VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ANALÝZA LOGŮ AKTIVNÍCH SÍŤOVÝCH PRVKŮ

LOG ANALYSIS OF ACTIVE NETWORK ELEMENTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. Petra Kajánková

AUTHOR

VEDOUCÍ PRÁCE — doc. Mgr. Karel Slavíček, Ph.D.

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Studentka: Bc. Petra Kajánková ***ID:*** 211550

Ročník: 2 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Analýza logů aktivních síťových prvků

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vyvinout software pro analýzu logů aktivních síťových prvků (přepínače, směrovače, ...) za účelem odhalování bezpečnostních incidentů. Vyvíjený software by měl zejména analyzovat konfigurační zásahy do monitorovaných zařízení a potenciální vliv těchto zásahů na bezpečnost komunikační infrastruktury. Systém bude koncipován jako otevřený a bude konfigurovatelný pro různé výrobce síťového hardware. Součástí řešení je i návrh strukturovaného popisu syslog zpráv souvisejících se změnou konfigurace jednotlivých typů zařízení a návrh klíčových částí konfiguračních souborů vybraných síťových prvků potřebných pro zajištění jejich bezpečnosti.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce.

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 19.5.2023

Vedoucí práce: doc. Mgr. Karel Slavíček, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Diplomová práce se zaměřuje na problematiku vyhledávání škodlivé činnosti pocházející od autorizovaných uživatelů, kteří mají oprávnění modifikovat zařízení. Teoretická část nejprve představuje základní síťový model ISO/OSI, společně s TCP/IP. Poté se zaměřuje na dva nejznámější protokoly pro práci s logy Syslog protokol a SNMP. Praktickou část práce tvoří teoretický návrh a následná realizace programu, jenž je schopen vyhodnotit záznamy o nastalých událostech na jednotlivých síťových zařízeních.

KLÍČOVÁ SLOVA

ISO/OSI, TCP/IP, Směrovače, Syslog, SNMP, Logování, MikroTik, Cisco System, Konfigurace, Zabezpečení

ABSTRACT

The master thesis focuses on the issue of searching for malicious activity originating from authorized users who have permission to modify devices. The theoretical part first introduces the basic network model ISO/OSI, together with TCP/IP. It then focuses on the two most well-known logging protocols, the Syslog protocol and SNMP. The practical part of the work consists of the theoretical design and subsequent implementation of the program that is able to evaluate records of events occuring on individual network devices.

KEYWORDS

ISO/OSI, TCP/IP, Routers, Syslog, SNMP, Logging, MikroTik, Cisco System, Configuration, Security

KAJÁNKOVÁ, Petra. *Analýza logů aktivních síťových prvků*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2030, 79 s. Diplomová práce. Vedoucí práce: doc. Mgr. Karel Slavíček, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Petra Kajánková
VUT ID autora:	211550
Typ práce:	Diplomová práce
Akademický rok:	2022/23
Téma závěrečné práce:	Analýza logů aktivních síťových prvků

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Zde bych ráda poděkovala vedoucímu diplomové práce panu doc. Mgr. Karlu Slavíčkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod **12**

1 Popis referenčních modelů **13**
1.1 Model ISO/OSI . . . 13
1.1.1 Fyzická vrstva . . . 13
1.1.2 Spojová vrstva . . . 15
1.1.3 Síťová vrstva . . . 18
1.1.4 Transportní vrstva . . . 26
1.1.5 Relační, prezentační a aplikační vrstva . . . 28
1.2 Model TCP/IP . . . 29

2 Řízení aktivních síťových prvků **30**
2.1 Logování událostí . . . 30
2.2 Protokol Syslog . . . 31
2.3 SNMP . . . 33
2.4 Porovnání protokolů SNMP a Syslog . . . 35

3 Praktická část **36**
3.1 Základní konfigurace síťových zařízení . . . 36
3.1.1 Nastavení zařízení od firmy Cisco . . . 36
3.1.2 Nastavení zařízení od firmy MikroTik . . . 38
3.2 Realizovatelné útoky . . . 40
3.3 Inicializace zařízení a vytvoření profilu uživatele . . . 44
3.4 Rozšíření programu pro další výrobce . . . 45

4 Popis přiloženého programu **47**
4.1 První fáze programu . . . 47
4.1.1 Podrobný popis zpracování souboru Architektura sítě . . . 49
4.1.2 Podrobný popis zpracování souboru Definice zařízení . . . 52
4.1.3 Načtení souboru Rozdělení logů pro další výrobce . . . 53
4.2 Druhá fáze . . . 54
4.2.1 Zpracování logů od firmy MikroTik . . . 56
4.2.2 Zpracování logů od firmy Cisco . . . 66
4.2.3 Zpracování logů dalších výrobců . . . 67
4.3 Další funkcionality programu . . . 68
4.4 Realizace testovací sítě . . . 69
4.4.1 Nastavení Syslog serveru . . . 69
4.4.2 Výsledky programu na vytvořené testovací síti . . . 72

Závěr **75**

Literatura **76**

A Popis přiloženého softwaru **79**

Seznam obrázků

1.1 Ukázka komunikace modelu ISO/OSI dvou fyzicky propojených zařízení. 14
3.1 Ukázka logu zařízení firmy Cisco. 37
3.2 Ukázka logu v zařízení MikroTik. 39
3.3 Příklad topologie sítě. 43
4.1 Obecný popis navrhovaného programu. 47
4.2 Obecné kroky vykonané při spuštění programu. 48
4.3 Ukázka zjednodušeného postupu vyhledání logů. 55
4.4 Zjednodušený příklad zpracování logu. 56
4.5 Příklad vygenerovaného logu pocházejícího ze síťového zařízení od firmy MikroTik. 57
4.6 Příklad logu ze síťového zařízení firmy Cisco. 66
4.7 Ukázka MikroTik logu popisující přidání nového skriptu. 72
4.8 Ukázka Cisco logu popisující přenastavení času. 72
4.9 Ukázka HP logu popisující změnu rozhraní. 73

Seznam tabulek

3.1 Příklad tabulky pro zadání cest vycházející z výše zobrazené topologie. 44
3.2 Příklad tabulky pro rozdělení informací v logu pro obr. 3.1 Ukázka logu zařízení firmy Cisco. 46
4.1 Příklad vyplněného souboru: Architektura sítě. 50
4.2 Příklad vytvoření duplicity záznamů v souboru Architektura sítě. . . 51
4.3 Chybový příklad vyplněného souboru: Definice zařízení. 53
4.4 Příklad vyplněného souboru: Předpis logů. 61
4.5 Příklad vyplněného souboru: Architektura sítě. 68
4.6 Příklad vyplněného souboru: Hodiny uživatelů. 68
4.7 Ukázka vyplněného souboru: Rozdělení logu v případě zařízení HP. . 73
4.8 Ukázka vyplněného souboru: Předpis logu v případě zařízení HP. . . . 73

Seznam výpisů

4.1 Ukázkový výpis při kontrole kompletní architektury s nastaveným přepínačem. 52
4.2 Ukázkový výpis terminálu při informování o uložení zálohy. 64
4.3 Ukázkový výpis v interaktivním módu programu. 65
4.4 Ukázkový výpis v případě neshody časového údaje. 69
4.5 Ukázkový výpis běhu nad realnými logy firmy MikroTik. 73
4.6 Ukázkový výpis běhu nad realnými logy firmy HP. 74
4.7 Ukázkový výpis běhu nad realnými logy firmy Cisco. 74

Úvod

V dnešní době je sběr a zpracování informací, společně s jejich následnou výměnou, součástí každé infrastruktury. Aby data mohla být doručována jednotlivým stanicím, nebo naopak směrována v síti, je třeba využít síťových zařízení – směrovačů a přepínačů. Vzhledem k tomu, že se jedná o nezbytné, a tedy i kritické prvky sítě, je zároveň třeba řešit i jejich bezpečnost.

Cílem diplomové práce je především zaměření se na možné útoky přicházející od autorizovaných uživatelů, kteří mají oprávnění k modifikaci nastavení a manipulaci s fyzickými zařízeními/spoji. Odhalení takovéhoto chování je postaveno na odesílání jednotlivých logů – záznamů o událostech, které na zařízení proběhly, do centrálního úložiště. Nezbytnou součástí je i následné vyhodnocení, zda je chování síťového prvku v souladu s nastavenými bezpečnostními pravidly.

Teoretická část práce nejprve popisuje strukturu sítě z pohledu referenčního modelu ISO/OSI se zaměřením na popis funkcionalit jednotlivých vrstev, který následně porovnává s druhým, praktičtějším modelem TCP/IP. Poslední teoretická část představuje dva pravděpodobně nejvyužívanější protokoly pro odesílání vytvořených logů – Syslog, SNMP. V neposlední řadě zhodnocuje výhody, nevýhody a vhodnost implementace na jednotlivé infrastruktury.

Výstup praktické části je zaměřen na návrh a realizaci aplikace sloužící k vyhledávání potenciálně nebezpečných událostí, zejména změn konfigurace aktivních síťových prvků. Tato část popisuje nejprve návrh konfigurace jednotlivých zařízení, aby byla schopna události nejen logovat, ale následně i odesílat na požadovaný, cílový server. V první fázi jsou definovány základní konfigurační příkazy pro dva vybrané výrobce síťových zařízení Cisco Systems a MikroTikls. Kapitola dále obsahuje příkazy umožňující logování příslušných sledovaných událostí. Program však ve výsledku nemá omezení pouze na logy uvedených výrobců a umožňuje vyhodnocení logu libovolného výrobce na trhu.

Druhou částí práce je následný návrh bezpečnostních profilů jednotlivých autorizovaných, kritických uživatelů a pravidel definujících nenormální chování administrátorů zařízení. Návrh aplikace tak je schopen záznamy o událostech nejen zpracovat, ale následně i posoudit, zda se jedná o legitimní chování.

Poslední kapitola popisuje samotnou praktickou realizaci, ve které je detailně popsán přiložený program v jednotlivých fázích svého běhu. Součástí této kapitoly je také návrh konfigurace Syslog serveru a následné otestování vytvořené aplikace na testovací síti.

1 Popis referenčních modelů

Práce se především zaměřuje na problematiku bezpečnosti síťových zařízení a jejich ochranu před škodlivou činností pocházející od autorizovaných uživatelů. Konkrétněji se jedná o zaznamenávání neobvyklých, ale i běžných událostí, které na těchto zařízeních vznikají. Tyto události jsou dále zpracovávány a vyhodnocovány. Teoretická část se tak nejprve zaměří na všeobecný popis dvou základních referenčních modelů: ISO/OSI[1], TCP/IP[2]. Tyto modely popisují nejen důvod využití síťových zařízení zmíněných výše, na kterých je diplomová práce postavena, ale dále i popisují jednotlivé procesy a protokoly, které jsou potřebné k funkčnosti samotné sítě.

1.1 Model ISO/OSI

Model ISO/OSI je prvním představeným modelem, jehož vznik je připsán organizaci International Organization for Standardization. Cílem této normy bylo v osmdesátých letech určitým způsobem standardizovat rychle se rozvíjející počítačové sítě z důvodu možné komunikace mezi nimi. Organizace se tak zaměřila především na vydání určitých pravidel, díky kterým spolu mohly dříve menší, uzavřené a nekompatibilní sítě začít komunikovat napříč celým světem. Norma byla oficiálně přijata v roce 1984 s označením ISO 7498. Zmíněný referenční model je však v dnešní době spíše teoretickým popisem jeho nástupce TCP/IP[3].

Jedná se o otevřený standard. V normě se nenachází žádné konkrétní specifikace protokolů a ani jejich technická řešení. Jsou zde popsány pouze důležité aspekty komunikace mezi dvěma uživateli. Průběh zpracování dat je rozdělen celkem do sedmi vrstev, jak zobrazuje obrázek níže.[1]

1.1.1 Fyzická vrstva

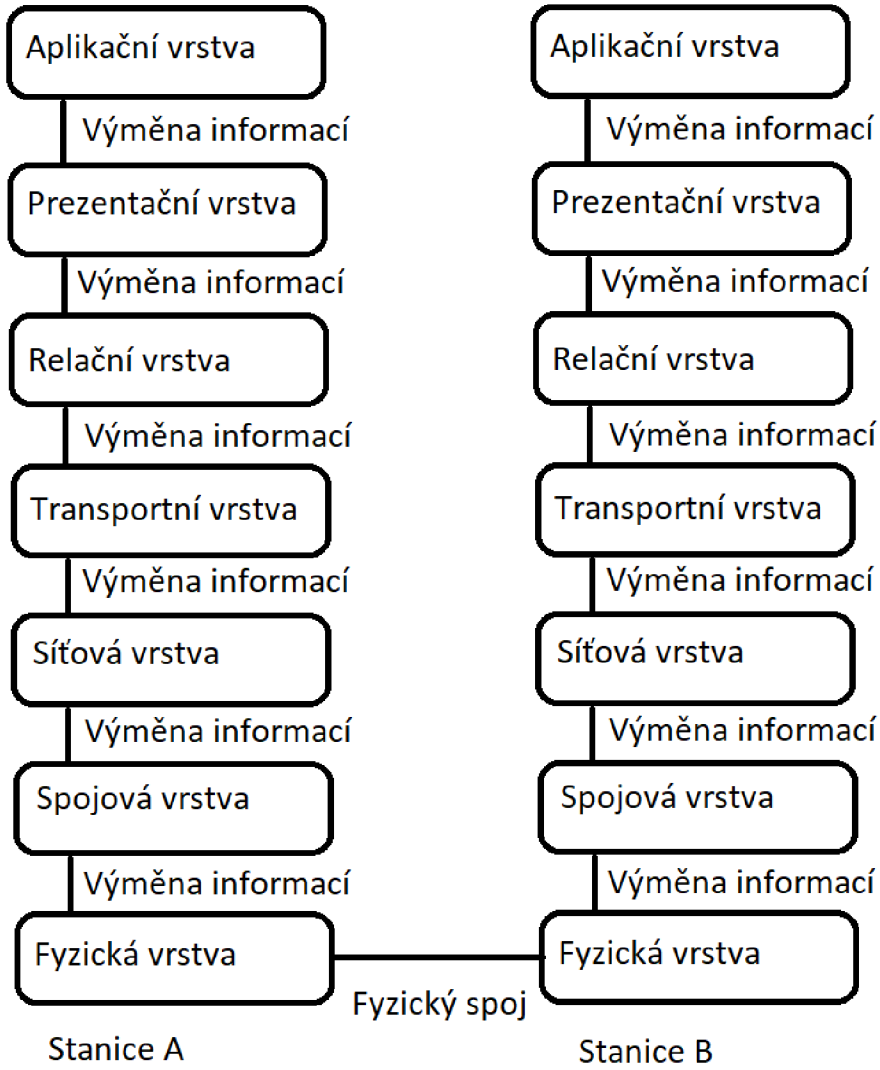
Fyzická vrstva je nejnižší vrstvou zmíněného modelu. V případě, že uživatel A navazuje komunikaci s uživatelem B, je poslední vrstvou, kterou data prochází od uživatele A, a zároveň první průchozí vrstvou při zpracování u uživatele B. Ze míněného vyplývá, že samotná vrstva pracuje s bitovou posloupností, tedy s převedením jednotlivých bitů do signálu vhodného pro daný přenos. Podmínkou funkčnosti celého modelu je i následné rozpoznání druhou stranou přijímaného signálu a schopnost

[1] Jedná se o zkratku složenou z: ISO – International Organization for Standardization, OSI – Open System Interconnection.

[2] TCP – Transmision Control Procol, IP – Internet Protocol.

[3] Je popsán v kapitole 1.2 Model TCP/IP.

reverzního procesu – převedení signálu na bitovou posloupnost. Dále vrstva zajišťuje modulaci, kódování, a především přístup k přenosovému médiu, synchronizaci, rozložení signálu, jež jsou popsány detailně dále.[2]



Obr. 1.1: Ukázka komunikace modelu ISO/OSI dvou fyzicky propojených zařízení.

V dnešní době lze přenosová média rozdělit do tří hlavních skupin: volný prostor, elektrické vodiče a optická vlákna. Pravděpodobně nejvyužívanějším médiem je aktuálně volný prostor založený na elektromagnetických vlnách. Jeho výhody spočívají zvláště v jednoduchosti. Z těchto důvodů je využíván pro televizní, rádiové vysílání, mobilní telefony, ovládání různých spotřebičů. Mezi nevýhody pak především patří náchylnost k rušení nebo omezená vzdálenost (dosah).

Elektrické vodiče oproti tomu využívají elektrické signály, které přenášejí přes

koaxiální/symetrický kabel[4]. Ze zmíněného vyplývá jeho hlavní nevýhoda – jestliže spolu chtějí komunikovat dva komunikační body (například stanice a senzor), musí být spolu propojeny fyzickým kabelem. Tyto kabely ve verzi se stíněním mají pak nevýhodu složitější implementace kvůli své velikosti (tloušťce). Stínění u vodičů pomáhá signál chránit před vnějším elektromagnetickým rušením – ovlivněním průchozího signálu jiným nežádoucím signálem. Metoda je tak vhodná především pro kratší přenosy.

Posledním médiem jsou optické sítě. Jedná se o nejrychlejší a pravděpodobně nejbezpečnější sítě. Signál je zde šířen pomocí odrazu světelných impulzů. V těchto sítích prakticky neexistuje šumové rušení. Jejich hlavní výhoda spočívá nejen v rychlosti, vzdálenosti – na vedení stovek km dosahují malý útlum[5], ale zároveň i bezpečnosti - odchytí-li úročník data, způsobí tím navýšení chybových stavů a tímto může být odhalen útok odposlechem. Z těchto důvodů jsou implementovány do páteřních tras.

Aby mohla být bitová sekvence vyjádřena do signálu a následně přenášena zmíněnými médii, musí být využito procesu kódování/modulace. Kódování, neboli přenos signálu v základním pásmu, je charakteristický využitím pravoúhlých impulzů. Tyto impulzy jsou získávány pomocí převodu bitové sekvence do libovolné varianty linkového kódu[6]. Dochází zde tak ke skokovým změnám hodnot.[3]

Oproti tomu modulace využívá nosnou frekvenci[7], která je následně spojitě v čase skládána s modulačním signálem[8]. Výsledný modulovaný signál má poté ovlivněnou vždy pouze jednu veličinu z důvodu samotného procesu modulace (spojení). Ovlivněnými veličinami mohou být:

- amplituda,
- frekvence,
- fáze.

1.1.2 Spojová vrstva

Druhou vrstvou modelu ISO/OSI je spojová (linková) vrstva. Ačkoliv ve zmíněném modelu je zakreslena jako jedna vrstva, z praktického hlediska se rozděluje na dvě podvrstvy – Logical Link Control[9], Media Access Control[10]. Tyto podvrstvy jsou

[4] V dnešní době jsou koaxiální kabely brány spíše jako zastaralá metoda.

[5] Útlum lze charakterizovat jako pokles amplitudy s rostoucí vzdáleností.

[6] Jedná se o vyjádření v rozsahu původního frekvenčního spektra.

[7] Jedná se o frekvenci typicky s vyšším kmitočtem, než je kmitočet modulačního signálu.

[8] Jedná se o datový signál.

[9] Označuje se zkratkou LLC.

[10] Označuje se zkratkou MAC.

spolu úzce spjaty. LLC, v překladu podvrstva Řízení logického spoje, se stará především o:

- vytvoření rozhraní mezi přenosovým médiem a vyšší vrstvou[11],
- multiplexování požadavků,
- řízení chybových stavů.

Potřeba multiplexace požadavků vychází z předpokladu, že uživatel zadá více procesů najednou. Například odešle-li email a zároveň i soubory, bude zde potřeba využít principu multiplexace – spojení více požadavků za účelem šetření přenosové kapacity.[2]

MAC podvrstva, neboli Řízení přístupu k přenosovému médiu, zajišťuje kódování, adresování, práci s rámci a řeší problematiku sdíleného média a jeho přístupu. Adresace zde probíhá na základě logických adres – jedná se o jedinečné označení každého zařízení. Její délka je 48 bitů. Dolních 24 bitů je dáno výrobcem a horních 24 bitů pak kódem síťové karty. Teoreticky by tak každé zařízení připojené k internetu mělo mít vlastní a jedinečnou logickou adresu. Tyto adresy jde však ručně přenastavit, či snadno podvrhnout, proto teoreticky v sítích mohou dočasné duplicity existovat (například v rámci útoku).[4]

Spojová vrstva pracuje s rámci. Rámcem jsou označovány pakety od síťové vrstvy, které tato vrstva zaobalí svým záhlavím složeným z:

- preambule – označuje začátek rámce,
- cílové a zdrojové adresy – informace, které zařízení data odeslalo a jaké zařízení je má přijmout,
- datové části – tato část je přebrána ze síťové vrstvy[12],
- kontroly chyb – probíhá zde jednoduchý výpočet pro odhalení chyb při přenosu. Nelze zde rozpoznat kybernetický útok – útočník je schopen tento součet nahradit novou hodnotou. Jestliže výpočet odhalí chybu, rámec může být zahozen, nebo opraven za přítomnosti opravných protichybových kódů,
- závěrečné sekvence – označení konce zápatí.

Adresace (doručení) rámců následně probíhá na základě logických adres s využitím přepínačů. Přepínač je fyzické zařízení, jehož úkolem je překlad IP adresy[13] na MAC adresu – mapování a zároveň i překlad MAC adres na IP adresy – reverzní mapování. Reverzní mapování je využito při odesílání dat mimo síť. Překlady adres mohou být:

- dynamického charakteru – zařízení dostalo náhodnou IP adresu. Tuto službu zajišťuje DHCP, jehož úkolem je na základě požadavku přidělit zařízení ná-

[11] V modelu ISO/OSI vždy platí, že vrstva pracuje pouze s daty sousedních vrstev.

[12] Typicky je tato část omezena. U Ethernetu je to například 1500 B, u ATM pak 48B.

[13] Tímto pojmem se zabývá kapitola 1.1.3 Síťová vrstva.

hodnou IP adresu z daného rozsahu[14],

- statického charakteru – mezi IP adresou a logickou adresou existuje určitý výpočet.

Dynamický překlad využívá ARP protokol – celým názvem Address Resolution Protocol. Přijde-li rámec na přepínač, zařízení nejprve vyhledává překlad ve své tabulce překladů, zda již nezná odpověď. Jestliže tomu tak není, odešle broadcastem[15] dotaz obsahující: zdrojovou IP adresu, cílovou IP adresu, zdrojovou fyzickou adresu, pole pro vyplnění cílové fyzické adresy. Tento dotaz získají všechny stanice, které jsou připojeny k danému směrovači. Odpověď odesílá unicastem[16] pouze ta stanice, která má hledanou IP adresu. Přepínač následně uloží překlad do své paměti a odešle data danému zařízení. ARP je schopen dynamicky reagovat na změny v síti jako jsou změny IP adres, odpojení zařízení.[5]

Poté, co je rámec doručen dané stanici, je nejprve vypočítána kontrolní sekvence. Následně je zkontrolována, zda se shoduje s uvedenou sekvencí v záhlaví. Jestliže jsou výpočty odlišné, zařízení využije jednu z uvedených technik řešící opakovaný přenos. Jedná se o:

- Stop and Wait ARQ[17],
- Selective Repeat ARQ,
- Go Back N ARQ.

Metoda Stop and Wait, je ze všech nejméně efektivní, probíhá-li přenos většího objemu dat, kdy nevadí určitá ztráta informací, z důvodu čekání na potvrzení doručení od druhé strany. Jestliže stanice B zašle rámec stanici A, stanice B vyčkává na kladné potvrzení bezchybného přenosu od stanice A. V druhém případě stanice B přenos opakuje. Tento postup aplikuje i v případě, nepřijde-li stanici B žádná odpověď od protější strany – například z důvodu zatížení sítě. Dochází tak k neefektivnímu přenosu za cenu nulové ztrátovosti. Tato metoda se v dnešních sítích příliš často nepoužívá a je nahrazena efektivnějšími metodami. Alternativou je využití klouzavého okna. Oproti zmíněné metodě odešle stanice B domluvený počet paketů a poté vyčkává na potvrzení o jejich doručení. I přestože se v tomto případě odešle více paketů, stále zde dochází k určitým prodlevám při čekání na potvrzení. Tato metoda je především využita v sítích, kde jsou zasílány informace v menší míře.

Metoda Go Back N nevyužívá potvrzování jednotlivých rámců, ale je založena na principu klouzavého okna. Tímto krokem je urychlena celá komunikace. Přijde-li stanici A rámec, jehož kontrolní výpočty budou odlišné, zašle protější stanici

[14]Detailnější popis je v kapitole 1.1.3 Síťová vrstva.

[15]Označuje druh provozu, kdy jsou data zaslána všem stanicím v dané síti.

[16]Označuje druh provozu, kdy dochází ke komunikaci klient – klient. Zpráva od stanice A bude zaslána pouze stanici B.

[17]ARQ označuje zkratku Automatic Repeat Request, ve volném překlad automatický požadavek na opakování přenosu.

číslo paketu. Následně vyčkává, dokud druhá strana neopakuje přenos vybraného paketu. Přijetí, zpracování zaslaného požadavku druhou stranou trvá určitý čas, a tak stanice B mezitím vysílá další pakety stanici A. Ta je však zahazuje. Jakmile stanice B zpracuje požadavek na opakovaný přenos, začne pakety odesílat znova od pořadového čísla uvedeného v požadavku. Ačkoliv v uvedeném příkladu je tato metoda efektivnější než předchozí, stále zde dochází k určitým prodlevám kvůli zahazování a nucenému opakování přenosu.

Poslední a zároveň nejvíce technicky náročnou metodou je Selective Repeat. Metoda využívá opakované odeslání pouze vybraných (chybových) paketů. V případě, že stanice A přijme chybový paket, zašle stanici B žádost o opakovaný přenos pouze vybraného paketu. V mezičase, na rozdíl od předchozí metody, jsou příchozí pakety ukládány do vyrovnávací paměti. Jakmile stanice B přijme požadavek na opakovaný přenos, zařadí paket znova do fronty k odeslání. Frontu k odeslání lze v tomto kontextu chápat jako pořadí určující postupné odesílání jednotlivých paketů. Druhá strana tak musí mít vyšší požadavky na velikost paměti, do které data v mezičase ukládá. Poté, co je straně A zaslán očekávaný paket, musí následně přeskládat přijaté pakety do správného pořadí a až v této chvíli je může začít zpracovávat.[2]

Zmíněnými kroky je zajištěno doručení rámců v lokální síti na spojové vrstvě, která následně přijatý rámec předává síťové vrstvě.

1.1.3 Síťová vrstva

Síťová vrstva je klíčová především v případech, vznikne-li požadavek uživatele na komunikaci/službu mimo síť, ve které se právě nachází. V tomto okamžiku jsou pakety[18] od uživatele směrovány na směrovač. Ten následně rozhodne o dalším směrování. Existují dva způsoby rozhodování dle typu sítě:

- Sítě se spojením – tento způsob směrování není příliš častý z důvodu jeho náročnosti a potřeby rezervace přenosové cesty. Není ani schopen dynamicky reagovat na změny v síti, jako jsou například výpadky uzlů, či přetížení zařízení. Nejdříve dojde k již zmíněné rezervaci přenosových prostředků a následně po vytvoření komunikačního okruhu jsou data odeslána. K cílové stanici tato data dochází v odeslaném pořadí a není nutno pořadí paketů přeskládávat. Pakety nemusí být opatřeny cílovými adresami, ale jsou identifikovány na základě identifikátoru daného okruhu.
- Sítě bez spojení – tato metoda je využívána téměř ve všech datových sítích dnešní doby. Sítě jsou schopny dynamicky reagovat na změny a nečekané události. Směrování zde probíhá na základě IP adres společně se směrovými tabulkami. Na základě výměny informací mezi směrovači tak nastávají situace, kdy

[18] Jedná se o označení datových souborů na této vrstvě.

pakety dochází druhé straně v přeházeném pořadí. Nedochází zde k rezervaci prostředků.

Samotné směrování probíhá na základě využití IP protokolů – v celém znění Internet Protocol. Tento protokol v dnešní době existuje ve dvou variantách – IPv4, IPv6. Jednotlivé verze nejsou mezi sebou kompatibilní, a tak je potřeba využít různých technik, které řeší jejich koexistenci.[2]

IPv4

Ze statistik společnosti Google ze září roku 2002 vyplynulo, že téměř 61 procent uživatelů této společnosti využívá stále adresy z rozsahu IPv4. Ačkoliv je tak v dnešní době rozsah protokolu IPv4 zcela vyčerpán, stále je rozšířenějším protokolem, než jeho nástupce IPv6.

IPv4 využívá ke směrování 32 bitovou adresu zařízení a 32 bitovou adresu sítě. Teoreticky tak mohou existovat čtyři miliardy těchto adres. Adresy jsou pak rozdělovány do dvou základních skupin, dle možnosti komunikace na:

- Veřejné IP adresy – Tyto adresy lze popsat jako globální identifikátor uzlu napříč celým internetem. Jsou směrovatelné, a tedy i viditelné pro celý internet. Typicky se jedná o adresy NATu, veřejných směrovačů.
- Privátní IP adresy - Nejedná se vždy o jedinečné adresy jako u spojové vrstvy. Na základě podsíťování[19] mohou vznikat duplicity adres v rámci celého internetu. Tyto duplicity nezapříčiní doručení paketů nesprávnému uzlu z důvodu základní podmínky – v rámci sítě musí mít každé zařízení jedinečnou IP adresu. Navíc tyto adresy nejsou směrovatelné z vnější sítě. Jestliže zařízení z privátní sítě komunikuje se serverem, typicky zde dochází k překladu IP adres – tedy náhradě privátní adresy za veřejnou.

IPv4 rozděluje rozsahy celkem do pěti tříd dle velikosti adresního prostoru. Postupem času se tato metoda dělení ukázala jako ne příliš efektivní. Vznikl zde problém nedostatku malých sítí a přebytku velkých sítí. Docházelo zde k rezervaci určitého rozsahu adres pro multicastové přenosy a experimentální účely. Jedná se o rozsahy 224 až 255. Tím byl zmenšen adresní prostor pro klienty. Na základě toho tak vzniklo podsíťování, neboli Subnetting. Jedná se o rozdělování jedné větší sítě do menších podsítí. Reverzním procesem je pak Supernetting - spojování menších sítí do větší.

Velikost hlavičky protokolu se pohybuje v rozmezí 20 až 60 bajty. Celková velikost paketu teoreticky může být až 65535 bajtů[20]. Jedná se o teoretickou velikost, protože protokol IPv4 dovoluje funkci fragmentace. Fragmentací je označován proces, dojde-li k situaci, kdy paket prochází sítí, jejíž maximální velikost je menší než

[19] Popsáno níže.

[20] V celkovém součtu je započítána i velikost hlavičky.

velikost celého paketu, a tak musí hraniční směrovač přijatý paket nejprve rozdělit na odpovídající jednotky. Ty následně zaobalí novou IP hlavičkou a až poté mohou být pakety vyslány do sítě.

Hlavička IPv4 je složena z:

- Verze IP protokolu – aktuálně jsou tak rozlišovány protokoly IPv4 a IPv6.
- Typu služby – identifikuje typ služby pro protokol QoS – Quality of Service. Jedná se o protokol zajišťující určitou kvalitu služeb. Dojde-li k zatížení směrovače, vznikne zde krátkodobý požadavek na zahazování určitého množství paketů. Ty mohou být vybrány dle třídy kvality. Platí zde pravidlo: má-li paket vyšší třídu kvality, existuje menší šance, že bude po cestě zahozen.
- Celkové délka IP datagramu – z důvodu proměnné délky paketů je potřeba rozpoznat jejich začátek a konec.
- Identifikace IP datagramu – proběhne-li proces fragmentování, je potřeba u příjemce jednoduchým způsobem rozpoznat související pakety. Tento identifikátor je využit k označení dané fragmentace.
- Příznaků – typicky se zde nastavuje bit související s fragmentací; je-li nulový, zařízení nemusí čekat na další části paketu. V opačném případě zařízení vyčkává na všechny části, následně je seskládán původní paket. Jestliže cílovému zařízení nedojdou všechny části paketu (část fragmentu byla zahozena, poškozena), jsou všechny části daného paketu zahozeny. Pakety nemusí vždy putovat stejnou cestou, a tak nemusí vždy nastat proces fragmentace.
- Posunutí fragmentu od začátku – v případě pozitivního nastavení příznaku s fragmentací je pole využito k označení pozice posunutí od začátku paketu.
- TTL[21] – označuje životnost paketu. Každým dalším skokem je životnost paketu snížena o jedna. Jestliže je pole nulové, směrovač paket zahodí a uživatel bude informován ICMP protokolem o jeho zahození[22].
- Volitelných položek záhlaví – jedná se především o informace pro směrovače. Pole dovoluje nastavit:
  - Požadavek na zaznamenávání cesty, kudy paket procházel – ve výpise budou zaznamenány pouze venkovní směrovače/servery s veřejnou IP adresou, privátní adresy mimo síť odesílatele se nezaznamenávají z důvodu bezpečnosti. Příkladem trasování paketu je například příkaz traceroute.
  - Požadavek na zaznamenání času – slouží ke zjištění zpoždění jednotlivých paketů, které je definováno jako čas potřebný od odeslání paketu až po jeho přijetí, zpracování.
  - Požadavek na cestu směrování – v poli lze explicitně nastavit úplnou,

[21] Celým názvem Time To Live.

[22] Jedná se o protokol Internet Control Message Protocol. Jeho detailnější popis se nachází v kapitole IPv6.

nebo částečnou cestu (IP adresy), kterou by měl paket putovat.

- Kontrolního součtu – při přenosu napříč sítěmi může dojít k rušení – poškození paketů a následné změně dat. Z podobných důvodů jako u spojové vrstvy je vypočítán kontrolní součet a rozhodnuto, zda paket bude předán vyšší vrstvě.

IPv4 byl prvním standardem, kterému se podařilo fakticky propojit internet a umožnit směrování napříč celým světem. Z dnešní doby se však jeho vlastnosti (proměnná velikost IP hlavičky, fragmentace) nejeví zcela efektivní, a proto vznikl v roce 1994 nový standard RFC 2460 s implementaci IPv6.[5]

IPv6

Protokol IPv6 vychází ze základů IPv4. Využívá hexadecimální zápis, společně se 128 bitovou adresou. Jedná se o tak veliký adresní prostor[23], který by se teoreticky neměl nikdy vyčerpat. Na jednoho uživatele vychází v přepočtu tisíce adres. IPv6 upouští od procesu klasické fragmentace z důvodu zpomalování přenosu dat[24] a zároveň zjednodušuje hlavičku IPv6 datagramu. Těmito kroky zmenšuje čas potřebný pro doručení dat.

Hlavička protokolu byla oproti protokolu IPv4 snížena na konstantních 40 bajtů. Zároveň byly vynechány položky týkající se velikosti záhlaví, kontrolních součtů a volitelných položek záhlaví. IPv6 hlavička obsahuje:

- Verzi IP protokolu – jedná se o zásadní informaci o verzi IP protokolu z důvodu nekompatibility IPv4 a IPv6. Jestliže by byl paket odeslán ze sítě s IPv6 adresami do sítě s IPv4 adresami, nebylo by možné provést krok doručení. K řešení této problematiky jsou využívány následující mechanismy:
  - Paket je zaobalen hraničními směrovači o další záhlaví, nachází-li se za směrovačem jiná verze zmíněného protokolu.
  - Využití zařízení NAT[25] k překladu adres. Cílem je pak vytvořit záznam v zařízení NAT o záměně IPv4 (IPv6) adresy za adresu opačné verze IPv6 (IPv4). Tímto krokem nemusí být přidáno další záhlaví. Při překladu ale dochází k určitému zpomalení.
- Identifikátor toku – jeho účelem je identifikace dat patřících k sobě dle zdrojové/cílové adresy. Toto pole není všemi směrovači podporováno, a tak může být ignorováno.
- Třídu provozu – definující prioritu paketů pro možné zahazování při zatížení. Tyto priority mohou být po cestě dalšími směrovači měněny. Jedná se pouze o informaci pro směrovač, na kterou nemusí brát při zahazování ohled.

[23] Adresní prostor lze vypočítat umocněním dvou na 128.

[24] Protokol IPv6 povoluje fragmentaci pouze za určitých podmínek. Detailně popsáno níže.

[25] Celým názvem Network Address Translation.

- Další záhlaví – slouží k identifikaci typu dalšího záhlaví. Může se jednat o identifikace TCP, UDP[26], nebo vnořeného záhlaví IPv6. Vnořené záhlaví je využito v případech, je-li potřeba využít procesu fragmentace. Tento proces může oproti IPv4 vykonat pouze zdrojový uzel. Žádné síťové zařízení po cestě v sítích IPv6 nemůže aplikovat proces fragmentace a je-li tedy velikost paketu větší než maximální průchozí velikost síti, je paket zahozen.[6]

Srovnání IPv4 a IPv6

Protokol IPv6 je úzce spjat s protokolem ICMP[27]. Jedná se o servisní protokol – nepřenáší žádná uživatelská data. Informuje pouze stanice o chybových (mimořádných) stavech, mezi které patři například: vypršení Time To Live (TTL), nedoručení paketu z důvodu neexistence cesty, zatížení sítě, informace o snížení rychlosti přenosu, zahození z důvodu velikosti paketu, nebo chybových údajů. Protokol ICMP je dále využíván k ověření dostupnosti stanic, či získání časových razítek. Tyto služby poskytuje ve verzi ICMPv4 a ICMPv6. Ve vyšší verzi je využíván k náhradě ARP protokolu – překladu IPv6 adresy na MAC adresu. Princip překladu pak zůstává podobný.

IPv6 v původním návrhu byl úzce spjat s protokolem IPsec[28]. Cílem bylo navýšit bezpečnost při přenosu pomocí šifrování na síťové vrstvě. Procesem šifrování je utajena původní informace, ke které by neměl mít přístup nikdo jiný než odesílatel a příjemce s využitím tajného klíče/tajemství. Většina známých protokolů pro zabezpečení přenosu zabezpečuje přenos dat za pomocí procesu šifrování až na vyšších vrstvách. Od návrhu však bylo časem opuštěno z důvodu složitosti pro zařízení na síti a zpomalení.

Nová verze IP protokol zavádí a řeší většinu nedostatků IPv4. Z těchto důvodů vzniká tlak od světových organizací, aby došlo k úplnému nahrazení IPv4 adres za IPv6 adresy. Aktuálně koexistují sítě obou druhů a je třeba využívat mechanismy, které jsou schopny zajistit jejich součinnost.

Přidělování IP adres

Aby v síti mohlo fungovat směrování popsané výše, musí platit zmíněná podmínka, že v dané síti má každé zařízení svou jedinečnou IP adresu. V případě IPv4, je za tímto účelem využit aplikační protokol DHCP – Dynamic Host Configuration Protocol.

[26] Detailně popsáno v kapitole 1.1.4 Transportní vrstva.
[27] Celým názvem Internet Control Message Protocol.
[28] Celým názvem IP Security.

Služba funguje na principu komunikace klient/server. Stanice pro komunikaci mimo svou síť požádá server o přidělení IP adresy pomocí zprávy DHCP Discover. Žádost je zaslána broadcastem z důvodu, že stanice netuší, kde se DHCP nachází. Existují případy, že server se na dané síti nenachází a směrovač (výchozí brána)[29] tak pracuje zároveň i jako Relay Agent. Dojde-li k zachycení této žádosti směrovačem, přeposílá žádost o přidělení pomocí unicastu do sítě s DHCP serverem[30]. DHCP následně pomocí DHCP Offer nabídne stanici konfiguraci. Tato zpráva může jít broadcastem nebo unicastem, zde záleží na konkrétním nastavení serveru. Server pomocí odeslaných zpráv sděluje stanici nabízenou IP adresu, masku sítě, dobu, po kterou bude IP adresa zapůjčena, IP adresu výchozí brány, adresy DNS serverů. Dojde-li stanici více odpovědí – na síti se nachází více serverů, klient typicky odpovídá na první přijatou odpověď. Následně zasílá zprávu severu DHCP Request, v které žádá přidělení nabízené IP adresy. Zpráva je odesílána broadcastem (stanice do této doby stále nemá přidělenou IP adresu). Jestliže získá potvrzení DHCP ACK, může klient až v této chvíli IP adresu začít používat.[4]

DHCP server může fungovat ve dvou variantách: přiděluje IP adresy pouze po určitou dobu, nebo přiděluje adresy na základě MAC adres trvale. V druhém případě je tak dané MAC adrese přiřazena jediná IP adresa po celou dobu existence. Tento systém se nazývá BOOTP – Bootstrap protokol. Nevýhodou je porušení předpokladu dynamičnosti sítě. Zmíněný přístup nelze využít například v otevřených, nezabezpečených bezdrátových sítích z důvodu, že se zde mohou denně přihlásit stovky zařízení. DHCP však přiděluje adresy jen z pevně definovaného rozsahu, a tak by brzy mohlo dojít k vyčerpání adres. Po vyčerpání všech možných IP adres by BOOTP začal žádosti ignorovat, či je začal odmítat pomocí zprávy DHCP NAK, a klienti by tak nemohli komunikovat mimo danou síť.

Klasický DHCP server pak propůjčuje IP adresy pouze na určitou dobu. Může se jednat řádově o minuty, hodiny, dny. Vždy záleží na daném konkrétním umístění. V zmíněném případě výše bude nastaven kratší interval než v případě domácích sítí. Po uplynutí poloviny doby zápůjčky stanice zažádá o prodloužení doby, po kterou může IP adresu používat, pomocí zprávy DHCP Request. Jestliže server neodpoví, zasílá stanice po uplynutí tří čtvrtin času tuto žádost broadcastem a žádá tak prodloužení u jakéhokoliv dalšího DHCP serveru. Takto je stanice schopna reagovat na možné výpadky DHCP serverů.

V případě, chce-li se stanice vzdát dříve než po uplynutí doby zápůjčky své IP adresy, zasílá zprávu DHCP Release. Server je tak informován, že danou IP adresu může přidělit jiné stanici. Zpráva musí být opět potvrzena serverem, aby

[29] Jedná se o hraniční směrovač na síti oddělující sítě.

[30] Unicast je v tomto případě využit, protože výchozí brána má informace o IP adrese DHCP serveru.

byla předčasně uvolněna.

Hlavními výhodami využití daného protokolu je především správa, snadné přidělení adres z pohledu stanic a jednoduché přečíslování sítě. Stačí v tomto případě pouze změnit rozsah přidělovaných adres.[2]

Oproti tomu IPv6 přináší možnosti bezestavové konfigurace bez využití výše uvedených mechanismů, stavové konfigurace s využitím DHCPv6 pro dočasné přidělení adres, nebo smíšené sítě využívající obou zmíněných způsobů. Protokol DHCPv6 nevyužívá všesměrové odesílání zpráv (broadcast), jež je využito v DHCPv4. Všeobecně protokol IPv6 podporuje typy komunikací:

- Unicast – komunikace mezi dvěma stanicemi: stanicí A společně se stanicí B.
- Multicast – tímto pojmem je myšlen skupinový přenos. Data jsou odeslána pouze vybraným členům dané skupiny. IPv6 do této skupiny přenosů zahrnuje i všesměrové vysílání a dovoluje na určitých adresách odeslání dat všem uzlům v síti, jedná se tedy i zároveň o všesměrový přenos (broadcast).
- Anycast – popisuje komunikaci postavenou na skupině výše. Data jsou odeslána vždy nejbližší stanici a tímto způsobem jsou šířena dále.

Protokol DHCPv6 komunikuje pomocí multicastu při přidělování jednotlivých IP adres. Dalším rozdílem je samotná identifikace stanic. DHCPv4 identifikuje stanice na základě MAC adres. Tyto adresy, ačkoliv jsou jedinečné, jdou změnit, podvrhnout a jsou na ně známy útoky. Z těchto důvodů IPv6 identifikuje jednotlivé stanice pomocí DUID – DHCP Unique Indetifier. Jedná se o jedinečný, trvalý identifikátor, který by neměl být spjat s technickým vybavením stanice – hardwarem a síťovým rozhraním. Zmíněný standard dovoluje tři možnosti ustanovení. Každá stanice může volit libovolný způsob i v rámci jedné sítě. DUID může být stanoveno těmito možnostmi:

- DUID-LLT – využívá spojení ethernetové adresy (MAC adresy) síťového rozhraní a libovolného časového okamžiku. Tato hodnota je následně uložena a je neměnná.
- DUID-EN – propojuje registrované číslo výrobce u organizace IANA[31] z veřejného seznamu, spolu s identifikátorem voleným výrobcem daného zařízení.
- DUID-LL – identifikátor DUID je spjat v případě této metody pouze s fyzickou adresou síťového rozhraní.

Význam zpráv je následně obdobný jako v případě DHCPv4.

Aby stanice měla možnost získat IPv6 adresu, musí být nejdříve přihlášena do sítě. Následně přichází krok vygenerování unikátní lokální linkové adresy[32]. Tato adresa je využívána pouze ke komunikaci v rámci dané sítě. Je složena propojením prefixu sítě a identifikátoru zařízení (MAC adresy). Klient nejprve musí vyčkat

[31] Celým názvem Internet Assigned Numbers Authority.

[32] Též označovaná jako link-local adresa.

na zprávu Router Advertisement od směrovače. Zmíněná zpráva Router Advertisement[33] a Router Solicitation[34] jsou součástí protokolu Neighbor Discovery Protocol, umožňujícího zasílání základních informací o síti. Směrovač v této RA zprávě oznamuje do sítě: prefix – typ využité konfigurace, dobu rozpětí zpráv Router Advertisement, dobu, po kterou bude daná výchozí brána platná, omezení TTL, které mají stanice ve svých paketech používat, ale i volitelné položky, mezi které může například patřit samotný prefix sítě, z kterého je generována lokální linková adresa, nebo omezení MTU[35]. Zmíněná zpráva je odesílána vždy v pravidelných intervalech.

V druhém případě stanice může zaslat zprávu Router Solicitation[36] na multicastovou adresu ff02::02. V této IPv6 skupině se nacházejí všechny směrovače v dané síti, a tak jsou aktivně informovány o situaci, že klient vyžaduje zaslání zprávy RA. Očekávanou reakcí je vyslání požadované zprávy do sítě. Poté, co je stanici doručena RA zpráva, zjišťuje z použitého příznaku typ konfigurace na síti. Může se jednat o příznaky: M – Managed Address configuration flag, O – Other configuration flag, anebo nemusí být příznak vůbec nastaven.

Je-li nastaven příznakový bit M, na síti se nachází DCPv6 server přidělující IP adresy. Klient tak musí výše zmíněným způsobem požádat o přidělení adresy z rozsahu.

V případě nastavení bitu O na jedna dochází ke kombinaci stavové a bezestavové konfigurace. Klient si sám generuje IP adresu popsaným způsobem níže, ale ostatní parametry získává od DHCPv6 serveru. Jedná se typicky o adresy DNS serverů.

Není-li nastaven ani jeden z příznakových bitů, znamená to, že klient bude muset přejít k bezestavové konfiguraci, též označované jako SLAAC – Stateless Address Autoconfiguration, jejíž detailní popis je v RFC dokumentu s číslem 4862. V tomto případě si stanice sama musí vygenerovat náhodnou IPv6 adresu. Následně využívá postupu z DAD – Duplicate Address Detection. Jedná se o mechanismus ze zmíněného protokolu Neigbor Discovery. Cílem je odhalení možných duplicit způsobených náhodným generováním. Ačkoliv je IPv6 adresní prostor veliký, stále existuje šance vzniku duplicit. Tyto situace by mohly vést k doručení nesprávným uzlům. Stanice pro zjištění duplicit vyšle ICMPv6 zprávu na multicastovou skupinu. V této skupině se nacházejí všechny uzly dané sítě. Za cílovou adresu zvolí svou náhodně vygenerovanou IPv6 adresu. V ICMP zprávě žádá překlad IP adresy na MAC adresu. Získá-li stanice odpověď na ICMP zprávu, znamená to, že adresa je již využita jiným uzlem. V tomto případě opakuje proces náhodného generování. V druhém případě, nepřijde-li stanici žádná odpověď, může tuto adresu využít sama ke komunikaci.

[33] Dále jen RA.
[34] Typ zprávy zmíněný níže.
[35] Celým názvem Maximum Transfer Unit, označující maximální velikost přenášených jednotek.
[36] Dále jen RS.

SLAAC je především využívaným způsobem v malých sítích pro snadnou správu. Pro větší sítě není příliš doporučován.[6]

1.1.4 Transportní vrstva

Transportní vrstva je první vrstvou, jež je součástí operačních systémů – tedy není realizována síťovými zařízeními. Její hlavní činností je zpracování požadavků od procesů na komunikaci. Jednotlivým procesům jsou pak na základě předchozího požadavku přidělovány porty, přes které jsou schopny komunikace s cílovou stanicí. Každý segment na transportní vrstvě je zaobalen záhlavím s číslem daného portu. Na síťové vrstvě je následně paketu do záhlaví přidána IP adresa. Kombinace IP adresy a portu se nazývá soket. Jedná se o jedinečný identifikátor komunikace napříč celým internetem[37].Porty jsou vybírány dle typu vyžádané služby.

Všeobecně se porty dělí do tří skupin: známé, registrované, dynamicky přidělené. Do skupiny známých portů jsou přiděleny známé aplikační protokoly/služby. Příkladem jsou protokoly http – port 80, https – port 443, dns – port 53, telnet – port 23. Skupinu registrovaných portů tvoří registrované služby u organizace IANA. Jedná se o celosvětovou organizaci dohlížející na správu síťových adres a funkcionality s tím sdružené. Příkladem jsou služby spojené se společností Microsoft, nebo službou Kerberos. Poslední skupinou jsou porty dynamicky přidělené, které nejsou rezervovány pro žádné služby. Typicky při komunikaci klient/server je vygenerován náhodný port z tohoto rozsahu pro danou komunikaci.[4]

TCP

Celým názvem Transmission Control Protocol. Jedná se o spolehlivý přenosový protokol využívaný službami, pro které by mohla být kritická ztráta informací, ale zároveň nevadí určité navýšení zpoždění z důvodu navazování spojení. Spolehlivost je zde zajištěna pomocí potvrzování přijatých segmentů. Data jsou zasílána v daném pořadí a měla by být i takto přijímána. Jestliže se z nějakého důvodu segment při přenosu poškodí, například z důvodu ovlivnění jiným signálem, protokol dovoluje opakované zasílání takto poškozených segmentů.

Spojení je navazováno pomocí zasílání příznaků. Zdrojová stanice nejdříve odešle segment s nastaveným příznakem SYN – Synchronize Sequence Number. Tím je cílová stanice informována o žádosti navázání spojení. Typicky jsou s touto informací odeslána i první data. Cílová stanice následně zasílá zprávy ACK – Acknowledgment. Zpráva potvrzuje navázání jednostranného spojení. Aby mohla cílová stanice taktéž zasílat data, navazuje identickým způsobem popsaným výše své jednostranné spojení se zdrojovou stanicí. K ukončení spojení jsou využity příznaky FIN – No more

[37]Tato jedinečnost platí pouze u veřejných adres.

Data from Sender a jejich potvrzení pomocí ACK. Spoj je definitivně ukončen až po přijetí potvrzující zprávy. Komunikace může probíhat jednostranně i oboustranně. To znamená, že i po ukončení jednostranného spojení, v případě oboustranné komunikace, mohou být data stále zasílána jednou stanicí. Dalšími využívanými příznaky jsou: URG – Urgent, označující urgentní data, PSH – push function, jedná se o data, která mají být bez čekání na další předána vyšší vrstvě, RST – reset the connection, slouží k odmítnutí spojení.[7]

UDP

Zkratka UDP označuje název User Datagram Protocol. Protokol je charakteristický opačnými vlastnostmi než TCP. Protokol UDP nenavazuje spojení a nezajišťuje ani spolehlivý přenos. Dosahuje celkově menšího zpoždění na síti. Je využíván především při přenosu velkých souborů, kde není kritické hledisko ztrátovost, ale naopak zpoždění. Typicky je využíván při přenosu multimediálních souborů, nebo při přenosu dat v reálném čase, kde výpadek (nedoručení) některých paketů, nemusí být uživatelem mnohdy ani viditelný a opakování chybně přenesených segmentů by způsobilo viditelné zpomalení služby. Tento přenos se v protokolu QOS – Quality of Service, označuje jako Best-effort. Jedná se o přenos, u kterého tedy není zajištěna kvalita a zároveň není garantováno jeho doručení. Typickým příkladem je přenášení hovorového signálu, u kterého i v případě výpadku jsou schopni se účastnici hovoru dorozumět. Z důvodu, že se jedná o službu fungující v reálném čase, není možné data přenášet opakovaně bez přílišného navyšování zpoždění.[7]

STCP

Dalším protokolem s obdobnými cíli jako u QUIC zmíněného níže – vyvinout protokol čerpající výhody UDP, TCP, je protokol STCP. Celým názvem Stream Control Transmission Protocol, schopný vykonávat služby multistreaming a multihosting.

Multistreaming umožňuje paralelní přenos více nezávislých toků dat. Data v jednotlivých tocích musí být nezávislá, aby celý proces byl efektivní. V situacích, kdy dojde k blokaci komunikace, by cílová stanice v případě závislých dat musela čekat na opakovaný přenos dalšími kanály, a tím by se protokol STCP přibližoval funkcionalitě TCP. Jednotlivé toky dále zajišťují spolehlivou službu – data jsou v daném toku přijímána v pořadí, v jakém byla odeslána.

Multihoming pak navyšuje odolnost proti selhání v případě připojení k vybrané síti. Je-li tak klient připojen k více sítím (IP adresám), snižuje se procentuální šance selhání, protože existuje možnost využití dalších cest.

Protokol je zároveň schopen rychle reagovat na změny. V pravidelných intervalech jsou zasílány zprávy o dostupnosti koncových stanic a tím je schopen zjistit

nedostupnost stanic. Navazování spojení na rozdíl od TCP probíhá pomocí prvotní žádosti klienta pomocí zprávy INIT. Touto zprávou klient žádá server o vytvoření spojení. Server potvrzuje spojení INIT ACK. Následují zprávy COOKIE ECHO a COOKIE ACK přenášející stavové cookies soubory.[4, 8]

QUIC

V dnešních sítích jsou pro přenos dat hojně využívány protokoly UDP, TCP. Protokol STCP pak je využíván především v uzavřených sítích. Nevýhodou TCP je zpoždění způsobené navazováním spojení, u UDP se následně jedná o nespolehlivý přenos. Cílem protokolu QUIC vyvinutého společností Google bylo vyvinout protokol řešící zmíněné nedostatky. QUIC snižuje celkové zpoždění při navazování spojení a zároveň navyšuje rychlost celého přenosu. Jeho existence je úzce spjata s protokolem https[38]. Podobně jako protokol STCP pak umožňuje službu multistreaming. Zde však počet jednotlivých proudů dat je proměnlivý i během přenosu. U STCP musí být definován na začátku komunikace.[4]

1.1.5 Relační, prezentační a aplikační vrstva

Relační, prezentační a aplikační vrstvy jsou obdobně jako transportní vrstva součástí operačního systému. Většinou ale nějaké konkrétní aplikace. Funkce relační vrstvy spočívá především ve vytváření, udržení a následném ukončení relací (spojení mezi zdrojovou a cílovou stanicí), synchronizaci stran a řízení přenášených dat pro nižší vrstvu. Zprostředkovává určité iterace mezi aplikační a transportní vrstvou. Oproti nižším vrstvám se při ukončování spojení tentokrát počítá s tím, že obě strany se nejprve dohodnou na ukončení spojení, a to je až poté následně ukončeno. Nižší vrstvy řeší komunikaci jednostranně, a tak může dojít k ukončení pouze jednoho komunikačního okruhu.

Cílem prezentační vrstvy jsou procesy šifrování, kódování a komprese dat. Aby byla možná komunikace dvou stran – například mezi klientem a serverem, vzniká zde předpoklad, že obě strany využijí stejný proces kódování. Kódování určuje reprezentaci dat vyšší vrstvě pro zobrazení uživateli. Typickými příklady využívaných kódů mohou být: ASCII – American Standard Code for Information Interchange, EBCDIC – Extended Binary Coded Decimal Interchange Code, nebo protokol společnosti Apple AFP – Apple Filing Protocol. Data následně mohou být přenášena v otevřeném tvaru – v čitelném tvaru pro všechny stanice, které data odchytí, nebo v šifrovaném – data jsou zakódována libovolnou šifrou a stávají se nečitelnými pro neoprávněné uzly nevlastnící klíč. Data však často obsahují nadbytečné informace,

[38] Jedná se o protokol pro šifrovaný přenos webových stránek.

které by při přenosu využívaly přílišnou přenosovou kapacitu. Z těchto důvodů se provádí proces komprese, zajišťující odstranění přebytečných informací z přenášených dat. Komprese existuje ve dvou přístupech: bezeztrátová a ztrátová. Bezeztrátová je využita v případech, je-li požadavek, aby data byla přesně shodná před procesem komprese, a i po dekompresi (reverzní proces ke kompresi). Typicky se jedná o případy přenosu textu, souborů. Ztrátová komprese pak funguje na principu trvalého odstranění informací, které již z komprimovaného souboru nelze obnovit. Tento přístup je využit kupříkladu v přenosu audiovizuálních dat, kdy lidské ucho slyší pouze frekvence od 16 do 20000 Hertzů (Hz), a tak nepostřehne ztrátu zvuku kolem hraničních hodnot. Na základě toho by bylo nadbytečné data přenášet, mohou tak být tak trvale odstraněna a přenosová kapacita snížena.

Poslední a zároveň jedinou vrstvou neposkytující žádná data vyšší vrstvě, je vrstva aplikační. Z pohledu celého modelu ISO/OSI zpřístupňuje jednotlivým síťovým aplikacím možnosti komunikace se servery. Její funkcionalita se odvíjí od potřeb daného procesu. Zároveň však jejím úkolem je následná interpretace přijatých dat od serverů k uživatelům. Běží zde několik nejznámějších a zároveň zásadních protokolů pro přenos dat. Jedná se o protokoly: http – Hypertext Transfer Protocol, https – Hypertext Transfer Protocol Secure, VoIP – Voice over Internet Protocol, FTP – File Transfer Protocol, SMTP – Simple Mail Transfer Protocol, DHCP, TELNET – Teletype over Network.[2]

1.2 Model TCP/IP

Standard ISO/OSI je však v dnešní době jen teoretický model popisu funkcí sítě. V praxi je využíván model TCP/IP definující rodinu protokolů, které mají zajistit funkce z popsaného teoretického modelu. Oproti ISO/OSI je řešení spolehlivosti komunikace přenecháno na jednotlivých stanicích. Model TCP/IP spojuje několik vrstev dohromady, a proto je složen pouze ze čtyř vrstev.

Vrstva síťového rozhraní spojuje funkce fyzické a spojové vrstvy. Tyto vrstvy neimplementuje operační systém, a tak v tomto protokolu nejsou detailně řešeny. Předpokládá se, že vrstva je tvořena jednoduchým hardwarovým ovladačem, závislým na využité přenosové technologii. Internetová a transportní vrstva jsou funkcionálně shodné se síťovou, transportní vrstvou, které byly popsány v předcházejícím modelu ISO/OSI. Poslední částí modelu je vrstva aplikační, propojující vrstvy relační, prezentační a aplikační. Ta je tvořena jednotlivými aplikačními protokoly a jejich požadavky na komunikaci.[2]

2 Řízení aktivních síťových prvků

Stěžejní součástí práce je nejen analýza provozu, ale i následné vyhledávání potencionálně nebezpečné aktivity administrátorů na síti. K odhalení takovéhoto chování jsou využívány signalizační protokoly. Ty umožňují zaznamenávat upozornění ze síťových prvků při výskytu jakékoliv události. Cílem této kapitoly je představení dvou nejběžnějších protokolů, na jejichž základě je postavena praktická část.

2.1 Logování událostí

Každé síťové zařízení na síti obsahuje určitá základní pravidla, specifikující jednotlivá oprávnění klientů. Na základě těchto pravidel by tak běžný uživatel neměl mít oprávnění (znalosti) ke konfiguraci hesel a následné modifikaci nastavení. Každá změna v konfiguraci by dále měla být zaznamenána logem. Log je tedy určitý zápis síťového prvku o události, která již proběhla, a má čistě informativní charakter. Může se jednat o události přihlášení, změny konfigurace, samotné vypnutí logování, ale i záznam o proběhlém vzdáleném přihlášení.

Všeobecně se logy dělí do skupin dle informativního charakteru na: informační, ladicí, varovné, chybové, pohotovostní. Informační popisují obecné události, které nejsou pro systém/infrastrukturu zařazeny do zbylých skupin, a tedy nikterak kritické. Mohou být však využity při zjišťování přesnějších detailů, co se dělo na síti v době kybernetického incidentu, nebo jako všeobecný přehled normálního provozu. Ladicí logy jsou využívány při vývoji systému pro odchycení nesprávných konfigurací. Logy označené systémem jako varovné se vztahují k chybějícím funkcím daného systému. Mohou ohrožovat jeho komplexní bezpečnost, či vytvářet potencionální rizika. Chybové logy jsou charakteristické obdobným způsobem jako předchozí skupina. Cílem je varování administrátorům o chybách ohrožujících funkcionality systému. Pohotovostní logy jsou z pohledu bezpečnosti nejkritičtější částí. Jejich hlavním úkolem je klasifikace potencionálně nebezpečných událostí a vydání následného varování. Na základě těchto logů mohou být útoky na sítě rozpoznány včas, což zvyšuje šance minimalizace ztráty aktiv[1].

Zmíněné logy mohou být dále zpracovávány ve dvou variantách. Textová podoba je vhodná především pro programy, očekávající určitou interaktivitu s uživatelem a není třeba využití dalších překladačů a zpracování dat[2]. Uživatel snadno rozpozná význam logu – jsou lehce čitelné. Druhou možností zaznamenávání logů je využití binární podoby. Binární formát logu se využívá z důvodu jednoduššího zpracování dalšími stroji a následného ukládání.

[1] Aktivem je myšlen hardware, software, data a další, které jsou majitelem považovány za cenné.
[2] Tento typ bude využit v praktické části diplomové práce.

Všechny firmy na trhu vyrábějící síťové prvky, například firmy Cisco, Miktotik[3] definují svou specifickou strukturu (vzhled) logu. Z všeobecného hlediska by log měl především obsahovat informace o tom, kdo jej vydal a která událost vyvolala zaznamenání logu. Dále existují předpoklady na požadavek využití jednotné časové zóny z důvodu možnosti zpětné synchronizace. Nebyl-li by dodržen tento požadavek, mohl by zde vznikat problém, kdy by kybernetická událost (útok) nebyla rozpoznána, protože přicházející logy z různých zařízení by časově nebyly shodné, a tak by systémem nebyly rozpoznány jako porušení určitých pravidel. Příkladem popsané situace by mohlo být pravidlo definující počet možných otevření spojení s příznakem TCP SYN za určitý časový okamžik. Časový okamžik vytvoření události je většinou řešen přiložením časového razítka. Toto razítko tak zajišťuje důkaz, že se logovaná událost odehrála v daný časový okamžik. Poslední stěžejní částí logu je samotná zpráva popisující sledovanou událost. Většina systému dovoluje nastavení vlastních chybových hlášek, a tak se často tyto informace napříč sítěmi liší. Aby logy mohly být odesílány do příslušných systémů k dalšímu zpracování, je potřeba využít protokoly umožňující takovéto služby.[9]

2.2 Protokol Syslog

Hlavní funkcionalitou protokolu Syslog je přenos logovaných událostí ze sítě k zařízení, kde jsou hromadně ukládány. Jedná se o princip protokolů klient/server. Ačkoliv existuje doporučení, že pro vyhledávání kybernetických událostí by informace ze sítě měly být na jednom místě, je potřeba vyřešit možnosti výpadku těchto centrálních úložišť, například pomocí redundantních serverů a zálohování informací. Těmito kroky je snížena možnost ztráty všech dat při výpadku, nebo napadení těchto serverů.

Dále je nutné při přenosu vyřešit problém samotného zabezpečení logů při přenosu. Protokol Syslog ve své standardizované verzi, v dokumentu RFC 3164, neobsahuje žádné mechanismy šifrování, a tak nezajišťuje zabezpečení dat. V této dokumentaci je pouze popsán přenos v textovém, otevřeném tvaru, což může vést k útokům, kdy útočník na síti delší časový úsek kopíruje logy z normálního provozu. Poté, co se mu podaří modifikovat uzel, začne za něj odesílat zachycené logy normálního provozu. Systém pro vyhodnocení událostí v případě, změní-li i útočník časové údaje, nemusí nic zjistit. Proti takovýmto útokům by pak stačilo podepisovat časová razítka digitálním podpisem. Ten je schopen zajistit autentizaci uživatelů/stanic a zároveň integritu dokumentu/logu. Šifrování provozu je řešeno na úrovni

[3]Touto problematikou se zabývá praktická část diplomové práce.

Aplikační vrstvy v modelu ISO/OSI s využitím protokolu TLS – Transport Layer Security.

Samotný přenos mezi stanicemi je zajištěn s využitím transportních protokolů, nejčastěji se jedná o protokoly UDP, TCP. Protokol je z hlediska transportní vrstvy spjat s portem číslo 514. Vzhledem k tomu, že ve zmíněné dokumentaci není řešena ani spolehlivost doručení, je na každém správci sítě, který transportní dokument bude zvolen. V případě UDP protokolu může nastat situace, kdy z důvodu přetížení sítě bude log zahozen a nevznikne zde ani záznam o jeho zahození.

Všeobecně protokol definuje třívrstvou architekturu systému: vrstva Syslog content – obsahuje informace k dané zprávě, vrstva Application – se stará o generování, analýzu a ukládání dat, a poslední vrstva Transport – odesílá a přijímá zprávy.

Protokol Syslog navíc definuje určité očekávané role sítě a dovoluje tímto oddělení systémů pro generování zpráv, následné analýzy a konečného uložení. Na každé zmíněné vrstvě výše jsou pak vykonávány další funkce. Předpokladem pro následující příklad je vznik události, u níž je nastaveno sledování. Nejprve původce vygeneruje zprávu (záznam) o dané události. Následně dochází ke směrování syslog zprávy mezi původcem a sběrateli, popřípadě dalšími předávacími zařízeními. Úkolem sběratele je sbírat jednotlivé zprávy a přichystat je pro další zpracování. Předávacími zařízeními mohou být zařízení pro přeposílání zpráv dalším podobným zařízením, nebo předání již konkrétním uživatelům. Dále je vybrán vhodný transportní protokol. Samotné směrování pak probíhá za pomoci principů vyplývajících z předcházející kapitoly (překlady IP adres, nalezení cíle). Přijímač přenosu, neboli cílová stanice, pak získává informace a dokáže na základě nich vyhodnotit aktuální stav sítě.

Protokol dále jednotlivým zprávám přiděluje číselnou prioritu. Na základě priorit pak může být rozhodováno, které logy při zatížení budou zahazovány. Protokol přebírá celkem sedm úrovní priorit. Nejnižší úrovní s číslem sedm je Debug úroveň. Jedná se o informace využívané při ladění systémů. Je-li systém již v provozu, nejsou tyto zprávy již zaznamenávány. Následuje Informational – informační úroveň, poskytující informace o tom, co se aktuálně děje na síti. Informace mohou být použity k sledování normálního provozu, měření propustnosti, zpoždění na síti. Priorita Notice – oznamující úroveň, specifikuje určité odchylky od normálního provozu. Především chování, které není ohrožující, ale pouze nezvyklé na danou stanici/uživatele. Pro tyto události není vyžadována okamžitá reakce. Čtvrtou úrovní je Warning – priorita varování, upozorňující na možné chyby, které za určitý časový okamžik nastanou. Příkladem může být postupné plnění paměti vedoucí k jejímu kompletnímu zaplnění. Reakce na takovéto události jsou vyžadovány v co nejkratší časový okamžik. Dalším stupněm priorit je Error – chybové události, upozorňující správce sítě na aktuální chyby na síti. Ty mohou ohrožovat normální funkcionalitu sítě, a proto musí být reakce na ně okamžité. Priorita Critical – označující kritické události, upo-

zorňuje na závažné problémy na síti, jako jsou výpadky systému, a často úzce souvisí s předposlední úrovní Alert – varující před kritickým stavem sítě. Příkladem je selhání záložních připojení nebo serveru. Poslední a nejvyšší prioritou je Emergency – neboli pohotovostní úroveň. Označuje události spojené s nefunkčností, které ovlivňují několik aplikací, serverů, webů – systém (síť) se stává nepoužitelným a je třeba okamžité reakce.

Výsledná priorita, která bude přiřazena ke zprávě, je vypočítána z následujícího vzorce[10]:

$$Priorita = Zařízení \cdot 8 + závažnost \tag{2.1}$$

Závažnost je přiřazena z výše popsaného odstavce. Správce však ještě může upřednostnit jednotlivé zařízení umělým navýšením úrovně priority. Je tak schopen sledovat například detailní informace kritických částí sítě.

Aby zprávy z protokolu byly stanicí přijaty, existuje v tomto standartu pravidlo o velikostech. Pravidlo určuje rozmezí velikosti od 480 do 2048. Jeden oktet je roven osmi bitům. Jestliže zasílaná zpráva má vyšší velikost, může být zařízeními buď zahozena, nebo uměle zkrácena. Ořezáním dojde k určité ztrátě dat a z těchto důvodů zde platí pravidlo, že významnější informace by měly být umísťovány na začátek zprávy, aby v případě dojde-li k zmíněnému procesu, nedošlo ke ztrátě informací.

Hlavička protokolu Syslog je standardizována dokumentem RFC 5234 a měla by tedy povinně obsahovat: verzi protokolu, číslo priority popsané výše, hostname stanice a domény, která data odeslala, a název aplikace, jež zprávu vytvořila, ID procesu, časovou značku a samotný popis události. V časové značce musí být definice využité časové zóny a označení začátku a konce řetězce.[10]

2.3 SNMP

Celým názvem: Simple Network Management Protocol je protokol s obdobnými vlastnostmi jako protokol Syslog. Jeho úkolem je přenos informací o tom, co se na síti děje, ale navíc dovoluje i vzdálené řízení zařízení, a to bez ohledu na typ hardwaru, softwaru. Jedinou podmínkou je zde fakt, že zařízení podporují SNMP. Jedná se o protokol běžící na aplikační vrstvě. K přenosu využívá transportní protokol UDP na portu 161 a 162.

Architektura SNMP definuje tři základní role – správce, manažer a agent. Správci komunikují s manažery a ti následně získávají informace od jednotlivých agentů na síti. Jedná se o hierarchický systém. Správcem je myšleno centralizované úložiště, kde jednotliví manažeři zasílají informace o tom, co se aktuálně děje na síti. Jeho úlohou je i následná analýza dat, vyvození potřebných alarmů a upozornění správce na podezřelé události. Úlohou manažera je komunikace s jednotlivými přiřazenými

agenty. Agenti pracují nad jednotlivými aplikacemi na zařízeních, z kterých přebírají informace. Manažeři společně s přidělenými klienty tvoří komunitu SNMP. Jedna stanice může zastupovat pozice manažera a agenta v případech, vykonává-li správu prvku na dálku.

Zasílané zprávy do centrálního úložiště jsou identifikovány dle názvů komunit, z jaké části sítě jsou zprávy odesílány. Název komunity je řetězec znaků o délce několika bajtů, který je náhodně vygenerován. V dřívějších verzích se tato hesla (názvy) komunit odesílaly v otevřeném formátu, což způsobovalo bezpečnostní slabinu. Na základě znalosti tohoto jména lze získat plný přístup k veškerým údajům, které komunita odesílá. V původních návrzích – ve verzi jedna a dva, tak mohl útočník snadno odchytit hesla a získat plný přístup k těmto údajům.

Od třetí verze je podporován proces šifrování a vyššího způsobu autentizace. K šifrování je využita symetrická šifra DES – Data Encryption Standard, nebo AES – Advanced Encryption Standard.

Kryptografie rozděluje dva základní druhy šifer – symetrické šifry, asymetrické šifry. Symetrické šifry využívají pro šifrování a dešifrování stejný klíč (tajemství). Klíč pak musí být utajen po celou dobu přenosu. Asymetrická kryptografie je pak postavena na dvou klíčích – veřejném a privátním. Platí zde předpoklad, že na základě znalosti jednoho klíče, nelze odvodit druhý.

Z pohledu bezpečnosti dle doporučení instituce NIST – Národní institut standardů a technologie je DES, sám o sobě se základní délkou (64 bitovým klíčem) pro šifrování, brán za prolomený. To znamená, že jde s dnešními běžnými nástroji prolomit zprávu zašifrovanou DES řádově v minutách s využitím útoku hrubé síly. Tento útok definuje postup, kdy útočník zkouší náhodné řetězce do doby, dokud není heslo prolomeno. V bezpečnostních doporučeních je proto uveden jeho bezpečnější nástupce AES.

K autentizaci jednotlivých komunit a zajištěni integrity zpráv[4] je využit protokol HMAC – Keyed-hash Message Authentication Code. HMAC je založen na využití bezpečného hashe, společně s tajným klíčem. S využitím logických operací a hashové funkce je pak tajný klíč propsán do výstupního bloku.

Manažeři v pravidelných intervalech zasílají agentům zprávy GetRequests, v kterých žádají informace o aktuálních stavech MIB – Management Information Base z jednotlivých agentů. MIB popisuje strukturu jednotlivých sledovaných parametrů. Toto spojení je vytvořeno obousměrně, a tak může probíhat komunikace typu klient/server. Manažeři dále využívají spojení k odeslání informace o změně těchto definovaných parametrů. Dojde-li ke změně stavu agenta, či k situaci, kterou agent vyhodnotí jako neobvyklou, odesílá zprávu trap manažerovi, ve které jej informuje

[4]Jedná se o rozpoznání modifikace zprávy.

o dané situaci v reálném čase. Spojení je tentokrát jednosměrné.[11]

2.4 Porovnání protokolů SNMP a Syslog

Standarty SNMP a Syslog jsou podobnými protokoly pracujícími s informacemi o událostech, které na síti nastaly. SNMP pak navíc dovoluje i určitou správu sítě. Syslog je schopen vyhodnocovat závažnost upozornění a filtrovat tak při zatížení na základě přidělených priorit. Podstatnou změnou mezi protokoly je ovšem typ komunikace. Zprávy Syslog jsou řazeny do fronty, a tak se může navyšovat zpoždění, než dojde k doručení do centrálního úložiště a následnému zpracování informací o události. SNMP naopak využívá komunikaci pomocí trapů v reálném čase a je schopen rychleji informovat o neobvyklých událostech, které nastaly. Každá síť má však individuální vlastnosti, a tak nelze definovat jediný protokol, který by byl ideálním řešením pro všechny sítě.

3 Praktická část

Cílem navrhovaného programu je odhalení potencionálně nebezpečných změn provedených samotnými administrátory, popřípadě uživateli, s oprávněním měnit konfigurační nastavení zařízení v dané síti. Program na základě vyhledávání konkrétních předpisů, které specifikuje manažer kybernetické bezpečnosti popřípadě jiný uživatel na podobné úrovni, bude moci upozornit například na přenastavení časového pásma, vypnutí jednotlivých rozhraní nebo změnu skriptovacích pravidel. Ve své podstatě by měl odhalit jakékoliv změny v konfiguračních souborech jednotlivých zařízení.

K tomu, aby však síťová zařízení – přepínače, směrovače, mohla na síti zachytávat určité události a následně je logovat (vytvářet záznamy), musí mít nastavena nejen pravidla, která takovéto chování definují, ale i základní nastavení popsané v následující podkapitole.

3.1 Základní konfigurace síťových zařízení

Všeobecně zařízení fungují na principu: je-li uživatel připojen k danému zařízení, může v konzoly vidět zachytávané události, které jsou nastaveny v původních pravidlech. Takovéto zachytávání je v defaultním nastavení povoleno, a tak pro výše zmíněné chování uživatel nemusí žádným způsobem modifikovat nastavení.

Aby však mohl navržený externí program procházet jednotlivé logy a vyhledávat v nich odchylky od normálního chování, nebo mimořádné události, je třeba logy odesílat do centrálního úložiště. Následující konfigurace bude vytvořena s předpokladem využití protokolu Syslog, popsaného v teoretické části práce.

3.1.1 Nastavení zařízení od firmy Cisco

Předchozí kapitola uvedla v protokolu Syslog celkem sedm definovaných úrovní bezpečnostních upozornění – kritická, mimořádná, varování, upozornění, oznámení, mimořádné události, chyby, informační úroveň a poslední je upozornění, které je spjato s režimem ladění. V původním nastavení síťových prvků by mělo být zakázáno odesílání takovýchto dat na server pro další zpracování.

Nejprve je tak potřeba nastavit IP adresu serveru na daných zařízeních, kam se jednotlivé vytvořené logy mají zasílat. Tato konfigurace je provedena následujícími příkazy:

```
R1(config)# logging host X.Y.W.Z
R1(config)# logging traps informational
```

Po zapnutí logování by měla být defaultně přednastavena úroveň informační, avšak program musí počítat s možností, že nebude implementován na nová zařízení,

a tak tyto možnosti mohou být pozměněny. Proto je součástí zmíněných příkazů nastavení úrovně logování, od které budou zasílány logy na server. Příkaz zajišťuje logování všech nižších úrovní od zadané úrovně. V tomto případě dojde k logování všech úrovní s výjimkou ladící.[12]

Dojde-li k vyřazení směrovače nebo výpadku konektivity, je potřeba zachytávané informace dočasně uložit do RAM paměti zařízení. RAM paměť je pouze dočasný typ paměti. To znamená, že v případě vypnutí či restartu zařízení je kompletně smazána.[13]

```
R1(config)# logging buffered 16384
R1(config)# logging buffered warnings
```

Aby proběhlo uložení vytvořených logů do paměti RAM, musí mít zařízení vyhrazené místo pro takováto data. Použitím uvedených příkazů dojde k rezervaci 16384 bajtů paměti pro Syslog zprávy. Z důvodu minimalizace plýtvání paměti pro méně závažné události se budou do RAM paměti ukládat logy od úrovně varování. Tím bude zajištěna dostatečná kapacita pro kritické události.

Vzhled logu od firmy Cisco je složen z:

- časového razítka,
- identifikace zařízení nebo příčiny vzniku systémové zprávy,
- závažnosti zachycené události,
- jednoznačného popisu zprávy,
- detailnějšího popisu události.

```
*Mar 04, 23:55:00:000: %LINK-5-CHANGED: Interface
GigabitEthernet0/0/1, changed state to up
```

Obr. 3.1: Ukázka logu zařízení firmy Cisco.

Napříč sítí by dále měl být čas na jednotlivých zařízeních synchronizován z důvodu možnosti spojování podezřelých událostí. Existují dva způsoby nastavení: manuální, nebo automatické. V případě automatického procesu je vytvořen NTP klient a čas bude převzat od NTP serveru.

Pro převzetí času z NTP serveru lze využít příkaz:

```
R1(config)# ntp server X.Y.W.Z
```

Následně je třeba povolit časová razítka ve zprávě Syslog. Druhou variantou, která může být využita namísto časových razítek, je pořadové číslo. To by znamenalo, přiřazování jednotlivým zprávám číslo z rozsahu, které je postupně inkrementováno o jedna. Tato možnost však v základní konfiguraci pro program nebude využita z důvodu potřeby sledování časových oken.

```
R1(config)# service timestamps log datetime msec
```

Položka závažnosti v ukázkovém logu pak popisuje stupeň závažnosti, popsaný v teoretické části. Rozhodnutí, do které skupiny daná událost bude začleněna, nemusí být žádným příkazem nastavováno.

Zdrojem zprávy je popsán proces, který logoval událost a vygeneroval následnou zprávu. Mezi nejčastěji využívané patří:

- %LINEPROTO – zdrojem je využívaný linkový protokol ze spojové vrstvy,
- %LINK – využívá se při změně stavu rozhraní,
- %SYS – pro obecné zprávy ze systému.

Celý seznam možných zdrojů je uveden v oficiálních dokumentech firmy Cisco.[13]

Jednoznačný popis události následně popisuje, zda došlo k zapnutí, vypnutí, či změně stavu sledovaného procesu nebo rozhraní. Za těmito body se nachází detailnější popis události, jehož součástí mohou být IP adresy, porty. Správce sám může pozměnit popis dané události a upravit jej pro specifikace své sítě.[12]

3.1.2 Nastavení zařízení od firmy MikroTik

Obdobnými příkazy pak lze nastavit veškeré výše zmíněné nastavení pro zařízení firmy MikroTik. Zde však dochází k menším odlišnostem. Nejprve je opět potřeba nastavit IP adresu serveru, na který se budou události logovat:

```
/system logging action set remote=X.Y.W.Z
```

Následně je potřeba nastavit sledovanou úroveň pro protokol Syslog. Příkazem níže proběhne nastavení obdobným způsobem jako u zařízení Cisco.[15]

```
/system logging action syslog-severity info
```

Dále je potřeba nastavení časových razítek jednotlivých logů pro odlišení událostí v dnech a konkrétních časech. Zařízení firmy MikroTik dovolují nastavit až dva NTP servery pro získání času. Primární server je vždy kontaktován první. Nezíská-li zařízení odpověď, například z důvodu výpadku uzlu, či při vzniku problému s komunikací, může kontaktovat sekundární server a požádat jej o aktuální čas. Za sekundární server je vhodné volit veřejně dostupné servery, u kterých je snížena šance nedostupnosti. Nastavení probíhá příkazy:

```
/system ntp client set enabled=yes primary-ntp=X.Y.W.Z
secondary-ntp=X.Y.W.Z
```

Specifikace sekundárního serveru však není povinnou součástí výše zmíněného příkazu.

Dále zde dochází k odlišnému značení časových značek firem Cisco a MikroTik. Zatímco časová značka firmy Cisco je složena z data a času, ve kterém se událost stala, zařízení firmy MikroTik využívá toto značení pouze u událostí z předchozích dnů. To znamená stane-li se sledovaná událost v daný den, odešle časové razítko pouze s údajem o hodině, minutách a sekundách. Odesílá-li se záznam o události starší několika dnů, je časová značka složena ze dne, měsíce, roku, hodin, minut a vteřin.

Všeobecně je tedy log MikroTiku složen z:

- časové značky,
- způsobu uložení logu,
- názvu protokolu/události, ke které log patří,
- významu události podle Syslogu,
- významu události podle firmy MikroTik,
- detailnějšího popisu události.

```
23:33:33 route, ospf, debug, raw 00 00 00 00 00 00
00 00 00 00 00 00
```

Obr. 3.2: Ukázka logu v zařízení MikroTik.

Logy taktéž obsahují poznámku o tom, kde jsou uloženy. V defaultním konfiguraci je nastaveno ukládání do paměti. Toto nastavení však lze přepsat a ukládat logy například pomocí emailu, disku, pouze je zobrazovat do konzole, či je odesílat na vzdálený server.[16]

Význam události může být značen pomocí protokolu Syslog, nebo pomocí rozložení bezpečnosti od firmy MikroTik. Tyto bezpečnostní úrovně se však lehce odlišují od specifikovaných úrovní u zařízení od firmy Cisco a popisu Syslogu sepsaného v teoretické části. MikroTik definuje celkem sedm svých úrovní možností logování bez bez ohledu na protokol, kterému je událost přidružena: kritická – critical, ladící – debug, chyby – error, informativní – info, varování – warning, sledování paketů – packet, sledování dat v paketech – raw. Významy úrovní informativní, ladící, varovné a kritické jsou shodné s firmou Cisco.

Sledování paketů umožňuje logování přicházejících, odcházejících paketů a zaznamenávání z jakých IP adres na jaké IP adresy jednotlivá zařízení komunikují. Nastavení je vhodné především v kritických sítích, kde jednotlivá zařízení mají dovoleno komunikovat jen s dalšími, vybranými zařízeními.

Úroveň sledování obsahu paketů pak dovoluje navíc i sledovat obsah jednotlivých paketů před zpracováním dalšími zařízeními. Dochází tak teoreticky u nastavených

zařízení k vytvoření duplicity celého paketu a následného odeslání ve zprávě na požadovaný server. Těmito kroky dochází k zvýšení možnosti zatížení sítě.

Cílem těchto značek je co nejvíce vystihnout a jednoduše popsat událost, která nastala. Jeden log tak může mít jiný význam události v protokolu Syslog a jinou úroveň definovanou správcem (MikroTikem). Výhodou druhé formy značení je to, že správce si sám jednoduše poupraví specifikaci jednotlivých úrovní.[15]

Pro navrhovanou aplikaci však v této kapitole není stěžejní specifikovat jednotlivé úrovně nebezpečí a přidružovat k nim dané protokoly dle jejich stupně zabezpečení.

3.2 Realizovatelné útoky

Hlavním cílem této práce je navrhnout program pro odhalení potencionálně nebezpečného chování samotných administrátorů. Roviny možných útoků lze rozdělit do dvou skupin. Do první skupiny jsou přiřazeny útoky, které lze programem odhalit, jak ukazuje následující příklad.

Správce může výše popsané nastavení pro odesílání na Syslog servery pomocí terminálu jednoduše vypnout, modifikovat, či může vypnout konkrétní rozhraní, přes které se data odesílají. Těmito kroky zabrání odeslání dat na server a následné analýze jeho chování. Zařízení by dle výše nastaveného mělo začít v tento moment logovat události do paměti. Administrátor po vypnuti odesílání vykoná škodlivé přenastavení, stažení, změnu dat, nebo konfiguračních souborů, a poté vymaže záznamy o proběhlých událostech v paměti. Tím je jeho chování pro program dočasně utajeno. Zařízení se tváří, že došlo pouze k výpadku při opětovném připojení.

Po realizaci popsaného útoku zařízení neodešle změnu svého stavu na Syslog server, a tak manažer bezpečnosti nemusí být o této změně ihned informován. Program však obsahuje určité mechanismy, kontrolující chování jednotlivých síťových zařízení. Jedná se například o princip časového sledování, což znamená, že neodešle-li se log ze zařízení za hlídaný časový interval, je vypsáno varovné hlášení, že ze zařízení o dané IP adrese nepřišel v daném úseku žádný záznam o událostech. Tímto krokem může být odhaleno: přenastavení IP adres, změna adresy Syslog serveru, vypnutí portu pro komunikaci.

V případě, že administrátor zasáhl do nastavení zařízení například přepsáním logovacích pravidel, přidáním skriptu, popřípadě nastavil zrcadlení obsahu, bude tato změna zaznamenána v konfiguračním souboru. Program bude uchovávat původní konfigurační soubory jednotlivých zařízení, čímž je schopen odhalit i takovéto změny.

Dalším možným útokem by následně mohlo být přenastavení povolených služeb, s kterými jsou jednotlivá zařízení spjata. Využití nezabezpečených protokolů typu

Telnet je však již monitorováno v bezpečnostních úrovních protokolu Syslog ve vysokých úrovních – tedy označené mezi závažnými událostmi. Dojde-li však k výše popsaným změnám – vypnutí stěžejního rozhraní, může být toto nastavení modifikováno, a tak nemusí být rozpoznáno vykonání požadavku k vybrané službě. Těmto krokům by všeobecně měly zabránit zálohy konfigurace.

Obecným problémem je ovšem fakt, že zálohy konfigurace jsou vytvářeny typicky jednou denně, či za delší časový úsek. Takže bude-li administrátor postupovat níže specifickým způsobem, bude téměř nemožné jeho chování odhalit, i přestože program navrhuje redundantní, kritické funkce, které budou sledovat: odchylky v konfiguraci, změny chování administrátora a změny v celkové topologii sítě.

Nejprve musí dojít k přihlášení administrátora ke konfiguraci síťového zařízení. Typicky jsou služby autentizace spjaty se servery typu Radius či Tacacs. Jedná se o servery, které na základě vytvořených účtů jsou schopny autorizovat jednotlivé příkazy. Jsou tak schopny rozpoznat jednotlivé administrátory a jejich zadávané konfigurační změny, a to i v případech, je-li více administrátorů přihlášeno k jedinému síťovému zařízení.

Po vypnutí portu, který je však monitorován níže uvedeným příkazem, se začnou události logovat do vyrovnávací paměti. To znamená, že se do paměti nahraje první záznam o vypnutí rozhraní. Poté by měl administrátor teoreticky čas vykonat škodlivou činnost. Zařízení by se v tuto chvíli tvářilo, že došlo kupříkladu k výpadku elektrické energie.

```
R1(config)#logging source-interface X
```

Posledním krokem může být navrácení nastavení do původního stavu s tím, že by byla vymazána mezipaměť, aby nebyl o škodlivé činnosti server informován. Zařízení by tak po opětovném zapnutí rozhraní pro odesílání zpráv Syslog neodeslalo žádné záznamy o událostech, protože by žádné logy nemělo uložené.

Předpokladem pro tento útok je také fakt, že správce je seznámen s časovým nastavením vytváření záloh, a tak ví, kdy musí nastavení vrátit do původního stavu, aby nebyla odhalena jeho činnost. V tomto konkrétním případě by nebylo možné odhalit na zařízení změny ani pomocí záloh konfigurace, byla-li by do daného časového intervalu vrácena do předchozího stavu.

Tímto specifickým útokem byla představena druhá skupina možných, realizovatelných útoků, které však program není schopen odhalit a varovat uživatele o nelegitimních, proběhlých změnách, z důvodu, že za sebou nezanechají potřebné logy. Program pouze v tomto případě může odhalit odchylku chování administrátora od běžného chování, popřípadě nepřítomnost žádného logu ze zařízení za kontrolovanou časovou dobu. Obě tyto funkcionality budou detailně popsány dále.

Záloha konfigurace pro zařízení Cisco může proběhnout následujícími příkazy:

```
R1(config)#archive
R1(config-archive)#path tftp://X.Y.W.Z/nazev_zalohy
R1(config-archive)#maximum 5
R1(config-archive)#write-memory
R1(config-archive)#time-period 4320
```

Výše uvedenými příkazy se nejprve přepne uživatel do konfiguračního módu pro vytváření záloh. Následně nastaví cestu složenou z IP adresy s využitím protokolu TFTP – Trivial File Transfer Protocol. Tento protokol slouží především pro přenos konfiguračních souborů. Jeho nevýhodou je komunikace v otevřeném tvaru. To znamená, že každý uživatel na síti může odchytit přenášený text, protože není řešeno jeho utajení - šifrování. Modifikovanou verzí, která využívá šifrování, je pak FTPS, kde je krok utajení zajištěn protokoly SSL, nebo TLS. Tato verze protokolu je však složitější a pro přenos záloh v uzavřené síti nadbytečná[1].

Následují příkazy vytvářející zápis samotné zálohy. Celý proces lze buďto vytvářet manuálně pomocí výše uvedeného příkazu, kterým ve chvíli zadání bude vytvořena první záloha konfigurace, nebo automatizovaně, jak je uvedeno v posledním příkazu. Nastavení času periody pro tento proces probíhá v minutách. V tomto konkrétním příkladě je nastaveno 4320, což odpoví třem dnům. Takže každé tři dny bude vytvořena nová záloha.

Zařízení firmy Cisco však zároveň dovoluje i monitoring jakékoliv stávající změny konfigurace pomocí výše zmíněného konfiguračního módu.

Uvedeným předpisem níže se nejprve uživatel přepne do konfiguračního souboru pro archivaci změn aktuálního nastavení. Následně je povoleno logování a vytváření samotných záznamů o změnách v konfiguračních souborech. V defaultním nastavení je povoleno až 100 záloh, což pro dané využití bude dostatečné. Bude-li však administrátor seznámen s nastavením zařízení, může i tuto funkcionalitu snadno vypnout.

Poslední příkaz specifikuje odeslání těchto záloh na server Syslog, jehož specifikace byly nastaveny v předchozích kapitolách. Těmito kroky může být částečně zabráněno útokům na pozměnění konfigurace v aktuálním čase, a ne až při porovnání celých záloh. Je-li navíc k ověření autentizace využit autentizační server, je toto chování snadnější odhalit, z důvodu, že každý příkaz je spojen s daným správcem.[14]

```
R1(config)#archive
R1(config-archive)#log config
R1(config-archive-log-cfg)#logging enable
R1(config-archive-log-cfg)#notify syslog
```

[1] Uzavřenou sítí je v tomto kontextu myšlena síť, ke které nemá přístup neautorizovaný uživatel.

Pro firmu MikroTik je nastavení obdobné. Nejprve je potřeba nastavit automatické vytváření záloh, které bude propojeno se skriptem, schopným odesílat zálohy na server. Příklad konfigurace je ukázán níže.

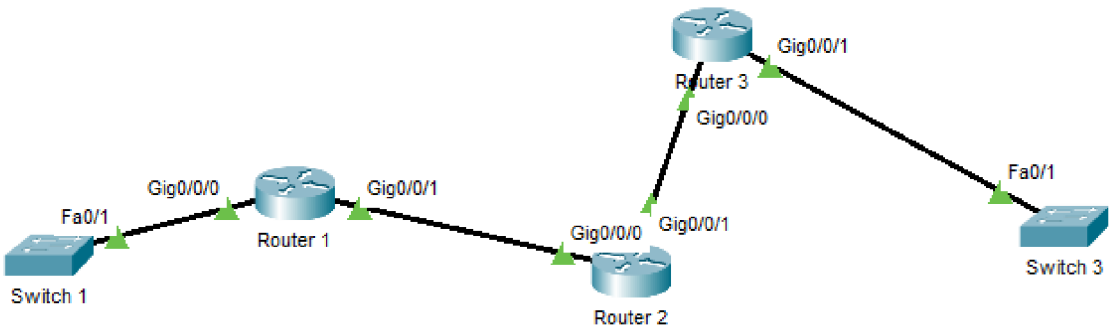
```
/system scheduler
add interval=3d name=zaloha
on-event=\system backup save name=jmeno_zalohy
```

Uvedený příklad skriptu je nastaven pro vytvoření automatické zálohy jednou za tři dny.[17]

Program však bude nabízet alternativu, a to možnost kontroly samostatně vytvářených záloh. V tomto případě tak bude stačit pouze namapovat složku souboru, kde budou jednotlivé zálohy uloženy. Následně program podle daného jména zálohy porovná změny se svou uloženou zálohou a vyhodnotí změny v konfiguraci.

Po implementaci popsaných kroků výše by mělo být určitým způsobem zabráněno nechtěným změnám konfigurace a manipulace se síťovými prvky. Jestliže však administrátor nebo útočník svou činnost zakryje, ať již odpojením fyzických zařízení od sítě či vypnutím portů pro komunikaci, nemusí být toto chování vždy odhaleno. Možné zabránění odpojení by mohlo být pomocí sledování směrovacích tabulek jednotlivých směrovačů při využití dynamického směrování. Z těchto záznamů by však šel opět odhalit pouze výpadek trasy nebo celého zařízení, a ne příčinu aktuálního stavu.

Aby program byl schopen realizovat i odhalení výpadku trasy, je třeba znalosti celé topologie sítě. To znamená, že uživatel bude muset zadat do předpřipraveného souboru topologii sítě. Tedy specifikovat určité charakteristiky: o jaké zařízení se jedná, s jakými dalšími zařízeními je spojeno a pod jakou IP adresou komunikuje se Syslog serverem, tak jak je zobrazeno na příkladu níže.



Obr. 3.3: Příklad topologie sítě.

Topologie byla vytvořena v testovacím prostředí Packet Tracer od firmy Cisco.

Jedná se o síť složenou ze tří směrovačů, dvou přepínačů a jejich spoji mezi rozhraním. V takovémto případě by měl program obsahovat níže popsanou tabulku.

Tab. 3.1: Příklad tabulky pro zadání cest vycházející z výše zobrazené topologie.

Označení zařízení	**Počáteční rozhraní a IP**	**Konečné rozhraní a IP**
Router 1	Gig0/0/0 192.168.20.1	Fa0/192.168.20.5
Router 1	Gig0/0/1 192.168.20.1	Gig0/0/0 192.168.19.1
Switch 1	Fa0/1 192.168.20.5	Gig0/0/0 192.168.20.1
Router 2	Gig0/0/0 192.168.19.1	Gig0/0/1 192.168.20.1
Router 2	Gig0/0/1 192.168.19.1	Gig0/0/0 192.168.17.1
Router 3	Gig0/0/0 192.168.17.1	Gig0/0/1 192.168.19.1
Router 3	Gig0/0/1 192.168.17.1	Fa0/1 192.168.17.5
Switch 3	Fa0/1 192.168.17.5	Gig0/0/1 192.168.17.1

Ačkoliv z pohledu směrování je rozložení jednotlivých IP adres na rozhraní nesmyslné, program nepotřebuje znalost kompletní topologie a jejich specifických adres na určitých rozhraních. K odesílání zpráv do Syslog serveru bude využito konkrétní rozhraní s konkrétní IP adresou, a tak je potřeba odlišit jednotlivá zařízení právě na základě známých IP adres pro daný server. Z těchto důvodů je v tabulce k danému rozhraní připsána adresa využívaná Syslog serverem.

Na základě těchto událostí může být rozpoznáno případné maskování útoku za pomoci odpojení fyzického kabelu, protože program může využít nejen znalosti zapojení, ale i samotného okolí kolem sledovaného síťového prvku, jak ukazuje popsaný příklad níže.

V případě fyzického útoku odpojením směrovače s označením Router 2 se spojem Gigabit Ethernet 1 by měla být doručena Syslog serveru zpráva z protějšího směrovače Router 3 o vypnutí rozhraní Gigabit Ethernet 1. Zároveň v tomto případě program ví, že Router 2 sousedí se směrovačem s označením Router 1. Od něj však zpráva o vypnutí rozhraní nepřišla. Je tedy pravděpodobné, že nenastala chyba na celém fyzickém prvku, ale pouze na konkrétním spoji. V tomto případě program vydá kritické varování o výpadku konkrétní linky. V případě výpadku i dalšího spoje je pravděpodobné, že došlo k výpadku celého zařízení a program by tak vydal varování i o této události.

3.3 Inicializace zařízení a vytvoření profilu uživatele

Předchozí podkapitola praktické části nabízí uživateli dle typu výrobce - MikroTik, Cisco - příklad základní konfigurace síťových zařízení. Ačkoliv však uživatel vybere

jednoho ze zmíněných výrobců, existuje možnost, že zařízení buďto část, či celou konfiguraci z nějakého důvodu nebude brát jako platnou a odmítne ji tak vykonat. Tato situace může nastat především v případě využívaní starších modelů, které nemusí podporovat dané konfigurační soubory. V tomto případě si tak uživatel musí sám, dle konkrétního typu zařízení, najít alternativní příkazy platné pro daný model a implementovat popsané bezpečnostní kroky.

Výsledný software monitoruje neobvyklé chování administrátorů, což může upozornit například na situace, kdy dojde k prolomení hesla. Aby bylo možné kontrolovat chování těchto jednotlivých správců, je potřeba vytvořit profil uživatele.

Nejprve je tedy potřeba nastavit parametry definující normální chování jednotlivých uživatelů sítě. Zmíněný profil bude složen z údaje obsahující běžnou pracovní dobu. Byl-li by přidán například údaj o IP adresách zařízení, na které je běžně přistupováno, mohlo by zde docházet k falešně pozitivnímu vyhodnocení z důvodu, že typicky jeden administrátor nespravuje sám dané zařízení, ale v případě potřeby se připojuje a následně konfiguruje různá zařízení.

Specifikace běžné pracovní doby může být proměnlivá v průběhu roku, z toho důvodu je umožněno provádět v jednotlivých profilech změny. Nastavení jednotlivých úseků proběhne samotným uživatelem. Vyskytne-li se tak záznam o události mimo specifickou dobu, bude vypsáno varovné hlášení o nalezené události.

Poslední fází programu je již samotné vyhledávání události dle specifikovaných pravidel výše. Ačkoliv výsledný program by měl odhalit potencionálně nebezpečnou činnost administrátorů/útočníků, existují i tak nadále možné cesty, kterými lze vytvořený log poškodit, smazat a těmito kroky zamaskovat svoji činnost.

3.4 Rozšíření programu pro další výrobce

Program bude primárně uzpůsoben pro síť složenou ze síťových zařízení od firem Cisco Systems a MikroTik. Bude-li se na síti nacházet síťové zařízení od jiných výrobců, je možné, že program by nebyl schopný vyhodnocovat takovéto logy z důvodu odlišného vzhledu – tedy z důvodu odlišného přeskládání rozpoznatelných, klíčových charakteristik v logu. Každý výrobce těchto zařízení si uzpůsobuje vzhled logů podle vlastních preferencí. Mezi další výrobce patří například firmy: Juniper Networks, Brocade Communations Systems.

Z těchto důvodů tak program bude obsahovat funkci pro možnost nové specifikace charakteristiky logů pro přidání dalších výrobců. V nové charakteristice bude třeba poskládat vzhled logu z polí: časové razítko, závažnost, text zprávy, proces/událost, kterou událost vyvolala a v neposlední řadě výrobce a typ samotného zařízení, od něhož zpráva přišla.

Tab. 3.2: Příklad tabulky pro rozdělení informací v logu pro obr. 3.1 Ukázka logu zařízení firmy Cisco.

Časové razítko	**Závažnost**	**Proces**	**Identifikátor zařízení**
Mar 04, 23:55:00:000	5	LINK	-

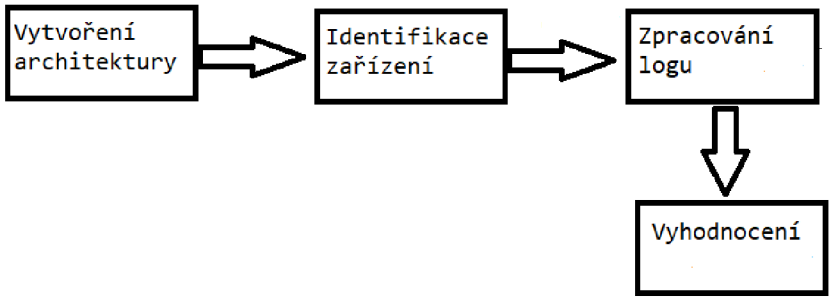
Událost	**Znění logu**
zapnutí rozhraní	Interface GigabitEthernet 0/0/1, changed state to up

Jestliže logy budou obsahovat další informace, může nastat situace, kdy program nebude uzpůsoben pro jejich zpracování. Z těchto důvodů je třeba vytvořit rozpoznání jednotlivých událostí a možné rozdělení řetězců. Program tak bude mít přiloženou tabulku, kde uživatel nejprve nakopíruje log příslušné odpovídající události a následně jej i sám rozdělí. Program díky tomu bude schopen rozpoznat, kde se jednotlivé klíčové charakteristiky nacházejí v původním řetězci a bude moci s těmito logy následně pracovat.

4 Popis přiloženého programu

Poslední část diplomové práce je především zaměřena na detailnější popis praktické realizace a vzniklé problematiky, vyplývající z teoretického návrhu předchozího textu. Kapitola podrobněji vysvětluje implementaci jednotlivých kroků, za pomoci algoritmů, využitých v přiloženém programu.

Aby mohl program plnit funkce vyplývající z předcházející kapitoly – tedy rozpoznat potenciálně nebezpečné chování administrátorů, projevující se prováděním nelegitimních změn na síťových zařízeních, je potřeba vykonat několik kroků. Prvním je kompletní namapování architektury sítě. Ta je poté využita především ke kontrole změn IP adres, rozpoznání komunikace se Syslog serverem a samotné kontrole chování zařízení. Dále program musí identifikovat výrobce společně s kategorií zařízení, na jehož základě může dojít k následnému vyhodnocení logované události. Zmíněný postup je detailněji rozebrán v následujícím textu.



Obr. 4.1: Obecný popis navrhovaného programu.

4.1 První fáze programu

Jak bylo uvedeno v předchozí kapitole, program bude celkově pracovat s pěti soubory, v nichž bude definována celá struktura sítě a programu. Jedná se o soubory popisující architekturu sítě, části specifikující jednotlivá zařízení a pravidla, dle kterých proběhne vyhledávání konkrétních událostí. V neposlední řadě se jedná o soubor určující rozdělení logů od různých výrobců. Pro všechny tyto soubory byl vybrán formát, schopný pracovat s tabulkovým typem dat. Jedná se o soubor csv – Comma-separated values.

Kód programu je uzpůsoben k vyhledávání těchto specificky umístěných a pojmenovaných souborů z důvodu, aby uživatele snadno upozornil na možné chyby během zadávání, dále odhalil jednoduché překlepy, neexistenci cest, popřípadě byl

schopen vyhodnotit chybějící údaje. Všechny tyto zmíněné části budou detailněji popsány dále.

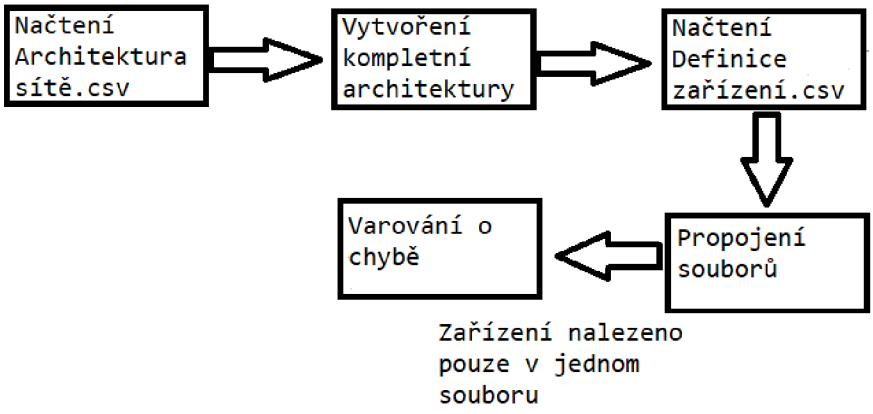
Nejprve proběhne importování cesty z operačního systému do proměnné, kterou program bude moci následně modifikovat. Importovaná cesta definuje vybrané složky v daném operačním systému, které musejí být projity ve chvíli, kdy dojde ke vzniku požadavku pro běh programu. Nejedná se o jeden očekávaný řetězec, uživatel bude moci uložit program do libovolné části paměti, kam program bude mít přístup.

Na základě získaného textového řetězce a znalosti názvu cílové složky „Soubory programu", je vytvořena očekávaná posloupnost, vedoucí ke kroku samotného načtení. V této složce by následně mělo být umístěno celkem pět souborů popsaných detailněji dále. Jedná se o základní podklady potřebné pro správný běh algoritmu.

Program se však nemůže spoléhat na skutečnost, že uživatel bude mít vždy všechny očekávané soubory, nebo například složky na místě, kde je očekává, a proto samotné načítání souborů je obaleno blokem schopným rozpoznat jakoukoliv nastalou chybu. V tomto případě se může jednat o chyby způsobené přístupem k neexistujícímu souboru a dalšími neočekávanými stavy.

Výše zmínění chybový blok tak nejprve rozezná, zda v jeho hlídaném bloku nastala neočekávaná událost – chyba. V případě nalezení této události ukončí vykonání aktuální části a přejde k vykonání kroků v následujícím bloku, který je určen k ošetření chybových stavů. Druhý blok je načten pouze v případě, nastane-li chyba v prvním bloku. Jedná se o metodu známou v programovacím jazyku Python jako: „try a except"[1].

Následující podkapitoly podrobněji popisují jednotlivé úlohy, které program musí vykonat, aby mohl přijatý log zpracovat, jak zobrazuje obrázek níže.



Obr. 4.2: Obecné kroky vykonané při spuštění programu.

[1]Jedná se o obdobnou funkci, jako try and catch pocházející z programového jazyka Java.

Níže popsané kroky proběhnou vždy na začátku programu z důvodu kontroly změny architektury sítě. Jestliže však uživatel nebude měnit strukturu sítě, program by v této fázi po prvotní kontrole a následném možném opravení, neměl vyhazovat další chyby.

Přidá-li však administrátor další zařízení na síť, které bude odesílat logy na Syslog server, program vypíše varování o nalezení neznámého zařízení. Zde ukončí svou činnost z důvodu, že popsané algoritmy nejsou schopny vyhodnotit, jakým způsobem byla změněna architektura a jestli vyplněná architektura stále odpovídá aktuální skutečné architektuře pracovní sítě.

4.1.1 Podrobný popis zpracování souboru Architektura sítě

První načítaný soubor popisuje detailně celou architekturu sítě. Ve vybrané složce je označen názvem „Architektura sítě.csv". Cílem souboru je definice struktury sítě, dle principů uvedených na teoretickém příkladu v předchozí kapitole.

Po načtení příslušných dat dojde prvnímu kroku kontroly, zabývající se zjištěním, zda samotný soubor obsahuje alespoň jeden řádek mimo předepsanou hlavičku. Jestliže by data obsahovala jediný řádek, lze předpokládat, že se jedná o řádek definující hlavičku celého tabulkového předpisu, a tedy soubor není vyplněn. Všechny takovéto stavy jsou označeny příslušnými chybovými hláškami popisujícími nejen místo vzniku (název souboru), ale i pravděpodobnou příčinu konkrétní chyby. Dojde-li během spuštění program k nálezu obdobných případů, ukončí po popsání chyby svůj běh.

Druhou část kontroly tvoří algoritmus, počítající počet údajů vyplněných na jednotlivých řádcích. Data jsou z daného souboru načítána po celých řádcích, v takovém formátu, v jakém byla zapsána. Jednotlivé řádky jsou od sebe odděleny příslušnými znaky, a tak program rozpozná, kde čtený řádek začíná a končí. Po načtení celého záznamu následuje uložení do inicializovaného listu bez jakýchkoliv úprav.

Poté, co jsou načtena veškerá data, jsou postupně procházeny jednotlivé záznamy ve zmíněném listu. V těchto záznamech jsou přepočítávány zadané údaje, vyplývající z jednotlivě vyplněných sloupců na daném řádku. Dojde-li k situaci, kdy řádek obsahuje méně údajů, program zahlásí chybový stav a ukončí svůj běh. V druhém případě – nachází-li se v daném řádku více údajů, program zpracuje pouze údaje, které požadoval zadat a nadbytečná data ignoruje. Zmíněná chybová situace – zadání méně údajů – je opatřena svou příslušnou hláškou pro co nejbližší možnou specifikaci problému.

Následně, co jsou načtená data strukturovaně zkontrolována, dojde ke kontrole duplicity cest. V uvedeném příkladu z předchozí kapitoly lze pozorovat, že existuje-li cesta z bodu A do bodu B, musí existovat zápis cesty i z bodu B do bodu A.

Jestliže by probíhala kontrola pouze na základě názvu rozhraní – například GigabitEthernet 0/0/0 a GigabitEthernet 0/0/0, mohlo by zde docházet k vyhledání duplicit, které spolu však nejsou souvislé. Proto algoritmus nejprve spojí daný název rozhraní s danou IP adresou síťového prvku. Tímto principem je vytvořena celá knihovna záznamů, složená z názvu rozhraní a dvou řetězců identifikujících začátek a konec cesty.

Algoritmus následně postupně načítá jednotlivé záznamy z popsané knihovny výše, a s každým záznamem vykoná následující kroky:

- vytvoření reverzní cesty,
- kontrola stávajících záznamů kvůli duplicitě,
- rozhodnutí o přidání nové cesty,
- načtení následujícího prvku.

Popsané kroky algoritmu jsou předvedeny na následujícím příkladu, vycházejícím z obrázku architektury sítě: Obr. 3.3 Příklad topologie sítě. Zadanými vstupními parametry v souboru jsou následující údaje uvedené v tabulce níže.

Program nejprve načte první záznam z tabulky, v tomto případě se bude jednat o řádek s označením R1. Záznam je složen ze dvou textových řetězců určujících začátek a konec cesty. Z řetězce definujícího konec cesty je vyjmuta IP adresa koncového zařízení, dle které je následně vyhledán identifikátor počátku reverzní cesty. Dále je vytvořena reverzní hledaná cesta, prostým přehozením řetězců, označující počátek a konec cesty.

Tab. 4.1: Příklad vyplněného souboru: Architektura sítě.

označení	**počáteční rozhraní, IP**	**konečné rozhraní, IP**
R1	Gig0/0/0 192.168.20.1	Fa0/1 192.168.20.5
S1	Fa0/1 192.168.20.5	Gig0/0/0 192.168.20.1
R2	Gig0/0/0 192.168.19.1	Gig0/0/1 192.168.20.1
R3	Gig0/0/0 192.168.17.1	Gig0/0/1 192.168.19.1
S3	Fa0/1 192.168.17.5	Gig0/0/1 192.168.17.1

Následuje postupné procházení celé tabulky záznamů a kontrola, zda zadaná IP adresa u konce cesty je shodná s načítanou IP adresou. Jestliže je nalezena shoda, je zároveň i nalezen identifikátor druhého zařízení. V opačném případě program odhalil chybu v zadání. Předpokladem pro správné fungování je fakt, že každé zařízení má alespoň jediný záznam v architektuře pro specifikaci jeho IP adresy. V uvedeném příkladě bude výstup této funkce identifikátor S1.

Po nalezení identifikátoru jsou načteny příslušné zadané záznamy z tabulky, které jsou spjaty s nalezeným údajem – v tomto případě tedy pouze druhý záznam. Ná-

sledně je zkontrolováno, zda nalezený identifikátor již obsahuje záznam o vytvořené reverzní cestě.

V případě kladné odpovědi je načten následující záznam[2]. V opačné situaci je záznam vytvořen a zapsán do struktury obsahující kompletní architekturu.

V uvedeném případě proběhne kontrola, zda síťový prvek s označením S1 obsahuje záznam: z Fa0/1 192.168.20.5 na Gig0/0/0 192.168.20.1. Záznam je již obsažen v zadané tabulce, a tak program přechází na třetí řádek, s kterým provádí obdobné kroky. Prvním připsaným záznamem bude údaj o reverzní cestě u identifikátoru R2. Zde chybí zápis reverzní cesty u R1: Gig0/0/1 192.168.20.1 na Gig0/0/0 192.168.19.1. Těmito jednotlivými kroky je program schopen vytvořit kompletní architekturu sítě.

Poslední částí kontroly je vyhledání duplicitních záznamů u jediného zařízení. Všechny síťové prvky mají určitý rozsah rozhraní (interfaců), která využívají ke komunikaci a přeposílání dotazů. Každé takovéto rozhraní má své jedinečné označení, jedná se například o: Fa0/1, Fa0/2, Fa0/3, Gig0/0/0, Gig0/0/1. Ačkoliv existují výrobci, kteří tento předpoklad nesplňují, program neočekává zadání jen vyjmenovaných interfaců, a tak uživatel může snadno odlišit v případě vytvoření VLAN sítí[3], jednotlivé rozhraní například řetězcem: Fa0/1.1.

Kontrola duplicit je postavena na logice, že v souboru s architekturou sítě je záznam cesty složen právě z IP adresy, ale i názvu rozhraní, uvedeného výše. Program tak prochází jednotlivá síťová zařízení a vyhledává, zda u něj neexistuje duplicitní záznam, shodující se v údaji o počátku cesty (rozhraní). Pokud je nalezena duplicita, uživatel je opět informován příslušnou hláškou. Tímto posledním krokem je zkontrolován první soubor.

Tab. 4.2: Příklad vytvoření duplicity záznamů v souboru Architektura sítě.

označení	**počáteční rozhraní, IP**	**konečné rozhraní, IP**
R1	Gig0/0/0 192.168.20.1	Fa0/1 192.168.20.5
R1	Gig0/0/0 192.168.20.1	Gig0/0/0 192.168.17.1

Pro možnosti reálného spuštění skriptu bylo vytvořeno uživatelské rozhraní za pomoci knihovny ArgParse. Tato knihovna dovoluje v příkazovém řádku, definici přepínačů, tedy volení budoucí funkcionality programu. V tomto případě byl přidán přepínač „control“, který umožňuje kontrolu kompletní architektury sítě. Tato kompletní kontrola je zavedena z důvodu, že program určité cesty generuje sám, dle

[2]Je zde předpoklad, že následující záznam existuje.

[3]Jedná se o rozdělení jedné logické sítě do více oddělených částí.

principů popsaných v předcházejícím textu, a tímto krokem může být uživatelem snadno odhalena chyba ve vyplnění souboru definujícího architekturu sítě.[18]

Po nastavení specifického přepínače „c“ program postupně s uživatelem prochází jednotlivá síťová rozhraní, u kterých vždy vypíše kompletní záznamy s ním spojené. Následně vždy vyčkává na kladné potvrzení správnosti daných záznamů. Jestliže uživatel nalezne nesmyslný, popřípadě chybový údaj, upozorní na tuto skutečnost program negativní odpovědí. Ten následně zde ukončí svůj běh z důvodu, že po opravě musí proběhnout opět výše zmíněné kroky.

Následující výpis ukazuje dotaz programu na správnost záznamů fiktivního zařízení.

Výpis 4.1: Ukázkový výpis při kontrole kompletní architektury s nastaveným přepínačem.

```
Dané zařízení je označeno jako: r2
gig0/0/0 192.168.19.1 gig0/0/0 192.168.20.1
gig0/0/1 192.168.19.1 gig0/0/0 192.168.21.1
Jedná se o veškeré a zároveň správně zadané cesty? y/n
```

4.1.2 Podrobný popis zpracování souboru Definice zařízení

Obdobným způsobem, jako byl načítán první soubor, je i načten druhý. Jedná se konkretně o soubor „Definice zařízení.csv". Zde nejprve proběhnou počáteční kontroly zadání. Narazí-li se na neúplný záznam, je ukončen běh programu. Paralelně také probíhá kontrola výskytu záznamů. Každý síťový prvek má v architektuře své označení – jméno, které však ve své podstatě nemusí být jedinečné. Aby mohlo docházet k jednodušší kontrole a propojování určitých souborů, je struktura programu postavena na logice, že každé zařízení má své jedinečné označení.

Další kontroly jsou poté zaměřeny na vyskytující se duplicity záznamů. Každý síťový prvek je v definici charakterizován dvěma povinnými údaji – výrobcem, označením a dvěma dobrovolnými údaji – kategorií a názvem zařízení. Dobrovolné údaje buďto program vůbec nenačítá, a ani nekontroluje, zda byly zadány – název zařízení, nebo v případě nezadání je nahradí prázdným řetězcem – kategorie.

Typ kategorie je zde zadáván z důvodu, aby mohly být odlišeny určité skupiny síťových zařízení napříč jedním výrobcem, ať už z důvodu kritického umístění, či rozpoznávání jen určité skupiny situací.

Aby nedocházelo v průběhu programu k chybám způsobeným přehozením velkých a malých písmen, jako je například označení výrobce: MikroTik/ mikrotik, převede program v průběhu načítání všechny textové řetězce na malá písmena. Tímto krokem je zabráněno nadbytečným chybám při kontrole řetězců.

Podmínka nalezení duplicity stojí na předchozím zmíněném předpokladu, že každé zařízení v souboru Architektura sítě má své jedinečné označení, a tak tedy nemůže dojít k situaci, kdy jedno zařízení má v definici zařízení dva různé záznamy, jak zobrazuje příklad níže.

Tab. 4.3: Chybový příklad vyplněného souboru: Definice zařízení.

název zařízení	**označení**	**výrobce**	**kategorie**
Router 1	R1	MikroTik	1
Switch 1	S1	Cisco	1
Router 1	R1	Cisco	1

V případě vyplnění souboru uvedenými záznamy v tabulce by program pozastavil svůj běh, a to z důvodu zapsané duplicity u prvku s označením R1. Tento záznam obsahuje dva různé údaje u výrobců, což by následně mohlo ovlivnit čtení údajů z logů a způsobit budoucí, chybové stavy. Obdobným způsobem je také vyhodnocen případ duplicitního označení, které se však shoduje pouze v údaji o výrobci a nikoliv již v údaji označujícím typ kategorie.

Poté, co jsou dokončeny veškeré popsané kontroly s negativními výsledky, může proběhnout první, základní propojení informací mezi různými soubory. Propoj proběhne mezi daty specifikujícími architekturu sítě a definici zařízení.

Nejprve jsou vyexportovány potřebné informace z množiny dat popisující kompletní architekturu sítě – označení jednotlivých zařízení, a následně je obdobným způsobem vytvořen druhý export, tentokrát však z dat detailněji popisujících jednotlivá zařízení. Následně probíhá jednoduchá kontrola, zda data z jednoho souboru jsou obsažena i v druhém a naopak. Tímto popsaným krokem je zabráněno situaci, kdyby nebyl nalezen výskyt duplicity prvků z důvodu, že se nachází pouze v jednom souboru. Program v případě nalezení takovýchto dat ukončí svůj běh a informuje uživatele o vzniklé chybě.

4.1.3 Načtení souboru Rozdělení logů pro další výrobce

Třetím načítaným souborem je specifikace vzhledu logů jednotlivých výrobců a jejich konkrétních kategorií. Cílem souboru je programu ukázat, jakým způsobem má rozdělit jednotlivé logy. Tento soubor je využit především v případech, kdy dochází ke zpracování dat výrobců mimo MikroTik a Cisco, kteří nemají v programu pevně definovanou strukturu dělení.

Nejprve i tentokrát probíhá základní propojení se souborem identifikujícím jednotlivá zařízení. Opět jsou vyhledávány duplicity dle algoritmu, který byl detailněji

popsán výše – exportovaná data z prvního souboru vyhledávají shodu v druhém souboru a následně proces probíhá inverzně. Jestliže algoritmus narazí na jednoho z výše zmíněných výrobců, nastane výjimka a pro danou jedinečnou sekvenci složenou z názvu výrobce zařízení a kategorie, kontrola duplicity neproběhne.

Zde mohou celkově nastat tři stavy, které jsou ošetřeny různými způsoby. Prvním možným výsledkem je, že soubory jsou správně vyplněny, a tak program může pokračovat bez zastavení ve svém běhu.

Druhý stav popisuje situaci, kdy ve specifikaci logů je zadán originální řetězec složený z názvu výrobce a kategorie. Tento řetězec však nemá shodu v souboru Definici zařízení. Vzhledem k tomu, že se může jednat o úpis (nechtěnou chybu), varuje program uživatele, že v souboru popisujícím rozdělení logů jsou nadbytečné informace, které budou ignorovány. Uživatel na základě vypsaných informací z této události vidí, který konkrétní výrobce a typ kategorie bude ignorován. Je tak schopen vyhodnotit, zda se jedná o chybu, či nikoliv. Program nepozastavuje po výpisu svůj běh.

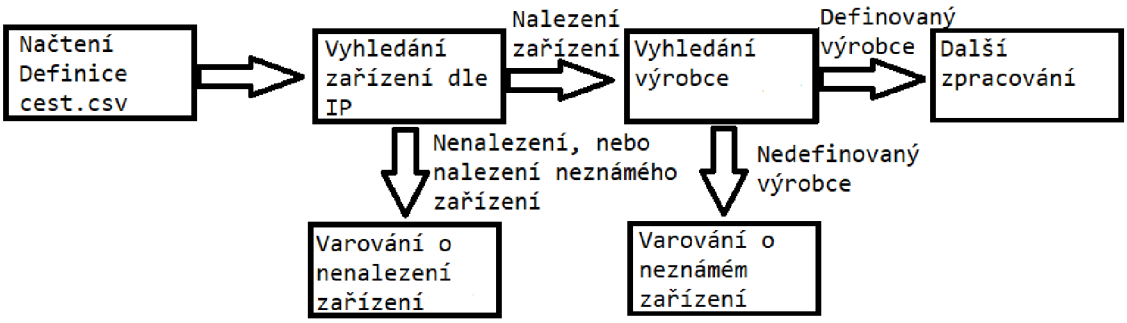
Posledním možným stavem je situace, kdy se řetězec (opět složen z označení výrobce a typu kategorie) vyskytuje pouze v souboru Definice zařízení. V tomto případě program ukončí svou činnost, nejedná-li se o zařízení Cisco nebo MikroTik, a varuje uživatele, že v architektuře sítě a definici zařízení je zmíněno zařízení, jehož rozdělení logů není uvedeno v souboru specifikujícím konkrétní rozdělení logů. Z těchto důvodů by se následně dané zařízení nemohlo zúčastnit dalšího zpracování, protože by program nebyl schopen rozdělit jejich příchozí logy na Syslog server.

4.2 Druhá fáze

Po načtení všech dříve popsaných souborů může program přejít do druhé části svého běhu. Cílem je načíst a zároveň propojit jednotlivé vyhledávané události s danými logy. Tímto způsobem může být nejen rozpoznáno kdo událost zapříčinil, ale popřípadě na ni i vykonat adekvátní reakci. Může se například jednat pouze o vypsaní chybové hlášky nebo o kontrolu konfigurace.

Rozdělení logů od firem Mikrotik a Cisco v uvedeném třetím souboru[4] slouží pouze jako ukázka, jakým způsobem by měl uživatel definovat nového výrobce. Program předpokládá, že logy těchto výrobců udržují určitý vzhled a při specifikaci vybraných logovaných problematik je schopen zajistit rozdělení dle své definované struktury. Předpokladem správné funkcionality je logování pouze vybraných událostí. Princip rozdělování je vysvětlen na příkladech uvedených níže.

[4]Jedná se o soubor s názvem: „Rozdělení logu.csv."



Obr. 4.3: Ukázka zjednodušeného postupu vyhledání logů.

Program pracuje celkově tedy s pěti soubory, kdy soubor s názvem „Definice cest.csv"je vyčleněn pro definici potřebných cest, za kterými se nachází logy jednotlivých zařízení, popřípadě soubory s konfigurací. Nejprve tak proběhne načtení dané cesty a následná kontrola, zda příslušná cesta existuje v systému, všechny chybové stavy jsou opět opatřeny chybovými hláškami.

Po vykonání prvního kroku by program měl být schopen se dostat do hlavní složky, kde jsou uloženy jednotlivé složky síťových klientů, pojmenované dle jejich IP adres, pod kterými zasílají informační logy Syslog serveru. Podrobné nastavení Syslog serveru popisuje následující podkapitola.

Aby mohl program zkontrolovat a případně upozornit uživatele na chybějící zařízení, musí nejprve načíst soubor specifikující úplnou architekturu sítě, kterou dříve vytvořil. Zde jsou načítána postupně jednotlivá zařízení a je kontrolováno, zda dané zařízení obsahuje svou složku logů v hlavní složce zařízení, na kterém program běží. V tomto případě program spoléhá na logiku, že na síťovém zařízení byl nastaven identifikátor, viditelný pro Syslog server v podobě jedné využívané IP adresy libovolného, vlastního rozhraní.

Nenalezne-li se shoda, tedy není nalezena složka zařízení pojmenovaná podle očekávaných IP adres, program upozorní uživatele na chybějící logy o konkrétním zařízení. Mohou tak být odhaleny například útoky přesměrování, čímž by na server nepřicházely zásadní informace o změně konfigurace, ale také se například může jednat o situace, kdy zařízení ještě z nějakých důvodů není nasazeno na síti, ale je již specifikováno v architektuře.

Jestliže vše proběhne v pořádku, program může pokročit s identifikací základních informací o daném zařízení, jehož logy budou načítány a rozděleny dle následující logiky.

4.2.1 Zpracování logů od firmy MikroTik

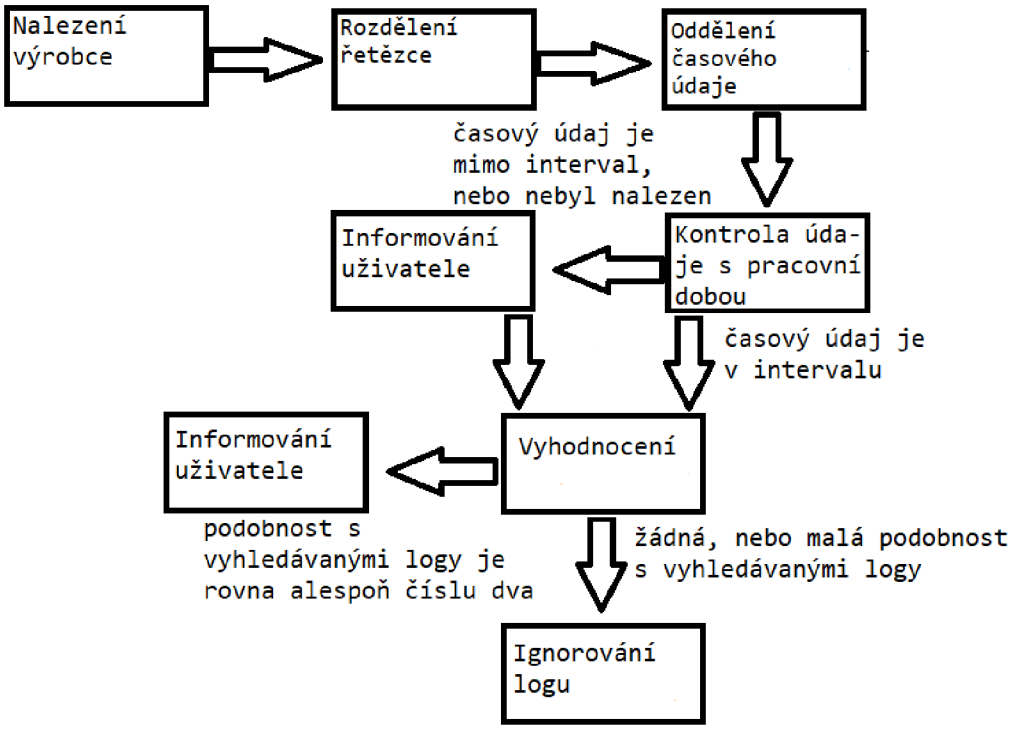
Jak uvádí předchozí kapitola popisující všeobecný návrh základního nastavení síťových zařízení od firmy MikroTik, odesílané logy udržují určitou teoretickou strukturu seskládaní. Pro identifikaci daného logu budou zejména kritické tři položky. Jedná se o:

- časovou značku,
- detailnější popis události,
- název události, ke které log patří.

Ačkoliv soubor s názvem „Rozdělení logů.csv"definuje možné rozdělení ukázkového logu, pocházejícího ze směrovače daného výrobce, pro běh programu je předpis v tomto konkrétním případě irelevantní.

Modifikuje-li tak uživatel jakýmkoliv způsobem definované rozdělení logů ze zařízení pocházejících od firmy MikroTik, popřípadě celý záznam vymaže, program bude dále rozdělovat logy dle své definované struktury popsané níže. Z tohoto dále vyplývá, že bude-li se na síti nacházet zařízení daného výrobce, v uvedeném souboru nemusí být definován princip rozdělení logu.

Následující uvedený příklad popisuje zpracování níže uvedeného logu, krok po kroku.



Obr. 4.4: Zjednodušený příklad zpracování logu.

Předpokladem celé funkcionality je situace, kdy Syslog server nevytváří neustále nové soubory pro nově příchozí logy, ale pouze do příslušných již vytvořených souborů připisuje další přicházející informace od zařízení, které následně rozděluje dle témat k nim přidružených[5].

Dalším předpokladem je také fakt, že výsledný program není nastaven pro neustálý běh, ale pouze pro cyklický - spouštění například v pravidelných intervalech, popřípadě spouštění za libovolný časový úsek. Z těchto důvodů je potřeba pracovat s časovými značkami uvedenými v jednotlivých lozích. K nastavení časového okamžiku určujícího kolik hodin zpětně mají být logy procházeny, aby nedocházelo zbytečně k situacím, kdy program bude procházet tisíce již jednou projitých a vyhodnocených záznamů, slouží dobrovolný přepínač, který může být nastaven při samotném spouštění skriptu v příkazové řádce.

Program očekává zadání číselného údaje v celých hodinách. Nastaví-li uživatel časový údaj přepínače na nulu, bude procházen celý soubor logů od začátku po konec. V případě nenastavení žádného údaje program vyhledá logy dvě hodiny zpětně od aktuálního systémového času[6]. Nastavení přepínače bude platné pro všechny procházené soubory[7].

```
2023-03-07T12:55:09.699250+01:00 router.lan
system,info changed script settings by admin
```

Obr. 4.5: Příklad vygenerovaného logu pocházejícího ze síťového zařízení od firmy MikroTik.

Rozdělení řetězce

Stěžejní součástí zpracování je prvotní oddělení časové značky od celého logu a poté oddělení popisu o samotné události, která zapříčinila vytvoření logu. Tyto informace jsou rozděleny napříč různými síťovými zařízeními firmy MikroTik, vždy znakem mezery. Ta tudíž může být využita k rozkouskování celého logu na menší podčásti, s kterými program dále pracuje. Vytvořený algoritmus pro zpracování předpokládá i do budoucna zachování této struktury.

[5]Tento předpoklad platí pouze z hlediska krátkodobého. V případě pohledu z dlouhodobého hlediska program počítá i se situací, kdy kupříkladu jednou týdně budou soubory přesunuty do úložiště a poté se vytvoří nové soubory pro přicházející logy. Postup v takovémto případě je popsán dále v textu.

[6]Systémovým časem je v tomto případě myšlen časový údaj v okamžiku spuštění, který bude převzat z daného zařízení, na němž program běží.

[7]Je tak třeba vyřešit správnou synchronizaci časů na jednotlivých sítových zařízeních.

Program využije k rozdělení logu funkci, jejíž cílem je postupné procházení zadaného řetězce znak po znaku. Narazí-li na znak shodující se s hledaným znakem, oddělí první část řetězce od zbytku původního řetězce a pokračuje s dalším vyhledáváním nad dosud neprojitými znaky. Výsledkem je celý list podřetězců, jejichž součástí není vyhledávaný znak.

Na uvedeném příkladu výpisu logu výše by tedy po aplikaci popsané funkce vznikl list řetězců neobsahující mezery. List by byl složen z osmi položek:

- 2023-03-07T12:55:09.699250+01:00,
- router.lan,
- system,info,
- changed,
- script,
- settings,
- by,
- admin.

Poté, co byla aplikována funkce rozdělení, se teoreticky programu podařilo oddělit časovou značku od celého logu, s kterou může být dále pracováno. Samotné oddělení spočívá v logice, že zasílané informace vždy začínají časovým údajem, který Syslog server poupraví dle svého nastavení, a tak programu stačí pouze načíst první údaj v listu.

V tomto kroku se následující funkce programu rozpadají do tří možností.

Je-li nastaven přepínač určující kolik hodin zpětně mají být události vyhodnoceny, jsou vytvořeny horní a dolní hranice tohoto časového intervalu. Do horní hranice je převzat aktuální systémový čas, následně je od něj odečteno číslo nastavené v přepínači. Tímto způsobem je vytvořena i spodní hranice intervalu. Poté už jen proběhne kontrola, zda oddělená časová značka zapadá do vytvořených hranic. V případě pozitivní odpovědi je celý log dále zpracován, v druhém případě je zahozen.

Příkladem uvedeného výpočtu výše může být nastavení přepínače na 48 hodin. V přepočtu se jedná o dva dny. Horní hranice bude převzata jako aktuální systémový čas, například 10 hodin dopoledne, sedmý den v roce. V tomto případě tak program bude vyhledávat logy pro zpracování dva dny dozadu. To znamená, že spodní hranici bude tvořit údaj: 10 hodin dopoledne pátý den v roce.

Obdobný způsob je využit i v případě nenastavení žádného časového údaje ve zmíněném přepínači. Tentokrát však pro vytvoření spodní hranice intervalu je využit časový interval dvou hodin.

Posledním možným případem je situace, kdy uživatel nastaví nulový časový údaj. V tomto případě nejsou časové značky jakýmkoliv způsobem zkoumány a logy jsou vždy zpracovány.

Po zpracování časové značky a rozhodnutí, zda log zapadá do časových hranic, tudíž nebyl pravděpodobně ještě vyhodnocen, může program přejít k vyhodnocení, zda se bude nadále zpracovávat, či nikoliv. Kód programu je uzpůsoben v případě MikroTiku ke zpracování logů týkajících se pouze témat ohledně změn na samotném systému a informativních změn. Tyto změny jsou schopny upozornit na situace, kdy dojde například k vypnutí rozhraní, vytvoření nebo smazání uživatele, vytvoření nového skriptu, úpravě stávajících skriptů, ale zaznamenávají i kritické změny postihující přenastavení času, změnu časové zóny a další.

Na základě příkladu výčtu logovacích schopností této skupiny byla uvedená témata vybrána jako dostačující pro specifické cíle programu. Vytvořená funkce prochází postupně ve smyčce jednotlivé položky výše uvedeného rozděleného logu v listu a vyhledává shodu aktuálně načteného prvku s řetězcem označujícím název události. V této fázi nemůže nastat negativní odpověď – tedy nenalezení žádné shody, protože Syslog server rozděluje logy dle témat a program je určen k vyhledání pouze konkrétního logovacího souboru, obsahujícího jen logy spjaté s daným tématem. Přesto je zde ošetřena i situace, kdy jsou programu předložena data bez výskytu této informace.

Poté, co je nalezena podoba řetězců, je odpočítána pozice začátku textového řetězce popisující detailněji nastalou událost. K odpočtu jsou využity informace o poloze výše načteného identifikátoru a znalost jeho délky. Každý znak v řetězci zabírá určitou číselnou pozici začínající od číslice nula. V uvedeném příkladu logu je na první pozici: číslice 2, na druhé: číslice 0, na třetí: 2 a na čtvrté: 3. Obdobným způsobem tak lze očíslovat i zbylé znaky tohoto logu.

Program využije znalost ohledně pozice začátku řetězce, označující druh nastalé události, který byl získán za pomocí funkce, vracející index aktuálního znaku, vzhledem k původnímu řetězci a k němu připočítá svou délku, společně s číslem jedna, označující znak mezery. Tímto postupem je teoreticky vypočítán začátek detailnějšího popisu o nastalé události.

V situaci, kdyby byl mezi zpracovanými podřetězci například další řetězec označující závažnost situace, nebude ovlivněna funkcionalita, nevyskytne-li se v hledaných klíčových slovech, jejichž popis je v následujících odstavcích.

V uvedeném příkladu logu je postupným procházením od konce seznamu získána pozice 42. K ní je následně přičtena ještě délka řetězce určující druh události „system, info“, společně s mezerou, tedy číslo 12. Po aplikaci operace sčítání je zjištěno, že detailnější popis události začíná na čísle 55. Hledaný textový řetězec je následně odpojen od zbytku logu a společně již s dříve odtrženou značkou vytvoří záznam v knihovně událostí. Opakováním výše zmíněného postupu je knihovna s událostmi postupně plněna zpracovanými logy.

Zpracování vytvořené knihovny událostí

Aby však uložené záznamy mohly být dále rozpoznány a popřípadě na ně mohla být vykonána adekvátní reakce, je potřeba načíst poslední soubor, který by měl být uživatelem modifikován. Jedná se o soubor s názvem „Předpis logů.csv", složeného z následujících prvků:

- zařízení a kategorie – každé specifické zařízení může mít pozměněné logovací hlášky, nebo může být žádoucí na zařízeních určitých skupin vyhledávat pouze určité situace. Z těchto důvodů musí být každý záznam spjat s konkrétním výrobcem a kategorií,
- akce – umožňuje programu příslušně reagovat odpovídajícími hláškami, či vykonávat adekvátní reakce. Aktuálně je program schopen vykonat celkem čtyři možné procesy, definované na základě písmen v příslušném sloupci. Jedná se o písmena:
  - i – informativní, označuje události, které nejsou zcela běžné a mohly by být způsobeny nekalou činností administrátora. Je to například úprava stávajících logovacích pravidel. Uživatel musí sám posoudit dopad těchto událostí na systém,
  - w – varování, popisuje události, které mohou mít kritické dopady na systém a program nemůže zajistit, že vykonané změny nebudou mít zároveň i dopady na výsledky programu. Jedná se například o změny způsobené přenastavením časové zóny, změnou času,
  - c – kontrola zálohy konfigurace, proběhne v případě kritických změn na zařízení. Popis postupné kontroly souboru se zálohami je popsán níže. Příkladem využití této události může být například při nalezení logu definující přidání nového uživatele, změně stávajících skriptu, či přidání nového skriptu,
  - a – kontrola architektury, v případě, dojde-li k vypnutí rozhraní, je program schopen vyhodnotit, které uzly na základě tohoto výpadku byly postihnuty. V podstatě se jedná o počátek a konec cesty. Ty jsou následně vypsány do terminálu.
- klíčové slovo – jedná se o rozhodující slova specifikující, ke které očekávané události bude log přiřazen. Soubor obsahuje již určité předvyplněné události, nad kterými byla celá funkcionalita programu testována.

Před načtením samotného souboru s klíčovými slovy je zkontrolováno, zda knihovna obsahující záznamy, které mají být dále procházeny, obsahuje alespoň jediný záznam. V případě, že je její velikost nulová, je vypsáno varování, že dané síťové zařízení za nastavený časový interval neodeslalo žádné záznamy o události. Následně je ještě vypsán poslední log zaznamenaný z tohoto zařízení. Uživatel tak musí sám

posoudit, zda se jedná o normální chování, nebo došlo ke kritické změně – například vypnutí rozhraní. Program po vypsání varování, neukončí svou činnost a pokračuje ve zpracování dalších zařízení.

Pokud i tato část pro kontrolované síťové zařízení skončí bez chybových hlášek, přejdou vybrané logy spadající do časového intervalu k procesu vyhodnocení. Vyhodnocení probíhá na základě shody klíčových slov z posledního načítaného souboru[8] s částí popisující nastalé události. Pod pojmem klíčová slova, jsou myšlena označení, která popisují danou událost. Na základě vyhodnocení celkové podoby mezi zadaným řetězcem a zaslaným logem je pak program schopen přiřadit událost k danému předpisu a vykonat adekvátní reakci. Uživatel tato klíčová slova sám definuje a mění.

Soubor navíc umožňuje právě na základě specifikace údaje s kategorií rozpoznání jen určitých, definovaných událostí. To znamená, že pokud bude běžné pro kategorii vnitřních směrovačů změna logovacích pravidel, ale naopak pro hraniční směrovače stejného výrobce tato aktivita nebude běžná, je možné tyto skupiny rozdělit dle identifikátoru kategorie.

Následně stačí definovat vybraného výrobce, kategorii a jednotlivé události, které u této skupiny mají být rozpoznávány. Platí zde pravidlo, že na jednom řádku může být specifikována pouze jedna událost, jeden typ a jeden výrobce, jak ukazuje příklad vyplnění níže.

Tab. 4.4: Příklad vyplněného souboru: Předpis logů.

výrobce	kategorie	reakce	klíčová slova		
MikroTik	1	c	new	script	scheduled
MikroTik	1	c	change	script	settings
MikroTik	1	c	change script settings scheduled		
MikroTik	2	c	new script scheduled		

Uvedený příklad zápisu logovaných událostí zobrazuje celkem dva způsoby zadání. V prvním případě se jedná o oddělení jednotlivých slov do příslušných sloupců a v druhém případě o zadání celého vyhledávaného řetězce do jediné buňky. Algoritmus pro načítání tohoto souboru je přizpůsoben k vyhledávání mezer mezi jednotlivými slovy v načtené buňce, a tedy předpokládá využití obou způsobů. Vyskytne-li se tak mezi slovy mezera, program tato slova oddělí a prochází dalšími kroky identicky jako v prvním případě.

[8]Jedná se o soubor: „Předpis logů.csv".

Popis způsobu vyhodnocení celkové shody

Po úspěšném načtení popsaného souboru lze přejít k samotnému posuzování celkové podobnosti. Nejprve je načten první log z knihovny, která byla vytvořena výše a obsahuje detailnější popis jednotlivých, nastalých událostí. Následně je k tomuto logu vyhledána skupina klíčových slov, definovaných uživatelem, shodující se s názvem výrobce a příslušnou kategorií. V dalším kroku stačí pouze načítat jednotlivé klíčové události a hledat shodu podřetězců v popisu nastalé události. Jestliže nastane shoda daného podřetězce se zaslaným logem, bude k počítadlu vybrané události přičteno číslo jedna, tak jak ukazuje příklad níže.

V uvedené tabulce, ukazující vyplnění posledního zmiňovaného souboru, jsou specifikovány celkem čtyři události. Jedná se o události specifikující nastavení opakujícího skriptu, změny v nastavení samotného skriptu nebo časových intervalů opakování. Také se zde vyskytují celkem dvě kategorie. Jedná se o kategorie označené čísly jedna a dva.

Při zpracování ukázkového logu – „Obr. 4.4: Příklad vygenerovaného logu pocházejícího ze síťového zařízení od firmy MikroTik", je nejprve nutné získat, na základě znalosti IP adresy a propojení souborů, údaje o výrobci společně s danou kategorií. Po získání těchto identifikátorů, v tomto případě výrobce MikroTik s definovanou kategorií číslem jedna, mohou být načteny ze souboru „Předpis logů.csv", příslušné rozpoznávané události. Není-li nalezen k dané dvojici – výrobce a kategorie – žádný předpis, je vyhozena chybová hláška a program ukončí svou činnost. Jedná se o kritickou chybu, která může například na základě překlepu způsobit nerozpoznání závažných událostí. Je-li v souboru pod danou dvojicí uložen alespoň jediný záznam, je pokračováno v následujících krocích.

Poté jsou již pouze procházeny postupně klíčové události a probíhá hodnocení celkové shody definované události s procházeným logem. V uvedeném příkladu je načten první předpis vypovídající o naplánování spouštění nově přidaného skriptu, který je dále rozdělen dle výše popsané logiky do listu ve tvaru:

- new,
- script,
- scheduled.

Tímto krokem jsou získány veškeré údaje potřebné k vyhodnocení celkové podobnosti. Ta je posuzována pomocí logiky spočívající v procházení jednotlivých položek listu a následném vyhledávání shody v detailnějším popisu logu. Celková podobnost uvedeného příkladu s výše vypsaným listem by byla rovna číslu jedna. Výpočet byl proveden na základě shody jediného slova nalezeného v předpisech. Jedná se o slovo skript. Obdobným způsobem je všem příslušným předpisům vypočítána vlastní hodnota podobnosti.

Popsaným způsobem byl program schopen vyhodnotit podobnost mezi jednotlivými událostmi. Algoritmus tak teoreticky neovlivní, bude-li log obsahovat informace navíc, jako jsou například údaje o závažnosti, nevyskytnou-li se mezi klíčovými slovy. V případě výskytu by došlo k nesprávné inkrementaci proměnné a na základě vyššího čísla by mohlo dojít k celkovému nesprávnému vyhodnocení – přidružení logu ke špatné události.

Z knihovny, kde jsou uložené jednotlivé předpisy, společně se svými hodnotami pravděpodobnosti, jsou vybrány pouze ty s nejvyšší hodnotou[9]. Nyní se kroky programu rozpadají celkem do čtyř větví. V prvním, nejjednodušším případě existuje pouze jediný předpis události, který souhlasí s daným logem. Program tak vypíše očekávané informace o zpracování a následně vykoná požadovanou akci popsanou výše.

V druhém případě se jedná o situaci, kdy pod nejvyšší hodnotou je několik předpisů událostí. To znamená, že program našel identický počet shod ve více než jednom definovaném předpisu. Tuto situaci zobrazuje příklad výše, kdy celková shoda tří klíčových slov nastala u předpisu na druhém a třetím řádku vyplněné tabulky s názvem: „Tab. 4.5: Příklad vyplněného souboru: Předpis logů". Shodnými slovy byly: change, script, settings.

Program nyní zkontroluje, zda se u některého z vybraných logů vyskytují všechna klíčová slova uvedeného předpisu. V případě pozitivní odpovědi je nalezen předpis logované události[10].

Jestliže nastane shoda opět ve více předpisech – tedy více celých předpisů je kompletně nalezeno v záznamu logu, program vypíše chybovou hlášku, že nebyl schopen rozpoznat danou událost a vykonat příslušnou akci. Zároveň varuje, že dané předpisy jsou buď identické nebo příliš podobné a uživatel by měl tyto možné události více specifikovat.

Pokud nebude ani jeden předpis kompletně obsažen v logované události, program opět vypíše chybovou hlášku o neschopnosti přiřazení pouze jediné události, jak tomu bylo i v předchozím případě. Uživatel tak musí sám rozhodnout, ke které události by měl být zařazen a popřípadě na něj vykonat adekvátní reakci.

Poslední možnou situací je stav, kdy log nesplní základní podmínku – shoda samotného popisu události s klíčovými slovy nebude větší než číslo jedna[11]. V tomto případě program přejde log bez jakéhokoliv dalšího zpracování. Dle čísla podobnosti lze odhadovat, že se jedná především o záznamy nastalých událostí, u kterých nejsou uživatelem definovány konkrétní předpisy, a tak není žádoucí, aby program upozorňoval na jejich výskyt.

[9] Jedná se o nejvyšší číslo.

[10] Tímto způsobem by byl vyřešen uvedený příklad výše.

[11] Pro splnění podmínky musí být shoda rovna alespoň číslu dva či více.

Popis vykonání reakce

Těmito popsanými kroky byl příkladový log přiřazen k události na řádku dva. Nyní je potřeba vykonat adekvátní reakci, kterou je v tomto případě kontrola změn v záloze konfigurace.

Program při vyhodnocení, zda se jedná o identické soubory, pracuje s dvěma zálohami: vlastní a nahranou samotným uživatelem. Mezi nimi jsou následně přes funkci porovnání vyhledávány jakékoliv změny. Jestliže jsou nějaké nalezeny, vypíší se do terminálu a uživatel musí na závěr sám posoudit možné kritické dopady na zařízení.

Nahrané jednotlivé konfigurace jsou načítány ze složky, kterou uživatel specifikuje v souboru „Definice cest.csv". Zde také k jednotlivým síťovým zařízením, jež jsou oddělena na základě identifikátorů IP adresy, kategorie a výrobce, přiřazuje označení samotné hledané zálohy – jejího očekávaného názvu. Může se jednat například o řetězec složený z IP adresy a využitého označení v architektuře sítě.

Algoritmus dále počítá se situací, kdy na jednotlivých zařízeních budou v pravidelných intervalech před spuštěním samotného běhu programu vytvářeny textové zálohy, ať už manuálně, či za pomoci skriptů. Nové zálohy vždy přepíšou již konkrétní uživatelem dříve definovanou zálohu v očekávané složce plynoucí z „Definice cest.csv". Tímto krokem však jednotlivé starší, neshodné zálohy nebudou ztraceny, protože vedlejší funkcí programu je uchování těchto záloh se systémovým časem běhu.

Při prvním běhu programu tak nejprve algoritmus zkontroluje, zda má již nahrané referenční zálohy pro možnou kontrolu. V této části se následující kroky rozpadají na dva možné stavy:

V načítané složce neexistuje doposud záloha vytvořená programem, a tak nemá v tomto případě konfiguraci pro porovnání. Z tohoto důvodu je výsledkem vzniklého stavu vytvoření referenční konfigurace a následné uložení do složky programu z nově příchozí zálohy nastavení. Uživatel je o vzniklé situaci informován, jak zobrazuje ukázkový výpis z terminálu níže. Jednotlivé soubory program pojmenovává dle jedinečného řetězce složeného z IP adresy využité pro komunikaci se Syslog serverem.

Výpis 4.2: Ukázkový výpis terminálu při informování o uložení zálohy.

```
Záloha konfigurace s nazvem:
192.168.22.1 2023.04.25_10.40.44 byla uložena do složky
zálohy.
```

Druhým případem je stav, kdy program nalezne ve své složce již dříve jím vytvořenou a následně uloženou očekávanou konfiguraci. Nyní jsou splněny veškeré očekávané vstupy do funkce zajišťující porovnání. Použitá funkce využívá modul

difflib, sloužící k porovnání dvou zadaných vstupů. Modul je schopen rozpoznat změny v jednom souboru vzhledem k druhému. Tímto postupem je možné rozeznat jak přidané, tak umazané řádky, slova konfigurace. Nalezené výsledky jsou vypsány do terminálu.

V tomto kroku program porovnal a nalezl potenciální změny v konfiguračním souboru daného zařízení. Ačkoliv se referenční společně se změněnou zálohou nastavení, nemusí být vždy identické, může se jednat o legitimní změny. Z těchto důvodů program umožňuje změnu referenční zálohy jednotlivých zařízení[12].

S touto funkcionalitou je spjat další přepínač s názvem „automatic“, jehož cílem je programu sdělit, zda má být spuštěn v interaktivním módu, či v automatickém. Jestliže nebude přepínač nastaven, poběží skript v automatickém módu a nebude očekávat žádnou interakci s uživatelem. Pokud tak za běhu nalezne neshody v nastavení, uloží tuto potenciálně správnou konfiguraci do složky s názvem „Zálohy“. Soubor pojmenuje dle aktuálního systémového času s kombinací IP adresy aktuálně kontrolovaného síťového zařízení. Daná IP adresa je využita pro komunikaci se Syslog serverem.

Jestliže je nastaven interaktivní mód, kód programu očekává určitou komunikaci s uživatelem. Jsou-li nalezeny jakékoliv změny v novém nastavení oproti vypovídající referenční záloze, je vypsán textový řetězec dotazující se, zda se jedná o legitimní změny a nově příchozí konfigurace má nahradit aktuální, referenční. V případě kladné odpovědi je starší referenční soubor daného zařízení přepsán jeho novější verzí. Je-li odpověď uživatelem negativní, je pouze informován, že nově příchozí záloha bude uložena do složky určené k uschování záloh identickým způsobem popsaným výše.

Následující výpis terminálu zobrazuje dotázání programu, zda má přehrát svou referenční zálohu, protože se neshoduje s aktuální, nahranou zálohou.

Výpis 4.3: Ukázkový výpis v interaktivním módu programu.

```
-# software id = gsfd-yjdf
+# software id = gsfd-yjdffff
 #
 # model = rbd52g-5hacd2hnd

 Chcete přehrát aktuálně příchozí konfiguraci s IP:
 192.168.20.1 jako novou refenční zálohu? y/n
```

Uživatel vytvořené soubory s modifikovaným nastavením oproti referenčním hodnotám spravuje sám. Řeší tedy například jak dlouho budou zálohy uchovávané. Jestliže by tato funkcionalita byla obsažena v kódu, zaneslo by se tímto krokem

[12]Jedná se o zálohu, s kterou je nově příchozí záloha kontrolována.

určité bezpečnostní riziko. Správci by tak stačilo znát pouze definované nastavení programu, následně by vykonal nelegitimní změny, které by zakryl za definovaný počet legitimních změn. V praxi by to znamenalo například nastavení programu pouze k uchování posledních pěti změněných souborů. Administrátor by nejprve vykonal nelegitimní změnu, po spuštění programu by záloha byla uložena do změněných záloh. Následně by zařízení vrátil do původního nastavení. Posledním krokem by bylo vykonání dalších pěti legitimních změn, čímž by záloha o nelegitimní změně byla smazána. Při kontrole logů z jednotlivých zařízení by pak uživatel nebyl informován o popsané situaci výše.

4.2.2 Zpracování logů od firmy Cisco

Ačkoliv vzhled logu firmy Cisco se liší od logu prvního zpracovávaného výrobce, princip rozdělení zůstává identický. Program zachovává strukturu algoritmů, které jsou uvedeny v předcházejícím příkladu. Ty pouze drobně upravuje. Jedná se o kroky vyplývající z obrázku „Obr. 4.4: Zjednodušený příklad zpracování logu":

- rozdělení řetězce na základě mezery,
- oddělení časové značky,
- vyhledání odkud začíná detailní popis události,
- kontrola shody s klíčovými slovy,
- vyhodnocení.

Uvedené kroky mohou být implikovány bez větších úprav na základě využití Rsyslogu, který například řeší odlišnosti zadávání časových značek výrobců MikroTik a Cisco, jenž byl uveden v předcházející kapitole. Rsyslog do logu přidává svou časovou značku. Tímto způsobem je udržen jednotný formát napříč různými výrobci.

```
2023-03-07T13:59:10.364014+01:00 %LINK-3-UPDOWN: Interface GigabitEthernet0/0/0,
changed state to down
```

Obr. 4.6: Příklad logu ze síťového zařízení firmy Cisco.

První změna nastává v kroku vyhledání začátku detailního popisu události, kde jsou tentokrát specifikovány události typu sys a line. Program podobným způsobem prochází vytvořený list neobsahující mezery a vyhledává zmíněné typy událostí. Jedná se o řetězec „sys“, událost typicky označuje systémovou činnost, nebo „link“, druhá možnost označuje změnu stavu linky či portu. Vybrané typy by měly být dostatečné pro specifické cíle programu a zároveň ekvivalentní záznamu události ze síťových zařízení firmy MikroTik.

Na základě výše popsané modifikace zároveň vyplývá i druhá změna, že program v tomto případě nebude mapovat soubory označené „system, info“, ale bude vyhledávat nové názvy „sys.log“, společně s označením „link.log“. Je už na vlastním uvážení uživatele, zda se rozhodne logy zapisovat pouze do jednoho souboru s očekávaným názvem, nebo zvolí rozdělení příchozích logů do dvou souborů.

Na základě popsané funkcionality tak v tomto případě dojde k rozdělení uvedeného logu výše na podřetězce:

- 2023-03-07T13:59:10.364014+01:00,
- %LINK-3-UPDOWN:,
- Interface,
- GigabitEthernet0/0/0,
- change,
- state,
- to,
- down.

Poté program pokračuje obdobným způsobem, jako tomu bylo u zpracování logů od firmy MikroTik. Zjistí velikost řetězce, ve kterém se nachází hledaný typ, připočítá jej k aktuální velikosti a tím je zjištěn začátek detailního popisu. Program tak bude ignorovat řetězce označující závažnost zachycené události nebo jednoznačný identifikátor zprávy. Následně již program zpracovává informace identickým způsobem jako u předcházejícího výrobce. Propočítá shody klíčových slov, vykoná adekvátní reakce a následně informuje uživatele o výsledcích.

Zásadním rozdílem mezi zařízeními firem Cisco a MikroTik je především způsob logování. Síťové prvky od výrobce MikroTik byly schopny kupříkladu rozeznat přidání nového skriptu, avšak z druhého zařízení od firmy Cisco byl, v případě přidání obdoby skriptu, pouze zaslán informační log o změně. Z těchto důvodů bude mít každý výrobce v souboru „Předpis logu.csv“ svou skupinu rozpoznávaných logů. Tyto skupiny se tak na první pohled mohou lišit, ale odchylky jsou způsobeny právě ve způsobu logování vybraných událostí.

4.2.3 Zpracování logů dalších výrobců

Posledním krokem, jehož funkcionalita musí být v programu ošetřena, je zpracování třetích výrobců, kteří taktéž vyrábějí síťové prvky, a tak mohou být na síti využity. Mezi nejznámější příklady patří Juniper Networks, Hewlett Packard Enterprise a další. Program bude očekávat v souboru „Rozdělení logu.csv“ zadání vyhledávaných typů události. Ve zmíněném souboru bude jako příklad zadání specifikace typu u zařízení od firmy MikroTik a Cisco. Jedná se pouze o ukázkové zadání. V případě modifikace informací u těchto výrobců nebudou mít změny žádný vliv na běh.

Uživatel by měl sám dle výše popsaného textu zvolit obdobné úrovně logování, aby byl program schopen dosáhnout sepsaných cílů, tedy odhalení potenciálně nebezpečné aktivity administrátorů. Poté musí ještě specifikovat jednotlivé rozpoznávané logy a jejich reakce na nalezení shody. Specifikace souboru logů se budou lišit napříč výrobci z důvodu, který byl popsán výše. Zpracování ukázkového logu třetího výrobce je zobrazen v kapitole „4.4.2 Výsledky programu na vytvořené testovací síti", kde je popsána i potřebná modifikace výše uvedených souborů.

Tab. 4.5: Příklad vyplněného souboru: Architektura sítě.

výrobce	kategorie	klíčová slova	
MikroTik	1	system,info	
Cisco	1	sys	link

4.3 Další funkcionality programu

V předchozí kapitole s názvem Praktická část, byly navrženy určité redundantní funkce, které nejsou kriticky spjaty s hlavní funkcionalitou programu. Slouží k další kontrole, pro co možná nejužší specifikaci normálního chování uživatelů. První takto navrženou funkcionalitou byla kontrola a specifikace pracovního času jednotlivých administrátorů na síti. Z těchto důvodů byl k programu přidán další soubor s názvem „Hodiny uživatelů.csv“. Cílem tohoto souboru je vydefinování jména administrátora, pod nímž vystupuje v lozích, a jeho běžné pracovní doby, jak zobrazuje příklad níže.

Tab. 4.6: Příklad vyplněného souboru: Hodiny uživatelů.

jméno uživatele	**začátek intervalu**	**konec intervalu**
admin	10:00	14:00
user1	9:00	15:00
user2	0:00	20:00

Program nejprve načte výše zmíněný soubor a vyhledá v detailnějším popisu události řetězec odpovídající jménu jakéhokoliv administrátora. Jestliže tento řetězec nebude nalezen, bude vypsáno varování pro uživatele, že daný log neobsahuje žádnou shodu s jmény uvedenými v souboru. Uživatel sám musí posoudit, zda se samotné jméno v logu nevyskytuje, či jej pouze zapomněl definovat, nebo se jedná

o nově přidaného uživatele, který doposud není definován. Program zde nezastaví svou činnost.

V případě nalezení shody jsou na základě jména vyčteny příslušné údaje – tedy je vytvořen časový interval, určující běžnou pracovní dobu. Poté je převzat čas z příslušného logu a následně je zkontrolováno, zda čas z logu zapadá do vytvořeného intervalu. Je-li odpověď kladná, program pokračuje ve své činnosti. V druhém případě vypíše uživateli varování obsahující: jméno, nastavení pracovní doby, čas pocházející z logu události. Uživatel musí opět sám posoudit, zda se jednalo o legitimní změny, či o neobvyklé chování.

Výpis 4.4: Ukázkový výpis v případě neshody časového údaje.

```
VAROVÁNÍ!!!!!
Časová značka není v rozmezí normální pracovní doby uživatele.
Normální časová doba práce:
8:00 16:00
Časová značka:
2023-03-26t03:51:10.364014+01:00
```

4.4 Realizace testovací sítě

Aby mohla být otestována funkčnost a reálné schopnosti navrhnuté aplikace, je potřeba zprovoznit alespoň jednoduchou síť, složenou ze Syslog serveru a několika síťových zařízení, které budou zasílat požadované logy o událostech. Na základě těchto dat pak bude vyvozeno chování aplikace při provozu. Program bude předpokládat, že nastavení Syslog serveru bude obdobné jako v níže popsaném postupu. Jako testovací síťové zařízení byly vybrány směrovače od firem Cisco, MikroTik a Hewlett-Packard Development Company[13].

4.4.1 Nastavení Syslog serveru

Z důvodu jednoduchosti a omezených prostředků byla pro zprovoznění Syslog serveru vybrána virtuální stanice, na kterou byl nahrán operační systém Ubuntu. Jedná se o otevřený operační systém[14], jehož základ tvoří linuxové jádro. Ubuntu vychází z distribuce Debian Linux a je vyvíjen oficiálně společností Canonical. Ta nabízí celkem tři možná řešení, lišící se ve funkcionalitě. Jedná se o verzi pro:

- Desktopové zařízení – navržena pro běžné uživatele,
- Servery – uzpůsobena pro využití v podnikových sítích, datacentrech,

[13]Dále jen HP.

[14]U otevřeného systému jsou zdrojové kódy volně dostupné.

- Cloudy – navržena pro běh aplikací v cloudu.[19]

Pro zprovoznění serveru byla nainstalována nejjednodušší verze pro desktopové zařízení z důvodu, že pro dosažení výše definovaného cíle je řešení dostačující.

Nejprve je potřeba aktualizace verze balíčků, aby instalace serveru mohla proběhnout nad aktuálními, podporovanými balíčky a nemohlo dojít k chybám způsobeným nekompatibilitou.

```
sudo apt update
```

Poté může být nainstalován a spuštěn samotný balíček Syslog za pomocí příkazů v terminálu:

```
sudo apt-get install rsyslog -y
sudo systemctl enable --now rsyslog
```

Ve výše uvedeném nastavení probíhá instalace balíčku RSyslog namísto obyčejného Syslog serveru. Jedná se o obdobu Syslog démona, jenž má v určitých směrech rozšířenou funkcionalitu oproti základní verzi. Rsyslog rozšiřuje například možnosti: filtrace na základě typu zpráv, konverzi logů, nebo IP adres, dále dovoluje využití protokolu TCP pro zajištění bezpečnějšího provozu[15].[20]

Po úspěšném spuštění serveru, posledním výše zmíněným příkazem, je potřeba provést konfigurační změny v souboru „rsyslog.conf“. Cílem je úprava konfigurace – tedy nastavení a přizpůsobení serveru daným požadavkům.

V tomto případě byl vytvořen záznam požadavku pro zachytávání určité podsítě IP adres a jejich následného uložení do souboru. Zde bylo využito definované šablony pro vytvoření jednotlivých složek dle příchozích IP adres[16]. Potřebná modifikace souboru je uvedena v řádcích níže.

Uvedenými příkazy dojde k nastavení příjmu syslog zpráv pro protokoly UDP běžícího na portu 514. Obdobným způsobem by pak mohl být nastaven provoz s využitím protokolu TCP.

```
#Provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")
```

Nyní je potřeba nastavit specifikaci konkrétních IP adres, podsítí nebo sítí, od kterých má Syslog server přijímat záznamy. V případě velkých sítí, kde administrátor nemá přehled o všech možných zařízeních, je vhodnější varianta vypsání pouze určitých IP adres[17]. V tomto konkrétním případě však plně postačí specifikace dané

[15] V základní verzi je podporován pouze protokol UDP.

[16] Jedna IP adresa má vždy svůj soubor logů.

[17] Především v případech využití sítí s IPv6.

podsítě, z důvodu, že se jedná o malou síť, a tak existuje přehled o připojených zařízeních. Specifikace může probíhat ve dvou variantách, a to filtrací adres na základě využitého IP protokolu IPv4 nebo IPv6.

V daném příkladu je vyobrazeno nastavení pro příjem pocházející z jakékoliv adresy protokolu IPv6, ale pouze dané podsítě IPv4. V praxi by se mohlo jednat o potenciálně nebezpečné nastavení z důvodu otevření portu a povolení komunikace se všemi IPv6 adresami. V tomto konkrétním případě se však jedná pouze o příklad, který se snaží poukázat na rozšířené možnosti nastavení.

```
$AllowedSender UDP, 192.168.38.0/24, [::0]/128
```

Posledními změnami v nastavení bude vytvoření následujícího principu chování, popsaného na příkladu sítě složené ze dvou síťových zařízení firmy MikroTik a logovacího serveru. Všechna síťová zařízení pro komunikaci využijí UDP protokol a port 514. Na logovací server otevřeného portu tak bude přicházet směs logů pocházejících z různých stanic. Z těchto důvodů následující nastavení specifikuje chování, dle kterého bude každé IP adrese vytvořena složka, do které budou postupně logy řazeny dle témat.

```
#Custom template to generate the log filemane dynamically
bansed on the client s IP address.
$template RemInputLogs, "cesta/%FROMHOST-IP%/%PROGRAMNAME.log"
*.* ?RemInputLogs
```

V textovém řetězci v případě implementace je třeba nahradit slovo cesta validní cestou v dané stanici.

Nyní by měl být Rsyslog server připraven pro příjem logů z ostatních síťových zařízení. Z důvodu možnosti kontroly napsané a přepsání stávající, běžící konfigurace, je nejdříve vhodné překontrolovat syntaxi nastavení kvůli výskytům možných chyb.

```
sudo rsyslogd -f /etc/rsyslog.conf -N1
sudo systemctl restart syslog-ng
```

Jestliže po kontrole konfigurace serveru nejsou vypsány žádné chyby, je pravděpodobné, že zadané nastavení bylo správné. V opačném případě je potřeba přejít opět do výše zmíněného souboru a dané nesrovnalosti opravit. Po zadání příkazu restartujícího server je nové nastavení nahráno na běžící Rsyslog.

V poslední části zprovoznění je potřeba povolit budoucí komunikaci na firewallu[18], která bude probíhat mezi serverem a síťovými zařízeními. Firewall lze chápat jako bezpečnostní proces, fungující jako brána mezi daným zařízením a zbylou částí sítě,

[18]Jedná se o součást instalovaného operačního systému.

která na základě definovaných pravidel propouští, přesměrovává, nebo zahazuje provoz. V tomto konkrétním případě by se tak mohlo stát, že by jednotlivé Syslog zprávy (logy), mohly být zahazovány, například na základě pravidla o nevyžádané komunikaci.[21, 22]

```
sudo ufw allow 514/udp
sudo ufw allow from 192.168.38.0/24 to any port 514 proto udp
```

Zapsáním těchto výjimek bude dovoleno komukoliv z dané podsítě komunikovat na port 514 přes protokol UDP.

V případě využití jiných stanic pro virtualizaci Syslog serveru je potřeba modifikovat nastavení obdobným způsobem, které je popsané v textu. Program bude uzpůsoben k chování a zpracování dat plynoucích z výše popsaného nastavení.[19]

4.4.2 Výsledky programu na vytvořené testovací síti

Po modifikaci původního nastavení Rsyslogu je následně třeba nastavit i síťová zařízení k odesílání informací na server. Nastavení by mělo odpovídat požadavkům plynoucím z výše popsaných kapitol.

V následujícím kroku byly na všech třech síťových zařízeních vytvořeny události, o kterých by měly informovat Syslog server zasláním logu o události. Na zařízení firmy MikroTik došlo k vytvoření nového skriptu:

```
2023-04-07T13:59:10.364014+01:00 router.lan system,info new
script added by admin
```

Obr. 4.7: Ukázka MikroTik logu popisující přidání nového skriptu.

U Cisco směrovače došlo k přenastavení času:

```
2023-04-07T13:57:10.841131+01:00 %SYS-6-CLOCKUPDATE: System clock has
been updated from 16:10:26 AAA Fri Jun 28 1996 to 06:10:26 CET Sat Jun
29 1996, configured from console by console.
```

Obr. 4.8: Ukázka Cisco logu popisující přenastavení času.

A poslední změnou bylo u zařízení HP nejprve vypnutí a následné zapnutí vybraného rozhraní, které bylo používáno ke komunikaci s dalšími prvky sítě.

Vzhledem k tomu, že program není uzpůsoben zpracovávat logy výrobce HP, musí dojít k úpravám souborů dle následující ukázek konfigurace, aby mohl být

[19] Jedná se především o zachování logiky ukládání příchozích logů.

správně rozdělen uvedený log výše. První modifikace tak nastane u souboru „Rozdělení logu.csv“, kde musí být specifikována klíčová událost, která bude vyhledávaná v logu. Následně je potřeba v souboru „Předpis logu.csv“ definovat vyhledávanou událost. V tomto konkrétním případě bude přidán jeden řádek od daného typu a výrobce, obsahující klíčová slova, na základě nichž může dojít k propojení daného logu s událostí.

```
2023-04-07T14:01:09.124007+01:00 router.lan system,info new script
added by admin 192.168.1.8 vlan:  Opto_FNB virtual LAN enabled
```

Obr. 4.9: Ukázka HP logu popisující změnu rozhraní.

Tab. 4.7: Ukázka vyplněného souboru: Rozdělení logu v případě zařízení HP.

výrobce	**typ**	**vyhledávané události**
HP	1	vlan

Tab. 4.8: Ukázka vyplněného souboru: Předpis logu v případě zařízení HP.

výrobce	**typ**	**klíčová slova**
HP	1	Opto_FNB virtual LAN enabled

Po vyplnění všech vstupních, požadovaných souborů může program přejít k vyhodnocení samotných záznamů o nastalých událostech, které jsou zobrazeny níže.

Výpis 4.5: Ukázkový výpis běhu nad realnými logy firmy MikroTik.

```
Nalezena shoda s chybovým logem
[mikrotik, 1, c, new, script, scheduled]
Rozpoznávaný log:
[new script scheduled by admin]
IP 192.168.20.1 oznaceni r1

Probíhá kontrola konfigurace
Zaloha z předchozího porovnání nebyla nalezena a proto nemůže
být aktuální záloha porovnána
Aktuální záloha bude nahrána pro možnost dalšího porovnání
Soubory zálohy byly prázdné a proto se naplnily aktuální
zálohou
```

Výpis 4.6: Ukázkový výpis běhu nad realnými logy firmy HP.

```
Nalezena shoda s informativním logem:
[hp, 1, i, opto_fnb virtual lan enabled]
Rozpoznávaný log:
[023-01-01t00:15:54+02:00 192.168.1.8 vlan:  opto_fnb
virtual lan enabled]
IP 192.168.22.1 oznaceni r4
```

Výpis 4.7: Ukázkový výpis běhu nad realnými logy firmy Cisco.

```
Varování: byla nalezena shoda s kritickým logem, který může
ovlivnit celý proces běhu programu.
[cisco, 1, w, system, clock has been updated]
Rozpoznávaný log:
[system clock has been updated from 16:10:26 aaa fri jun]
IP 192.168.21.1 oznaceni r3
```

Výstupem programu je v tomto případě uvedený výpis výše, který je ořezán o informativní, textové řetězce vyhodnocující časové údaje o přihlášení uživatele, který tyto změny vykonal z důvodu, že jsou již jednou ukázkové výpisy zobrazeny v předcházející podkapitole.

Jednotlivé akce byly odlišeny dle jejich nastavení v souboru „Předpis logu.csv“. V případě události přidání nového skriptu byla nahrána aktuální záloha do referenční, protože se jednalo o první běh na této síti a program tak nemohl dříve vytvořit svou referenční zálohu.

Po vypsání výsledku zpracování jednotlivých logů program ukončí svou činnost. Je tak na samotném uživateli, jakým způsobem vyhodnotí předložené výsledky.

Závěr

Teoretická část přináší přehled protokolů a postupů potřebných pro analýzu záznamů událostí aktivních síťových prvků. Praktická část nejprve popisuje návrh základní bezpečnostní konfigurace pro nastavení zaznamenávání potřebných událostí a následného odeslání logů do centrálního úložiště k dalšímu zpracování. Příklad jednotlivých příkazů je uveden pouze pro dva vybrané výrobce, a tak zvolí-li uživatel jiného výrobce, měl by sám implementovat obdobné kroky, které jsou popsány v práci. Součástí práce je také uvedení příkladů možných útoků, které by mohl autorizovaný administrátor vykonat. Software je uzpůsoben k rozpoznání nejen nefyzických, ale i fyzických útoků. Fyzické útoky jsou zde zahrnuty z důvodu, že mohou být využity k zakrytí škodlivé činnosti.

Praktickým výstupem práce je software pro analýzu logů síťových prvků s cílem odhalit potenciální útoky prováděné přímo administrátorem těchto prvků. Software je koncipovaný jako univerzální. Z důvodu dostupnosti prvků pro testování byla otestována implementace pro prvky Cisco a Mikrotik. Pro zařízení dalších výrobců je potřeba obdobným způsobem implementovat popis struktury logů. Výstupní software pracuje s kompletní architekturou sítě, která je vytvořena při zadání jen částečných údajů. Tyto znalosti dovolují programu vyhodnotit výpadky zařízení a linek, o kterých může následně pomocí hlášení vydat varování.

Vytvořený program navíc dovoluje síťová zařízení od jediného výrobce rozdělit do více segmentů, nad nimiž bude probíhat vyhledávání jen určitých skupin logovaných událostí. Tímto krokem tak program umožňuje teoreticky rozdělit síť na kritické a nekritické části. V kritických částech pak mohou být vyhledávány jiné události než v ostatních segmentech.

Součástí výstupních funkcionalit je dále i možnost porovnání doby konfiguračních změn s předem zadaným obvyklým časovým rozvrhem administrátorů.

Software byl testován na síti složené ze zařízení Cisco, MikroTik, HP a Rsyslog serveru. Na všech síťových zařízeních byly vytvořeny neobvyklé události, které mohou být pro samotné zařízení potenciálně nebezpečné. Jednalo se o události změny času, přidání skriptu, vypnutí a následné zapnutí rozhraní. Program byl schopen všechny tyto události vyhodnotit a vypsat detailní varování do konzole, které uživatele informovalo o proběhlých událostech na síti.

Všechny výše zmíněné kroky vedou ke zvýšení bezpečnosti sítě. Program není zaměřen na ochranu před kybernetickými hrozbami/útoky typu DDOS, DOS přicházející od útočníků, ale na odhalení potenciálně nebezpečného chování vlastních, autorizovaných uživatelů – samotných správců. Jednotlivé bezpečnostní procesy by však měly teoreticky i odhalit odcizení administrátorského účtu, či další nezmíněné útoky na základě nestandardních událostí nebo práce v nestandardní dobu.

Literatura

[1] *ELENDEZ, W. A. a E. L. PETERSEN. The upper layers of the ISO/OSI reference model (part II). Computer Standards Interfaces, 1999, str. 185–199.*

[2] *JEŘÁBEK, J. Komunikační technologie. Brno: Vysoké učení technické, 2013. ISBN 978-80-214-4713-4.*

[3] *PUŽMANOVÁ, R. Moderní komunikační sítě od A do Z. Druhé vydání. Brno: Computer Press, 2006. ISBN 80-251-1278-0.*

[4] *JEŘÁBEK, J. Pokročilé komunikační techniky. Brno: Vysoké učení technické, 2008. Skripta.*

[5] *HALSALL, F. Computer Networking and the Internet. Páté vydání. Edinburg: Addison-Wesley, 2005. ISBN 0-321-26358-8.*

[6] *SATRAPA, P. IPv6: Internetový protokol verze 6. Čtvrté vydání. Praha: CZ.NIC, z. s. p. o, 2019. ISBN 978-80-88168-46-1.*

[7] *KUMAR, S. a S. RAI. Survey on Transport Layer Protocols: TCP and UDP. International Journal of Computer Applications. 2012, str. 20-25.*

[8] *STEWART, R. a C. METZ. SCTP: new transport protocol for TCP/IP. IEEE Internet Computing, 2001, str. 64-69.*

[9] *KREPS, J. I love logs : event data, stream processing, and data integration. Sebastopol: O'Reilly Media, 2015. ISBN 978-1-491-90938-6.*

[10] *GERHARDS, R. RFC 5424: The Syslog Protocol. 2009.* [online]. Dostupné z URL:
<`https://www.rfc-editor.org/info/rfc5424`>.

[11] *CASE, J., M. FEDOR, M. SCHOFFSTALL a J. DAVIN. RFC 1098: Simple Network Management Protocol. 1989.* [online]. Dostupné z URL:
<`https://www.rfc-editor.org/info/rfc1098`>.

[12] *System Message Logging. www.cisco.com. March 11, 2008.* [online]. Dostupné z URL:
<`https://www.cisco.com/c/en/us/td/docs/routers/access/wireless/software/guide/SysMsgLogging.html#wp1054946`>.

[13] *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide. San Jose, USA: Cisco Systems.* [online]. Dostupné z URL:

<https://www.cisco.com/c/dam/en/us/td/docs/switches/lan/catalyst4500/XE3-11-0E/configuration/guide/xe-311-cg.pdf>.

[14] *Cisco IOS Embedded Syslog Manager Command Reference. San Jose, USA: Cisco Systems.* [online]. Dostupné z URL: <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/esm/command/esm-cr-book.pdf>.

[15] *Manual:System/Log. wiki.mikrotik.com. 2020.* [online]. Dostupné z URL: <https://wiki.mikrotik.com/wiki/Manual:System/Log>.

[16] *Diagnostics, monitoring and troubleshooting: Log. https://mikrotik.com/support.* [online]. Dostupné z URL: <https://help.mikrotik.com/docs/display/ROS/Log>.

[17] *Mikrotik RouterOS automatic backup and update script. forum.mikrotik.com.* [online]. Dostupné z URL: <https://forum.mikrotik.com/viewtopic.php?t=156466>.

[18] *LEKHONKHOBE, T. Argparse Tutorial. Python.org.* [online]. Dostupné z URL: <https://docs.python.org/3/howto/argparse.html>.

[19] *Ubuntu.com.* [online]. Dostupné z URL: <https://ubuntu.com/>.

[20] *HITCHCOCK, K. Logging. In: Linux System Administration for the 2020s: The Modern Sysadmin Leaving Behind the Culture of Build and Maintain. Berkeley, CA: Apress, 2022, str 241–257. ISBN 978-1-4842-7983-0.*

[21] *Install and Setup Rsyslog Server on Ubuntu 22.04. Kifarunix.com.* [online]. Dostupné z URL: <https://kifarunix.com/install-and-setup-rsyslog-server-on-ubuntu/>.

[22] *ZALENSKI, R. Firewall technologies. IEEE Potentials. 002, (21), str 24-29.* [online]. Dostupné z URL: <https://ieeexplore.ieee.org/document/985324>.

[23] *ZALENSKI, R. Firewall technologies. IEEE Potentials. 002, (21), str 24-29.* [online]. Dostupné z URL: <https://ieeexplore.ieee.org/document/985324>.

[24] *ZALENSKI, R. Firewall technologies. IEEE Potentials. 002, (21), str 24-29.* [online]. Dostupné z URL: <https://ieeexplore.ieee.org/document/985324>.

[25] *ZALENSKI, R. Firewall technologies. IEEE Potentials. 002, (21), str 24-29.* [online]. Dostupné z URL: <https://ieeexplore.ieee.org/document/985324>.

A Popis přiloženého softwaru

Praktickým výstupem diplomové práce je i samotná realizace uvedených postupů, jejichž cílem je odhalení nelegitimního chování jednotlivých administrátorů. Přiložený kód programu realizuje veškeré kroky popsané v předcházejících kapitolách práce. Součástí řešení jsou i předpřipravené soubory pro ukázkový běh programu společně s návodem pro spuštění.