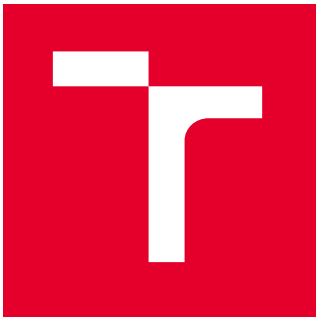


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

IMPLEMENTACE MONITOROVACÍHO SYSTÉMU PRO AKTIVNÍ PRVKY V UNIVERZITNÍM PROSTŘEDÍ

IMPLEMENTATION OF A MONITORING SYSTEM FOR ACTIVE COMPONENTS IN THE UNIVERSITY ENVIRONMENT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Patrik Ferko**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Viktor Ondrák, Ph.D.**
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA PODNIKATELSKÁ

Zadání diplomové práce

Ústav:	Ústav informatiky
Student:	**Bc. Patrik Ferko**
Vedoucí práce:	**Ing. Viktor Ondrák, Ph.D.**
Akademický rok:	2023/24
Studijní program:	Informační management

Garant studijního programu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Implementace monitorovacího systému pro aktivní prvky v univerzitním prostředí

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Navrhnout management počítačové sítě.

Základní literární prameny:

DONAHUE, Gary A. Kompletní průvodce síťového experta. Brno: Computer Press, 2009. 528 s. ISBN 978-80-251-2247-1.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009. 496 s. ISBN 978-80-247-2615-1.

KRETCHMAR, James M a Libor DOSTÁLEK. Administrace a diagnostika sítí: pomocí OpenSource utilit a nástrojů. Brno: Computer Press, 2004. 216 s. ISBN 80-251-0345-5.

ONDRÁK, Viktor, Petr SEDLÁK a Vladimír MAZÁLEK. Problematika ISMS v manažerské informatice. Brno: Akademické nakladatelství CERM, 2013. 377 s. ISBN 978-80-7204-872-4.

SOSINSKY, Barrie A. Mistrovství – počítačové sítě. Brno: Computer Press, 2010. 840 s. ISBN 978-80-251-3363-7.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2023/24

V Brně dne 4.2.2024

L. S.

doc. Ing. Miloš Koch, CSc. garant	doc. Ing. Vojtěch Bartoš, Ph.D. děkan

Abstrakt

Diplomová práca sa zameriava na výber a implementáciu monitorovacieho systému do univerzitného prostredia s cieľom zabezpečiť spoľahlivé monitorovanie a správu sieťových zariadení. V úvodnej časti práce je poskytnutý prehľad súčasného stavu vybranej univerzity, jej požiadaviek na monitorovanie a teoretických základov v oblasti monitorovania sietí. V implementačnej časti je podrobný výber vhodného monitorovacieho systému a identifikácia hlavných hrozieb, ktoré ovplyvňujú sieť univerzity. Nasleduje nastavenie vybraného monitorovacieho systému, ktoré zahŕňa iniciálne načítanie zariadení z dokumentačného systému NetBox. Okrem toho sa v tejto časti popisuje konfigurácia monitorovacích sond, notifikačných mechanizmov a priradenie práv pre užívateľské skupiny.

Klíčová slova

Počítačová sieť, monitorovanie siete, výber monitorovacieho systému, NetBox, Zabbix, konfigurácia SNMP

Abstract

The thesis focuses on the selection and implementation of a monitoring system in a university environment to ensure reliable monitoring and management of network devices. The introductory part of the thesis provides an overview of the current status of the selected university, its monitoring requirements and the theoretical background in network monitoring. The implementation section details the selection of a suitable monitoring system and the identification of the main threats affecting the university's network. This is followed by the setup of the selected monitoring system, which includes the initial loading of devices from the NetBox documentation system. In addition, this section describes the configuration of monitoring probes, notification mechanisms, and assignment of privileges for user groups.

Keywords

Computer network, network monitoring, monitoring system selection, NetBox, Zabbix, SNMP configuration

Bibliografická citácia

FERKO, Patrik. *Implementace monitorovacího systému pro aktivní prvky v univerzitním prostředí* [online]. Brno, 2024 [cit. 2024-05-06]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/156159. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Ing. Viktor Ondrák, Ph.D.

Čestné prehlásenie

Prehlasujem, že predložená práca je originálna a vypracoval som ju samostatne. Vyhlasujem, že citácie použitých zdrojov sú úplné a že som v práci neporušil autorské práva (v zmysle zákona č. 121/2000 Sb., o autorskom práve a právach súvisiacich s autorským právom).

V Brne dňa 13. 5. 2024

Bc. Patrik Ferko

autor

Poďakovanie

Týmto by som sa chcel poďakovať svojmu vedúcemu diplomovej práce Ing. Viktorovi Ondrákovi, Ph.D. za trpezlivosť, ochotu a cenné rady, ktoré posunuli túto prácu na odbornejšiu úroveň. Zároveň by som sa chcel poďakovať aj svojej rodine a kolegom z práce za podporu počas štúdia.

OBSAH

ÚVOD

V dnešnej digitálnej ére, kde spoľahlivá prevádzka siete predstavuje kľúčový pilier pre efektívne fungovanie organizácií a inštitúcií, získava monitorovanie sietí stále väčšiu dôležitosť. S rastúcou zložitosťou a objemom dátového prenosu v sieťach univerzít, je nevyhnutné mať na mieste robustný monitorovací systém, ktorý dokáže efektívne sledovať a diagnostikovať výkonnosť a funkčnosť sieťových zariadení. Táto diplomová práca sa zaoberá práve touto problematikou a snaží sa riešiť nastavenia monitorovacieho systému v univerzitnom prostredí.

V diplomovej práci sa zameriavam na výber a implementáciu monitorovacieho systému do prostredia univerzity. Zámerom je poskytnúť univerzite efektívne nástroje na monitorovanie a správu jej siete a sieťových zariadení. Práca sa venuje nielen technickej stránke implementácie, ale aj identifikácii kľúčových hrozieb, ktoré môžu ovplyvniť prevádzku siete. Cieľom je nastaviť monitorovací systém tak, aby bol schopný detegovať a reagovať na potenciálne problémy v sieti včas a efektívne.

V úvodnej časti práce sa bližšie preskúma súčasný stav vybranej univerzity a jej požiadavky na monitorovanie siete. Následne sa zameriam na teoretické východiská a v záverečnej časti budem popisovať konfiguráciu monitorovacích sond, notifikačných mechanizmov a priradenie práv pre užívateľské skupiny.

CIELE PRÁCE, METÓDY A POSTUPY SPRACOVANIA

Cieľom práce je návrh a implementácia monitorovacieho systému pre univerzitné prostredie s dôrazom na sieťové prvky od vendora Cisco.

Prvým krokom je identifikovať špecifické požiadavky univerzity na monitorovanie siete a sieťových zariadení. Na základe týchto požiadaviek bude vybraný vhodný monitorovací systém, ktorý bude schopný efektívne zabezpečiť sledovanie a správu sieťových prvkov.

Ďalším krokom je analýza kľúčových hrozieb, ktoré môžu ovplyvniť prevádzku siete. Na základe tejto analýzy budú navrhnuté opatrenia na predchádzanie týmto hrozbám a zabezpečenie bezpečného a spoľahlivého fungovania siete.

Následne bude implementovaný monitorovací systém v súlade s identifikovanými požiadavkami a bezpečnostnými opatreniami. To zahŕňa nastavenie sond, notifikácií a prístupových práv pre užívateľov.

1 ANALÝZA SÚČASNÉHO STAVU

V tejto kapitole budem prezentovať univerzitu, v rámci ktorej bude realizovaný projekt monitorovacieho systému pre aktívne prvky. Táto analýza bude zahrňovať dôležité informácie o spomínanej univerzite, aktuálny stav počítačovej siete, informačné systémy, ktoré sú využívané na univerzite, a nakoniec budú predstavené aj požiadavky investora.

1.1 Základné informácie o univerzite

Zvolenou vysokou školou je nemenovaná univerzita, sídliaca v Brne, ktorá patrí medzi najväčšie verejné české vysoké školy. Jej hlavným akademickým zameraním sú technické, ekonomické a sociálne vedy. V súčasnosti poskytuje študijné programy na deviatich fakultách, zahŕňajúc bakalárske, magisterské a doktorandské štúdiá pre viac ako 19 000 študentov. V rámci svojich učebných programov ponúka viac ako 70 rôznych študijných odborov a zaručuje svojim študentom prístup k významným výskumným a medzinárodným projektom, široké spektrum športových aktivít, možnosť študovať na zahraničných a partnerských univerzitách, odborné exkurzie a stáže, disponuje 8 knižnicami a poskytuje ubytovacie možnosti v internátoch. Každá fakulta má svoju vlastnú študentskú radu, ktorá zastupuje študentov v akademickom senáte a umožňuje im aktívne zapojiť sa do riadenia svojej fakulty. Výskumná činnosť na tejto inštitúcii sa hlavne uskutočňuje prostredníctvom národných a medzinárodných projektov, programov a výskumných centier. Okrem toho spolupracuje so zahraničnými univerzitami, Akadémiou vied České republiky, odborníkmi z praxe a súkromnými firmami. Univerzita je držiteľom certifikátov Európskej komisie ECTS Label a DS Label, ktoré svedčia o vysokom štandarde vysokoškolského vzdelávania. Navyše, dlhodobo sa umiestňuje na prestížnom rebríčku svetových univerzít QS TopUniversities. Jedným z jej dlhodobých cieľov je stať sa uznávanou medzinárodnou výskumnou univerzitou a excelentnou inštitúciou v českom vysokom školstve.

1.1.1 Ústav informačných služieb a výpočtovej techniky

Tento vysokoškolský inštitút funguje ako samostatná entita v rámci univerzity a má za úlohu spravovať informačné a komunikačné technológie na univerzite. Okrem toho sa venuje aj vedeckej a výskumnej činnosti v oblasti výpočtovej techniky, dátových

infraštruktúr a kybernetickej bezpečnosti. Jeho hlavnými úlohami sú prevádzka, monitorovanie a rozvoj IT služieb, komplexná správa hlavnej počítačovej siete a zabezpečenie funkčnosti výpočtových serverov.

1.1.2 Helpdesk

Univerzita prevádzkuje helpdesk pre koncových užívateľov, ktorými sú hlavne študenti a pedagógovia. Helpdesk je zameraný na odbavovanie L1 ticketov, teda dotazov a nízkoúrovňových problémov. Spracovanie týchto ticketov je riadené stanovenými procesmi a postupmi, ktoré smerujú k dvom možným výsledkom. V prípade, že helpdesk môže samostatne vyriešiť L1 ticket a poskytnúť odpoveď priamo koncovému užívateľovi, postupuje podľa definovaných návodov. V opačnom prípade, keď helpdesk nie je schopný samostatne vyriešiť ticket, prenesie ho priamo na príslušné oddelenie Ústavu informačných služieb a výpočtovej techniky, ktoré je priamo zodpovedné za danú službu alebo oblasť.

1.2 Súčasný stav

1.2.1 Počítačová sieť

Budovy

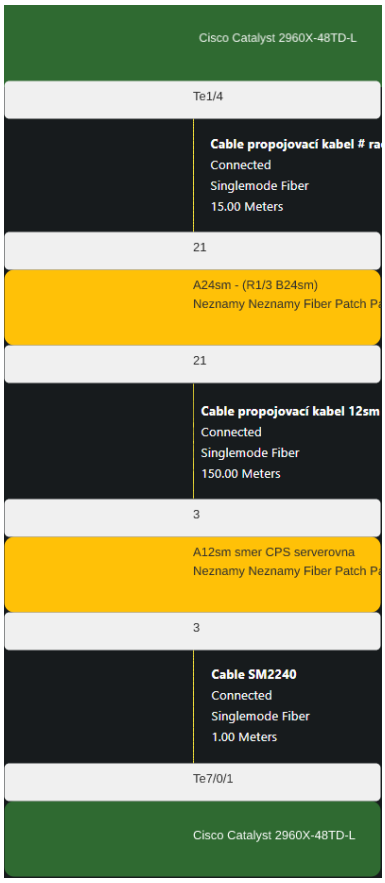
Fyzická infraštruktúra je rozdelená do 14 lokalít v rôznych častiach Brna, ktoré slúžia ako sídlo pre 9 fakúlt a 5 internátov. Každá fakulta má aspoň jednu budovu, kde sa nachádza minimálne jedno poschodie určené pre vedenie fakulty a jej zamestnancov. Prístup k týmto poschodiam je riadený prístupovými kartami pre zamestnancov a mechanickými kľúčmi. Budovy fakúlt sú vybavené slaboprúdovými rozvodňami, ktoré obsahujú aktívne a pasívne prvky počítačových sietí. Pre získanie prístupu k týmto priestorom je vyžadovaný mechanický kľúč, ktorý je uložený buď na recepcii alebo na pulte centrály ochrany (v prípade fakúlt s viacerými budovami). Odovzdávanie kľúčov je evidované v protokole prístupov, ktorý zaznamenáva dátum a čas vyzdvihnutia a vrátenia kľúča, spolu s identifikáciou osoby, ktorá o prístup požiadala. Pracovníci na recepcii a v centrále ochrany majú povinnosť preveriť totožnosť osoby na základe identifikačného preukazu

zamestnanca a skontrolovať, či sa osoba nachádza na zozname oprávnených osôb, ktorý je vytlačený na papieri a aktualizovaný každé dva mesiace.

Univerzita taktiež disponuje dátovým centrom, ktoré sa nachádza na fakulte informatiky a slúži na umiestnenie serverov, diskových polí a virtualizačnej infraštruktúry. Prístup do tohto dátového centra vyžaduje mechanický kľúč získaný na recepcii a preverenie osoby pri vstupe. Každý rozvádzač v dátovom centre je uzamknutý a k jeho odomknutiu je potrebný mechanický kľúč, ktorý je k dispozícii v skrinke pri vstupe do miestnosti.

Pasívna vrstva

Kabeláž horizontálneho vedenia na fakultách je zostavená z metalických káblov kategórie CAT 5E, ktoré operujú na rýchlostiach 1 Gbps. Tieto káble sú dostatočné na uspokojenie potrieb koncových staníc zamestnancov fakúlt, kamier a access pointov. Horizontálne vedenie v dátovom centre je zase vybavené metalickými káblami CAT 6A. Chrbticové prepojenia, konkrétne up-linky a down-linky, sú realizované optickými single mode vláknami. Redundantné trasy sú implementované prostredníctvom priamej metódy. Optické patch cordy typu LC-SC sa používajú na spojenie medzi aktívnymi prvkami a optickej distribučnej skrinkami (ODF), zatiaľ čo optické patch cordy typu E2000-E2000 slúžia na prepojenie medzi jednotlivými ODF na optických „križovatkách“.



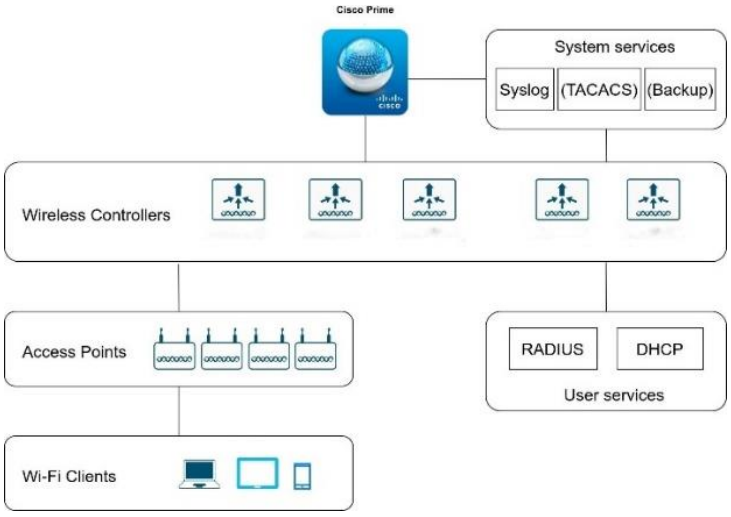
Obrázok 1: Výpis optickej trasy z IS Netbox (Zdroj: Vlastné spracovanie)

Na univerzite sa najčastejšie používajú rozvádzače s výškou 42 a 45U, pričom v dátovom centre sú preferované rozvádzače s výškou 47U. Väčšina týchto rozvádzačov nie je uzamknutá, a ich bočné a zadné časti sú úplne otvorené (výnimkou je dátové centrum). Každý rozvádzač určený pre slaboprúdové rozvodne je vybavený klimatizáciou a nepretržitými napájacími zdrojmi typu UPS so záložnou batériou. Aktívne prvky, ktoré disponujú redundantnými napájacími zdrojmi, sú pripojené do dvoch samostatných elektrických okruhov. Dátové centrum disponuje aj alternatívnym napájaním z dynamickej UPS (DUPS), ktorá je poháňaná dieselovým motorom.

Obrázok 2: DUPS (Zdroj: (16))

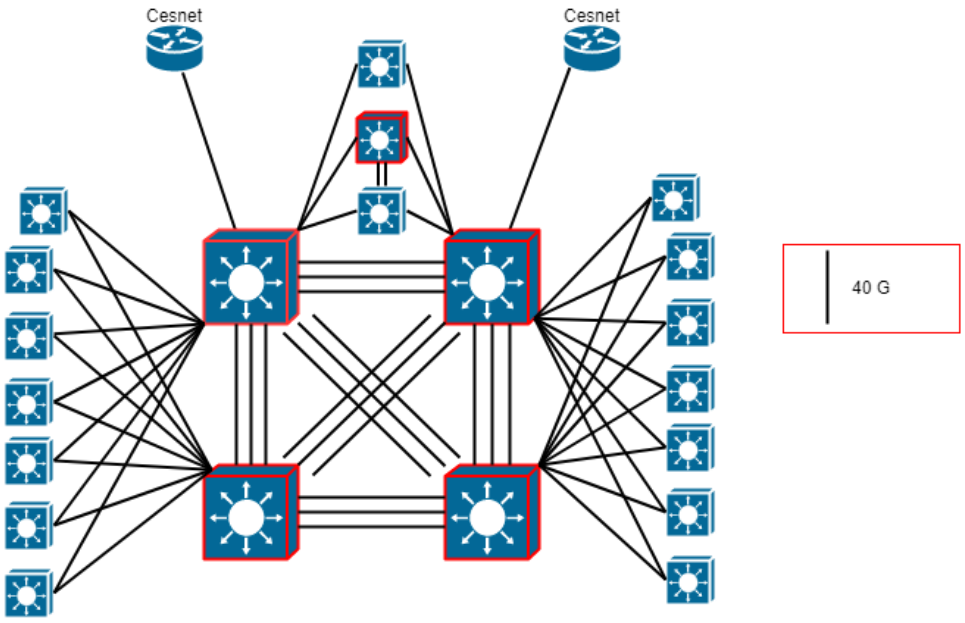
Wi-Fi

V súčasnosti je na univerzite rozmiestnených približne 3000 prístupových bodov od spoločnosti Cisco. Prevádzkované zariadenia patria najčastejšie do modelovej série Cisco Air AP2800 a sú napájané prostredníctvom PoE portov prístupových switchov v prístupovej vrstve siete. Univerzita poskytuje dve hlavné identifikácie siete (SSID) - Eduroam a UniverzitaXYZ. V prípade potreby, napríklad pri organizovaní konferencií, sú vytvárané dočasné SSID. V rámci prevádzky je využívaných 5 bezdrôtových kontrolérov Cisco 5520. Jeden z týchto kontrolérov pracuje na staršej verzií firmvéru, aby bolo zabezpečené zachovanie kompatibility so staršími access pointov, ktoré nepodporujú novšiu verziu. Kontroléry sú dostupné cez systém Cisco Prime, ktorý slúži na centrálne riadenie a správu týchto zariadení. Všetky kontroléry zároveň zasielajú logy na syslog server a synchronizujú ich s aplikáciou Cisco Prime pre efektívnejšie monitorovanie a správu siete.



Obrázok 3: Prehľad Wi-Fi topológie (Zdroj: Vlastné spracovanie)

Logická topológia



Obrázok 4: Logická topológia univerzity (Zdroj: Vlastné spracovanie)

Logická topológia sa skladá z troch logických vrstiev: prístupovej vrstvy, distribučnej vrstvy a chrbticovej vrstvy. Pripojenie do internetu má riešené cez ISP poskytovateľa Cesnet v dual homed prevedení.

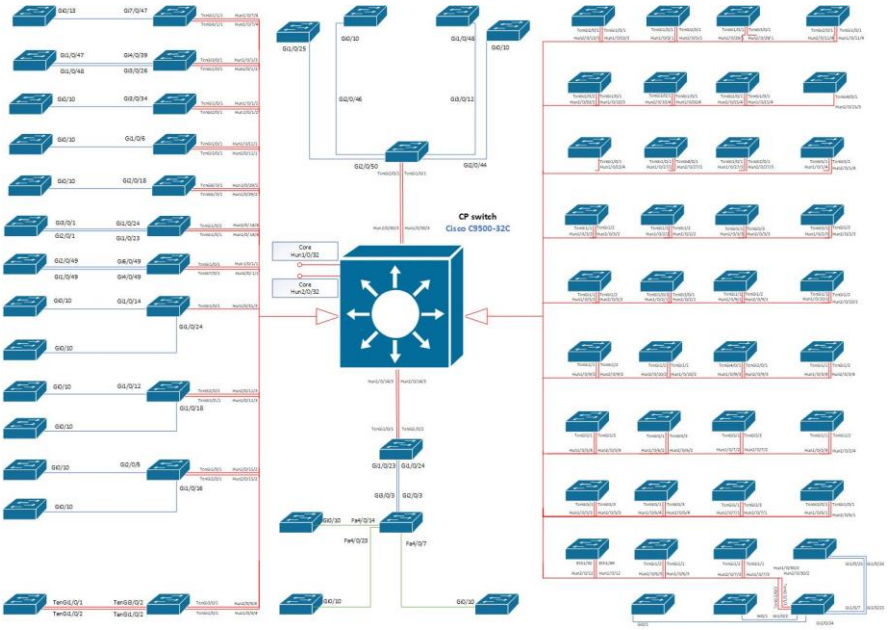
Chrbticovú vrstva je prepojená medzi sebou vo full mesh topológii troma 40G linkami do každého zariadenia a je tvorená L3 aktívnymi prvkami Cisco C9500. Každý aktívny prvok chrbticovej vrstvy sa nachádza na rozdielnych lokalitách. Distribučná vrstva je zložená z novších L3 Cisco C9500-24Y4C, ktoré znázorňujú individuálne jednu lokalitu (fakultu/internát). Distribučné switche sú do chrbticovej siete pripojené redundantne 40G linkami do dvoch rozdielnych chrbticových aktívnych prvkov.

Niektoré lokality sú riešené cez centralizovanú topológiu, ktorá je tvorená prístupovými switchmi poskytujúce pripojenie koncovým zariadeniam a centrálneho switchu, ktorý je redundantne pripojený 10G up-linkmi do distribučnej vrstvy.

Prístupová vrstva je po celej univerzite segmentovaná do VLAN, ktoré sú terminované a spravované na centrálnych switchov alebo na distribučnej vrstve. Na úrovni týchto VLAN sa vytvárajú ACL pravidlá pre obmedzovanie prevádzky siete. Najväčšia lokalita je rozsegmentovaná približne do 300 VLAN. Prístupová vrstva je tvorená približne 400 aktívnymi prvkami a sú tvorené modelmi Cisco 2960X series alebo novšími modelmi

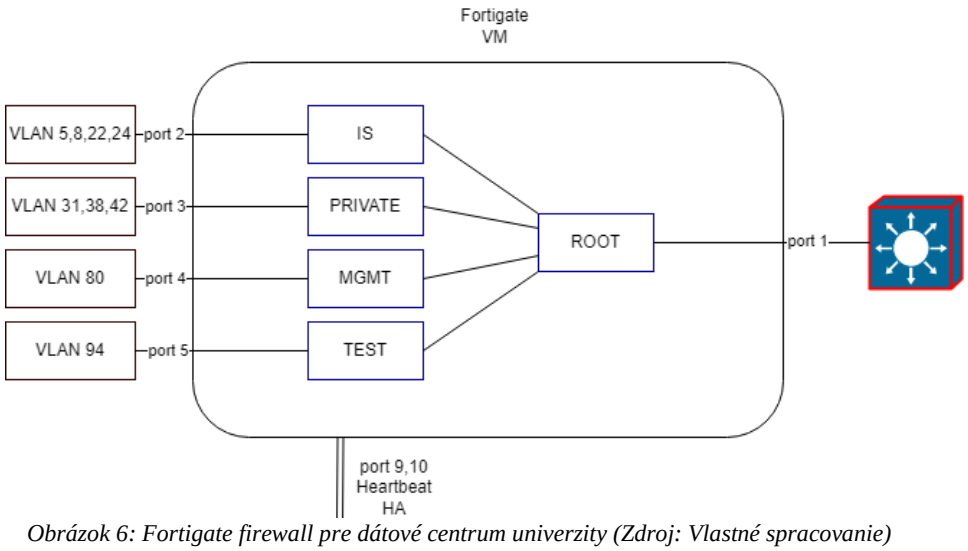
Cisco 9200 series. Redundancia na každej vrstve je riešená cez LACP protokol pre agregáciu liniek.

Univerzita má pre dátové centrum vytvorený bezpečnostný perimeter, ktorý chráni



Obrázok 5: Centralizovaná topológia lokality univerzity (Zdroj: Vlastné spracovanie)

virtualizovaný firewall Fortigate VM04 na clustery zložený z dvoch virtuálnych strojov v dvoch rozdielnych lokalitách. Stroje sú medzi sebou pripojené v HA režime dvoma linkami pre prípadný výpadok jedného z nich. Firewall je rozdelený do piatich virtuálnych domén (VDOM), z toho jedna je hlavná - *root*, pre služby univerzity – IS, pre externých zákazníkov – *private*, management aktívnych prvkov – *mgmt* a pre testovacie účely – *test*.



Obrázok 6: Fortigate firewall pre dátové centrum univerzity (Zdroj: Vlastné spracovanie)

Univerzita využíva jumphosta, ktorý slúži ako bezpečný prostriedok pre pripojenie k iným aktívnym prvkom v sieti. Tento prístup umožňuje sieťovým technikom pripojiť sa k jumphostovi a následne prepojiť sa s ostatnými zariadeniami v sieti prostredníctvom tohto hostiteľa. Okrem toho, jumphost plní aj úlohu sekundárneho DHCP servera pre adresné rozsahy v univerzitnej sieti. Na jumphostovi sú uložené Cron skripty, ktoré slúžia na vykonávanie rôznych kontrolných operácií nad aktívnymi prvky v sieti. Jedným z týchto skriptov, ktorý je spúšťaný pravidelne každý deň o piatej hodine ráno, je zálohovanie konfigurácií všetkých zariadení, vrátane SSH kľúčov a VLAN. Tento proces umožňuje centralizovať zálohy a zabezpečiť ich uchovanie v bezpečnom prostredí. Pre spustenie zálohovacieho skriptu je potrebné mať prístupové údaje do SNMP komunít, ktoré sú uložené na jumphostovi v XML formáte. Táto štruktúra je detailne popísaná na obrázku **č. 7**.

```
<device groups="XXX">
      <fqdn>XXX</fqdn>
      <snmp>
        <community version="2c" privilege="read-write">XXX</community>
        <community version="2c" privilege="read-only">XXX</community>
      </snmp>
      <filename>XXX</filename>
      <vendor>cisco</vendor>
      <type>WS-C2960S-48TS-L</type>
      <l2>true</l2>
      <l3>false</l3>
      <ipv6>false</ipv6>
      <backup>true</backup>
    </device>
```

Obrázok 7: XML štruktúra pre backup skript (Zdroj: Vlastné spracovanie)

Pokiaľ ide o TACACS, jumphosta možno použiť na riadenie riadenia prístupu a autentifikácie pre sieťové zariadenia, čo zahŕňa úlohy, ako je pridávanie a odstraňovanie používateľov alebo aktualizácia prístupových politík.

1.2.2 Dokumentačný systém NetBox

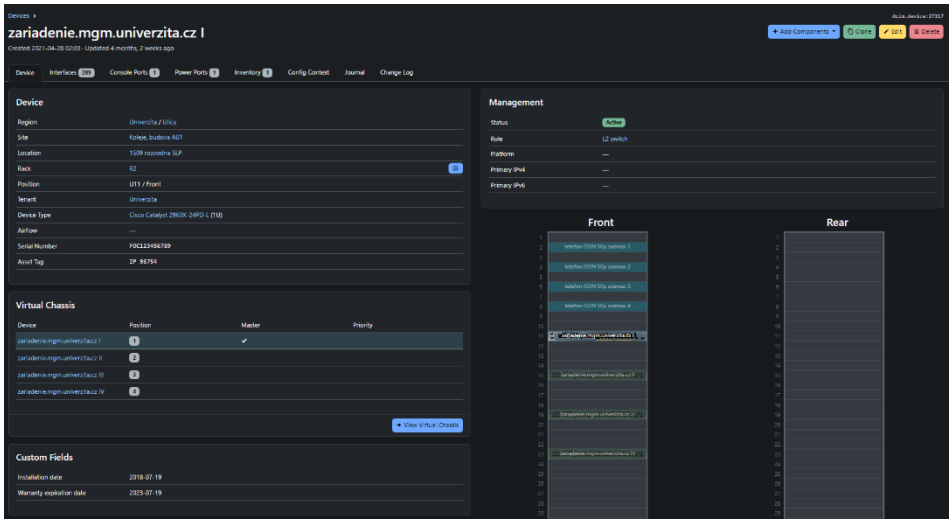
NetBox predstavuje open-source dokumentačný informačný systém, ktorý umožňuje systematické uchovávanie a riadenie informácií týkajúcich sa siete a príslušných zariadení. Jeho funkčnosť zahŕňa rôzne typy objektov pre podrobnú dokumentáciu, ako napríklad správa IP adries (IPAM), vlastníctvo zariadení, prenosové médiá a podobne. Štruktúra NetBox bola navrhnutá s dôrazom na jednoduché pridávanie a editovanie objektov, spolu s efektívnym vyhľadávaním informácií.

NetBox ponúka aj rozhranie API, ktoré umožňuje programátorom vytvárať vlastné aplikácie na základe informácií uložených v systéme. Toto API poskytuje prístup ku všetkým údajom a funkciám, umožňujúc jednoduchú integráciu s inými systémami. Okrem toho, NetBox umožňuje rozšírenie jeho funkcionality prostredníctvom pluginov, ktoré môžu byť vytvorené vo frameworku jazyku Python - Django. Tieto pluginy môžu pridávať nové typy objektov, rozširovať vlastnosti existujúcich objektov alebo pridávať nové funkcie do NetBoxu, čo umožňuje systém prispôsobiť individuálnym potrebám používateľov.

Univerzita aktívne využíva tento informačný systém na systematické uchovávanie informácií týkajúcich sa všetkých svojich aktívnych a pasívnych sieťových prvkov v rôznych lokalitách. NetBox slúži ako „source of truth“ pre tieto informácie. V súčasnej dobe je v systéme evidovaných viac než 700 aktívnych prvkov a viac ako 20 000 pasívnych prvkov vrátane koncových staníc. U aktívnych prvkov sa dokumentujú nasledujúce atribúty:

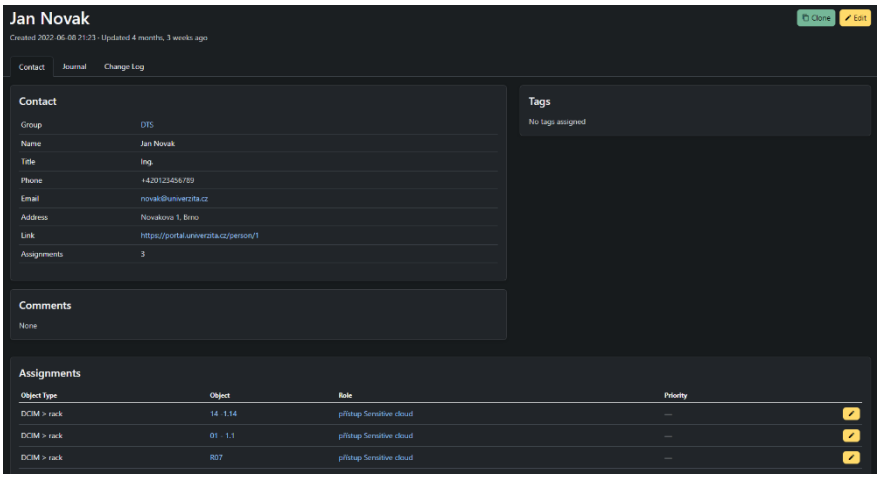
- FQDN zariadenia vo formáte - **zariadenie.mgm.univerzita.cz**
- Lokalita, kde sa zariadenie nachádza (Region->Site->Location->Rack->Unit pozícia)
- Vlastník zariadenia (tenant)
- Device type (model zariadenia)
- Miestnosť, v ktorom sa zariadenia nachádza
- Rack a jeho umiestnenie v ňom
- Sériové číslo
- Asset tag (pre inventarizáciu)
- Status (Online/Offline/Staged,...)

- Interface list
- Module bays (SFP moduly, Stack káble,…)



Obrázok 8: Zariadenie v IS NetBox (Zdroj: Vlastné spracovanie)

V prípade, že je zariadenie v tzv. stacku - súbor niekoľkých zariadení, ktoré sú prepojené riadiacimi linkami a ktoré tvoria jeden logický celok, je možné spojiť všetkých členov stacku do jedného objektu *virtual chassis.* Vytvoriť sa do stacku dajú aj zariadenia, ktoré sú v rôznych lokalitách. Tento objekt sa následne zobrazuje ako informácia u jednotlivých členov stacku. K zariadeniam je ďalej možné vytvárať voliteľné atribúty samotným užívateľom pomocou *custom fields*, ktoré rozširujú informácie o zariadeniach.



Obrázok 9: Kontakt v IS NetBox (Zdroj: Vlastné spracovanie)

V našom prípade sa do týchto polí zapisujú informácie o dátume inštalácie a dátumu záruky zariadenia.

Univerzita taktiež uchováva kontakty na zodpovedné osoby pre jednotlivé manažovateľné objekty, často to bývajú aktívne prvky a racky. Každý kontakt má

priradenú Contact Group do ktorej patrí, telefónne číslo, e-mailovú adresu, link na centrálny portál pre zamestnancov s detailnejšími informáciami o kontakte, a priradené objekty o ktoré je osoba zodpovedná.

Databázovú a serverovú časť dokumentačného informačného systému spravuje divízia informačných systémov. Zabezpečenie integrity a správnosti údajov o všetkých uložených objektoch je v kompetencii divízie IT infraštruktúry.

1.2.3 Cacti

Cacti predstavuje monitorovací systém, ktorý disponuje schopnosťou sledovať výkonnosť switchov a routerov v rámci siete. Užívatelia Cacti majú možnosť vytvárať grafy, ktoré monitorujú prenos dát, zaťaženie linky a rýchlosť prenosu na jednotlivých zariadeniach.

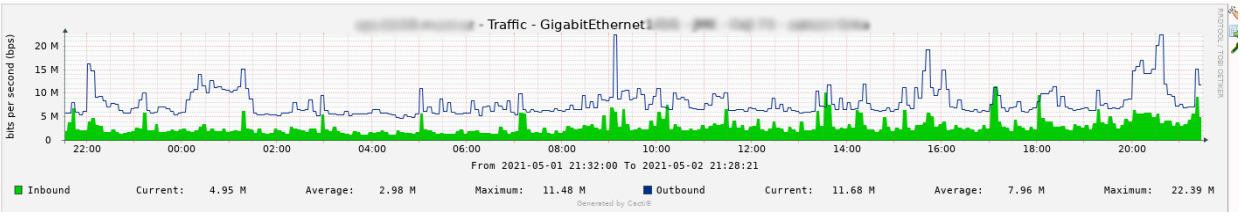
Systém Cacti využíva rôzne protokoly, vrátane SNMP a ICMP ping, na získavanie údajov o výkone zariadení a následne tieto údaje ukladá do platformy RRDtool pre neskoršie zobrazenie vo forme grafických vizualizácií. Tieto grafy umožňujú užívateľom monitorovať výkonnosť siete v reálnom čase a rýchlo identifikovať prípadné problémy s výkonom.

Univerzita využíva tento monitorovací systém výhradne na sledovanie grafov, ktoré zobrazujú agregovanú prevádzku na aktívnych sieťových prvokoch v rámci logických a fyzických portov. Tieto informácie slúžia na účely vytvorenia výročnej správy a sú taktiež zdieľané s Českým telekomunikačným úradom.

Nedostatky

- **Nepriehľadné GUI** – grafické rozhranie Cacti je či už pre správcu systému alebo pre samotných koncových používateľov vysoko nepriehľadné a je možné, že užívateľ strávi viacej času hľadaním ako samotným používaním
- **Nedostatočná podpora** – ako open-source projekt, neposkytuje tak časté aktualizácie pre monitorovací systém, ako by sa predpokladalo

- **Zložitosť konfigurácie** – kombinácia nepriehľadného používateľského grafického rozhrania a komplexnosť mnohých nastavení komplikuje efektívnemu používaniu tohto systému



Obrázok 10: Graf prevádzky v IS Cacti (Zdroj: Vlastné spracovanie)

1.2.4 Monitorovací systém Nagios

Nagios je monitorovací nástroj určený na sledovanie stavu IT infraštruktúry a služieb. Jeho použitie zahŕňa monitorovanie počítačov, sieťových zariadení, serverových aplikácií a ďalších prvkov IT.

V prípade, že systém zaznamená nezrovnalosť alebo problém, upozorní užívateľa prostredníctvom e-mailu, SMS správy alebo iných dostupných komunikačných kanálov. Nagios je k dispozícii ako open-source riešenie, napísané v jazyku C. Obsahuje konzolu na správu a konfiguráciu, ako aj webové rozhranie na prezeranie informácií o stave a monitorovaných prvkoch. Nagios môže byť ďalej rozšírený prostredníctvom rôznych pluginov, ktoré umožňujú prispôsobenie a monitorovanie špecifických služieb alebo prvkov.

Na univerzite je tento monitorovací systém využívaný ako hlavný nástroj na monitorovanie stavu aktívnych sieťových prvkov. V súčasnosti monitoruje iba službu hostalive, čo znamená, že sleduje dostupnosť zariadenia (online/offline). V prípade, že zariadenie prejde do stavu offline, notifikácia sa zasiela do centra pre monitorovanie IT infraštruktúry ústavu informačných služieb a výpočtovej techniky. Systém funguje na jedinom fyzickom serveri, čo predstavuje riziko pre celý informačný systém v prípade výpadku. V súčasnosti dochádza k postupnému vyraďovaniu tohto monitorovacieho systému na univerzite kvôli identifikovaným nedostatkom v jeho použití.

Nedostatky

- **Zložitosť konfigurácie** – konfigurácia a správa pre menej skúsených používateľov je časovo a vedomostne náročná, keďže neexistuje žiadna priehľadná a úplná dokumentácia pre vytváranie sónd, ich update alebo zoznam všetkých ponúkaných služieb
- **Náročnosť na výkon** – náročný na výkon pri veľkom množstve zariadení alebo služieb. Tento nedostatok sa prejavuje v spomalení alebo nestabilite monitorovacieho systému (pomalá odozva).
- **Nedostatok vylepšení** – Nagios ako open-source projekt je závislý na príspevkoch komunity pre vývoj a vylepšeniam, ktoré sú v súčasnosti na minimálnej úrovni. To spôsobuje, že niektoré vylepšenia alebo nové funkcie sa dostávajú do vydania pomaly alebo niekedy vôbec.
- **Nedostatok podpory** – keďže sa jedná o open-source projekt, neposkytuje žiadnu komerčnú podporu. Ak správca vyžaduje pomoc s nastavením alebo riešením problémov, musí hľadať pomoc v online komunite alebo si najať externého konzultanta.
- **Nedostatok integrácie s inými nástrojmi** – nie je schopný sa integrovať s inými nástrojmi na monitorovanie. To spôsobuje, že univerzita musí používať viacero rôznych nástrojov na riadenie infraštruktúry – napr. Cacti
- **Bezpečnostné diery** – kvôli nedostatočnej podpore od vývojárov a nie príliš častým aktualizáciám, sa v systéme objavujú bezpečnostné hrozby

1.2.5 Portál pre zamestnancov

S cieľom uľahčiť prístup k mnohým nástrojom a zjednodušiť ich používanie, boli nástroje, ktoré sú nutné pre každodennú prácu zamestnancov univerzity integrované do jedného webového portálu. Portál slúži ako centralizovaný bod pre prístup k týmto nástrojom, čo má za následok zvýšenie efektivity a pohodlia pre zamestnancov univerzity.

Pre prihlásenie sa na tento portál je používaný jednotný systém pre jednotné prihlasovanie (Single Sign-On - SSO) s využitím protokolu SAML (Security Assertion Markup Language). Na základe toho, v ktorom sektore univerzity zamestnanec pracuje, sú mu

pridelené individuálne práva prístupu k jednotlivým nástrojom. Portál ponúka rôzne nástroje, ktoré sú prístupné z jedného miesta. Niektoré z týchto nástrojov zahŕňajú:

- NetBox
- Dokumentačný portál Confluence
- Icinga
- Správa Fortigate firewallu
- Gitlab
- Ticket systém RT
- Netstork
- Perun
- VMware

1.3 Požiadavky investora

Univerzita požaduje nájdenie a implementáciu nového monitorovacieho informačného systému. Vzhľadom na snahu univerzity o optimalizáciu kapitálových nákladov, sa žiada, aby informačný systém bol postavený na open-source distribúcii. Za správu databázovej a serverovej časti tohto systému bude zodpovedná divízia informačných systémov, zatiaľ čo monitorovanie a konfigurácia bude v kompetencii divízie IT infraštruktúry, na ktorú sa bude táto práca opierať.

Pre načítanie dát do nového monitorovacieho informačného systému sa bude využívať dokumentačný systém NetBox ako hlavný zdroj údajov. Ďalším požiadavkom je, aby sa dáta z NetBoxu plnili do nového monitorovacieho systému automaticky. Nový monitorovací informačný systém by mal obsahovať sondy, ktoré sledujú kritické body aktívnych prvkov sieťovej infraštruktúry.

2 TEORETICKÉ VÝCHODISKA PRÁCE

V tejto kapitole sa budú prezentovať základné teoretické predpoklady, ktoré budú slúžiť ako základy pre implementáciu monitorovacieho systému pre sieťové aktívne prvky. V úvodnej časti sa budú rozoberať základy počítačových sietí, následne bude analyzovaná problematika informačnej bezpečnosti, správy siete, monitorovania a funkčného modelovania.

2.1 Počítačová sieť

Počítačová sieť predstavuje komunikačné spojenie medzi viacerými koncovými zariadeniami, umožňujúce prenos a prijímanie dát medzi týmito prvkami. Toto spojenie sa skladá z aktívnych a pasívnych prvkov, ako sú napríklad routery, switche, káble a patch panely. Skupinu týchto prvkov môžeme považovať za sieť, ak spĺňa nasledujúce atribúty: obsahuje prepojovací softvér, sieťový hardware, fyzické prenosové média a adresný systém (5).

2.2 Sieťová architektúra TCP/IP

V súčasnosti patrí medzi najpoužívanejšiu sieťovú architektúru. Komunikáciu na internete najčastejšie popisujú dva protokoly: TCP a IP, podľa ktorých aj táto architektúra vznikla. Tento model v prítomnosti popisuje skoro celý internet. Komunikácia v sieťach TCP/IP prebieha v nasledujúcich štyroch vrstvách:

1. Vrstva sieťového rozhrania,
2. Vrstva Internet,
3. Transportná vrstva,
4. Aplikačná vrstva

V koncových uzloch sú implementované všetky uvedené vrstvy, zatiaľ čo v aktívnych prvkoch, ako sú napríklad routery, sú implementované iba spodné dve vrstvy (vrstva sieťového rozhrania a vrstva Internet) (5, 6).

2.2.1 Vrstva sieťového rozhrania

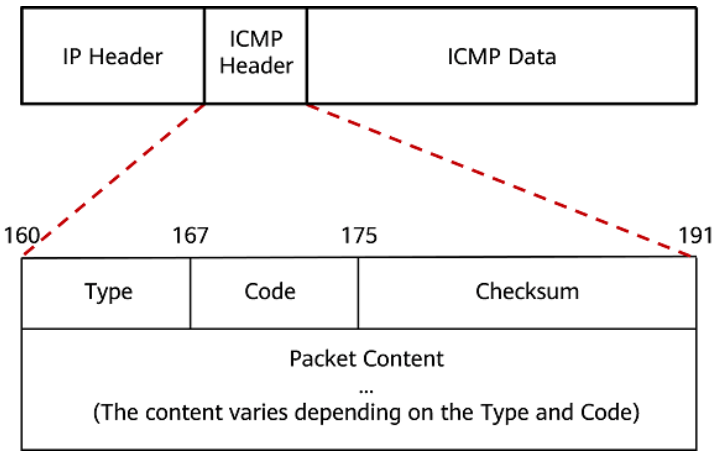
Nachádza sa v najnižšej vrstve TCP/IP architektúry. Jeho úlohou je odkazovať na fyzickú a linkovú vrstvu modelu ISO/OSI. Poskytuje prístup k fyzickému prenosovému médiu a jeho charakter sa priamo odvíja od konkrétnej implementácie, pričom sa môže líšiť v závislosti od použitej prenosovej technológie, ako sú napríklad Ethernet, Wi-Fi, FDDI a podobne (5, 6).

2.2.2 Vrstva Internet

Táto vrstva sa nachádza nad vrstvou sieťového rozhrania a zastáva svoje funkcie v rámci sieťovej vrstvy modelu ISO/OSI. Jej hlavnou úlohou je nájsť cestu pre prenos blokov dát medzi dvoma uzlami v sieti. Zodpovedá za smerovanie, prepojovanie, fragmentáciu a defragmentáciu datagramov, pričom poskytuje iba nespojovaný a nespoľahlivý prenos. V oblasti adresácie umožňuje používanie dvoch verzií IP, a to IPv4 a IPv6. Z týchto dvoch verzií je najčastejšie využívaná IPv4. Príklady protokolov, ktoré sa nachádzajú na tejto vrstve, zahŕňajú IP, ARP, RARP, ICMP, IGMP, OSPF, IGRP a IPSEC (5, 6).

ICMP

ICMP je protokol, ktorý definuje správy, ktoré systémy v IP sieti zasielajú ako požiadavky alebo reakcie na udalosti v procese prenosu dát. Tento protokol operuje nad IP protokolom, a preto sú pakety ICMP zabalené do IP paketov. Jeho hlavným účelom je signalizácia korektného doručenia paketov, žiadosti o zaslanie paketov a kontrola smerovania. Jedným z základných diagnostických nástrojov na kontrolu stavu zariadení v sieti je odoslanie ICMP Echo Request pomocou programu Ping. Dotazované zariadenie odpovedá na túto výzvu správou Echo Reply. ICMP správy sú generované na základe IP datagramov, pričom tieto správy sa nikdy nedelia a vždy sú obsiahnuté v jednom datagrame. Typ správy ICMP sa identifikuje podľa prvého poľa vloženého na pozícii 160 v pakete (5).



Obrázok 11: ICMP hlavička (Zdroj: (11))

2.2.3 Transportná vrstva

Táto vrstva prevyšuje vrstvu Internet a svojím účelom odpovedá transportnej vrstve podľa modelu ISO/OSI. Je zodpovedná za koncový prenos dát medzi komunikujúcimi procesmi a používa číslo portu na adresáciu. Poskytuje aplikáciám možnosť voľby medzi rýchlym, nespoľahlivým a nespojovaným prenosom pomocou protokolu UDP alebo spoľahlivým a spojovaným prenosom dát pomocou protokolu TCP. Táto vrstva ponúka dva hlavné typy služieb: datagram a stream.

V prípade datagramu sú dáta transportnej vrstvy predané cez socket už rozdelené na bloky, a transportná vrstva ich dodáva sieťovej vrstve v nezmenenej podobe. Túto službu zabezpečuje protokol UDP. Naopak, v prípade streamu sú dáta transportnej vrstvy predané cez socket ako spojitý prúd bytov. Transportná vrstva rozdelí dáta na bloky vhodné pre prenos a odovzdá ich sieťovej vrstve. Túto službu zabezpečuje protokol TCP (6, 7).

TCP

Je spojovo orientovaný protokol pre prenos na transportnej vrstve (L4), ktorý sa využíva na spoľahlivé doručovanie. Aplikácie na koncových uzloch v počítačovej sieti môžu vytvoriť spojenie pomocou TCP, čím umožňujú prenos dát medzi sebou. Tento protokol zabezpečuje spoľahlivé doručovanie, udržiavanie správneho poradia doručených paketov, potvrdzovanie prijatia paketov zo strany druhej strany, riadenie zahltenia a toku

dát. Okrem toho dokáže rozlišovať dáta určené pre viacnásobne súčasne spustené aplikácie na rovnakom koncovom zariadení. TCP využíva služby IP protokolu (8).

UDP

Je alternatívnym protokolom k TCP, fungujúcim na úrovni transportnej vrstvy (L4). Na rozdiel od TCP nevytvára priame spojenie medzi komunikujúcimi koncovými zariadeniami. Odosielateľ jednoducho odosiela paket, a následne sa stará aplikačný protokol o to, či bol paket úspešne doručený. Podobne ako TCP, aj UDP identifikuje aplikácie na koncovom zariadení pomocou portov. Pri tomto protokole sa neodporúča fragmentácia dát, aj keď je teoreticky možná. UDP umožňuje aj odosielanie broadcast alebo multicast paketov viacerým užívateľom súčasne. Je často využívaný pri "real-time" prenosoch, ako sú prenosy hudby, videa a podobne, pri ktorých je prenášaný veľký objem dát a potvrdenie doručenia by bolo pre sieť náročné (8).

2.2.4 Aplikačná vrstva

Je najvyššou vrstvou v architektúre TCP/IP, ktorá odpovedá relačnej vrstve, prezentačnej vrstve a aplikačnej vrstve relačného modelu ISO/OSI. Predstavuje vrstvu aplikácií, ktoré využívajú prenos dát v sieti. Zahrňuje súbor protokolov, ktoré poskytujú užívateľské a systémové sieťové služby. Sú tu implementované predovšetkým služby, ktoré sú často využívané aplikáciami a preto museli byť štandardizované. Medzi protokoly patria napríklad DNS, Telnet, FTP, TFTP, SMTP, HTTP, SIP a SNMP (5, 6).

SNMP

SNMP (Simple Network Management Protocol) je protokol navrhnutý na monitorovanie a správu aktívnych prvkov v počítačovej sieti. Je asynchrónny a transakčne orientovaný, postavený na klient/server modeli, kde porty 161 a 162 sú vyhradené na komunikáciu. Jeho základné časti sú:

1. ***Riadené objekty*** – jedná sa o individuálne prvky, ktoré sú predmetom správy, ako napríklad sieťové karty, aktívne prvky (switche, routre) a ďalšie.
2. ***Agenti*** – drobné softwarové moduly bežiace na riadených objektoch. Zbierajú informácie o objektoch a sieťovej komunikácii a ponúkajú ich prostredníctvom dotazov protokolu SNMP

3. ***Báza MIB*** (Management Information Base) – je databáza informácií o riadených objektoch. Väčšina poskytovaných dát prostredníctvom SNMP je v režime read-only, obsahujúc napríklad hardwarové informácie. Niektoré informácie sú modifikovateľné, ako je napríklad názov zariadenia.
4. ***SNMP manager*** – software pre tvorbu dotazov a zber dát (5, 8).

Súbory MIB (Management Information Base) sú organizované do hierarchickej názvovej štruktúry vo forme stromu. Každý uzol stromu je identifikátorom objektu (OID). Typy jednotlivých OID môžu byť v režime READ (hodnota je možné čítať), SET (hodnota je možné meniť), alebo oboje (5).

Existujú rôzne typy požiadavkov v rámci SNMP:

1. ***GetRequest*** – požiadavka od manažéra k agentovi pre načítanie premennej alebo zoznamu premenných (OID)
2. ***SetRequest*** – nastavenie nových hodnôt jednej premennej alebo viacerým premenným
3. ***GetNextRequest*** – zistenie informácie o dostupných premenných. Pri opakovaných dotazoch vracia lexikograficky nasledujúcu premennú v MIB
4. ***GetBulkRequest*** – optimalizovaná verzia GetNextRequest uvedená v SNMPv2, ktorá ponúka viacero iterácii GetNextRequest (5).

Autentizácia:

1. ***SNMPv1*** – poskytuje autentizáciu prostredníctvom "community stringu", ktorý funguje ako heslo pre prístup k informáciám na zariadení. Táto verzia však má obmedzené zabezpečenie, a preto nie je ideálna v prípade požiadaviek na vyššiu bezpečnosť.
2. **SNMPv2** – predstavil "party-based" zabezpečovací model, ktorý bol komplexný svojou funkčnosťou, ale nebol všeobecne akceptovaný verejnosťou. Jeho implementácia bola problematická a neprinášala dostatočnú úroveň zabezpečenia.
3. ***SNMPv2c*** – snažil sa riešiť nedostatky pôvodného SNMPv2. V autentizácii využíva rovnaký mechanizmus ako SNMPv1 s použitím "community stringu", čím poskytuje jednoduchšiu formu autentizácie.

4. ***SNMPv3*** – je najnovšou verziou a prináša podstatné vylepšenia v oblasti autentizácie a zabezpečenia. SNMPv3 umožňuje autentizáciu prostredníctvom metódy HMAC (Hash-based Message Authentication Code) a šifrovanie pomocou AES (Advanced Encryption Standard). Taktiež implementuje dva modely zabezpečenia: User Based Security Model (USM) pre autentizáciu a šifrovanie a View-Based Access Control Model (VACM) pre riadenie prístupu k informáciám (9).

Telnet

je starší protokol, ktorý funguje na porte 23. Tento protokol používa nezabezpečený prenos v "plain text" formáte pre prihlasovaciu autentifikáciu (používateľské meno a heslo) a pre prenos dát medzi komunikujúcimi zariadeniami. Často sa využíva na manuálne nadviazanie spojenia s otvorenými portmi služieb servera, ako napríklad HTTP (10).

SSH

Secure Shell (SSH) je zabezpečený protokol, ktorý využíva TCP port 22. Tento protokol poskytuje bezpečné a šifrované pripojenie pre správu vzdialených zariadení. V porovnaní s Telnetom, ktorý bol historicky používaný na pripojenia na správu, SSH prináša výrazné zlepšenia v oblasti bezpečnosti. Zabezpečuje autentifikáciu zariadenia pomocou silného šifrovania pre používateľské meno a heslo, a taktiež chráni prenášané dáta medzi komunikujúcimi zariadeniami (10).

2.3 Aktívne prvky

2.3.1 Switch

Je aktívny sieťový prvok, ktorý operuje na druhej vrstve ISO/OSI modelu a na adresáciu využíva MAC adresy. Neanalyzuje kompletné informácie (protokoly) prenášané v dátovej časti rámca, ako napríklad IPv4 alebo IPv6 ND pakety. Switch dynamicky vytvára tabuľku MAC adries skúmaním zdrojových MAC adries rámcov, ktoré prídu na port. Rámce smeruje na základe zhody cieľovej MAC adresy v rámci so záznamom v tabuľke MAC adries. Ak zdrojová adresa MAC nie je známa, pridá sa do tabuľky spolu

s číslom prichádzajúceho portu. Ak adresa už existuje, aktualizuje sa časovač obnovy pre túto položku v tabuľke. Cieľová MAC adresa určuje port, cez ktorý sa rámec odosiela. V prípade neznámej cieľovej MAC adresy sa rámec odosiela na všetky porty okrem prichádzajúceho (neznámy unicast) (10).

Switche používajú na prepínanie rámcov jednu z dvoch metód:

- ***Store-and-forward switching*** - táto metóda rozhoduje o preposielaní rámca po prijatí celého rámca a kontrole chýb rámca pomocou matematického mechanizmu kontroly chýb (CRC).
- ***Cut-through switching*** - táto metóda začína proces preposielania po určení cieľovej adresy MAC prichádzajúceho rámca a výstupného portu (10).

Dve z najzákladnejších nastavení switchu sú šírka pásma a duplexné nastavenia pre každý jednotlivý port prepínača. Na komunikáciu v sieti Ethernet sa používajú dva typy duplexných nastavení:

- ***Full-duplex*** - Oba konce pripojenia môžu súčasne odosielať a prijímať.
- ***Half-duplex*** - Naraz môže odosielať iba jeden koniec pripojenia.
- ***Autonegotiation*** - je voliteľná funkcia, ktorá umožňuje dvom zariadeniam automaticky dohodnúť najlepšiu rýchlosť a možnosti duplexu. Full-duplex sa vyberie, ak obe zariadenia majú túto schopnosť spolu s najvyššou spoločnou šírkou pásma (10).

Agregácia liniek

Predstavuje technologický prístup, ktorý slúži na spojenie viacerých fyzických ethernetových liniek do jedného logického spojenia. Táto technológia poskytuje viacero výhod, vrátane zvýšenej odolnosti voči chybám, efektívneho zdieľania záťaže, rozšírenia šírky pásma a zabezpečenia redundancie v komunikácii medzi switchmi, routermi a servermi.

Hlavnými cieľmi použitia agregácie liniek sú:

1. **Odolnosť voči chybám**: Technológia agregácie liniek umožňuje, aby bola sieť odolnejšia voči výpadkom jednotlivých liniek. V prípade poruchy na jednej linke

sa prevádzka automaticky prenáša na ostatné dostupné linky, čím sa minimalizuje vplyv chýb na konektivitu.

2. **Zdieľanie záťaže**: Agregácia liniek umožňuje efektívne zdieľanie sieťovej záťaže medzi viacerými fyzickými linkami. Týmto spôsobom sa optimalizuje využitie šírky pásma a zabezpečuje sa rovnomerné rozloženie komunikačnej záťaže.
3. **Zvýšenie šírky pásma**: Spojenie viacerých liniek do jedného logického spojenia zvyšuje celkovú šírku pásma, čo je výhodné pre aplikácie vyžadujúce vysokú prenosovú kapacitu.

Existuje viacero proprietárnych riešení pre implementáciu agregácie liniek, vrátane známych implementácií ako Etherchannel od spoločnosti Cisco, Aggregated Ethernet od Juniper alebo Eth-trunk od Huawei. Každé z týchto riešení ponúka špecifické vlastnosti a konfiguračné možnosti, prispôsobujúc sa potrebám konkrétnych sieťových prostredí (10).

2.3.2 Router

Je aktívnym prvkom, ktorý pracuje na úrovni sieťovej vrstvy ISO/OSI. Predtým, ako smerovač prepošle paket kamkoľvek, musí určiť najlepšiu cestu, ktorou sa má paket vydať. Smerovač spája viacero sietí, čo znamená, že má viacero rozhraní, z ktorých každé patrí do inej siete IP. Keď smerovač prijme paket IP na jednom rozhraní, určí, ktoré rozhranie sa použije na presmerovanie paketu do cieľa. Tento proces je známy ako smerovanie. Rozhranie, ktoré router používa na preposielanie paketov, môže byť konečným cieľom alebo to môže byť sieť pripojená k inému routeru, ktorý sa používa na dosiahnutie cieľovej siete. Primárnymi funkciami routera je určiť najlepšiu cestu na preposielanie paketov na základe informácií v jeho smerovacej tabuľke a posielať pakety smerom k ich cieľu. Obsahuje celkovo 4 komponenty: vstupné porty, prepínaciu štruktúru, výstupné porty a smerovací procesor (10).

2.4 Systém riadenia bezpečnosti informácii

ISMS (Systém riadenia bezpečnosti informácií) predstavuje systematický prístup k riadeniu citlivých informácií o spoločnosti s cieľom ich efektívnej ochrany. Tento prístup

zahŕňa ľudí, procesy a systémy IT aplikácií v rámci procesu riadenia rizík. ISMS tvorí súbor štandardov, odporúčaní, postupov a kontrol, ktoré majú za úlohu zabezpečiť bezpečnosť aktív spoločnosti. Jeho hlavným poslaním je pomáhať spoločnostiam rôznych veľkostí a odvetví udržať informácie o podnikových činnostiach v bezpečí. Tento systém je postavený na využití PDCA modelu, ktorý zahŕňa štyri hlavné etapy: ustanovenie ISMS, zavedenie a prevádzka ISMS, monitorovanie a preskúmavanie ISMS, a nakoniec údržba a neustále zlepšovanie ISMS. (4, 12).

2.4.1 Základné pojmy ISMS

Aktívum

Aktívum predstavuje všetko, čo má hodnotu pre vlastníka a môže byť ohrozené alebo úplne zničené v dôsledku hrozby. Aktíva rozdeľujeme na:

- ***Hmotné*** – movité a nemovité veci, finančné prostriedky, cenné papiere,…
- ***Nehmotné*** – software, informácie, autorské práva, patenty, služby, znalosti

V obecnej rovine môže byť aktívum proces, dej, udalosť ale aj samotný subjekt (4, 8).

Hrozba

Predstavuje situáciu, udalosť, aktivitu alebo subjekt s potenciálom spôsobiť nežiadúci incident, škodu alebo ovplyvniť udalosti nežiadúcim smerom. Hrozby môžu byť klasifikované do kategórií ako ľudské (úmyselné alebo neúmyselné), technické, technologické, fyzické a prírodné. Následky vyplývajúce z hrozby sa označujú ako dopad hrozby. Kľúčovými charakteristikami hrozby sú jej úroveň, ktorá sa posudzuje na základe faktorov:

- ***Nebezpečnosť*** – schopnosť hrozby spôsobiť škodu
- ***Prístup*** – možnosť hrozby pôsobiť na aktívum
- ***Motivácia*** – záujem o naplnenie hrozby (4, 8).

Zraniteľnosť

Zraniteľnosť aktíva predstavuje slabosť, nedostatok alebo chybu aktíva, ktorú môže hrozba využiť na uplatnenie nežiaduceho vplyvu. Vyjadruje, do akej miery je aktívum citlivé na vplyv konkrétnej hrozby a vzniká v dôsledku interakcie medzi hrozbou a

aktívom. Úroveň zraniteľnosti sa posudzuje podľa citlivosti, teda toho, do akej miery je aktívum náchylné na poškodenie danou hrozbou (4, 8).

Riziko

Riziko predstavuje potenciálnu hrozbu vzniku poškodenia, škody, straty, zničenia alebo iného nezdaru. Je výsledkom kombinácie hrozby a zraniteľnosti s následným dopadom na aktívum. Riziko vytvára stav neistoty, ktorý je vyjadrený pravdepodobnosťou nežiaduceho vývoja v rozmedzí hodnôt od 0 do 1. Nežiaduci vývoj nie je ani úplne istý, ani úplne nemožný. Za určitých podmienok je možné túto pravdepodobnosť opísať pomocou matematických nástrojov, ako sú štatistické funkcie a rozdelenie pravdepodobností, pokiaľ sú k dispozícii dostatočné vstupné informácie (4, 8).

Bezpečnostná udalosť

Môže sa definovať ako identifikovaný stav informačného systému, služby alebo počítačovej siete, ktorý by mohol narušiť pravidlá bezpečnostnej politiky alebo predstavovať zlyhanie protiopatrení. Môže ísť o neznámu, nepredpokladanú situáciu, ktorá má potenciálne ovplyvniť bezpečnosť. Samotný výskyt bezpečnostnej udalosti ešte nekvalifikuje ako bezpečnostný incident. Až po vyhodnotení môže byť bezpečnostná udalosť kvalifikovaná ako bezpečnostný incident (18).

Bezpečnostný incident

Bezpečnostný incident predstavuje jednu alebo viac nežiaducich bezpečnostných udalostí, ktoré s vysokou pravdepodobnosťou môžu narušiť podporu hlavných alebo podporných procesov organizácie alebo môžu viesť k ohrozeniu bezpečnosti informačného systému, pričom aktívum môže preukazovať zmenené vlastnosti (18).

Opatrenie

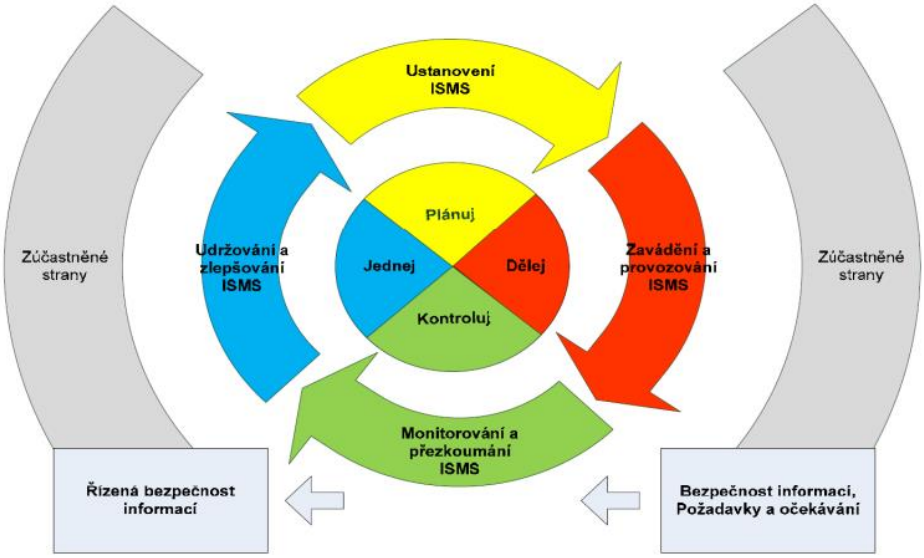
Opatrením sa rozumie postup, proces, fyzický prostriedok, služba alebo akýkoľvek iný prostriedok, ktorý je zameraný na zmiernenie dopadov hrozby alebo ich úplné odstránenie. Cieľom opatrení je znižovať riziko, a ich úroveň sa hodnotí ako rozsah opatrení. Efektívnosť opatrení v reálnom procese sa posudzuje na základe kritéria známeho ako účinnosť opatrení (8).

2.4.2 PDCA cyklus

Známy aj ako Demingov cyklus je metóda postupného zlepšovania kvality výrobkov, služieb, procesov, aplikácii alebo dát prebiehajúce formou opakovaného vykonávania štyroch základných činností:

- ***Plan*** – plánuj – plánovanie zamýšľaného zlepšenia
- ***Do*** – konaj – realizácia plánu
- ***Check –*** kontroluj – overenie výsledku realizácie oproti pôvodnému zámeru
- ***Act*** – jednaj – úpravy zámeru a vlastného prevedenia na základe overenia

Súčasťou modelu PDCA je taktiež dokumentácia každej jeho etapy ako jedna z kľúčových častí celého modelu. Procesy je nutné identifikovať, popísať a zdokumentovať, riadiť na základe dokumentácie, optimalizovať ich priebeh (4).



Obrázok 12: PDCA cyklus (Zdroj: (4))

2.4.3 Analýza aktív

Analýza aktív zahrnuje identifikáciu aktív a následné postupné hodnotenie. K tomu, aby bolo možné aktíva oceniť, je nevyhnutné ich predtým identifikovať. V procese

identifikácie sa odporúča zoskupiť všetky aktíva do logických celkov (napr. bezpečnostné aktíva, obchodné aktíva) a k nim priradiť vlastníka.

Následne je potrebné stanoviť stupnicu a hodnotiace kritéria, ktoré sa budú používať pri hodnotení aktív. Organizácia môže zvoliť, či preferuje stupnicu vyjadrenú finančne alebo kvalitatívnymi hodnotami. Hlavným princípom pri hodnotení sú náklady, ktoré môžu vzniknúť v dôsledku porušenia dôvernosti, integrity a dostupnosti. Na výpočet hodnotenia aktíva sa často využíva súčtový algoritmus, pri ktorom sa sčítajú všetky tri kritériá a výsledok sa delí tromi (4).

2.4.4 Analýza rizík

Analýza rizík sa uskutočňuje s cieľom identifikovať zraniteľné body v informačnom systéme organizácie. V rámci tohto procesu sa zostavuje zoznam hrozieb, ktoré môžu ovplyvniť informačný systém, a určujú sa riziká spojené s každým identifikovaným zraniteľným bodom a hrozbou. Účelom analýzy rizík je buď znížiť riziká na akceptovateľnú úroveň, alebo akceptovať zvyšujúce sa riziká tam, kde by minimalizácia nebola efektívna.

V prvom kroku sa identifikujú aktíva a ohodnotia sa s cieľom preskúmať potenciálne dopady narušenia dostupnosti, integrity a dôvernosti. V ďalšom kroku sa hodnotia hrozby a zraniteľnosti. Cieľom je vyhodnotiť potenciálne typy hrozieb, ktoré by mohli viesť k výskytu následkov identifikovaných v predchádzajúcej fáze. Hodnotenie hrozieb a zraniteľností spolu s ohodnotenými aktívami z prvého kroku umožňuje určiť úroveň rizík systému. Úroveň rizika pre každé aktívum alebo skupinu aktív sa stanovuje kombinovaním hodnoty aktíva s hodnotením hrozby a zraniteľnosti.

V poslednom kroku, na základe výpočtu úrovne rizika, sa vyberajú vhodné opatrenia. Tieto odporúčané opatrenia sú kategorizované podľa oblastí bezpečnosti, konkrétne IT bezpečnosti, komunikačnej bezpečnosti, personálnej bezpečnosti, administratívnej bezpečnosti a fyzickej bezpečnosti (4).

2.4.5 Riadenie rizík

Cieľom procesu riadenia rizík je identifikovať a kvantifikovať riziká, s ktorými je potrebné sa vyrovnať, a následne rozhodnúť o ich riadnom zvládnutí. Proces pozostáva z postupného vykonávania štyroch vzájomne prepojených etáp. Prvá fáza, stanovenie kontextu, zahŕňa vymedzenie základných aspektov, ako je výber metodiky na analýzu rizík, definícia rolí a zodpovedností v rámci procesov, a určenie spôsobov hodnotenia a

Obrázok 13: Riadenie rizík (Zdroj: (4))

riadenia rizík. Po analýze rizík sa identifikujú a kvantifikujú aktíva spolu s hrozbami a zraniteľnosťami, a stanovujú sa úrovne rizík. Pri vyhodnocovaní rizík sa tieto priorizujú a vyberajú sa optimálne opatrenia na ich znižovanie. V poslednej fáze sa rozhoduje o vhodných spôsoboch riadenia rizík, vrátane možností ako ich redukovať, transferovať, poistiť, a ďalšie (4).

2.5 Management siete

Správa siete je nevyhnutná pre efektívne fungovanie sietí rôznych rozmerov. S nárastom počtu aktívnych prvkov v sieti sa zvyšuje možnosť vzniku potenciálnych problémov pre správcu siete. Na dosiahnutie štandardizácie balíčkov pre správu siete sa využila klasifikácia FCAPS (5).

2.5.1 FCAPS

Akronym FCAPS bol vytvorený s cieľom zahrnúť kľúčové oblasti pre efektívnu správu sietí. Tieto oblasti sú reprezentované prvými písmenami nasledujúcich slov: **Fault** (chyba), **Configuration** (konfigurácia), **Accounting** and **Administration** (účtovanie a správa), **Performance** (výkon), **Security** (bezpečnosť). Využívanie akronymu FCAPS

pomáha správcom sietí zabezpečiť komplexnú správu siete a efektívne riešiť všetky relevantné aspekty jej fungovania (5).

Správa porúch

Správa porúch sa zaoberá identifikáciou a riešením softwarových alebo hardwarových chýb, ktoré môžu viesť k nežiaducim dôsledkom. Hlavným cieľom správy chýb je určiť čas vzniku chyby, izolovať ju a poskytnúť informácie potrebné na jej odstránenie. Systémy pre správu chýb operujú na základe detekcie špecifických udalostí spojených s chybovými stavmi alebo na princípe identifikácie chýb na základe určitej množiny udalostí (5).

Udalosti sú charakterizované určitým typom a môžu byť selektívne zaznamenávané. Dôležité udalosti sú následne logované alebo odosielané cez sieť pomocou protokolov aplikačnej vrstvy, prípadne špecifických proprietárnych protokolov niektorých výrobcov. Izolácia a podrobné porozumenie udalostiam predstavujú zásadnú časť správy chýb (5).

Okrem detekcie chýb sa v rámci správy chýb taktiež generujú alarmy. Alarm predstavuje stav detekovanej chyby, ktorý je kategorizovaný podľa typu a závažnosti, pričom sa zaznamenáva do databázy alebo sa zasiela formou oznámenia. V prípade, že zariadenie alebo systém automaticky odosiela alarm, takýto systém je označovaný ako pasívny systém pre správu. V situácii, kedy program vyžaduje informácie od zariadenia, ktorý reaguje na prezentačný signál (heartbeat), napríklad cez príkaz Ping, hovoríme o aktívnom systéme pre správu (5).

Existujú dva hlavné typy kategorizácie alarmov: digitálny a analógový. Digitálny alarm pracuje v binárnom systéme s dvoma stavmi, napríklad zapnutý (1) a vypnutý (0). Na druhej strane je analógový alarm charakterizovaný schopnosťou nadobúdať hodnoty v určitom rozsahu. Príkladom analógového alarmu môže byť sledovanie zaťaženia procesoru zariadenia. Tieto odlišné kategórie poskytujú širšie možnosti monitorovania a identifikácie stavov v systéme (5). Taktiež rozoznávame nasledujúce typy alarmov:

- **True Positive** - legitímny útok, ktorý spustí alarm.
- **False Positive** - udalosť signalizujúca vyvolanie alarmu, keď sa neuskutočnil žiadny útok

- **False Negative** - situácia, kedy nie je vyvolaný žiadny alarm, hoci k útoku skutočne došlo
- **True Negative** - udalosť, kedy sa neuskutočnil žiadny útok a nebola vykonaná žiadna detekcia (15).

V niektorých prípadoch je nevyhnutné identifikovať situáciu, v ktorej viaceré udalosti sú spôsobené rovnakým chybovým stavom, a to prostredníctvom procesu korelácie udalostí. Korelácia udalostí prebieha cez štyri fázy:

- **Filtrovanie** – vyčleňovanie irelevantných udalostí s cieľom redukcie informačného hluku
- **Agregácia** – odstránenie duplicitných výskytov rovnakej udalosti, čo zabezpečuje lepšiu prehľadnosť udalostí
- **Maskovanie** – skrytie udalosti, ktorá je následkom chyby a nie je relevantná pre skutočný chybový stav
- **Analýza hlavných príčin RCA** – metodika, ktorá využíva vzájomné závislosti medzi udalosťami na vytvorenie modelu prostredia. Tento model umožňuje objasniť úplné vysvetlenie chyby a identifikovať hlavné príčiny, ktoré k nej viedli (5).

Správa konfigurácie

Správa konfigurácie zahŕňa úlohy súvisiace so správou konfigurácie a identít systémov a užívateľov v sieti. Medzi úlohy, ktoré spadajú pod správu konfigurácie, patria:

- **Nastavenie koncových zariadení a aktívnych prvkov**: Definovanie a aktualizácia nastavení koncových zariadení a aktívnych prvkov v sieti
- **Inštalácia a konfigurácia softvéru (správa konfigurácie softvéru – SCM)**: Riadenie inštalácie a nastavenia softvéru vrátane procesov správy konfigurácie softvéru (SCM)
- **Správa rôznych užívateľov a skupín v sieti vrátane ich práv**: Riadenie identít užívateľov a skupín v sieti, vrátane pridelenia a správy oprávnení
- **Vyžadované aktualizácie a opravy softvéru a systémov**: Monitorovanie a implementácia nevyhnutných aktualizácií a opráv pre softvér a systémy

- **Provisioning dedikovaných sieťových pripojení**: Vytváranie a správa dedikovaných sieťových pripojení podľa potrieb siete
- **Dokumentácia konfigurácie všetkých sieťových súčastí**: Zaznamenávanie a udržiavanie dokumentácie týkajúcej sa konfigurácie všetkých komponentov v sieti (5).

Účtovná a evidenčná správa

Účtovná a evidenčná správa v rámci nástrojov pre správu siete sa zaoberá meraním využívaných dát s cieľom fakturácie zákazníkom alebo jednotlivým oddeleniam. Funkcie účtovania využívajú informácie o trendoch, ktoré sú poskytované nástrojmi na monitorovanie výkonu a umožňujú identifikovať, ktorí užívatelia alebo skupiny sú zodpovední za konkrétnu aktivitu a aké oprávnenia k nej majú pridelené. Niektoré zo sieťových funkcií, ktoré sa zhromažďujú na účely fakturácie, zahŕňajú:

- **Objem dát prenesených cez určité spojenie**: Miera využitia sieťového prenosu vzhľadom na konkrétne pripojenie
- **Počet konkrétnych udalostí súvisiacich s určitou aktivitou**: Sledovanie špecifických udalostí v sieti, ktoré majú dopad na fakturáciu
- **Počet spotrebovaných konkrétnych sieťových prostriedkov**: Množstvo využívaných sieťových zdrojov v súvislosti s konkrétnymi aktivitami
- **Hodnoty špičkového využitia**: Identifikácia maximálnych hodnôt využitia siete v určitom časovom období. (5).

Správa výkonu

Cieľom správy výkonu je zaistiť chovanie siete za štandardných podmienok a ponúknuť prostriedky potrebné k optimalizácii výkonu. Medzi najdôležitejšie prvky sledovania výkonu patria:

- Prevádzka siete funkcie použitého protokolu
- Miera kolízii
- Miera chýb rámcov
- Prevádzka siete funkcie uzlu (5).

Typickými nástrojmi pre sledovanie výkonu siete sú sledovacie programy – sniffery. Paketový sniffer zachytáva dáta prúdiace po sieti, takže sa dajú čítať a analyzovať ich obsah. Paketové sniffery sa dajú nakonfigurovať tak, aby zachytávali prevádzku v určitom segmente, na porte aktívneho prvku alebo na koncovom zariadení. Taktiež sa používajú na:

- Analýzu chýb siete
- Detekciu porušenia zabezpečenia
- Zhromažďovanie štatistických údajov o využití siete
- Zisťovanie použitých protokolov a filtrovanie paketov založených na pravidlách
- Zachytávanie relácii (5).

Správa bezpečnosti

Správa bezpečnosti v rámci nástrojov pre správu siete poskytuje prostriedky umožňujúce užívateľom a skupinám povoľovať alebo odopierať prístup k sieťovým zdrojom. Zabezpečenie siete spočíva na dvoch kľúčových funkciách: autentizácii užívateľov a systémov a ochrane dát posielaných po sieti prostredníctvom metód, ako je napríklad šifrovanie. Softvér pre správu zabezpečenia môže vytvárať infraštruktúru založenú na kľúčoch, ktoré zohrávajú úlohu v procese šifrovania a dešifrovania (5).

2.5.2 Event management

Medzi hlavné ciele event management patrí zodpovednosť za správu udalostí a zaručenie schopnosti detekovať udalosti, činiť ich zrozumiteľnými a na základe toho vykonať alebo iniciovať vhodnú akciu. Taktiež pomáha skrátiť trvanie výpadkov IT služieb vďaka ich včasnej identifikácii, pred tým ako výpadok pocíti užívateľ alebo k ich úplnému predídeniu a tým dokáže zvýšiť spokojnosť koncových užívateľov (4).

V prípade jeho absencie sa používajú užívatelia IT služieb ako nástroje pre identifikáciu výpadkov týchto služieb a vzniká vysoká reaktivita s nulovou proaktivitou pri riešení výpadkov (4).

Udalosť

Je zmena stavu, ktorá je významná z hľadiska riadenia konfiguračnej položky alebo služby IT. Taktiež sa tento pojem používa vo význame výstrahy alebo upozornenia pochádzajúcich od služby, konfiguračnej položky alebo monitorovacieho nástroja (4).

Úlohou monitorovacieho procesu je udalosti detekovať a roztriediť ich do nasledujúcich troch kategórii:

- **Informational** – udalosť, ktoré stačí zaznamenať pre účely ďalších analýz
- **Warning** – udalosti signalizujúce dosiahnutie konkrétnej prednastavenej prahovej hodnoty, čo umožňuje predchádzať vzniku incidentov (napr. zaťaženie RAM, CPU,...)
- **Exception** – udalosť, ktorá signalizuje neštandardný stav, ktorý často vyžaduje bezprostrednú akciu

2.5.3 Incident management

Incident management sa zaoberá včasnou detekciou incidentov, ich evidenciou a riadením životného cyklu. Jeho úlohou nie je skúmať príčiny incidentov (tú zodpovedá problém management), ale hľadať riešenia vrátane záložných (workaround), ktoré povedú k obnoveniu služby alebo k eliminácii, či aspoň obmedzeniu dôsledkov výpadku služby. Jeho vplyv spočíva predovšetkým v skrátení doby trvania incidentov a minimalizácii obchodných strát vyplývajúcich z ich existencie. Priamo neovplyvňuje počet incidentov, ale zameriava sa na efektívne riešenie a riadenie ich následkov. Incidenty sú detekované event managementom a užívateľmi kontaktujúci service desk, pričom je dôležité, aby najväčší podiel incidentov bol detekovaný prostredníctvom dohľadových nástrojov (4).

Hlavným cieľom incident managementu je čo najrýchlejšie obnovenie prevádzky služby súčasne s minimalizáciou dôsledkov výpadku pre užívateľov. Taktiež sa zameriava na zabezpečenie toho, aby poskytované služby zákazníkom spĺňali kvalitatívne požiadavky stanovené v dohodnutých úrovniach služieb – SLA (4).

Nedostatok incident managementu má viaceré negatívne dôsledky, vrátane:

- **Strata neriadených incidentov** – pri neefektívnom riadení incidentov dochádza k strate kontroly a predlžuje sa doba ich odstránenia, čím sa taktiež zvyšuje doba výpadku služby
- **Chýbajúce eskalačné procedúry** – absencia jasných postupov eskalácie môže spôsobiť, že drobné incidenty sa môžu premeniť na závažné, keďže nedochádza k dostatočnému rýchlemu riešeniu problémov
- **Zvýšená záťaž na zákaznícku podporu** – bez riadenia incidentov je zákaznícka podpora vystavená vyššej záťaži a tlaku na rýchle riešenie problémov
- **Nešpecifikované riešenia incidentov** – v prípade absentujúcich presných postupov incidenty môžu byť riešené odhadom namiesto presného postupu, čo môže viesť k neefektívnym a nedostatočným riešeniam (4).

Incident

Incident je neplánované prerušenie IT služby alebo zníženie jej kvality. Termín incident sa tiež vzťahuje na poruchu konfiguračnej položky, ktorá zatiaľ neovplyvnila poskytovanú službu (4).

Náhradné riešenie

Náhradné riešenie, tiež označované ako *workaround*, predstavuje dočasné obmedzenie alebo elimináciu dopadu incidentu alebo problému, pre ktorý v súčasnosti nie je k dispozícii úplné riešenie (4).

2.6 Podnikový informačný systém

Podnikový informačný systém má za účel splniť základné požiadavky podniku na adekvátnu podporu podnikových procesov prostredníctvom informačných a komunikačných technológií. Tento systém zahŕňa tri hlavné prvky: ľudí, informačné a komunikačné technológie (ICT) a dáta. Ľudia predstavujú významný prvok v informačnom systéme. Rozdeľujú sa na užívateľov informácii a personál zaoberajúci sa ICT. Užívatelia informácii sú pracovníci, ktorí priamo pracujú s informačným systémom a využívajú jeho výsledky. Títo pracovníci môžu pochádzať z rôznych oblastí podniku, ako sú personálny, ľudské zdroje (HR) a ďalšie oddelenia. Okrem toho zahŕňa manažérov z riadiacej štruktúry podniku a administratívnych pracovníkov, ktorí zbierajú dáta,

aktualizujú databázy a vytvárajú rôzne správy. ICT zahrňujú hardvérové a softvérové komponenty, ktoré sú súčasťou podnikového informačného systému. Tieto technológie umožňujú zbieranie, spracovanie a distribúciu informácií v rámci organizácie. Dáta predstavujú informácie, ktoré sú spracované a využívané v podnikovom informačnom systéme. Zahrňujú údaje o zákazníkoch, transakciách, zamestnancoch a ďalšie relevantné informácie potrebné pre beh podnikových procesov.

Rola užívateľa predpokladá:

- Samotnú obsluhu aplikačných softwarov a interpretáciu získaných výsledkov
- Sledovanie a analýzu stavu prevádzkových aplikácii
- Určenie väzieb danej aplikácie k ostatným aplikáciám (2)

Dáta predstavujú zaznamenané fakty o všetkých podstatných skutočnostiach, ktoré súvisia s aktivitami podniku. Dajú sa rozdeliť do troch kľúčových skupín:

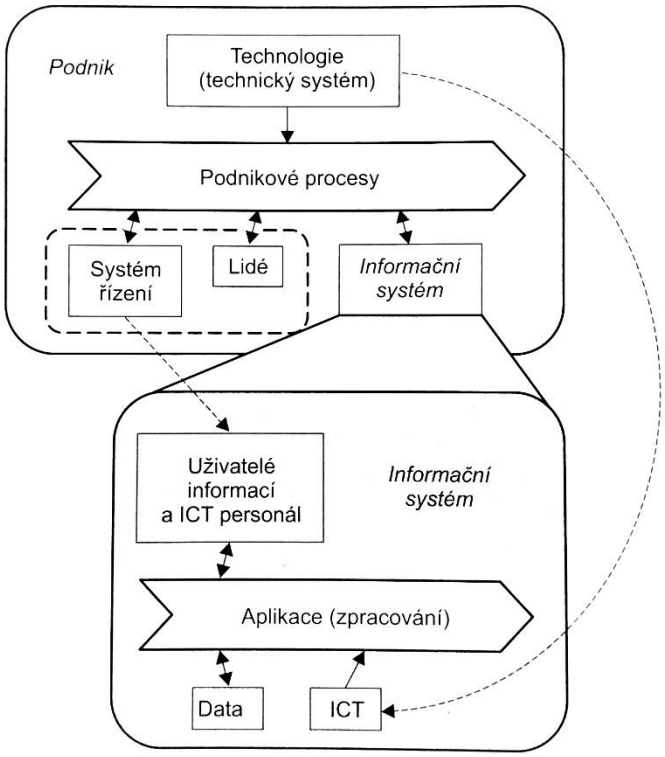
- **Dáta o spoločenských podmienkach podnikania** – do tejto skupiny zahrňujeme zaznamenané údaje o politických a štátnych očakávaniach v oblasti stability prostredia, rozvoji technológii. Taktiež sa tu zahrňujú údaje o faktoroch ovplyvňujúce výrobu napr. pracovná sila, materiál
- **Dáta o trhu** – zaraďujeme tu dáta o dopyte po komoditách podniku (produkty a služby)
- **Interné dáta** – tvoria predpoklady k tomu, aby mohol podnik reagovať vo svojom okolí. Do tejto skupiny patria plány a predpovede predaje alebo údaje formulujúce požiadavky ba alokáciu podnikových zdrojov (2).

ICT predstavujú širokú škálu technických prostriedkov a programového vybavenia. Technické prostriedky zahrňujú koncové zariadenia, samotné nosiče dát, komunikačné prostriedky (telekomunikačné a počítačové siete) a ďalšie špecializované zariadenia (2).

Kľúčovými vlastnosťami prvkov je úroveň schopnosti prvkov pružne a efektívne reagovať na požiadavky okolia systémov a vyjadrujú sa:

- **Schopnosťou byť integrovaný do väčšieho celku** tzn. vyjadrujeme možnosťou byť prepojený s iným prvkom a úrovňou kompatibility integrovaných prvkov

- **Úrovňou modularity** – stupňom spôsobilosti k zmene a k rozšíreniu bez závažných dopadov na celok
- **Úrovňou znalostí, skúsenosti a schopnosťami**, pričom sa očakáva, že disponujú technickými znalosťami a skúsenosťami, majú k dispozícii porozumenie, kde a jak nasadiť ICT tak, aby efektívne podporili ciele organizácie (2)



Obrázok 14: Podnikový informačný systém a jeho vzťah k podniku (Zdroj: (2))

2.7 XML jazyk

XML je značkovacím jazykom, označovaným za generický. Tento charakter znamená, že poskytuje inštrukcie, ktoré popisujú výlučne obsah dokumentu. XML sa dôkladne zameriava na štruktúru obsahu, umožňujúc vzdialeným systémom vymieňať a interpretovať takéto dokumenty bez potreby ľudského zásahu. Pre XML boli definované nasledujúce princípy:

- Formát XML musí byť použiteľný v rámci internetu (použiteľný kýmkoľvek)
- Formát XML by mal podporovať širokú škálu aplikácii
- Musí byť jednoduché vytvárať programy, ktoré manipulujú s dokumentmi v XML

- XML dokumenty by mali byť čitateľné a pochopiteľné aj pre človeka (2)

XML dokument sa z logického pohľadu skladá z prológu, deklarácii, elementov, komentárov a inštrukcii pre spracovanie inými aplikáciami.

XML je závislý na štruktúre, obsahu a integrite. Aby bol dokument považovaný za správne štruktúrovaný, musí spĺňať nasledujúce vlastnosti:

- Musí mať práve jeden root element.
- Neprázdne elementy musia byť ohraničené začiatočným a ukončovacím tagom. Prázdne elementy môžu byť označené tagom „prázdny element“.
- Všetky hodnoty atribútov musia byť uzavreté v úvodzovkách (') alebo (")
- Elementy môžu byť vnorené, ale nemôžu sa prekrývať (2)

```
1.  <?xml version="1.0"encoding ="UTF-8"?>
2.  <zbozi>
3.     <vyrobek id-zbozi="1">
4.        <nazev>Volta 75</nazev>
5.        <vykon>75</vykon>
6.        <zavit>E27</zavit>
7.        <jednotkovacena>10</jednotkovacena>
8.     </vyrobek>
9.     <vyrobek id-zbozi="2">
10.       <nazev>Volta 100</nazev>
11.       <vykon>100</vykon>
12.       <zavit>E27</zavit>
13.       <jednotkovacena>7</jednotkovacena>
14.    </vyrobek>
15. </zbozi>
```

Obrázok 15: Základná štruktúra XML (Zdroj: (2))

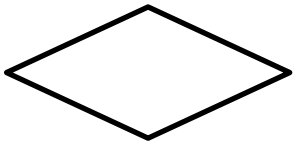
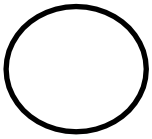
2.8 Funkčné modelovanie

Proces, ktorý zahrňuje súbor vzájomne pôsobiacich činností, ktoré transformujú vstupy na výstupy. V rámci informačného systému je možné procesy popisovať slovným popisom, grafickým znázornením alebo rozhodovacími tabuľkami.

2.8.1 Vývojový diagram

Vývojový diagram slúži ku grafickému zobrazeniu algoritmu programu krok za krokom. Pre znázornenie sa používajú grafické symboly spojené orientovanými šípkami. (14)

Tabuľka 1: Grafické symboly vývojového diagramu (Zdroj: Vlastné spracovanie)

Grafický symbol	Názov	Popis
	Terminál	Prvý a posledný krok v algoritme.
	Proces	Zobrazenie procesu v algoritme.
	Vstup / výstup dát	Zobrazenie vstupných alebo výstupných dát v algoritme.
	Orientovaná šípka	Grafické zobrazenie smeru algoritmu.
	Rozhodovací blok	Bod, kde výsledok rozhodnutia určuje ďalší krok.
	Cyklus	Zobrazenie cyklu s určitým počtom opakovaní.
	Podproces	Zobrazenie podprocesuv algoritme.
	Spojka	Zobrazenie ďalšieho kroku na inom výkrese

3 VLASTNÉ NÁVRHY RIEŠENIA

V tejto kapitole sa zameriam na implementáciu monitorovacieho systému pre univerzitu. V rámci tohto rozboru budem podrobne popisovať proces výberu monitorovacieho systému, analyzovať riziká spojené s aktívnymi prvkami, a následne sa venovať konfigurácii monitorovacieho systému. Táto konfigurácia zahŕňa získavanie údajov z NetBoxu a XML súboru, nastavovanie sond a príslušných upozornení a nakoniec nastavenie práv pre užívateľov.

3.1 Výber monitorovacieho systému

V tejto časti sa budem venovať popisu troch populárnejších open-source monitorovacích informačných systémov, ktoré sa často využívajú na sledovanie aktívnych prvkov v sieti. Na záver tejto podkapitoly vyberiem jeden z týchto systémov.

Univerzita požaduje nájdenie a implementáciu nového monitorovacieho informačného systému založeného na open-source distribúcii. Divízia informačných systémov bude zodpovedná za správu databázovej a serverovej časti tohto systému, zatiaľ čo monitorovanie a konfigurácia budú v kompetencii divízie IT infraštruktúry. Dôležitou súčasťou bude integrácia s dokumentačným systémom NetBox, ktorý bude slúžiť ako hlavný zdroj údajov. Nový monitorovací informačný systém bu mal byť konfigurovateľný tak, aby bol schopný sledovať kritické body a slabiny aktívnych prvkov sieťovej infraštruktúry.

Pri preskúmaní trhu s open-source monitorovacími systémami som identifikoval niekoľko významných možností, medzi ktorých patria Zabbix, Icinga a Prometheus. Tieto tri systémy predstavujú širokú škálu možností pre organizácie hľadajúce monitorovací systém založený na open-source technológiách.

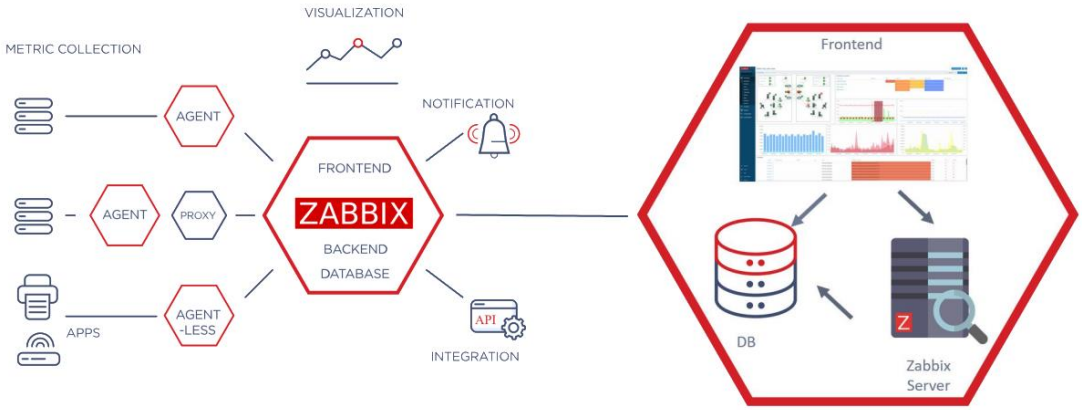
3.1.1 Zabbix

Zabbix rozdelený je do dvoch veľkých častí: servera a agentov. Server je umiestnený na jednom stroji, kde zbiera a ukladá štatistické údaje, a agenti sú umiestnení na tých strojoch, z ktorých sa údaje zbierajú.

Zabbix agenti podporujú pasívne (dotazovacie) a aktívne kontroly (trapovanie). Pasívne kontroly znamenajú, že server Zabbixu požiada o hodnotu od Zabbix agenta, a agent spracuje požiadavku a vráti hodnotu serveru Zabbixu. Aktívne kontroly znamenajú, že Zabbix agent požiada o zoznam aktívnych kontrol od serveru Zabbixu a potom pravidelne posiela výsledky. V prípadoch, kde nie je možné nainštalovať agenta, Zabbix ponúka základné monitorovanie bez agenta (napr. SNMP). S touto variantou je možné skontrolovať dostupnosť sieťových služieb a vykonávať vzdialené príkazy.

Na ukladanie údajov používa externú databázu. Databáza Zabbixu musí byť vytvorená počas jeho inštalácie. V súčasnosti sú podporované nasledujúce databázy: MySQL, PostgreSQL, Oracle, IBM DB2 a SQLite.

Na upozornenia využíva Zabbix už zabudovanú funkcionalitu pri ktorej je možné informovať zodpovedný personál o výskyte udalostí prostredníctvom mnohých rôznych kanálov. Systém upozorňovania umožňuje spravovať udalosti rôznymi spôsobmi: odosielanie správ, vykonávanie vzdialených príkazov, eskalácia problémov podľa flexibilne definovaných úrovní služby a podobne. Je tiež možné prispôsobiť správy na základe role prijímateľa tým, že sa vyberie, ktoré informácie zahrnúť, ako napríklad dátum, čas, názov hostiteľa, hodnota položiek, hodnoty spúšťačov, profil hostiteľa, história eskalácie a podobne (17).



Obrázok 16: Architektúra systému Zabbix (Zdroj: Vlastné spracovanie)

3.1.2 Icinga

Pôvodne vznikla ako odnož aplikácie na monitorovanie systému Nagios v roku 2009. Každý uzol v Icinge má jednu z troch rolí: master, satelit alebo agent. Hlavný stroj, ktorý bude zhromažďovať metriky, je registrovaný ako master. Stroje, ktoré sú monitorované - takzvané koncové body sú registrované ako agenti. Agenti Icinga vykonávajú monitorovacie skripty, ktoré vracajú stav vykonania skriptu, ako aj voliteľné údaje o výkonnosti späť do nadradenej jednotky.

Icinga databáza sa skladá z nasledujúcich komponentov:

- **icingadb**, ktorý je zodpovedný za odosielanie monitorovacích údajov na Redis server
- **Icinga DB démon**, ktorý synchronizuje údaje medzi Redis serverom a databázou
- **Icinga Web** s aktivovaným modulom Icinga DB Web, ktorý sa pripája k Redisu a databáze na zobrazenie a prácu s najnovšími údajmi

Obrázok 17: Komponenty systému Icinga (Zdroj: (18))

Samotná Icinga poskytuje možnosť zobrazenia jednoduchých grafov priamo v jej webovom rozhraní, ale pre pokročilejšie a interaktívne vizualizácie sa často využíva integrácia s nástrojmi na vizualizáciu dát, ako je Grafana.

Icinga má integrovaný systém notifikácií, ktorý upozorní v prípade, že sledovaný parameter prekročí definovanú hranicu alebo sa objaví iný problém. Notifikácie môžu byť odosielané prostredníctvom e-mailu, SMS, Slacku, a iných komunikačných kanálov.

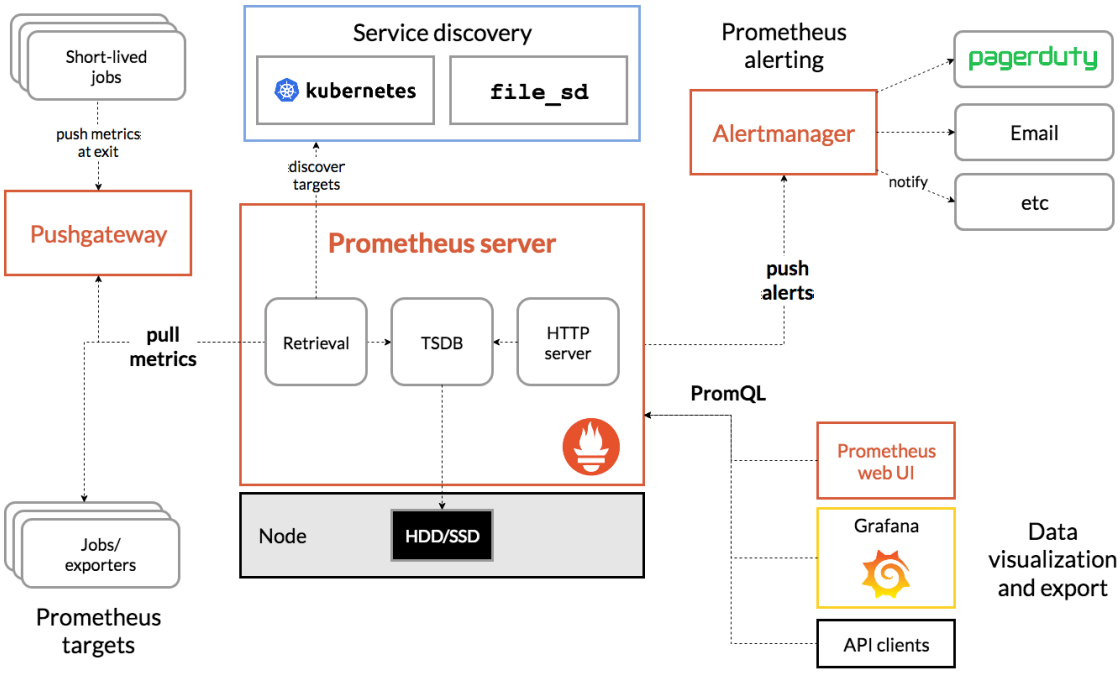
3.1.3 Prometheus

Prometheus je open-source monitorovací systém, ktorý poskytuje výkonný dotazovací jazyk, úložisko údajov a nástroje na vizualizáciu. Sleduje a zhromažďuje metriky v reálnom čase, ktoré zaznamenáva do databázy časových radov (TSDB). Tento nástroj ponúka multidimenzionálny dátový model, ktorý umožňuje definovať metriky pomocou názvu a/alebo značiek na ich identifikáciu ako súčasť jedinečného časového radu. Prometheus je implementovaný v programovacom jazyku Go a má licenciu Apache 2 so zdrojovým kódom dostupným na platforme GitHub.

Prometheus je vyvinutý tak, aby pravidelne získaval metriky z cieľového systému. Metriky je možné zbierať aj pomocou mechanizmu push. To môže byť potrebné v situáciách, keď zber metrík pomocou pull nie je možný. Napríklad keď sú monitorovacie služby chránené firewallom alebo sú monitorované služby pripojené k sieti periodicky a na krátky čas. Na tento účel sa používa špeciálny komponent pushgateway, ktorý sa inštaluje samostatne.

Aby mohol používateľ svoje grafy pozorovať a analyzovať, musí si nainštalovať vizualizačný nástroj Grafana, ktorý poskytuje veľké množstvo grafov a dashboardov a k tomu dostatočne veľkú customizáciu.

Pre spravovanie upozornení sa musí nainštalovať aplikácia Alertmanager. Upozornenia v Prometheus sú rozdelené na dve časti. Najprv sa definujú pravidlá upozorňovania na strane serveru Prometheus, ktoré budú následne odosielať upozornenia do nástroja Alertmanager (17).



Obrázok 18: Architektúra systému Prometheus (Zdroj: (17))

3.1.4 Zhrnutie

Pri rozhodovaní o výbere monitorovacieho systému je nevyhnutné zvážiť niekoľko kritérií a aplikovať multikriteriálne rozhodovanie s priradením váh k jednotlivým kritériám. Nasledujúce kritériá a ich váhy budú zohľadnené nasledovne:

- **Výkonnosť** (*váha **4***) – Schopnosť systému monitorovať a spracovávať maximálne množstvo sond.
- **Užívateľské rozhranie** (UI) (*váha **1***) – Prehľadnosť a použiteľnosť grafického prostredia pre jednoduché používanie systému.
- **Dokumentácia** (*váha **4***) – Dostupnosť a kvalita dokumentácie a návodov na použitie, konfiguráciu a údržbu systému.
- **Jednoduchosť** (*váha **2***) – Miera komplexnosti systému a počet komponentov, ktoré systém obsahuje.
- **Vizualizácia dát** (*váha **2***) – Schopnosť systému vizualizovať nazbierané údaje o stave siete.
- **Zbieranie dát** (*váha **3***) – Efektívnosť a množstvo spôsobov, ako systém dokáže zbierať údaje o stave siete a pracovať s nimi (SNMP, skripty atď.).

Výkonnosť

Zabbix aj Prometheus patria medzi výkonnejšie systémy, avšak Icinga zaostáva z dôvodu, že všetky sondy pre sledovanie koncových zariadení fungujú na princípe dotazovania pomocou monitorovacích skriptov písaných v jazyku Perl, ktoré vracajú stav vykonania skriptu. Pri vyššom počte sledovaných údajov napr. sledovanie všetkých portov 48 portového zariadenia, môže dochádzať k značnému klesnutiu výkonu dotazovacieho servera. Prometheus má decentralizovanú architektúru, kde každá inštancia je schopná ukladať svoje vlastné údaje. V niektorých prípadoch by to mohlo viesť k neefektívnemu využitiu zdrojov naopak, Zabbix má centralizovanú architektúru.

Zabbix aj Prometheus patria medzi výkonnejšie systémy, avšak Icinga vykazuje určité nedostatky z hľadiska výkonu, pričom všetky sondy pre sledovanie koncových zariadení pracujú na princípe dotazovania prostredníctvom monitorovacích skriptov napísaných v jazyku Perl, ktoré vracajú stav vykonania, a v prípade vyššieho počtu sledovaných údajov, ako je napríklad monitorovanie všetkých portov 48-portového zariadenia, môže nastať výrazné zníženie výkonu dotazovacieho servera. Prometheus sa vyznačuje decentralizovanou architektúrou, kde každá inštancia je schopná ukladať svoje vlastné údaje. V niektorých situáciách by to mohlo viesť k neefektívnemu využitiu zdrojov. Naopak, Zabbix disponuje centralizovanou architektúrou, čo môže byť v niektorých prípadoch považované za výhodu.

UI

Užívateľské rozhranie všetkých systémov je relatívne prehľadné a poskytuje značné množstvo informácií priamo na hlavnej obrazovke. Zabbix v navigačnom paneli ponúka bohaté množstvo nastavení a nástrojov, ktoré sú logicky usporiadané pre bežného užívateľa, avšak pre administrátora s veľkým množstvom položiek to môže znamenať, že konkrétnu vec nemusí nájsť okamžite. Prometheus sa vyznačuje jednoduchým dizajnom, ale v niektorých prípadoch môže pôsobiť dojmom obmedzeného množstva položiek v menu. Icinga má grafické rozhranie podobné Zabbixu, avšak občas sa vyskytujú grafické chyby, ktoré vedú k tomu, že niektoré položky nie sú pre užívateľa viditeľné.

Dokumentácia

Najpodrobnejšiu dokumentáciu zo všetkých poskytuje Zabbix, ktorý detailne popisuje všetky funkcionality vrátane bežných aj menej obvyklých použití, ktoré je možné v systéme nastaviť. Okrem toho ponúka užívateľom prístup k video tutoriálom. Prometheus má tiež dobre spracovanú dokumentáciu, avšak chýbajú jej niektoré príklady z reálneho použitia, ktoré by ocenili mnohí používatelia. Naopak, dokumentácia Icinga trpí nedostatkom podrobných a aktuálnych informácií. Navyše, komunitná podpora pre Icinga nie je tak rozsiahla a aktívna ako u oboch predchádzajúcich nástrojov.

Jednoduchosť

Prometheus zahŕňa viacero komponentov, ktoré sú všetky decentralizované. To znamená, že užívateľ alebo samotný systémový administrátor musí distribuovať svoje úlohy, ako je údržba, konfigurácia a monitorovanie, do viacerých oblastí. Krivka učenia pre Prometheus a Zabbix je výraznejšie plytšia v porovnaní s Icingou, ktorá je jednoduchšia pokiaľ ide o nastavenie sond a notifikácií.

Vizualizácia dát

Zabbix umožňuje užívateľom vytvárať vlastné grafy pridaním filtrov a obmedzení, agregovať viacero položiek do jedného grafu a taktiež vytvárať sieťové mapy, ktoré vizualizujú stav celej infraštruktúry. Na druhej strane, Prometheus a Icinga majú schopnosť využívať nástroj Grafana, čo im poskytuje rozsiahlejšiu možnosť prispôsobenia grafov a dashboardov v porovnaní s Zabbixom. Je dôležité však poznamenať, že Grafana je externým komponentom, a bez nej Icinga neponúka žiadne vizuálne zobrazovanie. V prípade Prometheusa je obmedzený na zobrazovanie iba jedného grafu naraz.

Zbieranie dát

Icinga ponúka natívne monitorovanie zariadení cez SNMP, ale pre monitorovanie komplexnejších metrík vyžaduje Icinga rôzne Nagios pluginy. Napríklad na získanie údajov v rámci agenta, ako je zaťaženie procesora, využitie pamäte a využitie disku, potrebuje mať Icinga nainštalovaný *NRPE* (Nagios Remote Plugin Executor) plugin, ktorý je zastaralý 4 roky a nie je moc udržovaný. Icinga nemá možnosť modifikovať dáta pri ich získaní. Zabbix používa na získavanie informácií *item key* položky. Napríklad

položka s názvom kľúča system.cpu.load zhromaždí údaje o zaťažení procesora. Výsledok sa dá prispôsobiť zadaním ďalších parametrov. Napríklad system.cpu.load[avg5] vráti priemerné zaťaženie CPU za posledných 5 minút. Taktiež všetky nazbierané dáta môžu prejsť úpravou pred svojím uložením do databázy pomocou tzv. Preprocessingu. Táto procedúra umožňuje dodatočné formátovanie získaných hodnôt, vrátane použitia regulárnych výrazov, validačných funkcií alebo JavaScriptových úprav. Prometheus využíva PromQL - výkonný dopytovací jazyk vyvinutý Prometheus tímom, na získavanie a agregáciu údajov časových radov z viacrozmerného dátového modelu v reálnom čase. však, rovnako ako Icinga, na zber rôznych typov metrík je potrebné nainštalovať množstvo modulov, ktoré nie sú natívne súčasťou systému, napríklad pre zbieranie údajov pomocou SNMP je nutné nainštalovať samostatný modul *SNMP_exporter*. Prometheus taktiež nemá možnosť modifikovať dáta pri ich získaní.

Výber systému

Tabuľka 2: Prehľad systémov (Zdroj: Vlastné spracovanie):

		Kritéria						Vážený priemer
		Výkonnosť	UI	Dokumentácia	Jednoduchosť	Vizualizácia dát	Zbieranie dát	
Monitorovací systém	**Zabbix**	1	2	1	3	2	1	**1.4375**
	Icinga	3	2	3	2	2	3	2.6875
	Prometheus	2	2	2	3	2	2	2.125

Icinga predstavuje monitorovací systém, ktorý sa spolieha na staršie metódy monitorovania. Vzhľadom na konkrétne požiadavky mojej implementácie, nejde o vhodnú voľbu. Naopak, Prometheus je navrhnutý s ohľadom na dynamickú a kontajnerovú architektúru mikroslužieb. Jeho decentralizovaná architektúra však môže spôsobiť komplexitu a vytvárať priestor pre potenciálne chyby. Taktiež vynucuje minimalistickejší prístup, čo znamená, že užívateľ musí vytvárať a implementovať ďalšie funkcie, ktoré by mohol považovať za nevyhnutné, nakoľko im chýbajú v základnom balíku. V porovnaní s Prometheusom má Zabbix omnoho rozsiahlejšiu sadu funkcií. Zabbix ponúka širokú škálu možností v základnej výbave vrátane podpory pre upozornenia, vizualizácie a automatizácie. Táto komplexná funkcionalita robí **Zabbix** vhodným výberom pre moju konkrétnu implementáciu monitorovacieho systému.

3.2 Analýza rizík aktívnych prvkov

V tejto časti sa budem venovať identifikácii hrozieb, ktoré môžu ovplyvniť aktívne prvky v sieťovej infraštruktúre. Začnem vytvorením hodnotiacej tabuľky, kde presne zaznačím hodnotu a úroveň dopadu každého rizika. Na základe týchto identifikovaných rizík sa následne vytvorí relevantný zoznam sond v monitorovacom systéme Zabbix, ktoré budú sledovať a monitorovať tieto potenciálne hrozby.

3.2.1 Identifikácia hrozieb a úrovní rizík

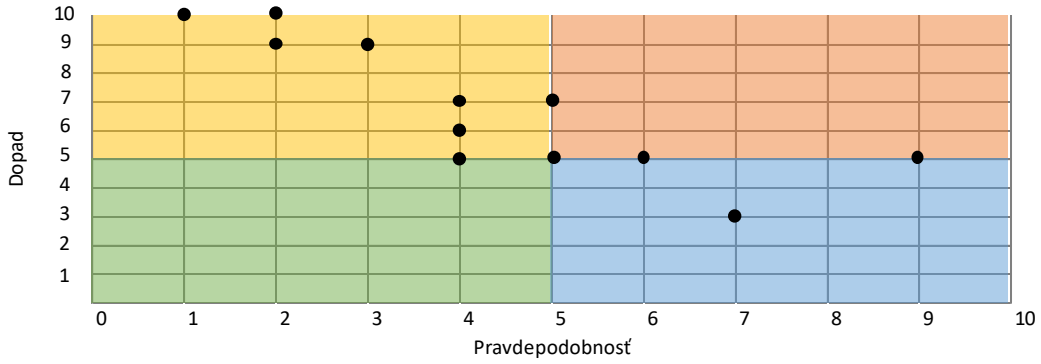
Na základe klasifikačnej tabuľky budem určovať pre každú identifikovanú hrozbu jej pravdepodobnosť výskytu a dopad na univerzitu.

Tabuľka 3: Klasifikácia pravdepodobností a dopadov incidentov (Zdroj: Vlastné spracovanie)

Pravdepodobnosť			Dopad	
Hodnota	Percentuálne	Slovné hodnotenie	Hodnota	Slovné hodnotenie
1 - 2	0% - 19%	Veľmi nepravdepodobné	1 - 2	Bezvýznamný
3 - 4	20% - 39%	Nepravdepodobné	3 - 4	Málo významný
5 - 6	40% - 59%	Pravdepodobné	5 - 6	Významný
7 - 8	60% - 79%	Viac pravdepodobné	7 - 8	Veľmi významný
9 - 10	80% - 100%	Veľmi pravdepodobné	9 - 10	Kritický

Tabuľka 4: Identifikácia hrozieb (Zdroj: Vlastné spracovanie)

Číslo	Hrozba	Scenár	Pravdepodobnosť	Dopad	Úroveň rizika
1	Výpadok prístupového zariadenia	Strata pripojenia pre koncových zákazníkov v určitých kanceláriach	5	7	35
2	Výpadok distribučného zariadenia (v prípade straty redundancie)	Strata vzdialeného pripojenia pre koncových zákazníkov pre určitú lokalitu	3	9	27
3	Výpadok chrbticového zariadenia (v prípade straty redundancie)	Strata vzdialeného pripojenia pre koncových zákazníkov pre celú univerzitu	2	10	20
4	Výpadok DC firewallu	Strata pripojenia pre dátové centrum	2	10	20
5	Výpadok člena clusteru firewallu	Strata redundancie a loadbalancingu	4	5	20
6	Výpadok up-linkového spojenia k ISP (v prípade straty redundancie)	Strata vzdialeného pripojenia pre koncových zákazníkov pre celú univerzitu	1	10	10
7	Výpadok up-linkového spojenia k chrbtici (v prípade straty redundancie)	Strata vzdialeného pripojenia pre koncových zákazníkov pre určitú lokalitu	3	9	27
8	Výpadok up-linkového spojenia k distrubúcii (v prípade straty redundancie)	Strata vzdialeného pripojenia pre koncových zákazníkov pre určité kancelárie	4	7	28
9	Neuložená konfigurácia	Vznik hrozby straty neuložených zmien při reštarte zariadenia	9	5	45
10	Vyťažené CPU na viac ako 90 %	Znížená funkcionalita managementu a control planu zariadenia	4	5	20
11	Vyťažená memory na viac ako 90 %	Znížená funkcionalita zariadenia pri ukladaní dát	5	5	25
12	Chybujúca linka k užívateľovi	Nekvalitné pripojenie	6	5	30
13	Výpadok zdroja zariadenia	Výpadok napájania pre zariadenie	4	6	24
14	Výpadok ventilátora zariadenia	Vznik hrozby prehriatia zariadenia	4	6	24
15	Výpadok access pointu	Výpadok pripojenia pre užívateľov v učebni	7	3	21
16	Výpadok wireless controllera (v prípade straty redundancie)	Výpadok bezdrôtového pripojenia pre celú univerzitu	2	9	18



Obrázok 19: Mapa rizík pred opatreniami (Zdroj: Vlastné spracovanie)

Z uvedených hrozieb vyplýva, že najväčší dopad na univerzitu by mohli mať výpadky zariadenia chrbticovej a distribučnej vrstvy zariadenia, výpadok firewallu v dátovom centre, výpadok uplinkového spojenia k internetovému poskytovateľovi alebo výpadok uplinkového spojenia ku chrbticovému zariadeniu. Tieto hrozby však nevykazujú tak vysokú hodnotu rizika, pretože pravdepodobnosť ich výskytu je relatívne nízka. Výpadky

týchto zariadení sú totiž zabezpečené dodatočnou redundanciou zariadení a spojení na rovnakej logické vrstve topológie.

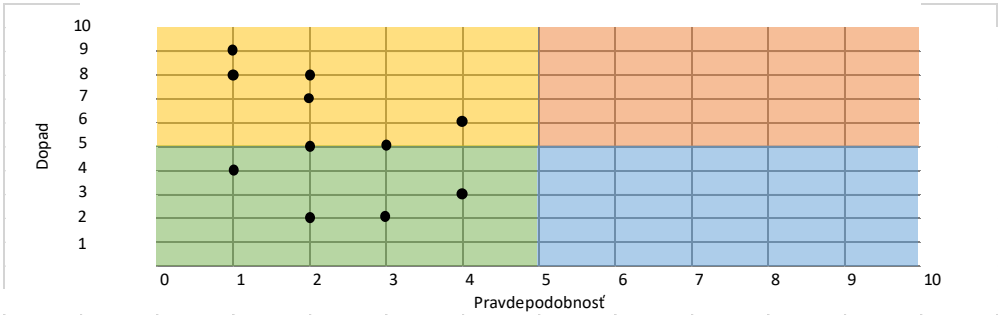
Najpravdepodobnejšími hrozbami, ktoré by mohli nastať, sú výpadky prístupového switchu, neuložená konfigurácia, chybujúca linka alebo výpadok access pointu. Tieto hrozby majú výrazne vyššiu hodnotu rizika, aj keď ovplyvňujú menšiu skupinu používateľov. Avšak títo používatelia sa často môžu zaradiť medzi študentov vo vyhradenom učebnom priestore s výučbou, ktorá prebieha v krátkych časových obdobiach.

3.2.2 Návrhy opatrení

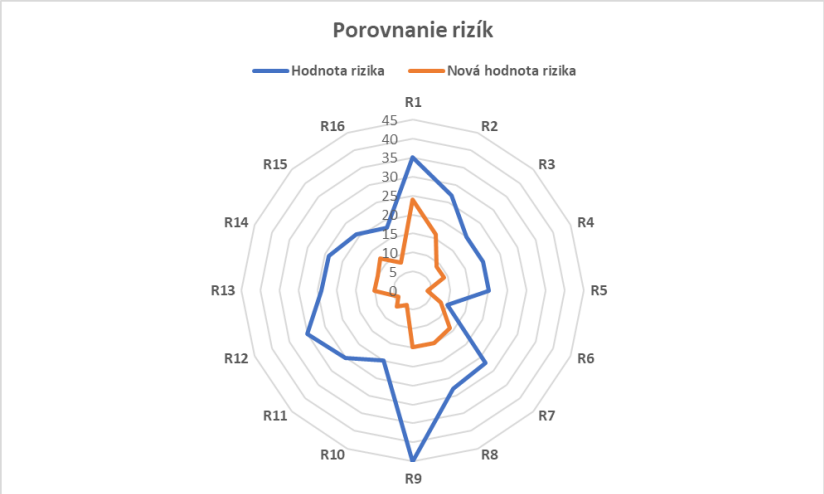
Cieľom tejto časti je poskytnúť základ pre implementáciu bezpečnostných opatrení, ktoré budú zabezpečovať spoľahlivú prevádzku siete a minimalizovať riziko výpadkov a bezpečnostných incidentov.

Tabuľka 5: Návrh opatrení (Zdroj: Vlastné spracovanie)

Číslo	Hrozba	Návrh opatrenia	Pravdepodobnosť	Dopad	Úroveň rizika
1	Výpadok prístupového zariadenia	Monitorovanie dostupnosti zariadenia, MAC tabuliek a jeho kritických súčastí	4	6	24
2	Výpadok distribučného zariadenia (v prípade straty redundancie)	Monitorovanie funkčnosti liniek vedúcich k zariadeniu, MAC tabuliek a kritických komponentov zariadenia	2	8	16
3	Výpadok chrbticového zariadenia (v prípade straty redundancie)	Monitorovanie funkčnosti liniek vedúcich k zariadeniu a kritických komponentov zariadenia	1	9	9
4	Výpadok DC firewallu	Zaistenie redundancie	1	9	9
5	Výpadok člena clusteru firewallu	Monitorovanie funkčnosti všetkých členov clusteru	1	4	4
6	Výpadok up-linkového spojenia k ISP (v prípade straty redundancie)	Monitorovanie vlastností uplinkových liniek k ISP	1	8	8
7	Výpadok up-linkového spojenia k chrbtici (v prípade straty redundancie)	Monitorovanie vlastností uplinkových liniek k chrbticovej vrstve	2	7	14
8	Výpadok up-linkového spojenia k distrubúcii (v prípade straty redundancie)	Monitorovanie vlastností uplinkových liniek k distribučnej vrstve	3	5	15
9	Neuložená konfigurácia	Monitorovanie stavu neuloženej konfigurácie	3	5	15
10	Vyťažené CPU na viac ako 90 %	Monitorovanie využitia CPU	2	2	4
11	Vyťažená memory na viac ako 90 %	Monitorovanie využitia memory	3	2	6
12	Chybujúca linka k užívateľovi	Monitorovanie vlastností liniek liniek až k užívateľom	2	2	4
13	Výpadok zdroja zariadenia	Monitorovanie funkčnosti zdrojov	2	5	10
14	Výpadok ventilátora zariadenia	Monitorovanie funkčnosti ventilátorov	2	5	10
15	Výpadok access pointu	Monitorovanie dostupnosti access pointu a jeho kritických súčastí	4	3	12
16	Výpadok wireless controllera (v prípade straty redundancie)	Monitorovanie dostupnosti wireless controlleru a jeho kritických súčastí	1	8	8



Obrázok 20: Mapa rizík po opatreniach (Zdroj: Vlastné spracovanie)



Obrázok 21: Porovnanie rizík pred a po opatreniach (Zdroj: Vlastné spracovanie)

Zo zistení vyplýva, že efektívnym opatrením na zmiernenie rizík je nasadenie monitorovania zariadení na úrovni, ktorá prekračuje jednoduchú kontrolu dostupnosti zariadenia (*hostalive*), ako bolo bežné v súčasnom monitorovacom systéme Nagios. Implementácia monitorovania týchto kritických oblastí siete umožní rýchlu detekciu a riešenie problémov, čím sa minimalizuje ich negatívny vplyv na univerzitnú infraštruktúru a používateľov.

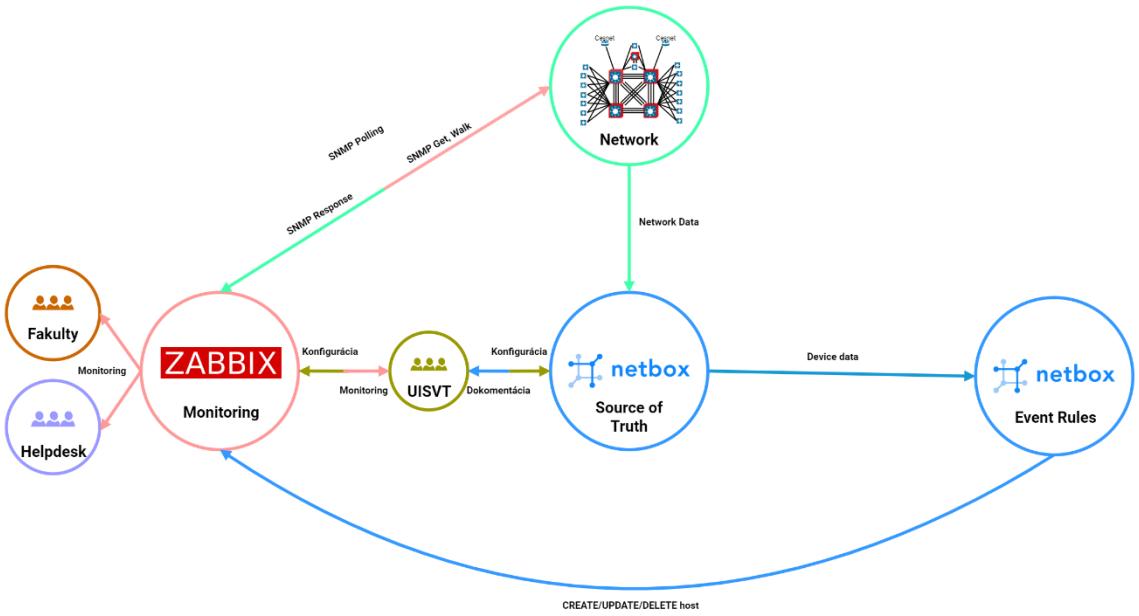
3.3 Nastavovanie monitorovacieho informačného systému

V tejto časti sa budem venovať nastavovaniu sledovacích funkcií vybraného monitorovacieho systému – Zabbix. V prvej časti bude popísaná časť získavania údajov z dokumentačného systému NetBox pre prvotnú konfiguráciu Zabbixu a jeho následnú aktualizáciu. Následne ukážem konfiguráciu sónd pre jednotlivé hrozby v Zabbixe spojené so sledovaním inventárnych atribútov aktívnych prvkov. Nakoniec bude predstavené riešenie pre helpdesk.

Nasledujúce nastavenia sa budú nastavovať v testovacom prostredí na virtualizačnej platforme Docker pre oba systémy, ktoré fungujú paralelne ako kontajnery. Zabbix bude v testovacom prostredí používať verziu **6.4.8** a NetBox verziu **3.7.1**.

V kapitole sa budem podrobne venovať konfigurácii sond a upozornení (triggerov) pre systém Zabbix, ktorý bude nasadený v architektúre typu Manager-Agent. Majorita dotazov sa budú vykonávať pomocou SNMP pollingu, čo znamená, že sa Zabbix (Manager) pravidelne dotazuje sledovaných zariadení (polls) v určitých časových intervaloch a zariadenia na tieto dotazy reagujú (response), čo umožňuje Zabbixu monitorovať ich stav a zbierať relevantné metriky. Na základe týchto údajov budú vytvárané triggerové udalosti, ktoré identifikujú anomálie a potenciálne problémy v sieti univerzity. Tieto triggerové udalosti slúžia ako hlavný mechanizmus na upozornenie správcov siete Ústavu informačných služieb a výpočtovej techniky a iných zainteresovaných strán (helpdesk a fakultné IT oddelenia) o možných problémoch alebo významných udalostiach.

Pre iniciačný import zariadení do Zabbixu bude použitý dokumentačný systém NetBox ako *„source of truth“*, ktorý reprezentuje skutočný stav univerzitnej siete. Po prvotnom načítaní údajov z NetBoxu do Zabbixu je nevyhnutné udržiavať tieto údaje v Zabbixu aktuálne, keďže sa očakáva častá zmena údajov v NetBoxe. Pre túto situáciu bude využitá funkcionalita NetBoxu nazvaná *Event Rules*.



Obrázok 22: Začlenenie Zabbixu (Zdroj: Vlastné spracovanie)

3.3.1 Iniciačný import zariadení do Zabbixu

Ústav informačných služieb a výpočtovej techniky v NetBoxe eviduje všetky sieťové aktívne prvky, ktoré sa v jej infraštruktúre používajú. Pre počiatočné naplnenie Zabbixu zariadeniami z NetBoxu je nutné najprv preddefinovať, ktoré atribúty budú z NetBoxu slúžiť pre nakonfigurovanie Zabbix hostov:

- Name – FQDN zariadenia
- Status – status zariadenia, ktoré môže nadobúdať hodnoty Offline, Active, Planned, Staged, Failed a Inventory. Monitorované zariadenia budú iba v statuse Active.
- Site – fakulta, v ktorej sa zariadenie momentálne nachádza

Pre monitorovanie zariadení vyžaduje Zabbix SNMP komunity, ktoré sa nachádzajú v XML súbore ***devices.xml*** na remote zariadení – jumphostovi.

Príkladový výsledný stav atribútov správne vloženého zariadenia do Zabbixu bude v nasledovnej štruktúre:

- **Host name** – zariadenie.mgm.univerzita.cz
- **Templates** – Template_1, Template_2, …
- **Host groups** – Právnická fakulta

- **Interfaces**
    - **Type** – SNMP
    - **DNS name** – zariadenie.mgm.univerzita.cz
    - **Connect to** – DNS
    - **Port** – 161
    - **SNMP version** – SNMPv2c
    - **SNMP community** – *{$SNMP_COMMUNITY}*
- **Tags**
    - **Name**: Fakulta
    - **Value**: Právnická fakulta
- **Macros**
    - **Macro –** *{$SNMP_COMMUNITY}*
    - **Value –** *****
    - **Type –** Secret

Pre naimportovanie zariadení do Zabbixu musia byť splnené nasledujúce **prerekvizity**:

- Nainštalované knižnice ***pynetbox*** a ***pyzabbix*** na jumphostovi
- Správne užívateľom vyplnené globálne premenné v skripte
- Priradený Primary IP pre každú device inštanciu v NetBoxe
- Vytvorený Custom Field, ktorý bude slúžiť na uchovávanie identifikačného čísla zariadenia v Zabbixe, aby bolo možné jeho ďalšie aktualizovanie prostredníctvom NetBoxu (zachovanie persistencie) **–** pre testovanie bol vytvorený Custom Field ***zabbix_id*** [obr. 23]
- Vytvorený Tag, na základe ktorého bude skript jednoznačne poznať, ktoré zariadenia majú byť do Zabbixu naimportované **–** pre testovanie bol vytvorený Tag ***ZABBIX***
- Vytvorený Config Context, pridelený zariadeniam na základe zvolených vlastností užívateľom, ktorého obsahom bude zoznam ID Zabbix hostgroup a zoznam ID Zabbix templatov s nasledujúcim formátom:

```
{
    "zabbix": {
        "groups": [
            "GROUPIP"
        ],
        "templates": [
            "TEMPLATEID"
        ]
    }
}
```

- Vytvorený Config Template, pomocou ktorého sa na základe prideleného Config Contextu zariadeniam, bude renderovať renderovacím templatovacím jazykom Jinja2 konfigurácia pre Zabbix. Template renderuje nasledujúce položky:
    - Obsah custom fieldu zabbix_id (ID Zabbix hosta)
    - FQDN hosta (*Host name*)
    - Primary IP hosta
    - Status hosta – v prípade, že Device inštancia nemá pridelený Tag nutný pre monitorovanie alebo má iný status ako Active, bude mu v Zabbixu pridelený stav *Disabled* (dočasné vypnutie monitoringu hosta)
    - Groupid z Config Contextu zariadenia
    - Site zariadenia (tag – *site: fakulta*)
    - Templates z Config Contextu zariadenia

Obsah Config Templatu sa nachádza **v prílohe CONFIG_TEMPLATE.J2**

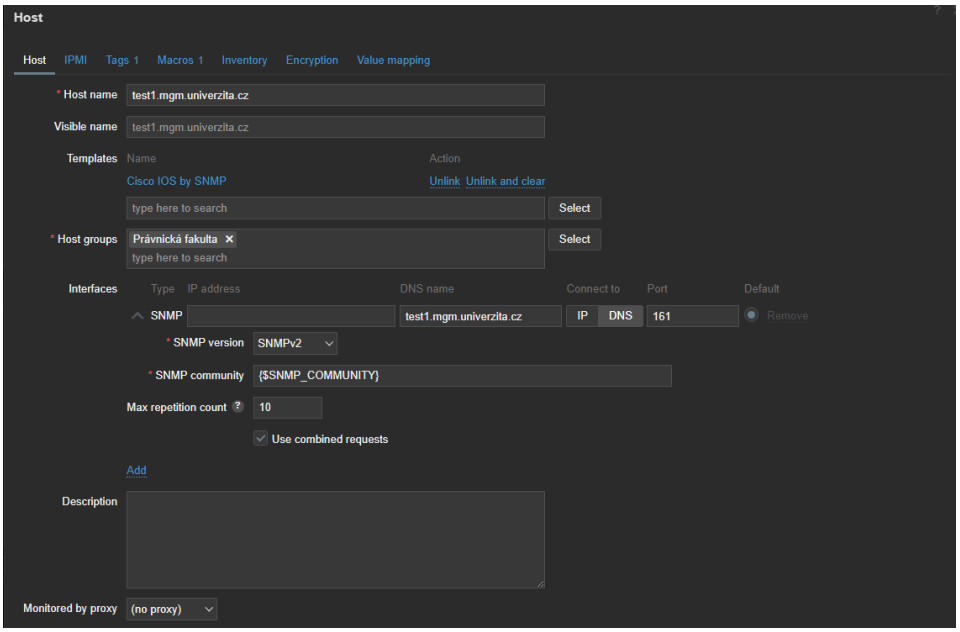
Pre tento import zariadení do Zabbixu som sa rozhodol vybrať možnosť použitia skriptu písaný v Python3 vzhľadom na jeho priamočiary účel. Import je možné vykonať aj pomocou funkcie Event Rule, ktorá je detailne popísaná v kapitole **3.3.2**, s použitím akcie Zabbix Update pri vynútenom aktualizovaní všetkých požadovaných NetBox inštancii.

Skript bude spustený z jumphosta, ktorý má umožnenú konektivitu k serveru NetBoxu.

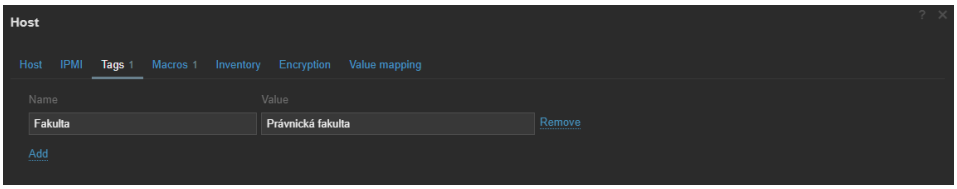
Skript funguje na princípe získavania dát z NetBoxu cez API pomocou *pynetbox* knižnice cez ktorú si vyselektuje zariadenia, ktoré obsahujú tag ***ZABBIX***. Popri selekcii je skript pripojený na Zabbix API pomocou knižnice *pyzabbix* cez ktorú sa vytvoria v Zabbixu hosti pre monitoring. Po pripojení na Zabbix a na NetBox sa vytvorí dictionary zo zoznamu SNMP komunít v súbore *devices.xml*, ktorý je uložený na rovnakom pracovnom adresári ako spúšťaný skript. Následne skript v cykle každej NetBox device inštancii priradí Config Template, ktorý sa vyrenderuje na základe inštancii priradeného Config Contextu. Vo vyrenderovanom Config Template sa následne priradí makru

{$SNMP_COMMUNITY}, ktoré slúži pre uloženie SNMP komunity, SNMP komunita z vytvoreného dictionary. Po upravení vyrenderovaného Config Templatu sa výsledok pošle na Zabbix API metódou create pre vytvorenie nového hosta. Z API odpovede sa následne priradí Zabbix host ID do príslušného custom fieldu device inštancie pre zachovanie persistencie.

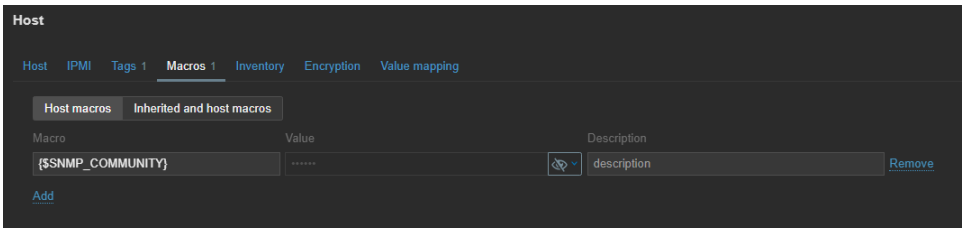
Vývojový diagram skriptu sa nachádza v **prílohe č. 1** a jeho obsah v prílohe **IMPORT_SCRIPT.PY**



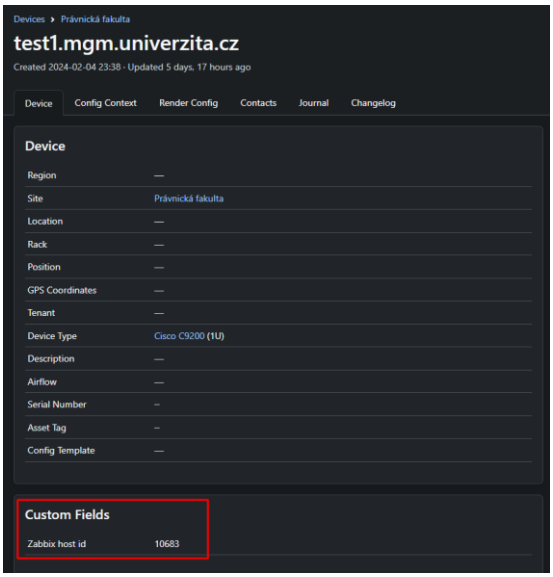
Obrázok 23: Obecné nastavenia vytvoreného testovacieho Zabbix Hosta (Zdroj: Vlastné spracovanie)



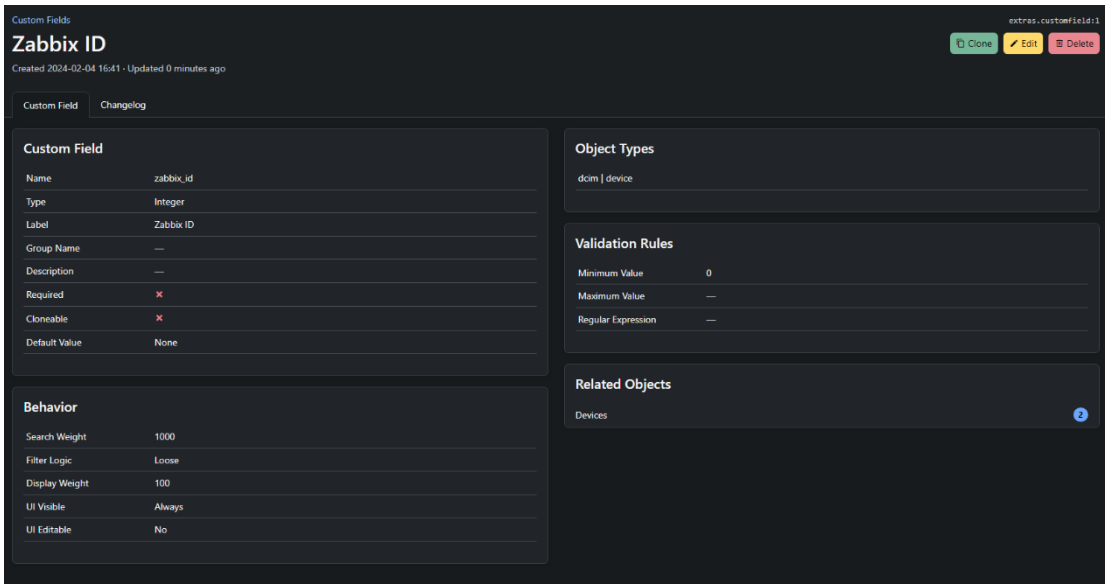
Obrázok 24: Nastavenia Tags testovacieho Zabbix Hosta (Zdroj: Vlastné spracovanie)



Obrázok 25: Nastavenia Macros testovacieho Zabbix Hosta (Zdroj: Vlastné spracovanie)



Obrázok 26: Priradený Zabbix host ID NetBox devicu (Zdroj: Vlastné spracovanie)



Obrázok 27: Nastavenia Custom Fieldu (Zdroj: Vlastné spracovanie)

3.3.2 Zabbix Provisioning

Po iniciačnom importe dát z NetBoxu do Zabbixu je nutné tieto dáta následne držať v Zabbixu aktuálne, keďže bude bežné, že sa dáta v NetBoxu budú často meniť – napr. zmena názvu, statusu, odobratie tagu, pridávanie nových zariadení, zmazanie zariadenia, atď. Zo strany Zabbixu môžu nastať 4 stavy, ktoré dokážu ovplyvniť hosta:

- **Create** – vytvorenie nového hosta, ktorý v Zabbixe neexistuje

- **Update** – aktualizovanie existujúceho hosta napr. zmena FQDN, zmena tagov, atď.
- **Delete** – trvalé zmazanie existujúceho zariadenia
- **Disable** – dočasné vypnutie monitoringu hosta

V NetBoxe môžu nastať 3 akcie, ktoré sa dajú s Device inštanciou vykonať:

- **Create** – vytvorenie novej Device inštancie
- **Update** – zmena atribútov existujúcej Device inštancie napr. Status, Tagy, Site, atď.
- **Delete** – trvalé vymazanie existujúcej inštancie

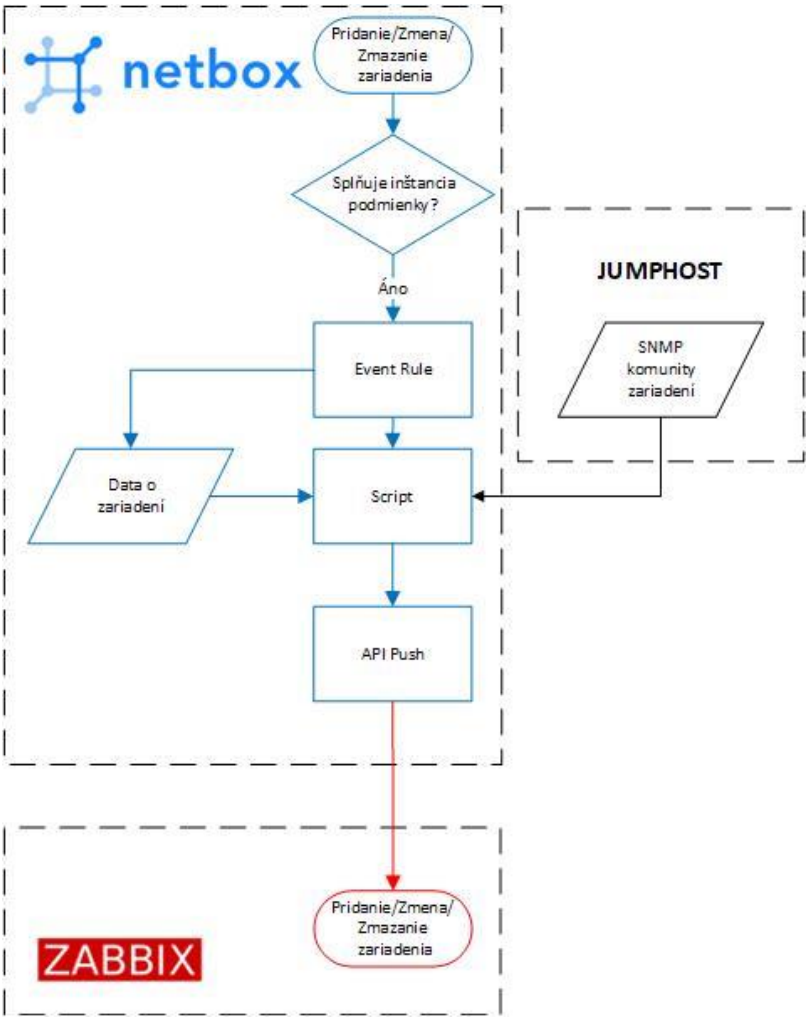
NetBox od verzie *3.7.0* obsahuje možnosť vykonávať určité funkcie (skripty alebo webhooky) v reakcii na zmeny interných objektov nazvanú ***Event Rules***. V mojom prípade použitia by išlo o automatickú konfiguráciu Zabbixu tak, aby začal monitorovať zariadenie, napríklad keď sa jeho prevádzkový stav zmení na *active,* a odstránil ho z monitorovania pri akomkoľvek inom stave pomocou použitia skriptu. Následne sa dá priradiť tento skript k ***Event Rule*** a NetBox bude skript automaticky spúšťať vždy, keď budú splnené nakonfigurované obmedzenia. Každá udalosť musí byť spojená aspoň s jedným typom objektu NetBox a aspoň s jednou udalosťou (napr. vytvorenie, aktualizácia alebo vymazanie).

Tabuľka 6: Zmapovanie udalostí pre Event Rules (Zdroj: Vlastné spracovanie)

	Zabbix akcie			
NetBox akcie	*Create*	*Update*	*Delete*	*Disable*
Create	X	X		
Update	X	X		X
Delete			X	

Pri NetBox akcii *Create,* by mohlo dôjsť k dvom Zabbix udalostiam – Create a Update, čo znamená, že pri vytvorení objektu v NetBoxe by sa host v Zabbixu vytvoril alebo aktualizoval (aktualizoval v prípade, že by inštancia pri vytváraní v NetBoxe už obsahovala hodnotu v custom fielde *Zabbix ID).* Pri akcii NetBox Update by sa host v Zabbixu aktualizoval, vytvoril (vytvoril v prípade, že v custom fielde *Zabbix ID* neexistuje hodnota, čo znamená, že v Zabbixu ešte neexistuje) alebo sa prepol do stavu

Disabled (dočasné vypnutie monitoringu hosta) v prípade, že inštancia Device sa nachádza v inom statuse ako *Active*. Pri akcii NetBox Delete sa host vymaže aj zo Zabbixu.



Obrázok 28: Zabbix provisioning (Zdroj: Vlastné spracovanie)

Podmienky – Zabbix Create

Pri vytváraní inštancie Device sa musia pre spustenie Event Rule splniť nasledujúce podmienky:

- Obsahuje tag *ZABBIX*
- Neobsahuje hodnotu v custom fielde *Zabbix ID*

```
{
    "and": [
        {
            "attr": "tags.name",
            "op": "contains",
            "value": "ZABBIX"
        },
        {
            "attr": "custom_fields.zabbix_id",
            "value": null
        }
    ]
}
```

Podmienky – Zabbix Update

Pri aktualizovaní inštancie Device sa musia pre spustenie Event Rule splniť nasledujúce podmienky:

- Obsahuje hodnotu v custom fielde *Zabbix ID*

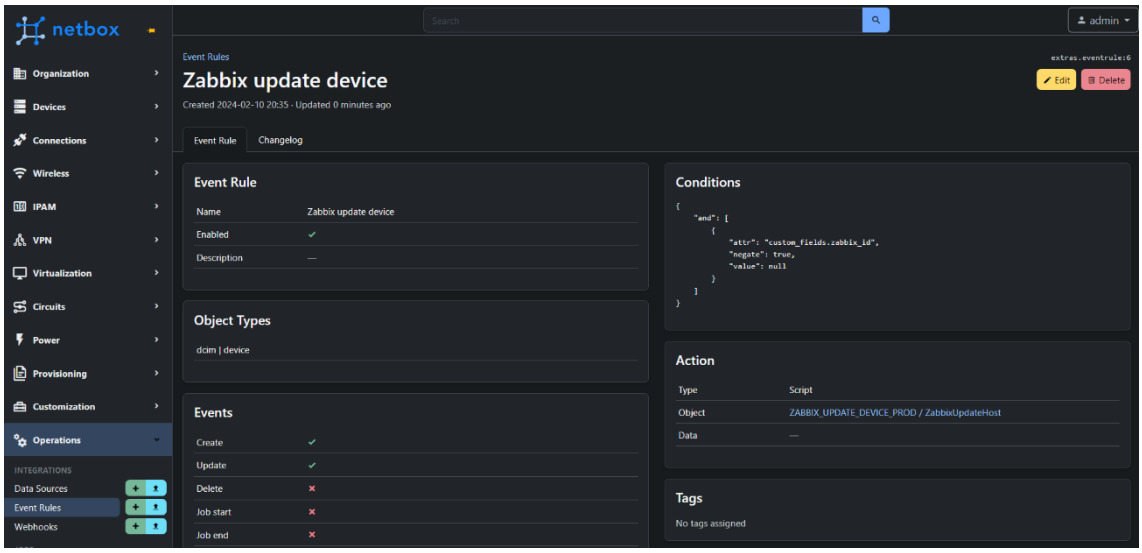
```
{
    "and": [
        {
            "attr": "custom_fields.zabbix_id",
            "negate": true,
            "value": null
        }
    ]
}
```

Podmienky – Zabbix Delete

Pri mazaní inštancie Device sa musia pre spustenie Event Rule splniť nasledujúce podmienky:

- Obsahuje hodnotu v custom fielde *Zabbix ID*

```
{
    "and": [
        {
            "attr": "custom_fields.zabbix_id",
            "negate": true,
            "value": null
        }
    ]
}
```



Obrázok 29: Event Rule Update (Zdroj; Vlastné spracovanie)

Prerekvizity pre správne fungovanie skriptov z NetBox serveru:

- Keďže skripty budú spúšťané z prostredia NetBoxu je nutné mať nainštalovanú knižnice ***paramiko*** a ***pyzabbix*** na NetBox serveri a NetBox workerovi
- Priradený Primary IP pre každú device inštanciu v NetBoxe
- Vytvorený Custom Field, ktorý bude slúžiť na uchovávanie identifikačného čísla zariadenia v Zabbixe, aby bolo možné jeho ďalšie aktualizovanie prostredníctvom NetBoxu (zachovanie persistencie) – pre testovanie bol vytvorený Custom Field ***zabbix_id***
- Vytvorený Tag, na základe ktorého bude script jednoznačne poznať, ktoré zariadenia majú byť do Zabbixu naimportované – pre testovanie bol vytvorený Tag ***ZABBIX***
- Vytvorený Config Context, pridelený zariadeniam na základe zvolených vlastností užívateľom, ktorého obsahom bude zoznam ID Zabbix hostgroup a zoznam ID Zabbix templatov s nasledujúcim formátom:

```
{
    "zabbix": {
        "groups": [
            "GROUPID"
        ],
        "templates": [
            "TEMPLATEID"
        ]
    }
```

- Vytvorený Config Template, pomocou ktorého sa na základe prideleného Config Contextu zariadeniam, bude renderovať renderovacím templatovacím jazykom Jinja2 konfigurácia pre Zabbix. Template renderuje nasledujúce položky:
    - Obsah custom fieldu zabbix_id (ID Zabbix hosta)
    - FQDN hosta (*Host name*)
    - Primary IP hosta
    - Status hosta – v prípade, že Device inštancia nemá pridelený Tag nutný pre monitorovanie alebo má iný status ako Active, bude mu v Zabbixu pridelený stav *Disabled* (dočasné vypnutie monitoringu hosta)
    - Groupid z Config Contextu zariadenia
    - Site zariadenia (tag – *site: fakulta*)
    - Templates z Config Contextu zariadenia

Obsah Config Templatu sa nachádza **v prílohe CONFIG_TEMPLATE.J2**

Skript – Zabbix Create

Skript sa spúšťa po aktivácii NetBox Event Rule, ktorý sa najprv pokusí pripojiť na Jumphosta z ktorého si natiahne všetky SNMP komunity vo forme dictionary a následne sa pripojí na Zabbix. Po tomto kroku si metódou GET vytiahne z NetBoxu Config Template, od Device inštancie Config Context a konkrétnu SNMP komunitu k danému zariadeniu. Následne sa Config Template vyrenderuje na základe Config Contextu a SNMP komunity. Daný vyrenderovaný Config Template sa následne použije pre vytvorenie hosta cez Zabbix API. Po vytvorení Zabbix hosta sa priradí jeho Zabbix ID do príslušného custom fieldu pre zachovanie persistencie.

Vývojový diagram skriptu sa nachádza v **prílohe č. 2** a jeho obsah v prílohe **ZABBIX_CREATE_DEVICE.PY**

Skript – Zabbix Update

Priebeh skriptu je rovnaký ako pri Zabbix Create, avšak po vyrenderovaní Config Templatu je nutné získať dáta o Zabbix interfaces a makrách, keďže sa jedná o samostatné komponenty Zabbix hostov, ktoré uchovávajú vlastné ID. Tieto komponenty je pre ich správne aktualizovanie nutné namapovať na položky vo vyrenderovanom Config Template. Vo vyrenderovanom Config Template sa Interfacom sa na základe

atribútov *dns*, *ip*, *main*, *port*, *type* a *useip* namapuje Zabbix interfaceid a makrám sa na základe atribútov *macro* a *type* namapuje hostmacroid. V prípade, že sa jedná o makro s SNMP komunitou, priradí sa mu z SNMP dictionary príslušná komunita. Tento Config Template sa následne pošle metódou Update na Zabbix API.

Vývojový diagram sa nachádza v **prílohe č. 3** a jeho obsah v prílohe **ZABBIX_UPDATE_DEVICE.PY**

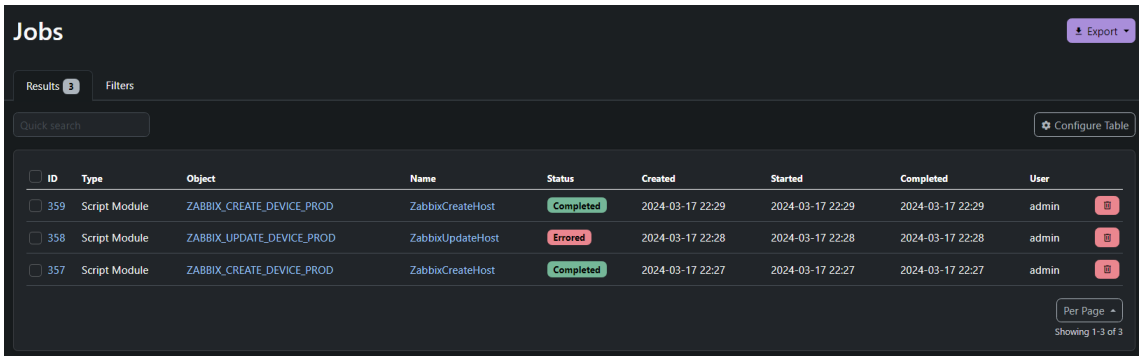
Skript – Zabbix Delete

Pri zmazaní Device inštancie sa script pripojí na Zabbix a na základe poskytnutého ID z Event Rule dát sa pošle na Zabbix metódou Delete API požiadavka, ktorá hosta vymaže.

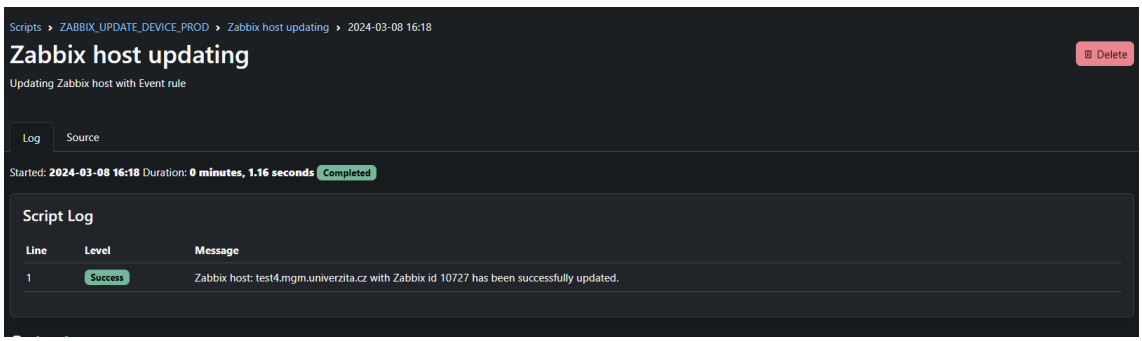
Vývojový diagram sa nachádza v **prílohe č. 4** a jeho obsah v prílohe **ZABBIX_DELETE_DEVICE.PY**

Kontrola

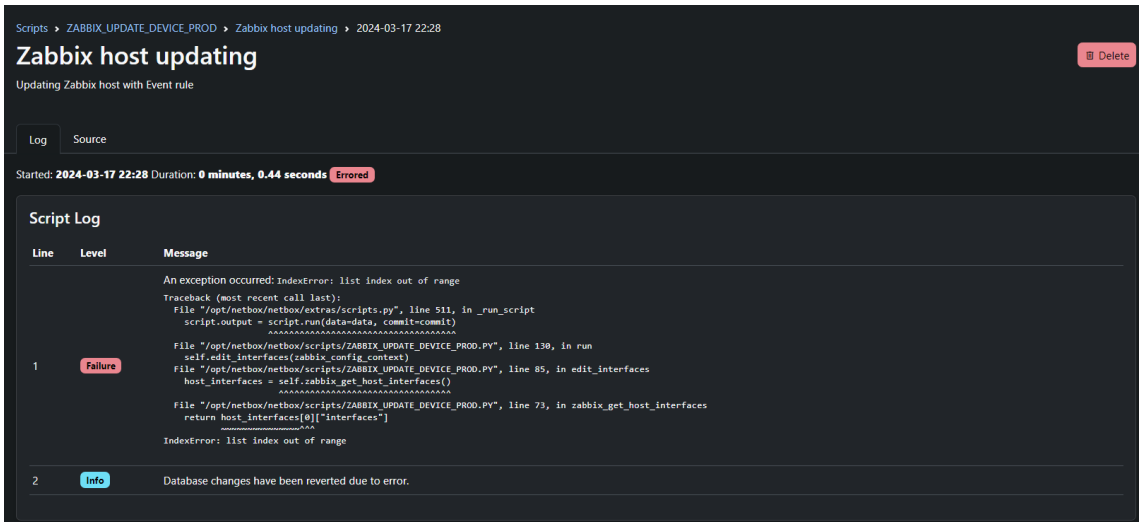
Pri každom spustení jednotlivých skriptov je zaznamenaný ich výsledok NetBoxu, ktorý poskytuje informácie administrátorovi o ich úspešnosti. Tieto logy sú zaznamenané v ponuke menu *Operations →Jobs*.



Obrázok 30: Prehľad NetBox skript logov (Zdroj: Vlastné spracovanie)



Obrázok 31: Prehľad úspešného spustenia NetBox skriptu (Zdroj: Vlastné spracovanie)



Obrázok 32: Prehľad neúspešného spustenia NetBox skriptu (Zdroj: Vlastné spracovanie)

3.3.3 Konfigurácia Zabbixu

V tejto kapitole sa budem podrobne venovať konfigurácii sond a upozornení (triggerov) pre systém Zabbix, ktorý bude nasadený v architektúre typu Manager-Agent. V tejto architektúre systém Zabbix pravidelne dotazuje sledované zariadenia v určitých časových intervaloch a zariadenia na tieto dotazy reagujú.

Prevažná časť dotazovania bude konfigurovaná pomocou SNMP oid, ktoré obsahujú hodnoty pre požadované (monitorované) atribúty. Zabbix identifikuje dva typy položiek (itemov): statické, ktoré poskytujú konštantné hodnoty ako teplota switchu alebo verzia firmwaru zariadenia, a dynamické, ktoré reprezentujú dynamicky meniace sa entity, ako napríklad rozhrania, ktorých počet sa líši medzi rôznymi zariadeniami. Dynamické položky sa konfigurujú v Zabbixu pomocou *Discovery rules*, v ktorých Zabbix vykoná sériu SNMP GETNEXT (SNMP Walk) požiadaviek na zistenie SNMP indexov jednotlivých entít. Tieto indexy sa následne dosadia do SNMP Get požiadavkov k dynamicky vytvoreným itemom, ktoré monitorujú požadované hodnoty. Triggery sa vytvárajú pre oba typy položiek.

```
[
  {
    "{#SNMPINDEX}": "1010",
    "{#SNMPVALUE}": "Switch 1 - Inlet Temp Sensor, GREEN "
  },
  {
    "{#SNMPINDEX}": "1011",
    "{#SNMPVALUE}": "Switch 1 - Outlet Temp Sensor, GREEN "
  },
  {
    "{#SNMPINDEX}": "1012",
    "{#SNMPVALUE}": "Switch 1 - HotSpot Temp Sensor, GREEN "
  },
  {
    "{#SNMPINDEX}": "2010",
    "{#SNMPVALUE}": "Switch 2 - Inlet Temp Sensor, GREEN "
  },
  {
    "{#SNMPINDEX}": "2011",
    "{#SNMPVALUE}": "Switch 2 - Outlet Temp Sensor, GREEN "
  },
  {
    "{#SNMPINDEX}": "2012",
    "{#SNMPVALUE}": "Switch 2 - HotSpot Temp Sensor, GREEN "
  }
]
```


Obrázok 33: Príkladový SNMP výpis Discovery rule pre teplotu komponentov (Zdroj: Vlastné spracovanie)

Každá položka môže byť upravená pred uložením do databázy pomocou tzv. *Preprocessingu* (predspracovania). Tento proces umožňuje ďalšiu úpravu získanej hodnoty, napríklad pomocou regulárnych výrazov, validačných funkcií, alebo JavaScriptových úprav.

Triggery majú v Zabbixe 6 kategórii závažnosti:

- **Not classified** – používa sa v prípadoch, kedy je neznáma závažnosť problému
- **Information** – používa sa pri udalostiach, ktoré nevyžadujú okamžitú pozornosť, ale poskytujú cenné informácie
- **Warning** – používa sa pri udalostiach, ktoré naznačujú potenciálny problém, ktorý vyžaduje pozornosť, ale nie je kritický
- **Average** – používa sa v prípadoch závažného problému, ktorý by sa mal riešiť čím skôr s cieľom predísť ďalším problémom
- **High** – používa sa v prípadoch kritického problému, ktorý vyžaduje okamžitú pozornosť, aby sa predišlo výrazným poruchám
- **Disaster** – používa sa pre označenie závažného incidentu, ktorý vyžaduje okamžitú pozornosť aby sa predišlo výpadku kritických systémov

Triggery sa spúšťajú na základe definovaných výrazov obsahujúce rôzne štatistické, historické, agregačné funkcie, ktorých argumentmi sú sledované itemy. Po spustení triggeru môžu nastať 3 varianty ako konkrétny trigger vyriešiť:

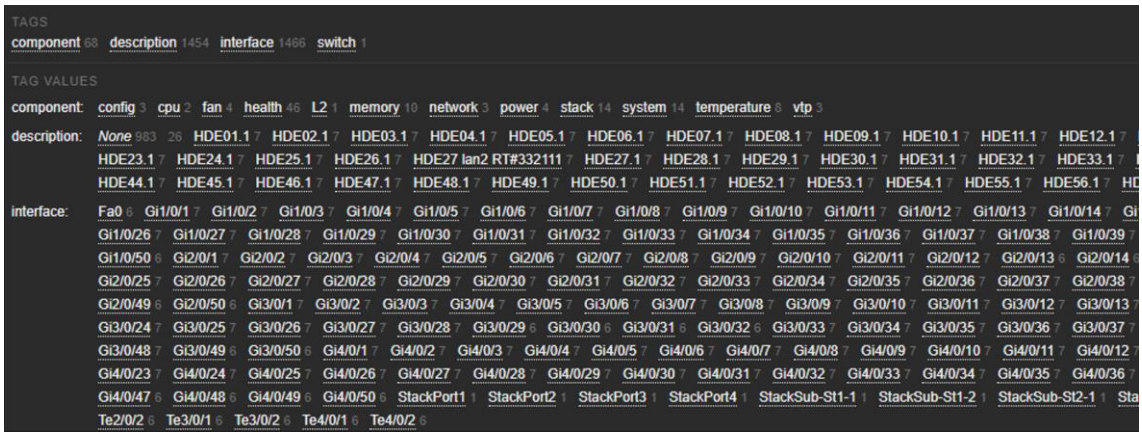
- **Expression** – definovaný výraz pre spustenie triggeru vyhodnotí pri novom načítaní hodnôt z itemov výsledok **False**
- **Recovery expression –** vytvorí sa nový výraz pre vyriešenie triggeru, ktorý sa líši od výrazu spustenia triggeru
- **None** (Manual close) – problém sa vyrieši až po ručnom uzavretí triggeru užívateľom, u ktorého je nutná užívateľská akcia

Rovnako ako pri hostoch, aj pri šablónach, itemoch a triggeroch je v Zabbixe možné pridávať **tagy**. Tieto tagy pomáhajú s filtrovaním a kategorizáciou monitorovaných objektov. Okrem toho sa dajú využiť aj na rozdelenie práv a zobrazení, kde sa môžu obmedziť skupiny užívateľov, ktorí môžu vidieť určitých hostov a udalostí. V mojom návrhu som sa rozhodol rozdeliť tagy pre dané objekty nasledovne:

- **Templates:**
  - ***template: software/hardware/interfaces*** - znázorňuje template, z ktorého item/trigger vznikol
  - ***class: network*** – znázorňuje oblasť infraštruktúry, v našom prípade network
  - ***device: switch/firewall/router*** – reprezentuje typ zariadenia
- **Items** – Itemy by mali vždy jeden tag - ***component***, ktorý by mal na výber nasledujúce hodnoty:
  - **system** – systémové itemy napr. firmware, uptime, atď.
  - **health** – itemy zariadenia, ktoré majú vplyv na jeho prevádzku (teplota, vetráky, zdroj, ...)
  - **L2/L3** – itemy, ktoré sledujú L2/L3 vlastnosti zariadenia (napr. MAC tabuľka, Routing table,...)
  - **config** – itemy, ktoré sledujú stav konfigurácie zariadenia
  - **cpu** – itemy sledujúce procesor
  - **memory** – itemy sledujúce pamäť
  - **stack** – itemy sledujúce stav stacku

- **temperature** – itemy sledujúce teplotu
- **power** – itemy sledujúce napájanie
- **fan** – itemy sledujúce vetráky

- **Items - Interfaces** – Interfacy neobsahujú tag component, ale svoje vlastné tagy:
  - ***description: {#IFALIAS}***
  - ***interface: {#IFNAME}***
  - ***portchannel: Po{#AGGRPORTENTRY}***



Obrázok 34: Ukážka interface tagov v Zabbixu (Zdroj: Vlastné spracovanie)

- **Trigger** – Rovnako ako pri itemoch, triggery budú mať tiež jeden tag - ***scope*** s nasledujúcimi hodnotami:
  - **notice** – trigger, ktorý nemá veľkú váhu a označuje udalosti, ktoré nemajú veľký vplyv (nemal by byť použitý v kombinácii s ***Availability*** a ***Perfomance***)
  - **security** – trigger naznačujúci situáciu spojenú s bezpečnosťou
  - **availability** – trigger, ktorý poukazuje na dostupnosť na samotné zariadenie či už po stránke control plane alebo data plane
  - **performance** – trigger, ktorý poukazuje na situáciu spojenú s nižším výkonom zariadenia
  - **L2/L3** – trigger spojený s L2/L3 funkciami boxu

Všetky itemy, Discovery rules a triggery sú konfigurované v rámci jedného objektu - šablóny (Template). V mojom návrhu som sa rozhodol rozdeliť sondy do štyroch rôznych šablón:

- **ICMP ping** – pre monitorovanie dostupnosti zariadenia
- **Cisco HW inventory** – pre monitorovanie hardvérových atribútov
- **Cisco SW inventory** – pre monitorovanie softvérových atribútov
- **Cisco Interfaces** – pre monitorovanie prevádzky, chýb a záťaže na linkách
- **Fortigate** – pre monitorovanie záťaže a HA statusu Firewallu

Template ICMP Ping

Jedná sa o template, ktorý obsahuje itemy pre sledovanie dostupnosti zariadenia z pohľadu Zabbix serveru, ktoré využívajú zabudované Zabbix funkcie.

ICMP Ping – Items

- **ICMP loss**

Monitoruje percentuálnu stratu ICMP paketov k sledovanému hostovi.

Tabuľka 7: Konfigurácia ICMP loss – Item (Zdroj: Vlastné spracovanie)

Tagy	component: health
Key	icmppingloss
Zabbix funkcia	icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>]
Interval	1 minúta

- **ICMP ping**

Monitoruje dostupnosť sledovaného hosta – vracia 2 hodnoty – 0 nedostupný, 1 dostupný.

Tabuľka 8: Konfigurácia ICMP ping – Item (Zdroj: Vlastné spracovanie)

Tagy	component: health
Key	icmpping
Zabbix funkcia	icmpping[<target>,<packets>,<interval>,<size>,<timeout>]
Interval	30 sekúnd

- **ICMP response time**

Monitoruje odozvu ICMP paketov v sekundách.

Tabuľka 9: Konfigurácia ICMP response time – Item (Zdroj: Vlastné spracovanie)

Tagy	component: health
Key	icmppingsec
Zabbix funkcia	icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>]
Interval	1 minúta

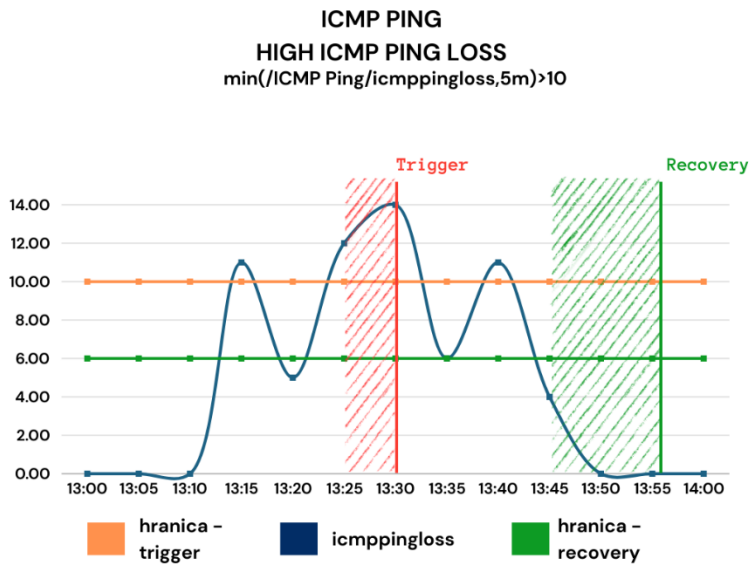
ICMP Ping – Triggers

- **Vysoký ICMP ping loss**

Trigger sa spustí v prípade, že za posledných 5 minút bola minimálna hodnota straty ICMP paketov väčšia ako 10%. Trigger sa automaticky vyrieši, ak za posledných 5 minút bola maximálna hodnota strát ICMP paketov menšia ako 6%.

Tabuľka 10: Konfigurácia High ICMP ping loss – Trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: availability scope: performance
Expression	min(/ICMP Ping/icmppingloss,5m)>10
Recovery	max(/ICMP Ping/icmppingloss,5m)<6
Dependency	ICMP Ping: Unavailable by ICMP ping



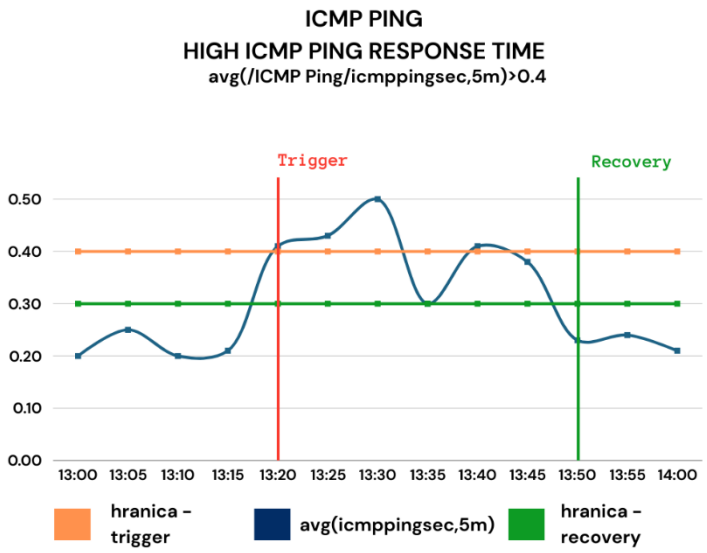
Obrázok 35: Vysoký ICMP ping loss - Trigger (Zdroj: Vlastné spracovanie)

- **Vysoký ICMP ping response time**

Trigger sa spustí v prípade, že za posledných 5 minút bola priemerná odozva ICMP paketov väčšia ako 0,4ms. Trigger sa automaticky vyrieši, ak za posledných 5 minút bola priemerná odozva ICMP paketov menšia ako 0,3ms.

Tabuľka 11: Konfigurácia Vysoký ICMP ping response time – Trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: availability scope: performance
Expression	avg(/ICMP Ping/icmppingsec,5m)>0.4
Recovery	avg(/ICMP Ping/icmppingloss,5m)<0.3
Dependency	ICMP Ping: Unavailable by ICMP ping



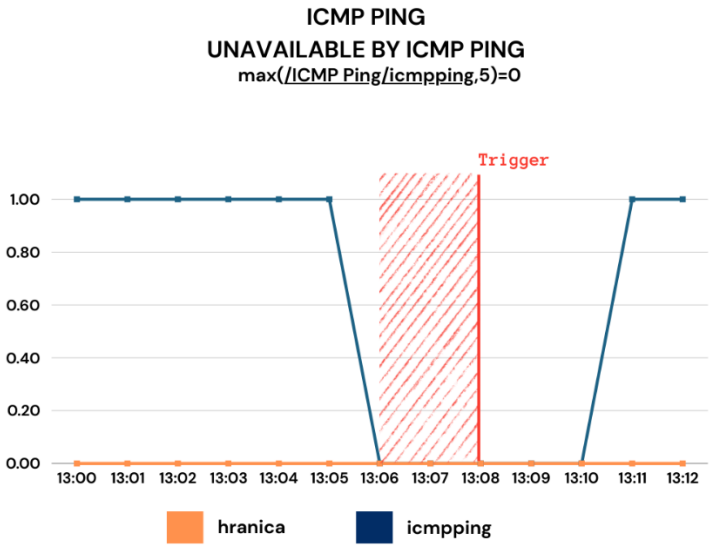
Obrázok 36: Vysoký ICMP ping response - Trigger (Zdroj: Vlastné spracovanie)

- **Nedostupné zariadenie**

Trigger sa spustí v okamihu, kedy posledných 5 maximálnych hodnôt dostupnosti hosta bolo rovné 0. Trigger sa vyrieši, ak ďalších 5 maximálnych hodnôt nebude rovné 0.

Tabuľka 12: Konfigurácia Nedostupné zariadenie – Trigger (Zdroj: Vlastné spracovanie)

Severity	High
Tagy	scope: availability
Expression	max(/ICMP Ping/icmpping,5)=0
Recovery	Expression



Obrázok 37: Nedostupné zariadenie - Trigger (Zdroj: Vlastné spracovanie)

Template Cisco HW Inventory

Jedná sa o template, ktorý obsahuje itemy pre sledovanie hardwarových vlastností sieťového zariadenia ako sú napríklad využitie CPU alebo uptime.

Cisco HW Inventory – Items

- **CPU záťaž**

Monitoruje využitie CPU zariadenia (priemer za posledných 5 minút – v CLI príkaz show processes cpu sorted) v percentách pomocou SNMP oid z MIB *OLD-CISCO-CPU-MIB*.

```
      #show processes cpu sorted
CPU utilization for five seconds: 41%/2%; one minute: 44%; five minutes: 37%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 179  1789907040   830153495       2156 15.11% 14.95% 14.94%   0 Hulc LED Process
 448   355351853   725937488        489  5.27%  5.23%  2.16%   0 SNMP ENGINE
 107   148264757   769740422        192  2.09%  2.20%  0.87%   0 hrpc <- response
 106    21170766   750640408         28  0.41%  0.23%  0.08%   0 hrpc -> request
 258   242160797    32915689       7357  0.35%  0.39%  0.41%   0 PI MATM Aging Pr
 141   134512364    33242128       4046  0.29%  0.30%  0.34%   0 hpm counter proc
 446    17784948    38977170        456  0.17%  0.22%  0.12%   0 IP SNMP
 195    74481492     6629659      11234  0.17%  0.17%  0.17%   0 HQM Stack Proces
 241    39896934    49340762        808  0.11%  0.09%  0.11%   0 Spanning Tree
 196    55109535    39632883       1390  0.11%  0.12%  0.11%   0 HRPC qos request
 329    23643089    95335929        247  0.05%  0.01%  0.00%   0 HULC DHCP Snoopi
 116      405941     5145897         78  0.05%  0.00%  0.00%   0 Hulc Port-Securi
  94    27421542  1514585309         18  0.05%  0.05%  0.05%   0 RedEarth Tx Mana
 132    14041480    32916442        426  0.05%  0.02%  0.00%   0 HLFM aging proce
  15    91684926   133895294        684  0.05%  0.12%  0.12%   0 ARP Input
 230    11950063    31708307        376  0.05%  0.17%  0.07%   0 IP Input
 441     8331887    38688178        215  0.05%  0.02%  0.00%   0 VLAN Manager
 147      164864    23188615          7  0.05%  0.00%  0.00%   0 hl3mm_mfib_ipv6
  91     6649278  1142803243          5  0.05%  0.02%  0.00%   0 Draught link sta
 137    21162663   569273696         37  0.05%  0.04%  0.01%   0 hpm main process
  49     7978579     7470329       1068  0.05%  0.00%  0.00%   0 Net Background
  21      127387     6629693         19  0.00%  0.00%  0.00%   0 IPC Event Notifi
  20           0           1          0  0.00%  0.00%  0.00%   0 IFS Agent Manage
  19         217          69       3144  0.00%  0.00%  0.00%   0 Entity MIB API
  18           0           2          0  0.00%  0.00%  0.00%   0 Policy Manager
  17           0           1          0  0.00%  0.00%  0.00%   0 AAA_SERVER_DEADT
  16      292207    35238476          8  0.00%  0.00%  0.00%   0 ARP Background
```

Obrázok 38: CLI výpis využitia CPU na modely Cisco 2960X (Zdroj: Vlastné spracovanie)

Tabuľka 13: Konfigurácia CPU záťaž – Item (Zdroj: Vlastné spracovanie)

Tagy	component: cpu component: health
Key	cpu.usage.overall
SNMP oid	1.3.6.1.4.1.9.2.1.58.0
Interval	5 minút

- **Voľná pamäť flash**

Monitoruje v bajtoch využitie pamäte flash, v ktorej je na Cisco zariadeniach uložený systém IOS, kópie konfiguračných súborov alebo informácie o VLAN-ách pomocou SNMP oid z MIB *CISCO-FLASH-MIB*.

Tabuľka 14: Konfigurácia Voľná pamäť flash – Item (Zdroj: Vlastné spracovanie)

Tagy	component: cpu component: health
Key	flash.mem.free
SNMP oid	1.3.6.1.4.1.9.9.10.1.1.4.1.1.5.1.1
Interval	30 minút

- **Uptime**

Monitoruje v stotinách sekundy (1/100) (následne prekonvertované do čitateľnej časovej jednotky pomocou *Unit: uptime*) uptime zariadenia od poslednej re-inicializácie systému pomocou SNMP oid z MIB *LBMSH-MIB*.

Tabuľka 15: Konfigurácia Uptime – Item (Zdroj: Vlastné spracovanie)

Tagy	component: system
Key	system.net.uptime[sysUpTime.0]
SNMP oid	1.3.6.1.2.1.1.3.0
Interval	1 hodina
Preprocessing	Custom multiplier: 0.01

- **Počet členov v stacku**

Monitoruje počet itemov *net.stack.member.status.[*]* z Discovery Rule **Stack členovia** pomocou ich sumy (musia byť aktívne pre ich zarátanie)

Tabuľka 16: Konfigurácia Počet členov v stacku – Item (Zdroj: Vlastné spracovanie)

Tagy	component: stack
Key	net.stack.member.count
Type	Calculated
Formula	sum(exists_foreach(//net.stack.member.status.[*]))
Interval	1 hodina

- **Sériové číslo**

Monitoruje sériové číslo zariadenia pomocou SNMP oid z MIB *ENTITY-MIB*.

Tabuľka 17: Konfigurácia Sériové číslo – Item (Zdroj: Vlastné spracovanie)

Tagy	component: system
Key	system.hw.serialnumber
SNMP oid	1.3.6.1.2.1.47.1.1.1.1.11.1
Interval	1 hodina

- **Model**

Monitoruje model zariadenia pomocou SNMP oid z MIB *ENTITY-MIB*.

Tabuľka 18: Konfigurácia Model – Item (Zdroj: Vlastné spracovanie)

Tagy	component: system
Key	system.hw.model
SNMP oid	1.3.6.1.2.1.47.1.1.1.1.13.1
Interval	1 hodina

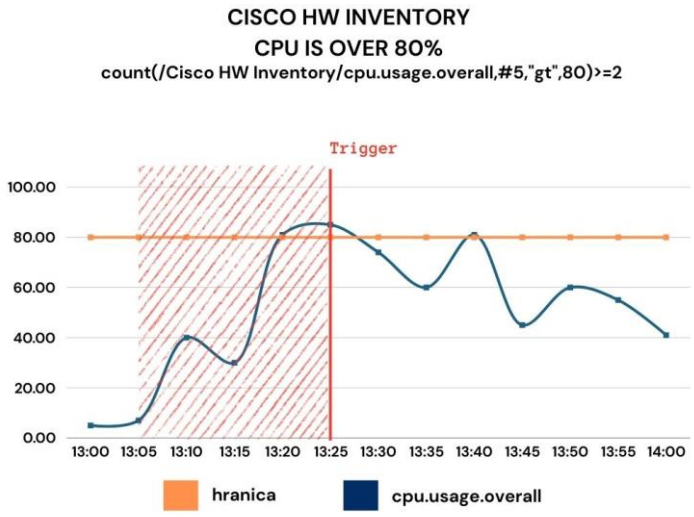
Cisco HW Inventory – Triggers

- **CPU využitie je nad 80%**

Trigger sa spustí v prípade, kedy za posledných 5 nameraných hodnôt bolo využitie CPU pri aspoň dvoch hodnotách väčšie ako 80%. Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí preskúmať zariadenie, zistiť, ktorý proces ho zaťažil a optimalizovať ho.

Tabuľka 19: Konfigurácia CPU využitie je nad 80% – Trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: performance
Expression	count(/Cisco HW Inventory/cpu.usage.overall,#5,"gt",80)>=2
Recovery	None



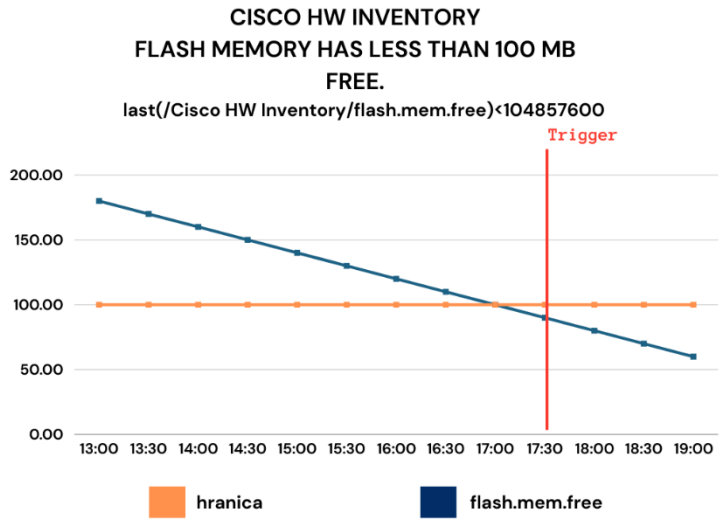
Obrázok 39: CPU využitie je nad 80% – Trigger (Zdroj: Vlastné spracovanie)

- **Flash pamäť má menej ako 100 MB voľného priestoru**

Trigger sa spustí v prípade, že posledná nameraná hodnota voľnej pamäte flash je menšia ako 100MB (104857600 bajtov). Trigger sa musí zatvoriť užívateľom, ktorý musí manuálne na zariadení uvoľniť pamäť.

Tabuľka 20: Konfigurácia Flash pamäť má menej ako 100 MB voľného priestoru – Trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: notice
Expression	last(/Cisco HW Inventory/flash.mem.free)<104857600
Recovery	None



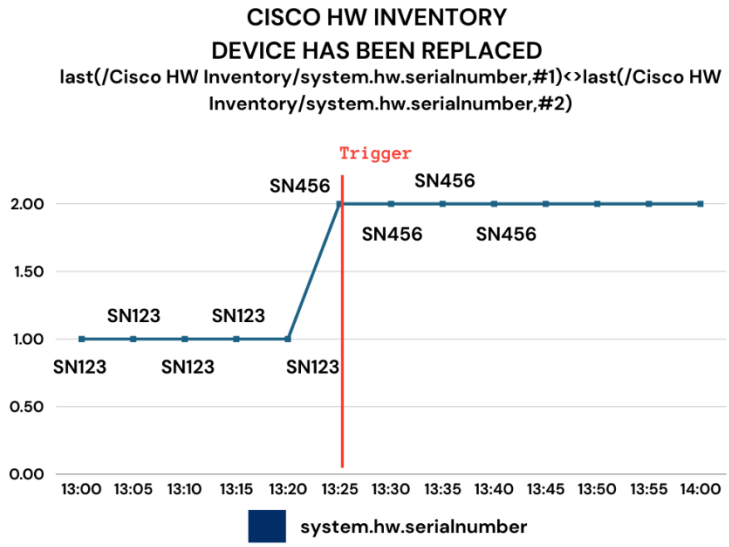
Obrázok 40: Flash pamäť má menej ako 100 MB voľného priestoru – Trigger (Zdroj: Vlastné spracovanie)

- **Zariadenie bolo vymenené**

Trigger sa spustí v prípade, že posledná hodnota sériového čísla sa nerovná predposlednej nameranej hodnote. Trigger sa musí zatvoriť užívateľom, ktorý musí manuálne overiť, čo sa so zariadením skutočne stalo (často sa stáva, že sa prehodil iný člen clusteru/stacku na mastera).

Tabuľka 21: Konfigurácia Zariadenie bolo vymenené – Trigger (Zdroj: Vlastné spracovanie)

Severity	Information
Tagy	scope: notice
Expression	last(/Cisco HW Inventory/system.hw.serialnumber,#1)<>last(/Cisco HW Inventory/system.hw.serialnumber,#2)
Recovery	None



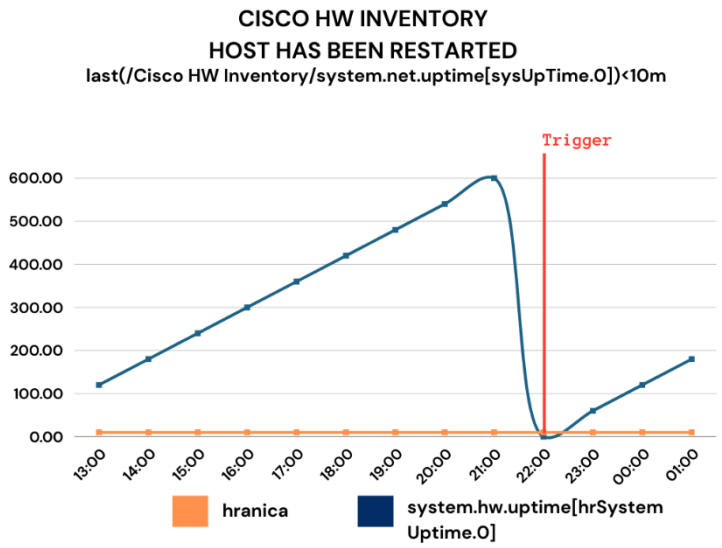
Obrázok 41: Zariadenie bolo vymenené – Trigger (Zdroj: Vlastné spracovanie)

- **Zariadenie bolo reštartované**

Trigger sa spustí v prípade, že posledná nameraná hodnota uptimu zariadenia bola menšia ako 10 minút. Trigger sa musí zatvoriť užívateľom, ktorý musí manuálne overiť, čo sa so zariadením skutočne stalo (môže nastať stav *false positive* v prípade, že pretečie counter).

Tabuľka 22: Konfigurácia Zariadenie bolo reštartované – Trigger (Zdroj: Vlastné spracovanie)

Severity	Information
Tagy	scope: notice
Expression	last(/Cisco HW Inventory/system.hw.uptime[hrSystemUptime.0])<10m
Recovery	None



Obrázok 42: Zariadenie bolo reštartované – Trigger (Zdroj: Vlastné spracovanie)

Cisco HW Inventory – Discovery rules

- **Teplota**

Jedná sa o Discovery rule, ktorý pomocou SNMP oid z MIB *CISCO-ENVMON-MIB*, vykoná SNMP walk, z ktorého zistí pre každý switch (standalone/v stacku) stav teploty ich komponentov. LLD macro **SNMPVALUE** obsahuje názvy komponentov switchu, u ktorých sa meria teplota.

Tabuľka 23: Konfigurácia Teplota – Discovery rule (Zdroj: Vlastné spracovanie)

Key	temperature.discovery
SNMP oid	discovery[{#SNMPVALUE},1.3.6.1.4.1.9.9.13.1.3.1.2]
Interval	1 deň

Discovery items

- **{#SNMPVALUE}: Teplota**

Monitoruje sledovaný komponent v stupňoch Celzia pomocou SNMP oid z MIB *CISCO-ENVMON-MIB.*

Tabuľka 24: Konfigurácia Teplota – Discovery Item (Zdroj: Vlastné spracovanie)

Tagy	component: health component: temperature
Key	sensor.temp.value[ciscoEnvMonTemperatureValue.{#SNMPINDEX}]
SNMP oid	1.3.6.1.4.1.9.9.13.1.3.1.3.{#SNMPINDEX}
Interval	5 minút

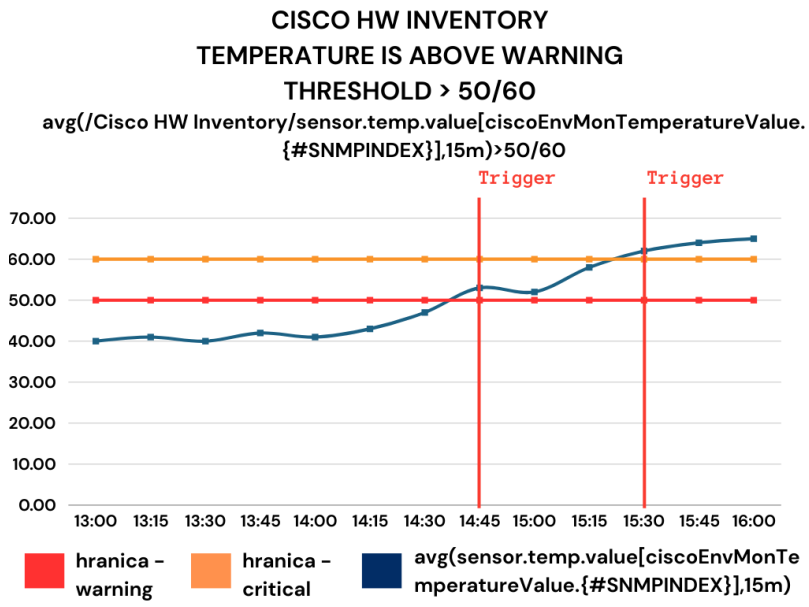
Discovery triggers

- **{#SNMPVALUE}: Teplota je nad povolenou hranicou**

Trigger sa spustí v prípade, že priemerná hodnota teploty komponentu za posledných 15 minút presiahla viac ako 50 stupňov Celzia. Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí overiť, čo sa so zasiahnutým komponentom stalo (skontrolovať umiestnenie zariadenia).

Tabuľka 25: Konfigurácia Teplota je nad povolenou hranicou – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: performance scope: availability
Expression	avg(/Cisco HW Inventory/sensor.temp.value[ciscoEnvMonTemperatureValue.{#SNMPINDEX}],15m)>50
Recovery	None



Obrázok 43: Teplota je nad povolenou hranicou – Discovery trigger (Zdroj: Vlastné spracovanie)

- **Zdroje**

Ide o Discovery rule, ktorý pomocou SNMP OID z MIB *CISCO-ENVMON-MIB*, vykoná SNMP walk, z ktorého zistí pre každý switch (standalone/v stacku) stav zdrojov. LLD macro **SENSOR_INFO** obsahuje názvy zdrojov switchu (napr. SW1 PS A,B,...).

Tabuľka 26: Konfigurácia Zdroje – Discovery rule (Zdroj: Vlastné spracovanie)

Key	psu.discovery
SNMP oid	discovery[{#SENSOR_INFO},1.3.6.1.4.1.9.9.13.1.5.1.2]
Interval	1 deň

Discovery items

- **{#SENSOR_INFO}: Stav zdroja**

Monitoruje stav sledovaného zdroja pomocou SNMP oid z MIB *CISCO-ENVMON-MIB*, ktorý môže nadobúdať 6 SNMP hodnôt:

1. *Normal*
2. *Warning*
3. *Critical*
4. *Shutdown*

5. *Not Present* (Nezobrazí sa pri SNMP walk)
6. *Not functioning* (Warning)

Tabuľka 27: Konfigurácia Stav zdroja – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	component: health component: power
Key	sensor.psu.status[ciscoEnvMonSupplyState.{#SNMPINDEX}]
SNMP oid	1.3.6.1.4.1.9.9.13.1.5.1.3.{#SNMPINDEX}
Interval	5 minút

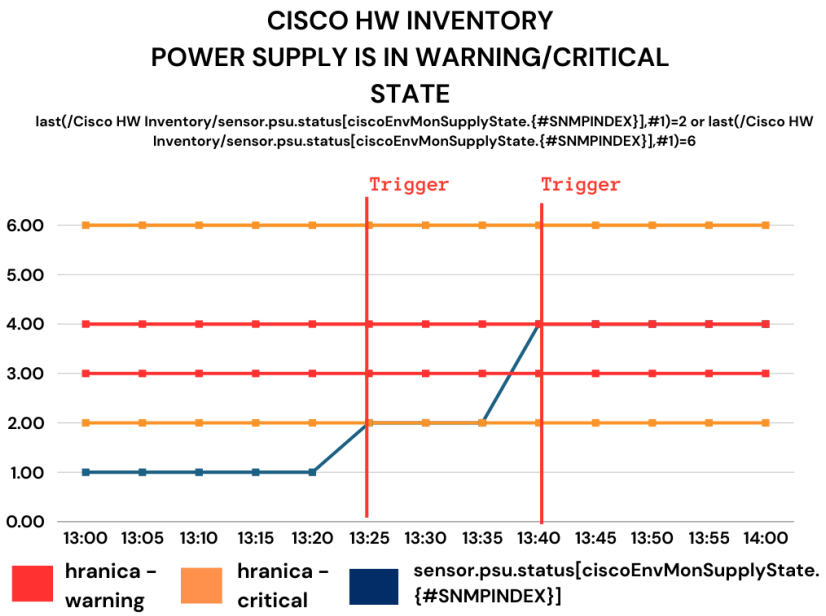
Discovery triggers

- **{#SENSOR_INFO}: Zdroj je vo varovnom stave**

Trigger sa spustí v prípade, že posledná hodnota stavu zdroja nadobudla hodnoty **2** (Warning) alebo **6** (Not Functioning). Pre trigger sledujúci kritický stav sa jedná o hodnoty **3** (Critical) a **4** (Shutdown). Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí overiť, čo sa so zasiahnutým zdrojom stalo.

Tabuľka 28: Konfigurácia Zdroj je vo varovnom stave – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Average
Tagy	scope: performance scope: availability
Expression	last(/Cisco HW Inventory/sensor.psu.status[ciscoEnvMonSupplyState.{#SNMPINDEX}],#1)=2 or last(/Cisco HW Inventory/sensor.psu.status[ciscoEnvMonSupplyState.{#SNMPINDEX}],#1)=6
Recovery	None



Obrázok 44: Zdroj je vo varovnom stave – Discovery trigger (Zdroj: Vlastné spracovanie)

- **Ventilátory**

Jedná sa o Discovery rule, ktorý pomocou SNMP OID z MIB *CISCO-ENVMON-MIB*, vykoná SNMP walk, z ktorého zistí pre každý switch (standalone/v stacku) stav ventilátorov. LLD macro **SENSOR_INFO** obsahuje názvy ventilátorov switchu (napr. SW1 FAN TR 1,2,...).

Tabuľka 29: Konfigurácia Ventilátory – Discovery rule (Zdroj: Vlastné spracovanie)

Key	fan.discovery
SNMP oid	discovery[{#SENSOR_INFO},1.3.6.1.4.1.9.9.13.1.4.1.2]
Interval	1 deň

Discovery items

- **{#SENSOR_INFO}: Status ventilátorov**

Monitoruje stav sledovaného ventilátora pomocou SNMP oid z MIB *CISCO-ENVMON-MIB*, ktorý môže nadobúdať 6 SNMP hodnôt:

1. *Normal*
2. *Warning*

3. *Critical*
4. *Shutdown*
5. *Not Present* (Nezobrazí sa pri SNMP walk)
6. *Not functioning* (Warning)

Tabuľka 30: Konfigurácia Status ventilátorov – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	component: health component: fan
Key	fan.status[fanState.{#SNMPINDEX}]
SNMP oid	1.3.6.1.4.1.9.9.13.1.4.1.3.{#SNMPINDEX}
Interval	5 minút

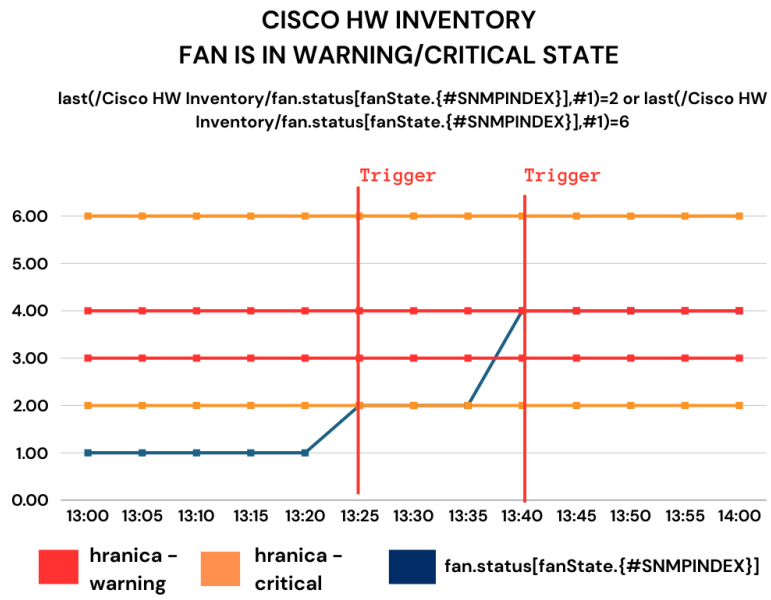
Discovery triggers

- **{#SENSOR_INFO}: Ventilátor je vo varovnom stave**

Trigger sa spustí v prípade, že posledná hodnota stavu ventilátora nadobudla hodnoty **2** (Warning) alebo **6** (Not Functioning). Pre trigger sledujúci kritický stav sa jedná o hodnoty **3** (Critical) a **4** (Shutdown). Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí overiť, čo sa so zasiahnutým ventilátorom stalo.

Tabuľka 31: Konfigurácia Ventilátor je vo varovnom stave – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Average
Tagy	scope: performance scope: availability
Expression	last(/Cisco HW Inventory/fan.status[fanState.{#SNMPINDEX}],#1)=2 or last(/Cisco HW Inventory/fan.status[fanState.{#SNMPINDEX}],#1)=6
Recovery	None



Obrázok 45: Ventilátor je vo varovnom stave – Discovery trigger (Zdroj: Vlastné spracovanie)

- **Stack členovia**

Ide o Discovery rule, ktorý pomocou SNMP OID z MIB *CISCO-STACKWISE-MIB*, vykoná SNMP walk, z ktorého zistí počet a poradie switchov v stacku. Popritom sa spustia ďalšie SNMP walky pre interfacy, ktoré splňujú filter majúc v LLD macre **IFNAME** string „*Stack*“. LLD macro **CSWSWITCHINFOENTRY** obsahuje poradie switchov a macro **IFOPERSTATUS** obsahuje operational status stackujúcej linky.

Tabuľka 32: Konfigurácia Stack členovia – Discovery rule (Zdroj: Vlastné spracovanie)

Key	net.stack.members.discovery
SNMP oid	discovery[{#CSWSWITCHINFOENTRY},1.3.6.1.4.1.9.9.500.1.2.1.1.1, {#IFOPERSTATUS},1.3.6.1.2.1.2.2.1.8,{#IFNAME},1.3.6.1.2.1.31.1.1.1.1]
Interval	1 deň
Filter	**{#IFNAME}** matches **Stack.***

Discovery items

- **Stack member {#CSWSWITCHINFOENTRY}: Status stack člena**

Monitoruje stav sledovaného stack člena pomocou SNMP oid z MIB *CISCO-STACKWISE-MIB*, ktorý môže nadobúdať 11 SNMP hodnôt (literárny zdroj č. **19**), z toho reprezentuje funkčný stav iba hodnota **4** (ready).

Tabuľka 33: Konfigurácia Status stack člena – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	component: stack
Key	net.stack.member.status.[{#SNMPINDEX}]
SNMP oid	1.3.6.1.4.1.9.9.500.1.2.1.1.6.{#SNMPINDEX}
Interval	5 minút

- **Interface {#IFNAME}: Stack interface status**

Monitoruje funkčný stav sledovaného stack interfacu pomocou SNMP oid z MIB *LBMSH-MIB*, ktorý môže nadobúdať 3 SNMP hodnoty:

1. *Up*
2. *Down*
3. *Testing*

Tabuľka 34: Konfigurácia Stack Interface status – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	component: stack interface: {#IFNAME}
Key	net.stack.if.status[ifOperStatus.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.2.2.1.8.{#SNMPINDEX}
Interval	5 minút

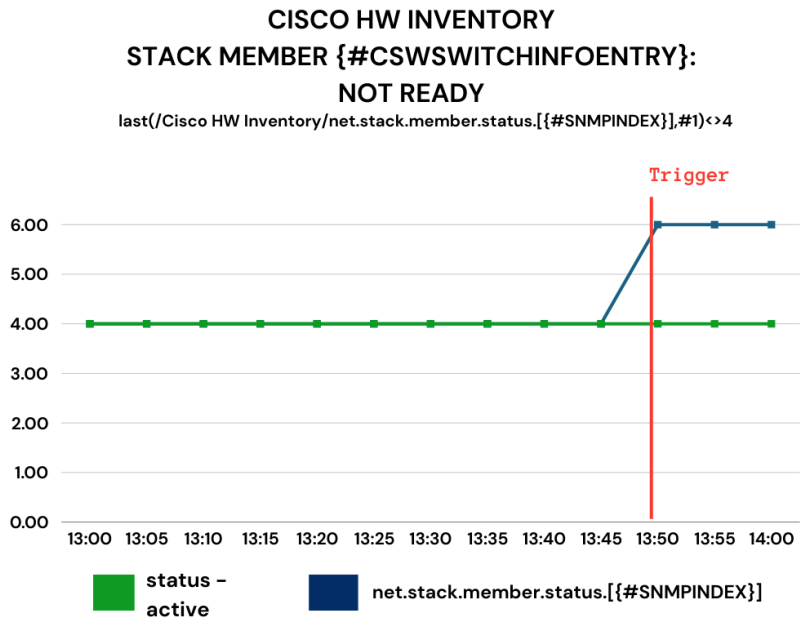
Discovery triggers

- **Člen stacku {#CSWSWITCHINFOENTRY}: Nie je pripravený**

Trigger sa spustí v prípade, že posledná hodnota stavu člena stacku nadobudla hodnotu, ktorá nie je rovná **4** (ready). Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí overiť, čo sa s členom stacku stalo.

Tabuľka 35: Konfigurácia Člen stacku nie je pripravený – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Average
Tagy	scope: availability
Expression	last(/Cisco HW Inventory/net.stack.member.status.[{#SNMPINDEX}],#1)<>4
Recovery	None



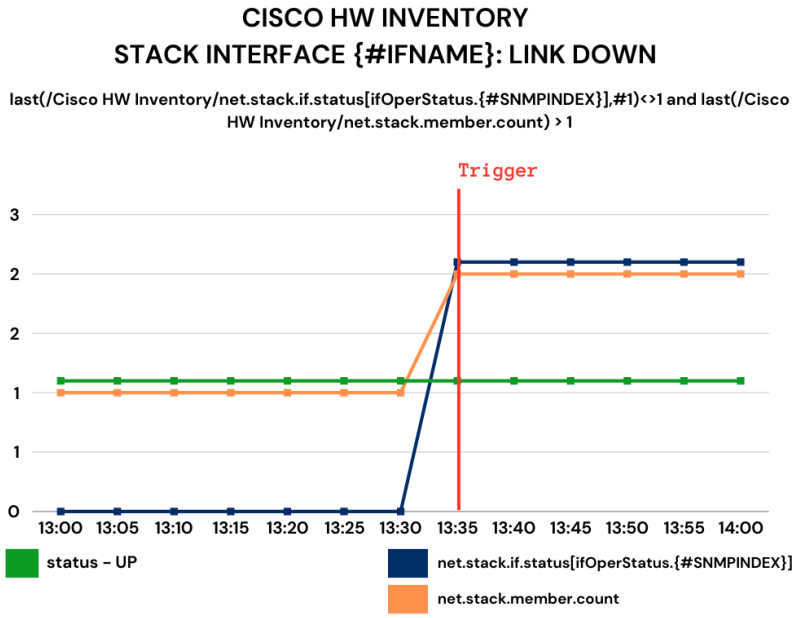
Obrázok 46: Člen stacku nie je pripravený – Discovery trigger (Zdroj: Vlastné spracovanie)

- **Stack interface {#IFNAME}: Link down**

Trigger sa spustí v momente, kedy posledná hodnota funkčného stavu stack linky nie je rovná **1** (Up) a počet členov v stacku je väčší ako 1 (už sa nejedná o standalone switch). Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí overiť, čo sa so stack interfacom stalo.

Tabuľka 36: Konfigurácia Stack interface Link down – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: availability
Expression	last(/Cisco HW Inventory/net.stack.if.status[ifOperStatus.{#SNMPINDEX}],#1)<>1 and last(/Cisco HW Inventory/net.stack.member.count) > 1
Recovery	None



Obrázok 47: Stack interface Link down – Discovery trigger (Zdroj: Vlastné spracovanie)

Template Cisco SW Inventory

Jedná sa o template, ktorý obsahuje itemy pre sledovanie softwarových vlastností sieťového zariadenia.

Cisco SW Inventory – Items

- **Využitie MAC tabuliek**

Monitoruje využitie MAC tabuľky switcha v percentách pomocou Zabbix SSH agenta, ktorý sa prihlasuje na zariadenie cez poskytnuté prihlasovacie údaje (TACACS, Radius, lokálny používateľ,…) a pomocou príkazu *show mac address-table count* si cez

JavaScript preprocessing vypočíta jej využitie. Switch v tomto príkaze uvádza 2 riadky obsahujúce počet využívaných MAC adries a počet voľných MAC adries. JavaScript si pomocou regex výrazu vyparsuje obe riadky a extrahuje ich hodnoty. Tieto hodnoty sú následne dosadené do vzorca:

$$\text{Využitie MAC tabuľky}(\%) = \frac{Total\ MAC\ Adress\ in\ Use}{Total\ MAC\ Adress\ in\ Use + Total\ MAC\ Address\ Space\ Available} \times 100$$

Na starších modeloch Cisco switchov sa uvádza iba údaj o počte voľných MAC adries – pre tento prípad sa v skripte vytvorili odhadované hodnoty využitia MAC tabuliek, ktoré sú spomenuté v JavaScripte v prílohe **MAC_TABLE_USAGE.JS**

```
Total Dynamic Address Count  : 5
Total Static  Address Count  : 7
Total Mac Address In Use     : 12
Total Mac Address Space Available: 16372
```

Obrázok 48: Výpis z príkazu show mac address-table count (Zdroj: Vlastné spracovanie)

Tabuľka 37: Konfigurácia Využitie MAC tabuliek – Item (Zdroj: Vlastné spracovanie)

Tagy	component: L2
Type	SSH Agent
Key	ssh.run[show macs, {HOST.HOST}, 22]
Executed script	Show mac address-table count
Interval	5 minút
Preprocessing	JavaScript: Script v prílohe **MAC_TABLE_USAGE.JS**

- **Operačný systém**

Monitoruje aktuálne nasadený firmware switcha pomocou SNMP oid z MIB *LBMSH-MIB*.

Tabuľka 38: Konfigurácia Operačný systém – Item (Zdroj: Vlastné spracovanie)

Tagy	component: system
Key	system.sw.os[sysDescr.0]
SNMP oid	1.3.6.1.2.1.1.1.0
Interval	1 hodina
Preprocessing	Regex: Version (.+), RELEASE

- **Running config – posledná zmena**

Monitoruje v stotinách sekundy (1/100) poslednú zmenu v konfigurácii (napr. 3 dni 15 minút a 10 sekúnd) pomocou SNMP oid z MIB *CISCO-CONFIG-MAN-MIB.*

Tabuľka 39: Konfigurácia Running config – posledná zmena – Item (Zdroj: Vlastné spracovanie)

Tagy	component: config
Key	run.config.last.changed
SNMP oid	1.3.6.1.4.1.9.9.43.1.1.1.0
Interval	5 minút
Preprocessing	Custom multiplier: 0.01

- **Running config – posledné uloženie**

Monitoruje v stotinách sekundy (1/100) posledné uloženie konfigurácie (napr. 3 dni 15 minút a 15 sekúnd) pomocou SNMP oid z MIB *CISCO-CONFIG-MAN-MIB.*

Tabuľka 40: Konfigurácia Running config – posledné uloženie – Item (Zdroj: Vlastné spracovanie)

Tagy	component: config
Key	run.config.last.saved
SNMP oid	1.3.6.1.4.1.9.9.43.1.1.2.0
Interval	5 minút
Preprocessing	Custom multiplier: 0.01

- **Running config – status uloženia**

Item, ktorý odčíta posledné hodnoty itemov posledného uloženia od poslednej zmeny.

Tabuľka 41: Konfigurácia Running config – status uloženia – Item (Zdroj: Vlastné spracovanie)

Tagy	component: config
Key	run.config.status
Type	Calculated
Formula	last(//run.config.last.changed)-last(//run.config.last.saved)
Interval	5 minút

- **Názov systému**

Monitoruje aktuálny názov systému (hostname) pomocou SNMP oid z MIB *CPQHOST-MIB*.

Tabuľka 42: Konfigurácia Názov systému – Item (Zdroj: Vlastné spracovanie)

Tagy	component: system
Key	system.name
SNMP oid	1.3.6.1.2.1.1.5.0
Interval	1 hodina

- **VTP mode**

Monitoruje aktuálny VTP mode (Client, Server) pomocou SNMP oid z MIB *CISCO-VTP-MIB*.

Tabuľka 43: Konfigurácia VTP mode – Item (Zdroj: Vlastné spracovanie)

Tagy	component: system
Key	system.vtp.mode
SNMP oid	1.3.6.1.4.1.9.9.46.1.2.1.1.3.1
Interval	12 hodín

- **VTP verzia**

Monitoruje nasadenú VTP verziu (v1, v2, v3) pomocou SNMP oid z MIB *CISCO-VTP-MIB*.

Tabuľka 44: Konfigurácia VTP verzia – Item (Zdroj: Vlastné spracovanie)

Tagy	component: system
Key	system.vtp.version
SNMP oid	1.3.6.1.4.1.9.9.46.1.2.1.1.11.1
Interval	12 hodín

- **VTP dómena**

Monitoruje VTP dómenu, v ktorej sa switch nachádza pomocou SNMP oid z MIB *CISCO-VTP-MIB*.

Tabuľka 45: Konfigurácia VTP dómena – Item (Zdroj: Vlastné spracovanie)

Tagy	component: system
Key	system.vtp.domain
SNMP oid	1.3.6.1.4.1.9.9.46.1.2.1.1.2.1
Interval	12 hodín

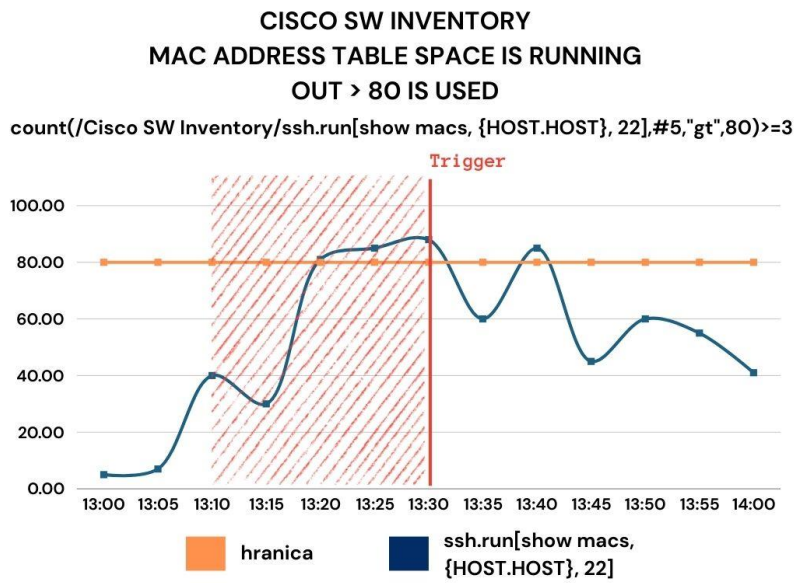
Cisco SW Inventory – Triggers

- **Priestor v MAC tabuľke je využitý > 80%**

Trigger sa spustí v prípade, že z posledných piatich nameraných hodnôt využitia MAC tabuľky boli aspoň 3 hodnoty väčšie ako 80%. Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí overiť a uvoľniť miesto MAC tabuľky.

Tabuľka 46: Konfigurácia Priestor v MAC tabuľke je využitý > 80% – Trigger (Zdroj: Vlastné spracovanie)

Severity	Average
Tagy	scope: performance scope: availability scope: L2
Expression	count(/Cisco SW Inventory/ssh.run[show macs, {HOST.HOST}, 22],#5,"gt",80)>=3
Recovery	None



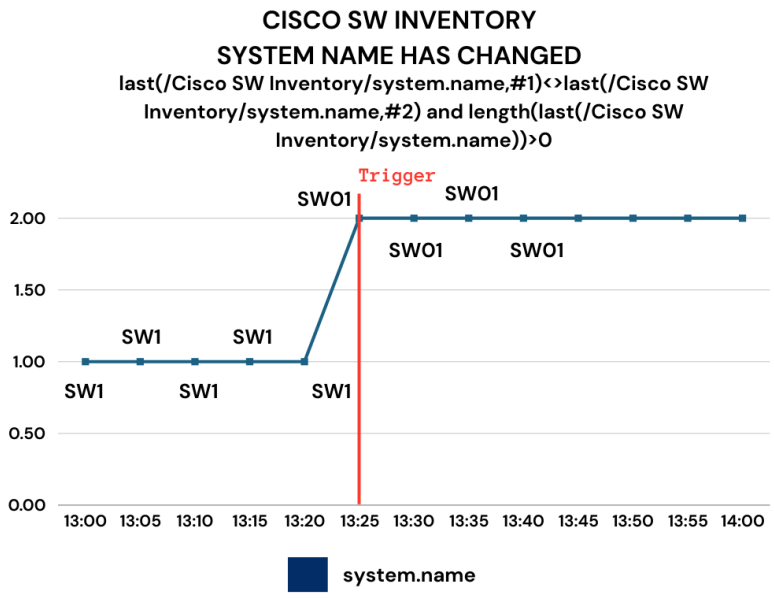
Obrázok 49: Priestor v MAC tabuľke je využitý > 80% – Trigger (Zdroj: Vlastné spracovanie)

- **Názov systému bol zmenený**

Trigger sa spustí v momente, že posledná nameraná hodnota názvu systému sa nezhoduje s predposlednou hodnotou a dĺžka názvu je väčšia ako 1. Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí potvrdiť, že premenovanie bolo úmyselné.

Tabuľka 47: Konfigurácia Názov systému bol zmenený – Trigger (Zdroj: Vlastné spracovanie)

Severity	Information
Tagy	scope: notice
Expression	last(/Cisco SW Inventory/system.name,#1)<>last(/Cisco SW Inventory/system.name,#2) and length(last(/Cisco SW Inventory/system.name))>0
Recovery	None



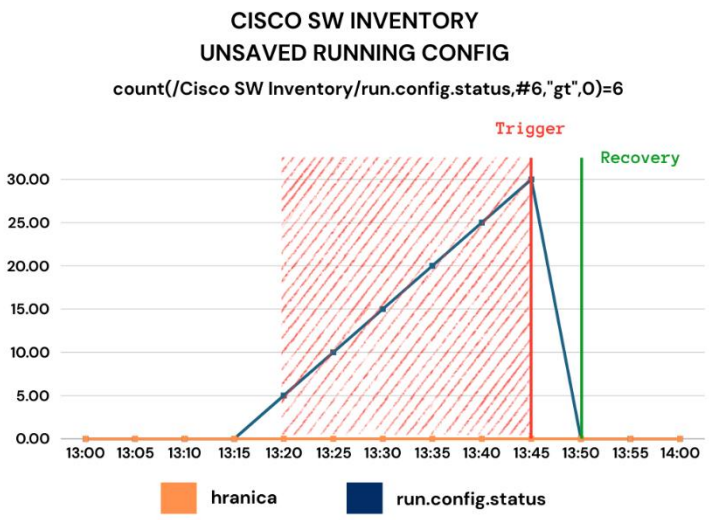
Obrázok 50: Názov systému bol zmenený – Trigger (Zdroj: Vlastné spracovanie)

- **Neuložená konfigurácia**

Trigger sa spustí vo chvíli, kedy je rozdiel medzi poslednou zmenou a posledným uložením (item *Running config – status uloženia)* väčší ako 0 za posledných 6 nameraných hodnôt (môže nastať stav false positive v prípade, že pretečie counter). Trigger sa automaticky vyrieši, ak posledná nameraná hodnota bude rovná alebo menšia ako 0.

Tabuľka 48: Konfigurácia Neuložená konfigurácia – Trigger (Zdroj: Vlastné spracovanie)

Severity	Information
Tagy	scope: notice
Expression	count(/Cisco SW Inventory/run.config.status,#6,"gt",0)=6
Recovery	last(/Cisco SW Inventory/run.config.status,#1)<=0



Obrázok 51: Neuložená konfigurácia – Trigger (Zdroj: Vlastné spracovanie)

Template Cisco Interfaces

Jedná sa o template obsahujúci itemy pre sledovanie záťaže a chybovosti portov sieťového zariadenia.

Cisco Interfaces – Discovery rules

- **Kontrola LACP liniek**

Jedná sa o Discovery rule, ktorý pomocou SNMP OID z MIB *IEEE8023-LAG-MIB*, vykoná SNMP walk, z ktorého zistí agregačné skupiny a linky, ktoré k ním patria. Popritom sa vykonajú ďalšie SNMP walky pre zistenie popiskov a názvov portov, ktoré sa v itemoch následne priradia príslušným členom agregačných skupín. LLD macro **IFNAME** obsahuje názov portu (Gi1/0/1) a macro **IFALIAS** obsahuje popis linky.

Tabuľka 49: Konfigurácia Kontrola LACP liniek – Discovery rule (Zdroj: Vlastné spracovanie)

Key	lacp.discovery
SNMP oid	discovery[{#AGGRPORTENTRY},1.2.840.10006.300.43.1.2.1.1.4,{#IFNAME},1.3.6.1.2.1.31.1.1.1.1,{#IFALIAS},1.3.6.1.2.1.31.1.1.1.18]
Interval	1 deň

Discovery items

- **Portchannel Po{#AGGRPORTENTRY}: {#IFNAME} admin status**

Monitoruje administratívny stav sledovaného stack interfacu pomocou SNMP oid z MIB *LBMSH-MIB*, ktorý môže nadobúdať 3 SNMP hodnoty:

1. *Up*
2. *Down*
3. *Testing*

Tabuľka 50: Konfigurácia Portchannel Interface admin status – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	description: {#IFALIAS} interface: {#IFNAME} portchannel: Po{#AGGRPORTENTRY}
Key	lacp.status[ifAdminStatus.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.2.2.1.7.{#SNMPINDEX}
Interval	5 minút

- **Portchannel Po{#AGGRPORTENTRY}: {#IFNAME} operational status**

Monitoruje funkčný stav sledovaného stack interfacu pomocou SNMP oid z MIB *LBMSH-MIB*, ktorý môže nadobúdať 3 SNMP hodnoty:

1. *Up*
2. *Down*
3. *Testing*

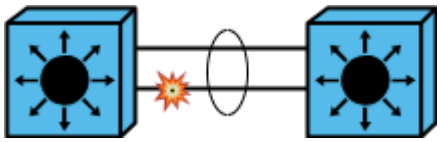
Tabuľka 51: Konfigurácia Portchannel Interface operational status – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	description: {#IFALIAS} interface: {#IFNAME} portchannel: Po{#AGGRPORTENTRY}
Key	lacp.status[ifOperStatus.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.2.2.1.8.{#SNMPINDEX}
Interval	5 minút

Discovery triggers

- **Portchannel Po{#AGGRPORTENTRY} {#IFNAME}: Link down**

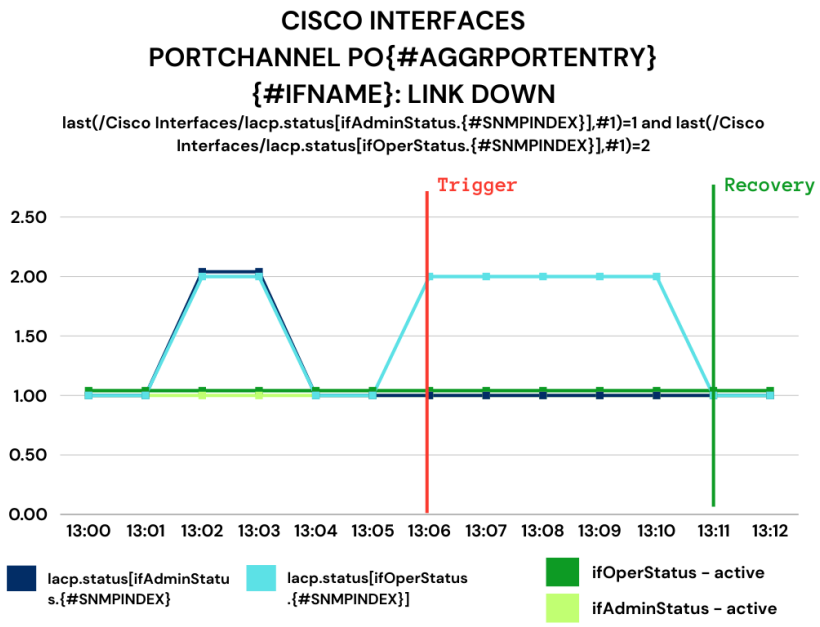
Trigger sa spustí v momente, kedy posledná hodnota funkčného stavu jednej z liniek agregačnej skupiny je rovná **2** (Down) a administratívny status konkrétnej linky je rovný **1** (Up). Trigger sa automatický vyrieši, ak posledná hodnota administratívneho statusu linky je rovná **2** (Down) – pre prípad, že sieťový technik zabudne vypnúť agregačnú linku pri sieťovom zásahu – alebo ak sa linka prepne späť do funkčného stavu s hodnotou **1** (Up).



Obrázok 52: Výpadok agregačnej linky (Zdroj: Vlastné spracovanie)

Tabuľka 52: Konfigurácia Portchannel Interface Link down – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Average
Tagy	scope: availability
Expression	last(/Cisco Interfaces/lacp.status[ifAdminStatus.{#SNMPINDEX}],#1)=1 and last(/Cisco Interfaces/lacp.status[ifOperStatus.{#SNMPINDEX}],#1)=2
Recovery	last(/Cisco Interfaces/lacp.status[ifAdminStatus.{#SNMPINDEX}],#1)=2 or last(/Cisco Interfaces/lacp.status[ifOperStatus.{#SNMPINDEX}],#1)=1



Obrázok 53: Portchannel Interface Link down – Discovery trigger (Zdroj: Vlastné spracovanie)

- **Kontrola chybovosti liniek**

Jedná sa o Discovery rule, ktorý pomocou SNMP OID z MIB *IF-MIB*, vykoná SNMP walky, z ktorých zistí SNMP Indexy všetkých portov a k ním ich popisky a názvy.

Tabuľka 53: Konfigurácia Kontrola chybovosti liniek – Discovery rule (Zdroj: Vlastné spracovanie)

Key	net.if.status.discovery
SNMP oid	discovery[{#IFALIAS},1.3.6.1.2.1.31.1.1.1.18,{#IFNAME},1.3.6.1.2.1.31.1.1.1.1]
Interval	1 deň

Discovery items

- **Interface {#IFNAME}({#IFALIAS}) err-disabled status**

Monitoruje výskyt chybných stavov, ktoré môžu linku prepnúť do stavu err-disabled, čo znamená, že systém zariadenia danú linku vypne do doby odstránenia chybného stavu. Chybné stavy sa monitorujú pomocou SNMP oid z MIB *CISCO-ERR-DISABLE-MIB*, ktorý môže nadobúdať 61 SNMP hodnôt (literárny zdroj č. **20**). Stav linky bez chybného stavu sa reprezentuje pod číslom **0**.

Tabuľka 54: Konfigurácia Interface err-disabled status – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	description: {#IFALIAS} interface: {#IFNAME}
Key	net.if.error[{#SNMPINDEX}]
SNMP oid	1.3.6.1.4.1.9.9.548.1.3.1.1.2.{#SNMPINDEX}.0
Interval	5 minút

- **Interface {#IFNAME}({#IFALIAS}): Prichádzajúce pakety s chybami**

Monitoruje počet prichádzajúcich paketov do interfacu, ktoré obsahovali chybu, pomocou SNMP oid z MIB *LBMSH-MIB.*

Tabuľka 55: Konfigurácia Interface prichádzajúce pakety s chybami – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	description: {#IFALIAS} interface: {#IFNAME}
Key	net.if.in.errors[ifInErrors.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.2.2.1.14.{#SNMPINDEX}
Interval	5 minút
Preprocessing	Change: Change per second

- **Interface {#IFNAME}({#IFALIAS}): Odchádzajúce pakety s chybami**

Monitoruje počet prichádzajúcich paketov do interfacu, ktoré obsahovali chybu, pomocou SNMP oid z MIB *LBMSH-MIB.*

Tabuľka 56: Konfigurácia Interface Odchádzajúce pakety s chybami – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	description: {#IFALIAS} interface: {#IFNAME}
Key	net.if.out.errors[ifOutErrors.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.2.2.1.20.{#SNMPINDEX}
Interval	5 minút
Preprocessing	Change: Change per second

- **Interface {#IFNAME}({#IFALIAS}): Operational status**

Monitoruje funkčný status interfacu pomocou SNMP oid z MIB *LBMSH-MIB.*

Tabuľka 57: Konfigurácia Interface Operational status – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	description: {#IFALIAS} interface: {#IFNAME}
Key	net.if.status[ifOperStatus.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.2.2.1.8.{#SNMPINDEX}
Interval	5 minút

- **Interface {#IFNAME}({#IFALIAS}): Admin status**

Monitoruje admin status interfacu pomocou SNMP oid z MIB *LBMSH-MIB.*

Tabuľka 58: Konfigurácia Interface Admin status – Discovery item (Zdroj: Vlastné spracovanie)

Tagy	description: {#IFALIAS} interface: {#IFNAME}
Key	net.if.status[ifAdminStatus.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.2.2.1.7.{#SNMPINDEX}
Interval	5 minút

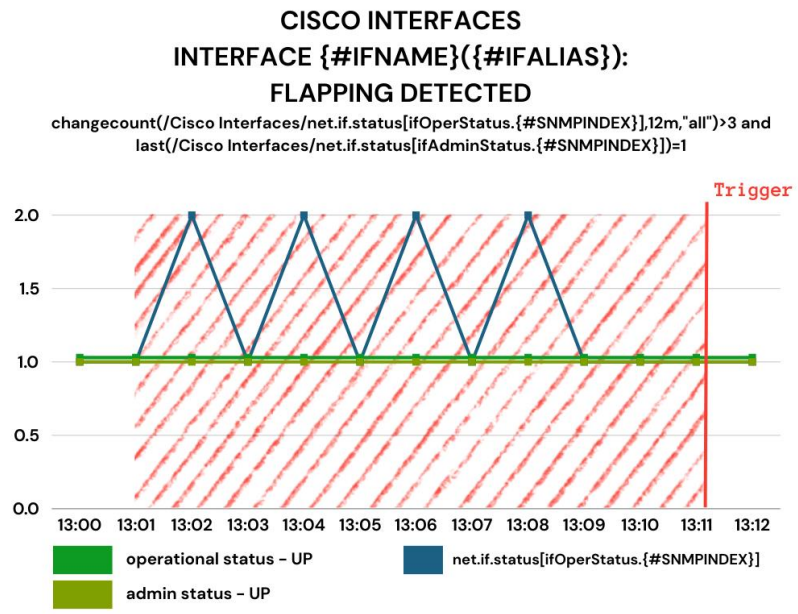
Discovery triggers

- **Interface {#IFNAME}({#IFALIAS}): Detekovaný flapping**

Trigger sa spustí vo chvíli, kedy sa funkčný stav portu zmenil v priebehu 12 minút viac ako 3x a jeho admin status bol počas merania Up (v prípade, že sieťový správca linku testuje pomocou vypínania a zapínania portu cez príkaz *shutdown*, trigger sa nespustí). Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí skontrolovať, čo sa s linkou v skutočnosti deje.

Tabuľka 59: Konfigurácia Interface Detekovaný flapping – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: performance scope: availability
Expression	changecount(/Cisco Interfaces/net.if.status[ifOperStatus.{#SNMPINDEX}],12m,"all")>3 and last(/Cisco Interfaces/net.if.status[ifAdminStatus.{#SNMPINDEX}])=1
Recovery	None



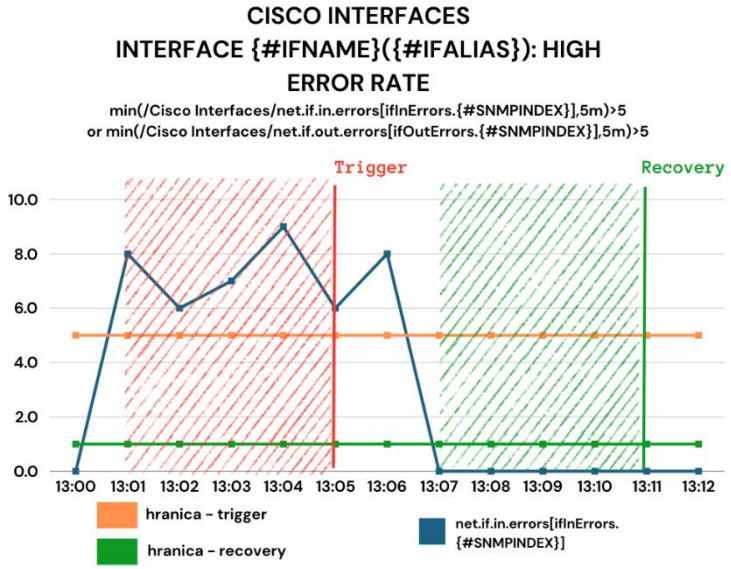
Obrázok 54: Interface Detekovaný flapping – Discovery trigger (Zdroj: Vlastné spracovanie)

- **Interface {#IFNAME}({#IFALIAS}): Vysoká miera chybovosti**

Trigger sa spustí vo chvíli, kedy minimálny počet prichádzajúcich alebo odchádzajúcich paketov s chybami bol za posledných 5 minút väčší ako 5. Trigger sa automaticky vyrieši, ak za posledných 5 minút bola maximálny počet paketov s chybami menší ako 1.

Tabuľka 60: Konfigurácia Interface Vysoká miera chybovosti – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: performance scope: availability
Expression	min(/Cisco Interfaces/net.if.in.errors[ifInErrors.{#SNMPINDEX}],5m)>5 or min(/Cisco Interfaces/net.if.out.errors[ifOutErrors.{#SNMPINDEX}],5m)>5
Recovery	max(/Cisco Interfaces/net.if.in.errors[ifInErrors.{#SNMPINDEX}],5m)<1 or max(/Cisco Interfaces/net.if.out.errors[ifOutErrors.{#SNMPINDEX}],5m)<1



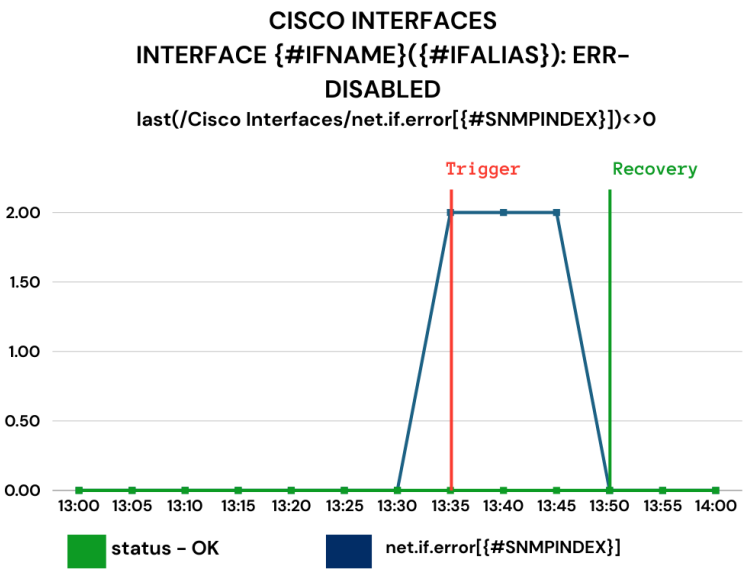
Obrázok 55: Interface Vysoká miera chybovosti – Discovery trigger (Zdroj: Vlastné spracovanie)

- **Interface {#IFNAME}({#IFALIAS}): Err-disabled**

Trigger sa spustí vo chvíli, kedy sa na linke vyskytne chybový stav (napr. BPDU paket príde na port so zapnutým BPDU guardom), čo znamená, že sa na linke objaví iná hodnota ako **0** (funkčný stav), ktorú Zabbix zaznamená ako poslednú nameranú. Trigger sa automaticky vyrieši, ak sa posledná hodnota rovná **0**.

Tabuľka 61: Konfigurácia Interface Err-disabled – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: availability
Expression	last(/Cisco Interfaces/net.if.error[{#SNMPINDEX}])<>0
Recovery	last(/Cisco Interfaces/net.if.error[{#SNMPINDEX}],#1)=0



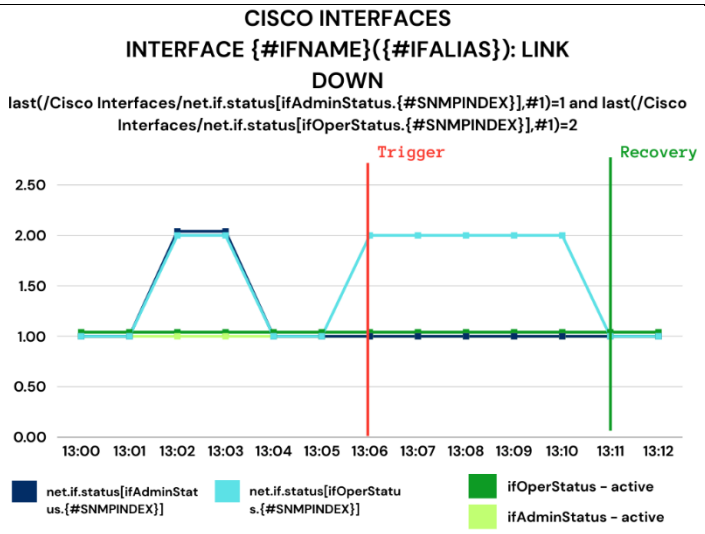
Obrázok 56: Interface Err-disabled – Discovery trigger (Zdroj: Vlastné spracovanie)

- **Interface {#IFNAME}({#IFALIAS}): Nefunkčná linka**

Trigger sa spustí vo chvíli, kedy je posledná hodnota admin statusu linky rovná hodnote **1** (Up) a funkčný stav linky je rovný hodnote **2** (Down). Trigger sa automaticky vyrieši, ak sa posledná hodnota funkčného stavu rovná **0**.

Tabuľka 62: Konfigurácia Interface Nefunkčná linka – Discovery trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: availability
Expression	last(/Cisco Interfaces/net.if.status[ifAdminStatus.{#SNMPINDEX}],#1)=1 and last(/Cisco Interfaces/net.if.status[ifOperStatus.{#SNMPINDEX}],#1)=2
Recovery	last(/Cisco Interfaces/net.if.status[ifOperStatus.{#SNMPINDEX}],#2)<>2
Dependency	Cisco Interfaces: Interface {#IFNAME}({#IFALIAS}): Flapping Detected



Obrázok 57: Interface Nefunkčná linka – Discovery trigger (Zdroj: Vlastné spracovanie)

Template Fortigate

Jedná sa o template, ktorý obsahuje itemy pre sledovanie Fortigate firewallu ako virtuálneho zariadenia.

Fortigate – Items

- **CPU záťaž**

Monitoruje poslednú nameranú hodnotu aktuálneho využitia CPU v percentách pomocou SNMP oid z MIB *FORTINET-FORTIGATE-MIB*.

Tabuľka 63: Konfigurácia CPU záťaž – Item (Zdroj: Vlastné spracovanie)

Tagy	component: cpu
Key	cpu.usage
SNMP oid	1.3.6.1.4.1.12356.101.4.1.3.0
Interval	5 minút

- **Využitie pamäte**

Monitoruje poslednú nameranú hodnotu aktuálneho využitia pamäte v percentách pomocou SNMP oid z MIB *FORTINET-FORTIGATE-MIB*.

Tabuľka 64: Konfigurácia Využitie pamäte – Item (Zdroj: Vlastné spracovanie)

Tagy	component: memory
Key	memory.usage
SNMP oid	1.3.6.1.4.1.12356.101.4.1.4.0
Interval	5 minút

- **Počet relácii**

Monitoruje poslednú nameranú hodnotu počtu relácii v globálnej VDOM pomocou SNMP oid z MIB *FORTINET-FORTIGATE-MIB*.

Tabuľka 65: Konfigurácia Počet relácii – Item (Zdroj: Vlastné spracovanie)

Tagy	system: sessions
Key	session.count
SNMP oid	1.3.6.1.4.1.12356.101.4.1.8.0
Interval	5 minút

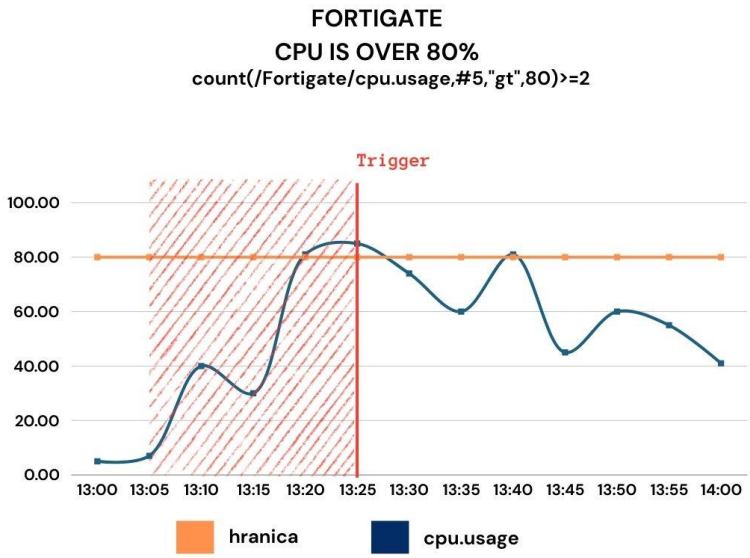
Fortigate – Triggers

- **CPU využitie je nad 80%**

Trigger sa spustí v prípade, kedy za posledných 5 nameraných hodnôt bolo využitie CPU pri aspoň dvoch hodnotách väčšie ako 80%. Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí zistiť, ktorý proces Fortigatu zaťažil CPU a optimalizovať ho.

Tabuľka 66: Konfigurácia CPU využitie je nad 80% – Trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: performance
Expression	count(/Fortigate/cpu.usage,#5,"gt",80)=2
Recovery	None



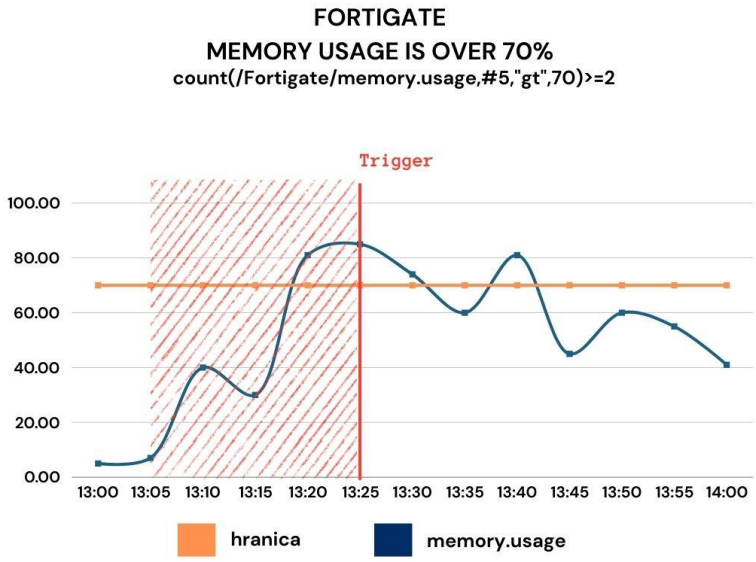
Obrázok 58: CPU využitie je nad 80% – Trigger (Zdroj: Vlastné spracovanie)

- **Využitie pamäte je nad 70%**

Trigger sa spustí v momente, kedy za posledných 5 nameraných hodnôt bolo využitie pamäte pri aspoň dvoch hodnotách väčšie ako 70%. Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí zistiť, ktorý proces Fortigatu zaťažil pamäť a optimalizovať ho.

Tabuľka 67: Konfigurácia Využitie pamäte je nad 70% – Trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: performance
Expression	count(/Fortigate/memory.usage,#5,"gt",70)=2
Recovery	None



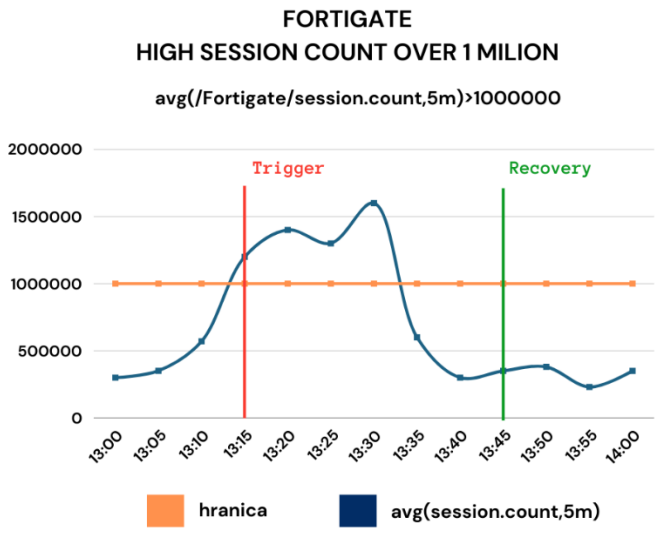
Obrázok 59: Využitie pamäte je nad 70% – Trigger (Zdroj: Vlastné spracovanie)

- **Počet relácii presiahol 1 milión**

Trigger sa spustí vo chvíli, kedy za posledných 5 minút bol priemerný počet relácii v global VDOM vyšší ako 1 milión. Trigger sa automatický zatvorí, ak sa za posledných 15 minút zmenšil priemerný počet relácii pod 1 milión.

Tabuľka 68: Konfigurácia Počet relácii presiahol 1 milión – Trigger (Zdroj: Vlastné spracovanie)

Severity	Warning
Tagy	scope: performance
Expression	avg(/Fortigate/session.count,5m)>1000000
Recovery	avg(/Fortigate/session.count,15m)<1000000



Obrázok 60: Počet relácii presiahol 1 milión – Trigger (Zdroj: Vlastné spracovanie)

Fortigate – Discovery rules

1. Interfaces

Jedná sa o Discovery rule, ktorý pomocou SNMP OID z MIB *IF-MIB*, vykoná SNMP walky, z ktorých zistí SNMP Indexy všetkých portov a k ním ich popisky a názvy.

Tabuľka 69: Konfigurácia Interfaces – Discovery rule (Zdroj: Vlastné spracovanie)

Key	lacp.discovery
SNMP oid	discovery[{#IFDESC},1.3.6.1.2.1.2.2.1.2,{#IFALIAS},1.3.6.1.2.1.31.1.1.1.1]
Interval	1 deň

Discovery items

- **Interface ({#IFALIAS}): Prijaté bity**

Monitoruje množstvo prijatých bitov v bps (bits per second) do interfacu pomocou SNMP oid z MIB *IF-MIB*.

Tabuľka 70: Konfigurácia Interface prijaté bity – Item (Zdroj: Vlastné spracovanie)

Tagy	interface: {#IFALIAS}
Key	fgt.int.in[ifHCInOctets.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.31.1.1.1.6.{#SNMPINDEX}
Interval	1 minúta
Preprocessing	Change per second --> Custom multiplier 8

- **Interface ({#IFALIAS}): Odoslané bity**

Monitoruje množstvo odoslaných bitov v bps (bits per second) z interfacu pomocou SNMP oid z MIB *IF-MIB.*

Tabuľka 71: Konfigurácia Interface odoslané bity – Item (Zdroj: Vlastné spracovanie)

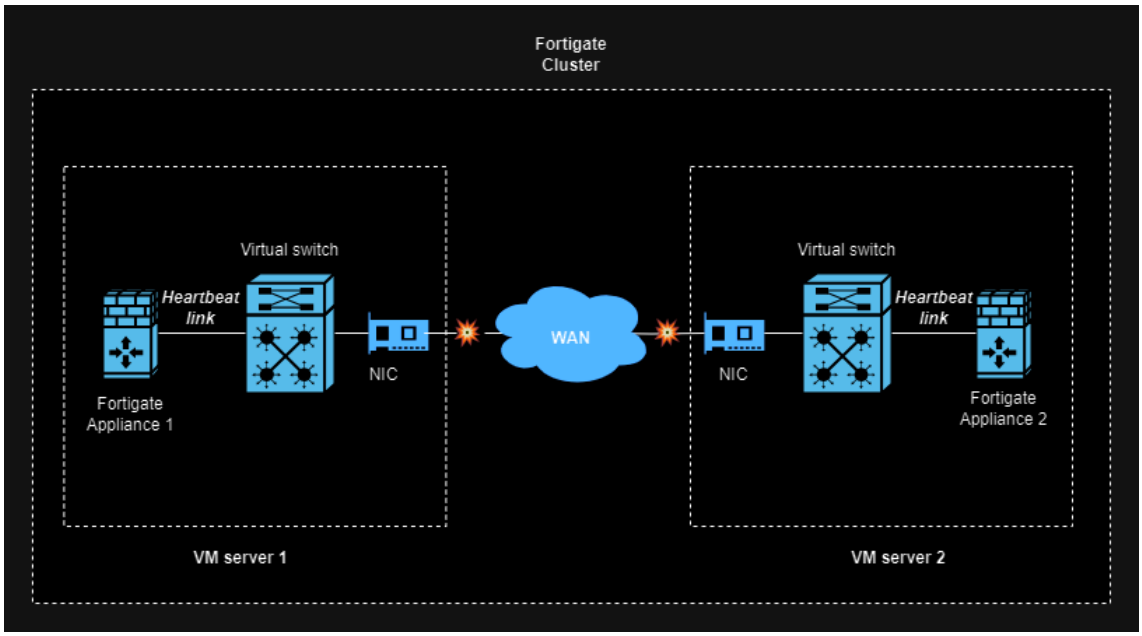
Tagy	interface: {#IFALIAS}
Key	fgt.int.out[ifHCOutOctets.{#SNMPINDEX}]
SNMP oid	1.3.6.1.2.1.31.1.1.1.10.{#SNMPINDEX}
Interval	1 minúta
Preprocessing	Change per second --> Custom multiplier 8

Discovery triggers

- **Druhý člen clusteru je offline**

Trigger sa spustí v momente, kedy na heartbeat linke nie je v poslednej nameranej hodnote žiaden prijatý alebo odoslaný bit, čo znamená, že druhý člen clusteru je offline. Trigger rozpozná zo všetkých portov z SNMP walku heartbeat linku pomocou Context makra (konfigurované v rámci templatu), do ktorého dosadí popis linky. Ak sa v popise linky nachádza „*HA interface*“, dosadí sa do makra hodnota **1** (tento popis linky sa do heartbeat linky na Fortigate nedoplňuje automaticky, **je nutné ho doplniť ručne**). Pre tento prípad nie je možné implementovať jednoduchú kontrolu stavu linky (Up alebo Down), keďže je virtuálny Fortigate pripojený do virtuálneho switchu na VM servery

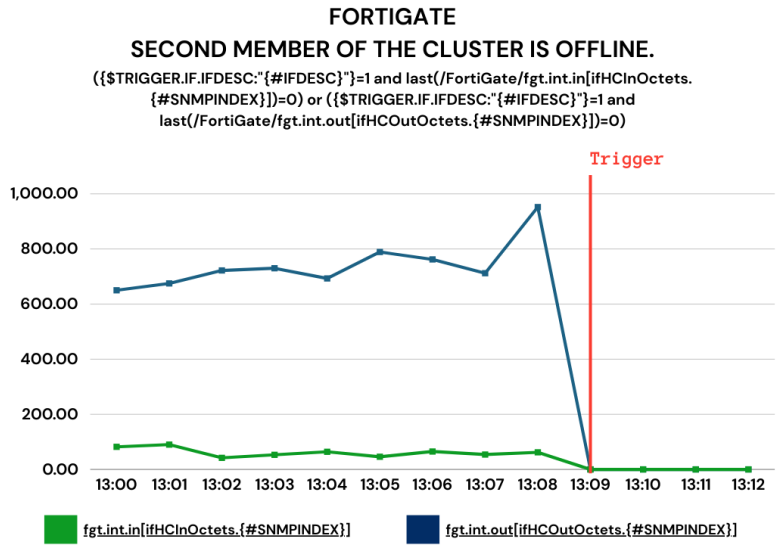
a pri výpadku druhého člena clusteru alebo celej heartbeat VLAN by stav linky ukazoval **Up**, keďže linka do virtuálnom switchu je stále **Up**. Trigger sa musí manuálne zatvoriť užívateľom, ktorý musí skontrolovať, čo sa s heartbeat linkou stalo a zistiť, či nenastal stav *split-brain*.



Obrázok 61: Výpadok heartbeat linky Fortigate clusteru (Zdroj: Vlastné spracovanie)

Tabuľka 72: Konfigurácia Druhý člen clusteru je offline – Trigger (Zdroj: Vlastné spracovanie)

Severity	High
Tagy	scope: performance
Expression	({$TRIGGER.IF.IFDESC:"{#IFDESC}"}=1 and last(/FortiGate/fgt.int.in[ifHCInOctets.{#SNMPINDEX}])=0) or ({$TRIGGER.IF.IFDESC:"{#IFDESC}"}=1 and last(/FortiGate/fgt.int.out[ifHCOutOctets.{#SNMPINDEX}])=0)
Recovery	None
Context macro	{$TRIGGER.IF.DESC} = 0 {$TRIGGER.IF.DESC:"HA interface"} = 1



Obrázok 62: Druhý člen clusteru je offline – Discovery trigger (Zdroj: Vlastné spracovanie)

3.3.4 Práva

Oprávnenia v Zabbixe závisia od typu užívateľa (*User type*), prispôsobených rolí užívateľa (*User roles*) a prístupu k hostom alebo triggerom, ktorý je určený na základe skupiny užívateľov (*User groups*).

Typy užívateľov sa delia na 3 typy:

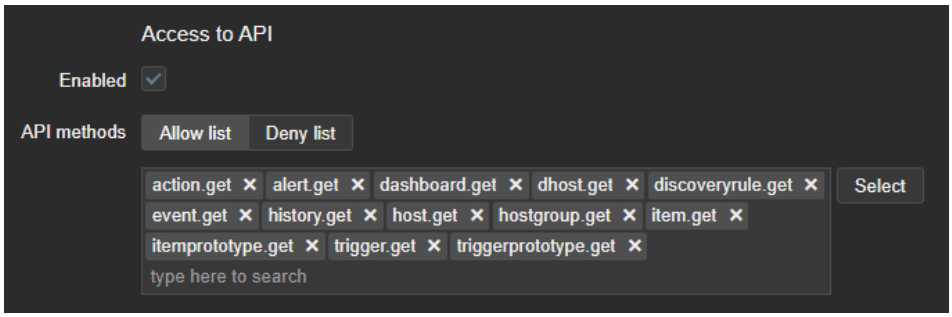
1. **User** – má obmedzené prístupové práva k špecifickým sekciám menu (Monitoring, Services, Inventory a Reports). Tento typ sa bude prideľovať všetkým zamestnancom helpdesku a fakultných IT oddelení, ktorí si požiadajú o prístup do Zabbixu.
2. **Admin** – má neúplné prístupové práva k sekciám menu (Monitoring, Services, Inventory, Reports, Data collection, Alerts). Admin bude prideľovaný zamestnancom divízie IT infraštruktúry Ústavu informačných služieb a výpočtovej techniky, ktorí sa budú starať o implementáciu nových sónd a ich údržbu a budú zároveň zahrnutí do monitorovania sieťovej infraštruktúry..
3. **Super Admin** – má prístup ku všetkým sekciám menu. Tento typ je vyhradený pre zamestnancov Ústavu informačných služieb a výpočtovej techniky, ktorí majú zodpovednosť za kontrolu správneho fungovania Zabbixu, vrátane konfigurácie

sond, kontroly užívateľov s typom Admin a správy užívateľov, typov médií pre notifikácie, skriptov a globálnych makier.

Typy užívateľov sa následne aplikujú užívateľským roliam v sekcii menu *User roles*, v ktorých sa dajú práva k sekciám menu v rámci typu užívateľa odoberať (nedajú sa pridávať menu sekcie, ktoré nie sú zahrnuté v type užívateľa). V rámci užívateľskej role sa nastavujú prístupy pre API a taktiež prístup k novým menu sekciám pri aktualizáciách Zabbixu. Nakonfigurované užívateľské role sa potom prideľujú konkrétnym užívateľom.

Pre univerzitné riešenie budú existovať 4 typy užívateľských rolí:

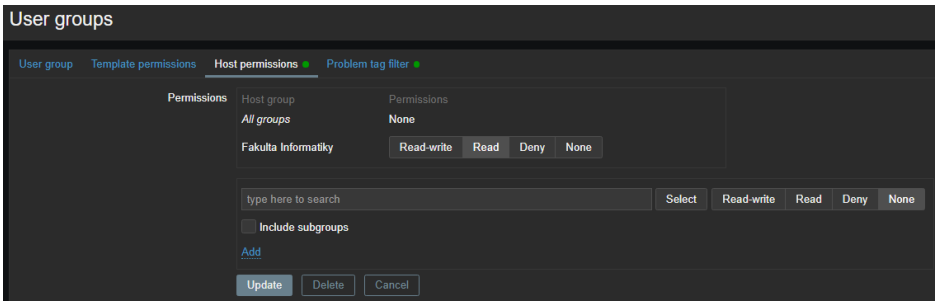
- **Fakulta** – bude aplikovaná zamestnancom fakultných IT oddelení
  - User type: User
  - Obmedzené API metódy viz **obr. 63**
  - Zákaz k novým menu sekciám
- **Helpdesk** – bude aplikovaná zamestnancom helpdesku
  - User type: User
  - Obmedzené API metódy viz **obr. 63**
  - Zákaz k novým menu sekciám
- **UISVT-Admin** – bude aplikovaná zamestnancom UISVT, ktorí sa budú starať o sondy a monitorovať sieť
  - User type: Admin
  - Neobmedzený prístup k API metódam (okrem vytvárania, aktualizovania a mazania hostov – k tomu je určený Zabbix provisioning viz *kapitola 3.3.2*)
  - Povolenie k novým menu sekciám
- **UISVT-Superadmin** – bude aplikovaná zamestnancom UISVT, ktorí sa budú starať o kontrolu správnej funkčnosti Zabbixu
  - User type: Super Admin
  - Neobmedzený prístup k API metódam
  - Povolenie k novým menu sekciám



Obrázok 63: Povolené API metódy (Zdroj: Vlastné spracovanie)

Obrázok 64: Užívateľská rola Fakulta (Zdroj: Vlastné spracovanie)

Prístupy k zoznamu hostov a triggerom sa nastavujú v skupinách užívateľov (*User groups*), v ktorých sa nastavuje, akú skupinu hostov (*Host group*) a aké triggery dokáže daná skupina vidieť. Tento obmedzený pohľad zabezpečuje, že zamestnanci majú prístup len k informáciám a funkciám, ktoré sú pre ich prácu relevantné. Nakonfigurované užívateľské skupiny sa potom prideľujú konkrétnym užívateľom.

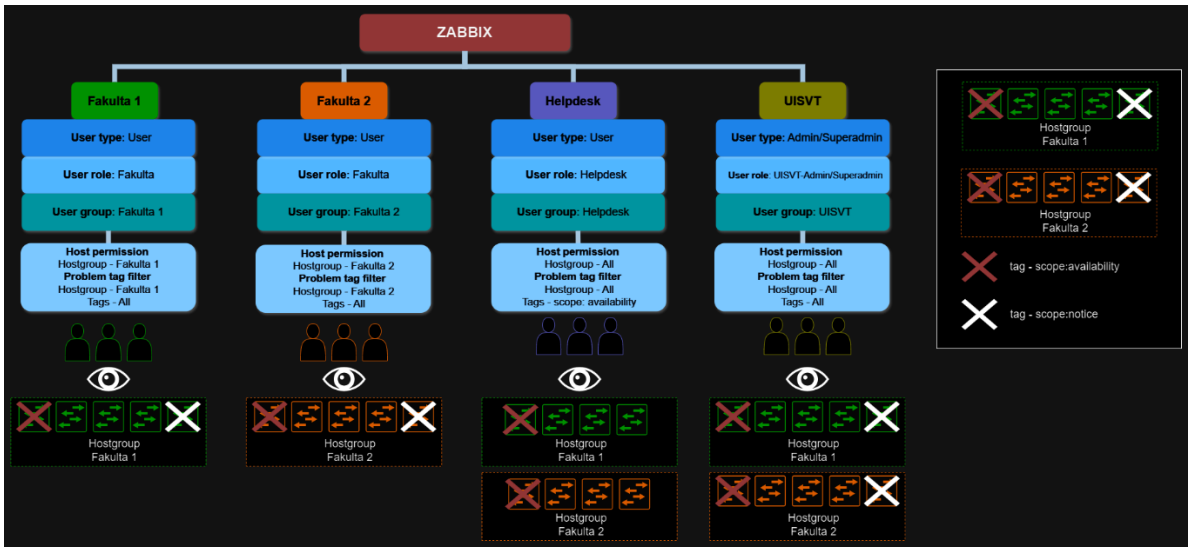


Obrázok 65: Príklad povolenia hostov pre užívateľskú skupinu (Zdroj: Vlastné spracovanie)

Každá fakulta bude mať svoju vlastnú užívateľskú skupinu, ktorá bude mať právo vidieť iba svoje skupiny hostov a k ním všetky ich triggery. Helpdesk bude vlastniť svoju užívateľskú skupinu, ktorá bude mať právo vidieť všetky skupiny hostov, avšak bude mať prístup iba k triggerom, ktoré vlastnia tag **scope: availability** viz *kapitola 3.3.3*. Helpdesk nepotrebuje vedieť o triggeroch, ktoré majú napríklad tag **scope: notice** (Triggery: Názov systému bol zmenený alebo Zariadenie bolo vymenené). Ústav informačných služieb a výpočtovej techniky (UISVT) bude vlastniť užívateľskú skupinu zvlášť pre Adminov a Superadminov (konkrétne divízia IT infraštruktúry), ktorí budú môcť vidieť všetky skupiny hostov a všetky typy triggerov.

Tabuľka 73: Rozdelenie trigger tagov (Zdroj: Vlastné spracovanie)

	Trigger tagy				
Užívateľské skupiny	*Notice*	*Security*	*Availability*	*Performance*	*L2/L3*
Fakulta	X	X	X	X	X
Helpdesk			X		
UISVT	X	X	X	X	X



Obrázok 66: Vizualizácia práv užívateľských skupín (Zdroj: Vlastné spracovanie)

3.3.5 Notifikácie

Notifikácie predstavujú dôležitú súčasť správy a sledovania stavu zariadení a služieb. Ich cieľom je informovať správcov a zodpovedné osoby o významných udalostiach a problémoch v infraštruktúre v reálnom čase. Notifikácie pre triggery (*Trigger actions*) sa v Zabbixe nastavujú pre udalosti, keď sa stav triggeru zmení zo statusu ***OK*** na ***PROBLÉM*** a opačne. Pre každú inštanciu trigger notifikácii je nutné nastaviť:

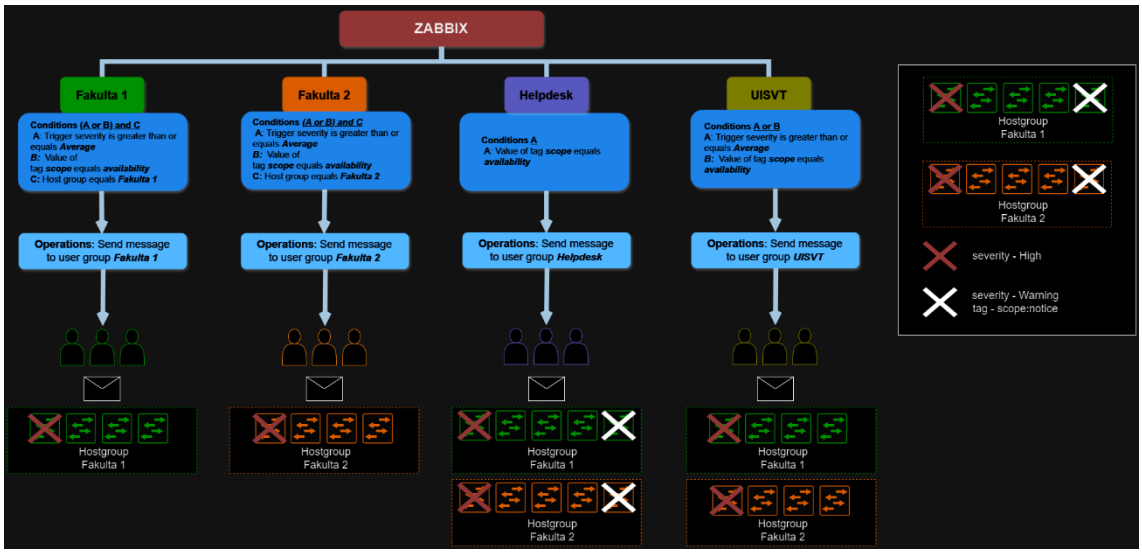
1. **Akcia** – definovanie podmienok, ktoré je nutné splniť pre odoslanie notifikácie. Pri implementácii notifikácií je kľúčové vybrať vhodné podmienky, pretože nie vždy je potrebné odosielať notifikácie pre každý trigger. Nesprávne nastavené notifikácie môžu spôsobiť spamovanie zainteresovaných strán, čo vedie k zvýšenej kognitívnej záťaži.
2. **Operácia** – nastavenie odosielania notifikácii definovaným užívateľom/užívateľským skupinám
3. **Typ média** – nastavenie doručovacieho kanálu používaného na odosielanie oznámení a upozornení zo Zabbixu

Pre univerzitné riešenie budú existovať pre každú fakultu, helpdesk a UISVT samostatné inštancie trigger notifikácii:

- **Fakulta**
    - **Akcia**: *(A or B) and C*

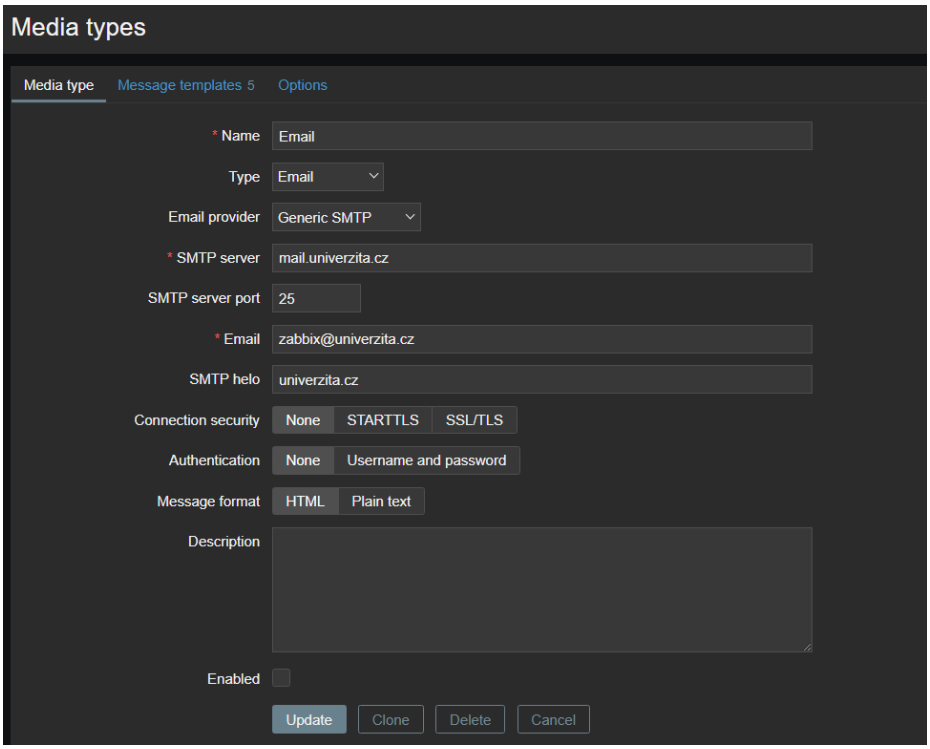
        - **A –** závažnosť triggeru je väčšia ako ***Average***
        - **B** – hodnota trigger tagu ***scope*** má hodnotu ***availability***
        - **C –** skupina hostov (*Host group*) sa zhoduje s názvom fakulty
    - **Operácia:** Odoslanie správ skupine užívateľov (*User group*), ktorá sa zhoduje s názvom fakulty
    - **Typ média:** Email
- **Helpdesk**
    - **Akcia**: *A*
        - **A** – hodnota trigger tagu ***scope*** má hodnotu ***availability***
    - **Operácia:** Odoslanie správ skupine užívateľov (*User group*) Helpdesku
    - **Typ média:** Email
- **UISVT**
    - **Akcia**: *A or B*
        - **A** – hodnota trigger tagu ***scope*** má hodnotu ***availability***
        - **B –** závažnosť triggeru je väčšia ako ***Average***
    - **Operácia:** Odoslanie správ skupine užívateľov (*User group*) Ústavu informačných služieb a výpočtovej techniky
    - **Typ média:** Email

Je nevyhnutné, aby všetky užívateľské skupiny dostávali notifikácie o udalostiach, ktoré upozorňujú na dostupnosť na samotné zariadenie. Vzhľadom na to, že fakulty a UISVT majú práva vidieť všetky triggery, je potrebné obmedziť notifikácie len na udalosti s vážnosťou vyššou ako je *Average*, čo naznačuje, že problém vyžaduje pozornosť a rýchle riešenie. Toto opatrenie pomôže minimalizovať spamovanie a zabezpečí, že notifikácie budú zaslané len v prípadoch, keď je to skutočne potrebné.



Obrázok 67: Vizualizácia notifikácii užívateľských skupín (Zdroj: Vlastné spracovanie)

Pre všetky notifikácie bude použitá Emailová forma, ktorá je na univerzite považovaná za oficiálny informačný kanál. Je dôležité, aby všetci užívatelia mali vyplnený email vo svojom užívateľskom profile. Príkladové nastavenie typu média *Email* sa nachádza na **obr. 68**



Obrázok 68: Nastavenie typu média Email (Zdroj: Vlastné spracovanie)

ZÁVER

Cieľom tejto diplomovej práce bol návrh a implementácia monitorovacieho systému pre univerzitné prostredie s dôrazom na sieťové prvky od vendora Cisco.

V rámci tejto práce som sa zameriaval na výber a implementáciu monitorovacieho systému do prostredia univerzity s cieľom poskytnúť efektívne nástroje na monitorovanie a správu siete a sieťových zariadení. Okrem technickej stránky implementácie som identifikoval kľúčové hrozby, ktoré môžu ovplyvniť prevádzku siete, a nastavil monitorovací systém tak, aby bol schopný detegovať a reagovať na potenciálne problémy v sieti včas a efektívne.

V úvodnej časti práce som preskúmal súčasný stav vybranej univerzity a jej požiadavky na monitorovanie siete, následne som sa zameriaval na teoretické východiská. V implementačnej časti som podrobne popísal import dát zo systému NetBox do monitorovacieho systému Zabbix, konfiguráciu monitorovacích sond, notifikačných mechanizmov a priradenie práv pre užívateľské skupiny, čím som vytvoril robustný a efektívny monitorovací systém pre univerzitné prostredie.

ZOZNAM POUŽITEJ LITERATÚRY

(1) DONAHUE, Gary A. Kompletní průvodce síťového experta. Brno: Computer Press, 2009, 528 s. : il. ISBN 978-80-251-2247-1.

(2) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009, 496 s. : il. ISBN 978-80-247-2615-1.

(3) KRETCHMAR, James M a Libor DOSTÁLEK. Administrace a diagnostika sítí: pomocí OpenSource utilit a nástrojů. Brno: Computer Press, 2004, 216 s. : il. ISBN 80-251-0345-5.

(4) ONDRÁK, Viktor, Petr SEDLÁK a Vladimír MAZÁLEK. Problematika ISMS v manažerské informatice. Brno: CERM, 2013, 377 s. : il, grafy, tab. ISBN 978-80-7204-872-4.

(5) SOSINSKY, Barrie A. Mistrovství – počítačové sítě. Brno: Computer Press, 2010, 840 s. : il. ISBN 978-80-251-3363-7.

(6) Sada protokolů TCP/IP [online]. [cit. 2022-10-10]. Dostupné z: https://moodle.sspbrno.cz/pluginfile.php/6413/mod_resource/content/1/tcpip.pdf

(7) ONDRÁK, V. Prednášky – počítačové sítě. Brno: VUT Fakulta podnikatelská, 2022

(8) JORDÁN, Vilém. Infrastruktura komunikačních systémů II: kritické aplikace. Brno: Akademické nakladatelství CERM, 2015, 232 stran : ilustrace. ISBN 978-80-214-5240-4.

(9) SNMP Programmer's Reference - TreckWiki. [online]. [cit. 2022-15-10]. Dostupné z: https://wiki.treck.com/SNMP_Programmer's_Reference

(10) NetAcad Course UI. NetAcad Course UI [online]. [cit. 2022-15-10]. Dostupné z: https://contenthub.netacad.com/srwe/1.3.1

(11) What is ICMP [online]. [cit. 2022-17-10]. Dostupné z: https://info.support.huawei.com/info-finder/encyclopedia/en/ICMP.html

(12) Co je ISO 27001? - Standard pro bezpečnost informací | NQA. Object moved [online]. [cit. 2022-22-10].Dostupné z: https://www.nqa.com/cs-cz/certification/standards/iso-27001

(13) Doucek, P.: Bezpečnost IS/ICT a proces globální integrace – Pročbezpečnost?, In: AT&P Journal, 01/2005, ISSN 1335-2237

(14) KOCH, Miloš a Bernard NEUWIRTH. Dátové a funkční modelovánií. Vyd. 4., rozšířené. Brno : Akademické nakladatelství CERM, 2010. s. 142 s. ISBN 978-80-214- 4125-5.

(15) False Positive, False Negative, True Positive and True Negative | v500 Systems - v500 Systems. How we can unlock information in documents with AI - v500 Systems [online]. Copyright © 2022 v500systems [cit. 19.11.2022]. Dostupné z: https://www.v500.com/false-positive-false-negative-true-positive-and-true-negative/

(16) Záložní napájení | PRONIX. PRONIX | Your Power System Integrator™ [online]. Copyright © 2018 [cit. 03.12.2022]. Dostupné z: https://www.pronix.cz/zalozni-napajeni/

(17) A Comprehensive Guide to Prometheus Monitoring. Online. Dostupné z: https://www.weave.works/blog/a-comprehensive-guide-to-prometheus-monitoring. [cit. 2023-11-18].

(18) Icinga DB. Online. Dostupné z: https://icinga.com/docs/icinga-db/latest/doc/01-About/. [cit. 2023-11-29].

(19) CISCO-STACKWISE-MIB. Online. MIB files repository. 2023. Dostupné z: https://www.circitor.fr/Mibs/Html/C/CISCO-STACKWISE-MIB.php#cswSwitchState. [cit. 2024-03-15].

(20) CISCO-ERR-DISABLE-MIB. Online. MIB files repository. 2023. Dostupné z: https://www.circitor.fr/Mibs/Html/C/CISCO-ERR-DISABLE-MIB.php#CErrDisableFeatureID. [cit. 2024-03-15].

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

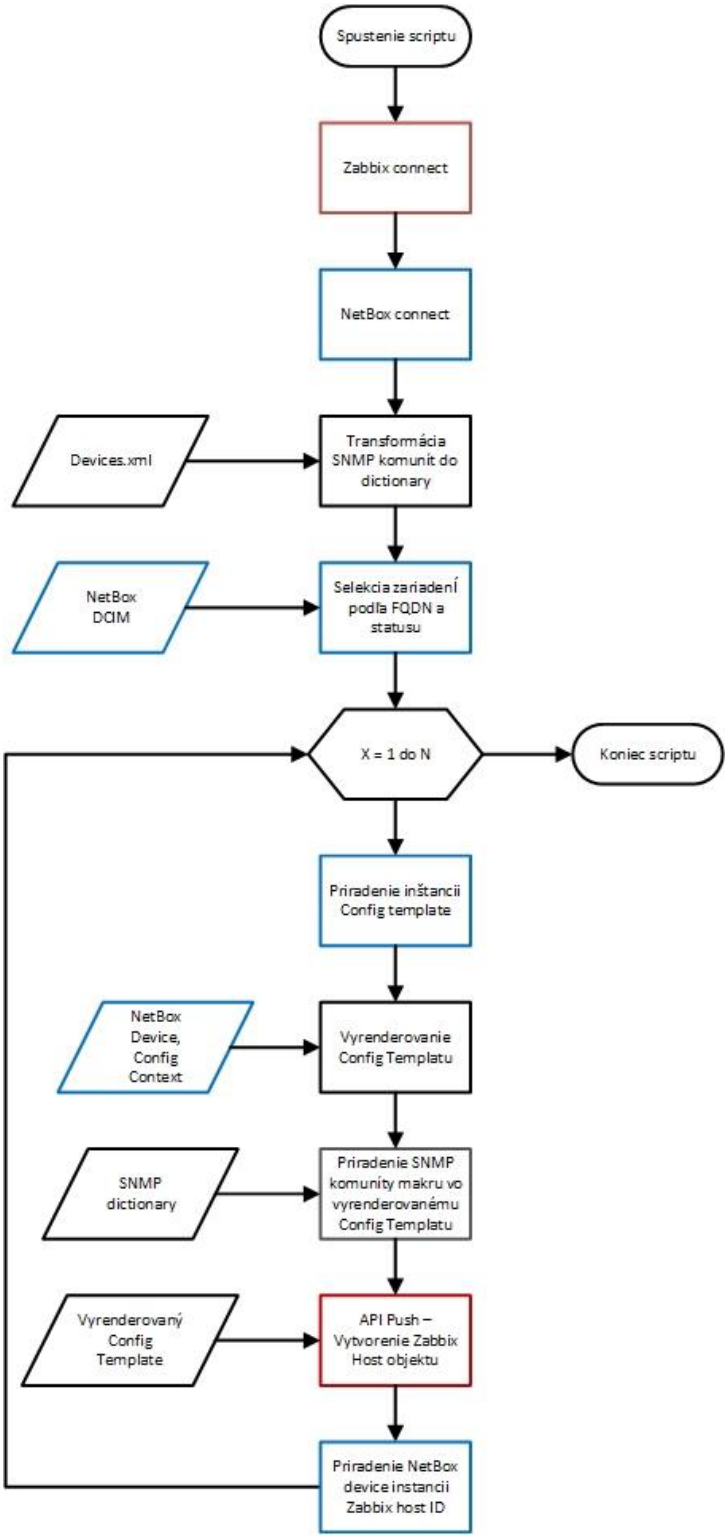
ZOZNAM POUŽITÝCH SKRATIEK

IP	Internet Protocol
SSH	Secure Shell
VLAN	Virtual Local Area Network
MAC	Media Access Control
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
BPDU	Bridge Protocol Data Units
SNMP	Simple Network Management Protocol
MIB	Management Information Bases
OID	Object Identifier
VTP	VLAN Trunking Protocol
LLD	Low-level discovery
CLI	Command Line
ICMP	Internet Control Message Protocol
HA	High availability
API	Application Programming Interface
SSID	Service Set Identifier
VM	Virtual Machine
UISVT	Ústav informačných služieb a výpočtovej techniky

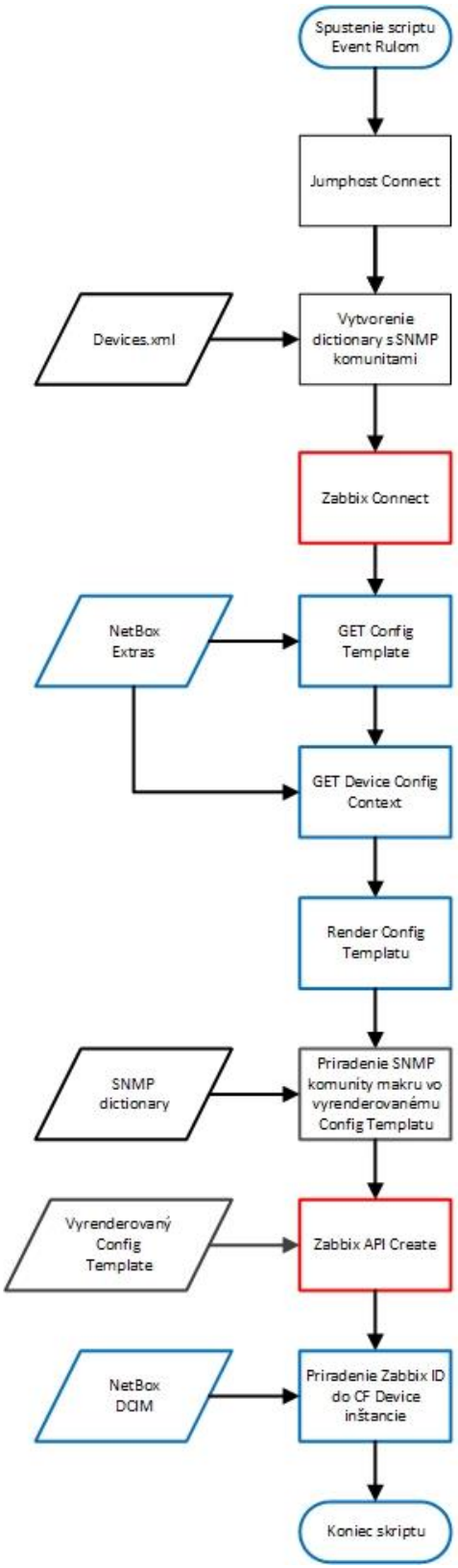
ZOZNAM PRÍLOH

PRÍLOHY

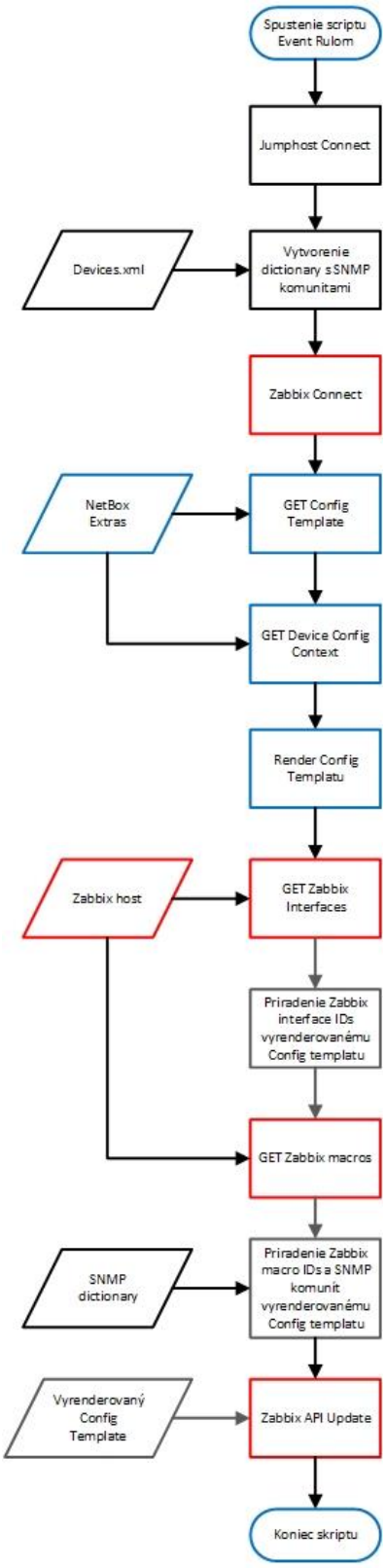
Príloha 1: Vývojový diagram – Importovací skript pre Zabbix



Príloha 2: Vývojový diagram – Skript – Zabbix Create



Príloha 3: Vývojový diagram – Skript – Zabbix Update



Príloha 4: Vývojový diagram – Skript – Zabbix Delete