Intrusion Detection Systemů*. Spolupráce již nasazených IDS a protokolu RADIUS by mohla nabízet v tomto ohledu významné zlepšení.

Dílčím cílem je navrhnout, jak vylepšit protokol RADIUS o možnost sledování chování uživatelů v distribuované síti s federativní autentizací se zaměřením na bezpečnost systému, a s využitím volitelných atributů protokolu RADIUS vyřešit, jak předávat tyto informace mezi jednotlivé autentizační servery.

4 TEORETICKÉ POZNATKY

4.1 Druhy autentizačních protokolů

Autentizační protokoly lze z hlediska síťového modelu ISO/OSI zařadit do dvou skupin. Autentizační protokoly druhé (linkové) vrstvy a autentizační protokoly sedmé (aplikační) vrstvy.

Mezi autentizační protokoly linkové vrstvy patří především protokol 802.1X a PPP protokol (Point-to-Point Procokol, RFC1661 [5]). Tento protokol slouží především pro autentizaci dvou uzlů mezi sebou (např. mezi klientem a přístupovým serverem [NAS]). Protokol PPP využívá služeb dalších linkových autentizačních protokolů. Jsou to především protokoly PAP (Password Authetication Protocol, RFC1334 [6]), CHAP (Chalenge Authentication Protocol, RFC1994 [7]) a EAP (Extensible Authentication Protocol, RFC3748 [8]). Protokol EAP má množství využití a také několik variant.

Vzhledem k tomu, že autentizační protokoly linkové vrstvy nemohou být směrovány, slouží pro komunikaci mezi dvěma přímo propojenými uzly. To je předurčuje jako nástroj pro komunikaci (autentizaci) s přístupovým serverem, kterým může být např. i bezdrátový přístupový bod sítě podle standardu IEEE 802.11 [9]. Pro autentizaci v rámci distribuované sítě jako celku nejsou vhodné z výše uvedeného důvodu.

Autentizační protokoly sedmé vrstvy ISO/OSI jsou mnohem propracovanější, robustnější a většinou nabízejí všechny tři pilíře technologie AAA (viz subkapitola 2.2). Mezi tyto protokoly v současné době patří RADIUS [10], DIAMETER [11], TACACS+ [12] a Kerberos [13].

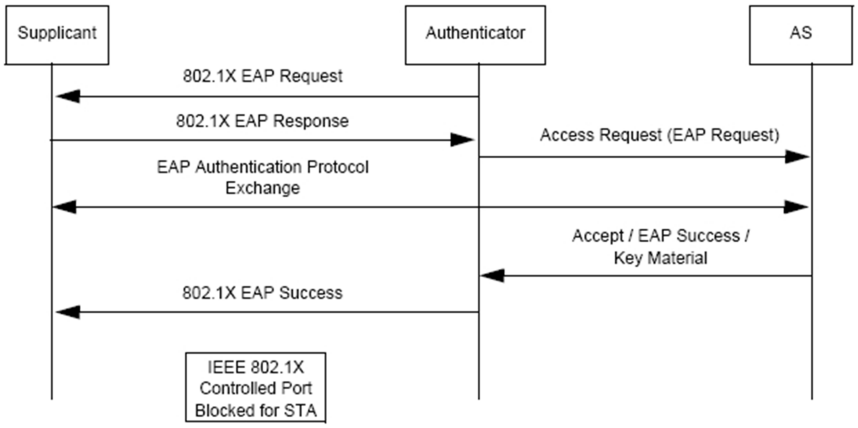
4.1.1 Bezpečnost podle standardu IEEE 802.1X

Standard IEEE 802.1X je obecným bezpečnostním standardem počítačových sítí, který stanovuje obecné postupy a metody pro zvýšení bezpečnosti datových přenosů.

Protokol 802.1X předpokládá, že připojení uživatele na fyzický nebo virtuální port aktivního prvku druhé vrstvy nelze považovat za bezpečné a obecné

nastavení portu je tak ve stavu „zavřeno“. Pro otevření portu je třeba, aby připojený uživatel absolvoval autentizační proceduru, která po úspěšné autentizaci umožní uživateli přístup do sítě. V rámci příslušného portu je povolen pouze provoz autentizačního protokolu PPP s příslušným podprotokolem (např. EAP), který slouží k autentizaci připojeného uživatele.

Průběh autentizace je zobrazen na obr. 5. Uživatel zde vystupuje v roli **supplicant** (prosebník) přístupový server NAS jako **authenticator**. AS označuje **autentizační server**.



obr. 5 Princip autentizace s použitím standardu 802.1X **(převzato z [14])**

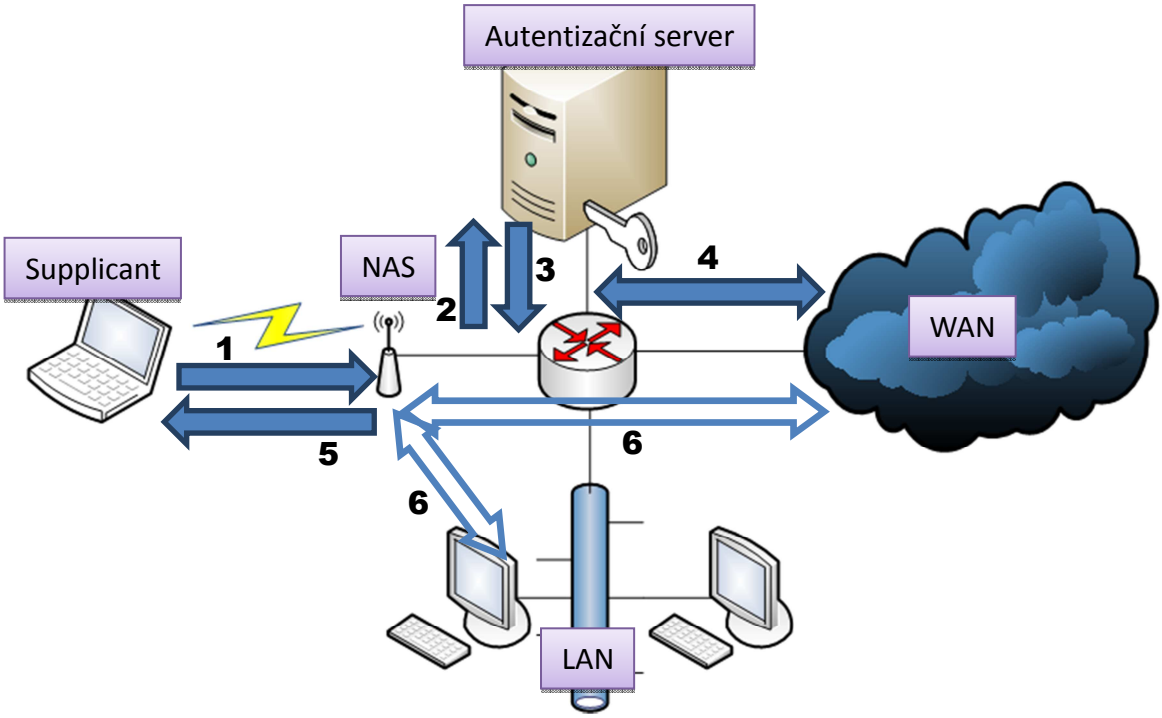
4.2 Protokol RADIUS

4.2.1 Vlastnosti a činnost protokolu

Protokol RADIUS (Remote Access Dial-in User Server) je distribuovaný autentizační protokol typu klient/server, který pochází z roku 2000 od společnosti Livingston Enterprises. Je to protokol podporující technologii AAA, přičemž autentizace a autorizace jsou shrnuty v RFC2865 [10] a auditování je v samostatném RFC2866 [15]. Protokol podporuje spolupráci s protokoly TACACS+ a Kerberos a je součástí nejrůznějších systémů, které požadují poměrně vysoký stupeň zabezpečení.

Výhodou tohoto protokolu – kromě velkého rozšíření – je také otevřenost standardu. V rámci činnosti protokolu RADIUS vystupují čtyři základní subjekty, viz obr. 6. Jedná se o uživatele, který požaduje přístup do sítě a který je

označován jako **supplicant** (prosebník). Dále zde je **klient**, kterým je příslušný přístupový server NAS (např. přístupový bod sítě WiFi). Neposledním je **autentizační server**, jenž obsahuje serverovou část systému RADIUS, a je na něm provozována **databáze** klientů, kteří mají mít přístup do sítě, vůči které je prováděna autentizace.



obr. 6 Přístup do sítě s využitím protokolu RADIUS

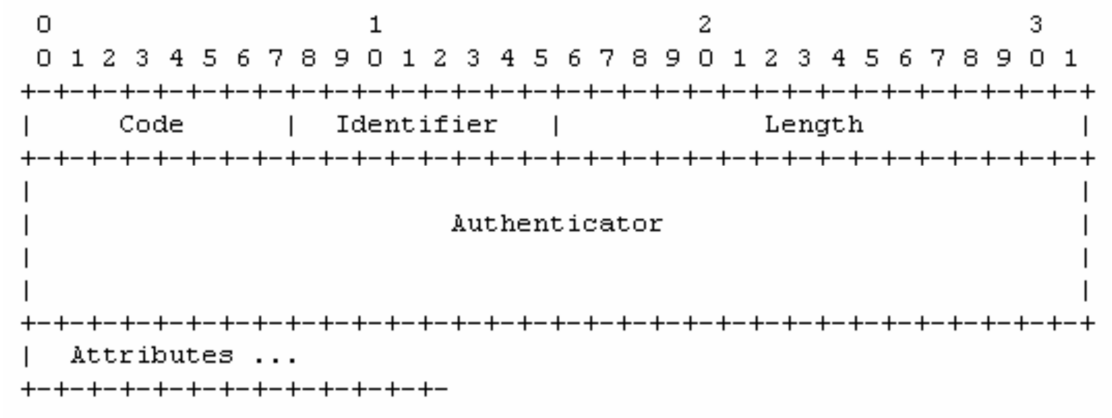
Protokol RADIUS pracuje nad protokolem UDP[v] [16] (port 1812 či 1645), pomocí kterého probíhá přenos zpráv mezi přístupovým serverem a zabezpečovacím serverem. Jak již bylo naznačeno výše, propojeny jsou autentizace s autorizací, přičemž accounting lze provozovat odděleně. Server přijímá od klientů autentizační informace (které získají protokolem PPP od supplicantů) a posílá jim zpět rozhodnutí o autentizaci (kladné nebo záporné). Autentizace probíhá na základě uživatelského jména a hesla, které je zasíláno zabezpečené pomocí algoritmu MD5 [17].

[v] Ačkoliv standard RFC2865 vysvětluje, proč byl pro přenos protokolu RADIUS využit protokol UDP, dnes se již toto řešení považuje za překonané a využívá se buď aplikačního protokolu RadSec [22], který pro autentizaci mezi servery využívá certifikátů, komunikaci plně šifruje a využívá transportní protokol TCP [32] (implementace Radiator [45]) nebo RADIUS proxy [44], kterou lze nasadit před jakoukoliv implementaci RADIUS (např. FreeRADIUS [30]) a která též využívá TCP.

Na obr. 6 je schematicky znázorněn průběh autentizace. V prvním kroku požádá supplicant o autentizaci příslušný NAS protokolem PPP-EAP, tento požadavek předá NAS protokolem RADIUS příslušnému RADIUS serveru (bod 2), který jej buď vyřídí sám (body 3 a 5) nebo jej předá k vyřízení příslušnému dalšímu RADIUS serveru podle realmu uživatele (body 4 a 5). V dalším kroku je pak v případě kladné autentizace otevřen požadovaný port NAS. Tím je umožněna uživateli komunikace do LAN či WAN (bod 6).

4.2.2 Typy a formát paketů

Do zprávy transportního protokolu UDP nebo TCP je zabalena vždy právě jedna RADIUS zpráva (viz obr. 7) a cílový port je nastaven na 1812, přičemž při odpovědi jsou cílový a zdrojový port prohozeny. Pole uvedená na obr. 7 budou přenášena zleva doprava.



obr. 7 Formát zprávy protokolu RADIUS (převzato z [10])

Zpráva RADIUS protokolu se skládá z několika polí, která jsou následující:

Code

Toto pole je dlouhé jeden oktet a identifikuje typ RADIUS paketu; může nabývat pro zprávy protokolu RADIUS hodnoty definované v odpovídajících RFC dokumentech (viz odstavec 4.2.2.1) a registrované institucí IANA[5]. Pokud by serveru RADIUS dorazil paket s jinou než definovanou hodnotou, bude zahozen.

Identifier

Pole Identifier obsahuje pseudonáhodně vygenerované číslo, které slouží pro rozpoznávání případných duplicitních paketů a je dlouhé jeden oktet.

Lenght

Tento parametr obsahuje délku paketu včetně všech polí. Je dlouhý dva oktety a smí nabývat hodnot od 20 do 4095. Oktety překračující maximální délku paketu určenou tímto parametrem budou považovány za výplň a ignorovány. Pakety kratší než dolní mez budou zahozeny.

Authenticator

Jedná se o 16 oktetů dlouhé pole, které slouží k autentifikaci paketů mezi NAS a AS. Toto pole se vytváří na základě sdíleného tajemství mezi NAS a RADIUS serverem s využitím algoritmu MD5.

Attributes

Pole Attributes obsahuje přenášené atributy, které de facto tvoří hlavní data příslušného paketu. Typy a význam některých atributů bude popsán dále.

4.2.2.1 Přehled typů paketů

Protokol RADIUS disponuje až 255 typy paketů (pole Code viz obr. 7 může obsahovat hodnoty 1-255). Registrací jednotlivých typů paketů a přidělováním jejich čísel se zabývá IANA. V současné době je obsazeno přibližně padesát typů paketů, zbytek je volný, výjimečně rezervovaný.

Není smyslem této práce uvádět celý výčet typů paketů, zde je výběr následujících nejzajímavějších (uvedené číslo je decimální hodnota pole Code):

1 Access-Request
2 Access-Accept
3 Access-Reject
4 Accounting-Request
5 Accounting-Response
10 Accounting-Message
11 Access-Chalenge

Vzhledem k jejich označení není zřejmě nutné tyto typy paketů detailněji popisovat; accounting pakety využívá RADIUS accounting server, access pakety jsou pak použity standardním RADIUS protokolem.

4.2.2.2 Volitelné atributy

Podobně jako lze v protokolu RADIUS definovat nové typy paketů, je možné definovat také jednotlivé atributy. Protokol nabízí opět celkem 255 možných atributů. Obsazených je aktuálně přibližně 140 atributů, zbytek je převážně volný.

4.3 Rozhodování v distribuovaných systémech

V rámci řešené problematiky dochází ke komunikaci mezi jednotlivými uzly distribuované sítě a k uzavírání rozhodnutí o autentizaci uživatelů. Z tohoto důvodu bylo nutné zabývat se problematikou rozhodování v distribuovaných systémech, zejména problematikou rozhodování v distribuovaných systémech s nespolehlivými uzly sítě. Tato problematika také souvisí s PKI (Public Key Infrastructure) [18], což je rozšířená metoda zajišťování síťové bezpečnosti, kterou lze využít v distribuovaných sítích. PKI poskytuje základní bezpečnostní úroveň, tj. bezpečný přenosový kanál a autentizaci komunikujících stran. Neznamená to však, že zpráva vybavená odpovídajícím certifikátem (podepsaná) je automaticky pravdivá. Je možné, že kompromitovaný uzel zašle podepsanou nepravdivou zprávu. PKI se také nezabývá hledáním společného konsensu. Na tuto problematiku se zaměřuje teorie **rozhodování v distribuovaných systémech** [19], [20].

4.3.1 Obecný úvod

Distribuovaný systém je systém nezávislých uzlů, který poskytuje uživateli dojem jednotného systému. [21]

Distribuované systémy obecně rozdělujeme na **synchronní** a **asynchronní**, přičemž podle [20] platí, že v asynchronním systému proces pracuje nezávisle na ostatních, zpoždění na komunikační lince je konečné, čas zpracování je započítáván do času přenosu, takže jej lze zanedbat a neexistuje v tomto systému žádná entita, která by poskytovala nějaký paměťový prostor nebo globální čas sdílený celou množinou procesů, které jsou součástí daného

systému. Distribuovaná počítačová síť patří mezi typické příklady asynchronních systémů s proměnnou topologií.

Aby bylo vůbec možné zajistit shodu (konsensus) v takovém systému, je třeba podle [20] zajistit několik podmínek. Zejména je třeba zajistit **uspořádané atomické rozesílání** zpráv. Tedy použít vhodný protokol, který zaručuje, že při odeslání zpráv n_1 *a* n_2 budou tyto zprávy přijaty všemi uzly v nezměněném pořadí. Takovým protokolem je hojně rozšířený protokol TCP, který ovšem není v rámci protokolu RADIUS standardně používán. Nicméně současné implementace protokolu RADIUS využívají subprotokolu RadSec [22], který pracuje výhradně na bázi protokolu TCP, a tedy lze předpokládat, že podmínka uspořádaného atomického rozesílání je dodržena.

V rámci problematiky řešené v této práci je třeba zajistit určitou míru shody mezi jednotlivými uzly, a tak zvýšit spolehlivost systému, resp. jeho vyšší odolnost proti poruchám. Při tom je třeba vzít do úvahy obecnou míru nespolehlivosti jednotlivých uzlů a možnost šíření chybových zpráv. Ne všechny uzly distribuovaného systému lze považovat za tzv. loajální, některé uzly mohou být zrádci. Za zrádce považujeme takový uzel (proces), který odesílá nesprávné informace nebo neodesílá žádné, a to záměrně nebo v důsledku poruchy uzlu či přenosové cesty.

Existují tři základní přístupy k řešení distribuovaného konsensu [21]:

1. **byzantský konsensus,**
2. **obecný konsensus,**
3. **interaktivní konzistence.**

V prvním případě platí, že pokud iniciační uzel zvolí nějakou hodnotu, rozešle ji všem uzlům, přičemž správně fungující (loajální) uzly se musí shodnout na stejné hodnotě. Tato hodnota bude shodná s iniciační, pokud je iniciační uzel loajální.

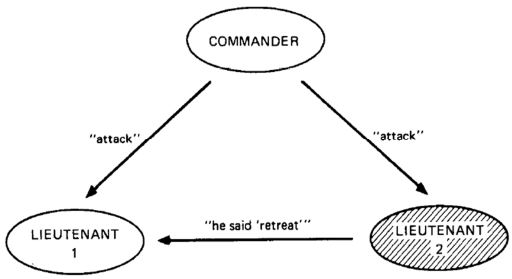
Druhý případ lze využít v případě, kdy každý uzel má nějakou iniciální hodnotu. Pak je třeba, aby se všechny loajální uzly shodly na nějaké hodnotě, přičemž pokud je jejich iniciální hodnota stejná, shodnou se právě na ní. Existuje celá řada algoritmů, které tento problém řeší. Některé z nich jsou uvedeny v [23].

V posledním případě má opět každý uzel nějakou iniciální hodnotu, přičemž loajální uzly se musí shodnout na společném vektoru. Hodnoty položek tohoto vektoru se shodují s iniciálními hodnotami jednotlivých uzlů. V řešené problematice existuje možné využití pro byzantský konsensus.

4.3.2 Byzantský konsensus

Byzantský konsensus řeší problém se shodou několika uzlů na jednom (navrženém) stavu [24], kdy v systému mohou vznikat náhodné netriviálně odhalitelné poruchy. Problém lze vysvětlit na příkladu **byzantských generálů**, kteří vydávají rozkazy armádě, ve které jsou zrádci. Jedná se tedy o distribuovaný systém velitelů, kteří mohou dostávat protichůdné rozkazy podle toho, zda je vydává subjekt loajální či zrádce. Tento algoritmus by mělo být možné za výše uvedených podmínek využít i v případě distribuované sítě, kde je možné předpokládat určitou míru nedůvěry nebo poruch mezi jednotlivými uzly a tedy možnost šíření nesprávných (i protichůdných) informací.

Pro vysvětlení problému s dosažením dohody mezi procesy v asynchronních distribuovaných systémech uveďme příklad se třemi procesy, kdy jeden z nich může být nespolehlivý, viz obr. 8. Pokud spolehlivý proces přijme během komunikace mezi procesy dvě rozdílné informace, není jisté, zda vznikla chyba od původně vysílajícího procesu nebo při předávání zprávy druhým procesem. Stejně tak, pokud proces přijme dvě stejné hodnoty, nemůže si být jistý, zda původně odeslané hodnoty byly skutečně totožné.

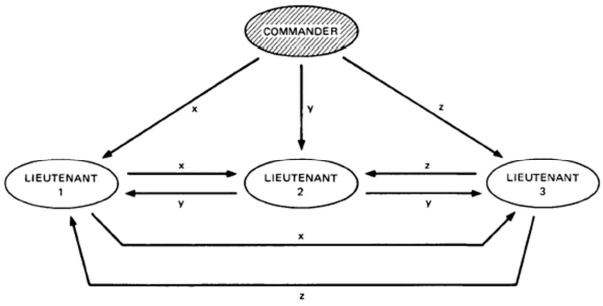


obr. 8 Tři uzly, zrádce důstojník (převzato z [24])

V tomto případě řešení neexistuje, a lze dokázat [24], že obecně neexistuje v případech, když $p \leq 3z$.

Definice proměnných: p je počet uzlů (procesů) a z je počet zrádců.

V případě čtyř uzlů, kdy zrádcem je například generál (podle obr. 9), pak buď rozešle tři různé rozkazy a uzly se shodnou, že generál je zrádce, nebo rozešle alespoň dva stejné rozkazy a pak se uzly shodnou na většinovém rozkazu.



obr. 9 Čtyři uzly, zrádce generál (převzato z [24])

Řešení problému existuje a lze podle [24] dokázat, že obecně existuje ve všech případech, kdy

$$\boldsymbol{p \geq 3z + 1} \qquad \textbf{(1)}$$

Protože není obecně známo, kolik je v systému zrádců, předpokládáme, že podmínka (1) splněna není a pak jednorázová výměna zpráv nemůže zajistit dosažení dohody mezi uzly [20]. Je třeba provést více výměn zpráv mezi uzly, tzv. **fází**. Z uvedeného příkladu plyne, že dostačující počet fází je dvě, a tedy lze dokázat, že počet fází výměn zpráv k pro nalezení řešení je obecně $k = z + 1$. Z toho ovšem plyne nutnost velkého množství přenesených zpráv mezi všemi uzly a tím i nemalé zatížení sítě při větším počtu uzlů.

Lze podle [20] dokázat, že tato cesta pro dosažení dohody generuje obecně exponenciální počet vyměňovaných zpráv v závislosti na počtu uzlů v síti. Základem činnosti běžného exponenciálního algoritmu je, že iniciální uzel rozešle zprávy všem ostatním uzlům (tedy $p - 1$ zpráv v první fázi; p je počet uzlů). V dalších fázích pak vždy všechny uzly, které v předchozí fázi obdržely nějakou zprávu, rozešlou zprávy všem ostatním uzlům kromě těch, od kterých zprávy obdržely. Ve druhé fázi bude rozesláno tedy $p - 2$ zpráv každým uzlem. Algoritmus se rekurzivně opakuje a v další fázi pak uzly rozešlou $p - 3$ zpráv atd. až do $k = z + 1$. Z toho je zřejmé, že pro jednoho zrádce je počet rozeslaných zpráv m

$$m = (p-1) + (p-1)(p-2) = p^2 - 2p + 1 \quad \textbf{(2)}$$

a pro dva zrádce pak

$$m = (p-1) + (p-1)(p-2) + (p-1)(p-2)(p-3) = p^3 - 5p^2 + 9p - 5 \quad \textbf{(3)}$$

Lze pak dovodit, že počet zpráv obecně je řádu

$$O(p^{z+1}) \quad \textbf{(4)}$$

Nárůst množství zpráv při růstu počtu (nespolehlivých) uzlů je tak značné.

Každá platná přijatá zpráva obsahuje jednoznačnou identifikaci uzlů, kterými prošla, a tedy je zřejmé, ze které pochází fáze. Na konci pak platná zpráva obsahuje nejvýše k identifikací, všechny jsou různé a identifikace daného uzlu se v nich nevyskytuje. Takovou zprávu pak může kterýkoliv uzel využít pro stanovení výsledku rozhodnutí. Míra „spolehlivosti“ informace obsažené v této zprávě je sice vysoká, nicméně v subkapitole 5.6 bude rozvedeno, proč je výše popsaný algoritmus pro nasazení ve vylepšeném protokolu RADIUS nevhodný. Jeho použití by přineslo především neakceptovatelný nárůst počtu přenesených zpráv, který je navíc výrazně závislý na počtu uzlů a na počtu výpadků při přenosu zpráv.

4.4 Petriho sítě

Aby bylo možné splnit cíle disertační práce, přičemž je mimo rozsah práce provést implementaci a nasazení inovovaného protokolu do provozu, bylo třeba najít vhodné simulační prostředí a vytvořit model inovovaného protokolu. Pro modelování a simulaci činnosti protokolu RADIUS v distribuovaném prostředí se ukázaly jako vhodný nástroj tzv. Petriho sítě [25]. Jedná se o dobře popsaný, rozvinutý a vyzkoušený nástroj pro modelování různých procesů včetně distribuovaných protokolů.

Dílčím problémem bylo nalezení vhodné softwarové implementace, která by byla dostatečně propracovaná a popsaná, aby zvládla modelovat hierarchické modely. Autor otestoval několik aplikací pro tvorbu modelů pomocí Petriho sítí, ale jako vhodná byla vybrána aplikace CPN Tools [26]. Jedná se o poměrně propracovanou aplikaci, která je zaměřena na barevné Petriho sítě. Výhodou je

dostupná dokumentace a množství příkladů. Tuto volně šiřitelnou aplikaci vyvíjí Eindhoven University of Technology v Holandsku.

4.4.1 Úvod do Petriho sítí

Petriho sítě vytvořil v roce 1962 Carl Adam Petri [25] a jedná se matematickou reprezentaci diskrétních distribuovaných systémů realizovanou formou orientovaných bipartitních grafů s ohodnocením [27]. Petriho sítě vznikly jako rozšíření modelovacích schopností konečných automatů [28]. V průběhu let se Petriho sítě postupně rozvíjely podle potřeb jejich uživatelů, přičemž dnes existuje značné množství různých variant Petriho sítí – časované, stochastické a také tzv. barevné Petriho sítě (Colored Petri Nets, CPN) [29].

Základem obecných Petriho sítí jsou **místa**, **přechody**, **hrany** a **tokeny**. Tokeny jsou prvky sítě umístěné v místech a přenáší se vytvořenou sítí v závislosti na stavu jednotlivých přechodů a hran. Místa obsahují určité množství tokenů, přechody provádějí s tokeny požadované operace a prostřednictvím hran se tokeny v Petriho síti přenášejí. Hrany mohou být vždy jen mezi místem a přechodem, nikoliv mezi dvěma místy či dvěma přechody.

Jako *síť* označujeme podle [27] trojici $N = (P, T, F)$, když *P* a *T* jsou disjunktní množiny a $F \subseteq (P \times T) \cup (T \times P)$ je binární relace. *P* je *množina míst* (places) sítě *N, T* je *množina přechodů* (transitions) a relace *F* je *tokovou relací* (flow relation) sítě *N*, označovanou též jako „hrany“.

Pokud vytvoříme bipartitní orientovaný graf reprezentací relace *F*, můžeme jej označit jako *graf sítě*. V grafu jsou místa kreslena obvykle ve tvaru kružnic nebo oválů a přechody ve tvaru obdélníků (příp. úseček).

Příklad sítě

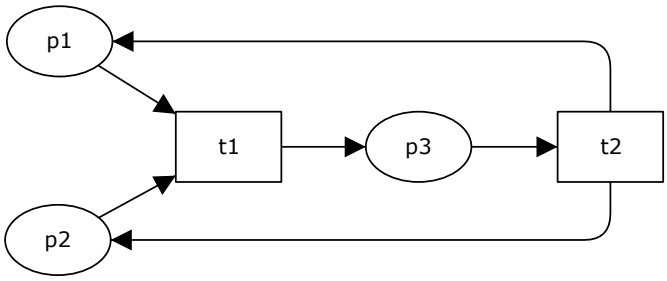
Mějme $N = (P, T, F)$, kde

$$P = \{p1, p2, p3\}$$

$$T = \{t1, t2\}$$

$$F = \{\langle p1, t1\rangle, \langle p2, t1\rangle, \langle p3, t2\rangle, \langle t1, p3\rangle, \langle t2, p1\rangle, \langle t2, p2\rangle\}$$

Graf takové jednoduché sítě bez prvků (tokenů) je na obr. 10.



obr. 10 Schéma sítě N

Vstupní a výstupní množinu prvků sítě *N* definujeme takto:

$^{\bullet}x = \{y | yFx\}$ se označuje jako *vstupní množina* (*preset*) prvku *x* a

$x^{\bullet} = \{y | xFy\}$ se označuje jako *výstupní množina* (*postset*) prvku *x*.

Pro každé $x, y \in (P \cup T)$ platí: $x \in {}^{\bullet}y \Leftrightarrow y \in x^{\bullet}$.

Petriho síť z obr. 10 můžeme popsat například vztahy ${}^{\bullet}t1 = \{p1, p2\}$ pro preset přechodu t1 a $p3^{\bullet} = \{t2\}$ pro postset místa p3.

4.4.2 Barevné Petriho sítě

Barevné Petriho sítě (CPN), (např. [29]) patří mezi vysokoúrovňové Petriho sítě a vyznačují se především tím, že dovolují uživateli modelovat značnou škálu problémů. Je to dáno tím, že CPN mohou obsahovat tokeny více typů (např. barev) a k nim je pak možné různě přistupovat. Tokeny mohou být např. číselné či složené z řetězce znaků. Klíčovou součástí barvených Petriho sítí jsou, kromě výše uvedených míst, hran, přechodů atd., tzv. **multimnožiny**, kde multimnožina *m* nad množinou *S* je funkce $m : S \rightarrow \mathbb{N}$, kde *m(s)* značí počet výskytů prvku *s* v multimnožině *m*. Multimnožinu *m* lze zapsat formální sumou $\sum_{s \in S} m(s)'s$. Například multimnožina obsahující tři výskyty prvku Q a dva výskyty prvku P bude zapsána jako 3'Q+2'P. Pro multimnožiny jsou definovány operace a predikáty viz např. [27].

Obecně lze nehierarchickou barevnou Petriho síť (CPN) definovat jako n-tici:

CPN = (Σ, P, T, A, N, C, G, E, I), kde

- *Σ* je konečná množina konečných neprázdných typů označovaných jako množina barev,
- *P* je konečná množina míst,

- *T* je konečná množina přechodů, když $P \cap T = \emptyset$,
- *A* je konečná množina hran, když $A \cap P = A \cap T = \emptyset$,
- *N* je uzlová funkce $N : A \to P \times T \cup T \times P$,
- *C* je funkce barev $C : P \to \Sigma$,
- *G* je funkce strážních podmínek $G : T \to EXPR$ taková, že

$$\forall t \in T : Type\big(G(t)\big) = \mathbb{B} \wedge Type\Big(Var\big(G(t)\big)\Big) \subseteq \Sigma,$$

- *E* je funkce hranových výrazů $E : A \to EXPR$ taková, že:

$$\forall a \in A : Type\big(E(a)\big) = C\big(p(a)\big)_{MS} \wedge Type\Big(Var\big(E(a)\big)\Big) \subseteq \Sigma,$$

přičemž *p(a)* je místo v *N(a)*,

- *I* je inicializační funkce $I : P \to CEXPR$ taková, že:

$$\forall p \in P : Type\big(I(p)\big) = C(p)_{MS}.$$

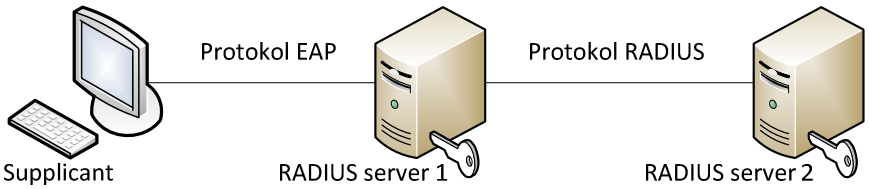
Ve výše uvedeném je $Type(v)$ označení pro typ proměnné v. Je-li V množina proměnných, pak $Type(V) = \{Type(v) | v \in V\}$. Typ výrazu *expr* je označen $Type(expr)$. Množina proměnných výrazů *expr je označena jako* $Var(expr)$. Jako uzavřený výraz označujeme výraz *expr* bez proměnných, tj. $Var(expr) = \emptyset$. Označme *EXPR* množinu všech výrazů přípustných ve zvoleném inskripčním jazyku. Označme dále *CEXPR* ⊆ *EXPR* množinu všech uzavřených výrazů přípustných ve zvoleném inskripčním jazyku a $\mathbb{B} = \{true, false\}$.

5 NÁVRH INOVACE PROTOKOLU RADIUS Z HLEDISKA SPOLEHLIVOSTI

5.1 Experimentální počítačová síť

Vzhledem ke stanoveným cílům práce bylo nutné nejprve vytvořit prostředí pro simulaci stávajícího protokolu RADIUS a posléze navrhnout možná řešení rozšíření tohoto protokolu. Následujícím krokem byla simulace vylepšeného RADIUS protokolu a vyhodnocení jeho činnosti.

Pro ověření činnosti protokolu RADIUS byla vytvořena jednoduchá experimentální počítačová síť, jejíž logické schéma je uvedeno na obr. 11.



obr. 11 Logické schéma experimentální sítě

Součástí této sítě je Supplicant, který vysílá požadavky na autentizaci směřované autentizačnímu serveru RADIUS2, přičemž tento požadavek je nejprve doručen serveru RADIUS1. To proběhne pomocí protokolu EAP [8], který je mezi Supplicantem a serverem RADIUS1 zaveden. Server RADIUS1 tento požadavek nejprve vyhodnotí z hlediska toho, zda je mu určen. To rozhodne podle textového řetězce *realm*, který je součástí přihlašovacího jména uživatele. Pokud mu rozhodnutí konkrétního požadavku na autentizaci nepřísluší, systém byl nakonfigurován tak, aby požadavek předal prostřednictvím protokolu RADIUS autentizačnímu serveru RADIUS2. Ten rozhodne, zda je nebo není požadavek konkrétního klienta oprávněný, a podle toho vydá autentizační stanovisko, které je stejnou cestou doručeno zpět Supplicantovi.

Na jednotlivých serverech byl použit jako RADIUS server software FreeRADIUS [30] a pro zachytávání přenášených paketů byl použit software WireShark [31]. Veškerý použitý software je volně šiřitelný.

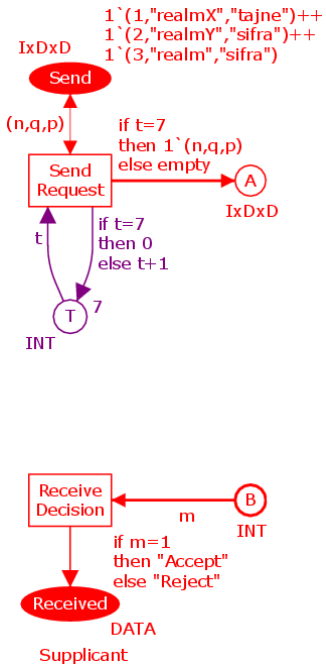
5.2 Simulace stávajícího protokolu

Pro ověřování základního chování RADIUS protokolu pomocí Petriho sítí byl na základě uvedené experimentální sítě vytvořen v aplikaci CPN Tools model s jedním suplikantem a dvěma RADIUS servery (viz obr. 13). Model slouží také jako východisko pro tvorbu simulace vylepšeného protokolu RADIUS. Tento model, přiložený jako softwarová příloha A.1, byl publikován v [E].

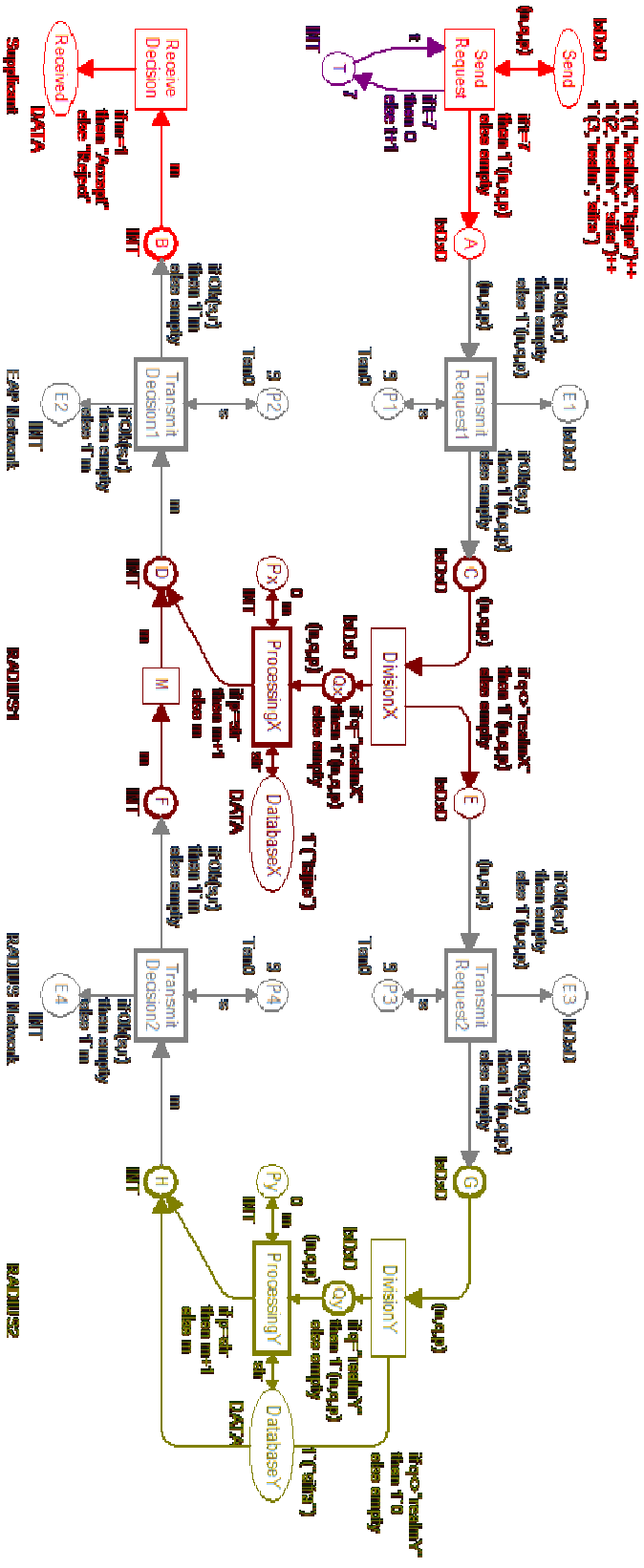
Jedná se o zjednodušený model činnosti reálně fungující sítě, který předpokládá, že klient zadal jeden požadavek na autentizaci do Supplicanta a tento požadavek je posléze modelem vyhodnocen. Pro zjednodušení a přehlednost nepracuje model se zpožděním jednotlivých prvků sítě, v přenášeném paketu bylo vynecháno uživatelské jméno (přenáší se pouze heslo a název *realm*u). Systém pro opakování ztracených paketů je pouze nastíněn. Jak je patrné z obr. 13 (strana 36), celý model je rozdělen do pěti odlišitelných částí (sekcí), jejichž činnost bude rozebrána podrobněji.

5.2.1 Sekce Supplicant

Sekce *Supplicant* je ohraničena pomocnými místy *A* a *B* a slouží jednak k odesílání požadavků, jednak k přijímání rozhodnutí.



obr. 12 Sekce Supplicant



obr. 13 Model činnosti sítě

Odesílání požadavků je reprezentováno místem *Send*, které obsahuje tři tokeny s výchozí hodnotou, která je tvořena datovým typem *Product*. Výchozí hodnota obsahuje pořadové číslo paketu, které je v tomto modelu neměnné. Dále obsahuje pole *realm* a pole *heslo*, která obsahují např. hodnoty „realmX“ a „tajne“.

Datový typ místa *Send* je deklarován jako

```
colset IxDxD = product INT * DATA * DATA;
```

Hrany, které mají přenášet token s požadavkem na autentizaci, mají inskripci obsahující `(n,q,p)`, kde proměnná n je typu *Integer*, a proměnné q a p jsou typu *String*.

Přechod *Send Request* slouží k odeslání tokenu (vybere se náhodně jeden z uvedených) a jeho činnost je omezována jednoduchým systémem čekání, který zajišťuje opětovné odeslání tokenu v případě, když na předchozí autentizační požadavek vlivem simulované chyby nedorazí odpověď. Z toho důvodu je také token stále udržován v místě *Send*. Systém pro opakované odesílání požadavků je reprezentován místem *T* a dvěma hranami. V místě *T* se inkrementuje do sedmi, což je předpokládaný počet kroků, během kterého by měla dorazit reakce. Po napočítání hodnoty sedm se požadavek znovu odešle a počítadlo se vynuluje.

Místo *Received* a přechod *Receive Decision* slouží pouze k formálnímu zobrazení přijatého rozhodnutí. To je reprezentováno hodnotou *m*, která může nabývat pouze hodnot 0 nebo 1, což se zobrazuje po přijetí jako text „Reject“ nebo „Accept“.

5.2.2 Sekce Network

Sekce *Network* jsou v modelu dvě a jsou odlišeny místy *A, B, C, D* a *E, F, G, H* a slouží pro přenos požadavku, resp. rozhodnutí. Jejich funkce by byla triviální nebýt toho, že veškeré přechody v těchto sekcích jsou vybaveny pravděpodobnostní funkcí a přenášený token je tak buď předán do dalšího modulu, nebo nenávratně skončí svou pouť na místě *E1 – E4*. Tím je simulována chyba v síti (ztráta RADIUS paketu nebo selhání serveru).

Rozhodnutí o další cestě tokenu je určováno pseudonáhodnou funkcí `Ok(s,r)`, která byla převzata z modelu *SimpleProtocol*, který je součástí aplikace CPN Tools [26]. Funkce je deklarována takto:

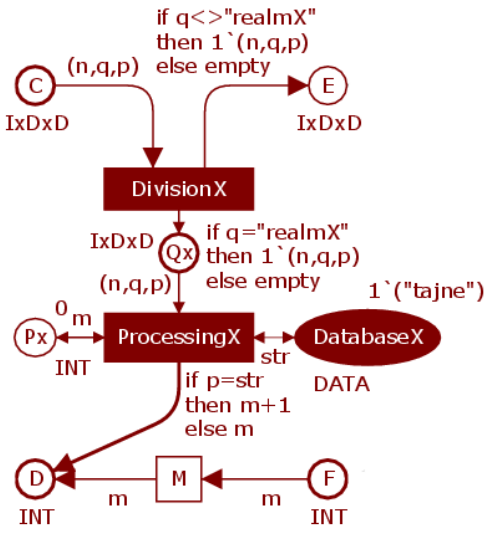
```
colset Ten0 = int with 0..10;
colset Ten1 = int with 1..10;
var s: Ten0;
var r: Ten1;
fun Ok(s:Ten0,r:Ten1) = (r<=s);
```

Význam této funkce je následující. Příkazy `colset` jsou definovány množiny `Ten0` a `Ten1`, které mohou obsahovat celá čísla v rozsahu 0-10, resp. 1-10. Příkazy `var` definujeme proměnnou `s`, resp. `r`, jenž patří do množiny `Ten0`, resp. `Ten1`. V posledním řádku deklarace je pak definována samotná funkce `Ok(s,r)`, která při provedení v modelu porovnává hodnoty proměnných `s` a `r`, když hodnota `s` je v modelu nastavena v místech *P1 – P4* na hodnotu 9. Hodnota proměnné `r` se při provedení funkce `Ok(s,r)` vygeneruje náhodně (to je dáno použitím deklarace `colset`) a porovná s hodnotou v proměnné `s`. Pak je vyhodnocena podmínka `(r<=s)` a podle jejího výsledku jsou provedeny inskripce na hranách sousedících s přechody *Transmit Request*, resp. *Transmit Decision*. Inskripce jsou vzájemně opačné, takže token pokračuje buď dále, nebo skončí na místě *E1 – E4* podle výsledku podmínky `(r<=s)`.

5.2.3 Sekce RADIUS1

Tato sekce provádí zpracování požadavků, které mu přísluší podle obsahu pole *realm*. Ostatní požadavky předává dále. Posouzení každého došlého požadavku provádí přechod *DivisionX* s pomocným místem *Qx*. Přijaté požadavky pak posuzuje přechod *ProcessingX*, který má k dispozici databázi klientů představovanou místem *DatabaseX*. Přechod provede porovnání došlého požadavku podle pole *heslo* se záznamem v databázi. Pomocné místo *Px* obsahuje hodnotu *m=0*, která je v případě záporného výsledku zaslána jako reakce, resp. v případě kladného výsledku je odeslána hodnota *m+1*.

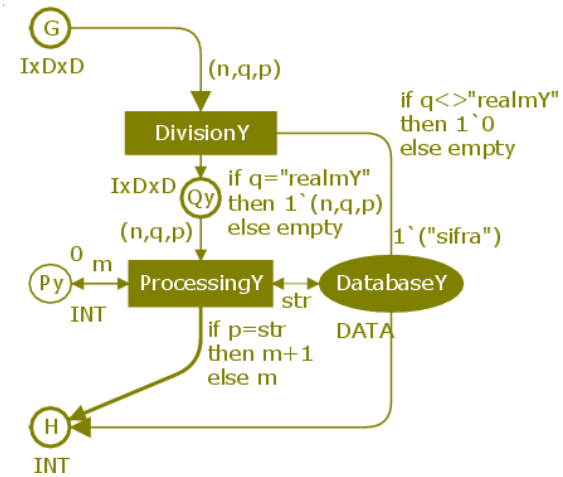
Pomocný přechod *M* slouží pouze k předávání rozhodnutí serveru *RADIUS2* do sekce *Supplicant*.



obr. 14 Sekce RADIUS1

5.2.4 Sekce RADIUS2

Sekce RADIUS2 pracuje velmi podobně jako sekce RADIUS1 s tím rozdílem, že přechod *DivisionY* rozhoduje, zda je došlý požadavek oprávněný k posouzení podle obsahu pole *realm* či nikoliv. Pokud je požadavek oprávněný, je předán k rozhodnutí přechodu *ProcessingY*, v opačném případě je zamítnut a zpět se odesílá hodnota *m=0*.



obr. 15 Sekce RADIUS2

Oblast přechodu *ProcessingY* pracuje obdobně jako v předchozím případě.

5.2.5 Závěr

Představený jednoduchý model dovede reagovat na náhodné požadavky o autentizaci podobně jako systémy vybavené stávajícím protokolem RADIUS. Pomocí tohoto modelu byly provedeny simulace základní činnosti a byla ověřena funkce protokolu RADIUS a schopnost CPN simulovat předložený problém (viz [E] a softwarová příloha A.1).

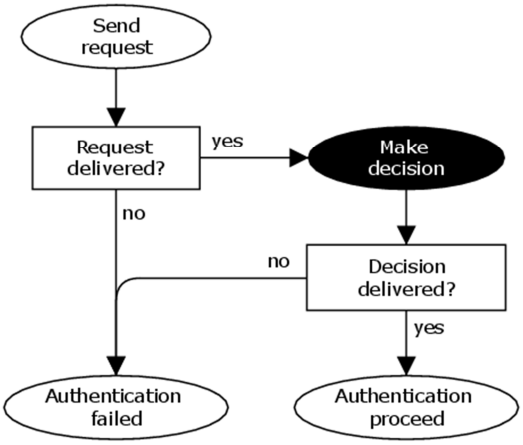
5.3 Návrh algoritmu řešení

5.3.1 Popis aktuálního algoritmu

Protokol RADIUS pracuje v distribuovaném prostředí tak, že autentizační požadavek uživatele předá k vyřešení domovskému autentizačnímu serveru tohoto uživatele prostřednictvím nespolehlivé počítačové sítě. Autentizační požadavek odešle server-odesílatel, ale domovský server uživatele jej nemusí vůbec obdržet v případě, že přenosová trasa je dočasně nedostupná nebo je domovský autentizační server mimo provoz (tyto případy označme souhrnně pojmem „**chybové stavy komunikace**“). Autentizační protokol RADIUS je součástí aplikační vrstvy podle modelu ISO/OSI a v jeho konstrukci obvykle nejsou nástroje, které by tyto chybové stavy komunikace ošetřovaly. V tomto směru se RADIUS spoléhá na protokoly nižší vrstvy. RADIUS byl primárně navržen pro práci nad transportním protokolem UDP [16], který rovněž chybové stavy komunikace ignoruje a v nižších vrstvách už nástroje na ošetřování nedoručených datagramů hledat nelze. RADIUS ve spolupráci s UDP tak nemá možnost chybové stavy komunikace řešit.

Poněkud jiná situace nastává pro implementace protokolu RADIUS pracující nad protokolem TCP, který má dostatečné nástroje na ošetřování chybových stavů komunikace (viz [32]) a který může ztracený (respektive nepotvrzený) segment dat odeslat znovu. Nicméně i protokol TCP po určité době trvání chybového stavu selže. Doba, po kterou musí být přenosová cesta nebo domovský server mimo provoz, není obecně stanovená a je závislá na aktuálním nastavení protokolu TCP na daném koncovém zařízení a na

průměrném zpoždění přenosové cesty[vi]. Z hlediska autentizace uživatele protokolem RADIUS může selhání autentizace obecně běžně nastávat.




```
DO send request
IF request delivered THEN
       DO make decision
       IF decision delivered THEN
        RETURN authentication proceed
       ELSE RETURN authentication failed
ELSE RETURN authentication failed
```

obr. 16 Diagram a pseudokód stávajícího algoritmu

Algoritmus chování protokolu RADIUS je názorně zobrazen na obr. 16. V případě doručení požadavku na autentizaci domovský autentizační RADIUS server požadavek porovná se svou databází uživatelů. Verifikuje uživatelské jméno a heslo a podle výsledku buď požadavek akceptuje, nebo jej zamítne. Své rozhodnutí poté obdobným způsobem, který byl popsán výše, odešle zpět serveru-odesílateli. Pokud rozhodnutí serveru-odesílateli dorazí, považujeme proces autentizace za úspěšný (bez ohledu na to, zda uživatel byl autentizován nebo byl jeho pokus zamítnut). I v případě doručování rozhodnutí však může dojít k selhání přenosové cesty.

Jednou z nevýhod stávajícího řešení je selhání autentizace, pokud dojde k chybovému stavu komunikace. Systém je v současnosti nastaven tak, že pokud je domovský autentizační server nedostupný nebo nereaguje v časovém limitu, server-odesílatel požadavek na autentizaci klienta zamítne, protože autentizace na vzdáleném serveru selže. Toto pravidlo je definováno na straně 4, bod 4.3.1 platné Roamingové politiky (viz odstavec 2.4).

[vi] Průměrné zpoždění přenosové cesty je však nutné určit. To probíhá tak, že doba cesty TCP segmentu mezi odesílatelem a příjemcem a potvrzení o doručení tohoto segmentu je odesílatelem měřena (RTT – Round Trip Time, [40]) a na jejím základě se vypočítávají pomocí Karnova, příp. Jacobsonova [41]) algoritmu doby pro opakované odeslání (RTO, retransmission timeout) příslušného segmentu, resp. příslušných nepotvrzených bytů.

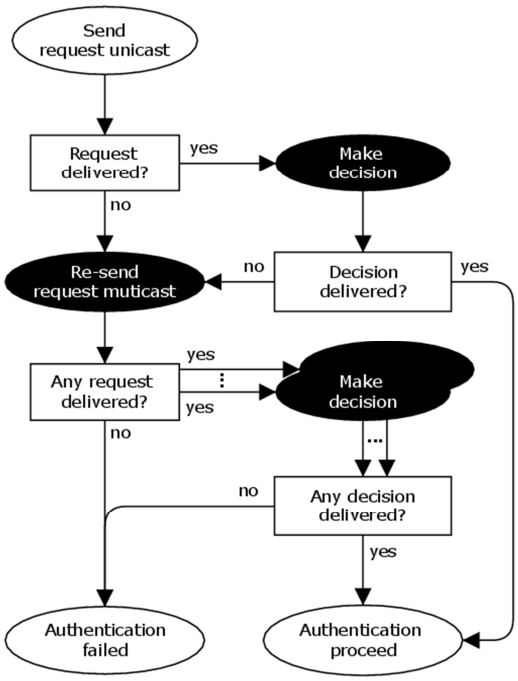
Dále bude popsán námi navržený algoritmus, který tento problém minimalizuje a zvyšuje spolehlivost systému, jak bude ukázáno v následujícím textu.

5.3.2 Popis vylepšeného algoritmu

Princip řešení je založen na faktu, že distribuovaná síť s federativní autentizací může být tvořena mnoha servery. Autentizační informace o uživateli tak může být uložena nejen v jeho domovském serveru (v serveru poskytovatele identity), ale i na jiných serverech (tedy – z hlediska daného uživatele – serverech poskytovatelů služeb). K uložení autentizační informace na jiný než domovský server dojde tehdy, když uživatel využije tento autentizační server poskytovatele zdrojů k přihlášení do sítě.

Tento server se obrátí na domovský server poskytovatele identity uživatele a po případné úspěšné autentizaci si informaci uloží do zvláštní databáze (viz dále), jak bylo popsáno v předchozím odstavci. Hlavní myšlenka vylepšení protokolu RADIUS uvažuje s možností využít autentizační informace uživatele uložené na všech dostupných serverech, kde by se mohly nacházet, nikoliv pouze na serveru poskytovatele identity. Vhodnou aplikací tohoto záměru do praxe může dojít ke snížení autentizačních selhání zejména u uživatelů, kteří využívají velký počet poskytovatelů zdrojů (tedy relativně hodně „cestují“).

Námi navržený algoritmus při výskytu chybového stavu komunikace kontaktuje nějakou formou (viz odstavec 5.3.4) určené servery v dané doméně (nejlépe všechny) a pokusí se provést autentizaci uživatele pomocí jejich informací. Výběr oslovených serverů bude závislý na dohodě mezi jednotlivými poskytovateli služeb. Algoritmus popsaného chování je zobrazen na obr. 17. Oslovené servery standardním způsobem na žádost reagují tak, jako by jim byla primárně určena. Nicméně, vzhledem k tomu, že nejsou správci realmu uživatele, o jehož autentizaci se žádá, nemohou použít databázi uživatelů, kterou vedou v roli poskytovatelé identity. Pro rozhodnutí musí použít databázi uživatelů, kteří jejich služby využili jako poskytovatele zdrojů. Tuto databázi označme „**cache databáze**“, protože záznamy v ní by měly sloužit pro opakované žádosti o autentizaci a měly by mít časově omezenou platnost (viz odstavec 5.3.3).




```
DO send request unicast
IF request delivered THEN
       DO make decision
       IF decision delivered THEN
        RETURN authentication proceed
       ELSE resend request multicast
ELSE resend request multicast
       IF any request delivered THEN
        DO make decision
       ELSE RETURN authentication failed
       IF decision delivered THEN
        RETURN authentication proceed
       ELSE RETURN authentication failed
```

obr. 17 Diagram a pseudokód vylepšeného algoritmu

Po provedení rozhodnutí oslovené servery odešlou serveru-odesílateli své stanovisko. Server-odesílatel nějakým způsobem doručené odpovědi vyhodnotí a odešle Supplicantovi konečnou odpověď. Způsob vyhodnocení může být různý – od „hlasování“ všech zúčastněných až po direktivní rozhodnutí serveru-odesílatele. Příslušnou instanci nového algoritmu zodpovědnou za výše uvedené rozhodování označme pojmem **adjudikátor**. Některé možnosti realizace adjudikátorů budou navrženy a experimentálně ověřeny v odstavci 5.4.6.

Navržený způsob řešení problému nebude vyžadovat změnu protokolu RADIUS jako takového, resp. nebude vyžadovat změnu jiných protokolů. Cíle lze dosáhnout výše popsanou změnou algoritmu chování koncových prvků, tedy úpravou softwarové implementace protokolu RADIUS a doplněním nového prvku – cache databáze (viz odstavec 5.3.3).

Předem lze dovodit, že tento způsob fungování povede k většímu zatížení sítě a zpoždění reakce na požadavek o autentizaci, ale nejspíše významně sníží možnost celkového selhání autentizace. Pro ověření těchto předpokladů a jejich

přibližnou kvantifikaci byl vytvořen model vylepšeného protokolu a na tomto modelu byly provedeny experimenty.

5.3.3 Cache databáze

Cache databáze je určena pro uchovávání autentizačních údajů uživatelů, kteří využijí autentizační server RADIUS jako poskytovatele služeb. Při dočasné nedostupnosti poskytovatele identity příslušného uživatele, pak mohou být záznamy cache databáze použity k autentizaci uživatele. Cache databáze by měla uchovávat jen ty autentizační záznamy, které prošly úspěšně autentizací u poskytovatele identity a jsou tedy „ověřené“. Strukturálně by se cache databáze neměla významně lišit od stávající databáze poskytovatele identity. Její záznamy by však měly mít časově omezenou platnost. Je na dohodě v rámci federace či na konkrétní implementaci protokolu, jak dlouhou „trvanlivost“ by měl záznam mít. Aby měla navržená inovace protokolu smysl, mělo by se jednat nejméně o měsíce (např. 1 – 3 měsíce), přičemž příliš krátká doba trvanlivosti záznamu může snížit využitelnost inovovaného protokolu. Naopak důvodem pro dočasnou platnost záznamů jsou případné změny v uživatelských účtech na straně poskytovatele identity (zrušení účtu, změna hesla), které se nemusí projevit v cache databázi poskytovatele služeb. Pro minimalizaci nežádoucích projevů (viz subkapitola 5.6), které tyto souvislosti mohou mít, je v odstavci 6.5 navrženo řešení formou poskytování informací o změnách v účtech mezi členy federace. Nicméně ne všichni členové musí nutně navržené řešení využívat, takže časové omezení záznamu má své opodstatnění.

Na tomto místě je vhodné zmínit, že ukládání záznamů v cache databázi má také bezpečnostní rovinu. Nicméně předpokládáme, že stávající schopnost autentizačních serverů poskytovatelů služeb zajistit bezpečné uložení a nakládání s daty v primární databázi uživatelů bude možné přenést i na záznamy v cache databázi.

5.3.4 Způsob komunikace mezi autentizačními servery

Jednotlivé autentizační servery sdružené ve federativní síti vzájemně komunikují přímo (z hlediska IP vrstvy), tedy bez použití centrálního uzlu, který by v zásadě popíral smysl distribuované sítě. Nicméně nadřazený RADIUS server a tedy nástin hierarchie zde existuje (viz obr. 4), ale tento server má za cíl především shromažďovat adresy jednotlivých členských serverů federace a poskytovat je těm serverům, kteří spolu aktuálně potřebují komunikovat.

Jak bylo popsáno v odstavci 5.3.2, při opakovaném pokusu o autentizaci potřebuje autentizační server oslovit okolní autentizační servery. Slovo „okolní“ v předchozí větě lze chápat ve smyslu serverů dané federace v národní doméně. Zajistit přesah hromadných autentizačních požadavků do jiné než národní domény je ve spolupráci nadřazených autentizačních serverů jednotlivých domén zřejmě možné, nicméně bylo by to z různých hledisek náročné (formální/legislativní, procesní, technické…). Do budoucna by to ovšem bylo z hlediska fungování systému jako celku potřebné.

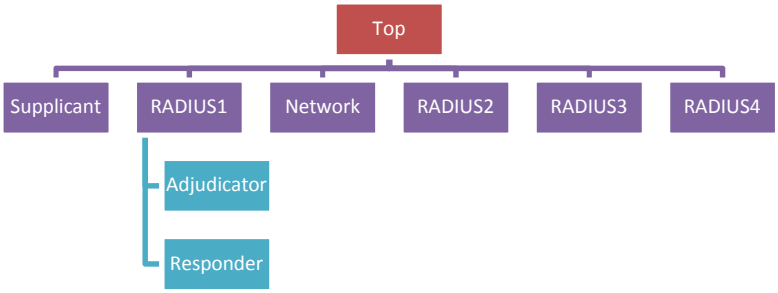
Oslovení zmíněných okolních serverů provede autentizační server buď souborem zpráv jednotlivým serverům (**unicast**) nebo skupinovou zprávou (**multicast**). Varianta s využitím multicastu je nepochybně atraktivní, neboť znamená mnohem méně paketů v síti. V této práci nicméně ověříme komunikaci s využitím unicastu, čímž získáme představu o nejvyšším množství paktů, kterým by mohl vylepšený protokol zatížit síť.

5.4 Model vylepšeného protokolu

5.4.1 Základní popis

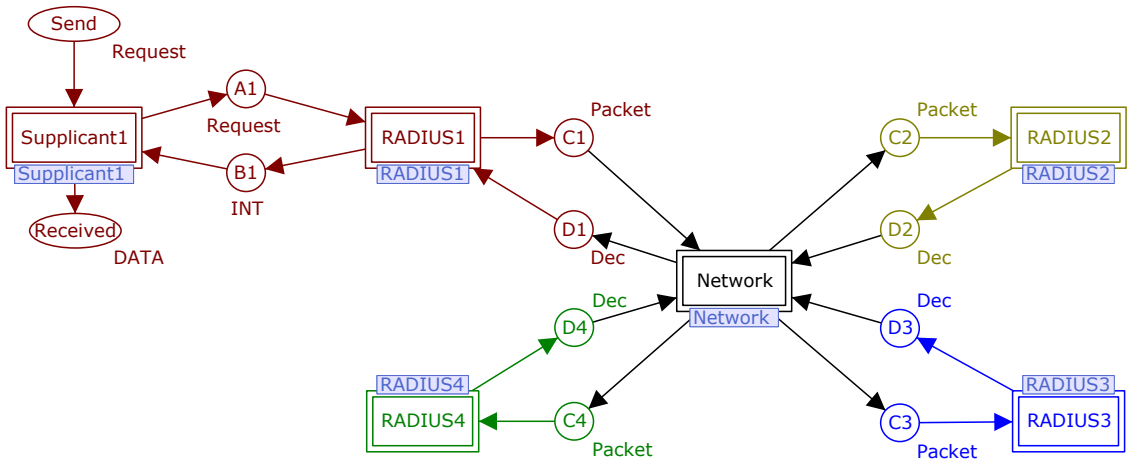
Model protokolu s výše navrženým algoritmem byl vytvořen v barevných Petriho sítích s pomocí aplikace CPN Tools a byl publikován viz [C, D]. Model je přiložen k této práci jako softwarové přílohy B.1 – B.3. Jedná se o hierarchický model, který má tři úrovně. Schéma s barevným zohledněním hierarchie sekcí je na obr. 18.

Zde je vidět šest základních sekcí modelu – *Supplicant1*, *RADIUS1* - *4* a *Network*. Sekce *RADIUS1* se dělí na dvě subsekce *Adjudicator*, která slouží pro rozhodování, a *Responder*, jenž generuje automatické záporné odpovědi.



obr. 18 Schéma hierarchického modelu

Jednotlivé sekce a subsekce budou popsány níže. Schéma modelu na úrovni *Top* je uvedeno na obr. 19. Patrné jsou základní sekce a jejich vzájemné propojení pomocí míst *A1* – *D4*. Místa *Send* a *Received* jsou součástí sekce *Supplicant1* a na úrovni *Top* jsou umístěna pro přehlednou kontrolu činnosti modelu.



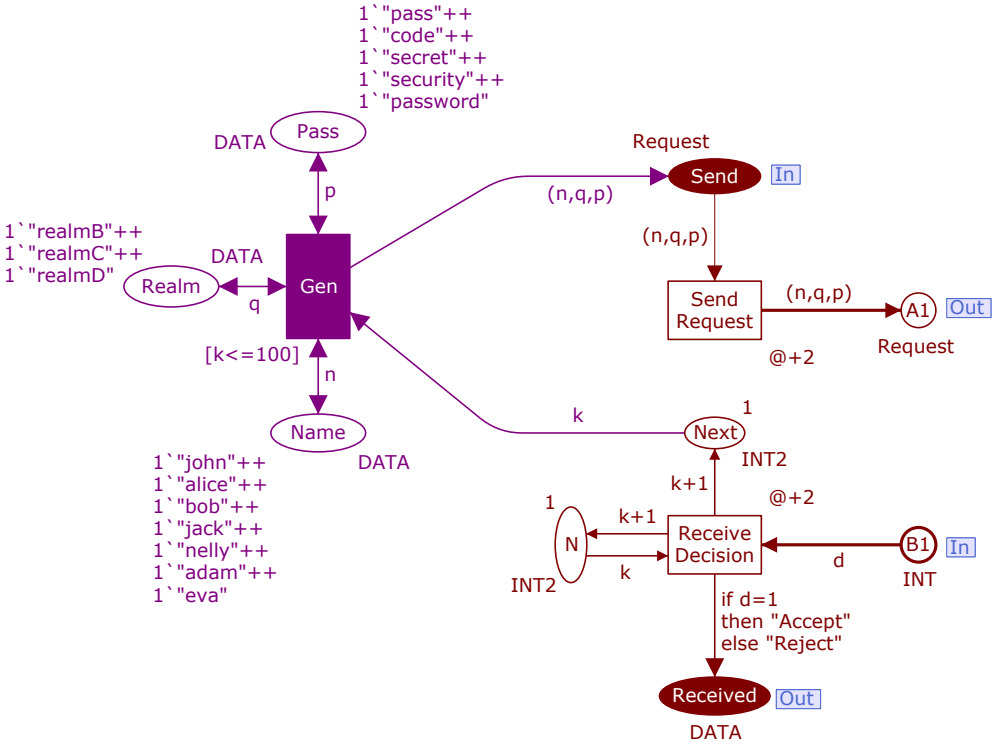
obr. 19 Schéma modelu (úroveň Top)

5.4.2 Sekce Supplicant1

Tato sekce se skládá ze tří částí. Jedna část zodpovídá za odesílání požadavků, druhá za jejich přijímání. Součástí druhé části je místo *N*, které si pamatuje počet přijatých paketů. To lze využít pro omezení celkového počtu paketů a tak simulaci ukončit. Poslední částí je oblast kolem přechodu *Gen*. Tato část je zodpovědná za generování náhodných požadavků. Požadavky se skládají ze tří částí. Jsou to *Name*, *Realm* a *Pass*. Požadavek je tedy tvořen

jménem uživatele, názvem jeho domovského realmu (například tul.cz) a heslem. Z obr. 20 je patrné, jaké náhodné kombinace může generátor sestavovat. Generátor vygeneruje další požadavek vždy až po obdržení odpovědi na předchozí odeslaný požadavek. O to se stará hrana s označením k. Její hodnotu sleduje strážník (**Guard**) uvedený v hranatých závorkách u přechodu *Gen*. Tento strážník při nesplnění podmínky `k<=100` deaktivuje trvale činnost přechodu *Gen*. Tím se zastaví generování požadavků a tím i činnost celého modelu. Takto se nastavuje počet paketů, které má model vyhodnotit.

Některé přechody či hrany v modelu obsahují formulaci `@+X`, kde `X` je časová hodnota. Tato hodnota se přičítá při průchodu přechodem nebo hranou k časové hodnotě tokenu a zpožďuje jej. Díky tomu může model pracovat s časem a zpožděním. Funkce *DEL1()*, resp. *DEL2()* přidává tokenu náhodné zpoždění v rozsahu 1-5, resp. 5-10 časových jednotek.



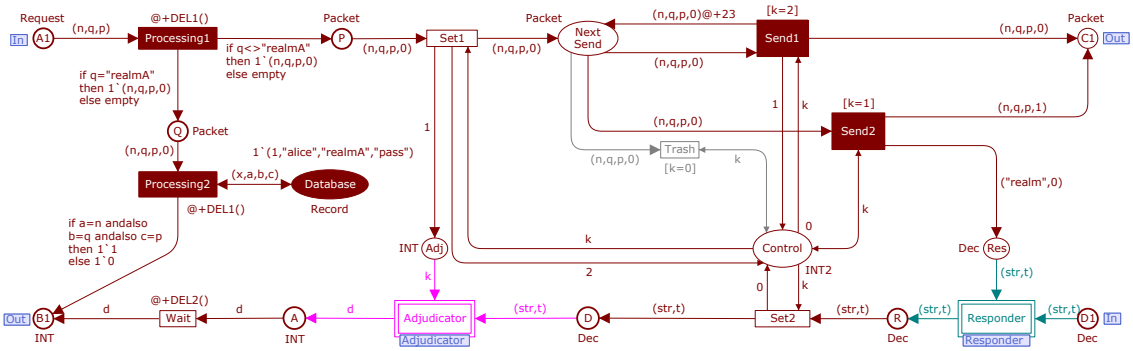
obr. 20 Sekce Supplicant

5.4.3 Sekce RADIUS1

Toto je nejsložitější sekce modelu. Oblast kolem přechodu *Processing2* je zodpovědná za řešení požadavků, které serveru *RADIUS1* přímo náležejí. Jsou

to požadavky, které obsahují řetězec „realmA“. V této konfiguraci modelu však takové požadavky nepřicházejí, protože je generátor požadavků v sekci *Supplicant* negeneruje. Jedná se o záměr, který na výsledky simulace nemá vliv, protože cílem simulace je sledování chování modelu v distribuovaném a nikoliv lokálním prostředí. Všechny přijaté požadavky na autentizaci tedy pokračují směrem k pomocnému místu *P* a dále přes přechod *Set*, který slouží k nastavení systému do počátečního stavu.

Tato sekce modelu byla vylepšena tak, aby zlepšovala spolehlivost systému jako celku. Požadavek, který domovský server nezpracoval v důsledku chyby, odešle server *RADIUS1* ještě jednou všem serverům v dané skupině (doméně)[vii]. Ostatní servery v doméně využijí svou databázi dříve uložených autentizačních informací (cache databáze, viz 5.4.4). Mezi těmito informacemi může být i informace o právě autentizovaném uživateli. Díky tomu pak autentizace uživatele může proběhnout úspěšně.



obr. 21 Sekce RADIUS1 (vylepšený protokol)

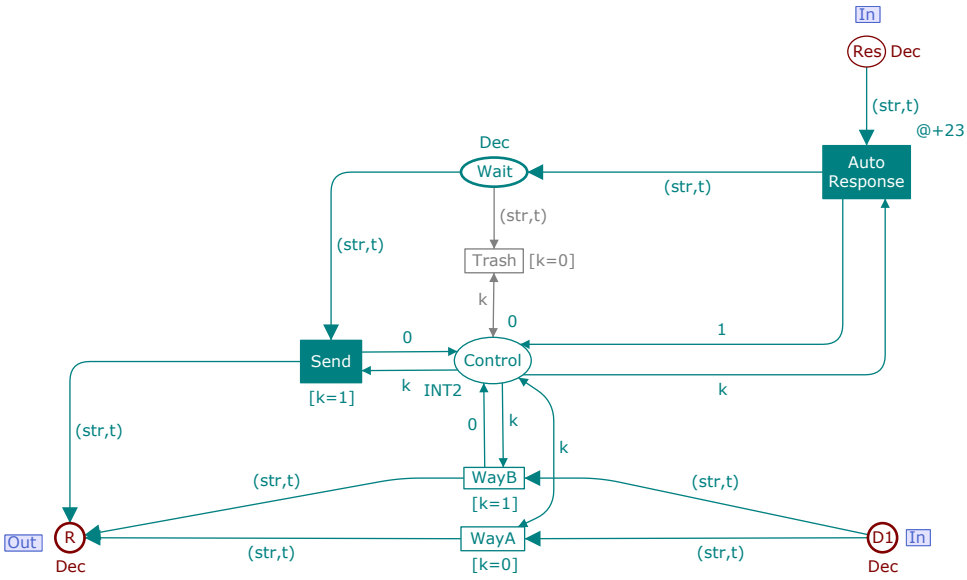
Přechod *Send1* odesílá pakety, které jsou určeny pouze přímému adresátovi – domovskému serveru autentizovaného uživatele. Zatímco požadavky byly tvořeny trojicí hodnot `(n,q,p)`, pakety jsou tvořeny čtveřicí hodnot `(n,p,q,t)`.

Odesílaný paket je složen ze čtveřice hodnot `(n,q,p,t)`, kde `t=0`. Hodnota `t=0` určuje, že paket bude doručen pouze domovskému serveru uživatele. Tento paket se zároveň uloží do místa *Next Send* pro případné pozdější opakované odeslání. Časová hodnota uloženého paketu je zvýšena o 23

[vii] Stávající protokol RADIUS odesílá požadavek na autentizaci jen jednou.

časových jednotek. Za tuto dobu dojde k odpovědi na odeslaný paket nebo k selhání (odpověď nepřijde). Místo *Control* se zároveň nastaví tak, že případné opakované odeslání paketu neproběhne již přes přechod *Send1*, ale přes přechod *Send2*. V případě, že autentizace proběhne a odpověď je doručena, přijde odpověď do místa *D1* a dále projde přes subsekci *Responder* (popis níže) do místa *R*. Průchodem odpovědi přes přechod *Set2* je místo *Control* nastaveno tak, že paket čekající v místě *Next Send* je zrušen (přijde do přechodu *Trash*, odkud již není návratu), protože nebude nutné jej už použít.

Pokud ale odpověď domovského serveru na odeslaný paket v časovém limitu nepřijde, bude odeslán totožný paket s nastaveným `t=1`. To povede k odeslání tohoto paketu všem serverům dané skupiny. *RADIUS1* bude čekat na jejich odpovědi. I tak však může nastat situace, že v časovém limitu nepřijde žádná odpověď. Na to musí *RADIUS1* reagovat tím, že sám vygeneruje zápornou odpověď.

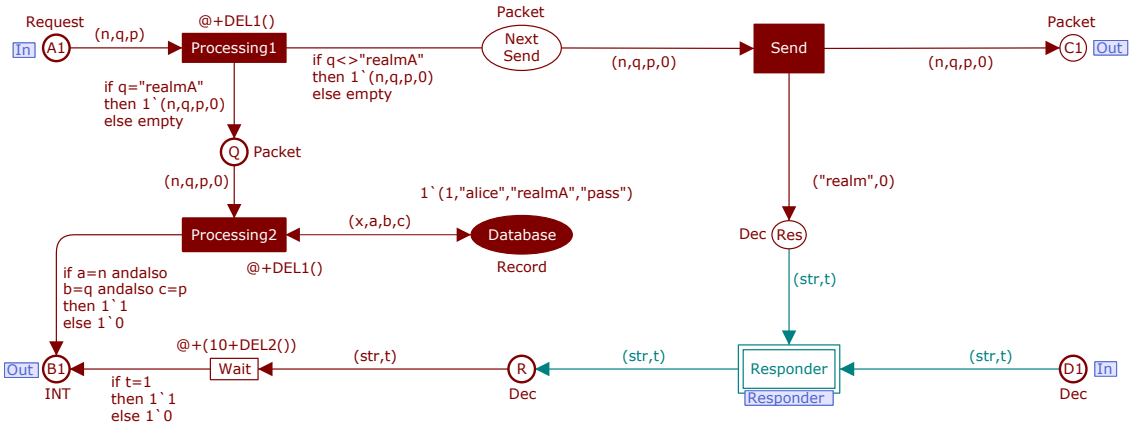


obr. 22 Subsekce Responder

O to se stará subsekce *Responder*, viz obr. 22. Průchodem paketu přechodem *Send2* v nadřazené sekci dojde k vygenerování záporné odpovědi přechodem *Auto Response* do místa *Wait*. Tato odpověď pak čeká 23 časových jednotek a podle stavu místa *Control* je pak buď odeslána jako odpověď (projde místem *Send*) nebo je zrušena (projde místem *Trash*). V případě, že odpověď přijde, je

tato odpověď předána do nadřazené sekce *RADIUS1*. První odpověď vždy prochází přechodem *WayB*, čímž nastaví místo *Control* tak, že automatická odpověď čekající v místě *Wait* bude zrušena a případné další odpovědi již prochází přes přechod *WayA*. Pokud žádná odpověď nepřijde v časovém intervalu, z místa *Wait* projde automatická záporná odpověď přechodem *Send* do nadřazené sekce a odtud dále až do sekce Supplicant.

Pro srovnání je na obr. 23 zobrazena sekce *RADIUS1* nevylepšeného protokolu. Tato sekce především zasílá požadavek na autentizaci pouze jednou (proto zde chybí oblast kolem přechodu *Send2*) a z toho důvodu také neobsahuje rozhodovací subsekci *Adjudicator*, která bude popsána dále v subkapitole 5.4.6.



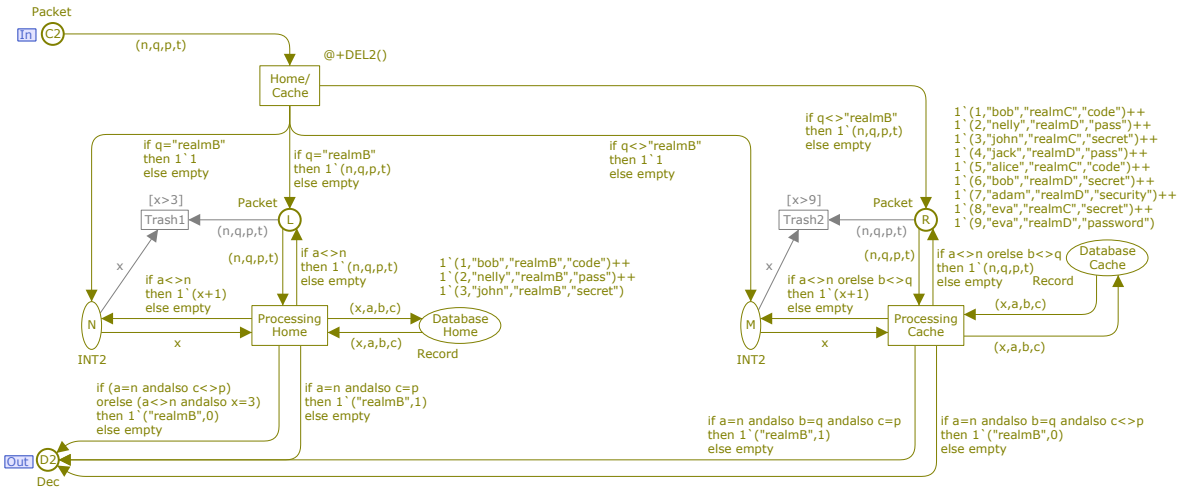
obr. 23 Sekce RADIUS1 (aktuální protokol)

5.4.4 Sekce RADIUS2

Tato sekce modelu reaguje na předané pakety a dělá rozhodnutí. Model obsahuje tři téměř totožné sekce *RADIUS2* až *RADIUS4*. Sekce se liší jen v tom, jaké záznamy jsou uloženy v jednotlivých databázích. Sekce *RADIUS3* a *RADIUS4* nebudou proto popsány.

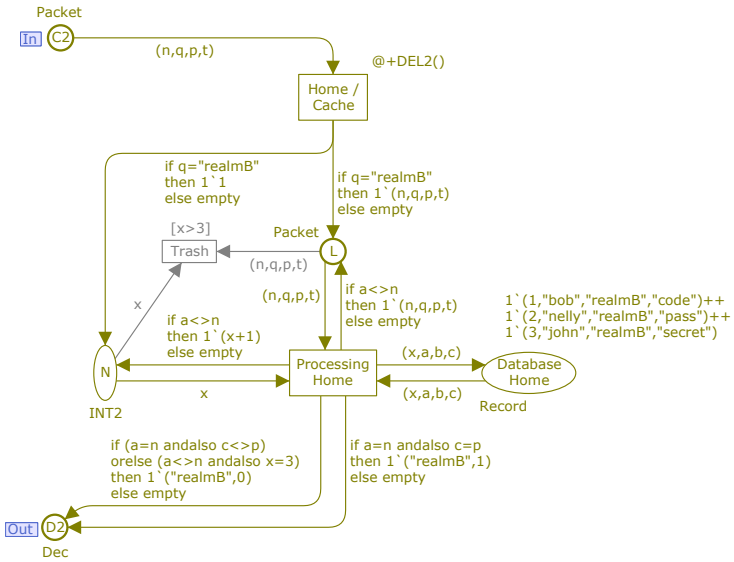
Aktuální RADIUS server porovnává informace z došlého paketu se svojí databází. Toto v modelu provádí oblast kolem přechodu *Processing Home* a místa *Database Home*. Pomocná místa zajišťují zapamatování došlých informací (místo *L*) a jejich porovnávání s příslušným záznamem v databázi. Každý záznam (čtveřice `x,a,b,c`) je označen hodnotou `x`. V místě *N* je uložena hodnota označující řádek, který má být z databáze vyzvednut. Vlastní

porovnání a rozhodnutí zajišťují hrany vedoucí z přechodu *Processing Home* do místa *D2*.



obr. 24 Sekce RADIUS2 (vylepšený protokol)

Vylepšený RADIUS server má ještě oblast kolem přechodu *Processing Cache* a *Database Cache*. Jedná se o cache databázi (viz 5.3.3), kde jsou uloženy záznamy o uživatelích, kteří se přes server RADIUS2 v minulosti úspěšně autentizovali svému domovskému RADIUS serveru. *Databáze Cache* obsahuje záznamy patřící do jiných realmů než je realm serveru RADIUS2 (tedy *RealmB*).



obr. 25 Sekce RADIUS2 (aktuální protokol)

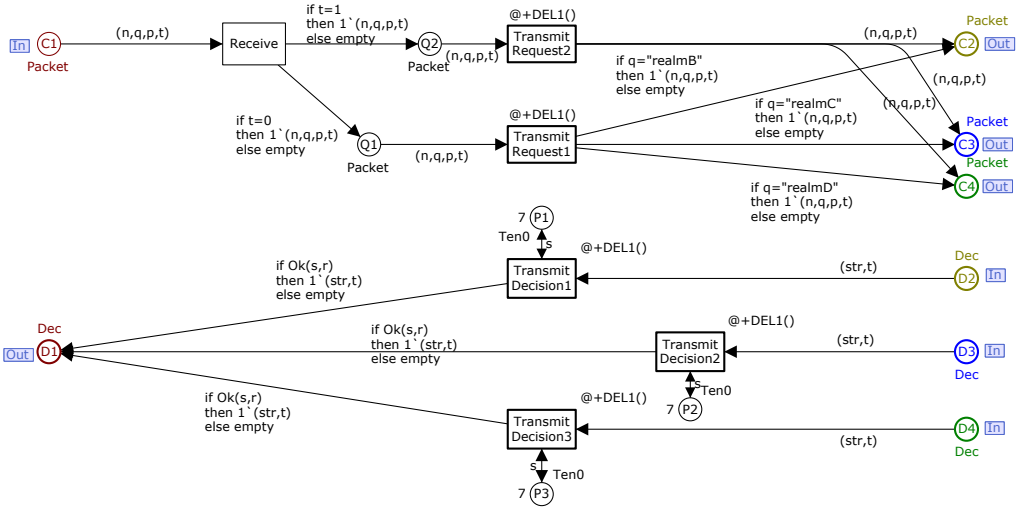
Činnost této oblasti je velmi podobná oblasti *Home*. Rozdíl je v tom, že tato oblast reaguje pouze na pakety, které byly odeslány opakovaně. Oblast

generuje kladné odpovědi autentizace, pokud přijatý požadavek odpovídá záznamu v databázi[viii], a záporné odpovědi, pokud odpovídá jméno uživatele (včetně realmu) a nesouhlasí heslo. Pokud databáze uživatele „nezná“, negeneruje žádnou odezvu, tím se snižuje počet paketů nutných k odeslání.

Na obr. 25 je pro srovnání zobrazena sekce RADIUS2 v modelu, který představuje aktuální verzi protokolu RADIUS. Část kolem přechodu *Processing Home* je totožná s vylepšenou verzí, druhá část není pochopitelně vůbec obsažena.

5.4.5 Sekce Network

Tato sekce slouží pro odesílání paketů od serveru *RADIUS1* k serverům *RADIUS2* až *RADIUS4* a pro odesílání rozhodnutí z těchto serverů do serveru *RADIUS1*. Oblast mezi místem *C1* a místy *C2*, *C3* a *C4* odesílá pakety. Pokud je v odesílaném paketu nastavena hodnota `t=0`, bude aktivní přechod *Transmit Request1*, který předá paket pouze tomu RADIUS serveru, kterému je určen podle hodnoty proměnné q (tedy podle realmu).



obr. 26 Sekce Network

V případě, že paket má nastavenu hodnotu `t=1`, předá přechod *Transmit Request2* tento paket všem serverům. V opačném směru pracují přechody *Transmit Decision1* až *Transmit Decision3*. Tyto přechody a k nim navazující

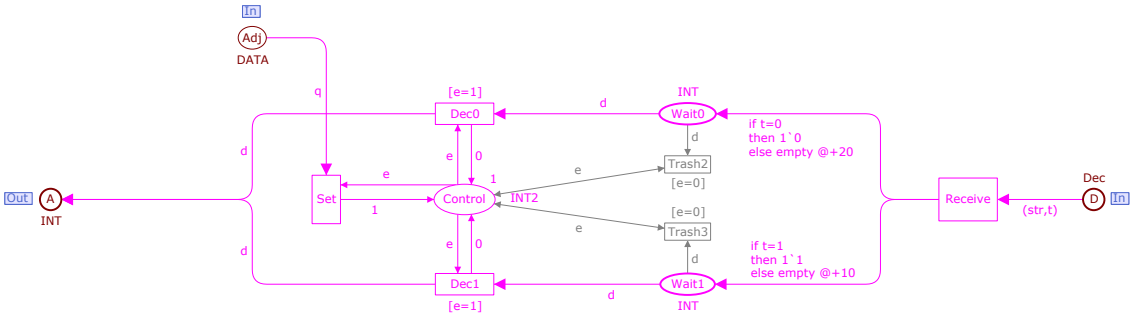
[viii] Databáze obsahuje náhodná data vložená staticky.

hrany jsou vybaveny pravděpodobnostní funkcí `0k(s,r)`. Tato funkce omezuje průchod paketu s určitou pravděpodobností. Pravděpodobnost úspěšného průchodu je dána hodnotou umístěnou v místech *P1* až *P3*. Na obr. 26 je nastavena náhodná hodnota 7, pravděpodobnost průchodu je tedy 70%. Tato pravděpodobnost zahrnuje všechny možné chyby a selhání serverů a sítě, které mohou vzniknout.

5.4.6 Subsekce Adjudicator

Tato subsekce je součástí pouze vylepšeného serveru RADIUS1. V případě, že systém obdrží více odpovědí, musí je nějakým způsobem zpracovat a rozhodnout. Za to zodpovídá rozhodovací subsekce označená jako **adjudikátor**. Adjudikátor představuje parametrizovatelnou složku algoritmu, kterou lze přizpůsobit požadavkům konkrétní situace. Ne všude musí být adjudikátor stejný, naopak umožňuje vyjádřit lokální politiku konkrétního poskytovatele služeb.

V této práci byly vytvořeny a ověřeny tři modelové případy adjudikátorů. Označovány budou jako **simple**, **base** a **vote**. V této subkapitole budou jednotlivé adjudikátory popsány z hlediska návrhu a činnosti. Adjudikátor má klíčový vliv na chování modelu jako celku, takže výsledky experimentů s jednotlivými adjudikátory budou popsány v subkapitole 5.5.



obr. 27 Simple adjudikátor

5.4.6.1 Simple adjudikátor

Tento adjudikátor obsahuje dvě cesty, obr. 27. Jednu pro kladné a druhou záporné odpovědi. Adjudikátor lze nakonfigurovat různě. V našem případě je pomocí časových konstant model nastaven tak, že záporné odpovědi čekají

v místě *Wait0*, zda nepřijde alespoň jedna odpověď kladná. Ta projde bez většího zpoždění přes přechod *Dec1*, kde nastaví místo *Control* tak, že záporné odpovědi budou zahozeny. Systém tedy preferuje kladnou autentizaci. Vstupní místo *Adj* slouží pouze pro nastavení adjudikátoru do výchozího stavu.

Oblast lze ale přepracovat i jinak, například lze preferovat zápornou autentizaci. Může se také stát, že jednou z příchozích odpovědí bude odpověď domovského serveru. Tato odpověď má však nyní stejnou „váhu“ jako odpovědi ostatních serverů.

Z uvedeného faktu vzniká **bezpečnostní riziko autentizace neaktuálního účtu**. To znamená, že v určitých situacích může systém povolit autentizaci uživatele, kterou by jeho domovský server (pokud by byl funkční) nepovolil. Může se to stát zejména v případě, že daný uživatel si změní heslo, ale po nějakém (relativně krátkém) čase použije z nějakého důvodu pro autentizaci původní heslo (např. „chybou ve své hlavě“ nebo to za něho udělá útočník). Výpadek jeho domovského serveru způsobí, že systém se pokusí uživatele autentizovat proti cache databázím ostatních RADIUS serverů, které však mohou mít uložen záznam o uživateli ještě s původním heslem (tedy neaktuální účet). Toto heslo však může být kompromitované, a tak tato situace je potenciálním bezpečnostním rizikem[ix]. Je však otázkou, zda je toto riziko natolik značné (muselo by dojít k sérii náhodných jevů, aby došlo ke skutečnému ohrožení uživatelského účtu), než aby byl tento způsob činnosti adjudikátoru označen za nepoužitelný.

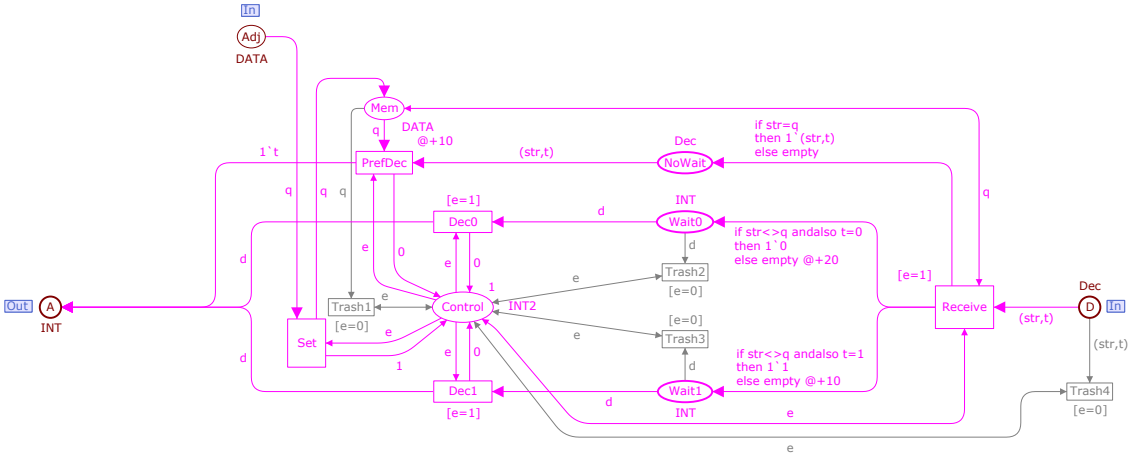
Závěrem je třeba upozornit na obecnou skutečnost, že nastavení doby platnosti cache záznamu ovlivňuje chování systému jako celku. Vhodná doba expirace cache záznamů by mohla být náplní dalšího rozvoje tohoto tématu a nových experimentů.

5.4.6.2 Base adjudikátor

Base adjudikátor se schová podobně jako adjudikátor Simple. Klíčovým rozdílem je však fakt, že primárně preferuje odpověď od domovského serveru,

[ix] Minimalizace tohoto rizika je navržena v subkapitole 6.5.

pokud taková přijde. V tom případě jsou ostatní odpovědi zahozeny. Odpovědi od domovského serveru jsou téměř bez zpoždění propuštěny k odchozímu místu *A*, zatímco ostatní odpovědi čekají, zda nějaká odpověď domovského serveru dorazí. Časovými konstantami na hranách směrem k místům *Wait0* a *Wait1* je dosaženo toho, že adjudikátor sekundárně preferuje kladné autentizace před zápornými. Abychom dosáhli toho, že adjudikátor bude preferovat odpovědi domovského serveru, je nutné, aby se do něho dostala informace, kterému domovskému serveru byl prvotní požadavek vlastně směřován. Toho se dosáhne tím, že z místa *Adj* přes přechod *Set* je do místa *Mem* tato informace předána ve formě řetězce s hodnotou „realm“.



obr. 28 Base adjudikátor

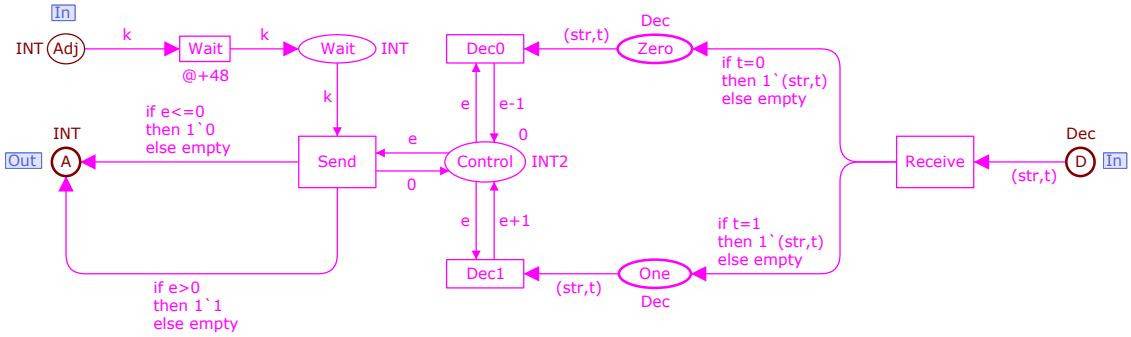
Této informace pak využívá přechod *Receive*, který vstupující pakety třídí dle toho, zda pocházejí od domovského serveru či nikoliv.

Takto navržený adjudikátor dává přednostní šanci domovskému serveru, pokud se po ztrátě paketu nebo dočasném výpadku znovu zapojí do rozhodovacího procesu, povolit nebo zamítnout autentizaci uživatele. V případě, že domovský server nereaguje ani na opakovanou výzvu, riziko autentizace uživatele zastaralým heslem, jak bylo popsáno v předchozím odstavci, přetrvává.

5.4.6.3 Vote adjudikátor

Ve třetím případě byl navržen „hlasovací“ adjudikátor. Tento se od předchozích liší tím, že sám generuje kladnou nebo zápornou autentizaci a mohl by tudíž i převzít funkci subsekce *Responder* (z důvodu unifikace sekce *RADIUS1*

s předchozími řešeními to však není provedeno). Hlasovací adjudikátor generuje odpověď na základě hlasování jednotlivých serverů, přičemž všechny servery mají v tomto případě stejnou váhu hlasu. Stejně tak odpovědi (kladné / záporné) mají stejnou váhu. Nepřijde-li v čekací lhůtě (viz přechod *Wait*) žádná odpověď nebo je kladných i záporných odpovědí stejně, vygeneruje adjudikátor zamítavé stanovisko. Nevýhodou tohoto řešení je fakt, že adjudikátor musí čekat dostatečně dlouho na pokud možno co největší množství odpovědí. Po provedení simulací a jejich vyhodnocení v subkapitole 5.5 bude patrné, že se jedná o časově nejpomalejší způsob řešení.



obr. 29 Vote adjudikátor

5.5 Ověření činnosti vylepšeného protokolu

Cílem předloženého modelu je simulovat činnost několika variant vylepšeného protokolu RADIUS v distribuovaném prostředí. Modely umožňují ověřit činnost vylepšení v různých konfiguracích a nastavení specifických parametrů (například časových konstant, pravděpodobností apod.), jak bylo naznačeno výše. Parametry a konfiguraci modelů je možné měnit a ověřovat tak hypotézy o chování počítačové sítě, například předpoklad, že s klesající mírou spolehlivosti sítě bude vylepšený RADIUS dosahovat lepších výsledků při autentizaci uživatele.

5.5.1 Vstupní a výstupní parametry modelů

Na úvod definujme vstupní a výstupní parametry. Vstupní parametry ovlivňují chování modelu a tím hodnoty výstupních parametrů, které sledujeme.
Mezi vstupní parametry modelu patří především:

- Počet autentizačních serverů v modelu
- Spolehlivost přenosové počítačové sítě
- Doba platnosti cache záznamu v databázi
- Velikost databází jednotlivých serverů (počet záznamů)
- Počet kombinací vstupních požadavků, které může Supplicant odesílat

Mezi výstupní parametry patří především:

- Úspěšnost autentizace uživatele
- Doba trvání autentizačního procesu
- Počet přenesených paketů
- Doba odezvy na jednotlivý požadavek o autentizaci

V následujících experimentech byly vstupní parametry zafixovány, kromě jednoho volného parametru – **spolehlivost počítačové sítě**. Tento parametr byl měněn a byly měřeny hodnoty výše uvedených výstupných parametrů.

5.5.2 Statistické zpracování naměřených dat

Vzhledem k tomu, že měřené parametry modelů jsou diskrétní náhodné veličiny, byly **úspěšnost autentizace uživatele**, **doba trvání autentizačního procesu** a **počet přenesených paketů** stanoveny jako **aritmetický průměr** z deseti jednotlivých měření pro každou hodnotu volného parametru.

Parametr **doba odezvy na jednotlivý požadavek o autentizaci** sice vykazuje podle naměřených hodnot ještě větší statistický rozptyl než předchozí parametry, nicméně pro stanovení aritmetického průměru bylo vždy použito pouze pěti naměřených hodnot. Důvodem k tomu je fakt, že odměření těchto parametrů s dostatečnou přesností by bylo nad míru časově náročné, protože je nutné měřit dobu odezvy na požadavek s kladnou autentizační reakcí a tento jev se vyskytuje zřídka. Naměření tohoto parametru s odpovídající přesností pro všechny varianty modelu by znamenalo řádově jednotky tisíc měření navíc.

Pro všechny naměřené hodnoty byla stanovena **střední kvadratická chyba aritmetického průměru**, určená všeobecně známým vztahem $\bar{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}(\bar{x}-x_i)^2}{n(n-1)}}$.

Naměřené hodnoty i s vypočtenou chybou jsou uvedeny v **příloze C** této práce.

5.5.3 Úspěšnost autentizace uživatele

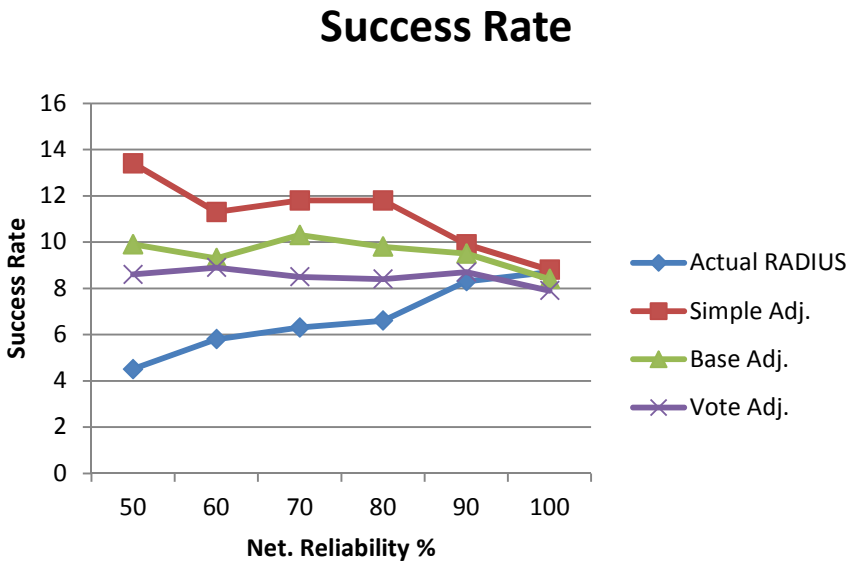
Tento parametr ukazuje úspěšnost autentizace pro sto požadavků o autentizaci pseudonáhodně vygenerovaných z dat umístěných v sekci *Supplicant*. Tato data jsou vždy pro všechny varianty modelu shodná. Model tedy vygeneruje náhodnou kombinaci uživatelského jména, realmu a hesla a tuto kombinaci zašle k autentizaci. Pokud je tato kombinace vstupních dat umístěna v databázi některého z autentizačních serverů (a je doručena), je autentizace považována za úspěšnou. Výhodou vylepšeného modelu je, že má možnost pokusit se o autentizaci na základě totožných vstupních dat opakovaně. Počet úspěšných pokusů se může zdát nízký, ale je třeba si uvědomit, že se jedná o pseudonáhodně sestavené kombinace vstupních dat.

Z grafu je patrné, že aktuální verze protokolu RADIUS má s klesající spolehlivostí sítě přibližně lineárně klesající úspěšnost autentizace. Vylepšený algoritmus protokolu naopak **vykazuje s klesající spolehlivostí sítě lineárně stoupající úspěšnost autentizace** (zejména pro Simple adjudikátor). Toto zásadně rozdílné **chování lze vysvětlit zvyšujícím se využíváním cache databází** jednotlivých poskytovatelů služeb při klesající spolehlivosti sítě. Protože s klesající spolehlivostí sítě dochází k častějším výpadkům přenosu dat, autentizační server poskytovatele služeb častěji opakuje výzvu o autentizaci a tím se zvyšuje šance na úspěšnou autentizaci. Prodlužuje se však doba autentizace a zvyšuje se počet paketů v síti.

Dosažené chování vylepšeného RADIUS protokolu nicméně může být výhodou při jeho nasazení v sítích s nízkou nebo výrazně kolísavou spolehlivostí (např. geograficky rozlehlé bezdrátové sítě v rozvojových zemích, bezdrátové sítě ve frekvenčně zarušených prostředích apod.).

Vylepšený protokol s Vote adjudikátorem má spíše konstantní průběh měřeného parametru. To je zřejmě dáno především množstvím hlasujících subjektů (tedy rozhodovacích RADIUS serverů), které je poměrně nízké. Výhodu při nasazení tohoto typu adjudikátoru by měli uživatelé, kteří relativně hodně „cestují“ mezi sítěmi a tedy mají šanci získat větší množství kladných

hlasů. Komplikací by ovšem byla častá změna hesla uživatelem, zejména pokud by nebyl implementován návrh podle subkapitoly 6.5.



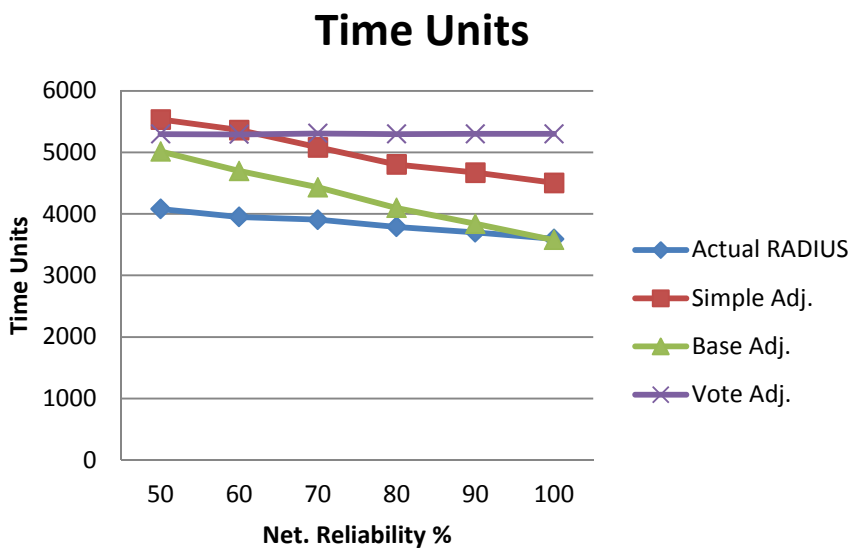
graf 1 Úspěšnost autentizace uživatele

Za zmínku stojí výsledky pro hodnotu spolehlivosti sítě 100%, kdy všechny varianty protokolu pracují podle aktuálního algoritmu a jejich výsledky by tak měly být shodné. Jejich nevýznamná rozdílnost je dána pouze chybou měření, která byla u jednotlivých měření stanovena, přičemž předložené výsledky jsou v rámci přípustné míry tolerance.

5.5.4 Doba trvání autentizačního procesu

Doba trvání autentizačního procesu je parametr, který udává celkovou dobu trvání autentizace pro sto pseudonáhodně vygenerovaných požadavků o autentizaci. Hodnota je uvedena v časových jednotkách, které jsou součástí CPN Tools. Srovnáním s reálnou činností protokolu by pak bylo možné získat skutečný čas. Z grafu 2 je vidět, že kromě Vote adjudikátoru mají všechny varianty protokolu tendenci s rostoucí spolehlivostí sítě lineárně snižovat časovou náročnost procesu autentizace. Base adjudikátor má pro hodnotu spolehlivosti sítě 100% dokonce přibližně shodnou časovou náročnost jako aktuální RADIUS protokol. To je dáno tím, že v Base adjudikátoru není třeba čekat na odpovědi ostatních serverů, pokud reaguje domovský server. V Simple

adjudikátoru je čekání na případné další reakce zavedeno a z toho důvodu je z grafu patrný konstantní rozdíl od Base adjudikátoru. Naproti tomu Vote adjudikátor má pevně stanovenou dobu čekání na „ukončení hlasování“, takže jeho charakteristika musí být konstantní, což jej znevýhodňuje vůči ostatním variantám pro použití ve spolehlivějších sítích.



graf 2 Úspěšnost autentizace

5.5.5 Počet přenesených paketů

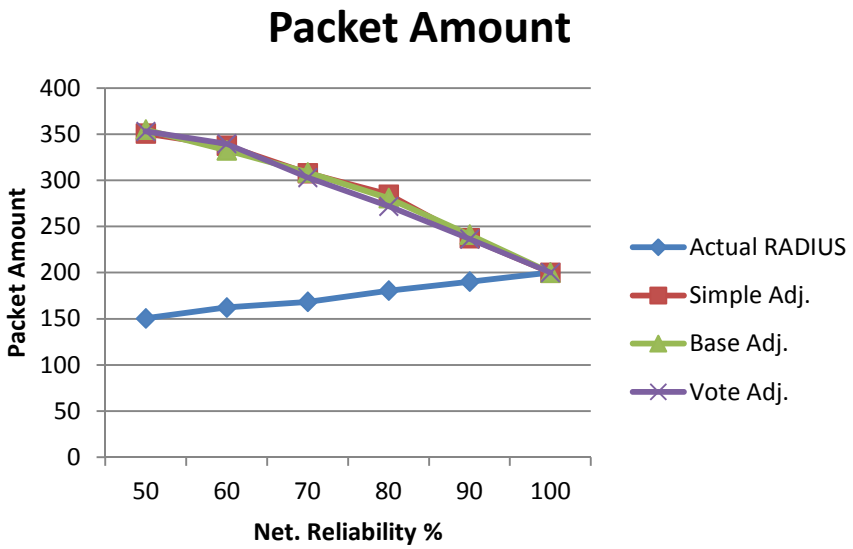
Tento parametr patří mezi nejdůležitější. V podstatě ukazuje, jak velké zatížení sítě při nasazení vylepšeného RADIUS protokolu můžeme očekávat. Metodika měření byla nastavena tak, že započteny byly odchozí pakety ze subsekce *Network* (a to v obou směrech). Chybové pakety, které se cestou v subsekci *Network* „ztratily“, se nezapočetly. Z výsledků uvedených v grafu pak plyne, že nezáleží na navrženém typu adjudikátoru, podstatná je pouze spolehlivost sítě. Pro uvažovanou nejnižší spolehlivost 50% dojde ke zvýšení přenosu paketů o cca 2,3násobek. Pro spolehlivost sítě 100% ke ztrátám paketů nedochází, a tak pro sto autentizačních požadavků a k nim náležejících odpovědí je počet přenesených paketů vždy právě 200.

Na základě naměřených dat je možné vyvodit empirický vztah mezi jednotlivými parametry sítě, který byl ověřen dosazením naměřených hodnot a který lze

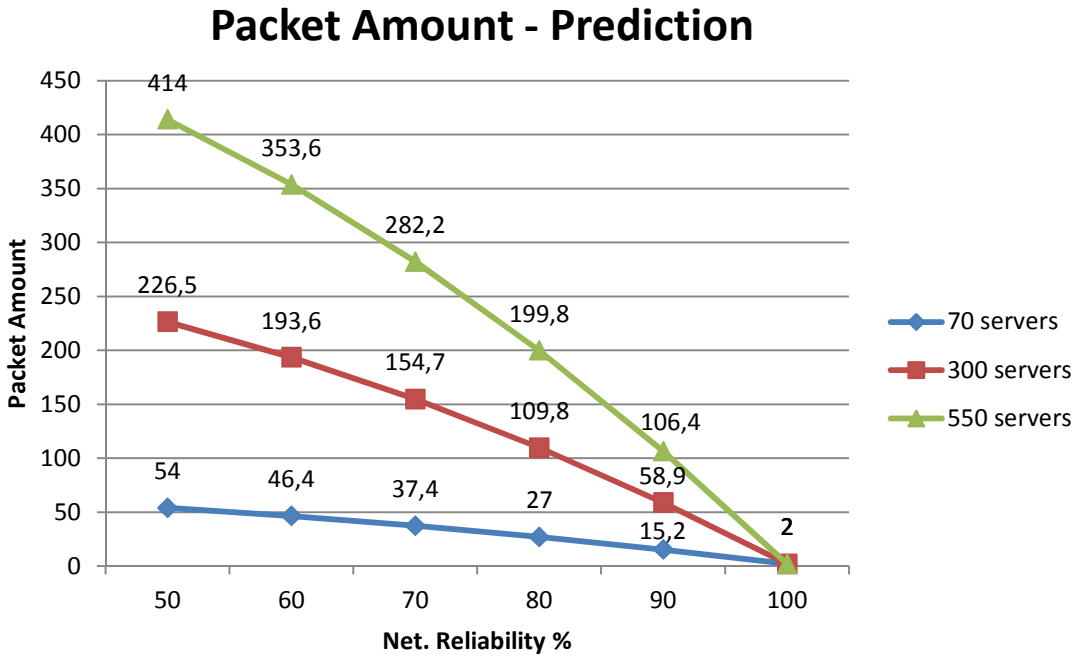
použít pro přibližnou predikci počtu přenesených paketů pro sítě s různými parametry.

$x = p + p.r + p.s.(1 - r) + r.[p.s.(1 - r)]$ *empirický vztah pro počet paketů*

$x = 1 + r + s - sr^2$ *upravený vztah pro jeden požadavek*



graf 3 Počet přenesených paketů - naměřeno



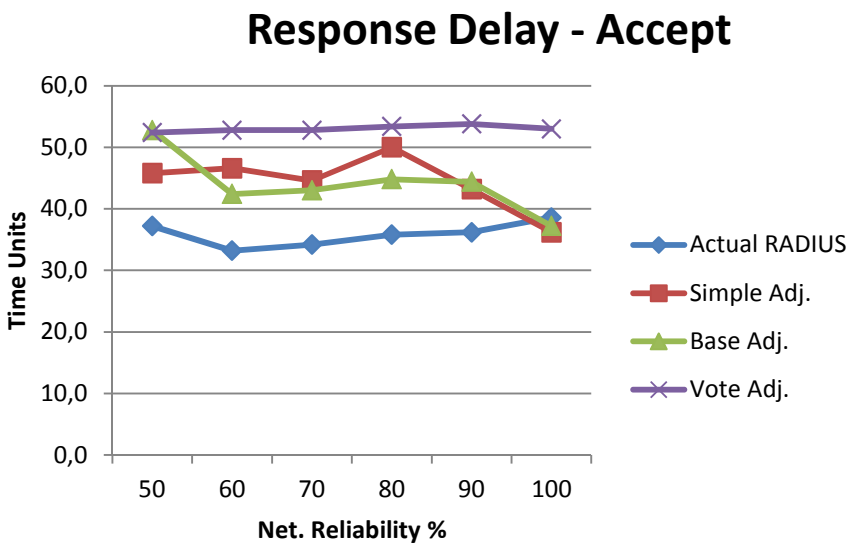
graf 4 Počet přenesených paketů - předpoklad

V uvedených vztazích je x počet paketů, p je počet autentizačních požadavků odeslaných do sítě, s je počet autentizačních serverů a r je spolehlivost sítě. Podle upraveného vztahu byl vytvořen graf 4, kde je počet autentizačních požadavků p v každém případě roven jedné a jednotlivé křivky odpovídají různým počtům autentizačních serverů (viz legenda grafu).

Z uvedených výsledků, které činí i pro poměrně rozsáhlé sítě (na úrovni států) řádově stovky přenesených paketů (tedy řádově stonásobky stávajícího stavu), je zřejmé, že nebude docházet k neúměrnému nárůstu počtu přenesených paketů a to ani pro případné nízké spolehlivosti sítě.

5.5.6 Doba odezvy na jednotlivý požadavek o autentizaci

V následujících grafech je zobrazena doba čekání na odpověď při požadavku o autentizaci. Vzhledem k tomu, že tyto parametry se vzhledem k velkému rozptylu měřených hodnot obtížně měří, je třeba je chápat spíše jako orientační. Metodika měření byla zvolena tak, že byl do modelu odeslán vždy jen jeden požadavek na autentizaci a pokud byla odpověď kladná, byl výsledný čas zahrnut do hodnot pro graf Accept a pokud byl výsledek záporný, byl výsledek zahrnut pro graf Reject. Každá hodnota v grafu je aritmetickým průměrem z pěti naměřených hodnot.

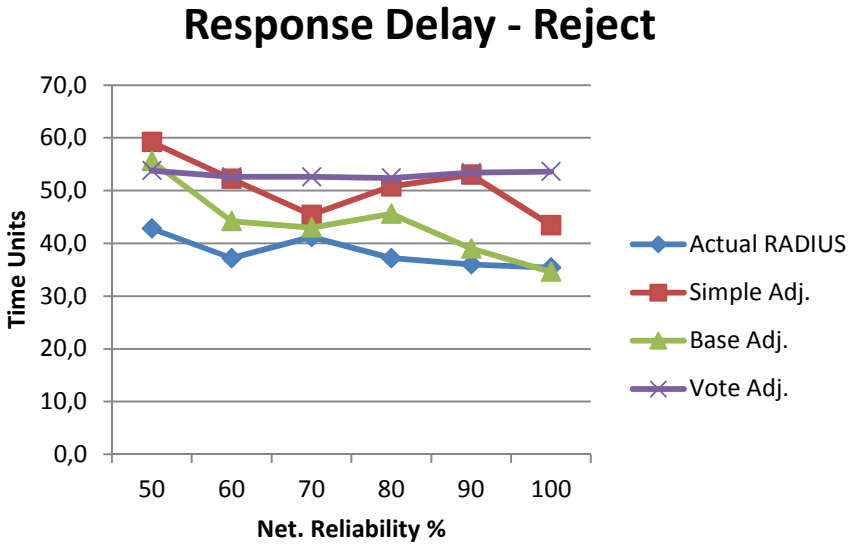


graf 5 Doba odezvy - Accept

Z grafů je patrné, že k výraznému nárůstu průměrné doby odezvy u vylepšeného protokolu proti aktuálnímu protokolu nedojde.

Z naměřených hodnot také vyplynulo, že doba odezvy se poněkud prodlouží při selhání domovského serveru. To koreluje s předchozí hypotézou, že vylepšený protokol bude provádět autentizační dotazy s mírně větší prodlevou. Stejně tak hypotéza, že doba odezvy pro kladné odpovědi bude kratší než pro záporné odpovědi, byla potvrzena.

Vzhledem k tomu, že na výši zpoždění autentizačního procesu má zásadní vliv konstrukce použitého adjudikátoru (nastavení jeho časových konstant), lze dovodit, že počet autentizačních serverů zapojených v síti bude mít na dobu odezvy autentizace marginální vliv.



graf 6 Doba odezvy - Reject

Opožděné reakce na opakovanou žádost o autentizaci, které do adjudikátoru mohou dorazit až po provedení rozhodnutí, jsou pak adjudikátorem zahozeny. Proto budou naměřené hodnoty přibližně stejné pro různý počet autentizačních serverů. Z toho důvodu bylo upouštěno od simulace závislosti doby odezvy na počtu autentizačních serverů.

5.6 Diskuse o dosažených výsledcích

Z výše uvedených měření je patrné, že vylepšený RADIUS protokol má ve srovnání s aktuálním protokolem poněkud odlišné chování. Zatímco **v aktuálním RADIUS protokolu parametr úspěšnost autentizace s klesající spolehlivostí sítě přibližně lineárně klesá, vylepšený protokol vykazuje opačný trend**. Je zjevné, že aktuální protokol není vhodný pro síťová prostředí, kde lze očekávat nízkou spolehlivost. Naopak ze zjištěných výsledků se dá předpokládat, že vylepšený protokol je pro prostředí s nespolehlivými sítěmi mnohem vhodnější. Navržený protokol sice vykazuje o něco delší prodlevy při řešení autentizačních požadavků (přibližně až o 30-35%, viz grafy), nejedná se ale o zásadní zpoždění, které by uživatele mohlo odradit od používání služeb sítě s nasazeným vylepšeným protokolem, neboť zpoždění je stále v řádu desetin sekundy.

Vylepšený protokol naopak v souladu s předpokladem, který mohl být předem učiněn, vykazuje zvýšení počtu přenesených paketů až o cca 200% oproti aktuálnímu protokolu. Toto zvýšení je nepřímo závislé na spolehlivosti sítě. Z tohoto faktu je možné usoudit, že by nasazení vylepšeného protokolu mohlo ohrozit fungování sítě tím, že při snižování spolehlivosti sítě (např. při přetížení směrovačů) by vylepšený RADIUS protokol zahlcoval systém dalšími pakety s požadavky na autentizaci a tím by roztočil spirálu vedoucí ke kolapsu sítě. Toto riziko lze považovat za jistých podmínek za reálné. Především se podle názoru autora může týkat softwarových implementací protokolu RADIUS, které by využívaly UDP pakety. UDP totiž jako takový na rozdíl od TCP postrádá nástroje pro řízení datového toku, resp. toto přenechává aplikačnímu protokolu. Implementace vylepšeného protokolu RADIUS, která by pro přenos využívala UDP datagramy, by mohla mít negativní vliv na provoz spolehlivé sítě. Toto riziko by však nemělo hrozit při implementaci vylepšeného protokolu RADIUS, který bude využívat protokol TCP. Tento protokol totiž obsahuje nástroje, které standardně využívá pro řízení datového toku a předchází tak zahlcení směrovačů. V tomto ohledu je také třeba poznamenat, že vhodně koncipovaná

topologie směrovačů v síti může předcházet problémům se zahlcením sítě (obecně, nejen v souvislosti s vyššími nároky vylepšeného protokolu RADIUS).

Navržený systém však může za určitých podmínek představovat už zmíněný **problém s autentizací na základě neaktuálního uživatelského účtu**, tedy za určitých podmínek může dojít k autentizaci například při použití zastaralého hesla, které může být uloženo v některé cache databázi a při výpadku domovského serveru se použije. Ačkoliv je toto riziko velmi malé, je tento problém řešen v subkapitole 6.5.

Z výše uvedených skutečností vyplývá, že se **nejeví vhodné využití byzantinských algoritmů** (viz podkapitola 4.3) pro řešení problematiky spolehlivosti protokolu RADIUS. Důvodem je jednak fakt, že tyto algoritmy jsou pro tuto problematiku nadmíru robustní. Mohou sice přinést do systému možnost přenosu kompletních autentizačních stanovisek jednotlivých serverů mezi všemi autentizačními servery navzájem, ale tato možnost se jeví jako redundantní. Konečné rozhodnutí je totiž vždy na RADIUS serveru poskytovatele služeb (na rozdíl od aktuálního protokolu, kde je rozhodnutí ve výhradní pravomoci poskytovatele identit) a šíření informací by spíše představovalo bezpečnostní riziko, protože přenášené zprávy obsahují autentizační stanoviska jednotlivých serverů, takže potenciální útočník by mohl snáze získat „přehled“ o prvcích systému. Dalším důvodem je fakt, že byzantinské algoritmy by byly mnohem náročnější na množství přenesených paketů. Při každém výpadku domovského serveru by totiž bylo nutné přenést při nasazení například šesti RADIUS serverů celkem patnáct paketů (pokud by všechny došly), zatímco předložený vylepšený protokol by vystačil jen s deseti pakety (a při nedoručení některého z nich by nevznikl následný problém). Při nárůstu počtu RADIUS serverů by se problém výrazně prohluboval (nejméně kvadraticky, viz subkapitola 4.3.2, vztah (3)). To by nepředstavovalo použitelné řešení.

6 NÁVRH INOVACE PROTOKOLU RADIUS Z HLEDISKA BEZPEČNOSTI

6.1 Základní východiska

Uživatelé distribuovaných sítí mohou měnit své chování ve vztahu k bezpečnosti jednotlivých částí sítě (připojených institucí) nebo sítě jako celku, například jinou než domovskou sít mohou zneužívat k útokům na vnitřní zdroje této sítě nebo na externí sítě.

Jestliže některý z uživatelů (definovaný konkrétním uživatelským účtem) *vykazuje chování, které porušilo bezpečnostní politiky daného poskytovatele služeb nebo by mohlo porušit bezpečnostní politiky domovského poskytovatele identity*, můžeme takové chování uživatele označit ve smyslu odstavce 2.1.3 jako **bezpečnostní incident**. V subkapitole 3.2 byl popsán problém, který bezpečnostně rizikovým chováním uživatelů vzniká a souvisí s problematikou předávání bezpečnostních informací.

V kapitole 5 bylo navrženo rozšíření protokolu RADIUS pro zlepšení spolehlivosti. Předložené řešení však umožňuje při splnění poměrně extrémních podmínek autentizaci na základě neaktuálního účtu, které bylo popsáno v odstavci 5.4.6.1. To by mohlo umožnit přístup do systému uživateli, kterému byla tato možnost poskytovatelem identity odebrána (například z důvodu spáchání bezpečnostního incidentu).

Je tedy třeba rozšířit protokol RADIUS o možnost zavádět zprávy přenášející informace o jednotlivých uživatelských účtech a chování uživatelů. Tento nástroj bude nutné doplnit o systém umožňující částečně automatickou formu sledování bezpečnostně-relevantní činnosti uživatelů v sítích jednotlivých členů federace. Informace o uživatelích v systému už nyní mohou shromažďovat jednotlivé zainteresované RADIUS servery v rámci svých účtovacích informací, nicméně systém účtování byl před lety navržen především z důvodu ekonomického provozu sítě a tedy se zaměřoval především na získání podkladů pro zpoplatnění uživatele za služby související s použitím sítě, proto

se nejeví jako vhodný pro sledování bezpečnostně-relevantní činnosti uživatelů a bude třeba jej rozšířit a vhodně propojit s jiným nástrojem pro odhalování bezpečnostních incidentů, mezi které patří především Intrusion Detection Systémy.

6.2 Analýza činnosti systému účtování

RADIUS Accounting se skládá z **accounting serveru** (AS) a **accounting klientů** (NAS). Accounting server RADIUS používá protokol UDP na portu 1813 (dříve 1646). Accounting subsystém protokolu RADIUS popisují standardy RFC2866 [15] a RFC2882 [33] pro přenos accounting informací mezi NAS a accounting serverem a mezi accounting servery navzájem. O účtování lze uvažovat jako o systému, který by mohl být více rozpracován tak, aby obsahoval architekturu pro správu uživatelů na federativní úrovni, kontrolu činnosti jednotlivých uživatelů a šíření informací o jejich chování.

Principem accounting systému jsou definované formáty zpráv (paketů), kterých je pro accounting pět typů (např. `Accounting-request`, `Accounting-response`, `Accounting-Message` atp). Tyto zprávy obsahují jeden nebo více atributů, kterých je pro accounting přibližně dvacet. Lze také definovat uživatelské zprávy a atributy. Pomocí těchto zpráv a atributů komunikují jednak server s klientem a jednak servery mezi sebou.

6.2.1 Komunikace mezi serverem a klientem

NAS (accounting klient) zašle accounting serveru `Accounting-request` paket, který obsahuje příslušnou událost. Accounting server odešle zpět `Accounting-response` paket, který obsahuje potvrzení o přijetí předchozího paketu.

Když se uživatel úspěšně připojí, je odeslán paket označovaný jako *Start*, viz obr. 30. Na účtovacím serveru je vytvořen záznam a informace z paketu jsou do něj uloženy. Záznam *Start* typicky obsahuje atributy `Session-Id`, `User-Name`, `Service-Type`, `Login-Service`, `Login-IP-Host`, `Acct-Delay-Time` a další relevantní informace o uživateli, resp. relaci.

```
Tue Jul 30 14:48:18 1996
Acct-Session-Id = "35000004"
User-Name = "bob"
NAS-IP-Address = 172.16.64.91
NAS-Port = 1
NAS-Port-Type = Async
Acct-Status-Type = Start
Acct-Authentic = RADIUS
Service-Type = Login-User
Login-Service = Telnet
Login-IP-Host = 172.16.64.25
```

obr. 30 Příklad záznamu Start (převzato z [13])

Když je relace ukončena, je vygenerován záznam *Stop*, viz obr. 31, který obsahuje podobné informace jako záznam *Start*, a navíc je v něm atribut obsahující délku trvání relace `Acct-Session-Time` v sekundách a atribut udávající důvod ukončení relace `Acct-Terminate-Cause`.

Tento postup může být podle [33] v některých případech rozšířen o přenos acounting paketů s jinými atributy, avšak na popsanou činnost účtování toto nemá výrazný vliv.

```
Tue Jul 30 14:48:39 1996
Acct-Session-Id = "35000004"
User-Name = "bob"
NAS-IP-Address = 172.16.64.91
NAS-Port = 1
NAS-Port-Type = Async
Acct-Status-Type = Stop
Acct-Session-Time = 21
Acct-Authentic = RADIUS
Acct-Input-Octets = 22
Acct-Output-Octets = 187
Acct-Terminate-Cause = Host-Request
Service-Type = Login-User
Login-Service = Telnet
```

obr. 31 Příklad záznamu Stop (převzato z [13])

6.2.2 Komunikace mezi acounting servery

Pro komunikaci mezi accounting servery byl v RFC2882 [33] definován typ paketu `Accounting-Message`. Tento typ paketu má jako ostatní typy paketů standardizovaný formát protokolu RADIUS, který je popsán v podkapitole 4.2.2.

Princip komunikace mezi RADIUS servery odpovídá způsobu komunikace, který byl popsán v předchozím odstavci. Tento typ paketu může přenášet jako data atributy definované v uvedených RFC. Může nést také uživatelské atributy, které jsou definovány dále v podkapitole 6.5.

6.2.3 Atributy účtování

Standardy RFC2866 [15] a RFC2882 [33] definují formát paketu a typy atributů, které se pro přenos accounting informací používají (viz též podkapitola 4.2.2). Každý atribut může obsahovat jeden nebo více parametrů (proměnných), jenž jsou vlastními nositeli informace obsažené v příslušném paketu.

6.2.3.1 Popis základních atributů účtování

Acct-Status-Type

`Acct-Status-Type` může nabývat několika hodnot, přičemž zajímavé jsou především dvě hodnoty: *Start* a *Stop*. Hodnota *Start* se generuje, když je relace uživatele započata, a hodnota *Stop* se generuje při ukončení příslušné relace.

Acct-Delay-Time

Obsahuje hodnotu v sekundách, která vyjadřuje dobu čekání od vzniku události do odeslání příslušného záznamu. Z tohoto údaje si může autentizační server přibližně vypočítat, kdy došlo k dané události.

Acct-Input-Octets a Acct-Output-Octets

Záznam o počtu přijatých (`Acct-Input-Octets`) a odeslaných (`Acct-Output-Octets`) oktetů během relace. Tyto hodnoty se objeví pouze ve *Stop* záznamech.

Acct-Session-Id

Jedná se o unikátní hodnotu, která slouží ke snadné identifikaci *Start* a *Stop* záznamů dané relace v log souboru.

Acct-Authentic

Tento parametr ukazuje, jakým způsobem byl přihlášený uživatel autentizován. Pokud je použit, může nabýt jedné ze tří hodnot: RADIUS, Local (autentizace místně pomocí NAS), Remote (autentizace s využitím nějakého externího protokolu).

Acct-Session-Time

`Acct-Session-Time` obsahuje délku trvání relace v sekundách. Atribut může být přítomen pouze v `Accounting-Request` paketu, kde `Acct-Status-Type` atribut nabyl hodnotu *Stop*.

Acct-Terminate-Cause

`Acct-Terminate-Cause` popisuje důvod ukončení relace. Tato informace je součástí paketu `Accounting-Request`, kde je nastaven `Acct-Status-Type` na *Stop*. Důvody mohou být následující:

1. User-Request — Ukončení na žádost uživatele, např. ukončením PPP.
2. Lost-Carrier — Při ztrátě signálu DCD[6], při chybě či zavěšení.
3. Lost-Service — Objeví se při ztrátě spojení mezi uživatelem a hostitelem.
4. Idle-Timeout — V případě vypršení doby nečinnosti.
5. Session-Timeout — Při vypršení časovače délky trvání relace.
6. Admin-Reset — Port / relace byla restartována administrátorem.
7. Admin-Reboot — Např. při restartu NAS administrátorem.
8. Port-Error — Při chybě na portu NAS, která vyžaduje konec relace.
9. NAS-Error — Při jiné chybě NAS, která vyžaduje konec relace.
10. NAS-Request — Při nechybových stavech, kdy ukončení požaduje NAS.
11. NAS-Reboot — Při nutném restartu NAS bez zásahu administrátora (crash).
12. Port-Uneeded — Když při využití více portů detekuje NAS slabé vytížení některého z nich.
13. Port-Preempted — Při nutnosti využít port NAS pro požadavek s vyšší prioritou.
14. Port-Suspend — Při pozastavení vituální relace NASem.
15. Service-Unavailable — NAS nemůže poskytnout požadovanou službu.
16. Callback — NAS ukončí aktuální relaci, když má vykonat zpětnou reakci pro novou relaci.
17. User-Error — Vstupní data od uživatele jsou chybná, např. se objeví PPP Configuration Request paket, když je již PPP relace navázana.
18. Host-Request — Relace byla ukončena na žádost hostitele normálně nebo při jeho výpadku.

Acct-Multi-Session-Id

Tento parametr je unikátním Accounting ID, kterým lze v log souboru označit několik vzájemně souvisejících relací. Každá ze souvisejících relací by pak měla mít vlastní `Acct-Session-Id`, ale stejné `Acct-Multi-Session-Id`.

Acct-Link-Count

Tento atribut obsahuje počet linků, které jsou součástí dané multilink relace, v době, kdy je vygenerován příslušný záznam. NAS může vložit atribut `Acct-Link-Count` do `Accounting-Request` paketu, který by mohl mít několik linků.

6.2.4 Zhodnocení systému účtování

Z výše uvedených atributů lze dovodit, že systém účtování je především zaměřen na zjištění, kdy přesně byla relace mezi uživatelem a NAS zahájena, kdy byla ukončena, jakým způsobem je uživatel připojen (např. pevnou linkou, komutovaným spojem atp.) a také jak velké množství dat bylo mezi uživatelem a NAS (resp. v opačném směru) přeneseno. Tyto informace jsou vhodné pro případné stanovení ceny za službu, nejsou ale v podstatě použitelné pro popis či snad dokonce detekci bezpečnostních incidentů. Možná snad jediný atribut `Acct-Terminate-Cause` by mohl být částečně použit ve vztahu k bezpečnostním incidentům, protože je zaměřen na popis důvodu pro ukončení dané relace. Nicméně většina parametrů tohoto atributu buď popisuje standardní důvody ukončení relace, nebo ukončení z důvodu selhání relace, avšak za těmito důvody lze spatřovat spíše technické než bezpečnostní chyby.
Z uvedeného plynou především dva závěry. **Jednak takto navržený a fungující systém účtování je pro šíření informací o bezpečnostních incidentech nevhodný a bude třeba jej rozšířit o nové atributy, aby mohl takovouto funkci plnit. Druhý závěr je, že systém není vůbec připraven na detekci bezpečnostních incidentů a jejich případné řešení. Na tuto záležitost tu jsou dobře rozvinuté IDS systémy, které disponují celou řadou vhodných detekčních metod a mají nástroje pro záznam událostí a jejich export do dalších systémů.** Využití těchto systémů v rámci rozšířeného účtovacího systému RADIUS protokolu by mohlo z bezpečnostního hlediska značně zdokonalit distribuované sítě s federativní autentizací založené na protokolu RADIUS.

6.3 Analýza systémů pro detekci průniku

6.3.1 Obecný úvod

Intrusion Detection Systém [34], [35] si definujme jako *obranný subsystém informačního systému pro detekci neobvyklých aktivit, které porušují nebo by mohly porušit bezpečnostní politiky systému.* IDS dosahuje této detekce

prostřednictvím monitorování síťového provozu a sledováním nejen aktivit, které přímo narušují bezpečnost systému, ale i aktivit, které by k narušení bezpečnosti mohly vést. Většina IDS při detekci neobvyklé aktivity provede zápis do systémového logu bezpečnostních událostí a v případě závažné události informuje administrátora systému. Existují také systémy označované někdy jako **Intrusion Prevention System** (IPS, příp. aktivní IDS apod.), které mohou násilně ukončit relaci, která bezpečnostní událost vyvolala (např. zasláním TCP datagramu s příznakem RST), a tím předcházet pokračování nebo dokončení potenciálního útoku. Rozdíly v činnosti mezi IDS a IPS jsou ve vztahu k této práci marginální, z toho důvodu zahrneme všechny IDS a IPS systémy pod jednotné označení IDS.

Z hlediska umístění v systému můžeme IDS rozdělit na:

- Host-Based-IDS (označované jako HIDS),
- Network-Based-IDS (NIDS).

Jak již napovídají jejich označení, HIDS jsou systémy umístěné na konkrétním hostiteli v systému (např. uživatelská stanice, aplikační server apod.) a slouží k detekci bezpečnostních událostí spojených s činností tohoto hostitele (resp. s uživateli, kteří jej aktuálně používají, z hlediska lokálního i vzdáleného). HIDS jsou softwarové aplikace, které sledují systémová volání v daném uzlu, práci souborového systému, činnost aplikací apod. HIDS mohou být součástí tzv. personálních firewallů [36], antivirů apod. Charakter jejich činnosti je převážně lokální a ve vztahu k řešenému problému nemohou přinést požadované přínosy, proto je dále nebudeme rozebírat.

NIDS jsou naopak systémy, které slouží pro monitorování síťového provozu jako celku (pokud možno). Mohou také sledovat chování jednotlivých hostitelů, prostřednictvím HIDS nasazených na hostitelích. NIDS se mohou nacházet na firewallech, routerech, switchích a dalších zařízeních pro administraci síťového provozu. Jejich aktivity jsou ve vztahu k této práci zajímavé.

6.3.2 Princip činnosti IDS

Současné IDS patří mezi základní součásti bezpečnosti počítačových systémů. Samy o sobě však nemohou být náhradou dalších bezpečnostních nástrojů

(firewallů, antivirových programů apod.). Tato práce není primárně zaměřena na problematiku IDS, které chápe jako vhodný zdroj informací o bezpečnosti systému, a proto cílem této subkapitoly je ilustrativní náhled do činnosti IDS.

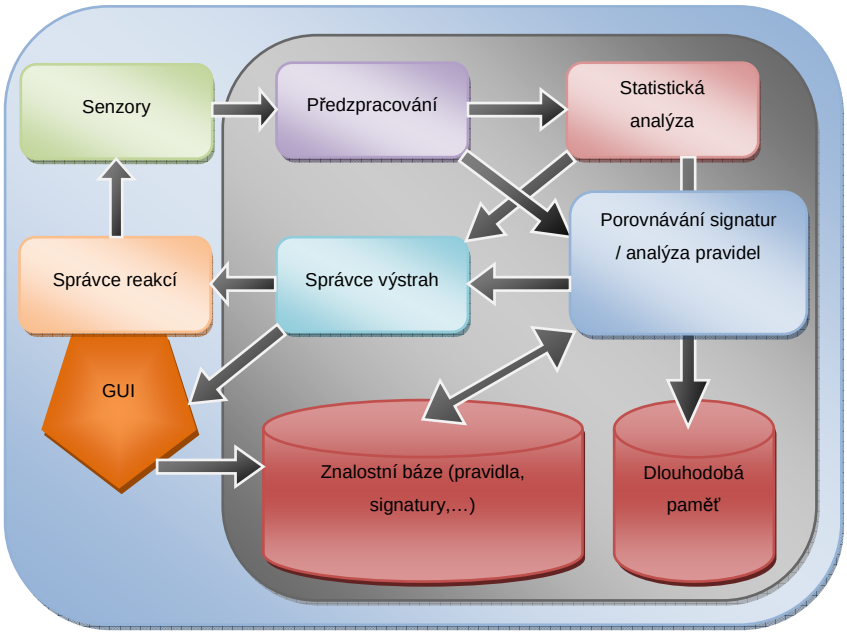
Mezi výhody IDS patří schopnost lépe zabezpečit počítačovou síť (informační systém). Pokud jsou správně nakonfigurovány a udržovány, pomáhají nalézat slabá místa v síti, poskytují možnost rychle reagovat na bezpečnostní incidenty a automatizovat odpovědi na tyto incidenty. Součástí IDS je také tvorba logů, možnost analýzy těchto logů a tvorba reportů.

Nevýhodou je zmíněná nutnost velmi dobré konfigurace a odladění systému IDS tak, aby nedocházelo k velkému množství falešných poplachů nebo naopak k ignoraci některých bezpečnostních incidentů. V případě sítí, které přenáší velké objemy dat, také mohou vznikat problémy s rychlostí vyhodnocování paketů systémem IDS.

Obecný IDS je složen podle [34] z několika základních částí, které jsou patrné na obr. 32. Jedná se o:

- Senzory – zařízení nebo softwarové agenty (jejich využití bude dále zmíněno v odstavci 6.4.1.3) umístěné ve vhodných místech v systému (např. na hostitelských počítačích či jako switche pro „odposlech“ veškerého provozu), slouží pro získávání informací z provozu sítě, z logů chování uživatelů apod.;
- Předzpracování – slouží k úpravám vstupních dat (dekódování, spojování fragmentů, oddělování hlaviček protokolů apod.);
- Statistická analýza – slouží pro statistickou analýzu běžného provozu a vytváření vzorců standardního chování sítě a jejích součástí; porovnává aktuální činnost sítě s uloženými vzorci chování;
- Porovnávání signatur / analýza pravidel – provádí analýzu aktivit získaných ze senzorů podle pravidel a signatur umístěných ve znalostní bázi;
- Znalostní báze – obsahuje uživatelsky vložené záznamy – signatury (otisky některých aktivit) a pravidla (jednoduché příkazy a podmínky pro analýzu vstupních dat);

- Správce výstrah – provádí analýzy reakcí a jejich filtrování (oddělení falešně pozitivních reakcí od skutečně pozitivních);
- Správce reakcí – zajišťuje centralizaci reakcí (výstrah, upozornění, oznámení) a přizpůsobuje činnost senzorů aktuálnímu stavu reakcí (zpětná vazba);
- Dlouhodobá paměť – ukládá především statistické informace o chování systému, které společně se systémem statistické analýzy provozu používá pro určitou formu „sebeučení“ a kontrolu aktuálního chování systému a jeho prvků;



obr. 32 Obecné schéma IDS

- Grafické uživatelské rozhraní (GUI), které nemusí být součástí všech systémů.

IDS používají několik hlavních přístupů k detekci průniků. Jednak je to porovnání signatur, jednak analýza pravidel a také u některých IDS analýza profilů.

Porovnání signatur (*Signature Matching*) pracuje na obdobném principu jako analýza signatur antivirových programů – tedy na faktu, že ve známých typech útoků se mohou vyskytovat známé řetězce znaků, na základě kterých lze tyto útoky detekovat. Tento způsob činnosti nebývá zdaleka tak efektivní, jak by se mohlo na první pohled zdát. Pravděpodobnost, že se přesným porovnáním

nalezeného řetězce s řetězcem umístěným ve znalostní bázi objeví útok, je poměrně malá. To je dáno tím, že je přirozenou snahou útočníků tento způsob odhalení obcházet a v útocích známé signatury nepoužívat. V tomto případě je možné nepoužívat přesné porovnání nalezeného řetězce se signaturou, ale porovnání částečné, které sice vykazuje častější detekce možných ohrožení, nicméně pochopitelně vyvolává častěji tzv. falešné poplachy (*False Positive Events*). Je tedy v tomto případě nutné dobře nastavit ohodnocení výše shody mezi nalezeným řetězcem a signaturou.

Analýza podle pravidel (*Rule Based Matching*) pracuje na principu pravidel uložených ve znalostní bázi (ve vhodném souboru) tato pravidla sestavuje administrátor přímo pro daný IDS. Většina oblíbených IDS má v rámci podpory uživatelů předpřipravená pravidla ke stažení, která si administrátor může samozřejmě upravit pro spravovaný systém nebo si vytvořit pravidla nová. Systém pravidel a jejich konfigurace je velmi podobná konfiguraci pravidel klasického firewallu nebo tzv. ACL (*Access Control Lists*) umísťovaných na vstupní/výstupní rozhraní routerů [37].

Příkladem může být toto zjednodušené pravidlo [35]:

`alert icmp` *`adresa_zdroje`* `any –>` *`adresa_cile`* `any (msg:“Large ICMP Packet“; dsize: >800; sid:103;)`

Toto pravidlo s identifikátorem 103 testuje velikost příchozího ICMP paketu a pokud je větší než nastavená mez (800 bytů), vygeneruje výstrahu a paket zaznamená do logu.

Správnou volbou a nastavením pravidel lze obecně dosáhnout velmi dobrých výsledků, kdy systém vydává výstrahu například až po sérii kroků útočníka, které by samostatně provedené výstrahy nevyvolaly.

Analýza profilů pracuje na principu strojového učení, tedy v tomto případě na tvorbě otisku „typického chování“ uživatele – vytvoření profilu chování uživatele. Tento otisk chování je pak porovnáván s aktuálním chováním uživatele a na základě shody, resp. neshody charakteristických markantů, je pak případně vyvolána výstraha. Příkladem může být sledování typického času a místa přihlašování uživatelů. Pokud je místo nebo čas výrazně odlišné od typického

chování (např. pokud se uživatel pokusí přihlásit v jiné kanceláři než obvykle, může být vyvolána výstraha, protože se může jednat o pokus o prolomení hesla).

6.3.3 Výstrahy IDS

Systémy IDS mají jednoznačně definované formáty výstrah (například hojně rozšířený **IDS Snort** [35]) a tyto výstrahy zaznamenávají textově do souboru logu (např. s názvem `alert.log` apod.). Vzhledem k textovému charakteru výstrah IDS je možné využít je pro předávání bezpečnostních informací v rámci protokolu RADIUS tak, že zejména ty nejzávažnější z nich (definované v tab. 1) budou předány společně s uživatelským jménem potenciálního útočníka do accounting systému protokolu RADIUS.

Číslo výstrahy	Označení výstrahy	Popis výstrahy
0	Attempted-admin	pokus o získání administrátorských práv
1	Attempted-user	pokus o získání uživatelských práv
2	Successful-admin	úspěšné získání administrátorských práv
3	Successful-user	úspěšné získání uživatelských práv
4	Web-application-attack	útok na webovou aplikaci
5	Attempted-dos	pokus o DOS
6	Denial-of-service	odhalení DOS
7	Successful-dos	DOS

tab. 1 Tabulka závažných výstrah

Ten je po vytvoření a implementaci nových atributů (zavedeny dále) bude moci předávat mezi členy federace. Závažné výstrahy byly definovány podle atributů systému IDS Snort, které byly subjektivně považovány za nebezpečné. Na základě zkušeností z praxe lze případně závažné atributy předefinovat.

6.4 Návrh řešení pro zlepšení bezpečnosti

Tato část práce byla částečně publikována v [A] a prezentována na [B].

6.4.1 Varianty komunikace mezi RADIUS a IDS

V této subkapitole budou nastíněny možnosti spolupráce mezi protokolem RADIUS a IDS nasazeným v konkrétní síti. Jednotlivé strategie přicházející

v úvahu stručně popíšeme a zdůvodníme výběr některé z nich. Mezi obecné strategie lze zařadit tři základní přístupy:

6.4.1.1 Metoda dotazování na aktuální stav (Pull strategie)

RADIUS protokol se dotazuje na aktuální stav zvoleného IDS v pravidelných časových intervalech.

- Výhody:
  - není potřeba žádná podpora na straně IDS (IDS nemusí ani vědět, že poskytuje další aplikaci své služby),
  - jednoduché řešení na straně protokolu RADIUS vyžadující ale novou SW implementaci protokolu,
- Nevýhody:
  - potenciálně pomalá odezva na vzniklý bezpečnostní incident (maximálně v délce dotazovacího intervalu),
  - nižší efektivita, je-li dotazovací interval příliš krátký (kratší než interval mezi bezpečnostními událostmi).
- Závěr:

Strategie je jednoduchá na implementaci a pro dále zkoumané řešení dostatečně efektivní. Navíc nevyžaduje zásahy do dalších entit kromě protokolu RADIUS.

6.4.1.2 Metoda subscribe-publish (Push strategie)

RADIUS protokol se registruje u IDS a při vzniku události je automaticky informován o změně stavu.

- Výhody:
  - prakticky okamžitá odezva na vzniklý bezpečnostní incident,
  - vyšší pružnost a škálovatelnost (u složitějších topologií),
- Nevýhody:
  - potřeba vhodně nastavit na straně IDS systém subscribe-publish (IDS obvykle umí posílat nějaká varování (e-mail, SNMP,...)) a na straně RADIUS serveru přijímat tato varování – potřeba změny implementace na straně protokolu RADIUS,

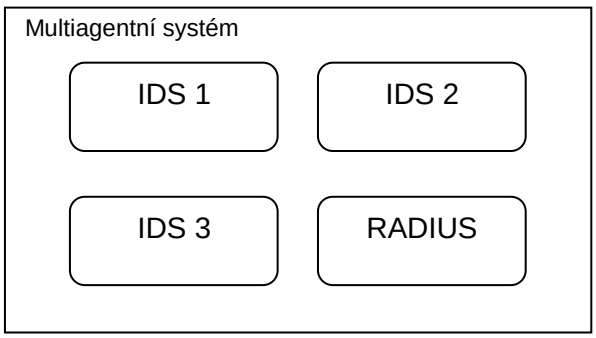
- možné vyšší zatížení systému jako celku při vyšší frekvenci událostí (události však mohou být filtrovány podle důležitosti nebo typu na straně publisheru nebo seskupovány při odesílání).

- Závěr:

Strategie vyžaduje zásahy do dalších entit kromě protokolu RADIUS a její přínosy nemohou vyvážit relativní složitost její implementace.

6.4.1.3 RADIUS jako softwarový agent v multiagentním IDS systému

RADIUS protokol spolupracuje se skupinou IDS instancí (agentů) při poskytování optimálního monitoringu bezpečnostních událostí a při jejich předávání mezi agenty vzájemně, viz obr. 33. RADIUS by tedy nebyl pouze konzumentem informací, ale mohl by příslušnému IDS poskytovat bezpečnostní informace získané z externích systémů. Toto řešení spolupráce je nejzajímavější, avšak náročné na implementaci, protože předpokládá nový pohled na IDS, na protokol RADIUS, na spolupráci mezi nimi a tvorbu nového multiagentního systému. Vytvoření nového multiagentního systému jen pro účely spolupráce s protokolem RADIUS by však nebylo z hlediska vynaložené energie efektivní. Jako vhodná se jeví integrace do vznikajícího multiagentního systému pracujícího s několika instancemi IDS v rámci konkrétní sítě, viz např. [38].



obr. 33 Příklad multiagentního systému

- Výhody:
  - obousměrná komunikace (RADIUS poskytuje vlastní informace za účelem dosažení optimální činnosti celého systému),
  - maximální integrace protokolu RADIUS do rozsáhlých IDS systémů,

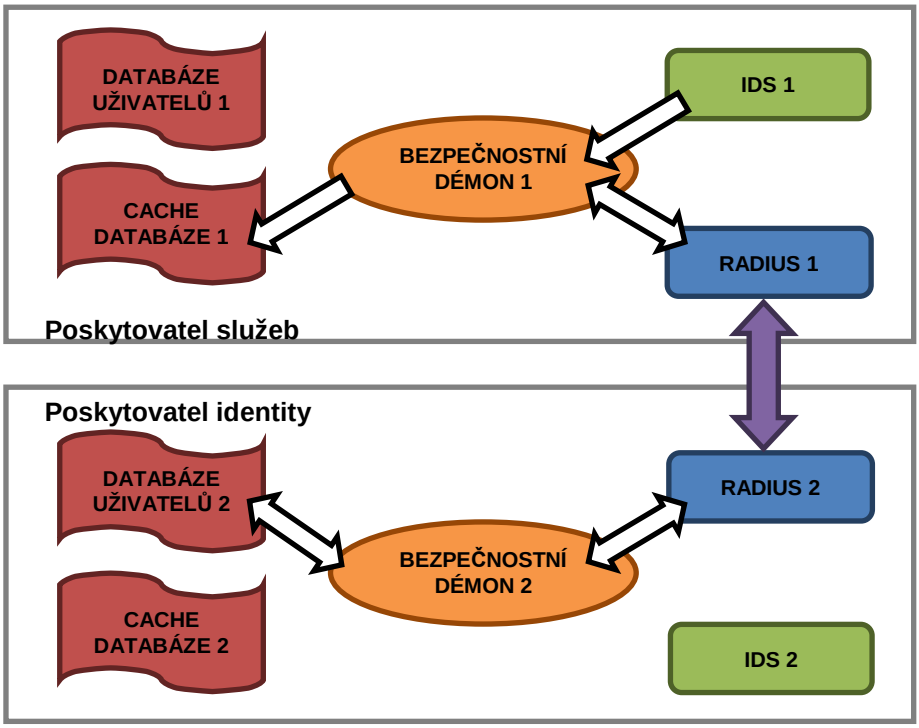
- Nevýhody:
  - nutnost podpory v IDS systému (řádově složitější než jednoduché zpracování eventů viz metody výše),
  - dvojitá odpovědnost protokolu RADIUS, která může vést ke kolizím zájmů; RADIUS je na jedné straně součástí systému RADIUS serverů, na straně druhé součástí multiagentního systému,
  - v souvislosti s předchozím bodem vzniká nutnost detailní specifikace pozice protokolu RADIUS v multiagentním systému a lokálního a globálního cíle.
- Závěr:

Strategie vyžaduje rozsáhlé zásahy do dalších entit kromě protokolu RADIUS. Nasazení této strategie by přineslo významné benefity pro celkovou bezpečnost počítačových systémů. Rozvoj předložené myšlenky může být tématem dalšího směřování výzkumu, nicméně rozsah nastíněné problematiky se nachází mimo rámec této práce.

6.4.1.4 Aplikace zvoleného řešení a zavedení nových atributů

Předávání vybraných výstrah z logovacího souboru do protokolu RADIUS accounting bude prováděno automatickým způsobem, přičemž existující možnosti byly naznačeny výše. Případný zásah administrátora by měl být zúžen jen na schválení předložených omezení uživatelů před odesláním dalším součástem federace. Pro svou relativní jednoduchost a snadnou implementaci do protokolu RADIUS byla jako mechanismus předávání informací mezi nasazeným IDS a protokolem RADIUS zvolena tzv. Pull strategie, viz odst. 6.4.1.1. Za získávání informací z IDS instance bude v rámci vylepšené implementace protokolu RADIUS zodpovědná nová entita, tzv. **bezpečnostní démon** (viz subkap. 6.4.2.5). Tento bezpečnostní démon bude ve stanovených intervalech prohledávat databázi výstrah příslušného IDS a vytěžovat informace podle klíčových slov výstrah (tab. 1, sloupec 2) zjištěné IDS. Přehled vztahů mezi prvky uvažovaného konceptu je na obr. 34.

Dále vygeneruje odpovídající bezpečnostní zprávu protokolu RADIUS Accounting opatřenou novými bezpečnostními atributy navrženými v tab. 2 a popsanými v kapitole 6.5. Na přijímací straně (tedy na serveru-příjemci, za kterého považujme server poskytovatele identity daného uživatele) bude úkolem vylepšené implementace protokolu (bezpečnostního démonu) podle



obr. 34 Princip předávání bezpečnostních informací

přijatého uživatelského jména zavést restrikce pro daného uživatele, například jej dočasně nebo trvale vyřadit z možnosti využívat služeb dané federace (pravidla určí lokální či federační politika, viz subkapitola 6.6).

6.4.2 Definice konkrétních případů a jejich procesních procedur

Navržený systém se zaměřuje na řešení následujících **případů**:

1. Uživatel způsobí bezpečnostní incident v síti poskytovatele služeb,
2. Uživatel způsobí bezpečnostní incident v síti svého poskytovatele identity,
3. Poskytovatel identity se rozhodne pro omezení uživatelského účtu z jiného důvodu,

4. Uživatel provede změnu hesla svého účtu a tuto změnu je třeba propagovat členům federace (z důvodu případného odstranění neplatného hesla z jejich databází), přičemž vlastní heslo se nepřenáší.

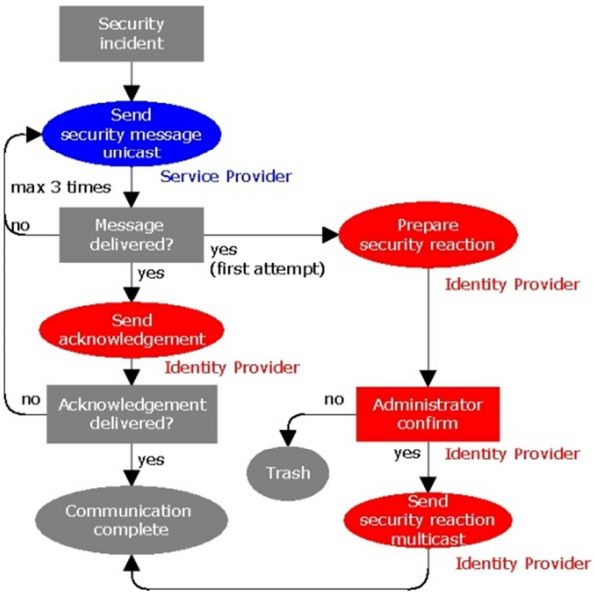
Pro řešení těchto případů budou navrženy atributy, viz tab. 2. Popis atributů je uveden v subkap. 6.5.

Číslo	Označení atributu	Popis atributu
145	Sec-Exchange	zapíná příjem bezpečnostních informací
146	Sec-Alert	předává bezpečnostní upozornění
147	Sec-Restriction	informuje o omezení už. účtu
148	Sec-Restriction-Period	informuje o délce omezení už. účtu
149	Sec-Restriction-Reason	informuje o důvodu omezení už. účtu
150	Sec-PassChange	informuje o změně hesla už. účtu

tab. 2 Definice nových atributů protokolu RADIUS

6.4.2.1 Procedury při řešení incidentu v cizí síti

Jedná se o nejdůležitější bezpečnostní situaci uvedenou výše jako **případ 1**. Uvažujeme, že bezpečnostní incident nastal v síti, kterou lze vůči uživateli, který jej způsobil, označit jako síť poskytovatele služeb.



obr. 35 Diagram činnosti algoritmu pro zasílání bezpečnostních zpráv

Bezpečnostní zprávu o incidentu generuje RADIUS server poskytovatele služeb, v jehož síti bezpečnostní incident nastal, a předává ji k řešení serveru poskytovatele identity příslušnému podle realmu uživatele, který incident

způsobil, a to standardní cestou formou unicast zprávy podle realmu uživatele. Zjednodušený diagram činnosti algoritmu je na obr. 35.

Odeslaná bezpečnostní zpráva obsahuje atribut `Sec-Alert` s hodnotou, která zařazuje způsobený incident (viz subkap. 6.5), a atribut `User-Name`, který jako hodnotu obsahuje jméno uživatelského účtu. Po přijetí zprávy bezpečnostní démon poskytovatele identity odešle potvrzovací zprávu odesílajícímu serveru obsahující atribut `Sec-Exchange` s hodnotou `3` a atribut `User-Name`. Pokud odesílající server tuto zprávu neobdrží, opakuje zaslání bezpečnostní zprávy znovu (nikoliv však nekonečně, například třikrát).

Dále bezpečnostní démon poskytovatele identity zajistí připravení reakce na bezpečnostní incident (restrikce na uživatelský účet) v souladu s interní bezpečnostní politikou poskytovatele identity, viz subkap. 6.6. Z procedurálního hlediska následuje rozeslání informace o restrikci uživatelského účtu nejlépe formou multicast zprávy všem členům federace. Zpráva obsahuje povinně atributy `Sec-Restriction` a `User-Name` a volitelně atributy `Sec-Restriction-Period` a `Sec-Restriction-Reason`. Rozeslání těchto zpráv však může trvat několik hodin i několik dní, protože připravená reakce na bezpečnostní incident může vyžadovat potvrzení administrátorem serveru.

6.4.2.2 Procedury při řešení incidentu v domovské síti

Případ 2 je pouze okrajovou situací **případu 1**, viz předchozí odstavec, kdy poskytovatel služeb je totožný s poskytovatelem identity. V tomto případě platí tedy přiměřeně předchozí odstavec. Není třeba generovat zprávu o bezpečnostním incidentu, bezpečnostní démon zajistí pouze připravení reakce na bezpečnostní incident (restrikce na uživatelský účet) v souladu s interní bezpečnostní politikou poskytovatele identity, viz subkap. 6.6, a rozešle informaci o restrikci uživatelského účtu nejlépe formou muticast zprávy všem členům federace. Zpráva obsahuje povinně atributy `Sec-Restriction` a `User-Name` a volitelně atributy `Sec-Restriction-Period` a `Sec-Restriction-Reason`.

6.4.2.3 Procedury při obecném omezení účtu

Situace uvedená jako **případ 3** patří také mezi okrajovou variantu **případu 1** a lze ji řešit přesně stejně jako v **případu 2**. Tento případ se využije v situacích, kdy se poskytovatel identity na základě nějakého administrativního popudu rozhodne omezit platnost uživatelského účtu. Příkladem takové administrativní situace může být přerušení studia studenta, který přestane být v právním vztahu k poskytovateli identity a tím by měl ztratit možnost využívat služeb distribuované sítě.

6.4.2.4 Procedury při propagaci změny hesla

Případ 4 lze využít pro záměr navržený v odstavci 5.3.2 provádět ve specifických případech autentizaci na serveru poskytovatele služeb. V takovém případě se jeví jako potřebné informovat servery poskytovatele služeb o změně hesla daného uživatelského účtu a předcházet tak případným neopodstatněným autentizacím na základě neaktuálního uživatelského účtu (viz odst. 5.4.6.1). Pokud uživatel změní heslo svého účtu, poskytovatel identity rozešle všem členům federace zprávu obsahující atributy `Sec-PassChange` a `User-Name`. Vlastní heslo se ve zprávě z důvodu bezpečnosti nepřenáší. V reakci na tuto zprávu zneplatní poskytovatel služeb příslušný záznam (existuje-li) ve své cache databázi.

6.4.2.5 Varianty bezpečnostních zpráv

Vzhledem k výše popsaným případům řešení je zřejmé, že v navrženém systému může být odesílán omezený počet smysluplných bezpečnostních zpráv, které mohou obsahovat konkrétní kombinace atributů.

Var.	Od.	Příj.	Atributy povinné a (volitelné)
A	PS	PI	`Sec-Alert,User-Name`
B	PI	PS	`Sec-Exchange, User-Name`
C	PI	PS	`Sec-Restriction, User-Name, (Sec-Restriction-Period), (Sec-Restriction-Reason)`
D	PI	PS	`Sec-PassChange, User-Name`
E	PS	PI	`Sec-Exchange`

tab. 3 Varianty bezpečnostních zpráv

Pro přehlednost uvádíme výčet přípustných bezpečnostních zpráv a jejich atributů v následující tabulce, kde PS označuje poskytovatele služeb a PI

označuje poskytovatele identity. Předpokládá se, že každá zpráva může obsahovat každý přípustný atribut nejvýše jednou.

Bezpečnostní zpráva A se použije pro odeslání upozornění na bezpečnostní incident, přičemž příjemce (poskytovatel identity) ji potvrdí odesláním bezpečnostní zprávy B.

Pro informace o omezeních uživatelského účtu využijí poskytovatelé identity hromadně odesílanou bezpečnostní zprávu C.

Bezpečnostní zprávu D hromadně odešlou poskytovatelé identity při změně hesla uživatelem.

Zpráva E se může v budoucnu využít k tomu, aby některý poskytovatel služeb byl ze systému bezpečnostních informací zcela nebo částečně vyřazen.

6.4.3 Koncept bezpečnostního démonu

Činnost bezpečnostního démonu můžeme rozdělit ze dvou hledisek. Jednak je to činnost z hlediska poskytovatele služeb a jednak činnost z hlediska poskytovatele identity. Výsledná implementace démonu však bude tvořit jeden celek. Technické detaily provedení a činnosti démonu budou součástí konkrétní softwarové implementace vylepšeného protokolu RADIUS.

6.4.3.1 Bezpečnostní démon z hlediska poskytovatele služeb

Z tohoto hlediska má démon řešit především situace podle **případu 1**, tj. prohledávání logovacího souboru výstrah v pravidelných intervalech (řádově hodiny) s cílem hledat závažné výstrahy navržené v tabulce na straně 76. Po nalezení výstrahy vygeneruje démon zprávu o bezpečnostním incidentu, která obsahuje atribut `Sec-Alert` a atribut `User-Name`. Tuto zprávu odešle serveru poskytovatele identity podle realmu uživatele, který bezpečnostní incident způsobil. Bezpečnostní démon také reaguje na přijaté zprávy s atributy `Sec-Restriction` a `Sec-PassChange`. Jeho úkolem je v těchto případech u záznamu, který se obsahově shoduje s atributem `User-Name`, nastavit/zrušit v databázi příznak dočasného omezení účtu (záznam se (ne)bude při autentizaci používat) na základě atributu `Sec-Restriction` (hodnota 0/1) nebo příslušný

záznam zrušit na základě atributu `Sec-Restriction` (hodnota 2) nebo `Sec-PassChange`.

6.4.3.2 Bezpečnostní démon z hlediska poskytovatele identity

Činnost démonu je zaměřena především na přijímání bezpečnostních zpráv od poskytovatelů služeb a s tím související změny záznamů v databázi uživatelů spočívající především ve změně příznaku, který způsobí omezení autentizace příslušného uživatele. Poté je vygenerována příslušná zpráva, která je odeslána poskytovatelům služeb.

Na základě pokynu administrátora ve smyslu **případu 2** či **případu 3** vygeneruje také démon zprávu s atributem `Sec-Restriction` při zavedení nebo zrušení restrikcí týkající se uživatelského účtu. Tyto restrikce administrátor zavádí/ruší na základě interní politiky poskytovatele identity.

Dále démon kontroluje změnu hesla uživatelských účtů, přičemž v případě změny hesla generuje příslušnou zprávu s atributem `Sec-PassChange` a rozesílá ji poskytovatelům identity. Tento vztah bezpečnostního démonu s databází uživatelů je snadno řešitelný (např. doplněním databáze uživatelů o parametr změny hesla, který bude sledován bezpečnostním démonem).

Posledním úkolem démonu by mělo být informování dotčených uživatelů, aby se mohli snadno dozvědět o svých pochybeních a restrikcích. Vhodným řešením by mohlo být budoucí propojení démonu s nějakou webovou aplikací, popř. automatické odeslání emailu přímo dotčenému uživateli.

6.4.4 Bezpečnostní aspekty navrženého systému

Nově navržený systém v mnohém rozšiřuje stávající činnost protokolu, která doposud spočívá, jak bylo již uvedeno, především v rovině účtovací. Na druhou stranu však rozšíření činnosti obvykle obecně zvyšuje bezpečnostní rizika systému.

V námi navrženém rozšíření lze spatřovat rizika v tom, že uživateli, který nic nezpůsobil a ani se o bezpečnostní incident nepokusil, bude omezena činnost uživatelského účtu neoprávněnou restrikcí. Anebo v opačném případě bude uživateli, který bezpečnostní incident způsobil, restrikce předčasně

neoprávněně zrušena nebo nebude vůbec zavedena. Tyto situace označme jako **neoprávněnou změnu stavu účtu.** K tomu může dojít jednak chybnou bezpečnostní zprávou a jednak útokem spočívajícím v neoprávněném zaslání bezpečnostní zprávy.

Prvnímu okruhu problémů – tedy odeslaná chybná zpráva – se zřejmě nelze zcela vyhnout (lidská chyba je možná prakticky vždy a administrátor může neúmyslně zaslat chybnou zprávu), ale problém lze je považovat za marginální. Technické chybě při přenosu zabrání běžně používané základní nástroje pro přenos zpráv a nižší vrstvy síťové architektury chybnou zprávu vůbec nedoručí.

Druhý okruh problémů – tedy zásah útočníka – souvisí s obecným zajištěním bezpečnosti na úrovni protokolu RADIUS. To je dnes již záležitost uspokojivě vyřešená prostřednictvím v úvodu zmíněného protokolu RadSec [22], který řeší bezpečnost komunikace mezi dvěma RADIUS servery prostřednictvím PKI a šifrování komunikace. Problematiku zajištění základní bezpečnosti komunikace mezi RADIUS servery považujeme za vyřešenou, není ji tedy třeba v rámci této práce dále rozvíjet.

6.5 Bezpečnostní zprávy a atributy

Navržené řešení inovace protokolu klade na stávající verzi systému RADIUS nové požadavky. S návrhem nových funkcionalit protokolu souvisí také potřeba nových nástrojů, které protokol bude nezbytně potřebovat, aby mohl navrženým požadavkům dostát. Mezi tyto nástroje patří zejména skupina nových atributů, které dovolí přenášet dosud nevyžadované informace. Z logiky fungování protokolu plyne skutečnost, že atributy musí být při komunikaci součástí některého typu zprávy protokolu RADIUS (viz odstavec 4.2.2). Stávající nabídka typů zpráv je podle IANA [39] poměrně široká. Nabízí se existující typ zprávy `Accounting-Message`, ale vzhledem k přehlednosti, a pro případ budoucího nárůstu atributů bezpečnostního typu, považujeme za vhodnější zavést nový typ zprávy (bezpečnostní).

6.5.1 Typ zprávy Security-Message

Zprávy jsou v rámci protokolu RADIUS definovány hodnotou pole **Code** viz odst. 4.2.2. Hodnoty tohoto pole registruje IANA, přičemž k termínu vzniku této práce byla volná například hodnota 35. V dalším textu budeme předpokládat, že nově zavedenému typu zprávy byla přidělena hodnota Code = 35. Je tedy možné definovat v rámci standardního formátu zpráv protokolu RADIUS nový typ bezpečnostní zprávy:

35 Security-Message

Tento typ zpráv může obsahovat atributy `User-Name`, `Sec-Exchange`, `Sec-Alert`, `Sec-Restriction`, `Sec-Restriction-Period`, `Sec-Restriction-Reason` a `Sec-PassChange`. V následujícím odstavci budou detailněji popsány uvedené bezpečnostní atributy.

6.5.2 Popis atributů bezpečnostního typu

Nově navržené **atributy bezpečnostního typu** (viz tab. 2) budou sloužit především pro výměnu bezpečnostních informací mezi členy federace. Jeden atribut mohou členové federace využít pro nastavení nebo omezení zasílání bezpečnostních informací. Pro návrh nových atributů je použit standardizovaný formát atributu obsažený v dokumentu RFC 2865, viz tab. 4.

0								1								2							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Type								Lenght								Value							
Value (cont)																							

tab. 4 Formát atributu protokolu RADIUS

6.5.2.1 Atribut Sec-Exchange

Tento atribut by měl sloužit pro možnost zapínat nebo vypínat přijímání bezpečnostních informací v rámci RADIUS Accounting. Implicitní hodnota by měla být 0 (*On*), pro zastavení příjmu zpráv slouží hodnota 1 (*Off*). Hodnota 2 (*Selected*) může sloužit pro nastavení zasílání pouze vybraných informací v rozsahu daném dohodou mezi provozovateli sítí, příp. podle implementace protokolu. Hodnota 3 *(Alert Accepted)* se použije jako potvrzení o doručení

zprávy s atributem `Sec-Alert`. Činnost atributu by měla být na systémové úrovni, tedy žádné bezpečnostní zprávy nebudou od odesílatele šířeny k těm příjemcům, kteří předem avizují nastavení hodnoty 0, resp. vybrané zprávy nebudou k příjemci šířeny, pokud avizuje hodnotu 2. Zavedení tohoto systému má charakter budoucího rozšíření, protože vyžaduje opatřit příslušnou implementaci protokolu nástroji, které budou zodpovídat za správnou činnost tohoto systému (mj. bude třeba vytvořit databázi členů bezpečnostního systému s nastavením jejich hodnot). Ponechme na dalších úvahách, na jaké úrovni by bylo optimální takový systém zavést a zda to bude mít očekávaný přínos. V bližší budoucnosti by měla implementace protokolu RADIUS umožnit vypnutí přijímání bezpečnostních zpráv jejich prostým zahazováním, ale tento postup rozhodně nelze doporučit.

Charakteristika atributu v souladu s RFC2865:

```
Type
145 for Sec-Exchange.
Lenght
3
Value
The Value field is one octet.
0 On
1 Off
2 Selected
3 Alert Accepted
```

6.5.2.2 Atribut Sec-Alert

Tento atribut zasílá bezpečnostní informace o uživateli, který způsobil bezpečnostní incident, jež je vyjádřen hodnotou uvedenou v poli *Value*. Hodnoty mohou být v průběhu času dále doplňovány. Zpráva, která obsahuje tento atribut, musí obsahovat také atribut `User-Name`.

Charakteristika atributu v souladu s RFC2865:

```
Type
146 for Sec-Alert.
Lenght
3
Value
The Value field is one octet.
0    Attempted-admin
1    Attempted-user
2    Successful-admin
```

```
3     Successful-user
4     Web-application-attack
5     Attempted-dos
6     Denial-of-service
7     Successful-dos
```

6.5.2.3 Atribut Sec-Restriction

Atribut by měl využívat poskytovatel identity uživatele, který způsobil v některé síti federace (včetně domovské sítě) bezpečnostní incident a byl mu omezen přístup do sítí federace. Dobu restrikce (dočasného odmítání přístupu do sítě federace) stanoví poskytovatel identity. Informace, zda uživatel má být omezen při využití služeb sítě, nemá být omezen nebo je jeho uživatelské konto zcela zrušeno, je uvedena hodnotou *Value*. Zpráva, která obsahuje tento atribut, musí obsahovat také atribut `User-Name`.

Charakteristika atributu v souladu s RFC2865:

```
Type
147 for Sec-Restriction.
Lenght
3
Value
The Value field is one octet.
0     User-unrestricted
1     User-restricted
2     User-deleted
```

6.5.2.4 Atribut Sec-Restriction-Period

Atribut by měl volitelně využívat poskytovatel identity uživatele, který způsobil v některé síti federace (včetně domovské sítě) bezpečnostní incident a byl mu omezen přístup do sítí federace. Datum ukončení restrikce vhodně zakódováno v poli *Value*. Zpráva, která obsahuje tento atribut, musí obsahovat také atributy `Sec-Restriction` a `User-Name`.

Charakteristika atributu v souladu s RFC2865:

```
Type
148 for Sec-Restriction-Period.
Lenght
4
Value
The Value field is two octets.
0-65535 End date of restriction.
```

6.5.2.5 Atribut Sec-Restriction-Reason

Atribut by měl využívat poskytovatel identity uživatele, který způsobil v některé síti federace (včetně domovské sítě) bezpečnostní incident a byl mu omezen přístup do sítí federace. Hodnota *Value* se přebírá z obdrženého atributu `Sec-Alert` a vyjadřuje důvod omezení uživatelského účtu. Tento důvod může využít příjemce pro vnitřní potřebu, např. v budoucnosti jej lze použít vnitřním bezpečnostním systémem příjemce (např. IDS). Zpráva, která obsahuje tento atribut, musí obsahovat také atributy `Sec-Restriction` a `User-Name`.

Charakteristika atributu v souladu s RFC2865:

```
Type
149 for Sec-Restriction-Period.
Lenght
3
Value
The Value field is one octet.
0    Attempted-admin
1    Attempted-user
2    Successful-admin
3    Successful-user
4    Web-application-attack
5    Attempted-dos
6    Denial-of-service
7    Successful-dos
255  Administrative-Reason
```

6.5.2.6 Atribut Sec-PassChange

Tento atribut slouží pro informování mezi členy federace o změně hesla konkrétního uživatele. Atribut rozešle poskytovatel identity při změně hesla uživatelem. To způsobí zneplatnění záznamů v cache databázi poskytovatelů služeb. Atribut je vždy zasílán společně s atributem `User-Name`. Atribut neobsahuje pole *Value*.

Charakteristika atributu v souladu s RFC2865:

```
Type
150 for Sec-PassChange.
Lenght
2
Value field not included.
```

6.6 Příklad bezpečnostních politik systému

V rámci navrženého systému je možné omezovat jednotlivým uživatelům přístup do sítě. Není pochybností o tom, že tento systém musí být nastaven transparentním způsobem tak, aby uživatelům sítě bylo předem jasné, za jakých podmínek k omezení může docházet, a tedy jaké sankce v případě porušení pravidel hrozí. Není záměrem zde nastavit obecně platná pravidla, neboť jsou to poskytovatelé identit, kteří nesou za registraci svých uživatelů zodpovědnost. Tedy poskytovatelé identit by především měli mít zájem na používání navrženého systému a na nastavení odpovídajících pravidel a sankcí. **Navržené bezpečnostní politiky** je tedy možné chápat jako **příklad či doporučení** k nastavení individuálních pravidel jednotlivých poskytovatelů identit (nicméně bylo by vhodné používat u jednotlivých poskytovatelů identit pravidla obdobná).

Jedním z hlavních cílů je zabránit opakovaným útokům, které nečiní uživatel vědomě, ale činí je např. vir, který počítač uživatele zahrne do botnetu[7]. V tomto případě má být hlavním smyslem restrikce donutit uživatele provést individuální bezpečnostní opatření. Vhodné je tedy kratší odpojení uživatele od služeb, spojené s informováním (např. „přes Váš účet byl proveden útok..., prosím spusťte antivir“). Po ukončení restrikce by byla při dalším útoku provedena trvalejší restrikce (s informací, kde se může uživatel informovat dál).

Číslo výstrahy	Označení výstrahy	Koeficient sankce k_s
0	Attempted-admin	3,0
1	Attempted-user	2,0
2	Successful-admin	3,0
3	Successful-user	2,0
4	Web-application-attack	1,0
5	Attempted-dos	1,0
6	Denial-of-service	1,0
7	Successful-dos	1,0

tab. 5 Návrh sankcí za porušení pravidel

Při restrikci uživatele navrhujeme vycházet z navržených bezpečnostních incidentů a časového omezení uživatelského účtu na násobky doby, kterou označme jako **základní dobu** (ZD). Hodnotu základní doby stanovme empiricky

na tři dny, což je zřejmě běžná doba, za kterou stihne uživatel sjednat nějakou nápravu věci. Za jednotlivé incidenty způsobené uživatelem poprvé by byla zavedena **restrikce na násobky základní doby** navržené v tab. 5 jako **koeficient k_s**.

Při opakování incidentu stejným uživatelem by se doba restrikce prodlužovala, což by mělo působit výchovně při neplnění povinností uživatele.

Vhodný způsob prodlužování doby restrikce by mohl vycházet z následující funkce:

$$t_r = 2^{n-1} \times k_s(L) t_b$$

kde

- t_r je celková doba restrikce ve dnech
- n je počet opakování útoků za určenou dobu (např. za jeden rok)
- t_b označuje základní dobu restrikce
- L je typ útoku $L \in \{0, 1, \dots, 7\}$, viz tab. 5 (číslo výstrahy)
- k_s je funkce mapující typ útoku na koeficient trvání restrikce $(k_s : \{0, 1, \dots, 7\} \rightarrow R)$, viz tab. 5.

Dále bylo uvedeno, že poskytovatel identity může omezit uživatelský účet z administrativních důvodů (např. při přerušení studia uživatele-studenta). Zde je doba omezení obvykle jeden, dva semestry, tedy několik měsíců.

Všechny důvody restrikcí a samotné restrikce by měly být pro uživatele vhodným způsobem prezentovány, např. formou emailu uživateli či po zadání nějakého osobního údaje (emailová adresa, datum narození apod.) prostřednictvím webové aplikace. Stejně tak je třeba propagovat na venek „z odstrašujících důvodů“, tedy jako prevenci, přehled restrikcí s důvodem jejího vzniku. V tomto případě je však třeba dbát na ochranu osobních údajů a zobrazovat pouze předem vhodně upravený přehled bez identifikačních údajů uživatelů, popř. pouze nějakou statistiku.

Závěrem je třeba uvést, že bezpečnostní incidenty mohou vzniknout špatným vyhodnocením chování uživatele či jinou vnitřní chybou systému. Z toho důvodu je třeba dát uživateli možnost odvolat se, případně své chování vysvětlit. Bezpečnostní politiky poskytovatelů identit by tak měly obsahovat pravidla pro

odvolání uživatelů a řešení těchto případných odvolání. Případné úspěšné odvolání uživatele by pak mělo znamenat snížení nebo zrušení restrikce, což vyvolá potřebu opětovného rozeslání bezpečnostní zprávy s atributem `Sec-Restriction-Period`.

7 ZÁVĚR

7.1 Zhodnocení dosažených výsledků

V souladu s definovanými cíli práce byly dosaženy následující hlavní výsledky:

1. Byl navržen algoritmus pro zajištění autentizace při výpadku domovského autentizačního serveru uživatele,
2. Navržený algoritmus byl simulován s použitím barevných Petriho sítí a byla ověřena jeho schopnost úspěšně řešit předložený problém v rozsáhlých sítích s nízkou spolehlivostí,
3. Bylo ověřeno na vytvořeném modelu, že algoritmus je schopen činnosti v asynchronním distribuovaném prostředí,
4. Byl navržen způsob řešení výměny informací o bezpečnostních událostech spojených s uživateli s využitím volitelných atributů protokolu RADIUS a systémů IDS.

Ad 1)

Navržený algoritmus předpokládá změnu způsobu chování protokolu RADIUS tak, že v případě výpadku autentizace z domovského serveru uživatele budou osloveny servery poskytovatelů služeb, které budou dovybaveny specifickou databází autentizačních informací. V případě výpadku autentizace z domovského serveru se odpovědnost za autentizaci přesouvá z poskytovatele identity na poskytovatele služeb. V tomto smyslu bude nutné také upravit Roamingovou politiku [4] nebo interní směrnice příslušné federace.

Ad 2)

Subkapitola 5.4 je věnována popisu vytvořeného modelu vylepšeného protokolu RADIUS. Tento model vytvořený v aplikaci CPN Tools je softwarovou přílohou B.1 – B.3 této práce. V subkapitole 5.6 jsou následně popsány a okomentovány výsledky provedených měření na předloženém modelu. Z nich je patrné, že navržený model je přínosný pro řešení předloženého problému a vytvoření jeho softwarové implementace nasazené do praxe by mělo význam pro zlepšení služeb zajišťovaných protokolem RADIUS.

Ad 3)

V rámci tvorby disertační práce byla pozornost věnována i algoritmům zajišťujícím konsensus v distribuovaném prostředí (především subkapitola 4.3) a bylo prověřeno, že navržený algoritmus nebude nucen používat nadmíru robustní algoritmy pro rozhodování v distribuovaných systémech a bude schopen činnosti i v asynchronním distribuovaném systému (tento závěr vychází se samotného modelu, který funguje jako asynchronní).

Ad 4)

V kapitole 6 bylo rozebráno, proč RADIUS Accounting protokol nelze použít pro vytvoření nástroje na sledování uživatelů. Bylo navrženo využít IDS a byly navrženy postupy, jak dosáhnout propojení těchto dvou nezávislých systémů a zajistit přenos bezpečnostně relevantních informací mezi členy federace. Implementace navrženého řešení by vyřešila jednak problém s rizikovým chováním uživatelů i s problémem s autentizací na základě neaktuálního uživatelského účtu (viz odst. 5.4.6.1).

7.2 Otevřené problémy disertační práce

Práce je zaměřena na analýzu problémů distribuovaných sítí s federativní autentizací založených na protokolu RADIUS a na hledání řešení předložených problémů a jejich ověření na modelu. Přirozeným pokračováním práce, které má následovat, je dotažení návrhu do funkční softwarové implementace, což není jen záležitost problematiky počítačových sítí, ale je třeba zapojit do týmu odborníky na návrh a tvorbu softwarových aplikací. Jednou z možných součástí budoucí implementace je, aby systém mohl propagovat zavedené restrikce uživatelů navenek. K tomu se nabízí propojení s vhodně koncipovanou webovou aplikací.

Mezi přímé problémy práce patří ověření vlivu parametrizace cache databází jednotlivých poskytovatelů služeb (zejména délka trvání záznamu) na úspěchu autentizace uživatele (viz zmínka na str.51). To by zřejmě bylo možné ověřit na modelu. Nejvhodnějším řešením by však bylo převedení dočasných záznamů v cache databázi na standardní záznamy, což ovšem znamená vyřešení

problému autentizace na základě neaktuálního uživatelského účtu (viz odst. 5.4.6.1). Toho je možné dosáhnout v podstatě pouze povinným využíváním systému předávání informací o uživatelích všemi členy federace.

V tomto směru by patrně bylo nemalým přínosem rozšíření systému předávání informací o uživatelích na obecnou distribuovanou správu uživatelů. V praxi to znamená zavedení dalších nových atributů protokolu RADIUS Accounting a rozšíření činnosti bezpečnostního démonu.

Poměrně rozsáhlou a zajímavou problematikou může být záležitost týkající se spolupráce mezi IDS a protokolem RADIUS v rámci multiagentních systémů, jak bylo naznačeno v odstavci 6.4.1.3. Tento problém si zaslouží širší rozbor včetně hlubší analýzy činnosti IDS a jeho jednotlivých částí.

SEZNAM OBRÁZKŮ, GRAFŮ A TABULEK

POUŽITÉ ZDROJE

[1] **Mashkov, V. a Fišer, J.** *Samokontrola a samodiagnostika na systémové úrovni.* Lviv : Ukrainian Academic Press, 2010. 978-966-322-169-4.

[2] **Sova, M.** *Federativní přístup k autentizaci.* Liberec : TU Liberec, 2005.

[3] **Mockapetris, P.** *Domain names - implementation and specification.* - : ISI, November 1987. RFC 1035.

[4] **CESNET, z. s. p. o.** Roamingová politika, verze 2.0 ze 14.7.2009. *eduroam.cz.* [Online] [Citace: 26. únor 2010.] www.eduroam.cz.

[5] **Simpson, W.** *RFC1661 - The Point-to-Point Protocol (PPP).* - : Daydreamer, 1994. RFC1661.

[6] **Lloyd, B. a Simpson, W.** *RFC1334 - PPP Authentication Protocols.* - : L&A and Daydreamer, 1992. RFC1334.

[7] **Simpson, W.** *RFC1994 - PPP Challenge Handshake Authentication Protocol (CHAP).* - : DayDreamer, 1996. RFC1994.

[8] **Aboba, B. a kol.** *RFC3740 - Extensible Authentication Protocol (EAP).* 2004. RFC3740.

[9] **IEEE.** Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. [Online] 2012. 802.11.

[10] **Rigney, C. a kol.** *RFC2865 - Remote Authentication Dial In User Service (RADIUS).* Livingston : The Internet Society, 2000. RFC2865.

[11] **Calhoun, P. a kol.** *RFC3588 - Diameter Base Protocol.* - : The Internet Society, 2003. RFC3588.

[12] **Carrel, D. a Grant, L.** *The TACACS+ Protocol.* - : Cisco Systems, 1997.

[13] **MIT.** MIT Kerberos Consortium. [Online] The MIT Kerberos Consortium, 2007. [Citace: 26. březen 2011.] http://kerberos.org.

[14] **The Institute of Electrical and Electronics Engineers, Inc.** IEEE Std 802.11i™-2004. [Online]

[15] **Rigney, C. a kol.** *RFC2866 - RADIUS Accounting.* Livingston : The Internet Society, 2000. RFC2866.

[16] **Postel, J.** *RFC768 - User Datagram Protocol.* místo neznámé : USC/Information Sciences Institute, 1980. RFC 768.

[17] **Rivest, R.** *RFC1321 - The MD5 Message-Digest Algorithm.* - : MIT Laboratory for Computer Science and RSA Data Security, Inc., 1992. RFC1321.

[18] **Adams, C. a Lloyd, S.** *Understanding PKI: Concepts, Standards, and Deployment Considerations.* Boston : Addison-Wesley Professional, 2003. 978-0-672-32391-1.

[19] **Tanenbaum, A. S. a Van Steen, M.** *Distributed Systems: Principles and Paradigms (2nd Edition).* - : Prentice Hall, 2006. 978-0132392273.

[20] **Motyčková, L.** *Distribuované systémy: výpočty v sítích.* Veletiny : SCIENCE, 1997.

[21] Distribuované systémy. *wiki.matfyz.cz.* [Online] MFF UK. [Citace: 16. březen 2010.] http://wiki.matfyz.cz/wiki/Distribuovan%C3%A9_syst%C3%A9my.

[22] **Open System Consultans Pty., Ltd.** RadSec: a secure, reliable RADIUS Protocol. [Online] [Citace: 11. červen 2009.] http://www.open.com.au/radiator/radsec-whitepaper.pdf.

[23] **Gilbert S., Kowalski D.** *Distributed Agreement with Optimal Communication Complexity.* Austin : Society for Industrial and Applied Mathematics (SIAM), 2010.

[24] **Lamport L., Shostak R., Pease M.** The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems.* Č. 3, 1982, Sv. 4.

[25] **Petri, C. A.** *Kommunikation mit Automaten.* Bonn : University of Bonn, 1962.

[26] **Westergaard M., Verbeek H.M.W.** CPN Tools Homepage. *CPN Tools.* [Online] [Citace: 6. 4 2011.] http://cpntools.org/.

[27] **Češka, M. a kol.** *Petriho sítě.* Brno : VUT v Brně, 2009.

[28] **Hopcroft, E. J., Rajeev, M. a Ullman, J. D.** *Introduction to Automata Theory, Languages, and Computation.* - : Addison-Wesley, 2001. 0-201-44124-1.

[29] **Jensen, K. a Kristensen, L. M.** *Coloured Petri Nets: Modelling and Validation of Concurrent Systems.* Berlin Heidelberg : Springer-Verlag, 2009. 978-3-642-00283-0.

[30] **The FreeRADIUS Server Project.** FreeRADIUS. [Online] [Citace: 11. 6 2010.] http://freeradius.org.

[31] **Wireshark Foundation.** WireShark. [Online] [Citace: 15. 6 2010.] http://wireshark.org/.

[32] **University of Southern California.** *RFC793 - Transmission Control Protocol.* Arlington : DARPA, 1981. RFC793.

[33] **Mitton, D.** RFC2882 - Network Access Servers Requirements: Extended RADIUS Practices. [Online] 2000. [Citace: 23. 8 2011.] http://tools.ietf.org/html/rfc2882. RFC2882.

[34] **Endorf C., Schultz E., MelanderJ.** *Intrusion Detection & Prevention.* Emeryville : McGraw-Hill, 2004. 0-07-222954-3.

[35] **Roesch, M.** Snort. [Online] SourceFire. [Citace: 20. 11 2011.] http://www.snort.org/.

[36] **Perkins, Ch. a Strebe, M.** *Firewally a proxy-servery: Praktický průvodce.* Praha : Computer Press, 2003. 80-722-6983-6.

[37] **Odom, W., Healy, R. a Mehta, N.** *Směrování a přepínání sítí.* Praha : Computer Press, 2009. 978-80-251-2520-5.

[38] **Řehák, M. a kol.** *Multi-Agent Approach to Network Intrusion Detection.* Estoril : -, 2008. ISBN 978-0-9817381-3-0.

[39] **Internet Corporation For Assigned Names and Numbers.** Internet Assigned Numbers Authority (IANA). *Internet Assigned Numbers Authority (IANA).* [Online] [Citace: 11. 10 2012.] http://www.iana.org/.

[40] **Paxton, V. a kol.** *RFC 6298 - Computing TCP's Retransmission Timer.* Berkley : Internet Engineering Task Force, 2011. RFC 6298.

[41] **Jacobson, V. a kol.** *RFC 1323 - TCP Extensions for High Performance.* Berkley : Network Working Group, 1992. RFC1323.

[42] **Kim, I. G. and Choi, J. Y.** Model Checking of RADIUS Protocol in Wireless Networks. *IEICE Trans. Commun.* 2005, Vols. E88-B, No.1, pp. 397-398.

[43] **Orkáč, R.** IDS Snort. [Online] 19. 6 2006. [Citace: 10. 12 2011.] http://www.cs.vsb.cz/grygarek/SPS/projekty0506/Snort.pdf.

[44] **Microsoft.** RADIUS Proxy. [Online] Microsoft, 2012. [Citace: 28. březen 2012.] http://technet.microsoft.com/en-us/library/cc731320.aspx.

[45] **Open System Consultans Pty., Ltd.** *Radiator RADIUS Server: Installation and Reference manual for Radiator version 4.11.* místo neznámé : Open System Consultants Pty. Ltd., 2012.

[46] **Pužmanová, R.** *TCP/IP v kostce.* - : Koop, 2010. 978-80-7232-388-3.

PŘEHLED PUBLIKAČNÍ ČINNOSTI AUTORA

[A] Jelínek J., Satrapa P.: Návrh inovace protokolu RADIUS z hlediska bezpečnosti, In: sborník příspěvků 12. ročníku doktoradské konference, Hradec Králové, 2012, ISBN 978-80-7435-185-3

[B] Jelínek J., Satrapa P.: Návrh inovace protokolu RADIUS z hlediska bezpečnosti, příspěvek na 12. ročníku doktoradské konference, Hradec Králové, 10.5.2012

[C] Jelínek J., Satrapa P., Fišer J.: Simulation of enhanced RADIUS protocol in Colored Petri nets, In: Proceedings of 11th International Scientific Conference Informatics 2011, Rožňava, ISBN 978-80-89284-94-8

[D] Jelínek J., Satrapa P., Fišer J.: Simulation of enhanced RADIUS protocol in Colored Petri nets, příspěvek na konferenci, Informatics 2011, 11th International Scientific Conference, Rožňava, 17.11.2011
[E] Jelínek J., Satrapa P.: Simulation of RADIUS protocol in Colored Petri nets, In: Proceedings of Networking 1 – Theory and Practice, ŽU Žilina, 2011, ISBN 978-80-554-0494-3
[F] Barilla J., Jelínek J., The Mathematical Model of the Chemical Phase of Radiobiological Mechanism, WDS05 Proceeding of Contributed Papers, Part III, 620-624, Praha 2005

Výukové materiály

a. Heller L., Jelínek, J., Simr P.: Práce s počítačem, skriptum, 2011
b. Jelínek, J., Barilla, J., Sýkorová, K.: Základy informatiky s testováním ECDL, skriptum, 2007
c. Jelínek J.: Principy počítačových sítí I, skriptum, 2005
d. Fiala F., Jelínek J.: Sériový port PC v laboratorních úlohách, skriptum, 2003

Účast v řešených projektech

a. Vyrovnávací kurz IT kompetencí pro studenty 1. ročníků PřF UJEP, CZ.1.07/2.2.00/07.0432, projekt ESF OP VK, 2009-2012, celkový objem: 3.564.594,-Kč, **hlavní manažer projektu**
b. Posilování kompetencí VŠ pracovníků pro rozvoj konkurenceschopnosti vysokého školství v Ústeckém kraji, CZ.1.07/2.2.00/07.0117, projekt ESF OP VK, 2009-2012, **spoluřešitel**
c. To je věda, seznamte se - podpora systematické práce s žáky a studenty v oblasti vědy, výzkumu a vývoje, CZ.1.07/2.3.00/09.0121, projekt ESF OP VK, 2009-2012, **spoluřešitel**
d. Tvorba a realizace ŠVP z přír. předmětů v Ústeckém kraji, ESF OP VK, 2007-2008, **spoluřešitel**
e. Plné uplatnění strukturovaných st. programů na PřF UJEP, roz. projekt MŠMT, 2006, **spoluřešitel**
f. Zřízení „Local Cisco Networking Academy“ na UJEP, FR CESNET, 2006, **hlavní řešitel**
g. Příprava a rozvoj strukturovaných st. programů, rozvojový projekt MŠMT, 2005, **spoluřešitel**
h. Příprava SP Informatika pro ZŠ – komb. forma, rozvojový projekt MŠMT, 2004, **hlavní řešitel**
i. Moderní multimediální učebna KI, rozvojový projekt FRVŠ, 2003, **hlavní řešitel**
j. Příprava SP Informatika pro ZŠ – komb. forma, rozvojový projekt MŠMT, 2003, **hlavní řešitel**

Vedení obhájených závěrečných prací

1	HELLER diplomová	Lukáš	Historie internetu	25.3.2004
2	KOLAŘÍK bakalářská	Tomáš	Telekomunikační a síťové technologie	1.5.2004
3	PYTLÁK bakalářská	Michal	Návrh LAN pomocí technologie WiFi	1.5.2004

4	MAŠEK bakalářská	Jan	Složení a funkce osobního počítače	1.5.2004
5	HŘEBEJK bakalářská	Tomáš	Docházkové systémy a jejich aplikace v praxi	10.11.2004
6	NOVOTNÝ bakalářská	Michal	Srovnání bezdrátových technologií a návrh WLAN sítě	21.4.2005
7	BLÁHA bakalářská	Tomáš	Zabezpečení WiFi sítí	7.11.2005
8	DUŠEK bakalářská	Jan	Dynamické směrovací protokoly v IP sítích	20.4.2006
9	GASSER bakalářská	Martin	Obnova ztracených dat	20.4.2006
10	KOPENEC bakalářská	Martin	Přenosová média počítačových sítí	20.4.2006
11	SLAVÍČEK bakalářská	Marek	IP telefonie a zabezpečení kvality služeb (QoS)	20.4.2006
12	CHORVÁTH bakalářská	Petr	MS Windows server 2003: Terminálové služby	9.11.2006
13	PEKLO bakalářská	Petr	Správa sítě s Windows Serverem: Microsoft Exchange Server	1.4.2007
14	HYKŠ bakalářská	Pavel	Správa sítě s Windows Server 2003: Vzdálená správa systému	26.4.2007
15	PRIVARA bakalářská	Petr	Současné procesory Intel a AMD	26.4.2007
16	ŠIC bakalářská	Petr	Správa počítačových sítí na Základních školách	26.4.2007
17	ŠPAČEK bakalářská	Václav	Správa sítě s Windows Serverem: Sdílený datový prostor	26.4.2007
18	ŠVINGR bakalářská	Ivo	Bezpečnost na internetu: Adware & Spyware	26.4.2007
19	JANEČEK bakalářská	Tomáš	Počítačové rozhraní a komunikace s periférierni	24.4.2008
20	PUCHOLT bakalářská	Karel	Grafické karty současnosti	24.4.2008
21	MACHO bakalářská	Jan	Multimediální centrum	23.4.2009
22	MŇUK bakalářská	Lukáš	Hardwarová rozhraní	12.11.2009
23	DURCHÁNEK bakalářská	Martin	Realizace sítě pro Základní školu	29.4.2010
24	RULF bakalářská	Martin	Zabezpečení počítačových sítí s protokolem IEEE 802.1x	29.4.2010

25	VAICOVÁ bakalářská	Zuzana	Bezdrátové technologie WiMAX a jejich využití	29.4.2010
26	KLOFEC bakalářská	Jan	QoS - řízení a vizualizace datových toků	28.4.2011
27	HAMOUZ bakalářská	Ondřej	Směrování mezi autonomními systémy s využitím protokolu BGP	28.4.2011

SEZNAM ZKRATEK

[1] CESNET — organizace pro provozování a rozvoj páteřní akademické počítačové sítě České republiky

[2] RADIUS — Remote Authentication Dial-In User Service, autentizační, autorizační a účtovací protokol

[3] IEEE 802.1X — protokol pro zabezpečení fyzického přístupu do počítačové sítě

[4] NAS — Network Access Server, přístupový bod do sítě (nejčastěji AP WiFi sítě nebo přepínač Ethernetu)

[5] IANA — The Internet Assigned Numbers Authority. http://iana.org

[6] DCD — Data Carrier Detect, kontrolní signál podle normy RS-232

[7] Botnet — skupina softwarových agentů nebo robotů, které slouží jako zdroj útoků na nějakou síť nebo její součást

SEZNAM PŘÍLOH

Softwarová příloha A.1 — Zjednodušený model aktuálního protokolu RADIUS

Softwarová příloha A.2 — Hierarchický model aktuálního protokolu RADIUS

Softwarová příloha B.1 — Model vylepšeného protokolu RADIUS – adjudikátor Simple

Softwarová příloha B.2 — Model vylepšeného protokolu RADIUS – adjudikátor Base

Softwarová příloha B.3 — Model vylepšeného protokolu RADIUS – adjudikátor Vote

Příloha C — Experimentálně naměřené hodnoty