

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Monitoring tiskáren na platformě Centreon

Bakalářská práce

Vedoucí práce:
Ing. Martin Pokorný, Ph.D.

Tomáš Honek

Brno, 2017

Na tomto místě bych chtěl poděkovat firmě CDC Data, s. r. o. za poskytnuté zadání pro vypracování této bakalářské práce a konzultantovi Ing. Tomáši Holomkovi za ochotu a odborné rady při konzultacích. Také bych chtěl poděkovat vedoucímu mé práce Ing. Martinovi Pokornému, Ph.D. za cenné rady, ochotu i čas strávený při konzultacích a v neposlední řadě mé rodině za podporu během celého studia.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Monitoring tiskáren na platformě Centreon** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

Brno, 30. 12. 2016

.....

Abstract

Honek, T. Monitoring of printers on Centreon platform. Bachelor thesis. Brno, 2017.

This bachelor thesis deals with an issue of network printers monitoring on Centreon platform. The result of the work is a programmed plugin, which leads to detection the specific printer type using SNMP protocol, it analyzes the actual status of the device and gained values returns in the Centreon system's processable form. Eventually, the plugin is integrated into the monitoring system of the Centreon in an experimental network and its functionality has been verified on the production devices.

Keywords

Centreon monitoring system, monitoring of network printers, SNMP, plugin, Perl programming language

Abstrakt

Honek, T. Monitoring tiskáren na platformě Centreon. Bakalářská práce. Brno, 2017.

Tato bakalářská práce se zabývá problematikou monitoringu tiskáren na platformě Centreon. Výsledkem práce je naprogramování pluginu, který vede k detekci konkrétního typu tiskárny pomocí SNMP protokolu, analyzuje aktuální stav zařízení a získané hodnoty vrátí ve formě zpracovatelné systémem Centreon. Na závěr je plugin integrován do monitorovacího systému Centreon v experimentální síti a ověřena funkčnost na produkčních zařízeních.

Klíčová slova

monitorovací systém Centreon, monitorování síťových tiskáren, SNMP, plugin, programovací jazyk Perl

Obsah

1	Úvod	11
2	Cíl	12
3	Analýza uživatelských požadavků	13
4	Přehled existujících řešení a literatury	15
4.1	Kritéria vyhledávání	15
4.2	Přehled existujících prací	15
4.3	Přehled odborných knižních publikací	17
4.4	Přehled odborných článků na Internetu	18
5	Popis technologického aparátu	20
5.1	Základy TCP/IP	20
5.2	SNMP	21
5.3	MIB	22
5.4	Monitorovací systém Centreon	22
5.5	Programovací jazyk Perl	23
5.6	VPN	24
5.7	SSH	24
6	Návrh řešení	25
6.1	Schéma přenosu informací z tiskárny k uživateli	25
6.2	Vývojový diagram pluginu	26
6.3	Analýza přístupu k datům zařízení	30
6.3.1	Přístup k datům zařízení	30
6.3.2	Míra uveřejnění informací	30
7	Implementace	32
7.1	Vytvoření přístupu ke sledovaným zařízením a monitorovacímu systému Centreon	32
7.1.1	Síťová topologie přístupu k zařízení	32
7.1.2	Vytvoření přístupu k serveru přes SSH	33
7.1.3	Vytvoření SSH tunelu	34
7.1.4	Připojení ke vzdálenému serveru a webovému rozhraní Centreonu	35
7.2	Instalace monitorovacího systému Centreon	36
7.3	Implementace pluginu pro monitoring tiskáren	45
7.4	Vložení pluginu do monitorovacího systému Centreon	54
7.4.1	Přidání monitorovaného zařízení	54
7.4.2	Nastavení příkazu pro spuštění pluginu	56
7.4.3	Konfigurace služby v monitorovacím systému	57

7.4.4	Aktualizace konfigurace monitorovacího systému	59
7.4.5	Vytvoření performance grafu	59
8	Testování	61
8.1	Principy testování	61
8.2	Testování – webové rozhraní tiskáren	61
8.3	Testování – firma CDC Data, s. r. o.	64
8.4	Shrnutí testování	66
9	Ekonomické zhodnocení	67
10	Závěr	68
11	Seznam zdrojů	69
	Přílohy	73
A	CD – Zdrojový kód pluginu	74

1 Úvod

Pro úspěšné zvládnutí správy se stal monitoring nezbytnou součástí počítačových sítí. Díky neustálému rozvoji informačních technologií máme k dispozici široké spektrum technologií, protokolů a oblastí k monitorování. Celý monitorovací systém můžeme postavit na vytvoření skriptů na základě vlastních znalostí a zkušeností, nebo si vybrat z nabídky komerčních produktů. V případě vytvoření vlastních skriptů budeme mít detailní přehled o tom, jak monitoring probíhá, a můžeme se soustředit pouze na komponenty zařízení, které jsou pro nás důležité. Dalším z důležitých důvodů pro jeho nasazení lze zahrnout i bezpečnost a ochranu firemních dat. Firmy zabývající se touto problematikou v současném období používají nejrůznější open source softwary, se kterými se může naučit pracovat i zkušenější uživatel počítače. Pro snadnější pochopení je k některým softwarům k dispozici sada základních pluginů.

Bakalářská práce byla zadána prostřednictvím firmy CDC Data, s. r. o., konkrétně jejich pracovníkem a odborníkem v oblasti počítačových sítí a monitoringu Ing. Tomášem Holomkem, který v této bakalářské práci působí v roli konzultanta. Tato společnost se zaměřuje na systémovou integraci pro IT infrastrukturu se specializací na kvalitu poskytovatele služeb. Jejich preferencí je návrh a realizace řešení pro zákazníky, které odpovídá zadaným požadavkům i příslušnému prostředí (CDC Data, s. r. o., 2016). V současnosti jde o jedinou akreditovanou firmu na území České Republiky, která pro dohled nad informačními technologiemi využívá monitorovací systém Centreon (Centreon, 2016c). Z tohoto důvodu se ve své práci budu zabývat tvorbou pluginu pouze pro monitorovací systém Centreon. Důležitým faktem je, že tato firma poskytla naší fakultě zadání pro řešení tří bakalářských prací. Ve své práci se zabývám monitoringem tiskáren na platformě Centreon. Zbylé dvě práce zpracovávají kolegové Roman Opluštil, který se zabývá monitoringem diskových polí na platformě Centreon a Marek Hevier, jehož úkolem je monitoring serverů na platformě Centreon. Pro instalaci a seznámení se se základními vlastnostmi monitorovacího systému Centreon jsme s kolegy využili týmové spolupráce. Přesněji řečeno je monitorovací systém Centreon společným základem pro tyto tři bakalářské práce, ale v konečné fázi se každý z nás zaměřuje na monitoring odlišných zařízení (tiskárny, diskové pole, servery). Jako další zásadní odlišnost lze uvést použité technologie ve výše zmíněných pracích. Při monitoringu tiskáren a diskových polí se používá protokol SNMP pro získání dat z MIB tabulek příslušných zařízení. Na druhou stranu u monitoringu serverů je použita technologie parsování XML souborů.

2 Cíl

Na základě požadavků firmy CDC Data, s. r. o., práce pokrývá analýzu, návrh, implementaci pluginu v programovacím jazyce Perl, který provede detekci konkrétního modelu tiskárny, analyzuje aktuální stav zařízení a získané hodnoty vrátí ve formě zpracovatelné systémem Centreon.

Po návrhu řešení bude výsledný plugin implementován v experimentální síti a jeho funkčnost ověřena na produkčních zařízeních ve spolupráci s firmou CDC Data, s. r. o. Na závěr je zpracováno ekonomické zhodnocení.

3 Analýza uživatelských požadavků

Tato kapitola se zaměřuje na konkretizaci požadavků, zadaných zadavatelem bakalářské práce, které bude výsledný plugin na platformě Centreon umět zpracovat a vhodným způsobem interpretovat jejich výstup.

Zadavatelská firma si vytváří vlastní pluginy pro správu různých typů síťových zařízení, které pocházejí od různých výrobců. Základním požadavkem tvorby mého pluginu je kontrolovat stav tiskáren Konica Minolta bizhub C220, 4050, C3350, 283, C308, 4000P a C224e. Součástí každého z těchto zařízení jsou informace o typu, sériovém čísle a nepřehledné množství dalších informací, o kterých běžný uživatel nemusí mít na první pohled ani zdání. Dalším požadavkem od zadavatele je získání informací o zařízení jako jsou:

- typ monitorované tiskárny
- sériové číslo
- kontrola spotřebního materiálu (tonery, válce a jiné)
- chybový stav
- počítadla vytisknutých stránek i v podobě performance dat pro generování grafů

Důraz je kladen na to, aby vytvořený plugin uvedené informace uměl vhodným způsobem (z hlediska časového, přehlednosti a bez zásahu do zdrojového kódu) zpracovat a interpretovat jejich výstup pomocí monitorovacího systému Centreon pro každé zařízení zvlášť. Významnou vlastností pluginu je i jeho snadná rozšiřitelnost sledovaných informací o zařízeních, případně přidání dalších zařízení ke sledování pro náročnější zákazníky.

Nejsnadnějšími způsoby, jak určit o jaké zařízení se jedná, je zjistit jeho typ nebo sériové číslo. Zjištění typu monitorované tiskárny má význam pro uživatele i pro lepší orientaci při monitoringu zejména pokud sledujeme stav více tiskáren současně. O struktuře MIB (Management information base) tabulek jednotlivých tiskáren lze říci, že jsou si podobné, avšak nikoliv do detailů stejné. Pro představu to lze vysvětlit na jednoduchém příkladu: levná tiskárna zcela jistě nebude mít tolik funkcí jako dražší tiskárna a tudíž i struktura MIB tabulek levnější tiskárny bude o něco menší. Z tohoto důvodu je nejdříve nutné u daného pluginu rozpoznat typ monitorované tiskárny pro získání správných dat, neboť získání informací ze stejných MIB tabulek pro všech sedm typů tiskáren je neefektivní a v mnoha případech může způsobit zjištění nesprávných informací o zařízení. Sériové číslo je ojedinělé identifikační číslo zařízení, které mu bylo přiděleno jeho výrobcem.

Kontrola spotřebního materiálu patří mezi nejdůležitější a zcela běžnou součást monitoringu tiskáren. Téměř každý uživatel zná situaci, kdy v průběhu tisku většího množství potřebných materiálů zjistí, že má nedostatek některé z barev. Pro menší firmy tento spotřební materiál není zrovna levnou záležitostí a může se stát, že si ho nebudou moci dovolit nakoupit ve větším množství současně. Pro předcházení

těmto problémům je vhodný monitoring tonerů, válců a dalších komponent tiskáren. Sledování spotřebního materiálu a chybových stavů u tiskáren zmíněných na začátku této kapitoly bude cílem mé práce, ačkoli každá z tiskáren nemusí obsahovat úplně stejné komponenty a může být odlišné i jejich množství.

Další naprosto běžnou součástí monitoringu tiskáren je výpis chybových hlášení. Mnohokrát se stane, že se porouchá určitá část tiskárny a je nutný servisní zásah, aby došlo k odstranění závady. Bez monitoringu tohoto režimu by bylo velice pracné hledat určitou chybu.

Jelikož není možné zveřejnit informace z použitých dokumentací tiskáren, nejsou uvedeny příklady pro komponenty spotřebního materiálu ani chybových stavů.

Z ekonomické stránky firem používajících tiskárny je velice vhodné mít přehled o počtu vytisknutých stran papíru. Může se jednat o celkový počet vytisknutých stran, počet vytisknutých stran za měsíc, vytisknuté strany na uživatele a jiné. V mé práci se zaměřím na sledování celkového počtu vytisknutých stran. Jejich počet budu odlišovat i podle typu barvy: jednobarevný, dvoubarevný, černobílý či čtyřbarevný tisk, proto se jako poslední zaměřím na monitorování těchto statistik spolu s vytvořením performance grafů a jejich přehledné zobrazení v prostředí monitorovacího systému Centreon.

4 Přehled existujících řešení a literatury

4.1 Kritéria vyhledávání

Pro nalezení zdrojů v rámci této kapitoly bylo použito několik klíčových slov (centreon, nagios, monitoring, perl, snmp, network printers, konica minolta). Pro vyhledávání byly nejčastěji použity zdroje jako theses.cz, www.google.cz a books.google.com. Mezi další využitě zdroje vyhledávání lze zařadit scholar.google.com, www.amazon.com a www.seznam.cz.

Časové hledisko:

- **Závěrečné práce**

Za aktuální považuji závěrečné práce od roku 2010. Starší řešená problematika prací může fungovat na rozdílných principech než jsou aktuálně používané, nebo se již vůbec nepoužívají.

- **Odborné knižní publikace**

Zabývající se SNMP (Simple Network Management Protocol), MIB a programovacím jazykem Perl mohou být i starší, neboť základy a principy na kterých stojí, jsou pořád stejné i přesto, že se neustále vyvíjí.

- **Odborné články na internetu**

Za vhodné považuji články uveřejněné od roku 2013. Čím jsou aktuálnější, tím lepší je orientace v aktuálně řešených problémech a standardech. Jejich aktuálnost je pro mě důležitá i z hlediska vývoje a používání stále novějších verzí softwarů (např. monitorovacího systému Centreon).

4.2 Přehled existujících prací

Cílem kapitoly je obeznámit se s již existujícími dostupnými zdroji v podobě závěrečných bakalářských a diplomových prací. Poté bych měl mít dostatečný přehled o stavu v oboru, ve kterém se moje práce nachází a získat potřebné informace o základních principech a používaných technologiích. V případě, že se někdo jiný zabýval problematikou na podobné téma, je možné se jeho prací inspirovat. Pokud ovšem již existuje řešení pro moji práci, psaní této bakalářské práce by bylo zcela zbytečnou záležitostí.

Management počítačové sítě ISP (Filip Svoboda) – Bakalářská práce

Svoboda (2014) se zabývá managementem počítačové sítě poskytovatele připojení k internetu. Praktickou částí je instalace monitorovacího systému Nagios na server, díky kterému dojde ke zkvalitnění managementu sítě a případným upozorněním na výpadky síťových zařízení.

Přínosem je kapitola 2.4, ve které popisuje výhody a nevýhody dostupných softwarů pro správu sítí. Ve třetí kapitole se věnuje vhodnému výběru a následné instalaci softwaru Nagios v kombinaci s Centreonem.

Monitoring lokálnej siete prostredníctvom Nagiosu (Gábor Szabo) – Bakalářská práce

Szabo (2012) nejdříve popisuje co všechno a jakým způsobem je možné v síti monitorovat. V praktické části se zaměřuje na monitoring multifunkční tiskárny Konica Minolta bizhub C360 prostřednictvím Nagiosu. Výsledkem jeho práce je napsaný plugin v jazyce Java a jeho implementace pro kontrolu hladiny tonerů, množství papíru v zásobníku a doby provozu.

Velmi vhodná práce, ze které získám znalosti o tom, jak lze sledovat specifické údaje konkrétní tiskárny pomocí pluginu přes systém Nagios.

Monitorování aktivních prvků počítačové sítě (Zdeněk Habrman) – Diplomová práce

Habrman (2013) se zabývá možnostmi dohledových systémů, zejména těmi, které používají standardní protokoly SNMP a ICMP. Vybraný dohledový systém Cacti implementuje na univerzitní počítačovou síť.

Příslušná práce mi poslouží k získání základních znalostí o dohledových systémech. Důležitou částí je i závěr práce, ve kterém je řešena prezentace výsledků monitorování v podobě performace grafů.

Monitorování počítačových sítí a aplikací pomocí programu Nagios (Miroslav Mihok) – Diplomová práce

Mihok (2010) ve své práci využívá bezplatného systému Nagios k monitorování počítačových sítí. V teoretické části se zmiňuje o tom, proč je monitoring vhodný a jeho způsoby použití. Podrobněji je popsán použitý protokol SNMP a struktura MIB. V praktické části práce se věnuje instalaci a použití monitorovacího systému Nagios, ve spojení s odesláním reportů formou e-mailu nebo SMS zpráv a výsledky generuje ve formě grafů.

Užitečné pro moji práci je blíže se seznámit s monitorovacím systémem Nagios. Největším přínosem je zobrazení statistických výsledků ve formě grafů, které ve své práci využívám pro počítání vytisknutých stránek.

Optimalizace nákladů na tisk ve firmě (Patrik Fišer) – Bakalářská práce

Fišer (2012) se ve své práci zabývá redundancí nákladů na tisk ve firmě. Věnuje se problémům s rostoucím počtem tiskáren a jejich neefektivnímu využívání ve firmě. V praktické části řeší konkrétní návrh a implementaci pro firmu pomocí monitorovacího systému HP Web Jetadmin.

Získaným poznatkem pro moji práci je návrh řešení, jak dosáhnout efektivního využívání tiskáren v místní síti a také, jak snížit náklady firmy při tisku a výsledek vhodným způsobem umět interpretovat.

Správa heterogénných síťových prvků (Tomáš Szaniszlo) – Bakalářská práce

Szaniszlo (2012) se zabývá správou síťových prvků. Popisuje používané protokoly a nástroje univerzitního systému Fakulty informatiky na Masarykově univerzitě. V práci popisuje i systémy na monitorování stavu sítě. Cílem jeho práce je vylepšit správu používaného systému o nové vlastnosti a funkce.

V této práci jsou důležité zejména používané protokoly a rozšíření si znalosti z oblasti monitorovacích systémů, jejichž důležité vlastnosti jsou v práci popsány.

Správa různorodých síťových zařízení pomocí SNMP protokolu (Jiří Malík) – Diplomová práce

Malík (2011) se zabývá monitorováním, konfigurací a ovládáním různorodých síťových zařízení od různých výrobců pomocí protokolu SNMP. Klade důraz na flexibilitu systému v možnosti jeho libovolného rozšíření.

Velice důležitou součástí je třetí kapitola, kde je velmi detailně popsán protokol SNMP a struktura MIB tabulek.

Závěrem je nutné říci, že nebyla nalezena žádná bakalářská či diplomová práce zabývající se monitoringem tiskáren na platformě Centreon v počítačové síti. Tuto mezeru se snažím zaplnit vypracováním své bakalářské práce. V současnosti existuje pouze několik prací zabývajících se monitoringem na platformě Nagios, ze kterého se systém Centreon od určité verze oddělil a působí jako nezávislý systém. Tyto práce mi mohou posloužit k pochopení problematiky v oblasti monitoringu a jako podnět k vytvoření mého pluginu.

4.3 Přehled odborných knižních publikací

Účelem kapitoly je vyhledat odborné knižní publikace k nastudování základních principů a technologií, které ve své práci využívám. Aktuálnost těchto zdrojů nehraje až tak velkou roli, protože jde především o pochopení problematiky, zejména protokolu SNMP, MIB tabulek a základů programovacího jazyka Perl.

Essential SNMP, 2nd Edition (Mauro a Schmidt, 2005)

Obsahuje detailní popis protokolu SNMP. Nejdříve je vysvětleno, co to je SNMP a k čemu se používá. Nejvíce zajímavou se jeví především druhá a třetí kapitola. Zde se můžeme dočíst o verzích protokolu, čím se od sebe liší a nechybí ani popis struktury MIB tabulek.

Mistrovství - počítačové sítě (Sosinsky, 2010)

V této knize je dobré zaměřit se na úvodní kapitolu, zabývající se úvodem do světa počítačových sítí. Další významnou je čtvrtá kapitola, vysvětlující co je to TCP/IP a všechny potřebné protokoly, které sem spadají. V rámci poslední kapitoly se dozvíme základy o správě a diagnostice sítě.

Myslíme v jazyku Perl (Dařena, 2005)

Obsahem je stručná charakteristika jazyka určená od začátečníků po zkušenější uživatele. Tato odborná publikace se zabývá i popisem standardních funkcí jazyka včetně příkladů.

Slouží k získání základních poznatků o struktuře a možnostech jazyka, které jsou potřebné při programování pluginu.

TCP/IP Protocol Suite (Fourth Edition) (Behrouz, 2010)

Úvod je věnován popisům modelů OSI a TCP/IP a porozuměním základním síťovým technologiím. Druhá až čtvrtá část knihy se věnuje síťové, transportní a aplikační vrstvě protokolu TCP/IP a podrobně popisuje všechny protokoly a služby, které do příslušných vrstev spadají. Zajímavostí pro nás může být pátá část, popisující protokol nové generace IPv6.

Velký průvodce protokoly TCP/IP a systémem DNS (Dostálek a Kabelová, 2000)

Na začátku knihy jsou popsány síťové protokoly OSI a TCP/IP a detailně rozebrány jejich vrstvy. Dočteme se zde zajímavé informace o směrování, IP adrese i IP adrese nové generace. Významnými jsou především kapitoly devět a deset, směřující k velmi podrobnému popisu protokolů TCP a UDP.

4.4 Přehled odborných článků na Internetu

Kapitola slouží k vyhledávání návodů, dokumentací, postupů při instalaci a konfiguraci prvků používaných v současnosti. Především aktuálnost dokumentace Centreonu k instalaci nejnovějších verzí a operačního systému Debian je pro nás velice důležitá. Informace o principech a technologiích vyhledávaných v předchozí kapitole mohou být i staršího data.

SNMP Technical Reference (Microsoft, 2003)

Na stránkách od společnosti Microsoft lze získat přehled o protokolu SNMP. V odkazu „What Is SNMP?“ se dozvíme o jeho základních vlastnostech. U „How SNMP Works“ o komunikaci SNMP agenta a SNMP manažera a vysvětlení k čemu slouží MIB tabulka.

Cílem je získat obecné znalosti a pochopení funkčnosti protokolu SNMP. Monitorované údaje mají tiskárny uloženy v MIB tabulkách, proto pro nás je důležité seznámit se i s tímto pojmem.

The Debian Administrator's Handbook (Hertzog a Mas, 2015)

Tato příručka popisuje v kapitole 4 instalaci operačního systému Debian krok za krokem. Je zde popsáno vše od počátku instalace přes výběr jazyka, země, klávesnice, rozdělení pevného disku a mnoho dalšího až vytvoření prvního uživatele.

Welcome on Centreon Documentation (Centreon, 2016h)

Stránka poskytující dokumentaci k monitorovacímu systému Centreon, která je dostupná v anglickém i francouzském jazyce. Obsahuje detailní návod na instalaci potřebných součástí, která se odlišuje podle konkrétního operačního systému. První velice důležitou součástí tohoto systému je Centreon Engine (Centreon, 2015i). Ovšem pro správnou funkčnost systému nesmíme zapomenout na Centreon Broker (Centreon, 2016ch).

Jedná se o velmi přehledně zpracovanou dokumentaci, ve které najdeme vše potřebné od instalace až po správný chod monitorovacího systému Centreon.

Začínáme s monitoringem sítě (Bouška, 2009)

Bouška ve svém článku řeší základy monitoringu. Popisuje, co vše je možné na síti sledovat, jaké oblasti nás nejvíce zajímají a používané technologie. Detailněji jsou popisovány reporty monitoringu.

Slouží k rozšíření obecných znalostí o monitoringu. Zjistíme zde jaké zařízení můžeme monitorovat, pomocí jakých technologií a možné způsoby interpretace výstupů.

5 Popis technologického aparátu

5.1 Základy TCP/IP

Počítače pro vzájemnou komunikaci v počítačových sítích používají síťové protokoly. V prostředí Internetu se prosadila sada síťových protokolů TCP/IP, počítající pouze se čtyřmi vrstvami, na rozdíl od složitějšího modelu OSI složeného z vrstev sedmi. Stal se součástí všech operačních systémů (původně byl navržen pro Unix).

Na Internetu se pro síťové protokoly používají normy RFC (Request For Comments) a v současnosti jich existuje několik tisíc. Zde je několik příkladů RFC pro protokoly TCP/IP: RFC 791 – IPv4, RFC 2460 – IPv6, RFC 768 – UDP, RFC 793 – je základním protokolem pro TCP, přičemž TCP specifikace byly od té doby upraveny a revidovány, např. zlepšení chování pro kontrolu zahlcení v RFC 5681, RFC 3782, RFC 3390 a RFC 3168 (Fall a Stevens, 2012).

Vrstva síťového rozhraní odpovídá vrstvě linkové a fyzické v OSI modelu. Jejím úkolem je zajištění přístupu k přenosovému médiumu a řízení datového spoje. TCP/IP může být používán na jakémkoliv typu sítě, protože implementace této vrstvy závisí na přenosové technologii, která je použita v příslušné síti.

Síťová vrstva se stará o přenos paketů mezi všemi uzly v síti od odesílatele k příjemci. K tomu je nezbytné, aby zvládla směrování prostřednictvím směrovačů a tzv. brán u uzlů v síti, které spolu přímo nesousedí. Je implementována ve všech prvcích sítě, ať jde o směrovače či koncová zařízení. Hlavním protokolem je IP protokol a z tohoto důvodu bývá tato vrstva často označována v odborné terminologii jako IP vrstva. Jeho úkolem je přenos datagramů mezi počítači na základě IP adres. Každé záhlaví datagramu obsahuje adresu odesílatele a příjemce, proto může síť přenášet každý datagram samostatně a k příjemci nemusí dorazit ve správném pořadí. Zajištění spolehlivosti obstarává vyšší vrstva (protokol TCP) nebo příslušné aplikace. Mezi další známé protokoly této vrstvy můžeme zařadit: ARP, RARP a ICMP. Rozsáhlým popisem protokolů IPv4, IPv6, ARP a RARP se zabývá Comer (2014) v kapitole 5 a 6.

Transportní vrstva je implementována na koncových počítačích a poskytuje jim služby, které přizpůsobují chování sítě potřebám aplikací. Rozdíl těchto služeb se liší v jejich spolehlivosti. Zatímco protokol TCP poskytuje spolehlivé doručení, protokol UDP se za správné doručení nezaručuje.

Protokol TCP dopravuje TCP segmenty konkrétním aplikacím spuštěných v koncových zařízeních. Důležitým rysem je zajištění spolehlivého doručení dat, případným opakováním přenosu, a doručení ve správném pořadí. Z toho důvodu se před začátkem přenosu k navázání spojení používá třicestný handshake, kde se obě strany dohodnou na číslu sekvence. Využití TCP protokolu bývá v místech, kde se upřednostňuje správnost doručovaných dat před rychlostí jejich doručení, ať už se jedná o bankovní transakce, e-mail nebo komunikační protokol SSH. Navazováním spojení a strukturou TCP segmentu se podrobněji zabývá Behrouz (2010).

Protokol UDP při přenosu datagramů neposkytuje záruku za jejich doručení, proto bývá označován jako nespolehlivý. Neprovádí se kontrolní součet a neposílá se potvrzení o správném doručení odesílateli, z čehož plyne malé zatížení sítě. Spolehlivost (potvrzení doručení, správnost přijatých datagramů a doručení ve správném pořadí) místo protokolu UDP přebírají aplikace, které ho používají. Nasazení protokolu je vhodné tam, kde je vyžadována jednoduchost a počítá se s určitou mírou ztráty datagramů. Jako konkrétní příklad lze uvést volání přes Internet (VoIP), DNS, protokol SNMP či streamování videa (Builder, 2003).

Aplikační vrstva TCP/IP vznikla spojením aplikační, prezentační a relační vrstvy odpovídající OSI modelu. Obsahuje všechny protokoly, které poskytují konkrétní služby pro uživatele. Některé z nich vyžadují protokol TCP na transportní vrstvě (FTP, Telnet atd.), jiné protokol UDP (DHCP, TFTP). Zvláštním případem je DNS, který může pracovat s jakýmkoli protokolem transportní vrstvy (Lammler, 2010).

5.2 SNMP

Protokol SNMP (Simple Network Management Protocol) je jednoduchý, široce rozšířený a používaný se pro správu sítě. Základním konceptem je použití pojmů SNMP manažer a SNMP agent. Manažer (obvykle hostitel) řídí a monitoruje řadu agentů (směrovač, server, switch, tiskárna a jiné). Protokol pracuje na nejvyšší aplikační vrstvě OSI modelu, aby bylo možné sledovat zařízení vyrobená různými výrobci a instalovaná na různých fyzických sítích. SNMP typicky využívají správcovské počítače (manažeři), které sledují počítače či jiná zařízení v síti. Agent poskytující pomocí SNMP informace manažerovi je spuštěný na sledovaných zařízeních. Komunikace je dosažena prostřednictvím jednoduché interakce mezi manažerem a agentem. Úkolem agenta je shromažďovat a uchovávat informace o stavu zařízení v databázi. SNMP pracuje ve dvou režimech. V prvním režimu pro kontrolu sledovaného zařízení vznášá manažer požadavky agentovi (formou zpráv SNMP GetRequest) na zaslání požadovaných informací. Obdržené informace jsou poté na monitorovací straně důkladně zpracovány a výsledkem procesu je vyhodnocení stavu zařízení (např. router – pracuje správně, přetížený, nedostupný). V druhém režimu odesílá oznámení (SNMP Trap) agent směrem k manažerovi a to pokud si všimne něčeho neobvyklého (Behrouz, 2010).

Počáteční verzí SNMP je verze 1 (dále jen SNMPv1). Ačkoli jde o historicky nejstarší verzi, je stále hlavní implementací SNMP, kterou mnozí dodavatelé podporují. SNMPv1 využívá bezpečnostní mechanismus založený na hesle – vyžaduje pouze textovou komunitu (community string). Nejsou zde žádná bezpečnostní opatření od někoho s přístupem k síti. V průběhu svého vývoje prošel tento protokol několika proměnami s cílem aktualizovat řadu maličkostí. V důsledku toho vznikl SNMPv2 jako přímá náhrada a aktualizace SNMPv1 s novými funkcemi pro zabezpečení. Jeho součástí je také podprotokol SNMPv2c, který ovšem pro zabezpečení používá stejný mechanismus jako SNMPv1 a je jeho „lehčí“ verzí. Obě verze jsou kompatibilní s prvotní verzí, ale SNMPv2 má i možnost pracovat po nastavení proxy

jako SNMPv1. Nejnovější verzí protokolu, s cílem poskytnout mnohem vyšší úroveň zabezpečení oproti předchozím verzím je SNMPv3. Jejím hlavním přínosem pro správu sítě je zajištění bezpečnosti formou autentizace (kryptografickou hašovací funkcí MD5 nebo SHA) a šifrováním DES, 3DES a AES na ochranu dat (Wilkins, 2011).

5.3 MIB

MIB je sada objektů, uspořádaných v hierarchické stromové struktuře definované standardem ASN.1 (Abstract Syntax Notation One), které se používají ke správě příslušného zařízení. V závislosti na vykonávané funkci se liší počet jeho MIB tabulek. Výrobci zařízení jsou přidělovány podstromy (subtree), jejichž následnou strukturu si sami vytvářejí v neomezené velikosti. Každá část stromové struktury nese označení v podobě textového popisu a hodnoty typu číselného integeru. Ke konkrétním informacím se přistupuje prostřednictvím tohoto číselného integeru tzv. OID (Object Identifier), specifikující cestu od kořene (root) až k danému objektu (Mauro a Schmidt, 2005, s. 23).

Podrobnější přehled o struktuře MIB a přístup k informacím tiskáren Konica Minolta je uveden v kapitole „6.3 Analýza přístupu k datům zařízení“.

5.4 Monitorovací systém Centreon

Centreon (až do roku 2015 známý jako Merethis) je open source software pro komplexní monitorování podnikových sítí a IT systémů. Centreon (2016b) poskytuje základní informace o společnosti a jejich vývoji monitorovacího systému od prvopočátku.

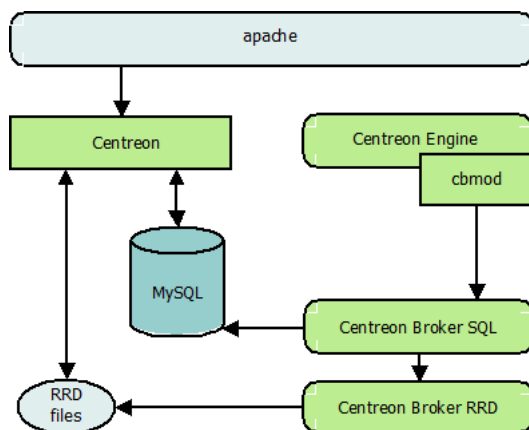
Tento monitorovací systém se oddělil od Nagiosu verze 3.2.3 a nyní působí jako samostatný monitorovací systém. Jeho největší předností oproti systému Nagios je kvalitní webové rozhraní. Přehledně na jednom místě zobrazuje aktuální stav monitorovaných zařízení. Centreon se skládá ze dvou základních částí, Centreon Engine a Centreon Broker, které jsou pro uživatele volně dostupné. Za rozšířené součásti a služby jako jsou Centreon MBI (Monitoring intelligence for business) Centreon BAM (Business Activity Monitoring) a Centreon Map si náročnější uživatelé připlatí (Centreon, 2016g).

Hlavním úkolem Centreon Engine je sledovat hostitele a služby na síti a upozornit nás v případě, kdy dojde k nějaké závadě. V případě problému je možnost nastavit okamžité informování uživatele prostřednictvím e-mailu, SMS zprávy nebo jinou uživatelem definovanou metodou. Jediným požadavkem pro funkčnost Centreon Engine je instalace na jakýkoli operační systém Unix.

Centreon Broker se stará o ukládání informací získané Centreon Enginem do zabezpečené databáze. O klíčovou vlastnost Centreonu, kterou je vytváření grafů z informací získaných sledováním za určité časové období, se dříve staral software Centstorage. Tento rys nyní převzal Centreon Broker, který umožňuje vytváření grafů rychlejším a efektivnějším způsobem. Detailní návod instalace, použití a další

zajímavé informace k těmto dvěma základním kamenům monitorovacího systému poskytuje Centreon (2015j) a Centreon (2016ch).

Na obrázku níže můžete vidět základní strukturu monitorovacího systému Centreon.



Obrázek 1: Struktura monitorovacího systému Centreon (Centreon, 2015a, s. 18)

Klíčovým prvkem pro komplexní funkčnost monitorovacího systému je vytváření vlastních pluginů, které provádí kontrolu různorodých síťových zařízení, dle potřeby uživatele. Jedním z možných přístupů, jak provádět monitoring, je možnost získat data ze zařízení pomocí protokolu SNMP. Jako příklady monitorování můžeme uvést zatížení procesoru, využití disku a dostupnost zařízení. Výstupní data pluginu musí jednoznačně specifikovat závadu na konkrétním zařízení, tato data jsou Centreon Enginem dále zpracovávána a nakonec přehledně zobrazena ve webovém rozhraní Centreonu, na základě kterých uživatel závadu odstraní.

5.5 Programovací jazyk Perl

Perl je interpretovaný, programovací jazyk. Stal se velmi populárním nástrojem pro vytváření webových aplikací (v kombinaci s HTML, CSS, Javascriptem atd.) formou CGI skriptů (Common Gateway Interface – protokol, který propojuje externí aplikace s webovým serverem). Lze ho využít i jako vhodný nástroj pro zpracování velkého množství textu (parsování XML souborů).

Perl nevyžaduje deklaraci proměnných, definování vlastních datových typů a neexistuje zde typová kontrola. Základem jsou proměnné typu: skaláry, pole skalárů a hashe skalárů (Dařena, 2012). Mimo tyto úplné základy je nutné se k vytvoření pluginu naučit základní syntaxi jazyka u cyklů, podprogramů a práci s regulárními výrazy. Nejen úvodem do tohoto programovacího jazyka včetně ukázkových příkladů se zabývá Dařena (2005) a Satrapa (2000). K tomuto jazyku je dostupná celá řada specializovaných modulů i s dokumentací v CPAN, které programátorům výrazně ulehčují práci a jsou volně dostupné (CPAN, 2010).

5.6 VPN

Virtuální privátní síť vytváří bezpečné propojení mezi koncovým zařízením a serverem, umístěným v privátní počítačové síti organizace, prostřednictvím veřejné počítačové sítě. Autentizace z obou stran probíhá při navazování spojení pomocí digitálních certifikátů. Hlavní výhodou tohoto typu vzdáleného přístupu je umožnění koncovému zařízení zajistit fyzickou přítomnost v privátní síti. Není tedy nutné svoji práci vykonávat pouze v místě privátní sítě organizace, ale například i z domova.

Jak autentizace mezi dvěma zařízeními probíhá a jejím podrobnějším popisem se blíže zabývá Bouška (2014).

Kromě autentizace používá VPN k zabezpečení autorizaci, kontrolu integrity dat a anti-reply. Autorizací rozumíme přidělení práv uživatelům, aby mohli provádět některé operace či povolení přístupu k informacím. Při kontrole integrity dat je odhaleno, zda nedošlo při jejich přenosu k neočekávaným změnám a zajištění kompletnosti dat. O ochranu proti zachycení dat třetí stranou (hackery) a jejich změně následované vložení do proudu dat mezi zdrojovým a cílovým počítačem, se stará anti-replay.

5.7 SSH

SSH (Secure Shell) přináší bezpečnou (šifrovanou) formu komunikace mezi dvěma vzdálenými počítači. Použijeme ho v případě, kdy se chceme na dálku připojit ke vzdálenému serveru (prostřednictvím přihlašovacích údajů), k provádění různých operací nebo jen pro přenos souborů. Nejprve si obě zúčastněné strany vymění verzi protokolu, kterou používají, následně server posílá svůj veřejný hostitelský klíč a veřejný server-klíč, který je automaticky generován v předem daných intervalech a ukládán na disk. Spojení vznikne vytvořením klíče klienta, který je zašifrován klíčem svým i klíčem serveru a odeslán serveru. Výsledkem je klíč používaný pro šifrování veškeré komunikace (Moučka a Peša, 2011).

Tento komunikační protokol byl navržen jako vhodná náhrada za špatně zabezpečené protokoly a služby typu Telnet, které posílají data včetně hesla v nebezpečné formě a umožňují jejich snadné získání pomocí odposlechu při přenosu nedůvěryhodnou počítačovou sítí jako je například Internet (dsl.cz, 2016).

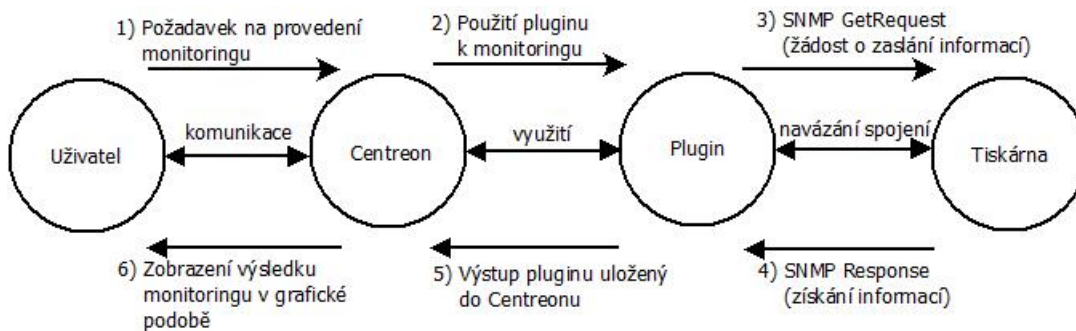
Způsob, jakým autentizace a navázání spojení mezi počítačem a serverem probíhá, je blíže popsán v kapitolách „7.1.2 Vytvoření přístupu k serveru přes SSH“ a „7.1.3 Vytvoření SSH tunelu“.

6 Návrh řešení

Cílem kapitoly je získat úplnou představu o funkci vytvářeného pluginu od jeho spuštění až k získání výsledků monitoringu. Poté bude následovat kapitola věnující se jeho samotné implementaci. Pro lepší představu je vytvořen návrhový diagram znázorňující stavy, které mohou v průběhu monitorovaného zařízení nastat. Ke každému stavu v diagramu následuje detailní popis jeho činnosti. Závěr kapitoly se věnuje analýze přístupu, která povede k získání informací z jednotlivých zařízení prostřednictvím MIB tabulek.

6.1 Schéma přenosu informací z tiskárny k uživateli

Cílem podkapitoly je získat ucelenou představu o funkci a vzájemné vazbě jednotlivých prvků, které jsou součástí monitoringu. Schéma prvků včetně slovního popisu je zobrazeno na obrázku č. 2.

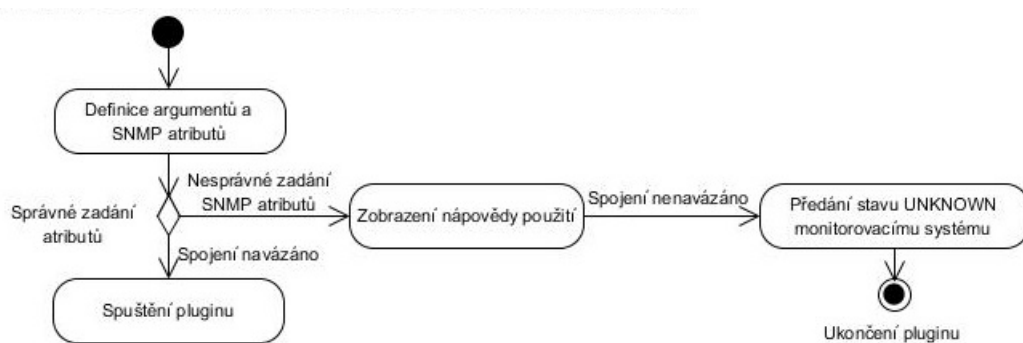


Obrázek 2: Schéma prvků monitoringu pro přenos informací

Celý průběh monitoringu začíná vznesením požadavku uživatele na ověření správné funkčnosti zařízení pomocí monitorovacího systému Centreon. Centreon na tento požadavek reaguje spuštěním příslušného pluginu, který je navržený pro získání informací z konkrétního zařízení. Po úspěšném navázání spojení se zařízením (protokolem SNMP) plugin odešle požadavek *SNMP GetRequest* pro získání informací o aktuálním stavu zařízení. Monitorované zařízení na tuto žádost odpoví ve formě *SNMP Response* a získané informace jsou následně pluginem zpracovány. Poté následuje zaslání informací ve struktuře, které monitorovací systém Centreon rozumí a dokáže ji vhodným způsobem interpretovat. Na závěr celého procesu jsou výsledky monitoringu přehledně zobrazeny v grafické podobě monitorovacím systémem Centreon prostřednictvím webového prohlížeče.

6.2 Vývojový diagram pluginu

Vývojový diagram lze rozdělit na tři základní části. Úvodní část se zabývá definicí argumentů a SNMP atributů, které jsou potřebné pro spuštění pluginu a navázání spojení se sledovaným zařízením. Druhá část diagramu je věnována specifikaci komponent, které jsou kontrolovány, a přístupu k nim. Poslední část se stará o to, jakým způsobem jsou výsledky monitoringu zobrazeny uživateli.



Obrázek 3: Návrh vývojového diagramu – část první

Nápověda je k dispozici pro pochopení významu jednotlivých argumentů a způsobu zadávání, aby byly správně předány pluginu.

Zobrazení nápovědy: `perl check_printers_konica_minolta.pl -help`

Spuštění pluginu: `perl check_printers_konica_minolta.pl -mode: errors -hostname: 192.168.20.10 -snmp_community: public -snmp_version: 2c`

- **Definice argumentů a SNMP atributů**

Pro spuštění naprogramovaného pluginu je nutné, aby uživatel správně zadal všechny potřebné argumenty, ať už jde o jejich nastavení v prostředí monitorovacího systému Centreon, nebo jejich zadání při spuštění pluginu z příkazového řádku. Všechny argumenty musí být zadány v následujícím tvaru: `-nazev_predavaneho_argumentu: hodnota`.

Seznam všech povinných argumentů: `-mode`, `-hostname`, `-snmp_version` a `-snmp_community`. Nepovinné argumenty využitelné pouze pro mód `consumables` jsou `-w` a `-c`.

- **Zobrazení nápovědy použití**

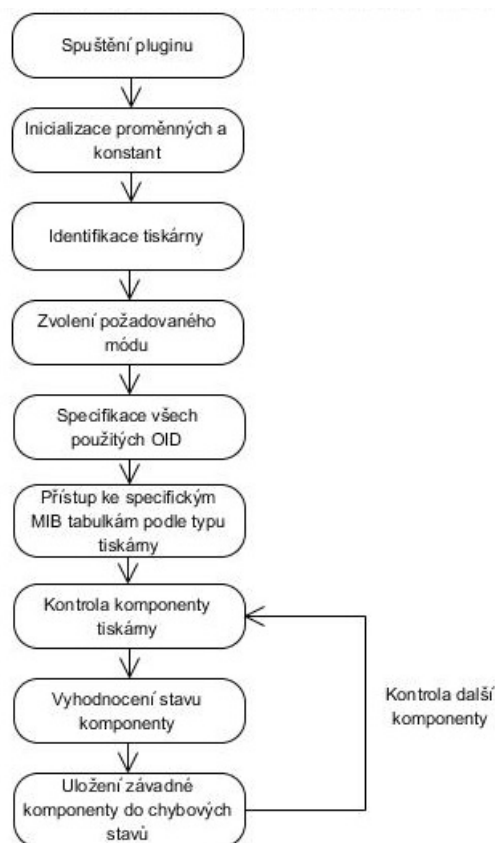
Při nesprávném zadání potřebných argumentů či SNMP atributů se uživateli zobrazí nápověda, aby věděl, jakým způsobem plugin spustit.

- **Předání stavu UNKNOWN monitorovacímu systému**

Současně se zobrazením nápovědy dochází k odeslání stavu *UNKNOWN* monitorovacímu systému a jeho zobrazení ve webovém rozhraní Centreonu následované ukončením pluginu.

- **Spuštění pluginu**

Po správném zadání všech potřebných argumentů následuje spuštění pluginu.



Obrázek 4: Návrh vývojového diagramu – část druhá

- **Inicializace proměnných a konstant**

Nejdříve probíhá definování chybových stavů, které jsou předávány monitorovacímu systému při zjištění nějakého nedostatku. Přesněji řečeno jsou to stavy: *OK*, *WARNING*, *CRITICAL*. Poté jsou definovány další proměnné a konstanty, které jsou použity.

- **Identifikace tiskárny**

Klíčovou roli pro správnou funkci pluginu je zjištění typu tiskárny. Od toho se odvíjí počet monitorovaných komponent, provádění výpočtů procentuální vyčer-

panosti spotřebního materiálu a počítání vytisknutých stránek podle typu barev vztahující se pouze k určité tiskárně, neboť každá tiskárna nemusí mít stejné komponenty ani stejnou strukturu MIB tabulek.

- **Zvolení požadovaného módu**

Pro přehlednost uživatele je navíc monitorování každé tiskárny rozděleno do tří základních módů *errors*, *consumables* a *counters*, jejichž názvy byly zvoleny po konzultaci se zadavatelem z firmy CDC Data, s. r. o. Toto rozdělení poskytuje uživateli obrovskou výhodu, protože si může vybrat pouze konkrétní oblast, kterou chce sledovat.

- **Specifikace všech použitých OID**

Pro přístup k informacím získaných z tiskárny potřebuje každý režim přistupovat k odlišným MIB tabulkám, proto je nutné na počátku specifikovat OID čísla používaná pro daný režim a uložit je do vhodné datové struktury.

- **Přístup ke specifickým MIB tabulkám podle typu tiskárny**

Následuje přístup k MIB tabulkám, ze kterých jsou zjišťovány informace o komponentách tiskárny. Přístup probíhá na základě určení typu tiskárny, aby bylo jasné ke kterým MIB tabulkám přistupovat.

- **Kontrola komponenty tiskárny**

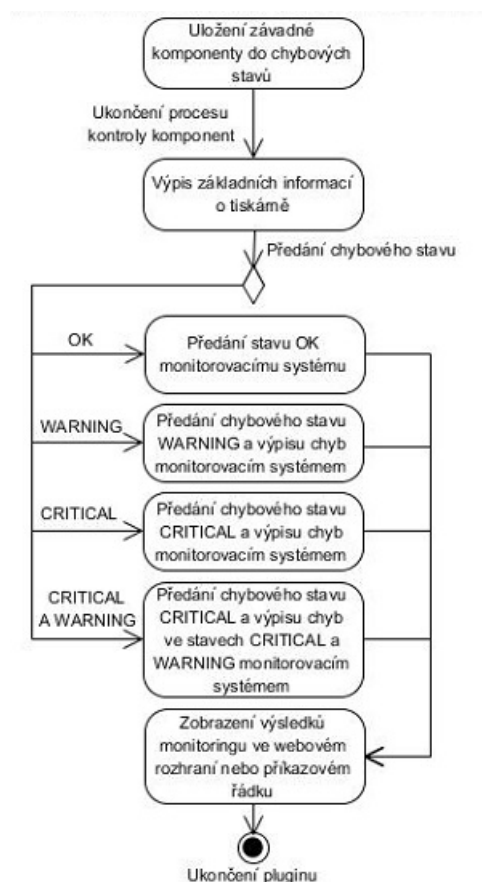
Dalším krokem je zjištění potřebných informací o určité komponentě tiskárny, které vedou k určení jejího aktuálního stavu. Tyto údaje nalezneme přístupem ke specifikovaným MIB tabulkám z předchozího kroku.

- **Vyhodnocení stavu komponenty**

Po nalezení všech potřebných informací v kombinaci s pomocnými výpočty a porovnáními hodnot určíme aktuální stav komponenty.

- **Uložení závadné komponenty do chybových stavů**

Po důkladném vyhodnocení aktuálního stavu zmíněného v předchozím kroku, je komponenta uložena do jednoho ze stavů: *OK*, *WARNING* nebo *CRITICAL*.



Obrázek 5: Návrh vývojového diagramu – část třetí

- **Výpis základních informací o tiskárně**

Pro přehlednost jsou jako první vypsaný informace identifikující tiskárnu.

- **Předání stavu *OK* monitorovacímu systému**

Současně s předáním základních informací je monitorovacímu systému předán stav *OK* v případě, kdy jsou všechny komponenty v pořádku.

- **Předání chybového stavu *WARNING* a výpisu chyb monitorovacím systémem**

Současně s předáním základních informací je monitorovacímu systému předán stav *WARNING* a výpis všech závadných komponent.

- **Předání chybového stavu *CRITICAL* a výpisu chyb monitorovacím systémem**

Současně s předáním základních informací je monitorovacímu systému předán stav *CRITICAL* a výpis všech závadných komponent.

- **Předání chybového stavu *CRITICAL* a výpisu chyb ve stavech *CRITICAL* a *WARNING* monitorovacím systémem**

Současně s předáním základních informací je monitorovacímu systému předán stav *CRITICAL*, neboť má vyšší prioritu než *WARNING*. Následuje výpis všech závadných komponent obou stavů.

- **Zobrazení výsledků monitoringu ve webovém rozhraní nebo příkazovém řádku**

Nakonec jsou všechny informace přehledně zobrazeny prostřednictvím webového prohlížeče v monitorovacím systému Centreon nebo v příkazovém řádku. Záleží pouze na nás, kde plugin spustíme. Tímto způsobem dojdeme k závěrečnému výsledku, následovaného ukončením pluginu.

6.3 Analýza přístupu k datům zařízení

Účelem této podkapitoly je analýza způsobu, kterým se přistupuje k datům zařízení.

6.3.1 Přístup k datům zařízení

Navázání spojení pro přístup pluginu k datům zařízení je realizováno protokolem SNMP, jehož základní funkce a existující verze již byly popsány v kapitole „5.2 SNMP“. Komunikace monitorovaných tiskáren (Konica Minolta bizhub C220, 4050, C3350, 283, C308, 4000P a C224e) probíhá prostřednictvím protokolu SNMP verze 2c. Tato informace je získána z uživatelských manuálů tiskáren Konica Minolta (2009, 2013, 2013, 2010, 2015, 2013, 2013).

SNMP definuje operace nad MIB tabulkami. Úkolem MIB je popsat jejich objekty, s kterými je možné pracovat, jejich typ (Integer, Octet String, Counter atd.) a definuje celkem čtyři přístupy k nim (read only, read and write, write only a not accessible). V případě našeho monitoringu je prováděn pouze přístup *read only*, tudíž nemůže dojít ke změně, nebo záměrnému poškození stávajících dat.

6.3.2 Míra uveřejnění informací

Pro přístup k MIB tabulkám obsažených tiskárnami nejdříve musíme porozumět jejich základní struktuře a uspořádání informací v nich. Poté můžeme přikročit k získávání informací z těchto MIB tabulek protokolem SNMP.

V podstatě existují dva způsoby, jakými je možné získat MIB tabulky zařízení. Někteří výrobci je distribuují ve formě .mib souborů na svých oficiálních stránkách. Druhý způsob spočívá ve využití SNMP protokolu založeného na modelu klient / server, kdy se SNMP klient ptá na stav jednotlivých proměnných a SNMP agent implementovaný v zařízení mu poskytne odpověď. Komunikace probíhá formou otázek a odpovědí, kde proměnné odpovídají OID číslování (HW group, 2016).

Výrobce použitých tiskáren Konica Minolta nezveřejňuje MIB tabulky zařízení na svých oficiálních stránkách, ale pro účely bakalářské práce byly k těmto tiskárnám poskytnuty firmou CDC Data, s. r. o dokumentace, obsahující MIB tabulky s detailním popisem objektů pro každou z monitorovaných tiskáren. Jelikož je CDC Data, s. r. o. oficiálním distributorem firmy Konica Minolta, získává dokumentace jejich zařízení na požádání. Vzhledem k tomu, že není možné zveřejnit jakékoliv informace nacházejících se v těchto dokumentacích, není popsán způsob, jakým jsou jednotlivé informace ze zařízení protokolem SNMP s použitím konkrétních OID čísel pro dotazování získávány.

7 Implementace

Kapitola se zabývá popisem technologií sloužících pro přístup ke sledovaným zařízením a k serveru, na kterém běží monitorovací systém Centreon. Ačkoli je tento systém již nainstalován na vzdáleném serveru, s jeho instalací, které se v této kapitole také věnuji, jsem se seznámil již v předmětu AVT (Aplikace Vývojových Technik), sloužící jako příprava pro bakalářskou práci. Nejdůležitější částí je vytvoření hosta v monitorovacím systému Centreon, implementace pluginu a jeho následné reálné použití.

7.1 Vytvoření přístupu ke sledovaným zařízením a monitorovacímu systému Centreon

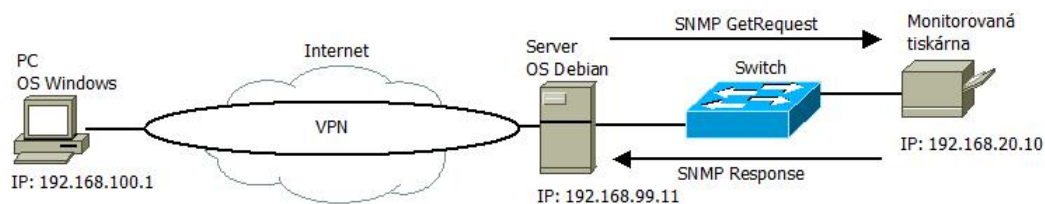
Běžnou praxí je instalace a nasazení monitorovacího systému na server umístěný v lokální počítačové síti, z důvodu jednoduchého a efektivního sledování příslušných síťových zařízení.

Při vývoji pluginu je třeba mít přístup k monitorovacímu systému Centreon a sledovaným zařízením v reálném čase, neboť mezi důležitou součástí vývoje patří neustálé ověřování a testování správnosti hodnot na zařízeních vykazovaných pluginem. Pro účely vytváření pluginu byl v rámci bakalářské práce poskytnut přístup do podnikové sítě firmy CDC Data, s. r. o. přes VPN, ve které se nachází server s monitorovacím systémem Centreon prostřednictvím SSH tunelu. Součástí podnikové sítě jsou i příslušná zařízení, jejichž kontrola aktuálního stavu je při vývoji pluginu ověřována a testována. Na závěr je vytvořen plně funkční plugin integrovaný do monitorovacího systému Centreon a implementovaný v podnikové síti způsobem běžným pro praxi.

Pro vysvětlení následujících podkapitol (zabývajících se síťovou topologií, vytvořením přístupu k serveru až po přihlášení do monitorovacího systému) jsou použity smyšlené IP adresy a standardně používaná čísla portů přiřazená běžným službám (SSH – 22, HTTP – 80), aby bylo možné ukázat, jakým způsobem se přístup realizuje.

7.1.1 Síťová topologie přístupu k zařízení

Síťová topologie přístupu k zařízení, zobrazena na obrázku č. 6, nám blíže znázorňuje vzdálený přístup k serveru a zařízení pro získání lepší představy o způsobu komunikace mezi použitými síťovými prvky.

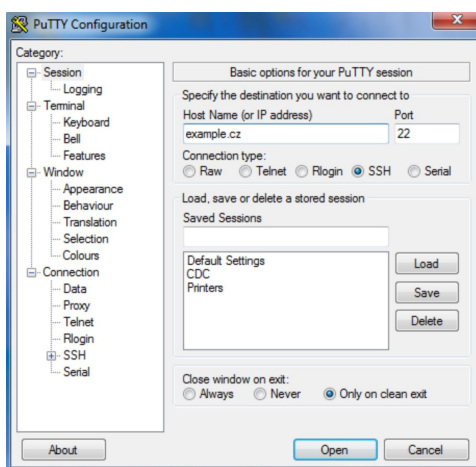


Obrázek 6: Topologie sítě

Síťová topologie se skládá ze tří hlavních prvků: počítač, server a tiskárna. Z počítače je uskutečněn vzdálený přístup na server pomocí vytvoření SSH tunelu na základě požadavku uživatele. Tento přístup může být realizován z libovolného počítače či notebooku s přístupem k Internetu. Na serveru s operačním systémem Debian běží monitorovací systém Centreon. Díky navázání spojení počítače se serverem můžeme ovládat server z příkazového řádku. Toho využijeme zejména při implementaci pluginu, kdy potřebujeme přistoupit na sledované zařízení a získat z něho konkrétní data pomocí protokolu SNMP, nebo pro spuštění pluginu zadáváním příkazů z příkazového řádku. Jedná se o daleko jednodušší a rychlejší způsob ověřování funkčnosti pluginu než jeho zdlouhavé vkládání do složky určené pro pluginy monitorovacího systému a čekání na zobrazení výsledků ve webovém rozhraní Centreonu. V konečné fázi je plugin integrován do monitorovacího systému Centreon, pro jehož zobrazení v grafické podobě využijeme webový prohlížeč počítače.

7.1.2 Vytvoření přístupu k serveru přes SSH

K úspěšnému vytvoření přístupu ke vzdálenému serveru potřebujeme odpovídající program, který nám dokáže zajistit bezpečné připojení. Pro tento účel byl vybrán program PuTTY, umožňující se bezpečně připojit nejen ke vzdálenému linuxovému systému z operačního systému rodiny Windows.

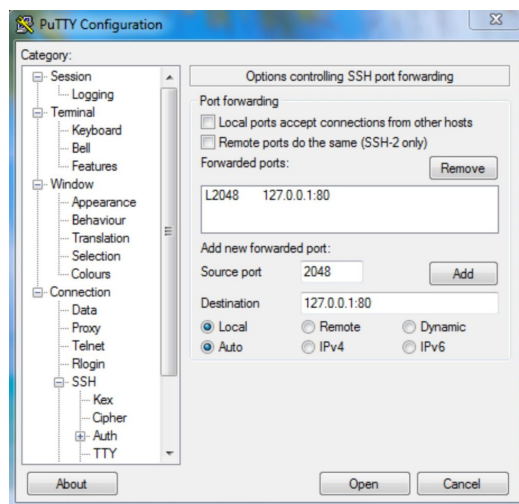


Obrázek 7: Nastavení údajů pro vytvoření SSH spojení v programu PuTTY

Při spuštění se nám zobrazí nabídka *Session*, ve které vyplníme požadované údaje, jak vidíme na obrázku č. 7, abychom dosáhli vytvoření klasického SSH spojení. Do políčka *Host Name (or IP address)* vložíme údaje (*example.cz*) o vzdáleném serveru, se kterým chceme navázat spojení. Odpovídající SSH démon nainstalovaný na serveru standardně naslouchá na portu 22, zda-li se s ním někdo nepokouší navázat SSH spojení. Poslední položkou, kterou v této sekci musíme vybrat je *Connection type* – SSH.

7.1.3 Vytvoření SSH tunelu

Pro přístup k webovému rozhraní monitorovacího systému Centreon instalovaného na vzdáleném serveru potřebujeme vytvořit SSH tunel zajišťující veškerou interakci mezi počítačem a serverem. Nastavení pro vytvoření SSH tunelu znázorňuje obrázek č. 8.

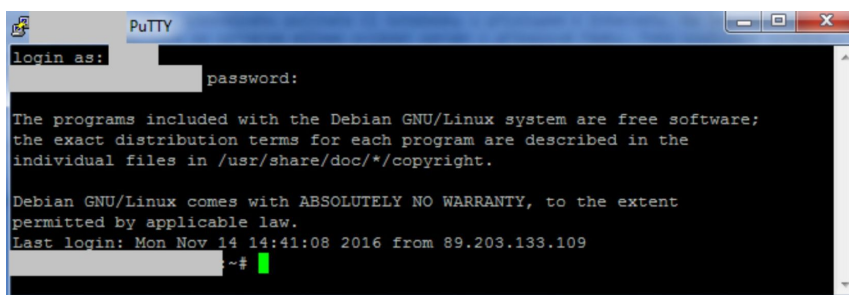


Obrázek 8: Nastavení údajů pro vytvoření SSH tunelu v programu PuTTY

Nastavení SSH tunelu probíhá v sekci *Connection – SSH – Tunnels*. Pro rozlišení jednotlivých služeb na počítači používají protokoly TCP a UDP tzv. síťové porty, rozdělené do tří skupin: (dobře) známé porty, registrované porty a dynamické a soukromé porty. Do pole *Source port* zadáme libovolné číslo (námi vybrané číslo 2048), vyskytující se v rozsahu portů jedné z těchto skupin, které není obsazeno jinou službou. Jelikož je monitorovací systém přímo na vzdáleném serveru, vyplníme políčko *Destination* adresou 127.0.0.1:80, což je IP adresa localhostu na serveru (právě používaný počítač). Číslo 80 za adresou localhostu odpovídá dobře známému portu služby HTTP využívané internetovým prohlížečem, přes který přistupujeme k webovému rozhraní monitorovacího systému Centreon. Kliknutím na tlačítko *Add* se vytvoří forwardovaný port, kterým probíhá komunikace. Abychom nemuseli tyto údaje zadávat při každém spuštění programu PuTTY, lze si je uložit zadáním názvu v sekci *Saved Sessions* (obrázek č. 7) a kliknutím na tlačítko *Save*.

7.1.4 Připojení ke vzdálenému serveru a webovému rozhraní Centreonu

Po definování údajů potřebných k vytvoření SSH spojení a SSH tunelu se nyní připojíme k serveru – jako terminál, kliknutím v programu PuTTY na tlačítko *Open* a zadáme přihlašovací údaje v podobě loginu a hesla, jak je uvedeno na obrázku č. 9. Po přihlášení můžeme server ovládat zadáváním příkazů do příkazového řádku.



Obrázek 9: Připojení ke vzdálenému serveru pomocí programu PuTTY

Po vytvoření SSH spojení se pokusíme přihlásit do prostředí Centreonu. Port 2048 na našem počítači čeká ve stavu LISTEN. Otevřeme si libovolný webový prohlížeč a do adresního řádku vložíme `http://localhost(nebo 127.0.0.1):2048` a stiskneme *Enter* na klávesnici. V tuto chvíli se požadavek odešle z počítače portem 2048 vzdálenému serveru na port 80 standardně používaného službou HTTP a otevře se webové prostředí Centreonu běžící na serveru v našem prohlížeči (obrázek č. 10).



Obrázek 10: Přihlášení se do monitorovacího systému Centreon

7.2 Instalace monitorovacího systému Centreon

Monitorovací systém Centreon je možné instalovat pouze na linuxových operačních systémech. Abych daný systém Centreon mohl nainstalovat i na svém počítači, rozhodl jsem se pro operační systém Debian, konkrétně distribuci Debian Wheezy, jehož instalaci zde nepopisuji. Následuje popis instalace monitorovacího systému Centreon za použití instalačních balíků typických pro tento operační systém.

Pro instalaci monitorovacího systému nejsou použity nejnovější komponenty z důvodu možné nestability systému. Po doporučení zadavatelem bakalářské práce Ing. Tomášem Holomkem byly použity již otestované a stabilní verze hlavních komponent: Centreon Web 2.5.2, Centreon Engine 1.4.6, Centreon Broker 2.6.3 a Centreon Clib 1.4.0.

Následuje detailní návod krok za krokem, jakým způsobem jednotlivé komponenty nainstalovat, a jehož výsledkem je funkční monitorovací systém Centreon. Veškerá instalace probíhá pod superuživatelem (root) zadáváním příkazů do terminálu. Pro instalaci byly použity tyto instalační návody: Centreon Web (Centreon, 2015j), Centreon Broker (Centreon, 2015d), Centreon Engine (Centreon, 2013f) a Centreon Clib (Centreon, 2015e).

1. Instalace prerekvizit

Operační systém Debian, jakožto balíčkovací systém, nemusí nutně obsahovat všechny potřebné balíčky (archive obsahující programy pro instalaci, odinstalaci a aktualizaci) pro instalaci našeho monitorovacího systému Centreon. Prvotním krokem je tedy instalace všech prerekvizit, které jsou monitorovacím systémem využívány.

```
root@Debian:~# apt-get install sudo tofrodos bsd-mailx lsb-release
mysql-server libmysqlclient15-dev apache2 apache2-mpm-prefork
php5 php5-mysql php-pear php5-ldap php5-snmp php5-gd rrdtool
librrds-perl libconfig-inifiles-perl libcrypt-des-perl libdigest-
-hmac-perl libdigest-sha-perl libgd-gd2-perl snmp snmpd libnet-
snmp-perl libsnmp-perl
```

2. Úprava souboru s repozitáři

Při instalaci prerekvizit může nastat situace, že instalovaný balík není dostupný, ale jiný balík se na něj odkazuje. Jinými slovy balík nemá kandidáta pro instalaci. Tento případ je typickou ukázkou chybějících cest k repozitářům (servery odkud mohou být staženy a nainstalovány potřebné balíčky). Základní repozitáře jsou dostupné v každé linuxové distribuci. Pro distribuci Debian jsou umístěny v `/etc/apt/sources.list`, který můžeme otevřít libovolným textovým editorem. Abychom mohli nainstalovat námi potřebné balíky, upravíme soubor do následující podoby a uložíme.

```
deb http://ftp.cz.debian.org/debian/ wheezy main
deb-src http://ftp.cz.debian.org/debian/ wheezy main

deb http://security.debian.org/ wheezy/updates main contrib
deb-src http://security.debian.org/ wheezy/updates main contrib

# wheezy-updates, previously known as 'volatile'
deb http://ftp.cz.debian.org/debian/ wheezy-updates main contrib
deb-src http://ftp.cz.debian.org/debian/ wheezy-updates main
      contrib
deb http://http.debian.net/debian/ wheezy main contrib non-free
deb http://http.debian.net/debian wheezy-backports main

# squeeze backport because of php
deb http://ftp.debian.org/debian/ squeeze main contrib non-free
deb http://security.debian.org/ squeeze/updates main contrib non-
  free
```

K dokončení je nutné nainstalovat SNMP MIB, neboť soubory MIB nejsou k dispozici ve výchozím nastavení Debianu. Zadáme proto následující příkazy:

```
root@Debian:~# apt-get update
```

```
root@Debian:~# apt-get install snmp-mibs-downloader
```

Nyní již přejdeme k instalaci monitorovacího systému.

3. Instalace Centreon Engine

Vytvoření uživatele a skupiny

K dispozici jsou dva způsoby, kterými lze provést instalaci Centreon Engine. Instalace pomocí oficiálních balíčků z CMS (Centreon Enterprise Server) úložiště, nebo instalace použitím zdrojů. V našem případě použijeme druhou variantu a je tudíž nutné splnit prerekvizity v podobě vytvoření uživatele systému a skupiny pro bezpečnostní účely.

```
root@Debian:~# groupadd centreon-engine
```

```
root@Debian:~# useradd -g centreon-engine -m -r -d /var/lib/
  centreon-engine centreon-engine
```

Upozornění: Vytvořený uživatel a skupina budou použity i v dalších krocích, proto se nedoporučuje měnit jejich název.

Instalace prerekvizit a Centreon Clib

Důležitou součástí je instalace balíčků *build-essential* a *cmake* pro vytvoření kompilačního prostřední programovacího jazyka C++ a přečtení zdrojů pro kompilaci.

```
root@Debian:~# apt-get install build-essential cmake
```

Následuje instalace knihovny Centreon Clib společně pro všechny Centreon produkty napsané v jazyce C/C++. Z oficiálních stránek download.centreon.com si stáhneme požadovanou verzi v archivu, pro jehož rozbalení použijeme následný příkaz (ve složce, do které byl tento soubor stažen).

```
root@Debian:/home/xhonek/Stazene# tar xzf centreon-clib-1.4.0.tar.gz
```

Nyní pokračujeme konfigurací Centreon Clibu, jehož konfigurační skripty se nachází v adresáři *build*. Konfiguraci provedeme příkazem *cmake* a nastavením proměnných, jak uvádí následující kód:

```
root@Debian:/home/xhonek/Stazene/centreon-clib-1.4.0/build# cmake \
-DWITH_TESTING=0 \
-DWITH_PREFIX=/usr \
-DWITH_PREFIX_LIB=/usr/lib \
-DWITH_PREFIX_INC=/usr/include/centreon-clib \
-DWITH_SHARED_LIB=1 \
-DWITH_STATIC_LIB=0 \
-DWITH_PKGCONFIG_DIR=/usr/lib/pkgconfig .
```

V tomto kroku probíhá kontrola existence a využitelnosti prerekvizit. Pokud zde něco není v pořádku, zobrazí se odpovídající chybové hlášení. Pokud je instalace úspěšná, zobrazí se její souhrnné informace.

Jakmile konfigurace proběhne v pořádku, provedeme kompilaci (příkaz *make*) a dokončíme instalaci (příkaz *make install*).

Tímto je dokončena instalace Centreon Clibu jakožto prerekvizita Centreon Engine, v jehož instalaci budeme nyní pokračovat.

Centreon Engine - dokončení instalace

Nyní stáhneme požadovanou verzi Centreon Engine 1.4.6 opět z oficiálních stránek, poté soubor stejným způsobem rozbalíme a pokračujeme na jeho konfiguraci do složky *build*, kde zadáme následující příkaz:

```
root@Debian:/home/xhonek/Stazene/centreon-engine-1.4.6/build# cmake \
> -DWITH_PREFIX=/usr \
> -DWITH_PREFIX_BIN=/usr/sbin \
> -DWITH_PREFIX_CONF=/etc/centreon-engine \
> -DWITH_PREFIX_LIB=/usr/lib/centreon-engine \
> -DWITH_USER=centreon-engine \
> -DWITH_GROUP=centreon-engine \
> -DWITH_LOGROTATE_SCRIPT=1 \
> -DWITH_VAR_DIR=/var/log/centreon-engine \
> -DWITH_RW_DIR=/var/lib/centreon-engine/rw \
> -DWITH_STARTUP_DIR=/etc/init.d \
> -DWITH_PKGCONFIG_SCRIPT=1 \
> -DWITH_PKGCONFIG_DIR=/usr/lib/pkgconfig \
> -DWITH_TESTING=0
```

Posledními dvěma kroky jsou kompilace příkazem *make* a instalace Centreon Engine příkazem *make install*.

4. Instalace Centreon Broker

Vytvoření uživatele a skupiny

Při instalaci ze zdrojů je stejně jako u Centreon Engine doporučeno z bezpečnostních důvodů vytvořit uživatele a skupinu.

```
root@Debian:~# groupadd centreon-broker
```

```
root@Debian:~# useradd -g centreon-broker -m -r -d /var/lib/
  centreon-broker centreon-broker
```

Upozornění: Vytvořený uživatel a skupina budou použity i v dalších krocích, proto se nedoporučuje měnit jejich název.

Instalace prerekvizit

V rámci prerekvizit je nutné nainstalovat: kompilační nástroje, RRDTool, Qt framework a GnuTLS následujícím příkazem:

```
root@Debian:~# apt-get install build-essential cmake librrd-dev
  libqt4-dev libqt4-sql-mysql libgnutls28-dev
```

Centreon Broker - dokončení instalace

Příslušná verze Centreon Broker 2.6.3 je stažena z oficiálních stránek, rozbalena příkazem *tar xzf centreon-broker-2.6.3.tar.gz* a abychom mohli pokračovat konfigurací, přistoupíme do složky *build*. Konfigurace se provede následujícím příkazem.

```
root@Debian:/home/xhonek/Stazene/centreon-broker-2.6.3/build# cmake
\
>   -DWITH_DAEMONS='central-broker;central-rrd' \
>   -DWITH_GROUP=centreon-broker \
>   -DWITH_PREFIX=/usr \
>   -DWITH_PREFIX_BIN=/usr/sbin \
>   -DWITH_PREFIX_CONF=/etc/centreon-broker \
>   -DWITH_PREFIX_INC=/usr/include/centreon-broker \
>   -DWITH_PREFIX_LIB=/usr/lib/nagios \
>   -DWITH_PREFIX_MODULES=/usr/share/centreon/lib/centreon-broker
\
>   -DWITH_STARTUP_DIR=/etc/init.d \
>   -DWITH_STARTUP_SCRIPT=auto \
>   -DWITH_TESTING=0 \
>   -DWITH_USER=centreon-broker .
```

Nakonec provedeme kompilaci příkazem *make* a instalaci Centreon Engine příkazem *make install*.

5. Instalace Centreon Web

V této fázi instalačního procesu jsou všechny potřebné součásti nainstalovány. Nyní se pustíme do instalace samotného monitorovacího systému Centreon. Nejdříve si stáhneme Centreon 2.5.2 z oficiálních stránek, rozbalíme ho, přesuneme se do jeho složky a zahájíme instalaci příkazem:

```
root@Debian:/home/xhonek/Stazene/centreon-2.5.2# ./install.sh -i
```

Jestliže jsou všechny prerekvizity správně nainstalovány, neměli bychom mít žádný problém v této fázi instalace. Dále je zachycen pouze výběr nejdůležitějších kroků instalace Centreonu. Podrobný popis celé instalace je uvedený v instalačním návodu Centreon Web.

```
Do you accept GPL license ?
[y/n], default to [n]:
> y
```

Instalace hlavních komponent:

```
-----
                Please choose what you want to install
-----

Do you want to install : Centreon Web Front
[y/n], default to [n]:
> y

Do you want to install : Centreon CentCore
[y/n], default to [n]:
> y

Do you want to install : Centreon Nagios Plugins
[y/n], default to [n]:
> y

Do you want to install : Centreon Snmp Traps process
[y/n], default to [n]:
> y
```

Definice instalačních cest:

```
-----
                Start CentWeb Installation
-----

Where is your Centreon directory?
default to [/usr/local/centreon]
> /usr/share/centreon
Path /usr/share/centreon                                OK

Where is your Centreon log directory
default to [/usr/local/centreon/log]
```



```
> /usr/share/centreon
Path /usr/share/centreon OK

Where is your Centreon etc directory
default to [/etc/centreon]
> /var/log/centreon
Path /var/log/centreon OK

Where is your Centreon binaries directory
default to [/usr/local/centreon/bin]
> /var/log/centreon
Path /var/log/centreon OK

Where is your Centreon data informations directory
default to [/usr/local/centreon/data]
> /usr/share/centreon/data
Path /usr/share/centreon/data OK
```

Zadání uživatele pro centreon-engine a centreon-broker:

```
What is the Monitoring engine user ?
> centreon-engine

What is the Broker user ? (optional)
> centreon-broker

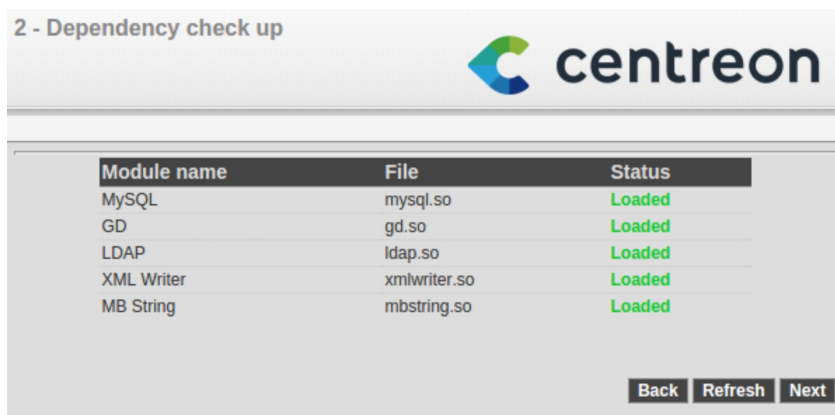
What is the Monitoring engine log directory ?
> /var/log/centreon-engine
```

Další nastavení méně důležitých kroků nalezneme v instalačním návodu Centreon Web.

6. Dokončení instalace pomocí webového prohlížeče

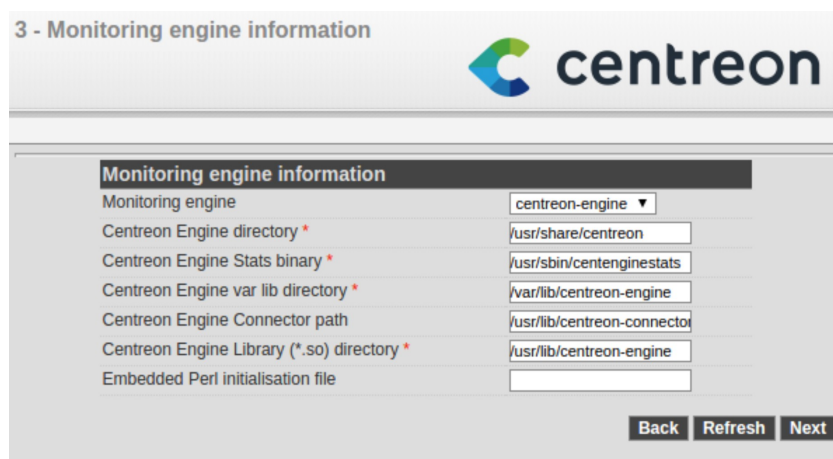
Po dokončení instalace Centreon Web bychom se měli ujistit, že Apache a MySQL server jsou v provozu předtím, než budeme pokračovat.

Instalaci dokončíme v grafickém rozhraní dostupném otevřením libovolného webového prohlížeče. Zde zadáme adresu *http://127.0.0.1(nebo localhost)/centreon* a zobrazí se nám úvodní instalační okno. Pokračujeme kliknutím na tlačítko *Next*.



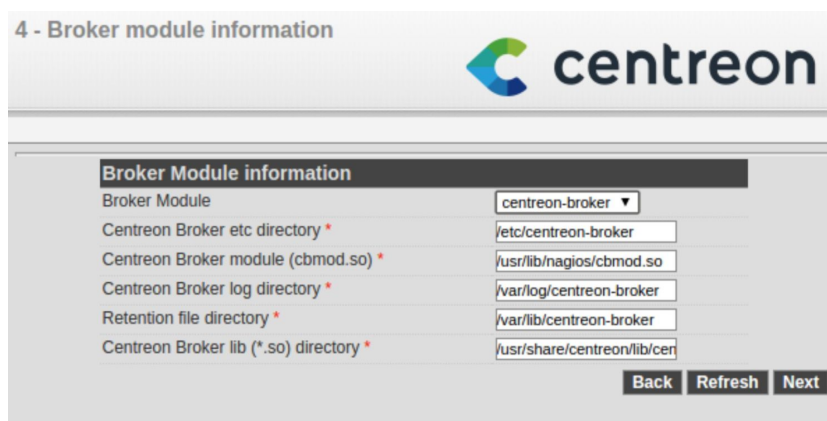
Obrázek 11: Centreon – ověřování závislostí

Kontrola všech potřebných nainstalovaných balíčků vyjadřuje status *Loaded* a pokračuje se tlačítkem *Next*. V opačném případě jsou chybějící balíčky doinstalovány a tlačítkem *Refresh* jejich stav obnoven a pokud je vše v pořádku pokračujeme v instalaci.



Obrázek 12: Centreon – volba monitoring engine

V prvním kroku zvolíme *centreon-engine* a díky tomu se nám zobrazí další políčka s výchozími hodnotami cest. Abychom mohli pokračovat dále, musíme požadované cesty (označené hvězdičkou) nastavit stejně jako na obrázku č. 12. Při nastavení špatné cesty se nám zobrazí chybová hláška a nelze přejít na další krok instalace. V opačném případě pokračujeme tlačítkem *Next*.



4 - Broker module information

centreon

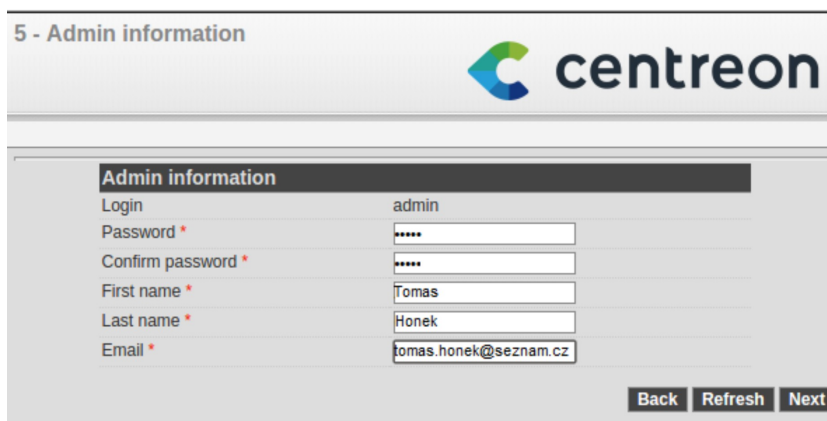
Broker Module information

Broker Module	centreon-broker
Centreon Broker etc directory *	/etc/centreon-broker
Centreon Broker module (cbmod.so) *	/usr/lib/nagios/cbmod.so
Centreon Broker log directory *	/var/log/centreon-broker
Retention file directory *	/var/lib/centreon-broker
Centreon Broker lib (*.so) directory *	/usr/share/centreon/lib/cer

Back Refresh Next

Obrázek 13: Centreon – volba broker module

Stejným způsobem upravujeme výchozí hodnoty cest po zvolení *centreon broker* modulu podle obrázku č. 13. Po správném nastavení přejdeme tlačítkem *Next* na další krok týkající se vyplnění údajů o administrátorském účtu.



5 - Admin information

centreon

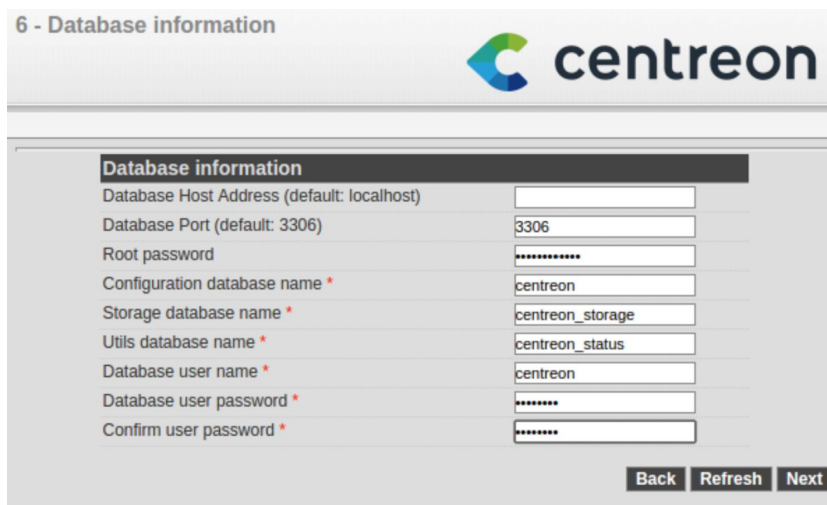
Admin information

Login	admin
Password *	*****
Confirm password *	*****
First name *	Tomas
Last name *	Honek
Email *	tomas.honek@seznam.cz

Back Refresh Next

Obrázek 14: Centreon – vyplnění údajů o administrátorovi

U administrátorského účtu je požadováno vyplnění následujících informací: heslo (včetně ověření opětovným zadáním hesla), jméno, příjmení a e-mail. Inspirovat se lze obrázkem č. 14. Tlačítkem *Next* přistoupíme k vyplnění údajů o databázi monitorovacího systému Centreon.



6 - Database information

Database information

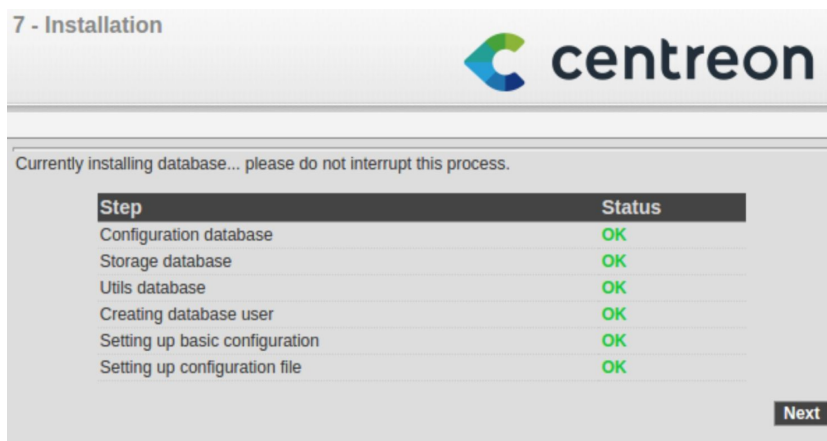
Database Host Address (default: localhost)	<input type="text"/>
Database Port (default: 3306)	3306
Root password	*****
Configuration database name *	centreon
Storage database name *	centreon_storage
Utils database name *	centreon_status
Database user name *	centreon
Database user password *	*****
Confirm user password *	*****

Back Refresh Next

Obrázek 15: Centreon – vyplnění údajů o databázi

Položky s výchozími hodnotami ponecháme beze změny. Pro vytvoření uživatele databáze zadáme pouze tyto údaje: login uživatele, heslo (včetně ověření opětovným zadáním hesla) a také heslo roota (superuživatele) systému. Zobrazení možného zadání údajů můžeme vidět na obrázku č. 15.

Tímto již máme všechny potřebné informace nastaveny a kliknutím na tlačítko *Next* spustíme instalaci databáze. V některých případech může proces instalace požadovat změnu nastavení serveru MySQL přidáním příkazu `innodb_file_per_table = 1` do konfiguračního souboru. Instalace je úspěšně dokončena pokud u provedených operací, jako na obrázku č. 16, svítí zeleně stav *OK*.



7 - Installation

Currently installing database... please do not interrupt this process.

Step	Status
Configuration database	OK
Storage database	OK
Utils database	OK
Creating database user	OK
Setting up basic configuration	OK
Setting up configuration file	OK

Next

Obrázek 16: Centreon – instalace databáze

Posledním krokem potvrzujeme kliknutím na tlačítko *Finish* ukončení instalačního procesu a jsme přesměrováni na přihlašovací okno monitorovacího systému Centreon, do kterého se můžeme za použití loginu a hesla přihlásit.

7.3 Implementace pluginu pro monitoring tiskáren

Cílem je navázat na kapitolu „6 Návrh řešení“, ve které byla nastíněna obecná funkce pluginu a pro lepší pochopení navržen i vývojový diagram se striktním popisem všech stavů, které mohou nastat na cestě od spuštění pluginu až po získání výsledků monitoringu. Nyní jsou jednotlivé stavy vývojového diagramu pomocí programovacího jazyka Perl převedeny do programového kódu a ty nejdůležitější z nich podrobně vysvětleny. Stavům vývojového diagramu odpovídají dále tučně zvýrazněné nadpisy. Informace, které nemohou být ve zdrojovém kódu zobrazeny jsou nahrazeny obecným řetězcem. Naprogramovaný plugin je k dispozici v příloze.

Definice argumentů a SNMP atributů

Pro správnou funkci pluginu musí být uživatelem zadány potřebné argumenty a SNMP atributy, jak bylo ukázáno v návrhu řešení u stejnojmenného názvu stavu.

```
if (!@ARGV) { #bez zadanych parametru
  usage();   #napoveda pouziti
  exit(3);   #hodnota exit kodu
};
```

Jestliže uživatel spustí plugin bez parametrů, zůstane pole *@ARGV*, kterým se všechny argumenty předávají do pluginu, prázdné, uživateli se zobrazí nápověda a návratové hodnotě exit kódu odpovídá stav UNKNOWN.

Jakmile jsou parametry zadány a skript je spuštěn, dochází k jejich ukládání do hashe *%inputValues*, jehož klíčem je řetězec názvu předávaného parametru uvedený ve složených závorkách a hodnotou předaná hodnota polem *@ARGV*. V této hash struktuře jsou uloženy všechny parametry zadané při spuštění skriptu, abychom je později mohli použít.

```
while (@ARGV){
  switch(shift){
    case '-mode:' {
      $inputValues{mode} = lc(shift);}
    case '-w:' {
      my $value = shift;
      if((0<=$value) && ($value<=100)){
        $inputValues{warning} = $value;
      }else{
        $inputValues{warning} = 20;
      }
    }
    case '-c:' {
      my $value = shift;
      if((0<=$value) && ($value<=100) && ($value<$inputValues{warning})){
        $inputValues{critical} = $value;
      }else{
        $inputValues{critical} = 10;
      }
    }
    case '-hostname:' {
      $inputValues{hostname} = shift;
    }
  }
}
```

```

        case '-snmp_community:' {
            $inputValues{snmpCommunity} = shift;
        }
        case '-snmp_version:' {
            $inputValues{snmpVersion} = lc(shift);
        }
    }
}
# vychozi nastaveni hodnot pro warning a critical
$inputValues{warning} = 20 if (!exists $inputValues{warning});
$inputValues{critical} = 10 if (!exists $inputValues{critical});

```

Při zadávání hodnot potřebujeme mít parametry (mód a verze SNMP protokolu) v předem stanoveném tvaru, tzn. kontrolovat vstupy zadané uživatelem, k čemuž nám zde slouží funkce *lc* (lowercase) pro převod všech písmen argumentu na malá.

Pro stavy *WARNING* a *CRITICAL* si může uživatel nastavit hodnoty (v rozmezí 0–100 %) dle vlastního uvážení, přičemž hodnota stavu *WARNING* musí být větší než pro stav *CRITICAL*. Aby bylo dodrženo rozmezí hodnot a předchozí podmínka, jsou hodnoty těchto parametrů předány nejdříve do proměnné *\$value* ke kontrole. V případě splnění podmínky se hodnoty uloží do hashe *%inputValues*, při jejím nesplnění jsou nastaveny na výchozí hodnoty 20 a 10.

Nesprávné zadání SNMP atributů, Zobrazení nápovědy použití, Předání stavu UNKNOWN monitorovacímu systému, Ukončení pluginu

Pro navázání spojení s tiskárnou pomocí protokolu používaného pro správu sítě SNMP, se používá modul *Net::SNMP*. Po zavolání konstruktoru *SNMP::Session* dochází k zapouzdření získaných parametrů uložených ve struktuře hash *%inputValues* pro navázání spojení se zařízením pomocí správné SNMP verze a komunity a IP adresy daného zařízení.

```

# SNMP relace, kontrola jestli je spojeni navazano nebo ne
my ($session, $error) = Net::SNMP->session(
    -community => $inputValues{snmpCommunity},
    -hostname   => $inputValues{hostname},
    -version    => $inputValues{snmpVersion},
);

# spojeni neni navazano
if (!defined $session) {
    printf "Error SNMP: %s.\n", $error;
    usage();
    exit(3);
}

```

V situacích (chybně zadané argumenty, nedostupnost zařízení), kdy nedochází k navázání spojení *SNMP::Session* s monitorovanou tiskárnou, se uživateli zobrazí chybové oznámení, nápověda, exit kód (stav UNKNOWN) a plugin je následně ukončen.

Správné zadání SNMP atributů, Spuštění pluginu, Inicializace proměnných a konstant

Po správném zadání všech argumentů a navázání spojení s tiskárnou můžeme přistoupit k získávání informací o monitorovaném typu tiskárny.

Na počátku jsou definovány všechny proměnné a konstanty, s pomocí kterých později přistoupíme ke specifickým informacím tiskárny nebo k ukládání výsledků monitoringu.

```
# struktura pro ulozeni chybovych komponent
my %states = (
    OK => {},
    WARNING => {},
    CRITICAL => {}
);
#celkovy stav tiskarny
my $overallState;
```

Pro ukládání výsledků stavů monitorovaných komponent je vytvořena vhodná datová struktura obsahující tři stavy: *OK*, *WARNING* a *CRITICAL*. Například pokud se v rámci monitoringu zjistí, že určitá komponenta tiskárny vykazuje závažnou chybu, je uložena do datové struktury pro stav *CRITICAL*. Základní informace o tiskárně a zjištěné výsledky monitoringu dle provedeného módu (chybové stavy, komponenty s nízkou kapacitou spotřebního materiálu či informace o celkovém počtu vytisknutých stran) jsou ukládány do proměnné *\$overallState*, která je výstupem monitoringu pro všechny módy.

Získávání informací o tiskárnách je realizován na základě přístupu k jejich MIB tabulkám za pomoci OID čísel. V datové struktuře *%listOfOIDs* jsou obsažena všechna OID čísla použitelná ke kontrole aktuálního stavu tiskárny pro jakýkoliv mód zvolený uživatelem.

```
# struktura pro ulozeni vseh pouzitych OID cisel
# nazvy klicu struktury %listOfOIDs jsou vymyslene
my %listOfOIDs = (
    printerDescription           => 'x.x.x.x.x',
    serialNumberOfPrinter        => 'x.x.x.x.x',
    errorCondition                => 'x.x.x.x.x',
    errorDescription              => 'x.x.x.x.x',
    descriptionOfComponents      => 'x.x.x.x.x',
    stateOfComponents            => 'x.x.x.x.x',
    maximumCapacityOfComponents => 'x.x.x.x.x',
    totalPrintedPages            => 'x.x.x.x.x',
);
```

Klíči datové struktury jsou programátorem vymyšlené názvy popisující, ke kterým částem MIB tabulek či konkrétním objektům se hodnoty OID čísel vztahují. Protože není možné zveřejnit OID čísla z dokumentací výrobce tiskáren Konica Minolta, jsou tato čísla nahrazena tvarem obecných řetězců.

Identifikace tiskárny, Zvolení požadovaného módu, Specifikace všech použitých OID, Přístup ke specifickým MIB tabulkám podle typu tiskárny, Kontrola komponenty tiskárny, Vyhodnocení stavu komponenty, Uložení závadné komponenty do chybových stavů

Pro sledování specifických komponent zařízení je možné plugin spouštět ve více režimech (jejich názvy jsou zvolené programátorem). Mezi hlavní přínosy módů patří větší přehlednost pro uživatele, který si tímto krokem určí pro něho významné komponenty monitoringu, lépe strukturovaný zdrojový kód a rychlejší dobu kompilace pluginu při spuštění.

Přehled jednotlivých módů:

- *errors* – kontrola aktuálního stavu všech komponent tiskárny (probíhá výpisem varovných a kritických chyb následovaného popisem vzniklého problému)
- *consumables* – kontrola spotřebního materiálu včetně souvisejících komponent (tonery, válce, kazety a jiné)
- *counters* – kontrola celkového počtu vytisknutých stránek tiskárnou (součet celkového počtu vytisknutých a kopírovaných stránek)

Závěrem je nutné zdůraznit, že výše zmíněné módy jsou použitelné pro všechny typy monitorovaných tiskáren.

Nejdříve se zaměříme na zjištění základních informací o tiskárně. K tomuto účelu jsme použili podprogram *getBasicInfo()*.

```
# ulozeni zakladnich informaci o tiskarne
$overallState = "xxxxx: " . getBasicInfo(xxxxx) . ", xxxxx: " . getBasicInfo(
    xxxxx) . ", ";

# FUNKCE PODPROGRAMU: navraci konkretni informaci ziskanou z MIB tabulky
zarizeni
# VSTUP => OID cislo tabulky, ze ktere chceme prislusnou informaci ziskat
# VYSTUP <= Informace ziskana z MIB tabulky zarizeni.
sub getBasicInfo{
    my $oidNumber = shift;
    # ulozeni specificke informace ziskane pozadavkem get_request podle
    vlozeneho OID cisla
    my $result = $session->get_request(-varbindlist => [ $oidNumber ]);
    # navraceni ziskane informace (hodnota hashe)
    return @{$result}{$oidNumber};
}
```

Do proměnné *\$overallState* jsou postupně ukládány základní informace o tiskárně složené z názvů a hodnot získaných po provedení podprogramu *getBasicInfo()*, kterému je předloženo OID číslo tabulky, ze které chceme příslušnou informaci o zařízení získat.

Po získání základních informací o tiskárně následuje monitoring specifických komponent tiskárny podle zvoleného módu.

```
# provedeni monitoringu podle vstupniho modu
switch($inputValues{mode}){
  case 'errors' {
    # kontrola chybovych stavu tiskarny
    checkErrorStates(descriptions => getListOfComponents(xxxxx),
                    states       => getListOfComponents(yyyyy));

    # vypis chyb v textove podobě, predani stavu monitorovacimu systemu a
    ukonceni pluginu
    getOverallState(warningErrors => getNumberOfErrors("WARNING"),
                  criticalErrors => getNumberOfErrors("CRITICAL"));
  }
  case 'consumables' {
    # kontrola stavu komponent spotrebnih materialu
    checkConsumables(componentDescriptions=> getListOfComponents(xxxxx),
                    componentStates     => getListOfComponents(yyyyy));

    # vypis chyb v textove podobě, predani stavu monitorovacimu systemu a
    ukonceni pluginu
    getOverallState(warningErrors => getNumberOfErrors("WARNING"),
                  criticalErrors => getNumberOfErrors("CRITICAL"));
  }
  case 'counters' {
    # kontrola celkoveho poctu vytisknutych stran a jejich vypis
    checkPrintedPages(getListOfComponents(xxxxx));
  }
  else {print "You entered unknown mode name!";
        usage(); # napoveda pouziti
        exit(3); # hodnota exit kodu
  }
}
```

Nyní se zaměříme na analýzu kontroly chybových stavů tiskárny a komponent spotřebního materiálu prováděnou podprogramy *checkErrorStates()* a *checkConsumables()*. Po jejich vysvětlení následuje rozbor podprogramu *getOverallState()*, který vypisuje všechna chybová hlášení a jejich stav v textového podobě, předá je monitorovacímu systému a ukončí plugin.

Při kontrole chybových stavů tiskárny jsou podprogramu předloženy dva parametry. Hodnotami těchto předávaných parametrů jsou tabulky informací obsahující názvy chybových stavů a stupně závažnosti chyb (jde-li o chybu menšího rozsahu či chybu kritickou) vykazovaných tiskárnou. Příslušné tabulky informací jsou získány podprogramem *getListOfComponents()* na základě konkrétního OID čísla.

```
# FUNKCE PODPROGRAMU: navruci tabulku informaci ze zarizeni
# VSTUP => OID cislo MIB tabulky, kterou chceme ziskat
# VYSTUP <= MIB tabulka informaci - struktura hash
sub getListOfComponents{
  my $oid = shift;
  # ziskani tabulky informaci ze zarizeni pozadavkem get_table podle
  vlozeneho OID cisla
  my $result = $session->get_table(-baseoid => $oid);
  return $result;
}
```

```

# FUNKCE PODPROGRAMU: kontrola chybovych stavu tiskarny
# VSTUP => $params{descriptions} - tabulka informaci obsahujici nazvy
#         chybovych stavu - hash struktura
#         => $params{states} - tabulka informaci obsahujici stupne zavaznosti
#         chyb - hash struktura
# VYSTUP <= Neni zadny, podprogram pouze ulozi chybove stavy do struktury
#           %states.
sub checkErrorStates{
  my %params = @_;
  my $descriptions = $params{descriptions};
  my $states = $params{states};

  while (each %{$states}) {
    # separace cisla chyboveho stavu - ulozeno v parametrech $1 a $3
    $_ =~ /((\d)*)\.((\d)*)/;
    # ukladani nazvu chyby do %states podle stupne zavaznosti chyby
    switch (${$states}{$_}){
      case x {
        #prirazeni nazvu chyby k jejimu cislu
        $states{CRITICAL}{"$1.$3"} = ${$descriptions}{xxxxx . "$1.$3"};
      }
      case x {
        #prirazeni nazvu chyby k jejimu cislu
        $states{WARNING}{"$1.$3"} = ${$descriptions}{xxxxx . "$1.$3"};
      }
    }
    # tiskarna nachazejici se v uspornem rezimu - neni povazovano za chybu
    if((grep {"xxxxx" eq $_} $states{WARNING}{"$1.$3"}) ||
      (grep {"xxxxx" eq $_} $states{WARNING}{"$1.$3"})){
      delete $states{WARNING}{"$1.$3"};
    }
  }
};

```

Celý proces kontroly probíhá v cyklu, kdy jsou jednotlivé chybové stavy tiskárny postupně zpracovány a podle identifikace stupně jejich závažnosti uloženy do struktury `%states` uchováající chybové stavy. Nejdříve zjistíme číslo chybového stavu tiskárny. V dalším kroku rozlišíme, o jak závažnou chybu se jedná. Nakonec k číslu chybového stavu tiskárny přiřadíme její název a společně uložíme do struktury `%states` pro určený typ chyby. Chyba tiskárny nacházející se v úsporném režimu není považována za chybu. V případě jejího nalezení je z této datové struktury vymazána.

Nyní přejdeme na vysvětlení podprogramu používaného pro kontrolu komponent spotřebního materiálu.

```
# FUNKCE PODPROGRAMU: kontrola stavu komponent spotřebního materiálu
# VSTUP => $params{componentDescriptions} - tabulka informací obsahující názvy
#         komponent - hash struktura
#         => $params{componentStates} - tabulka informací obsahující aktuální
#         hodnoty komponent spotřebního materiálu - hash struktura
# VYSTUP <= Není zadný, podprogram pouze uloží chybové stavy do struktury
#           %states.
sub checkConsumables{
    my %params = @_;
    my $descriptionsOfComponents = $params{componentDescriptions};
    my $statesOfComponents = $params{componentStates};
    my $maxCapacityOfComponent;
    my $percentageValue;

    while (each %{$statesOfComponents}) {
        # separace čísla komponenty - uloženo v parametru $1
        $_ =~ /((\d)*)/g;
        # maximální kapacita komponenty
        $maxCapacityOfComponent = maximumCapacity(oid => xxxxx, numberOfComponent
            => $1);
        # aktuální hodnota spotřebního materiálu komponenty převedena na procenta
        $percentageValue = convertValue(actualValue => %{$statesOfComponents}{$_},
            maxCapacity => $maxCapacityOfComponent, numberOfComponent => $1);
        # ukládání názvu komponenty do %states podle aktuálního množství spotřebního
        # materiálu komponenty
        switch ($percentageValue){
            case ['xxxxx', 'xxxxx', $inputValues{warning}..100] {
                $states{OK}{ "$1" } = %{$descriptionsOfComponents}{xxxxx} . "$1";
            }
            case ['xxxxx', $inputValues{critical}+1..$inputValues{warning}-1]{
                $states{WARNING}{ "$1" } = %{$descriptionsOfComponents}{xxxxx} . "$1";
            }
            case [0..$inputValues{critical}] {
                $states{CRITICAL}{ "$1" } = %{$descriptionsOfComponents}{xxxxx} . "$1";
            }
        }
    }
}
}
```

Podprogram přebírá, stejně jako u kontroly chybových stavů tiskárny, dva parametry, jejichž hodnotami (opět získané podprogramem *getListOfComponents()*) jsou tabulky informací uchováující názvy komponent a aktuální hodnoty spotřebního materiálu příslušných komponent. Jsou zde kontrolovány jednotlivé komponenty a podle zůstatku spotřebního materiálu jsou zařazeny do jednoho ze stavů OK, WARNING nebo CRITICAL ve struktuře chybových stavů *%states*. Komponenta může nabývat hodnot v rozmezí 0–100 % nebo pouze určitý stav. Procentní hodnoty pro výchozí nastavení jsou určeny takto: 100–20 % – stav OK, 19–11 % – stav WARNING a 10–0 % – stav CRITICAL. Výchozí nastavení nemusí být pro všechny uživatele vyhovující, proto si procentní rozmezí jednotlivých stavů může každý určit podle svých potřeb zadáním argumentů při spuštění pluginu. Aby mohla být

aktuální hodnota spotřebního materiálu příslušné komponenty převedena na procenta, potřebujeme znát její maximální hodnotu, kterou zjistíme podprogramem *maximumCapacity()*. Poté můžeme převést aktuální hodnotu komponenty podprogramem *convertValue()* na procenta, k čemuž využívá předané parametry obsahující aktuální hodnotu komponenty, maximální možnou kapacitu a číslo komponenty. Podprogramy zabývající se zjištěním maximální kapacity komponenty a převedením aktuální hodnoty na procenta již dále nejsou do podrobně vysvětleny.

Začneme zjištěním čísla kontrolované komponenty. Po zjištění procentuálního stavu komponenty výše zmíněnými podprogramy rozhodneme o přiřazení a uložení komponenty do jedné ze skupin chybových stavů struktury *%states*, kam k číslu komponenty přiřadíme její název.

Výpis základních informací o tiskárně, Předání stavu OK monitorovacímu systému, Předání chybového stavu WARNING nebo CRITICAL a výpisu chyb monitorovacím systémem, Předání stavu CRITICAL a výpisu chyb ve stavech CRITICAL a WARNING monitorovacím systémem, Zobrazení výsledků monitoringu ve webovém rozhraní nebo příkazové řádce

V této fázi již máme všechny chybové stavy tiskárny nebo komponenty spotřebního materiálu rozděleny a uloženy v datové struktuře *%states*. Poslední krok úspěšného monitoringu sestává z informování uživatele výpisem závadných komponent a předání stavu monitorovacímu systému podprogramem *getOverallState()*.

```
# FUNKCE PODPROGRAMU: vypis chyb v textove podobe, predani stavu monitorovacimu
#                       systemu a ukonzeni pluginu
# VSTUP => $params{warningErrors} - celkovy pocet chyb typu WARNING
#       => $params{criticalErrors} - celkovy pocet chyb typu CRITICAL
# VYSTUP <= Vypis vysledku monitoringu.
sub getOverallState{
    my %params = @_;
    my $numberOfWarningErrors = $params{warningErrors};
    my $numberOfCriticalErrors = $params{criticalErrors};

    if (($numberOfWarningErrors != 0) && ($numberOfCriticalErrors != 0)){
        $overallState .= 'States CRITICAL: ';
        foreach my $check_oid (keys %{$states{CRITICAL}}){
            $overallState .= $states{CRITICAL}{$check_oid} . ', ';
        }
        $overallState .= 'States WARNING: ';
        foreach my $check_oid (keys %{$states{WARNING}}){
            $overallState .= $states{WARNING}{$check_oid} . ', ';
        }
    };
    print $overallState;
    exit 2;
}elsif($numberOfCriticalErrors != 0){
    $overallState .= 'States CRITICAL: ';
    foreach my $check_oid (keys %{$states{WARNING}}){
        $overallState .= $states{CRITICAL}{$check_oid} . ', ';
    }
    print $overallState;
    exit 2;
}
```

```

elseif($numberOfWarningErrors != 0){
    $overallState .= 'States WARNING: ';
    foreach my $check_oid (keys %{$states{WARNING}}){
        $overallState .= $states{WARNING}{$check_oid} . ', ';
    }
    print $overallState;
    exit 1;
}elseif(($numberOfWarningErrors == 0) && ($numberOfCriticalErrors == 0)){
    $overallState .= 'Error status is OK.';
    print $overallState;
    exit 0;
}
}

```

Podprogramu jsou nejdříve předány hodnoty obsahující celkový počet chyb typu *WARNING* a *CRITICAL*, od nichž se odvíjí splnění jedné z příslušných podmínek a provedení příkazů jejího bloku.

Vše si nyní ukážeme na příkladu. Představme si, že jsou podprogramu předány parametry se dvěma závadnými komponentami chybových stavů *WARNING* a jednou komponentou chybového stavu *CRITICAL*. Dochází ke splnění podmínky *if ((\$numberOfWarningErrors != 0) AND (\$numberOfCriticalErrors != 0))* a provedou se příkazy jejího bloku. Na počátku jsme si do proměnné *\$overallState* vložili textový řetězec obsahující základní informace o tiskárně. K tomu nyní přidáme řetězec „States CRITICAL: “ následovaný popisem závadné komponenty tohoto typu uložené ve *\$states{CRITICAL}*. Stejným způsobem připojíme řetězec „States WARNING: “ a dvě závadné komponenty typu *WARNING*. Nakonec se celý obsah této proměnné vypíše, monitorovacímu systému se předá hodnota exit kódu 2 reprezentující závažnější chybový stav *CRITICAL* a plugin je následně ukončen. Dalšími možnými předanými exit kódy jsou exit 1 – stav *WARNING* nebo exit 0 – stav *OK*. Výsledky jsou pro uživatele zobrazitelné ve webovém rozhraní monitorovacího systému Centreon nebo pomocí příkazového řádku.

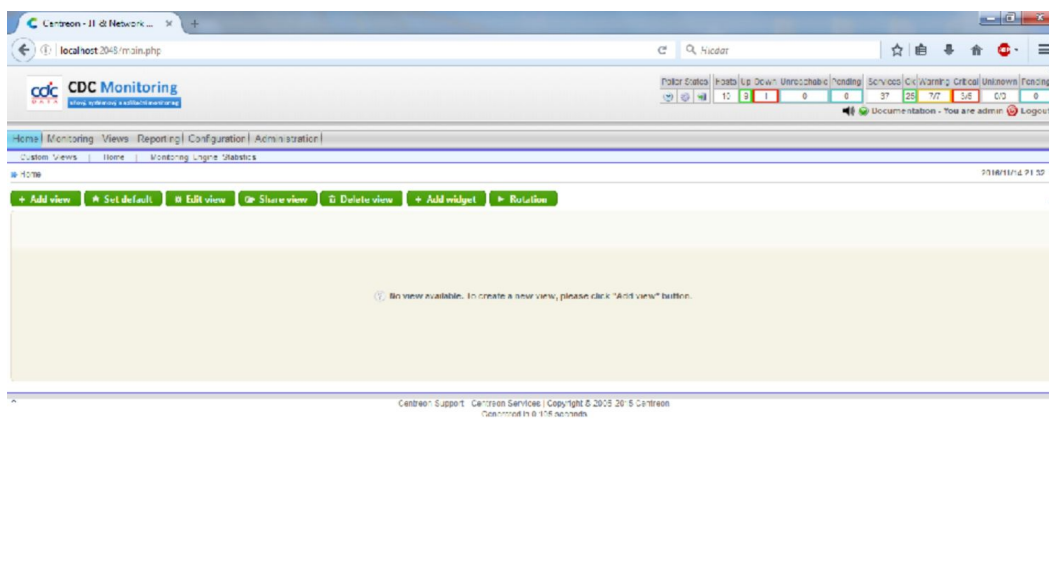
V této chvíli již mám popsány způsoby, kterými probíhá kontrola chybových stavů a spotřebního materiálu podle zvoleného módu zadaného uživatelem při spuštění pluginu. Jako poslední je vysvětlena kontrola celkového počtu vytisknutých stran podprogramem *checkPrintedPages()*. Aby nebyly prozrazeny informace z MIB tabulek zařízení, není u tohoto podprogramu zobrazena ukáзка zdrojového kódu a tento kód není ani součástí pluginu odevzdaného v příloze. Hlavní odlišnost oproti předchozím dvěma režimům spočívá v nepředávání chybového stavu monitorovacímu systému, tudíž stav nabývá vždy hodnoty *OK* a dochází pouze k výpisu výsledků v textové podobě. Při kontrole celkového počtu vytisknutých stran představuje klíčovou roli zjištění vytisknutých stran pro jednotlivé typy tisků (jednobarevný, dvoubarevný, čtyřbarevný a černobílý) tiskáren. Tiskárny s malým počtem funkcí obsahují pouze některé typy tisků, multifunkční všechny čtyři typy. Výstupem podprogramu *checkPrintedPages()* je informace pro uživatele o stavu celkového počtu vytisknutých stran jednotlivých typů tisku, za kterým následuje znak (|), a řetězec obsahující jednu nebo více metrik performance dat sloužící pro generování grafů.

Tímto jsme dokončili popis stavů vývojového diagramu implementovaných pluginem od spuštění po zobrazení výsledků monitoringu. Následuje kapitola věnující se integraci pluginu do monitorovacího systému Centreon.

7.4 Vložení pluginu do monitorovacího systému Centreon

Účelem podkapitoly je popsat způsob implementace naprogramovaného pluginu v grafickém prostředí monitorovacího systému Centreon. Popis se zaměřuje pouze na implementaci tiskárny Konica Minolta bizhub C220. Přidání ostatních typů tiskáren do monitorovacího systému Centreonu je téměř identickou záležitostí.

Na začátku vytvoříme SSH spojení, blíže popsané v kapitole „7.1 Vytvoření přístupu ke sledovaným zařízením a monitorovacímu systému Centreon“ k tomu, abychom se do monitorovacího systému mohli přihlásit zadáním přihlašovacích údajů (login a heslo) a byli připuštěni na úvodní stránku (obrázek č. 17).



Obrázek 17: Centreon – úvodní okno

7.4.1 Přidání monitorovaného zařízení

Monitorované zařízení přidáme do prostředí Centreon vytvořením nového hosta: přesuneme se do *Configuration – Hosts* a klikneme na tlačítko *Add*. Zobrazí se nám formulář v sekci *Host Configuration*, kde definujeme zvýrazněné části podle obrázků č. 18 a č. 19.

Modify a Host	
General Information	
Host Name *	Printer_Konica_Minolta_C220
Alias *	printer_C220
IP Address / DNS *	<input type="text"/> Resolve
SNMP Community & Version	public 2c ▼
Monitored from	Central ▼
Host Templates	Add a new entry + Nothing here, use the "Add" button
Create Services linked to the Template too	<input type="radio"/> Yes <input checked="" type="radio"/> No
Host Check Properties	
Check Period *	24x7 ▼
Check Command	check_host_alive ▼ ⓘ
Args	<input type="text"/>
Max Check Attempts *	5
Normal Check Interval	1 * 60 seconds
Retry Check Interval	1 * 60 seconds
Active Checks Enabled	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Default
Passive Checks Enabled	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Default

Obrázek 18: Centreon – vytvoření hosta v sekci General Information a Host Check Properties

Do pole *Host Name* vložíme název, pod kterým se nám tiskárna zobrazuje v prostředí Centreonu. Jako další vyplníme *Alias* a *IP address*, abychom se mohli na danou tiskárnu připojit. Pro získávání informací ze zařízení pomocí protokolu SNMP definujeme *SNMP Community* a *SNMP Version*. Výběrem z nabídek v *Check Period* a *Check Command* si určíme, po jakou dobu se monitoring provádí a výběrem *check_host_alive* kontrolujeme dostupnost zařízení. Číslo zadané v *Max Check Attempts* určuje počet provedených dotazů na zařízení před vyhodnocením jeho stavu zobrazením *OK*, *WARNING* nebo *CRITICAL*. V případě nedostupnosti se zobrazí stav *UNKNOWN*.

Notification					
Notification Enabled	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Default				
Contact additive inheritance	<input type="checkbox"/>				
Linked Contacts *	<table border="1"> <thead> <tr> <th>Available</th> <th>Selected</th> </tr> </thead> <tbody> <tr> <td>Admin_Admin Obsluha User</td> <td>Guest</td> </tr> </tbody> </table>	Available	Selected	Admin_Admin Obsluha User	Guest
Available	Selected				
Admin_Admin Obsluha User	Guest				
Contact group additive inheritance	<input type="checkbox"/>				
Linked Contact Groups *	<table border="1"> <thead> <tr> <th>Available</th> <th>Selected</th> </tr> </thead> <tbody> <tr> <td>Supervisors</td> <td>Guest</td> </tr> </tbody> </table>	Available	Selected	Supervisors	Guest
Available	Selected				
Supervisors	Guest				
Notification interval *	1 * 60 seconds				
Notification Period *	24x7 ▼				
Notification Options *	<input checked="" type="checkbox"/> Down <input checked="" type="checkbox"/> Unreachable <input checked="" type="checkbox"/> Recovery <input checked="" type="checkbox"/> Flapping <input type="checkbox"/> Downtime Scheduled				
First notification delay	<input type="text"/> * 60 seconds				

List Form

Obrázek 19: Centreon – vytvoření hosta v sekci Notification

Hlášení o monitoringu zařízení se odesílají uživateli, kterého v poli *Linked Contacts* a *Linked Contact Groups* vybereme. Na závěr definujeme *Notification Interval*, *Notification Period* a *Notification Options*, abychom věděli, jak často, po jakou dobu a jaké stavy se budou uživateli v hlášeních odesílat. Po ukončení nastavení klikneme na tlačítko *Save* a nový host se přidá do seznamu již vytvořených hostů.

7.4.2 Nastavení příkazu pro spuštění pluginu

Pro automatické spuštění pluginu monitorovacím systémem Centreon, musíme definovat příkaz jako při spuštění z příkazového řádku.

Abychom mohli tento krok provést, potřebujeme umístit naprogramovaný plugin do adresáře, ve kterém jsou na počátku uloženy pouze výchozí pluginy používané monitorovacím systémem. K tomu použijeme program WinSCP, jehož hlavní funkcí je bezpečný přenos souborů mezi počítačem a vzdáleným serverem. Plugin vložíme do adresáře */usr/lib/nagios/plugins*.

Pokračujeme nastavením příkazu pro spuštění pluginu v prostředí Centreon přesunem do sekce *Configuration – Commands*. Kliknutím na tlačítko *Add* se zobrazí formulář, který vyplníme podle obrázku č. 20.

Obrázek 20: Centreon – nastavení příkazu

Nastavení příkazu je uloženo v prostředí Centreonu pod názvem zadaném v poli *Command Name*. Pole *Command Line* je totožné s příkazovým řádkem, ze kterého jsme plugin spouštěli v implementační fázi. Definujeme cestu, název a parametry pluginu. Pro spuštění pluginu napsaného v programovacím jazyku Perl slouží slovo *perl*. Za ním následuje hodnota *\$USER1\$*/, určující cestu k adresáři, o které jsme se zmínili ve druhém odstavci této podkapitoly. Za adresář napíšeme název spouštěného pluginu, v našem případě *check_printers_konica_minolta.pl*. Poté už zadáváme pouze parametry pro správné navázání spojení se zařízením a parametry potřebné pro správnou funkci pluginu.

Existují dva způsoby zadávání parametrů. V prvním způsobu zadává parametry uživatel ručně při vytváření monitorovací služby. Druhý způsob spočívá v automatickém předávání parametrů, jejichž hodnoty byly dříve zadány uživatelem, monitorovacím systémem Centreon. Celý seznam standardních hodnot systému Centreon

nalezneme ve třetí rolovací nabídce na obrázku č. 20. U parametru `-mode:` se předává hodnota `$ARG1$` později definovaná uživatelem. Stejný princip platí pro další dva parametry `-w:` a `-c:` s hodnotami `$ARG2$` a `$ARG3$`. Ostatní hodnoty pro parametry jsou předávány systémem Centreon. Pro parametr `-hostname:` nalezneme a vložíme hodnotu `$HOSTADDRESS$` ze třetí rolovací nabídky. Tato hodnota obsahuje IP adresu monitorovaného zařízení, kterou jsme definovali při vytváření hosta. Parametry `-snmp_community:` a `-snmp_version:` přebírají standardně definované hodnoty `$_HOSTSNMPCOMMUNITY$` a `$_HOSTSNMPVERSION$` systémem Centreon, jejichž hodnoty byli stejně jako IP adresa zadány při vytváření hosta.

Z důvodu přehlednosti při předávání více hodnot uživatelem současně slouží pole *Argument Descriptions*. Zde si ke každé hodnotě slovně napíšeme její význam. Tímto je vytvoření příkazu pro spuštění pluginu ukončeno, zbylá políčka ponecháme ve výchozím nastavení a vše uložíme kliknutím na tlačítko *Save*.

7.4.3 Konfigurace služby v monitorovacím systému

V současné chvíli máme nastaveny všechny potřebné prerekvizity a nic nám nebrání ve vytvoření služby monitorující aktuální stav zařízení. Přistoupíme tedy do sekce *Configuration – Services* a kliknutím na tlačítko *Add* zobrazíme formulář pro vytvoření nové služby. Jednotlivá políčka vyplníme podle obrázků č. 21 a č. 22.

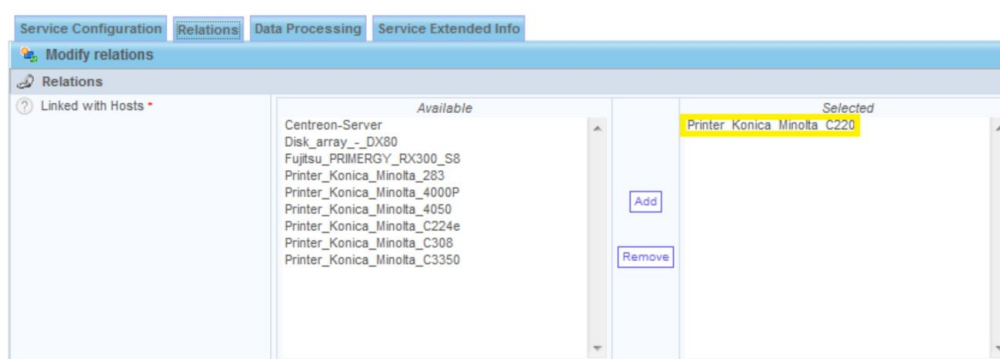
Service Configuration		Relations	Data Processing	Service Extended Info								
Modify a Service												
General Information												
Description *	C220 components consumables											
Service Template												
Service State												
Is Volatile	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Default											
Check Period *	24x7											
Check Command *	check_printer_components_konica_minolta											
Args	<table border="1"> <thead> <tr> <th>Argument</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>-mode: you can enter errors, consumables or counters</td> <td>consumables</td> </tr> <tr> <td>-w: state WARNING</td> <td>20</td> </tr> <tr> <td>-c: state CRITICAL</td> <td>10</td> </tr> </tbody> </table>				Argument	Value	-mode: you can enter errors, consumables or counters	consumables	-w: state WARNING	20	-c: state CRITICAL	10
Argument	Value											
-mode: you can enter errors, consumables or counters	consumables											
-w: state WARNING	20											
-c: state CRITICAL	10											
Max Check Attempts *	5											
Normal Check Interval *	3 * 60 seconds											
Retry Check Interval *	1 * 60 seconds											
Active Checks Enabled	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Default											
Passive Checks Enabled	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Default											

Obrázek 21: Centreon – vytvoření služby v záložce Service Configuration v sekci General Information a Service State

Obrázek 22: Centreon – vytvoření služby v záložce Service Configuration v sekci Notification

Pro uživatelskou přehlednost by měl název služby, zadaný do pole *Description*, obsahovat informace o typu zařízení a použitém módu. V poli *Check Period* definujeme dobu, po kterou bude monitoring prováděn. Z rolovací nabídky *Check Command* zvolíme název pluginu *check_printer_components_konica_minolta*, jehož nastavení jsme provedli v kapitole „7.4.2 Nastavení příkazu pro spuštění pluginu“. Po této volbě se nám do pole *Args* načte tabulka parametrů včetně jejich popisu, které mají být zadány uživatelem. Jejich hodnoty zadáváme do sloupce *Value*. V prvním parametru volíme jeden z typů módu: *errors*, *consumables* nebo *counters*. Pro parametry hraničních hodnot mezi stavy volíme hodnoty pohybující se v intervalu 0–100. Význam hodnoty *Max Check Attempts* jsme si vysvětlili při vytváření hosta. Navíc zadáme hodnoty pro *Normal Check Interval* – interval kontroly zařízení při stavu *OK*, a *Retry Check Interval* – interval kontroly zařízení při ostatních stavech. Poté zbývá zaškrtnout odesílané stavy a vybrat uživatele, kterému se budou hlášení posílat.

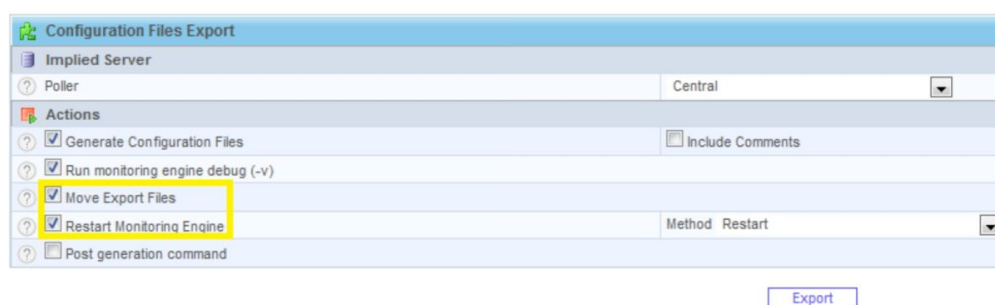
V posledním kroku vybereme zařízení vytvořené v kapitole „7.4.1 Přidání monitorovaného zařízení“, pro které bude monitoring realizován. Toho docílíme přechodem na záložku formuláře *Relations* (obrázek č. 23).



Obrázek 23: Centreon – vytvoření služby v záložce Relations

7.4.4 Aktualizace konfigurace monitorovacího systému

V předchozích kapitolách bylo vysvětleno nastavení příkazu pro jeho automatické spouštění, vytvoření hosta a služby monitorující zařízení. Aby se provedené změny promítly v monitorovacím systému, je zapotřebí aktualizovat konfiguraci restartováním Centreon Engine. Z tohoto důvodu přistoupíme do sekce *Configuration – Monitoring Engines*, označíme zaškrtačací políčka dle znázornění na obrázku č. 24 a klikneme na tlačítko *Export*.



Obrázek 24: Centreon – restartování Monitoring Engine

Po bezchybném restartování konfigurace se vypíše celkový přehled provedených akcí a stav *OK*. Restartování je vyžadováno pokaždé, když zasáhneme do konfigurace systému.

7.4.5 Vytvoření performance grafu

Performance grafy vytváříme pouze pro mód *counters*. Po dobu celého monitoringu se zobrazuje stav *OK*, neboť se nezaměřujeme na překročení určitých hodnot, ale pouze na výpis celkového počtu, který může nabývat libovolné kladné hodnoty. Prostý výpis aktuálního počtu vytisknutých stran má pro uživatele pouze informativní charakter, z kterého nevyčte, kolik stran bylo vytisknuto za určitou dobu (např. den, týden, měsíc) od počátku monitoringu. Pro tento účel jsou časově závislá data pomocí RRDTool nástroje zpracovávána a ukládána do databáze MySQL a z nich dokáže

vytvořit jakkoliv modifikovaný graf. Grafy jsou vytvářeny monitorovacím systémem automaticky po nastavení všech důležitých součástí. Z grafického znázornění je poté uživatel schopen vyčíst mnohem více informací.

Plugin vrací výsledky o celkovém počtu vytisknutých stran ve svém výstupu zasláním čitelného řetězce, za kterým následuje znak (|) a potom řetězec obsahující jednu nebo více metrik performance dat. Ukázkový výstup pluginu může vypadat například takto:

Počet vytisknutých stran: černobíle=100, barevně=25 | cb=100; bar=25

Pro každý typ tisku, který obsahuje daná tiskárna, vytváříme křivku grafu. Nastavení každé křivky provádíme monitorovacím systémem Centreon v sekci *Views – Graphs – Curves*, ve které definujeme tiskárnu, mód (*counters*) a název metriky. Pro zobrazení hodnoty křivky metriky přistoupíme do sekce *Views – Graphs – Metrics*, kde si zvolíme tiskárnu, mód a název metriky. Tímto nastavením se automaticky vygeneruje graf pro daný mód tiskárny, který si můžeme zobrazit.

8 Testování

Cílem kapitoly je ověřit funkčnost implementace pluginu vloženého do monitorovacího systému Centreon. Výsledkem monitoringu každého módu pluginu je jeden ze tří stavů a výpis vzniknutých chyb. Stavů definujeme: OK – komponenty pracují správně, WARNING – nalezena závada menšího rozsahu nebo CRITICAL – nalezená závada velkého rozsahu znemožňující provádět běžné operace tiskárny (tisknout, kopírovat atd.).

8.1 Principy testování

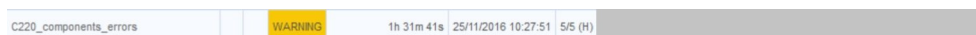
Testování monitorovaných tiskáren probíhalo dvěma odlišnými způsoby. Tiskárny umožňující přímé připojení k síti zahrnují vestavěné webové rozhraní, které poskytuje správu tiskárny z libovolného prohlížeče v síti podle práv přihlášeného uživatele (Public User, Registered User nebo Administrator). Pro účely testování stačilo přihlášení pouze jako *Public User*, abychom mohli zobrazit informace o stavu tiskárny (chybové stavy, spotřební materiál, počet vytisknutých stran). Způsob připojení na webové rozhraní tiskáren není blíže popsáno. Druhý způsob testování probíhal přímo ve firmě CDC Data, s. r. o. ve spolupráci s konzultanem práce Ing. Tomášem Holomkem. Účelem bylo na vybraných tiskárnách záměrně způsobit závadu a kontrolovat zda-li se tato závada promítla do prostředí monitorovacího systému Centreon v určitém čase.

8.2 Testování – webové rozhraní tiskáren

V tomto případě se ověřuje, zda-li hodnoty vykazované pluginem odpovídají reálným hodnotám tiskáren. Následuje ukázkové testování tiskárny Konica Minolta bizhub C220. Pro ostatní tiskárny probíhá testování identickým způsobem. Popisy chybových stavů vykazovaných tiskárnou v monitorovacím systému Centreon na následujících obrázcích nemohou být zveřejněny, neboť se jedná o stavy získané z MIB tabulek zařízení, jejichž dokumentace nejsou veřejné.

Testování chybových stavů

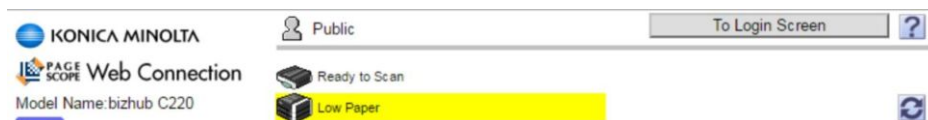
Po integraci pluginu do monitorovacího systému Centreon vidíme na obrázku č. 25, že tiskárna v módu *errors* vykazuje chybu či více méně závažných chyb a proto se nachází ve stavu *WARNING*.



Obrázek 25: Centreon – test chybových stavů tiskárny KM bizhub C220

Webové rozhraní tiskárny, zobrazené na obrázku č. 26, nás upozorňuje na chybu *Low Paper* (zvýrazněno žlutou barvou) vykazovanou tiskárnou. Konkrétní informace o tom, který zásobník má nedostatek papíru zjistíme kliknutím na toto chybové

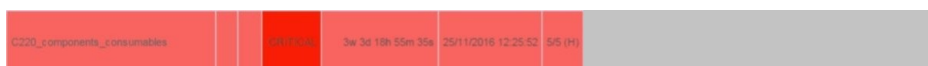
hlášení. Chyba vykazovaná ve webovém rozhraní tiskárny se shoduje s chybou zobrazenou v systému Centreon.



Obrázek 26: Webové rozhraní tiskárny KM bizhub C220 – chybové stavy

Testování spotřebního materiálu

Pro účely testování tohoto módu jsou nastaveny všem tiskárnám hranice rozlišující stavy na: 100–20 % => OK, 19–11 % => WARNING a 10–0 % => CRITICAL. Na obrázku č. 27 je zobrazeno, že tiskárna vykazuje kritický stav spotřebního materiálu pro jednu či více komponent. Tiskárna ovšem může současně vykazovat i varovný stav pro další komponenty.



Obrázek 27: Centreon – test spotřebního materiálu tiskárny KM bizhub C220

Na obrázku č. 28 vidíme aktuální hodnotu spotřebního materiálu komponent. V kritickém stavu (rozmezí 0–10 %) se nachází tyto komponenty: Toner – Cyan, Drum Unit – Black, Developing Unit – Black a Fusing Unit. Ve varovném stavu (rozmezí 11–19 %) se nachází Developing Unit – Cyan, Magenta a Yellow. I v tomto případě jsou pluginem v monitorovacím systému Centreon vypsány zmíněné komponenty spotřebního materiálu.

Consumables	
Toner	Status
Yellow	24%
Magenta	46%
Cyan	5%
Black	96%
Drum Unit	
Status	
Cyan	54%
Magenta	54%
Yellow	54%
Black	4%
Developing Unit	
Status	
Cyan	16%
Magenta	16%
Yellow	16%
Black	7%
Consumables	
Status	
Ozone Filter	32%
Fusing Unit	8%
Image Transfer Belt Unit	32%
Transfer Roller Unit	32%
Waste Toner Box	Ready
Staple Cartridge	Ready

Obrázek 28: Webové rozhraní tiskárny KM bizhub C220 – spotřební materiál

Testování celkového počtu vytisknutých stran

V tomto případě je výsledkem monitoringu stav *OK*, zachycený na obrázku č. 29 a také celkový počet vytisknutých stran (součet vytisknutých a kopírovaných) pro všechny typy tisku (nezobrazeno).



Obrázek 29: Centreon – test počtu vytisknutých stran tiskárny KM bizhub C220

Abychom získali správný výsledek celkového počtu vytisknutých a kopírovaných stran tiskárny, musíme sečíst hodnotu v řádku *Total* z tabulky *Copy Counter* s hodnotou ze stejnojmenného řádku tabulky *Print Counter* pro daný typ tisku (např. pro Full Color = 7389 + 70442) podle obrázku č. 30. Tento způsob provádíme pro získání celkového počtu všech typů tisku. Po porovnání jednotlivých výsledků s hodnotami vykazujícími monitorovacím systémem Centreon jsou hodnoty shodné.

Copy Counter					
	Full Color	Black	Single Color	2 Color	Total
Total	7389	83855	0	4	91248
Large Size	933	16554	0	1	17488

Print Counter				
	Full Color	Black	2 Color	Total
Total	70442	277955	105	348502
Large Size	8101	29811	27	37939

Obrázek 30: Webové rozhraní tiskárny KM bizhub C220 – počet vytisknutých stran

Tímto způsobem byla ověřena správnost hodnot vykazovaná pluginem pro všechny tři módy tiskárny KM bizhub C220. Stejným způsobem je nyní provedena kontrola zbývajících monitorovaných tiskáren a výsledky kontroly jsou znázorněny v tabulce č. 1. Pokud se hodnoty vykazované pluginem shodují s hodnotami webových rozhraní tiskáren je do tabulky zapsán stav *OK*.

Tabulka 1: Tabulka ověření výstupních hodnot pluginu

	errors	consumables	counters
C220	OK	OK	OK
4050	OK	OK	OK
C3350	OK	OK	OK
283	OK	OK	OK
C308	OK	OK	OK
4000P	OK	OK	OK
C224e	OK	OK	OK

V případě testování srovnáváním výstupních hodnot pluginu, po jeho implementaci a integraci do monitorovacího systému, s hodnotami webových rozhraní tiskáren byla prokázána jeho správná funkčnost.

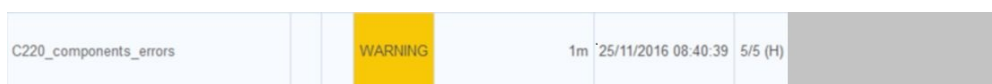
8.3 Testování – firma CDC Data, s. r. o.

Cílem tohoto způsobu testování je ověřit funkčnost pluginu v reálném čase. Pro vykonávané testy byla zvolena doba monitoringu na 180 sekund tzn. doba, za kterou dochází k detekci chybových stavů a jejich promítnutí v systému Centreon od počátku monitoringu. Nastavení intervalu monitoringu je pro každého uživatele individuální záležitostí.

Test 1: KM bizhub C220 – simulace nedostatku papíru v zásobníku

První testovací případ se zabývá testováním tiskárny simulací nedostatku papíru pro tisk. Před zahájením testování vykazuje monitorovací systém Centreon příslušné tiskárny pro mód *errors* stav *OK* a výpis „*Error status is OK*“.

V čase 08:40:00 jsme záměrně způsobili závadu odebráním papíru ze zásobníku tiskárny. Po uplynutí 39 sekund se tato závada projevila v systému Centreon, stav *OK* byl nahrazen chybovým stavem *WARNING*, jak můžeme vidět na obrázku č. 31, a v poli pro výpis informací o zařízení se objevil název způsobené chyby. Uživatel je zobrazením popisu chyby v prostředí Centreonu informován o tomto nedostatku, který tak může jednoduše opravit.



Obrázek 31: Centreon – reálný stav komponent vykazovaný systémem Centreon po způsobení závady odebráním papíru ze zásobníku

Test 2: KM bizhub C220 – simulace otevření krytu

Druhý testovací případ navazuje na konec prvního testu výchozím stavem podle obrázku 31. Nyní jsme způsobili závažnou chybu otevřením krytu tiskárny. V takovém stavu by měla nastat situace, kdy tiskárna neumožňuje tisk a uživatel musí být o této chybě neprodleně informován. I tento typ chyby se zobrazuje ve webovém rozhraní tiskárny.

Chybu jsme způsobili v čase 08:50:30 a za 66 sekund se projevila v monitorovacím systému (obrázek č. 32), kdy monitorovaná služba změnila ze stavu *WARNING* na stav *CRITICAL*, který obdržel od pluginu, neboť urgentní závady pro správnou funkci tiskárny je třeba řešit okamžitě. Nyní tiskárna vykazuje dvě chyby ve stejném okamžiku, kdy první informuje o otevřeném krytu a druhá o nedostatku papíru v jednom ze zásobníků.

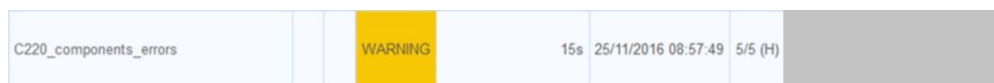


Obrázek 32: Centreon – reálný stav komponent vykazovaný systémem Centreon po způsobení závady otevřením krytu tiskárny KM bizhub C220

Test 3: KM bizhub C220 – simulace zavření krytu

Za počátečním stav třetího testovacího případu považujeme výsledek druhého testu (obrázek č. 32). Test se zaměřuje na odstranění závady (zavřením krytu v čase 08:57:00) způsobené v druhém testu. Poté by se měl výstup pluginu opět vrátit do stavu *WARNING* obsahující pouze chybu informující na nedostatek papíru v jednom ze zásobníků tiskárny.

Ve výsledku tohoto testu (obrázek č. 33) se opravená závada správně ukázala v systému Centreon. Kritická závada byla odstraněna, chybový stav *CRITICAL* byl nahrazen stavem *WARNING* a i popis zobrazoval pouze jedinou chybu.



Obrázek 33: Centreon – reálný stav komponent vykazovaný systémem Centreon po odstranění závady zavřením krytu tiskárny KM bizhub C220

Test 4: KM bizhub 283 – simulace otevření krytu

Před začátkem čtvrtého testovacího případu se tiskárna nachází ve stavu *WARNING* a vykazuje chybu týkající se nedostatku papíru v jednom ze zásobníků. I zde se pokusíme nasimulovat závažnou chybu otevřením krytu (standardně zobrazitelná i ve webovém rozhraní) tiskárny v čase 09:00:00.

V rozmezí necelé minuty se nám tato chyba projevila v systému Centreon změnou stavu na *CRITICAL*, jak můžeme vidět na obrázku č. 34, a do výpisu byl přidán popis informující o otevření krytu tiskárny.

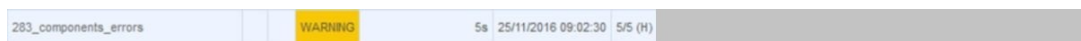


Obrázek 34: Centreon – reálný stav komponent vykazovaný systémem Centreon po způsobení závady otevřením krytu tiskárny KM bizhub 283

Test 5: KM bizhub 283 – simulace zavření krytu

Za počáteční stav pátého testovacího případu považujeme kritický stav, který nastal po otevření krytu tiskárny (obrázek č. 34). Nyní se pokusíme, stejně jako v případě tiskárny KM bizhub C220, zavřít kryt a opravit tak závažnou chybu tiskárny.

Jak můžeme vidět na obrázku č. 35 i takhle chyba byla uživatelem odstraněna, stav *CRITICAL* změněn na *WARNING* a výpis opět obsahuje pouze informaci o nedostatku papíru v zásobníku.



Obrázek 35: Centreon – reálný stav komponent vykazovaný systémem Centreon po odstranění závady zavřením krytu tiskárny KM bizhub 283

8.4 Shrnutí testování

Při testování tiskáren porovnáváním hodnot vykazovaných pluginem s reálnými hodnotami zobrazenými webovými rozhraními byla prokázána správná funkčnost všech monitorovaných tiskáren. Bezchybné testování v podobě předání správného chybového stavu a výpisu závadných komponent v reálném čase proběhlo i v případě testování tiskáren Konica Minolta bizhub C220 a 283 ve firmě CDC Data, s. r. o. po záměrném způsobení a následném odstranění závad blíže popsanych v jednotlivých testovacích případech. V době testování byly všechny tiskárny k dispozici a bylo možné je otestovat. Závěrem tedy lze shrnout, že plugin je plně funkční pro dané typy tiskáren.

9 Ekonomické zhodnocení

Obsahem kapitoly je z finančního pohledu zhodnotit navržené řešení monitoringu pro fiktivní firmu na základě jejich požadavků. Předpokládáme, že práci spojenou s realizací kompletního monitoringu vyplývajících z požadavků firmy zvládne zkušený pracovník v této oblasti (správce počítačové sítě), jehož hrubý měsíční plat je 42 300 Kč – což přepočteno na hodinovou sazbu činí 265 Kč (Ministerstvo práce a sociálních věcí, 2016).

Náklady firmy spojené s monitoringem můžeme rozdělit na několik skupin, kde hlavní z nich zahrnují náklady na instalaci monitorovacího systému Centreon, implementaci a testování pluginu pro určitý počet zařízení a služby spojené s monitoringem.

Nejdříve se zaměříme na počáteční náklady spojené s instalací monitorovacího systému Centreon. Počítáme-li s tím, že firma již disponuje dostatečným hardwarem tj. serverem s operačním systémem rodiny Unix, na který je možné tento monitorovací systém nainstalovat. Základní součásti systému jsou k dispozici na oficiálních stránkách Centreonu s licencí open source. Jestliže by byly firmou kladeny speciální požadavky na instalaci zpoplatněných částí systému Centreon, cena instalace se může vystoupat až na několik desítek tisíc korun. V případě instalace zdarma dostupných částí uvažujeme pouze platbu za práci odborníka instalujícího monitorovací systém na server firmy. Při uvažované době instalace pohybující se mezi 4–8 hodinami, v závislosti na schopnostech odborníka, a hodinové sazbě 265 Kč firmu zprovoznění monitorovacího systému přijde zhruba na 1060–2120 Kč.

Do druhé skupiny nákladů patří lidská práce související s implementací, testováním pluginu s ohledem na konkrétní počet monitorovaných zařízení a nasazením pluginu do systému Centreon běžícím na serveru zadavatelské firmy. I v této situaci předpokládáme, že firma již disponuje zařízeními, na kterých chce realizovat monitoring. Uvažujme, že zadavatelská firma vznesla požadavek na monitoring stejného počtu (5–7) zařízení, jako u vytvářeného pluginu v rámci této bakalářské práce. Navržené vhodné řešení monitoringu pro takový počet zařízení zahrnující analýzu, implementaci pluginu, otestování jeho správné funkčnosti na produkčních zařízeních a nasazení pluginu do reálného provozu zabere celkově více než jeden pracovní týden (přes 40 hodin). Výsledný čas nezávisí pouze na počtu monitorovaných zařízení, ale také na požadavcích zadavatelské firmy. Při vynaloženém čase zahrnujícím 40 hodin práce se náklady firmy vyšplhaly na částku 10 600 Kč.

Zatímco doposud byly všechny uvažované náklady jednorázového charakteru, nyní se zaměříme na náklady pravidelně (měsíčně, čtvrtletně atd.) vynakládané na služby provozu a údržby zařízení pracovníkem firmy realizující monitoring. Tyto náklady nelze blíže specifikovat, neboť se odvíjejí od více faktorů, do kterých řadíme celkový čas opravy, podle rozsahu vzniklých chyb, nákup nových komponent nahrazujících závadné (nebo komponenty s dosluhující životností), dopravu a plat odborného pracovníka.

10 Závěr

Cílem práce bylo na základě analýzy uživatelských požadavků firmy CDC Data, s. r. o. naprogramovat plugin, který vede k detekci konkrétního modelu tiskárny, analyzuje aktuální stav zařízení a získané hodnoty vrátí ve formě zpracovatelné monitorovacím systémem Centreon. Součástí práce byla i integrace pluginu do systému Centreon v experimentální síti a otestování jeho funkčnosti na produkčních zařízeních.

Firmou kladené požadavky na funkci pluginu se podařily implementovat a otestovat na všech zařízeních Konica Minolta bizhub C220, 4050, C3350, 283, C308, 4000P a C224e v experimentální síti. Plugin získává informace o aktuálním stavu zařízení prostřednictvím protokolu SNMP určeného pro správu sítě.

Testováním pluginu na produkčních zařízeních byla prokázána jeho správná funkce na všech zařízeních, které byly k dispozici jak v implementační, tak i v testovací fázi. U testování porovnáváním hodnot vykazovaných pluginem s reálnými hodnotami z webových rozhraní tiskáren byl ověřen správný výstup pluginu. Při testování tiskáren Konica Minolta bizhub C220 a 283 byla ověřena i správná funkce pluginu, integrovaného v systému Centreon, v reálném čase.

Podle slov zadavatele plugin, který byl vyvíjen v rámci této bakalářské práce, nahradí dosud používaný zastaralý plugin, jež neumožňuje takový rozsah možností monitoringu zařízení. Tento plugin bude nasazen do zkušebního reálného provozu jednomu zákazníkovi v průběhu ledna 2017 a v případě bezproblémového chodu, popřípadě po odhalení a odladění chyb, bude postupně k dispozici i dalším zákazníkům.

11 Seznam zdrojů

- BEHROUZ, A. *TCP/IP Protocol Suite*. Fourth Edition. [New York]: McGraw-Hill, 2010. 1029 s. ISBN 978-0-07-337604-2.
- BOUŠKA, P. *VPN 8 – Dvou-faktorová autentizace s certifikátem*. [online]. 2014. [cit. 2016-09-06]. Dostupné z: <http://www.samuraj-cz.com/clanek/vpn-8-dvou-faktorova-autentizace-scertifikatem/>.
- BOUŠKA P. *Začínáme s monitoringem sítě*. [online]. 2009. [cit 2016-09-06]. Dostupné z: <http://www.samuraj-cz.com/clanek/zaciname-s-monitoringem-site/>.
- BUILDER.CZ *Protokol UDP 1. část*. [online]. 2003. [cit. 2016-09-08]. Dostupné z: <http://www.builder.cz/rubriky/c/c-/protokol-udp-1-cast-156226cz>.
- CDC DATA, S. R. O. *Co umíme*. [online]. 2016. [vid. 2016-09-05]. Dostupné z: <https://www.cdc.cz/co-umime>.
- CENTREON. *Centreon Documentation Release 2.6.4*. [online]. 2015a. [cit. 2016-09-26]. Dostupné z: https://documentation.centreon.com/media/pdf/centreon/2.6.x/centreon_en.pdf.
- CENTREON. *Company*. [online]. 2016b. [vid. 2016-09-26]. Dostupné z: <https://www.centreon.com/en/company/>.
- CENTREON. *Produkty*. [online]. 2016c. [vid. 2016-10-11]. Dostupné z: <https://www.centreon.com/company/partners/>.
- CENTREON. *Installation – Centreon Broker*. [online]. © 2012-2015d. [cit. 2016-11-12]. Dostupné z: <https://documentation.centreon.com/docs/centreon-broker/en/3.0/installation/index.html#using-sources>.
- CENTREON. *Installation – Centreon Clib*. [online]. © 2013-2015e. [cit. 2016-11-12]. Dostupné z: <https://documentation.centreon.com/docs/centreon-clib/en/1.4/installation/index.html#using-sources>.
- CENTREON. *Installation – Centreon Engine*. [online]. © 2013f. [cit. 2016-11-12]. Dostupné z: <https://documentation.centreon.com/docs/centreon-engine/en/1.4/installation/index.html#using-sources>.
- CENTREON. *Solutions – centreon*. [online]. 2016g. [vid. 2016-12-12]. Dostupné z: <https://www.centreon.com/solution/>.
- CENTREON. *Welcome on Centreon Documentation*. [online]. 2016h. [cit. 2016-09-05]. Dostupné z: <https://documentation.centreon.com/>.
- CENTREON. *Welcome to Centreon Broker's documentation!*. [online]. © 2012–2016ch. [cit. 2016-09-06]. Dostupné z: <https://documentation.centreon.com/docs/centreon-broker/en/2.11/>.

- CENTREON. *Welcome to Centreon Engine's documentation!*. [online]. © 2013–2015i. [cit. 2016-09-06]. Dostupné z: <https://documentation.centreon.com/docs/centreon-engine/en/latest/>.
- CENTREON. *Welcome to Centreon's documentation!*. [online]. © 2015j. [cit. 2016-09-06]. Dostupné z: https://documentation.centreon.com/docs/centreon/en/2.8.x/installation/from_sources.html.
- COMER E. *Internetworking with TCP/IP Volume One*. 6th Edition. [New Jersey]: Addison-Wesley, 2014, 744 s. ISBN 978-0-13-608530-0.
- DAŘENA, F. *Myslíme v jazyku PERL*. [Praha]: Grada, 2005. 700 s. ISBN 978-80-247-6390-3.
- DAŘENA, F. *Perl*. [online]. 2012. [cit. 2016-09-06]. Dostupné z: <https://akela.mendelu.cz/darena/Perl/>.
- DOSTÁLEK L. A KABELOVÁ A. *Velký průvodce protokoly TCP/IP a systémem DNS*. [Praha]: Computer Press, 2000. 435 s. ISBN 80-7226-323-4.
- DSL.CZ *SSH – bezpečné používání vzdáleného počítače a kopírování dat*. [online]. 2106. [cit. 2016-10-19]. Dostupné z: <http://www.dsl.cz/jak-na-to/jak-na-ssh>.
- FALL R. A STEVENS R. *TCP/IP Illustrated, Volume 1*. 2nd Edition. [Ann Arbor]: Addison-Wesley Professional, 2012. 1056 s. ISBN 978-0-321-33631-6..
- FIŠER, P. *Optimalizace nákladů na tisk ve firmě*. Jihlava, 2012. Bakalářská práce. Vysoká škola polytechnická Jihlava. Katedra elektrotechniky a informatiky. Vedoucí práce Ing. Bc. Michal Vopálenský, Ph.D.
- HABRMAN, Z. *Monitorování aktivních prvků počítačové sítě*. Zlín, 2013. Diplomová práce. Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky. Vedoucí práce Ing. Jiří Korběl, Ph.D.
- HERTZOG, R. A MAS, R. *The Debian Administrator's Handbook*. [online]. © 2003–2015 [cit. 2016-09-06]. Dostupné z: <https://debian-handbook.info/download/stable/debian-handbook.pdf>.
- HIETANIEMI, J. *Comprehensive Perl Archive Network*. [online]. © 1995–2010. [cit. 2016-09-06]. Dostupné z: <http://www.cpan.org/>.
- HW GROUP *Síťové komunikační protokoly Damocles*. [online]. 2016. [vid. 2016-11-24]. Dostupné z: http://www.hw-group.com/products/damocles/dam_protocols_cz.html.

- KONICA MINOLTA *bizhub C360/C280/C220 User's guide [Network Administrator]*. [online]. © 2009. [cit 2016-11-28]. Dostupné z: <https://www.manualslib.com/manual/432385/Konica-Minolta-Bizhub-C220.html?page=205#manual>.
- KONICA MINOLTA *bizhub C368/C308/C258 User's Guide*. [online]. © 2015. [cit 2016-11-28]. Dostupné z: http://manuals.konicaminolta.eu/bizhub-C368-C308-C258/EN/contents/opkey_693_S.html.
- KONICA MINOLTA *bizhub C3850/C3350 User's Guide Applied Functions*. [online]. © 2013. [cit 2016-11-28]. Dostupné z: <https://www.manualowl.com/m/Konica%20Minolta/bizhub-C3350/Manual/406758>.
- KONICA MINOLTA *bizhub C554e/C454e/C364e/C224e User's Guide*. [online]. © 2013. [cit 2016-11-28]. Dostupné z: http://manuals.konicaminolta.eu/bizhub-C554e-C454e-C364e-C284e-C224e/EN/contents/opkey_1042_S.html.
- KONICA MINOLTA *bizhub 423/363/283/223 User's Guide [Security Operations]*. [online]. © 2010. [cit 2016-11-28]. Dostupné z: http://www.manualsprinter.com/manuals/konica/Konica_Minolta_bizhub_363.pdf.
- KONICA MINOLTA *bizhub 4750/4050 User's Guide Applied Functions*. [online]. © 2013. [cit 2016-11-28]. Dostupné z: <https://www.manualowl.com/m/Konica%20Minolta/bizhub-4050/Manual/406780>.
- KONICA MINOLTA *Konica Minolta bizhub 4700P/4000P/3300P*. [online]. 2013. [cit 2016-11-28]. Dostupné z: <http://it.wvu.edu/files/d/86555cfe-3cab-4620-8683-c11bdc6c66d5/bizhub-4700p-spec-and-install-guide.pdf>.
- LAMMLE T. *CCNA: Cisco Certified Network Associate Study Guide*. 6th Edition. [Brno]: Computer Press a.s, 2010. 928 s. ISBN 978-80-251-2359-1.
- MALÍK, J. *Správa různorodých síťových zařízení pomocí SNMP protokolu*. Brno, 2011. Diplomová práce. Masarykova univerzita. Fakulta informatiky. Vedoucí práce RNDr. Tomáš Rebok, Ph.D.
- MAURO, D. A SCHMIDT, K. *Essential SNMP*. 2nd Edition. [Sebastopol]: O'Reilly Media, 2005. 462 s. ISBN 978-0-596-00840-6.
- MICROSOFT *SNMP Technical Reference*. [online]. 2003. [cit 2016-09-06]. Dostupné z: [https://technet.microsoft.com/en-us/library/cc787561\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc787561(v=ws.10).aspx).

- MIHOK, M. *Monitorování počítačových sítí a aplikací pomocí programu Nagios*. Zlín, 2010. Diplomová práce. Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky. Vedoucí práce Ing. Miroslav Matýsek, Ph.D.
- MINISTERSTVO PRÁCE A SOCIÁLNÍCH VĚCÍ 2522 – *Systémoví administrátoři, správci počítačových sítí*. [online]. 2016. [cit 2016-10-03]. Dostupné z: <http://www.mpsv.cz/ISPVvypis.php?kzams=3381&ok=Zobrazit+informace&sfera=2&sz=2&txt=>.
- MOUČKA B. A PEŠA R. *Zabezpečené spojení se vzdáleným počítačem*. [online]. 2011. [cit 2016-11-07]. Dostupné z: <http://webserver.ics.muni.cz/bulletin/articles/249.html>.
- SATRAPA, P. *Perl pro zelenáče*. [Praha]: Neocortex, 2000. 224 s. ISBN 80-86330-02-8.
- SOSINSKI, B. *Mistrovství – počítačové sítě*. [Brno]: Computer Press, 2010. 840 s. ISBN 978-80-251-3363-7.
- SVOBODA, F. *Management počítačové sítě ISP*. Brno, 2014. Bakalářská práce. Vysoké učení technické v Brně. Fakulta podnikatelská. Vedoucí práce Ing. Viktor Ondrák, Ph.D.
- SZABO, G. *Monitoring lokálnej siete prostredníctvom Nagiosu*. Brno, 2012. Bakalářská práce. Masarykova univerzita. Fakulta informatiky. Vedoucí práce doc. RNDr. Eva Hladká, Ph. D.
- SZANISLO, T. *Správa heterogénných sieťových prvkov*. Brno, 2012. Masarykova univerzita. Fakulta informatiky. Vedoucí práce RNDr. Jan Kasprzak.
- WILKINS, S. *SNMP Concepts and Configuration*. [online]. 2011. [cit 2016-10-03]. Dostupné z: <http://www.ciscopress.com/articles/article.asp?p=1730888>.

Přílohy

A CD – Zdrojový kód pluginu