**Vysoká škola ekonomická v Praze**
**Fakulta informatiky a statistiky**
**Katedra informačních technologií**

Studijní program: Aplikovaná informatika
Obor: Aplikovaná informatika

# Monitoring IT infrastruktury

**BAKALÁŘSKÁ PRÁCE**

Student : Filip Zíbar
Vedoucí : RNDr. Igor Čermák, CSc
Oponent : Ing. Aleš Kruczek, Ph.D.

**2018**

## Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne 2. května 2018 . . . . . .

Filip Zíbar

## Poděkování

Rád bych poděkoval vedoucímu práce RNDr. Igoru Čermákovi, CSc. za cenné připomínky a rady při psaní bakalářské práce. Také děkuji své rodině a přátelům za jejich neustálou podporu.

# Abstrakt

Tato práce pojednává o monitoringu IT infrastruktury a jeho dopadu na provoz služeb. Cílem práce je poskytnout komplexní pohled na problematiku monitoringu a seznámit tak čtenáře s principy, možnostmi a dostupnými nástroji. Dále je také popsán proces implementace a konfigurace nástroje Zabbix do produkčního prostředí reálné společnost. Při konfiguraci nástroje je kladen důraz na odfiltrování nerelevantních informací, vhodné nastavení prahových hodnot, triggering a alerting.

Výstupem práce je srovnání vybraných monitorovacích nástrojů a doporučený postup implementace libovolného monitorovacího nástroje. Práce proto může být využita společnostmi při výběru a implementaci monitorovacího nástroje do vlastního prostředí.

**Klíčová slova:** Monitoring, konfigurační položka, alert, trigger, Zabbix, Nagios, Solarwinds

# Abstract

This thesis focuses on the IT infrastructure monitoring and its impact on the service operation. The goal of this thesis is to provide the reader with a complex overview on the matter of monitoring. For this purpose, crucial principles, options and available tools are described first. The process of configurating Zabbix into the production environment of a real company is discussed next. Emphasis is given on filtering out irrelevant information, correctly setting up thresholds, triggers and alerts.

The output of this thesis is a comparison of selected monitoring tools and a recommended procedure for implementation any monitoring tool. The thesis can therefore be used by companies when selecting and implementing a monitoring tool into their own environment.



**Keywords:** Monitoring, configuration item, alert, trigger, Zabbix, Nagios, Solarwinds

# Obsah

1 Úvod ..... 8
2 Monitoring ..... 10
2.1 Principy monitoringu ..... 12
2.1.1 Metriky ..... 13
2.1.2 Prahové hodnoty ..... 14
2.1.3 Triggering ..... 14
2.2 Vazba monitoringu na Provoz služeb dle ITIL ..... 16
2.2.1 Event Management (Správa událostí) ..... 16
2.2.2 Incident Management (Správa incidentů) ..... 17
2.2.3 Problem Management (Správa problémů) ..... 17
2.3 Bezpečnostní monitoring ..... 18
2.3.1 Behaviorální analýza síťového provozu ..... 19
2.3.2 Log Management ..... 19
2.3.3 SIEM ..... 20
3 Způsoby monitorování ..... 21
3.1 Aktivní, pasivní, proaktivní a reaktivní monitorování ..... 21
3.2 S agentem ..... 21
3.3 Bez Agenta ..... 22
3.3.1 SNMP ..... 22
3.3.2 IPMI ..... 23
3.3.3 ICMP ..... 24
3.3.4 WMI ..... 24
3.3.5 NetFlow ..... 25
4 Monitorovací nástroje ..... 26
4.1 Nagios Core & Nagios XI ..... 26
4.2 Zabbix ..... 28
4.3 SolarWinds Server & Application Monitor ..... 29
4.4 Srovnání nástrojů ..... 30
4.4.1 Metodika ..... 30
4.4.2 Srovnání nástrojů ..... 30
4.4.3 Závěr srovnání ..... 32
5 Implementace monitorovacího nástroje Zabbix do produkčního prostředí reálné společnosti. ..... 34
5.1 Analýza prostředí a současného stavu ..... 35
5.2 Definice požadavků ..... 36
5.2.1 Upřesnění požadavků – Monitoring provozu ..... 36
5.2.2 Upřesnění požadavků – Způsob monitorování ..... 37
5.2.3 Upřesnění požadavků – Monitorovací nástroj ..... 37
5.3 Výběr vhodného monitorovacího nástroje ..... 37
6 Konfigurace Zabbixu ..... 38
6.1 Logické seskupení prvků ..... 38
6.2 Uživatelské přístupy a oprávnění ..... 39
6.3 Šablony ..... 39
6.3.1 Monitorované CI ..... 41
6.3.2 Prahové hodnoty ..... 43
6.3.3 Triggering ..... 45
6.3.4 Vizualizace dat ..... 50
6.3.5 Retence dat ..... 51
6.4 Alerting ..... 52

6.4.1 Eskalace ..... 54
6.5 Monitoring Windows služeb ..... 55
6.6 Testování ..... 57
6.7 Operativní požadavky ..... 59
6.8 Vyhodnocení konfigurace ..... 59
7 Doporučený postup implementace libovolného monitorovacího nástroje ..... 61
8 Závěr ..... 64
Seznam použitých zkratek ..... 66
Seznam literatury ..... 67
Seznam obrázků a tabulek ..... 71
Seznam obrázků ..... 71
Seznam tabulek ..... 71
Příloha 1 – Rešerše zdrojů a stávajících prací ..... 72
Příloha 2 – Trend vyhledávání Nagios Core & XI, Zabbix a SAM ..... 74
Příloha 3 – Test eskalačního pravidla pro spuštění zastavené tiskové služby ..... 75

# 1 Úvod

IS/IT hraje v současné době významnou roli pro konkurenceschopnost podniku. Díky IS/IT je možné zvyšovat a udržovat požadovanou úroveň kvality služeb, které jsou zákazníkům poskytovány. Dosažení určité kvality služeb by však bylo nemožné bez řádné správy IT infrastruktury. [1] Podstatnou roli při správě IT infrastruktury hraje zejména její monitoring. Díky neustálému monitorování jednotlivých prvků a jejich aktuálního stavu je tak možné reagovat na významné události, tzv.: incidenty, s velice krátkou reakční dobou, nebo jim úplně předcházet.

Dostupná řešení, v podobě současných monitorovacích nástrojů, nabízejí širokou škálu funkcionalit a možností, jak dosáhnout komplexního dohledu nad IS/IT. A to jak z pohledu provozu, za účelem monitorování výkonnosti a dostupnosti, tak i z pohledu bezpečnosti. V obou zmíněných případech nabízejí současné monitorovací nástroje rozsáhlé možnosti pro podporu analýzy a korelace v podobě agregace nasbíraných dat, jejich vizualizace do grafů či tabulek nebo pravidelného reportování. Na základě těchto údajů je dále možné lépe plánovat finanční výdaje a investice do IS/IT. Zároveň také dochází k eliminaci potenciálních ztrát např.: v případě porušení SLA v důsledku výpadku.

Cílem práce je poskytnout komplexní pohled na problematiku monitoringu a seznámit tak čtenáře s principy, způsoby a možnostmi monitoringu včetně srovnání vybraných monitorovacích nástrojů. Hlavním cílem práce je implementace a konfigurace nástroje Zabbix do produkčního prostředí reálné společnosti tak, aby byly splněny stanovené požadavky a výsledná konfigurace vyhovovala potřebám operátorů IT. Na základě tohoto procesu je posléze autorem práce sestaven doporučený postup pro implementaci libovolného monitorovacího nástroje. Toto jsou tedy i očekávané výstupy práce.

V práci jsou nejprve uvedeny obecné principy monitoringu a jeho přínosy pro provoz služeb dle ITIL. Následně jsou zmíněny i způsoby bezpečnostního monitoringu. Další část práce se věnuje představení jednotlivých způsobů monitorování a standardů, které je pro tyto účely možné využít. Následně jsou představeny a srovnány 4 vybrané monitorovací nástroje a jsou stanovena doporučení pro výběr jednotlivých nástrojů. V praktické části je popsán proces implementace a konfigurace monitorovacího nástroje Zabbix do produkčního prostředí reálné společnosti. Nejdříve je provedena analýza

prostředí a definice požadavků na monitoring. Při konfiguraci nástroje je brán ohled na vhodné definice prahových hodnot, konfiguraci triggeringu a alertingu.

Práce předpokládá základní znalosti čtenáře v oblasti informačních technologií. Vzhledem k budoucímu vývoji uvedených monitorovacích nástrojů je také práce omezena na aktuálně dostupné verze.

# 2 Monitoring

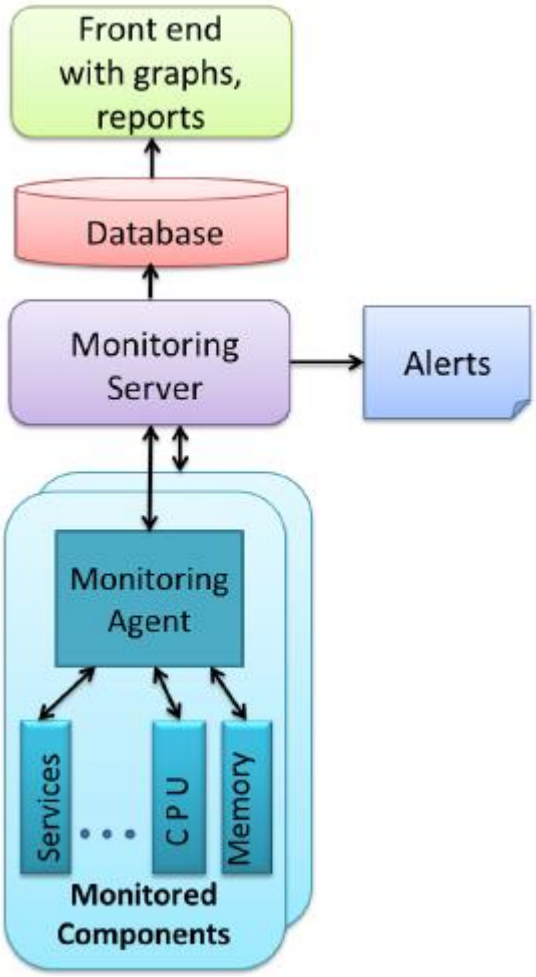
S pojmem monitoring se setkáváme v nejrůznějších odvětvích a v nejobecnější formě jej můžeme definovat jako systematický sběr a vyhodnocování informací o stavu sledované jednotky po určitou dobu. Monitoring ve smyslu IS/IT definuje ITIL [1, str. 332] jako „*Opakované sledování konfigurační položky (dále jen „CI"), služby IT nebo procesu za účelem detekce událostí a zjištění aktuálního stavu.*". Jako konfigurační položku pak ITIL označuje jakoukoliv komponentu nebo aktivum, které je potřeba spravovat za účelem dodávky IT služeb. Konfigurační položku představuje např.: software, hardware, IT služba, budova, personální zdroje nebo dokumentace. [1] Dominic Ogbonna [2, loc. 582] pak definuje monitoring IT infrastruktury jako „*proces kontroly zdraví systému, zahrnujících sběr dat o využití zdrojů, výkonu a dostupnosti systému. Zároveň také ukládání dat, jejich analýzu a alerting při přesahu definovaných prahových hodnot.*" Dále pak pokračuje a zmiňuje, že „*efektivní monitoring by měl pokrývat všechny prvky IT infrastruktury a jejich vzájemné závislosti.*"

Za účelem monitorování IS/IT existuje v současné době nespočet nástrojů, které nabízejí širokou škálu funkcionalit a možností, jak dosáhnout komplexního dohledu nad IT infrastrukturou. K dosažení tohoto cíle sbírají informace o stavu CI z různých zdrojů za pomoci různých protokolů a způsobů. Díky možnosti agregace nasbíraných dat a jejich vizualizace do grafů, tabulek nebo pravidelného reportování představují monitorovací nástroje také cenný zdroj informací při řešení dlouhodobých problémů či plánování budoucího rozvoje IS/IT.

Přestože existuje opravdu nespočet monitorovacích nástrojů, lze jejich architekturu obecně shrnout do následujících komponent: [3]

1) *Monitorovací server* – Centrální jádro monitoringu. Řídí intervaly a způsob sběru dat, ukládá data do databáze, vyhodnocuje data, provádí filtrace a korelace nasbíraných dat a výstupy v podobě vizualizace.
2) *Zdroje dat* – Windows nebo Linux/Unix agenti, aplikace, služby, prvky sítě.
3) *Databáze* – Úložiště nasbíraných dat a konfigurace monitorování.
4) *Uživatelské rozhraní* – Umožňuje přístup ke konfiguraci nástroje a monitorování či nasbíraným datům a jejich vizuální podobě (grafy atp.).

Zjednodušené schéma architektury monitorovacích nástrojů je zachyceno na *obrázku č.1*. V závislosti na rozsahu IT infrastruktury, která je monitorována, se pak také odvíjí i rozsáhlost monitorovacího nástroje a tím pádem i jeho architektura. *Distribuované monitorování* umožňuje monitorování rozsáhlé IT infrastruktury implementací více instancí monitorovacích serverů, které pracují v režimu Master (centrální, řídící server) a Slave (odlehčené instance, které poslouchají řídící server tím, že vykonávají sběr dat, které následné zasílají řídícímu serveru). Tímto způsobem dochází k zachování centrálního řízení monitorování a zároveň ke snížení zátěže sítě. [4]




*Obrázek 1 - Zjednodušená architektura monitorovacích nástrojů, Zdroj: [3]*

## 2.1 Principy monitoringu

Jak již bylo zmíněno, monitoring spočívá ve sběru informací o stavu CI a následném vyhodnocení jejího aktuálního stavu. Sběr těchto dat probíhá na třech úrovních: [2]

1) *Úroveň Businessu* – Monitorování dat podstatných pro business např.: počet objednávek, počet zákazníků, počet registrovaných uživatelů, počet provedených plateb.
2) *Úroveň služeb* – Monitorování výkonu aplikací, korektního provedení transakcí, odezvy a schopnosti plnit požadavky businessu. Vyhodnocení stavu probíhá na základě např.: uživatelské spokojenosti, počtu provedených transakcí, odezvě aplikace atp.
3) *Úroveň zdrojů* – Monitorování využití zdrojů IT infrastruktury aplikacemi a službami. Vyhodnocení stavu je založeno na interních metrikách využití výpočetních zdrojů např.: využití CPU, paměti RAM, HDD, sítě atp.

Tyto úrovně se navzájem ovlivňují např.: zvýšení počtu zákazníků a počtu provedených plateb na úrovni businessu klade vyšší požadavky na úroveň služby (služba musí být schopná vykonat vyšší počet transakcí za stejnou dobu), což se projeví jako vyšší požadavky na úrovni zdrojů. Například v důsledku nedostatečných výpočetních zdrojů (CPU), může dojít při nárůstu požadavků úrovně služeb na úroveň zdrojů, ke zpomalení jejich zpracování tzn.: zvýší se doba odezvy aplikace (CPU nezvládá požadavky zpracovávat), což se projeví na spokojenosti zákazníka či koncového uživatele. Pro další účely této práce je relevantní zejména úroveň zdrojů.

V následujících *kapitolách (2.1.1–2.1.4)* jsou zmíněny jednotlivé problematiky, které je vhodné brát v úvahu za účelem pochopení principů monitoringu a následné konfiguraci monitorovacího nástroje. Pozn.: Pro účely této práce je využito přenesených anglických výrazů tzn.: *Alert/Alerting*, Trigger / Triggering namísto překladu do českého jazyka tzn.: *Upozornění*, respektive *Spoušť/Spouštění*.

### 2.1.1 Metriky

Metriky představují jednotlivé ukazatele hodnot, které jsou sledovány za účelem zjištění a vyhodnocení aktuálního stavu CI [5]. *Tabulka č. 1* zachycuje příklady obvyklých metrik využívaných k monitorování IT infrastruktury

| **Prvek** | **Metriky** |
|---|---|
| *Server* | Využití CPU (- celkové, na uživatele, na systém, IOWAIT), CPU load, dostupnost serveru, využití RAM, zaplnění HDD, počet procesů, informace z Unixových utilit např.: sar, iostat či vmstat atd. |
| *Datacentrum* | Využití el. Energie, chlazení, využití šířky pásma, |
| *Síť/router/switch* | Využití šířky pásma, chybové stavy, přenesené pakety/s, aktivní porty |
| *SAN* | IOPS, alokace úložného prostoru |
| *NAS* | Operace/s, Latence, využití kapacity úložného prostoru, kapacita souborového systému |
| *HDD* | Čas čtení/zápisu/hledání, přenesené MB |
| *Web* | Počet požadavků, počet požadavků ve frontě, počet provedených transakcí |
| *Databáze* | Počet aktivních připojení, aktivních transakcí, dotazy/s, délka fronty, čtení/s, zápis/s |

*Tabulka 1 – Metriky monitorování IT infrastruktury, zdroj: [2], upraveno*

Ke správnému vyhodnocení stavu CI je také potřeba rozumět tomu, v jakých jednotkách je konkrétní metrika zaznamenána a jaký údaj daná hodnota představuje např.: prostý údaj v konkrétním čase, průměr nebo maximum hodnot za definovaný interval či jiný agregát. Tyto hodnoty je možné dále rozdělit na absolutní a procentuální, nebo v některých případech i na kvalitativní: [2][5][6]

- *Absolutní hodnota* – Představuje údaj o kvantitativním množství. Jednotky se odvíjí od monitorované CI. Např.: Jednotka udávající informaci o zaplnění místa na HDD je Byte.
- *Procentuální hodnota* – představuje procentuální údaj o využití monitorované CI. Např.: Server má přiděleno 64GB RAM z toho je aktuálně využito 20GB. Z toho vyplývá, že je využito 33,33% RAM.
- *Kvalitativní* – textový údaj

Výhodou procentuálního vyjádření hodnot je sjednocení jednotek do jednoduché a obecně známé podoby, ze které je možné ihned odvodit, zdali je aktuální stav položky v přijatelných mezích.

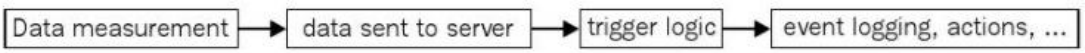
### 2.1.2 Prahové hodnoty

Prahová hodnota představuje hodnotu, jejíž dosažení či překročení má význam z hlediska provozu služeb. Překročení prahové hodnoty je tedy událost, která by měla způsobit odeslání alertu (dále popsáno v *kapitole 2.1.4*) nebo vykonání kontrolního opatření [1]. Příkladem prahových hodnot je např.: využití dostupné kapacity HDD z více jak 95%. Tento stav může zabránit korektnímu ukládání nových dat, či spuštění některých aplikací. Prahovou hodnotou může být také např.: odezva serveru je po dobu 5min vyšší než 320ms.

### 2.1.3 Triggering

Trigger (v českém překladu spoušť) představuje mechanismus pro zahájení kontrolního opatření. [7] Je to tedy např.: telefonát uživatele na ServiceDesk. Na tuto událost reaguje operátor ServiceDesku vykonáním činnosti, za účelem odstranění problému uživatele tzn.: provede opatření.

Trigger, v kontextu monitorovacích nástrojů, doplňuje vzniklé události v důsledku překročení prahových hodnot, či jiných změn CI o definici jejich významu a činností, kterých monitorovací nástroj vykoná za účelem návratu CI do běžného stavu. Triggering tedy představuje činnost filtrace nevýznamných událostí a korelaci významných událostí s definovanou sadou opatření. Pokud je na základě proběhlé filtrace a korelace vyhodnocen stav CI jako vyžadující kontrolního opatření, monitorovací nástroj buď alertuje operátory, kteří situaci dále vyhodnotí, nebo opatření vykoná automaticky např.: spuštění zastavené služby, restartování aplikace nebo operačního systému. [1][8][9][10] Na *obrázku č. 2* je zobrazena logika triggeru monitorovacího nástroje Zabbix.




*Obrázek 2 – Logika triggeringu monitorovacího nástroje Zabbix, zdroj: [3]*

Pro úspěšný triggering je důležité zejména správně rozlišit význam jednotlivých událostí. Joe Rogers [11] vidí jako vhodné rozlišovat události, které mají následující význam:

- *Informational* – Představuje stav, který nepředstavuje ohrožení (např.: proběhl update monitorovacího agenta).

- *Minor* – Představuje stav, který by měl být řešen, ale není naléhavý (např.: zbývá 15% volného místa na disku).
- *Major* – Představuje stav, který by měl být přednostně řešen. Pokud tomu tak nebude, hrozí výpadek služby (např.: nadměrné vytížení procesoru serveru může vést ke zpomalení aplikací/služeb).
- *Critical* – Aktuální nebo bezprostřední riziko selhání služby nebo zařízení (např.: výpadek serveru a nedostupné aplikace/služby).

ITIL definuje alert jako *„notifikaci o dosažení či překročení prahové hodnoty, o změně některé položky nebo o výskytu chyby.“* [1, str. 311] Účelem alertu (v českém překladu upozornění) je tedy upozornit operátory IT na významné události. [12] Alerty jsou odeslány monitorovacími nástroji na základě definovaných triggerů (viz *kapitola 2.1.3*). Odeslání je možné prostřednictvím různých komunikačních kanálů např.: e-mail, sms nebo Jabber. [2][3][12] Při konfiguraci alertů je zejména důležité se zaměřit na následující otázky:

*1) Koho alertovat?*

Alert by měl být směrován osobám, které mají potřebné schopnosti k řešení dané události. Není však vhodné alertovat pouze jednoho operátora IT. Naopak vhodnějším řešením je upozornit skupinu operátorů, nebo integrovat alerting se ServiceDeskem. Při odeslání alertu do ServiceDesku tak dojde k založení ticketu, který může být následně přiřazen konkrétnímu operátorovi a po vyřešení uzavřen. [1][2][12]

*2) Kdy alert odeslat?*

Čím více alertů operátoři denně obdrží, tím je větší šance, že dojde k jejich ignorování. Není proto potřebné alertovat události, které svojí závažností nevyžadují bezprostřední opatření. Okamžité alertování událostí může ale také vést k záplavě false positive notifikací. Např.: chvilkové vytížení CPU na 90% je v pořádku, pokud však toto vytížení trvá delší dobu, může mít vliv na rychlost aplikace. Pokud to povaha události dovolí tzn.: nejedná se o kritickou událost, je vhodné s odesláním alertu vyčkat na další data. Tímto způsobem je identifikace incidentu přesnější, ovšem na úkor rychlosti detekce. [12]

*3) Jaké informace zahrnout do alertu?*

Informace obsažené v alertu by měli jednoznačně identifikovat vzniklou událost a obsahovat potřebné informace k určení přesného významu a závažnosti dané události, za účelem zvolení vhodného opatření. Jedná se tedy např. o: odkaz na dokumentaci incidentu [1]. Zároveň by také alert měl obsahovat: název CI, její aktuální stav, naměřenou hodnotu včetně jednotek, časový údaj vzniku události, význam události atp.

## 2.2 Vazba monitoringu na Provoz služeb dle ITIL

Information Technology Infrastructure Library (ITIL) je soubor procesů a best-practice postupů dostupných jako sada knižních publikací. Díky ITILu je možné lépe plánovat, využívat a zefektivňovat využití služeb IT a IT infrastruktury. Cílem Provozu služeb (Service Operation) dle ITIL je zajištění kvalitní a efektivní dodávky služeb s ohledem na udržení bezchybného stavu služeb. [13]

Následující kapitoly popisují vybrané procesy Provozu služeb a přínos monitoringu.

### 2.2.1 Event Management (Správa událostí)

Jako event – událost – je dle ITIL [1] chápána jakákoliv změna stavu, která má význam z hlediska správy konfigurační položky či IT služby. Cílem tohoto procesu je správa událostí po dobu jejich životního cyklu a vyhodnocení aktuálního stavu konfigurační položky, na základě čehož je zvoleno vhodné opatření. Nevýznamné události nevyžadují vykonání žádného opatření, naopak významné události jsou předány do procesu správy incidentu viz. *kapitola 2.2.2.* ITIL definuje význam událostí jako: *Informational, Warning a Exception.*

#### Přínos monitoringu

*„Správa událostí pracuje s událostmi, které jsou specificky generovány k tomu, aby byly monitorovány. Monitoring jednak sleduje tyto výskyty, ale také aktivně sleduje stavy konfiguračních položek, které události negenerují.*“ [1, str. 59] Např.: využití RAM. Přínosem monitoringu pro správu událostí je tedy zejména automatizovaná a značně rozšířená možnost monitorování CI. Dále pak také centralizovaná správa a úložiště událostí, nebo v některých případech i vykonání automatického opatření v reakci na vzniklou událost.

### 2.2.2 Incident Management (Správa incidentů)

Incidentem je neplánovaný výpadek nebo omezení IT služby. Jako incident je vnímána i porucha konfigurační položky, která ještě neovlivnila provoz IT služby. Hlavním cílem procesu správy incidentů je obnova provozu IT služby do běžného režimu s minimálním dopadem na provoz businessu, nikoliv samotné odstranění příčiny. Správa incidentů spravuje životní cyklus incidentů a udává jaké kroky podstoupit během jednotlivých fází. [1]

#### Přínos monitoringu

Z hlediska businessu by mělo k identifikaci incidentu dojít dříve, než dojde k dopadu na zákazníky nebo na uživatele. Za tímto čelem by měl být implementován monitoring všech podstatných CI pro provoz služeb tak, aby bylo možné zahájit proces správy incidentů s co nejkratší reakční dobou. [1]

Jak bylo popsáno v kapitole 2.1.4 operátoři IT jsou monitorovacími nástroji, za pomoci alertů, informováni o výskytu významných událostí včetně relevantních informací, které usnadňují diagnostiku incidentů, a tedy i jeho následné řešení. V závislosti na povaze daného incidentu a nastavení monitorovacího nástroje, je možné některé incidenty řešit vykonáním automatického opatření např.: spuštění zastavené služby nebo restartování OS. Tímto způsobem je značně snížena reakční doba na vzniklý incident a případně úplně eliminován dopad incidentu na business.

### 2.2.3 Problem Management (Správa problémů)

*Správa problémů je proces zodpovídající za správu životního cyklu všech problémů. ITIL definuje problém jako příčinu jednoho či více incidentů.* [1, str. 97] Hlavním cílem procesu správy problémů je minimalizace dopadu incidentů na business včasným odstraněním chyb v IT infrastruktuře a zamezení jejich opakovanému vzniku. Důležitým vstupem informací je správa incidentů. Každý záznam problému by tak měl obsahovat křížový odkaz na zapříčiněné incidenty.

Správu problémů je možné z hlediska přístupu k řešení problémů rozdělit na:

- *Reaktivní* – Řeší problémy v reakci na již vzniklé incidenty.

- *Proaktivní* – Spočívá v identifikaci a řešení chyb ještě před tím, než zapříčiní vznik incidentu. Mezi aktivity proaktivní správy problémů patří např.: pravidelné kontroly logů, grafů a analýzy trendu provozu IT infrastruktury. [1]

### Přínos monitoringu

Podstatným krokem v procesu správy problémů je analýza nasbíraných dat za účelem přesné identifikace důvodu vzniku problému a jeho následného odstranění. [14] Přínosem monitorovacích nástrojů jsou právě funkcionality, které tuto analýzu podporují tzn.: vizualizace a korelace nasbíraných dat nebo reporting. Například pravidelný reporting nejčastěji se vyskytujících událostí může pomoct odhalit problém na IT infrastruktuře. Nebo korelace hlášených incidentů uživateli, spojená s analýzou trendu využití paměti RAM, může pomoct odhalit nedostatečnou disponibilní kapacitu RAM na konkrétním serveru. Tento poznatek by měl být dále využit k navýšení paměti RAM na daném serveru a úpravě triggerované prahové hodnoty.

## 2.3 Bezpečnostní monitoring

Prozatím byl uvažován pouze monitoring z pohledu provozu služeb, ovšem je patřičné zmínit i bezpečnostní monitoring. Bezpečnostní monitoring spočívá ve sběru a analýze informací za účelem detekce podezřelé aktivity a tím pádem zamezení či včasného odhalení bezpečnostního incidentu [15]. Bezpečnostní incident definuje řada norem ISO/IEC 27000:2016 jako „*jednu nebo sérii nechtěních či neočekávaných bezpečnostních událostí, u kterých existuje vysoká pravděpodobnost kompromitace činností organizace a ohrožení bezpečnosti informací.*“. [16, str. 6] Bezpečnostní událost poté norma definuje jako „*identifikovatelný stav systému, služby nebo sítě, který indikuje možné narušení bezpečnosti informací, porušení bezpečnostních politik nebo selhání bezpečnostních opatření.*“. [16, str. 6]

Informační bezpečnost spočívá v zachování konfidence, integrity a dostupnosti informací. [16] Nejen v důsledku nařízení Evropského parlamentu a Rady (EU) 2016/679 označovaného jako *Obecného nařízení o ochraně osobních údajů* (dále jen „*GDPR*“), které vstoupí v účinnost dne 25.5.2018 je na bezpečnostní incidenty kladen velký důraz, jelikož ohrožují jak dobré jméno společnosti, tak údaje a zájmy zákazníků. GDPR, mimo jiné, ukládá povinnost správcům údajů, oznamovat porušení zabezpečení

osobních údajů Úřadu pro ochranu osobních údajů, a to nejpozději do 72h od okamžiku, kdy se o incidentu dozvěděli. [17]

Následující kapitoly pojednávají o způsobu monitoringu se zaměřením na bezpečnost provozu a informací.

### 2.3.1 Behaviorální analýza síťového provozu

Behaviorální analýza síťového provozu (*network behaviour analysis,* dále jen *NBA*) je zaměřena na monitorování síťového provozu a detekci anomálií, které mohou představovat potenciální bezpečnostní incident. K tomuto účelu je nejdříve stanoven dataset běžného provoz sítě tzv.: *profil chování*. NBA nástroje jsou tedy nejdříve implementovány do sítě za účelem pasivního monitoringu síťového provozu např.: který počítač komunikuje s kterým serverem, jaké protokoly a porty jsou využity a kdy k této komunikaci dochází. Případně je možné tento profil chování také sestavit manuálně. Během této fáze je důležité, aby nebyl síťový provoz nijak kompromitován. Na základě takto stanoveného profilu chování jsou posléze detekovány a alertovány jakékoliv anomálie. Pro účely NBA je využit protokol NetFlow respektive ISPFX standard, který je dále popsán v *kapitole 3.3.5*.

NBA, na rozdíl od Systémů pro odhalení průniku (tzv.: *IDS*), nepracuje na základě již známých útoků, zranitelností, nebo škodlivého kódu tzv.: *signatur*. Umožňuje tedy i odhalení *malwaru*, který není doposud IDS nebo antiviry znám tzv.: zero-day *worm*. Pomocí NBA je také možné odhalit např.: nežádoucí aktivity uživatelů, jako je pokus o upload interních dat na filehostingové servery. [18]

### 2.3.2 Log Management

Logy představují záznamy o událostech prvků IS/IT infrastruktury (např.: aplikačním softwarem, operačními systémy, routery, firewally atp.), kterými jsou generovány pro účely auditu bezpečnosti a provozu, nebo za účelem laděn. Logy relevantní z pohledu bezpečnosti jsou např. logy o:

- Přihlášení/odhlášení uživatele, a to včetně nezdařilých pokusů o přihlášení.
- Navázané spojení ke službě, aplikaci, serveru, pc apod.
- Přístupy uživatelů ke složkám a souborům.
- Pády aplikací, služeb, systémů nebo různé chybové zprávy.

Přenos logů probíhá pomocí různých protokolů např.: Syslog, Windows event log, nebo SNMP. Jak Syslog tak Windows event log pak také představují standardizovaný formát logů. Výpis záznamu ze Sysloglogu o připojení uživatele přes SSH je uveden níže:

```
Jan 2 08:44:54 ns1 sshd2[23661]: User anton, coming from 65.211.15.100, authenticated.
```

Log management potom představuje disciplínu centralizovaného sběru, ukládání, a uchovávání logů z různých zdrojů. [19]

### 2.3.3 SIEM

Log management je zaměřen pouze na sběr a uchování logů. *Security Information and Event Management* (*Správa bezpečnostních událostí a informací*, dále jen *SIEM*), nejčastěji představuje SW nástroj, který rozvíjí podstatu log managementu a doplňuje:

- Převod logů z různých formátů do jednotné podoby tzv.: *parsing*.
- Možnosti pokročilé analýzy, filtrace a korelace logů v reálném čase na základě definovatelných pravidel (v terminologii SIEM se využívá pojem „*Korelační pravidla*“)
- Možnosti alertingu, reportingu a vizualizace dat.

SIEM tak představuje auditní a monitorovací nástroj, pomocí kterého je možné provádět analýzu historických aktivit, reporty požadovaných informací, nebo alerting podezřelé aktivity v reálném čase. Vybrané možnosti SIEM nástrojů jsou např.: [20]

- *Alerting pokusu vniknutí do interní sítě.*
    - Korelační pravidlo: více jak 5 neúspěšných pokusů o autentizaci z jednoho počítače za 1 min.
    - Zdroje událostí: Active Directory, Syslog, RADIUS, Aplikace
- *Alerting uložení nebezpečného souboru na server*
    - Korelační pravidlo: uložení spustitelného souboru na server (.exe, .jar, .cgi, .asp) z neznámého zdroje
    - Zdroje událostí: Syslog, Windows event log
- *Reporting aktivních uživatelů*
    - Korelační pravidlo: alespoň jedna úspěšná autentizace uživatele za poslední měsíc.
    - Zdroje událostí: Active Directory, RADIUS

# 3 Způsoby monitorování

Podle toho, jakým způsobem jsou data pro zjištění aktuálního stavu CI sbírána, lze monitorování rozdělit na monitorování s agentem a bez agenta. Dále je také možné přihlédnout k režimu, v jakém je stav konfiguračních položek monitorován a vyhodnocován. Takto lze rozdělit způsob monitorování na aktivní a pasivní, proaktivní a reaktivní [1].

## 3.1 Aktivní, pasivní, proaktivní a reaktivní monitorování

V případě *aktivního monitorování* je využito pravidelného dotazování monitorovacího nástroje na aktuální stav CI. Aktivní monitorování klade vyšší nároky na zdroje IT infrastruktury a mělo by být využito k monitorování kritických zařízení pro provoz služeb. Aktivního monitorování je také možné využít k ověření diagnózy při řešení incidentů.

V případě *pasivního monitorování* je k zjištění aktuálního stavu CI využito generovaní a přenosu pouze těch událostí, které splňují definovaná kritéria, např.: překročení prahové hodnoty. Například pomocí SNMP trap.

*Proaktivní monitorování* je využito za účelem sledování vzorů událostí, které mohou indikovat potenciální incident. Cílem proaktivního monitorování je zamezení výskytu incidentů, potažmo problémů. Jedná se tedy např. o: analýzy trendů vývoje stavu CI atp.

*Reaktivní monitorování* spočívá ve vykonání opatření v důsledku reakce na již vzniklé události. K vykonání opatření dochází v důsledku triggerování viz *kapitola 2.1.3* [1].

## 3.2 S agentem

Monitorování s agentem vyžaduje instalaci programu (tzv.: agenta) na monitorovaný prvek (počítač, server), který sbírá požadované údaje a odesílá je na monitorovací server, nebo tyto data odesílá na jeho vyžádání. Výhodou tohoto způsobu je, že pomocí agenta je možné získat širší škálu a podrobnější údaje než v případě monitorování bez agenta. Údaje je také možné sbírat v kratších intervalech a ke sběru dochází lokálně na monitorované jednotce. Nevýhodou je potřeba instalace samotného agenta, v důsledku čehož přibývá další spuštěný proces, což může mít vliv na výkon daného prvku. [21]

## 3.3 Bez Agenta

Jeremy Sherwood [22] vidí monitorování bez agent spíše než jeho úplnou absenci, jako využití již přítomného agenta pro správu, který je zabudován do software zařízení, nebo je již přítomný jako vlastnost pro správu systému (SW/HW) a nepotřebuje dodatečnou instalaci. Příkladem může být využití SNMP protokolu nebo IPMI standardu, kdy na monitorovaném prvku musí být přítomný SNMP agent, respektive IPMI sensor. Nejedná se však o agenta samotného monitorovacího nástroje. Dále je možné monitorovat dostupnost serveru pomocí ICMP protokolu (např. ping) nebo vlastní služby serveru např.: HTTP, FTP, SMTP. [23]

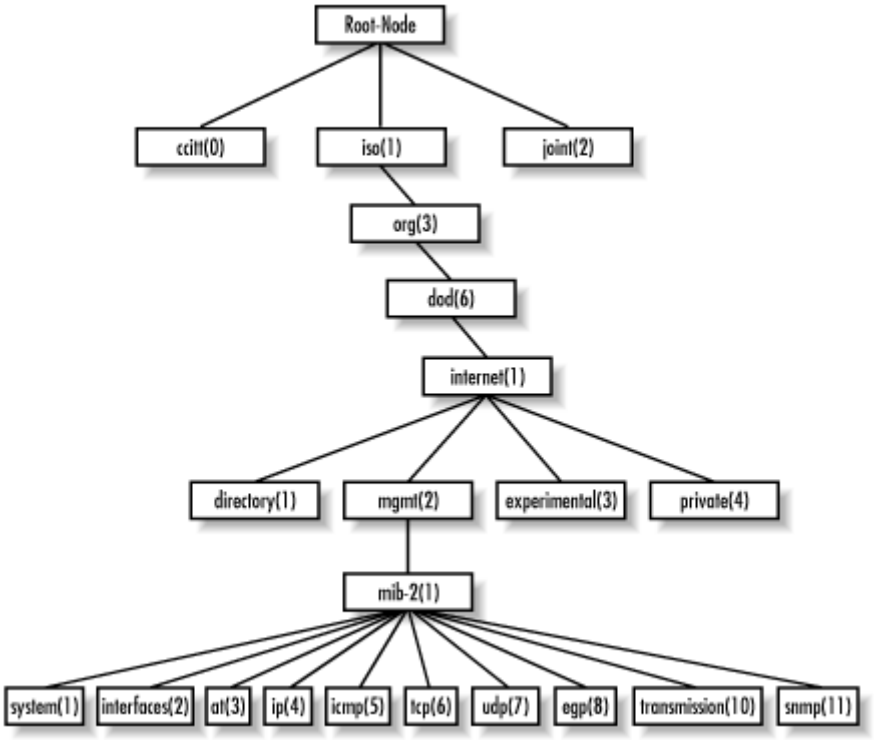
Následující kapitoly popisují nejčastější protokoly a standardy, kterých je možné využít za účelem monitoringu bez agenta.

### 3.3.1 SNMP

*Simple Network Managemet Protocol* (SNMP) popsaný v RFC 1157 [24] je jednoduchý protokol, pomocí kterého lze získat informace o stavu síťových prvků např.: switche, routery, tiskárny, ale i server atp. Jak již bylo zmíněno výše, k monitorování pomocí SNMP je zapotřebí přítomnost SNMP agenta na daném zařízení.

- *SNMP Agent* – zpracovává dotazy SNMP managera tím, že získává informace o lokálním systému tzn: musí být přítomný na daném prvku a zasílá je jako odpověď zpět správci.
- *SNMP Manager*– Je jakýkoliv systém, který je konfigurován tak, aby pomocí dotazování získával informace o síťovém prvku od agenta.

SNMP Agent si udržuje Informace o systému v dotazovatelé stromové struktuře tzv.: *Management Information Base* (MIB). MIB obsahuje jednotlivé objekty, ve kterých jsou uloženy informace např. o: síťových rozhraních, nebo době provozu síťového prvku (tzv.: *up-time*). Každý z těchto objektů je jednoznačně identifikován číselným identifikátorem tzv.: *Object Identifier* (OID). OID představuje číselnou posloupnost, kdy jednotlivá čísla představují průchod objekty strukturou MIB. Například údaj o up-time síťového prvku má OID 1.3.6.1.2.1.1.3 viz *obrázek č. 3.*




*Obrázek 3 – Struktura MIB, Zdroj: [25]*

Komunikace SNMP probíhá pomocí následujících příkazů [25]:

- *Get* – Dotaz na hodnotu určitého OID poslaný správcem agentovi.
- *GetNext* – Dotaz na hodnotu následujícího OID.
- *Set* – Požadavek správce na modifikaci hodnoty OID agentem.
- *GetBulk* – Dotaz na několik hodnot OID najednou.
- *Response* – Odpovědi agenta správci.
- *Trap* – Zpráva zaslána agentem správci, pokud některá z hodnot překročila určenou hranici.
- *Inform* – Potvrzení přijetí Trap zprávy.

### 3.3.2 IPMI

Standard *Intelligent Platform Management Interface* (IPMI) umožňuje monitoring stavu HW komponent (např.: teploty, příkon napájení, rychlosti otáček větráků atd.) serverů případně jiných prvků s podporou IPMI a jejich vzdálené ovládání (např.: vypnutí a zapnutí zamrzlého serveru odpojením od napájení). IPMI dále umožňuje sběr logů a inventarizaci HW komponent. Funkcionalita IPMI je zajištěna díky speciálnímu procesoru – *Baseboard Management Controller* (BMC) – umístěnému na základní desce

serveru. BMC je s ostatními HW komponenty základní desky propojen pomocí jednoduché sběrnice. [26]

### 3.3.3 ICMP

K nejzákladnějšímu monitoringu úplně dostačuje informace o dostupnosti, respektive nedostupnosti daného síťového prvku. Tuto informaci lze získat díky příkazu PING, který využívá zprávu typu *Echo Request* protokolu ICMP (*Internet Control Message Protocol*). ICMP protokol využívají síťové prvky k odesílání chybových či provozních zpráv např.: o nedostupnosti služby nebo cílového prvku. Mezi zprávy, které lze využít k monitoringu patří: [27]

- *Echo Request* – Požadavek na odpověď síťového prvku.
- *Echo reply* – Odpověď na Echo Request.
- *Destination unreachable* – informace o nedostupnosti cíle
    - *0 = Net unreachable* – nedostupná síť
    - *1 = host unreachable* – nedostupný host
    - *2 = protocol unreachable* – nedostupný protokol
    - *3 = port unreachable* – nedostupný port
    - *4 = fragmentation needed* – potřeba fragmentovat datagram
    - *5 = source route failed* – nedostupný zdrojová routa
    - Kódy 0, 1, 4 a 5 vrací výchozí brána.
    - Kódy 2 a 3 odesílá cílový host.

### 3.3.4 WMI

Pomoci rozhraní *Windows Managemen Instrumentation* (WMI) je možné získávat údaje o stavu CI z prostředí MS Windows. WMI bylo vytvořeno společností Microsoft jako sada specifikací založených na CIM, které umožňují standardizovanou správu zařízení s operačním systémem MS Windows. WMI tedy představuje rozhraní, pomocí kterého je možné získat informaci např. o: stavu služby, nainstalovaném SW, využití HDD apod. [28]

Struktura WMI se skládá ze tří komponent: [29]

- *WMI Consumer* – Aplikace nebo programovací či skriptovací jazyky, které umožňují dotazování na WMI objekty např.: PowerShell, .NET, C, C++.

- *WMI infrastructure*
  - Object manager – Spravuje a dohlíží na použité instance WMI v systému.
  - *WMI* repository – Databáze všech dostupných WMI objektů, které se nacházejí v daném systému.
- *WMI Provider* – Prostředník, který vykonává interakci mezi sledovanými objekty a *Object managerem* a poskytuje požadovaná data *WMI Consumerem*. *WMI Provider* je např.: ovladač HW, DLL knihovna, registr atd. [30]

### 3.3.5 NetFlow

NetFlow je protokol vyvinutý společností CISCO, který umožňuje monitoring síťového provozu na základě IP toků. Pakety, které se shodují zdrojovou a cílovou IP adresou, číslem zdrojového a cílového portu a typem protokolu jsou seskupeny do jednoho záznamu IP toku. Tento záznam je dále doplněn o vstupní, respektive výstupní rozhraní a průběžně aktualizován o počet přenesených paketů a bytů:

Architektura NetFlow se skládá z následujících prvků:

- *NetFlow exportér* – zaznamenává NetFlow záznamy a odesílá je kolektoru.
  - *NetFlow* kompatibilní router nebo switch tzn.: aktivní sonda
- *NetFlow kolektor* – server, který ukládá, agreguje vizualizuje a reportuje informace o síťovém provozu.

Na základě NetFlow v9 vznikl standard Internet Protocol Flow Information Export (IPFIX), který definuje, jak formátovat záznamy o IP toku a jak tyto záznamy přenášet z exportéru do kolektoru. [31]

V současné době je také možné využít tzv.: pasívních netflow sond. Pasivní sonda je specializované HW zařízení, které je zapojené do sítě prostřednictvím mirroring portu centrálního routeru nebo switche, nebo pomocí pasivního rozbočovače tzv.: TAP. Pasivní sondy rozšiřují možnost monitorování IP toku o položky vyšších vrstev L5-L7 např.: http informace, DNS dotazy atd. Způsobem svého zapojení navíc nijak neovlivňují výkon sítě. [32]

# 4 Monitorovací nástroje

Faktu, že je monitoring provozu IS/IT v současné době již obvyklou součástí organizací nasvědčuje i dostupnost množství komerčních i open-source monitorovacích nástrojů na trhu. Následující kapitoly představují 4 vybrané monitorovací nástroje z pohledu podporovaných platforem, přístupu k monitorování a cenové politiky. Srovnání těchto nástrojů je dále provedeno v *kapitole 4.4*.

## 4.1 Nagios Core & Nagios XI

Monitorovací nástroj *Nagios* vznikl v roce 2002 z projektu s názvem *NetSaint*. Instalace Nagios serveru je oficiálně podporována na OS CentOS nebo RHEL. Jako databázi využívá MySQL nebo MariaDB. V případě verze Nagios XI (viz níže) je k dispozici také před-konfigurovaná VM appliance. Monitorování je možné za pomoci jak Windows či Linux/Unix agentů, tak i bez agenta. [33]

Monitorování pomocí Nagiosu je založeno na tzv.: *Pluginech*. Plugin představuje externí komponentu – kód/příkaz, který rozšiřuje funkcionalitu Nagiosu [34]. Například k dotazování na Linux agenta (NRPE) dochází pomocí pluginu *check_nrpe*. Nagios spustí plugin se vstupními argumenty (požadované informace), který kontaktuje agenta. Agent vykoná příkaz na kontrolu položky a výsledek zašle zpět Nagios serveru. Obdobně umožňují pluginy dotazování např.: pomocí SNMP. [35]

Monitorované prvky rozděluje Nagios dle následné logiky. V obou případech je možné seskupovat prvky do logických skupin: [34]

- *HOSTS* – Představují prvky sítě (fyzické či virtuální) např.: server, router, switch atp.
- *SERVICES* – Logické prvky, které jsou vázány na hosty – představují jejich funkcionalitu, nebo umožňují využití této funkcionality např.: proces služby SSH, který je spuštěn na serveru.

Nagios je dostupný ve dvou verzích: [36]

- *Nagios Core* – Open-source jádro nástroje. Veškerá konfigurace jak serveru, tak monitorování probíhá editací konfiguračních souborů v CLI a tato změna se projeví až po restartování Nagios serveru. Nagios Core je zdarma.
- *Nagios XI* – Placená verze nástroje, která rozšiřuje Nagios Core o implementaci dalších funkcionalit např.: konfiguračních průvodců, pomocí kterých je možné konfigurovat

sledování položek i samotný nástroj z webového rozhraní atp. Ve ceně XI je také zahrnuta technická podpora.

Cenová politika Nagiosu XI je založena na zakoupení roční licence a jejím následném obnovení. Cena licence, respektive obnovení je dána počtem uzlů (jakýkoliv prvek IT infrastruktury s IP adresou), které je možné monitorovat. Za $1,500, respektive +$750 v případě obnovy, je k dispozici upgrade na *XI Enterprise* edici, která nabízí dodatečné vlastnosti např.: rozšířené možnosti reportingu, audit změn nastavení nebo možnost hromadné změny nastavení.

Obnovení licence je možné buď v režimu „*Jen údržba*“, kdy zákazník získá přístup pouze k aktualizacím Nagiosu XI, nebo v režimu „*údržba & podpora*“, ve kterém získá zákazník i technickou podporu. *Tabulka č. 2* zachycuje ceny licencí a její obnovy pro různé počty uzlů. V případě potřeby dodatečné technické podpory je možné licencovat balíček řešení 5, respektive 10 incidentů ročně viz *tabulka. č. 3.* [36]

| | | Obnova licence | |
|---|---|---|---|
| **Počet uzlů** | **Cena licence** | **Jen údržba** | **Údržba & podpora** |
| 100 uzlů | $1,995 | $645 | $1,695 |
| 200 uzlů | $3,495 | $1,045 | $2,795 |
| 300 uzlů | $4,995 | $1,495 | $3,995 |
| 400 uzlů | $5,995 | $1,795 | $4,795 |
| 500 uzlů | $6,995 | $2,095 | $5,595 |
| 1000 uzlů | $11,995 | $3,595 | $9,595 |
| Neomezeně | $19,995 | $5,995 | $15,995 |
| Enterprise upgrade | +$1,500 | +$750 | +$750 |

*Tabulka 2 – Ceny licencí Nagios XI, zdroj: [36], vlastní zpracování*

| | Způsob nahlášení/řešení | |
|---|---|---|
| **Počet incidentů** | **po telefonu** | **po-emailu** |
| 5 | $1,995 | $995 |
| 10 | $2,995 | $1,495 |

*Tabulka 3 – Ceny technické podpory Nagiosu, zdroj: [36], vlastní zpracování*

## 4.2 Zabbix

První stabilní verze open-source monitorovacího nástroje *Zabbix* byla vydána v roce 2004. Nástroj je aktualizovaný v pravidelných 6 měsíčních intervalech a během Q2 2018 dojde k vydání verze 4.0. [37] Zabbix server je možné provozovat na OS CentOS, Debian, Oracle Linux, RHEL nebo Ubuntu. Jako databázi je možné využít MySQL nebo PostgreSQL. K dispozici je také již před-konfigurované VM Applience. K monitorování je možné využít Windows i Linux/Unix agenta. Pokud není instalace agenta na zařízení možná, Zabbix dokáže monitorovat zařízení i bez agenta pomocí např.: SNMP, IPMI, VMware API, atp. [38]

Zabbix využívá za účelem sběru dat tzv.: položky (*items*). Tyto položky představují jednotlivé zdroje dat a jednotky naměřených údajů. Položkou je např.: celková kapacita HDD v Bytech. V závislosti na způsobu monitorování – aktivní/pasivní, s agentem/bez agenta – dochází k pravidelnému dotazování Zabbix serveru na hodnoty dané položky, resp. k zasílání hodnot agentem na Zabbix server. [3] Tyto položky je možné shlukovat do šablon, které jsou přiřazeny sledovaným prvkům (hostům). Tímto způsobem se značně urychluje proces nastavení monitorování nového hosta, či změna monitorovaných položek – úprava jedné položky v šabloně se aplikuje na všechny hosty, kteří mají danou šablonu přiřazenou. [8]

Jelikož se jedná o open-source nástroj, je možné provozovat Zabbix bezplatně. Placená je v případě Zabbixu jen volitelná technická podpora. K dispozici je pět stupňů podpory uvedených v *tabulce č. 4.* Cenová kalkulace konkrétního stupně pak dále závisí na rozsahu implementace nástroje v organizaci. [39]

| | Stupeň technické podpory | | | | |
|---|---|---|---|---|---|
| **Položka podpory** | **Bronze** | **Silver** | **Gold** | **Platinum** | **Enterprise** |
| Počet incidentů | 4 | 8 | ∞ | ∞ | ∞ |
| Počet kontaktních osob | 1 | 1 | 2 | 3 | 5+ |
| Dostupnost podpory (h x d) | 8 x 5 | 8 x 5 | 8 x 5 | 24x7 | 24x7 |
| Garantovaný čas reakce | 2 dny | 1 den | 4 hodiny | 4 hodiny | 4 hodiny |
| Telefonická podpora | ne | ano | ano | ano | ano |

*Tabulka 4: Stupně technické podpory, zdroj: [39], vlastní zpracování*

## 4.3 SolarWinds Server & Application Monitor

*Server & Application Monitor* (SAM) je monitorovací nástroj společnosti Solarwinds, která se zaměřuje na vývoj a distribuci SW nástrojů pro komplexní správu IS/IT infrastruktury. SAM je zaměřený na monitoring serverů a aplikací. Jeho provoz je možný na Windows server 2012, 2012 R2 nebo 2016 a jako databázi využívá OrionSQL. Monitorování s agentem probíhá za pomoci SolarWinds Orion agenta, který je dostupný pro MS Windows i Linus/Unix distribuce. [40] K dispozici je také možnost monitorování bez agenta.

Logika monitorování pomocí SAM je založena na šablonách. Šablony, stejně jako v případě Zabbixu, představují soubor položek a metrik (SAM označuje tyto položky jako tzv.: *komponenty*), které nástroj monitoruje. Položka je např.: stav služby nebo využití CPU. SAM umožňuje monitorování fyzických zařízení i aplikací např.: OS, databáze, Webové servery, Emailové servery, virtualizační platformy atd. Na komunitních stránkách Solarwinds je k dispozici ke stažení více jak 800 již před-konfigurovaných šablon. Výhodou SAM je také možnost využívat vlastních scriptů a scriptů vytvořených pro Nagios. [41]

Cenová politika SAM je založena na počtu položek, které daná licence dovolí monitorovat. K dispozici je 6 licencí, jejichž ceny i počet monitorovaných položek je zachycen v *tabulce č. 5.* [42]

| Licence | Počet položek | Cena |
|---|---|---|
| SAM AL150 | 150 | € 2,440 |
| SAM AL300 | 300 | € 4,720 |
| SAM AL700 | 700 | € 8,635 |
| SAM AL1100 | 1100 | € 13,195 |
| SAM AL1500 | 1500 | € 17,760 |
| SAM ALX | ∞ | € 29,410 |

*Tabulka 5 – Licence SolarWinds SAM, zdroj: [42], vlastní zpracování*

## 4.4 Srovnání nástrojů

V této kapitole je provedeno srovnání výše zmíněných monitorovacích nástrojů. Za tímto účelem byli využity informace obsažené v kapitolách věnujících se jednotlivým nástrojům tzn.: *4.1-4.3,* dostupných demo verzí na stánkách výrobců a autorem zpracované tabulky srovnání daných nástrojů viz *tabulka č. 6.* Účelem srovnání je zhodnotit čtyři nejčastěji vyhledávané nástroje (*viz příloha č. 2*) a stanovit doporučení pro výběr daných nástrojů pro různé případy.

### 4.4.1 Metodika

Srovnání je provedeno na základě vlastností a poskytovaných funkcionalit daných nástrojů. Jednotlivé vlastnosti, respektive funkcionality jsou rozděleny do 5 kategorií, které analyzují různá kritéria, která mohou hrát roli při výběru konkrétního nástroje. Každému kritériu jsou přiděleny maximálně dva body. Jeden bod je stržen, pokud je kritérium omezené, nebo je některý nástroj v libovolném smyslu zvýhodněn. Dva body jsou potom strženy, pokud dané kritérium není podporováno, nebo jsou ostatní nástroje v libovolné smyslu zvýhodněny. Jelikož mají kritéria různou váhu při rozhodování např.: cena nástroje je pro výběr důležitější než možnost tvořit grafy, je každé kategorii přidělena váha, která tento fakt zohledňuje. Maximálně je možné získat 64 bodů. Výsledek srovnání je potom vytvořen součtem přidělených bodů vynásobených váhou dané kategorie. *Tabulka č. 6* zachycuje přidělené body a váhy jednotlivým kategoriím..

| Kategorie | Popis | Max. bodů | Váha |
|---|---|---|---|
| *Technologická stránka* | Podporované platformy | 6 | 0,05 |
| *Sběr dat* | Zdroje dat – agent, protokoly | 20 | 0,25 |
| *Monitoring* | Možnosti monitoringu – triggering, alerting apod. | 16 | 0,2 |
| *Zpracování dat* | Vizualizace, reporting | 8 | 0,1 |
| *Uživatelská stránka* | UI, Cena, Cenová politika | 14 | 0,4 |

*Tabulka 6 - Kategorie srovnání nástrojů, zdroj: autor*

### 4.4.2 Srovnání nástrojů

*Tabulka č. 7* zachycuje srovnání vybraných funkcionalit nástrojů Nagios Core, Nagios XI, Zabbix a SAM.

| | | Nástroj | | | |
|---|---|---|---|---|---|
| | **Kritérium** | ***Nagios Core*** | ***Nagios XI*** | ***Zabbix*** | ***SAM*** |
| *Technologická stránka* | *Server OS* | CentOS, RHEL | CentOS, RHEL | CentOS, RHEL, Debian, Ubuntu,… | Win server |
| | *Agent OS* | Win/Linux/Unix | Win/Linux/Unix | Win/Linux/Unix | Win/Linux/Unix |
| | *DB* | flat files | MySQL/MariaDB PostgresQL | MySQL, PostgreSQLI BM DB2 | OrionSQL |
| *sběr dat* | *Agentem* | ano | ano | ano | ano |
| | *Bez agenta* | ano | ano | ano | ano |
| | *IPMI* | ano[1] | ano[2] | ano | ano |
| | *ICMP* | ano[1] | ano[2] | ano | ano |
| | *WMI* | ano[1] | ano[2] | přes agenta | ano |
| | *SNMP/SNMP traps* | ano[1]/ne | ano[2]/ano[2] | ano/ano | ano/ano |
| | *NetFlow* | ne | ne | ne | ne |
| | *VMware API* | ne | ano[2] | ano | ano |
| | *Web monitoring* | ne | ano[2] | ano | ano |
| | *Monitorování logů* | ne | ano[2] | ano | ano |
| *Monitoring* | *Alerting* | email | email,sms[3] | email,sms[4] | email, sms[3] |
| | *Prahové hodnoty* | ano | ano | ano | ano |
| | *Triggering* | ano | ano | ano | ano |
| | *Kontrolní opatření*[5] | ano | ano | ano | ano |
| | *Závislosti*[6] | ano | ano | ano | ano |
| | *Plánování kapacit* | ne | Ano[7] | omezeně | ano |
| | *Predikce* | ne | ano[7] | ano | ano |
| | *Stupně závažnosti* | 4 dané | 4 dané | definovatelné | definovatelné |
| *Zpracování dat* | *Dashboardy* | ano[1] | ano | ano | ano |
| | *Grafy* | ano[1] | ano | ano | ano |
| | *Mapy* | ano[1] | ano | ano | ano |
| | *Reporting* | ne | ano | omezeně | ano |
| *Uživatelská stránka* | *konfigurace* | CLI | UI wizard | UI wizard | UI wizards |
| | *Víceuživatelské UI* | ne | ano | ano | ano |
| | *Intuitivní UI* | ne | ne | ano | ano |
| | *Komunita* | největší | největší | malá | velká |
| | *Cenová politika* | zdarma | počet uzlů | zdarma | počet položek |
| | *Cena* | zdarma | od $1,995 | zdarma | od $2,440 |
| | *Technická podpora* | placená | v ceně | placená | v ceně |

*Tabulka 7 - Srovnání Nagios Core, Nagios XI, Zabbix, SAM, zdroj: autor, [3][35][36][37][38][39][40][41][42][43][44][44]*

[1] Prostřednictvím pluginu, který je potřeba nainstalovat
[2] Prostřednictvím pluginu, který je již přítomný
[3] Prostřednictvím message gate operátora
[4] Prostřednictvím gsm modemu, připojenému k Zabbix serveru
[5] V kontextu monitorovacích nástrojů bývá označováno jako event handling
[6] Schopnost nástroje alertovat pouze kořenovou příčinu události
[7] Pouze v případe Enterprise upgradu

Shrnutí výsledků je zachyceno v *tabulce č. 8*. Výsledek doplněný o systém vah je poté uveden v *tabulce č. 9*.

| | Nástroj | | | |
|---|---|---|---|---|
| **Kategorie** | *Nagios Core* | *Nagios XI* | *Zabbix* | *SAM* |
| *Technologická stránka* | 4 | 6 | 6 | 4 |
| *Sběr dat* | 8 | 18 | 17 | 18 |
| *Monitoring* | 10 | 13 | 15 | 16 |
| *Zpracování dat* | 3 | 8 | 7 | 8 |
| *Uživatelská stránka* | 7 | 10 | 11 | 9 |
| *Celkem bodů* | 32 | 55 | 56 | 54 |

*Tabulka 8 – Výsledek srovnání Nagios Core, Nagios XI, Zabbix a SAM, zdroj: autor*

| | | *Nástroj* | | | |
|---|---|---|---|---|---|
| **Kategorie** | *Váha* | *Nagios Core* | *Nagios XI* | *Zabbix* | *SAM* |
| *Technologická stránka* | 0,05 | 0,2 | 0,3 | 0,3 | 0,2 |
| *sběr dat* | 0,25 | 2,25 | 4,5 | 4,25 | 4,5 |
| *monitoring* | 0,2 | 2 | 2,6 | 3 | 3,2 |
| *zpracování dat* | 0,1 | 0,3 | 0,8 | 0,7 | 0,8 |
| *Uživatelská stránka* | 0,4 | 2,8 | 4 | 4,4 | 3,6 |
| *Celkem bodů* | 1 | 7,55 | 12,2 | 12,65 | 12,3 |

*Tabulka 9 – Vážený výsledek srovnání Nagios Core, Nagios XI, Zabbix a SAM, zdroj: autor*

### 4.4.3 Závěr srovnání

Z uvedeného srovnání nástrojů v *kapitole 4.4.2*. vyplývá, že se svou funkcionalitou a nabízenými vlastnostmi velice podobají. Největším rozdílem je však úsilí, kterého je potřeba vynaložit, za účelem dosažení stejné úrovně dohledu s daným nástrojem. V tomto hraje roli hlavně způsob konfigurace, nutnost stahovat pluginy/šablony či přehlednost UI. Dále je velkým rozdílem cena a cenová politika a velikost komunity. Jako nejméně výhodný se jeví být Nagios Core, Vzhledem k jeho uživatelské nepřívětivosti a dostupnosti jiných open-source nástrojů např.: zmíněný Zabbix, je možné ho doporučit pouze k monitorování velice malého rozsahu. Jedinou výhodnou je možnost plynulého přechodu na placenou verzi Nagios XI. [45]

Nejvýhodněji se jeví být Zabbix, který poskytuje stejné funkcionality jako oba placené nástroje, avšak s výhodou možnosti bezplatného provozu (myšleno pouze z pohledu

licencí). V Závislosti na rozsahu implementace, se ale také odvíjí cena placené podpory. Proto vidím Zabbix jako vhodný pro malé až středně velké organizace, které požadují nástroj s širokou škálou funkcionalit a přívětivým UI, které ale nemohou pro monitoringu obětovat finanční zdroje. V případě většího rozsahu implementace doporučuji zakoupení některého stupně placené podpory.

Nagios XI a Solarwinds SAM se výrazně liší jen v cenně a cenové politice. Vzhledem k licencování počtu monitorovaných položek se SAM jeví být vhodný pro účely monitoringu buď malého rozsahu, nebo libovolného rozsahu, kde ovšem nehraje roli cena nástroje. Je to např.: monitoring datových center, kde má monitoring zásadní význam pro provoz služeb. Nagios XI potom vidím jako vhodný pro monitoring středního až velikého rozsahu, kde je při rozhodování brána v úvahu i cena.

# 5 Implementace monitorovacího nástroje Zabbix do produkčního prostředí reálné společnosti.

Následující část této práce se zabývá implementací a následným rozvojem konfigurace monitorovacího nástroje Zabbix do produkčního prostředí reálné společnosti. (Jméno společnosti bylo za účelem udržení anonymity změněno na ABC s.r.o, z tohoto důvodu jsou také začerněny některé části v uvedených screenshotech). Cílem následujících kapitol není komentovat konfiguraci v celém rozsahu, nebo popsat návod instalace nebo konfigurace Zabbixu. Cílem této části je naopak, na omezeném vzorku monitorovaných CI, obecně popsat přístupu k implementaci a konfiguraci monitorovacího nástroje, a to ve smyslu a rozsahu uvedeném v *kapitole 2.1-2.1.4.* Konfigurace je proto provedena a popsána s důrazem na pochopení monitorovaných CI, jejich metrik a správného určení prahových hodnot. Při konfiguraci triggerů je zohledněn význam a závažnost jednotlivých událostí, a to i z pohledu eliminace false positive událostí. Při konfiguraci alertingu bylo dbáno na eliminaci nerelevantních informací pro jednotlivé operátory.

Jednotlivé kapitoly tedy odpovídají myšlenkovému pochodu a aspektům, které byly brány v úvahu při výběru nástroje, definici požadavků a posléze samotné konfiguraci monitorování. Nejprve je proto provedena analýza prostředí a současného stavu. Na základě této analýzy jsou stanoveny a upřesněny požadavky na monitoring a monitorovací nástroj. Dále je uveden postup konfigurace monitorovacího nástroje Zabbix tak, aby výsledná konfigurace splnila stanovené požadavky a zároveň vyhovovala potřebám operátorů.

## 5.1 Analýza prostředí a současného stavu

IT infrastruktura společnosti je značně integrování do infrastruktury sesterské a potažmo mateřské společnosti. Pro účely této práce je popsána pouze část infrastruktury, na kterou byly kladeny požadavky na monitoring.

Aktivní síťové prvky např.: switche a routery jsou ve správě dodavatelské společnosti. Serverová část infrastruktury je postavena na virtualizační platformě VMware. Provoz virtuálních serverů potom zajišťují 3 fyzické hypervisor servery, na kterých je virtualizováno 67 virtuálních serverů. Z virtuálních serverů je 49 produkčních tzn.: jejich provoz je kritický pro provoz služeb. Mezi operačními systémy jsou zastoupeny jak MS Windows Servery (2008 – 2016), tak i Linux/Unix distribuce – převážně se jedná o Oracle, CentOS a RHEL. Vzhledem k různorodosti serverů je dále provedeno jejich logické seskupení, které je uvedeno v *kapitole 6.1.* V topologii sítě je vytvořena tzv.: Demilitarizovaná zóna, ve které jsou umístěny webové servery. Virtuální servery jsou průběžně replikovány do DR lokality, která je tvořena z jednoho fyzického DR serveru. Zálohu dat obstarává jeden fyzický backup server.

V době stanovení požadavků tvořilo IT oddělení 6 IT operátorů, kteří jsou rozdělení do následujících týmů:

1) Aplikační tým – zpracovává události a požadavky v rámci interních aplikací.
2) Infrastrukturní tým – zpracovává události a požadavky na úrovni zdrojů a obecných systémů.

V době stanovení požadavků byl již ve společnosti přítomný nástroj pro aplikační monitoring, který monitoruje dostupnost interních aplikací a systémů. Vykonává také definované scénáře a prověřuje správnost jejich provedení. Alerting je integrován s instant messaging clientem.

## 5.2 Definice požadavků

Za účelem doplnění monitoringu IT infrastruktury na úrovni zdrojů a bezpečnostního monitoringu byly vedením IT oddělení stanoveny požadavky v následujícím rozsahu:

1) Monitoring provozu
    a. Monitoring vybraných produkčních serverů ve smyslu dostupnosti a využití zdrojů.
2) Bezpečnostní monitoring
    a. Kompletní log management MS SQL serverů a Linux Oracle serverů.

Pro účely bezpečnostního monitoringu bylo využito služeb externího dodavatele. Postup implementace bezpečnostního monitoringu není tedy dále brán v úvahu.

### 5.2.1 Upřesnění požadavků – Monitoring provozu

Vzhledem k velice stručným požadavkům je potřeba upřesnit, jaké CI je za účelem jejich splnění monitorovat a způsob jakým je možné monitorování provádět. Pro výběr vhodného nástroje je také potřeba stanovit požadavky na samotný monitorovací nástroj.

Monitorování dostupnosti je možné pomocí protokolu ICMP a příkazu ping viz kapitola 3.3.3. Za tímto účelem není potřeba instalace agenta na daný prvek. Pomocí up-time prvku je dále možné zjistit, zdali byl prvek restartován. Up-time prvku je možné monitorovat agentem, SNMP, WMI nebo VMware API.

K monitorování využití zdrojů je potřeba monitorovat využití paměti RAM, HDD, CPU a provozu na síťovém rozhraní. Za tímto účelem je možné využít buď lokálně nainstalovaného agenta, SNMP v případě Linux/Unix i Windows serverů, WMI v případě Windows serverů (např. instance třídy *Win32_PerfFormattedData*), nebo tyto informace monitorovat na úrovni virtualizační platformy VMware pomocí VMware API.

Jako doplňující požadavek bylo autorem práce stanoveno monitorování switchů a routerů, které, i když jsou ve správě dodavatelské společnosti, je vhodné monitorovat alespoň z pohledu dostupnosti. Dále byl také autorem práce stanoven požadavek na monitorování vybraných služeb Windows serverů – např.: tisková služba na tiskovém serveru, nebo služba dhcp server na dhcp serveru apod. Tato problematika je dále popsána v *kapitole 6.5.*

### 5.2.2 Upřesnění požadavků – Způsob monitorování

Téměř všechny servery je možné monitorovat pomocí lokálně nainstalovaného agenta. Výjimku však tvoří Doménové řadiče, jejichž správa je v rukou mateřské společnosti, servery umístěné v DMZ (z důvodu zachování co nejnižší průchodnosti do interní sítě – agent potřebuje průchodný port TCP 10050) a VMware infrastruktura – ESXi a vCenter. Vzhledem k definovaným požadavkům je však ve všech případech možné využít monitorování pomocí ICMP a VMware API. Takto je zachována stejná úroveň dohledu i bez nutnosti lokálně nainstalovaného agenta.

### 5.2.3 Upřesnění požadavků – Monitorovací nástroj

Na základě požadavků popsaných v *kapitole 5.2.1* a *5.2.2* vznikly také požadavky na funkcionalitu monitorovacího nástroje. Při výběru vhodného nástroje byla proto požadována podpora monitoringu s agentem i bez agenta. V druhém případě byla požadována zejména podpora ICMP, SNMP, WMI a VMware API. Důvodem pro tento výběr byla také možnost poukázat na jiný způsob než na již několikrát, v kvalifikačních pracích, popsaný SNMP a také zájem autora o seznámení se s touto možností.

Vhodný monitorovací nástroj by také měl mít databázový backend pro účely uchování historických dat pro podporu analýzy a korelací. Dále také víceuživatelské UI, které by mělo být uživatelsky přívětivé a intuitivní. Při rozhodování byl preferován open-source nástroj s možností placené lokální podpory.

## 5.3 Výběr vhodného monitorovacího nástroje

Na základě srovnání nástrojů v *kapitole 4.4* probíhal užší výběr vhodného nástroje mezi Nagios XI a Zabbixem. S přihlédnutím k požadavkům na bezpečnostní monitoring viz *kapitola 5.2* a požadavkům uvedených v *kapitole 5.2.3*, zejména tedy možnosti lokální podpory, byl vybrán monitorovací nástroj Zabbix, který byl externím dodavatelem integrován do vlastního log management nástrojem. Oba nástroje byly tak součástí stejné implementace.

# 6 Konfigurace Zabbixu

Zabbix byl spolu s log management nástrojem externího dodavatele implementován v režimu Proof of Concept. Pro Zabbix tento režim představoval monitorování pěti serverů na základě oficiálních šablon pro OS Windows, Linux a VMware a zaškolení dvou operátorů. Následující kapitoly popisují proces rozvoje monitoringu Zabbixem tak, aby výsledná konfigurace splňovala definované požadavky a vyhovovala potřebám operátorů.

## 6.1 Logické seskupení prvků

Jako první byly vytvořeny tzn.: Host Groups, které slouží k logickému seskupení monitorovaných prvků (dále označeno jako „*Hosts*“). Rozdělení monitorovaných hostů bylo provedeno jednak na základě operačního sytému, tak na základě aplikací a systémů, které jsou na jednotlivých serverech implementovány. Tímto rozdělením byly zohledněny různorodé požadavky na zdroje, způsob provozu i budoucího vývoje současných i budoucích hostů.

Na základě těchto skupin je možné dále např.: agregovat počet událostí dle jednotlivých skupin hostů. Dále jsou skupiny hostů také využity pro účely uživatelských oprávnění (*viz kapitola 6.2*) a alertingu (*viz kapitola 6.4*). Jmenná konvence byla definována jako: *ABC #LOGIC.GROUP #HOST.TYPE např.: ABC Application Servers. Tabulka č. 10* zachycuje seznam vytvořených skupin hostů.

| ***Skupina Hostů*** | **Popis** |
|---|---|
| *ABC Application Servers* | MS Windows servery s interními aplikacemi |
| *ABC Citrix Servers* | Terminálové Citrix Servery |
| *ABC Exchange Servers* | MS Windows servery s MS Exchange rolí |
| *ABC Linux Servers* | Běžné Linux servery |
| *ABC Network Nodes* | Síťové prvky – switche, routery |
| *ABC Oracle Servers* | Servery s OS Oracle linux |
| *ABC SQL Servers* | Databázové MS Windows servery s MS SQL serverem |
| *ABC VMware Infrastructure* | VMware ESXi servery a vCenter servery |
| *ABC Windows Servers* | Běžné MS Windows servery |
| *Virtual Machines* | Servery monitorované přes VMware API (název skupiny nelze měnit) |

*Tabulka 10 - Host groups, zdroj: autor*

## 6.2 Uživatelské přístupy a oprávnění

Každému operátorovi IT byl přidělen přístup do nástroje Zabbix. operátoři byli rozděleni do logických skupin dle interního rozdělení pro řešení incidentů. Každé uživatelské skupině byly poté přidělena oprávnění ke čtení konkrétní skupiny hostů tzn.: možnost prohlížet zaznamenané události a konfiguraci. Výjimku tvoří pouze skupina *ABC Zabbix Admins*, která má absolutní práva do celého Zabbixu. Tímto způsobem bylo dosaženo odfiltrování nerelevantních informací pro konkrétní operátory. Uživatelské skupiny jsou také dále využity pro účely alertingu (viz kapitola 6.4).

*Tabulka č. 11* zachycuje vytvořené uživatelské skupiny s počtem přiřazených operátorů Jmenná konvence pro uživatelské skupiny byla definována jako: *ABC #LOGIC.GROUP Admins*.

| Uživatelská skupina | Počet operátorů |
|---|---|
| *ABC APP Admins* | 6 |
| *ABC Citrix Admins* | 3 |
| *ABC Exchange Admins* | 4 |
| *ABC Linux Admins* | 3 |
| *ABC Network Admins* | 4 |
| *ABC Oracle Admins* | 5 |
| *ABC SQL Admins* | 6 |
| *ABC VMware Admins* | 4 |
| *ABC Windows Admins* | 3 |
| *ABC Zabbix Admins* | 2 |

*Tabulka 11 - Uživatelské skupiny, zdroj: autor*

## 6.3 Šablony

Jedním způsobem, jak nakonfigurovat monitorování hostů, je provést nastavení přímo na daném hostu. Tento způsob není však příliš vhodný, jelikož se musí opakovat s každým přidaným hostem do Zabbixu. Šablony tzv.: Templates představují sadu definovaných entit, které mohou být hromadně aplikované na monitorované hosty. [3] V šabloně lze definovat:

- *Applications* – Slouží ke kategorizaci monitorovaných položek do logických kategorií např.: položky, které monitorují přidělenou paměť RAM a její využití, jsou seskupeny v aplikaci „*Memory*“. Umožňují další stupeň agregace.
- *Items* – Monitorované CI. Např.: Celková kapacita paměti RAM - *vm.memory.size[total].*
- *Triggers* – Nastavené Triggery.
- *Graphs* – Vizualizace dat do grafů.
- *Screens* – Vizualizace do tzv.: obrazovek – umožňuje seskupovat více grafů, tabulek atd.
- *Low Level Discovery* – Automatické vytvoření výše popsaných entit např. pro: filesystem, CPU, Windows služby.

Jako východisko k dalšímu nastavení, jak již bylo zmíněno, byly využity oficiální šablony pro OS Windows, Linux a VMware, které jsou také dostupné z komunitních stránek Zabbix Share[8]. Na základě těchto šablon byla pro každou skupinu hostů vytvořena oddělená šablona, ve které byly dle potřeby upraveny/přidány/odebrány některé z výše popsaných entit. *Tabulka č. 12* zachycuje vytvořené šablony. Pro šablony byla definována jmenná konvence jako: *Template ABC OS #HOST.OS #HOST.SYSTEM.*

| **Šablona** | **Přidělena hostům** |
|---|---|
| *Template ABC ICMP Ping* | Všem |
| *Template ABC OS Linux* | Ze skupiny ABC Linux Servers |
| *Template ABC OS Oracle* | Ze skupiny ABC Orace Servers |
| *Template ABC OS Windows* | Ze skupiny ABC Windows Servers |
| *Template ABC OS Windows Application* | Ze skupiny ABC Application Servers |
| *Template ABC OS Windows Citrix* | Ze skupiny ABC Windows servers |
| *Template ABC OS Windows Disk ** | Servery s kapacitou HDD větší jak 500GB |
| *Template ABC OS Windows Exchange* | Ze skupiny ABC Exchange Servers |
| *Template ABC OS Windows Services* | Hostům, na kterých jsou monitorovány Windows služby. Dále popsáno v kapitole 7.5 |
| *Template ABC OS Windows SQL* | Ze skupiny ABC SQL Servers |
| *Template ABC VMware ESXi* | ESXi serverům |
| *Template ABC VMware Vcenter* | Vcenter serverům |
| *Template ABC VMware VM* | Serverům monitorovaných přes VMware API |

*Tabulka 12 - Seznam definovaných šablon, zdroj: autor*

[8] www.share.zabbix.com

### 6.3.1 Monitorované CI

Aby bylo možné splnit požadavky uvedené v *kapitole 5.2.1*, je potřeba definovat, jaké konkrétní CI je za tímto účelem nutné monitorovat a jaké jsou jejich metriky.

1) **Dostupnost serveru**

Dostupnost síťového prvku je monitorována pomocí příkazu PING na protokolu ICMP. Host je dostupný (1), pokud vrátí alespoň jeden paket jako odpověď na *Echo Request.* Sleduje se i doba odezvy jednotlivých paketů v sekundách a procento ztracených paketů.

Up-time je monitorován ve tvaru dd:hh:ss

2) **Využití zdrojů** (Paměť RAM, HDD, CPU)

U paměti RAM je monitorována celková a dostupná kapacita v Bytech. Dostupná kapacita je také sledována v procentuálním vyjádření. Jako paměť je brán také Swap oddíl, u kterého je monitorována jeho celková a dostupná kapacita v Bytech.

Monitorování využití HDD probíhá na základě celkové, dostupné a využité kapacity v Bytech. V případě dostupné kapacity je sledováno i procentuální vyjádření. V případě linux serverů je vhodné také monitorovat disponibilní kapacitu pro i-uzle, které uchovávají metadata o adresářích a souborech

Pro účely monitorování využití CPU je sledováno jeho procentuální využití. Vhodné je také monitorovat zátěž procesoru tzv.: CPU load. CPU load udává, zdali se vytváří fronta procesů ke zpracování v důsledku nedostatečné kapacity CPU. Metrika CPU loadu je teoreticky kladné reálné číslo, které by se však mělo pohybovat v mezi 0.00 – 1.00. V případě, že je naměřená hodnota vyšší než 1.00, znamená to, že se tvoří „fronta“. Dlouhodobě vysoký CPU load indikuje nedostatečný výpočetní výkon. [45]

U síťového rozhraní je monitorován vstupní, respektive výstupní provoz v bps.

Vyhodnocení stavu Windows služeb probíhá na základě číselných kódů, uvedených níže:

*„0 - running, 1 - paused, 2 - start pending, 3 - pause pending, 4 - continue pending, 5 - stop pending, 6 - stopped, 7 – unknown“* [14]

*Tabulka č. 13* zachycuje shrnutí monitorovaných CI pro standardní Windows servery. Tabulka obsahuje stanovený požadavek na monitorování, kategorizaci položek do aplikací, název položky, výraz položky v Zabbixu, interval sběru, metriku a způsob sběru.

| **Požadavek** | ***Aplikace*** | ***Položka*** | ***key*** | ***Interval*** | ***Metrika*** | ***Sběr*** |
|---|---|---|---|---|---|---|
| *Dostupnost* | *ICMP* | Ping | icmpping | 1m | 1/0 | icmp |
| | | Ping loss | icmppingloss | 5m | % | icmp |
| | | Ping response time | icmppingsec | 5m | s | icmp |
| | *General* | System uptime | system.uptime | 1m | hh:mm:ss | agent |
| *Paměť* | *Memory* | Total memory | vm.memory.size[total] | 1h | B | agent |
| | | Free memory | vm.memory.size[free] | 1m | B | agent |
| | | Available memory % | vm.memory.size[pavailable] | 1m | % | agent |
| | | Total swap space | system.swap.size[,total] | 1h | B | agent |
| | | Free swap space | system.swap.size[,free] | 1m | B | agent |
| *HDD* | *Filesystem* | Total space | vfs.fs.size[{#FSNAME}, total] | 1h | B | agent |
| | | Used space | vfs.fs.size[{#FSNAME}, used] | 1m | B | agent |
| | | Free space | vfs.fs.size[{#FSNAME}, free] | 1m | B | agent |
| | | Free space % | vfs.fs.size[{#FSNAME}, pfree] | 1m | % | agent |
| *CPU* | *CPU* | CPU load (1 min average) | system.cpu.load[percpu,avg1] | 1m | n | agent |
| | | CPU load (5 min average) | system.cpu.load[percpu,avg5] | 1m | n | agent |
| | | CPU load (15 min average) | system.cpu.load[percpu,avg15] | 1m | n | agent |
| | | CPU Util | perf_counter[\Processor(_Total)]\% Processor Time] | 30s | % | agent |
| *Provoz na síťových rozhraních* | *Network interfaces* | Incoming net traffic | Net.if.in[{#IFNMAE}] | 1m | bps | agent |
| | | Outgoing net traffic | Net.if.out[{#IFNMAE}] | 1m | bps | agent |

*Tabulka 13 - Seznam monitorovaných CI u standardních Windows serverů, zdroj: autor*

### 6.3.2 Prahové hodnoty

Pro korektní stanovení prahových hodnot byl brán ohled na potřeby i budoucí vývoj jednotlivých serverů. Při definici bylo preferováno procentuální vyjádření hodnot, jelikož sjednocuje jednoty do jednoduché a obecně známé podoby, ze které je možné ihned odvodit, zdali je aktuální stav položky v přijatelných mezích.

1) **Dostupnost**

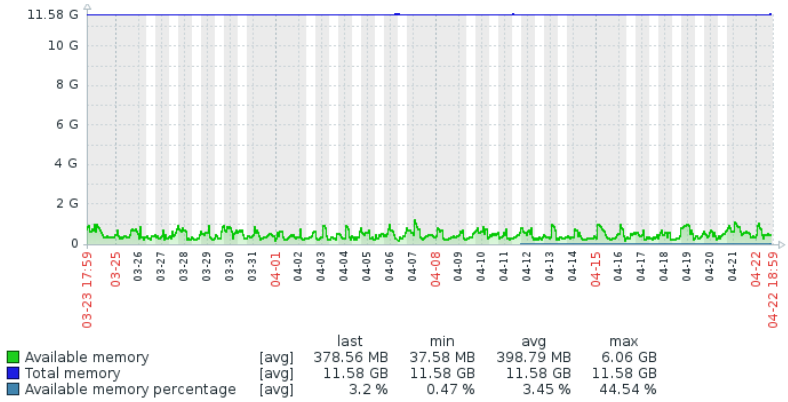
Za účelem monitorování dostupnosti prvku byly ponechány definované prahové hodnoty z oficiální šablony tzn.:

a. Dostupnost – Alespoň jeden ze tří po sobě jdoucích dotazů na dostupnost prvku musí být úspěšní tzn.: vrátí *„1“*.
b. Odezva – Průměrná doba odezvy prvku za 5 minut by neměla být vyšší než 0.15 sekund.
c. Ztráta paketů – Průměrná ztráta paketů za 5 minut by měla být vyšší než 20 paketů.
d. Restart – Uptime prvku by se oproti poslední hodnotě neměl zmenšit.

2) **Využití zdrojů** (Paměť RAM, SWAP, HDD, CPU, Síťové rozhraní)

Za účelem monitorování využití zdrojů serverů bylo do šablon doplněno monitorování procentuálního vyjádření využití RAM, HDD, CPU a stanoveny níže uvedené prahové hodnoty.

a. **Paměť** – Pro standardní severy by nemělo být dostupné paměti RAM po dobu 10, respektive 5 minut méně než 10%, respektive 5% a méně než 500 MB Swapu.
    i. Výjimku zde tvoří databázové MS Windows a databázové Linux Oracle servery viz *obrázek č. 4,* které si alokují více paměti RAM bez dopadu na výkon (viz obrázek č. 4). Pro tyto Servery by nemělo být dostupné paměti RAM po dobu 1h méně než 1% a taktéž 500MB Swapu.




*Obrázek 4 – Dlouhodobé využití RAM databázového serveru, zdroj: autor*

b. **HDD** – U většiny serverů je kapacita HDD omezena na 50 GB, při potřebě větší kapacity je využito fileserveru. U standardních serverů by proto nemělo být méně než 10%, respektive 5% dostupné kapacity HDD. Výjimku zde však tvoří následující servery:

   i. Servery s větší kapacitou HDD jak 500 GB, u kterých byly prahové hodnoty posunuty na 2% a 1% dostupné kapacity.
   ii. MS Exhchange servery, pro které byly prahové hodnoty stanoveny na 15% a 11% dostupné kapacity. A to za účelem zajištění včasného uvolnění kapacity bez dopadu na odesílání, respektive příjem emailů.
   iii. Linux Oracle servery, pro které byly prahové hodnoty stanoveny, na základě doporučení dodavatele, na 20% a 5% dostupné kapacity.
   iv. Pro i-uzle u Linux serverů jsou ponechány stejné prahové hodnoty tzn.: 20% respektive 5%.

c. **CPU**

   i. Pro CPU load byla ponechána prahová hodnota z výchozí šablony tzn.: průměrný CPU load za 5 minut by neměl být vyšší než 5.
   ii. Celkové využití CPU serveru by nemělo být po dobu 15 minut vyšší než 95%, dále by také dlouhodobé využití CPU (po dobu 8h) nemělo být vyšší jak 35%.
   iii. U MS Exchange a Linux Oracle serverů bylo také doplněno monitorování využití CPU pro jednotlivá jádra. Prahová hodnota byla ponechána stejně jako v bodě ii.

d. **Síťové rozhraní**

   i. Pouze pasivní monitorování, není potřeba stanovit prahovou hodnotu

### 6.3.3 Triggering

Jak bylo popsáno v *kapitole 2.1.3*, trigger představuje mechanismus pro zahájení kontrolního opatření. V Zabbixu jsou triggery definovány pomocí monitorovaných CI, definovaných prahových hodnot a významu události. Každému triggeru je tedy přidělen význam (v kontextu Zabbixu se jedná o závažnost), podle toho, jaké riziko představuje spuštění daného triggeru pro provoz služeb. Na základě informací popsaných v *kapitole 2.1.3* a interní analýzy byly stanoveny níže uvedené stupně závažnosti:

1) *Debug* – Ladění položek, využito v případě log managementu.
2) *Informational* – Změny informačního charakteru. Nevyžadují vykonání kontrolní akce.
3) *Warning* – Nepředstavují bezprostřední riziko pro provoz služeb. Nemusí vždy vyžadovat vykonání kontrolní akce. Kontrolní akce může být vykonána s časovou prodlevou.
4) *Major* – Představují bezprostřední riziko pro provoz služeb. Kontrolní akci je potřeba vykonat přednostně.
5) *Critical* – Aktuální selhání nebo výpadek služby. Kontrolní akce musí být vykonána okamžitě.
6) *Fatal* – Stejná definice jako Critical, ovšem s rozdílem, že je tento stupeň přidělen jen triggerům, které indikují zhroucení celého systému tzn.: nedostupnost ESXi nebo databáze ERP systému.

Jelikož jsou názvy triggerů využity pro účely alertingu (dále popsáno v *kapitole 6.4*), musí název triggeru jednoznačně identifikovat událost. Za tímto účelem byly všechny triggery přejmenovány dle jmenné konvence:

*#HOST.NAME: #CI #TRIGGER.CONDITION.SIMPLIFIED* *(#TIME.PERIOD) např.: *Server1: Free memory < 10%* nebo *Server2: CPU util > 95% for 15min.*

*Tabulka č. 14* zachycuje shrnutí názvů definovaných triggerů pro standardní MS Windows servery, včetně kategorizace do aplikací, přidělené závažnosti a názvu triggeru.

| Aplikace | Závažnost | Název Triggeru |
|---|---|---|
| Filesystem | MAJOR | {HOST.NAME}: Free disk space on {#FSNAME} < 5% |
| | WARN | {HOST.NAME}: Free disk space on {#FSNAME} < 10% |
| Memory | CRITICAL | {HOST.NAME}: Free swap space < 500M |
| | MAJOR | {HOST.NAME}: Free memory < 5% |
| | WARN | {HOST.NAME}: Free memory < 10% |
| CPU | CRITICAL | {HOST.NAME}: CPU util > 95% for 15min |
| | WARN | {HOST.NAME}: CPU util > 35% for 8hours |
| | MAJOR | {HOST.NAME}: CPU load > 5 for 5min |
| ICMP | Critical | {HOST.NAME}: is unavailable by ICMP |
| | WARN | {HOST.NAME}: Response time > 0.15s for 5min |
| | WARN | {HOST.NAME}: Ping loss > 20% for 5min |
| General | MAJOR | {HOST.NAME}: Has just been restarted |

*Tabulka 14 - Seznam definovaných triggerů pro standardní MS Windows, zdroj: autor*

V *kapitole 2.1.4* byla naznačena problematika časných alertů. Za účelem ošetření případu, kdy se monitorovaná CI pohybuje na hraně prahové hodnoty a často jí přesahuje, následně se však ihned vrátí do definovaných mezí (tzv.: *flapping)*, v důsledku čehož vznikají false positive události, byly triggery doplněny o definici tzv.: *Hystereze*.

Hystereze rozšiřuje výraz triggeru o podmínku, kdy nastane návrat CI z problémového stavu do bezproblémového stavu. Například problémový stav u využití RAM serveru nastane, je-li dostupné paměti RAM v průměru za 5 minut méně než 5%. Návrat z problémového stavu však nastane i v případě, je-li v průměru za 5 minut dostupné i 5,01% paměti RAM. Pomocí hystereze bylo definováno, že návrat z problémového stavu nastane až v případě, je-li po dobu 10 min v průměru dostupné více jak 10% paměti RAM. Výrazy definic triggerů pro standardní MS Windows servery, doplněných o hysterezi, jsou zachyceny v *tabulce č. 15*. Obdobně byla nastavena hystereze i v ostatních šablonách pro ostatní servery.

| Název triggeru | Výraz triggeru |
|---|---|
| {HOST.NAME}: Free disk space on {#FSNAME} < 5% | {Template ABC OS Windows:vfs.fs.size[{#FSNAME},pfree].last(0)}<5 |
| {HOST.NAME}: Free disk space on {#FSNAME} < 10% | {Template ABC OS Windows:vfs.fs.size[{#FSNAME},pfree].las(0}<10 |
| {HOST.NAME}: Free swap space < 500M | {Template ABC OS Windows:system.swap.size[,free].last(0)}<500M |
| {HOST.NAME}: Free memory < 5% | ({TRIGGER.VALUE}=0 or {Template ABC OS Windows:vm.memory.size[pavailable].avg(5m)}<5) and {TRIGGER.VALUE}=1 and {Template ABC OS Windows:vm.memory.size[pavailable].avg(10m)}<10 |
| {HOST.NAME}: Free memory < 10% | ({TRIGGER.VALUE}=0 or {Template ABC OS Windows:vm.memory.size[pavailable].avg(10m)}<10) and {TRIGGER.VALUE}=1 and {Template ABC OS Windows:vm.memory.size[pavailable].avg(15m)}<15 |
| {HOST.NAME}: CPU util > 95% for 15min | (({TRIGGER.VALUE}=0 and {Template ABC OS Windows:perf_counter[\Processor(_Total)\% Processor Time].avg(15m)}>95) or {TRIGGER.VALUE}=1) and {Template ABC OS Windows:perf_counter[\Processor(_Total)\% Processor Time].avg(10)}>75 |
| {HOST.NAME}: CPU util > 35% for 8hours | {Template ABC OS Windows:perf_counter[\Processor(_Total)\% Processor Time].avg(8h)}>35} |
| {HOST.NAME}: CPU load > 5 for 5min | {Template ABC OS Windows:system.cpu.load[percpu,avg1].avg(5m)}>5 |
| {HOST.NAME}: is unavailable by ICMP | {Template ABC ICMP Ping:icmpping.max(#3)}=0 |
| {HOST.NAME}: Response time > 0.15s for 5min | {Template ABC ICMP Ping:icmppingsec.avg(5m)}>0.15 |
| {HOST.NAME}: Ping loss > 20 for 5min | {Template ABC ICMP Ping:icmppingloss.min(5m)}>20 |
| {HOST.NAME}: Has just been restarted | {Template ABC OS Windows:system.uptime.change(0)}<0 |

*Tabulka 15 - Seznam definovaných triggerů pro MS Windows, zdroj: autor*

Pro databázový Linux Oracle server jsou pak triggery zachyceny v *tabulce č. 16.*

| **Název triggeru** | **Výraz triggeru** |
|---|---|
| {HOST.NAME}: Free disk space on {#FSNAME} < 5% | {Template ABC OS Linux Oracle:vfs.fs.size[{#FSNAME},pfree].last(0)}<5 |
| {HOST.NAME}: Free disk space on {#FSNAME} < 20% | {Template ABC OS Linux Oracle:vfs.fs.size[{#FSNAME},pfree].last(0)}<20 |
| {HOST.NAME}: Free inodes on {#FSNAME} < 5% | {Template ABC OS Linux Oracle:vfs.fs.inode[{#FSNAME},pfree].last(0)}<5 |
| {HOST.NAME}: Free inodes on {#FSNAME} < 20% | {Template ABC OS Linux Oracle:vfs.fs.inode[{#FSNAME},pfree].last(0)}<20 |
| {HOST.NAME}: Free swap space < 500M | {Template ABC OS Linux Oracle:system.swap.size[,pfree].last(0)}<500 |
| {HOST.NAME}: Free memory < 1% | (({TRIGGER.VALUE}=0 and {Template ABC OS Linux Oracle:vm.memory.size[pavailable].avg(1h)}<1) or {TRIGGER.VALUE}=1) and {Template ABC OS Linux Oracle:vm.memory.size[pavailable].avg(1h)}<2 |
| {HOST.NAME}: CPU util on core {#CPU.NUMBER} > 95% for 15min | (({TRIGGER.VALUE}=0 and {Template ABC OS Linux Oracle:system.cpu.util[{#CPU.NUMBER}, system, avg1].avg(15m)}>95) or {TRIGGER.VALUE}=1) and {Template ABC OS Linux Oracle:system.cpu.util[{#CPU.NUMBER}, system, avg1].avg(10m)}>75 |
| {HOST.NAME}: CPU util on core {#CPU.NUMBER} > 35% | {TRIGGER.VALUE}=0 and {Template ABC OS Linux Oracle:system.cpu.util[{#CPU.NUMBER}, system, avg1].avg(8h)}>35 |
| {HOST.NAME}: CPU load > 5 for 5min | {Template ABC OS Linux Oracle:system.cpu.load[percpu,avg1].avg(5m)}>5 |
| {HOST.NAME}: is unavailable by ICMP | {Template ABC ICMP Ping:icmpping.max(#3)}=0 |
| {HOST.NAME}: Response time > 0.15s for 5min | {Template ABC ICMP Ping:icmppingsec.avg(5m)}>0.15 |
| {HOST.NAME}: Ping loss > 20 for 5min | {Template ABC ICMP Ping:icmppingloss.min(5m)}>20 |
| {HOST.NAME}: Has just been restarted | {Template ABC OS Windows:system.uptime.change(0)}<0 |

*Tabulka 16 - Seznam definovaných triggerů pro DB Linux Oracle, zdroj: autor*

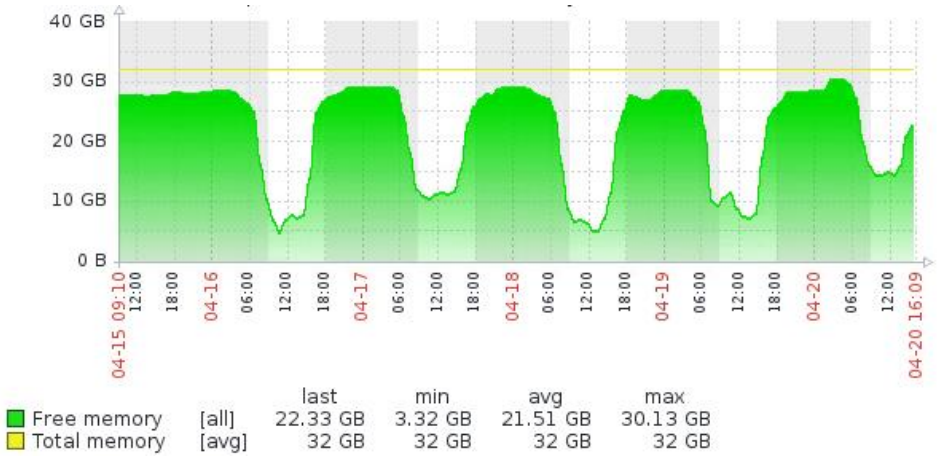
Pro servery monitorované přes VMware API bylo potřeba definici triggerů trochu poupravit. Triggery zachyceny v *tabulce č. 17.*

| **Název triggeru** | **Výraz triggeru** |
|---|---|
| {HOST.NAME}: Free disk space on {#FSNAME} < 5% | {Template ABC VMware VM:vmware.vm.vfs.fs.size[{$URL},{HOST.HOST},{#FSNAME},pfree].last()} <5 |
| {HOST.NAME}: Free disk space on {#FSNAME} < 10% | {Template ABC VMware VM:vmware.vm.vfs.fs.size[{$URL},{HOST.HOST},{#FSNAME},pfree].last()} <10 |
| {HOST.NAME}: Free memory < 5% | (({TRIGGER.VALUE}=0 and {Template ABC VMware VM:vmware.vm.memory.size.usage.guest.percent[{$URL},{HOST.HOST}].avg(5m)}>95) or {TRIGGER.VALUE}=1) and {Template ABC VMware VM:vmware.vm.memory.size.usage.guest.percent[{$URL},{HOST.HOST}].avg(10m)}>90 |
| {HOST.NAME}: Free memory < 10% | (({TRIGGER.VALUE}=0 and {Template ABC VMware VM:vmware.vm.memory.size.usage.guest.percent[{$URL},{HOST.HOST}].avg(10m)}>90) or {TRIGGER.VALUE}=1) and {Template ABC VMware VM:vmware.vm.memory.size.usage.guest.percent[{$URL},{HOST.HOST}].avg(15m)}>85 |
| {HOST.NAME}: Swapped memory > 3.5G | {Template ABC VMware VM: vmware.vm.memory.size.swapped [{$URL},{HOST.HOST}}>3.5G |
| {HOST.NAME}: CPU util > 95% for 15min | (({TRIGGER.VALUE}=0 and {Template ABC VMware VM:vmware.vm.perfcounter[{$URL},{HOST.HOST},"cpu/usage[average]"].avg(15m)}>95) or {TRIGGER.VALUE}=1) and {Template ABC VMware VM:vmware.vm.perfcounter[{$URL},{HOST.HOST},"cpu/usage[average]"].avg(10m)}>75 |
| {HOST.NAME}: CPU util > 35% for 8hours | {Template ABC VMware VM:vmware.vm.perfcounter[{$URL},{HOST.HOST},"cpu/usage[average]"].avg(8h)}>35 |
| {HOST.NAME}: is unavailable by ICMP | {Template ABC VMware VM:icmpping.max(#3)}=0 |
| {HOST.NAME}: Ping loss > 20% for 5min | {Template ABC VMware VM:icmppingloss.min(5m)}>20 |
| {HOST.NAME}: Response time > 0.15s for 5min | {Template ABC VMware VM:icmppingsec.avg(5m)}>0.15 |
| {HOST.NAME}: has just been restarted | {Template ABC VMware VM:vmware.vm.uptime[{$URL},{HOST.HOST}].change(0)}<0 |

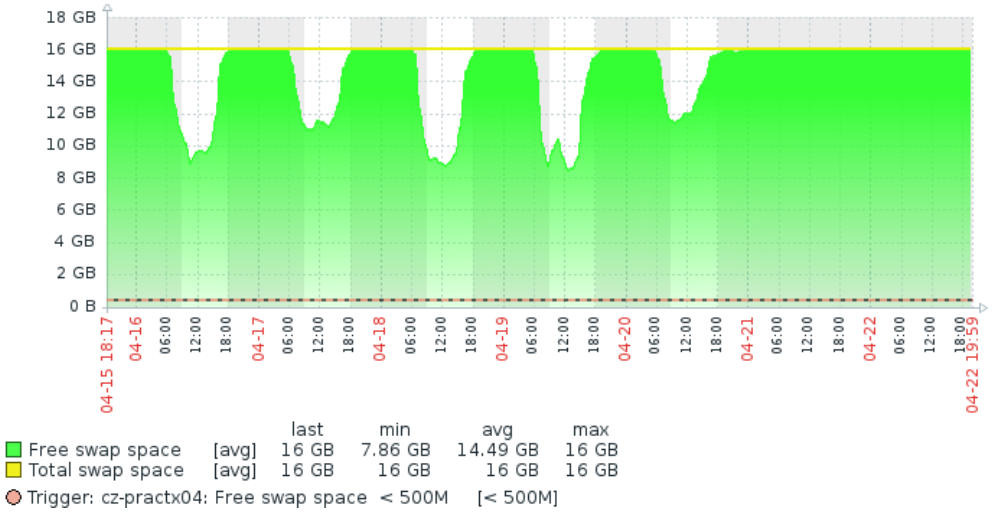
*Tabulka 17 - Seznam definovaných triggerů pro servery monitorované přes VMware API, zdroj: autor*

### 6.3.4 Vizualizace dat

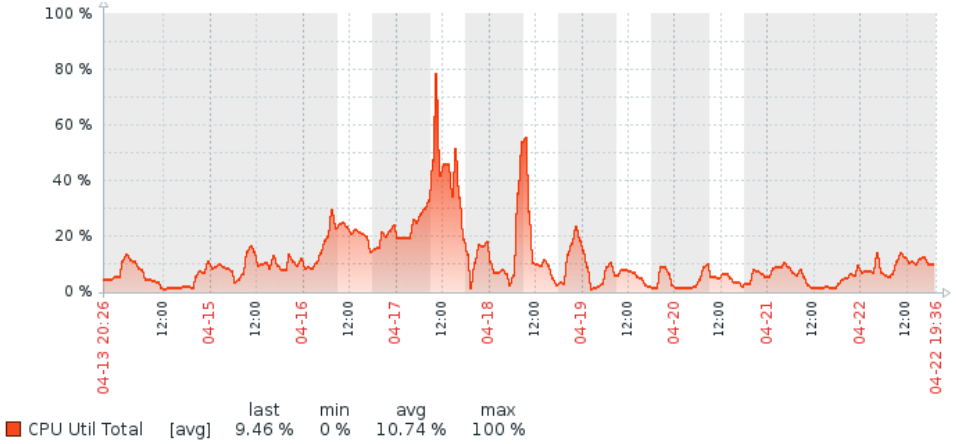
Za účelem vizuální podpory, nejen při řešení problémů, ale i pro zhodnocení kapacit monitorovaných serverů, byly doplněny grafy pro sledování využití RAM, SWAP a CPU viz *obrázek č. 5, č 6* respektive *obrázek č 7*. Graf pro sledování využití HDD a provozu na síťových rozhraních byl již přítomný ve výchozí šabloně.




*Obrázek 5 - Graf využití RAM serveru, zdroj: autor*



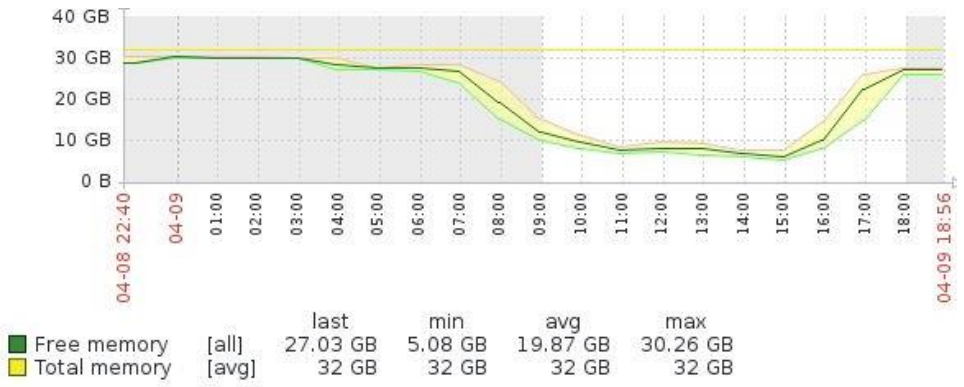

*Obrázek 6 - Graf využití SWAP serveru, zdroj: autor*




*Obrázek 7 - Graf využití CPU serveru, zdroj: autor*

### 6.3.5 Retence dat

Aby bylo možné využívat a dále analyzovat i historicky nasbíraná data pro účely např.: odhadu budoucího trendu kapacit, jsou nasbíraná data ukládána do databáze. Aby však zůstala velikost databáze v rozumných mezí, byla nastavena retence. Pro účely krátkodobé analýzy jsou uchovávána všechny data za poslední týden. Pro účely dlouhodobého uchování je uchováván pouze trend vývoje dané položky za 365 dní. V trendu jsou uchovávány jen průměrná, minimální a maximální hodnota za hodinu. Pokud jsou v grafu zobrazena historická data, zobrazí se tyto agregovaná dat viz *obrázek č. 8*.




*Obrázek 8 - historický trend vývoje využití paměti RAM, zdroj: autor*

## 6.4 Alerting

Alerting byl nakonfigurován tak, aby byli jednotliví operátoři informováni pouze o relevantních událostech tzn.: operátor ze skupiny *ABC Windows Admins* obdrží pouze události ze skupiny hostů *ABC Windows Servers*. Obdobně byly vytvořeny definice pro všechny skupiny hostů a uživatelů. Jednotlivé definice alertů byli pojmenovány dle jmenné konvence: *Report events to #USER.GROUP. Tabulka č. 18* zachycuje seznam jednotlivých alertů. Pro eskalační pravidla byla pak definována jmenná konvence: *Escalate #SEVERITY events to ServiceDesk.* Eskalace je dále popsána v *kapitole 6.4.1.*

| Seznam Definovaných alertů | Události ze skupiny hostů | Alertuje uživatele |
|---|---|---|
| *Report events to ABC Application Admins* | ABC Application Servers | ABC Windows Admins |
| *Report events to ABC Citrix admins* | ABC Citrix Servers | ABC Citrix Admins |
| *Report events to ABC Exchange Admins* | ABC Exchange Servers | ABC Exchange Admins |
| *Report events to ABC Linux Admins* | ABC Linux Servers | ABC Linux Admins |
| *Report events to ABC Network Admins* | ABC Network nodes | ABC Network Admins |
| *Report events to ABC Oracle Admins* | ABC Oracle Servers | ABC Oracle Admins |
| *Report events to ABC SQL Admins* | ABC SQL Servers | ABC SQL Admins |
| *Report events to ABC VMware Admins* | ABC VMware Infrastructures | ABC VMware Admins |
| *Report events to ABC Windows Admins* | ABC Windows Servers, Virtual Machines | ABC Windows Admins |
| *Report events to ABC Zabbix Admins* | ABC Zabbix Server | ABC Zabbix Admins |
| *Escalate critical events to ServiceDesk* | Všech výše popsaných | ServiceDesk |
| *Escalate major events to ServiceDesk* | Všech výše popsaných | ServiceDesk |

*Tabulka 18 - Seznam definovaných alertů, zdroj: autor*

Samotné tělo Alertu bylo dále upraveno tak, aby jednoznačně identifikovalo danou událost a obsahovalo další relevantní informace pro určení přesného významu a závažnosti události. *Obrázek č. 9* zachycuje konfiguraci názvu, předmětu a těla alertu. Zachycené nastavení na *Obrázku č. 9* definuje alert s názvem „*Report events to Windows Admins*“. Předmět i tělo je komponováno za pomoci maker, jejichž popis je následný: [14]

- *{TRIGGER.NAME}* – Název triggeru, který zapříčinil odeslání alertu (viz *kapitola 6.3.3*)
- *{TRIGGER.STATUS}* – Aktuální stav triggeru – nabývá hodnot „OK“ nebo „PROBLEM“
- *{TRIGGER.SEVERITY}* – Přidělená závažnost danému triggeru
- *{TRIGGER.DESCRIPTION}* – Popis triggeru
- {EVENT.TIME} – Čas vzniku události
- {EVENT.DATE} – Datum vzniku události

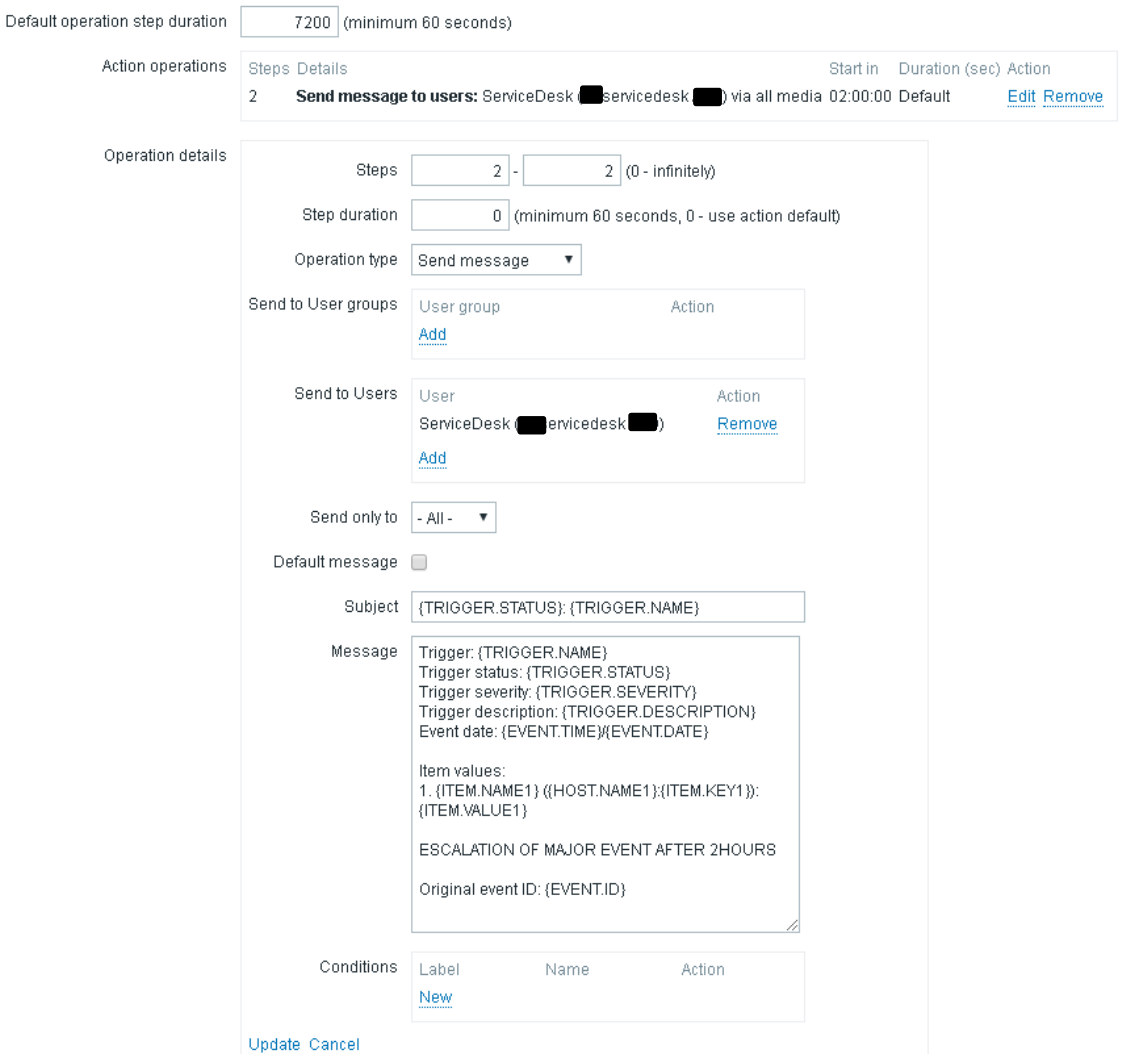
- *{ITEM.NAME#}* – Název sledované položky
- *{ITEM.VALUE#}* – Hodnota sledované položky
- *{HOST.NAME#}* – Název hostu, na kterém došlo k události
- {EVENT.ID} – Přidělené ID událost




*Obrázek 9 - Konfigurace alertu, zdroj: autor*

### 6.4.1 Eskalace

V *kapitole č. 2.1.4* byla nastíněna možnost integrovat alerting s interním ServiceDeskem. Aby nedocházelo k zakládání ticketů i pro nevýznamné události, byly definovány dvě obecná eskalační pravidla, která jsou již zachycena v *tabulce č. 18* – jedná se o „*Escalate major events to ServiceDesk*“ a „*Escalate critical events to ServiceDesk*“. Tyto pravidla eskalují události s vyšší závažností do interního ServiceDesku v případě, trvají-li déle jak 2h, respektive 20min. Pokud tedy byl první, bezprostřední alert ignorován a událost nebyla řešena, dojde k odeslání alertu na emailovou adresu ServiceDesku a založení ticketu. Eskalace však neproběhne, pokud byla událost uznána za nevýznamnou a označena tak v Zabbixu (tzv.: *acknowledged*). *Obrázek č. 10* zachycuje eskalaci major událostí.




*Obrázek 10 - Eskalace major událostí do ServiceDesku, zdroj: autor*

## 6.5 Monitoring Windows služeb

Jako jeden z doplňujících požadavků na monitoring bylo monitorování stavu vybraných Windows služeb. Za tímto účelem bylo vytvořeno i automatické opatření v podobě spuštění zastavené služby.

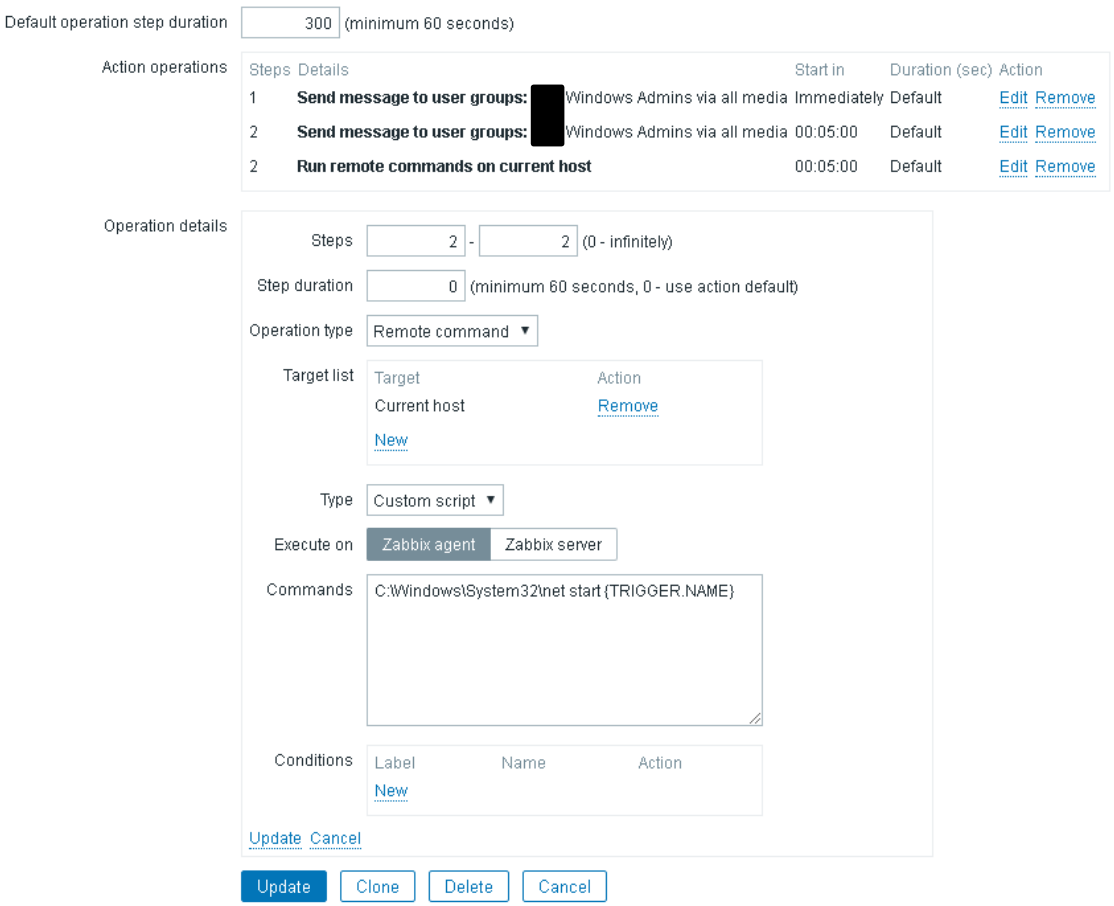
Pro dodržení přehlednosti a konvencí popsaných v *kapitolách 6.1-6.4*, byla vytvořena nová šablona s názvem *Template ABC OS Windows Services,* ve které byla vytvořena aplikace *Windows services*. Tato šablona byla posléze přidělena všem hostům, u kterých jsou monitorovány některé Windows služby. Na daném hostu byla poté vytvořena položka *service.info[#SERVICE.NAME],* která byla přiřazena do aplikace *Windows Services*. Na základě této položky byl vytvořen trigger *{HOST.NAME:service.info[{#SERVICE.NAME}].last()}=6*, který vyvolá spuštění kontrolního opatření, nachází-li se daná služba ve stavu *stopped*.

Aby bylo možné nastavit pouze jeden univerzální příkaz pro všechny monitorované služby, byla porušena jmenná konvence u daných triggerů. Tyto triggery byly pojmenovány dle monitorovaných služeb např.: „*Spooler*“ v případě tiskové služby. V příkazu pro spuštění služby pak bylo využito makro {TRIGGER.NAME}. Výsledný příkaz pro spuštění služby má podobu:

```
c:\windows\system32\net start {TRIGGER.NAME}
```

Aby však zůstala zachována jmenná konvence alertů, byl každému triggeru doplněn popis ve tvaru *{HOST.NAME}: #SERVICE.NAME is not running.* Název alertu je pak vytvořen z maker *{TRIGGER.STATUS}: {TRIGGER.DESCRIPTION}.*

I když je možné vykonat automatické opatření bezprostředně po výskytu dané události, je žádoucí, aby byla zachována kontrola nad spuštěnými příkazy. Za tímto účelem bylo nastaveno eskalační pravidlo, které nejprve alertuje operátory a až v případě, trvá-li tento stav 5 minut, spustí výše uvedený příkaz. O spuštění příkazu jsou operátoři taktéž informováni. Toto eskalační pravidlo je zachyceno na *obrázku č. 11.*




*Obrázek 11 - Automatické spuštění zastavené Windows služby, zdroj: autor*

## 6.6 Testování

Otestování výše popsaného nastavení bylo provedeno na vyhrazené testovací stanici. Metodika testování spočívala v umělém vytvoření podmínek, které zapříčinili překročení definovaných prahových hodnot viz *kapitola 6.3.2*. Metodika a způsob testování je popsán níže. Pro účely testování byla vytvořena samostatná logická skupina, jejímž členem byla jen testovací stanice. Stanici byla přidělena testovací šablona *Template ABC OS Windows TEST*. Aby nebyli o testování alterování ostatní operátoři, byla také vytvořena samostatná definice alertu. Takto byla otestována i konfigurace alertingu. Výsledky tesů jsou vždy uvedeny pod daným požadavkem na monitoring. Jedná se o *tabulky výsledků č. 19, 20, 21 a 22* a *obrázek č. 12*. Anonymizovaný výpis eskalačního pravidla pro spuštění zastavené tiskové služby je uveden v *příloze č. 3*.

1) **Dostupnost**

a. Dostupnost – Stanice byla odpojena od sítě
b. Odezva – Pakety Echo reply byly odeslány se zpožděním [46]
c. Ztráta paketů – Pakety Echo Request byly zahozeny [46]
d. Restart – stanice byla restartována

| Požadavek | Simulace události | Vznik a detekce událost | Poznámka |
|---|---|---|---|
| *Dostupnost* | 14:15 | 14:18:28 | 3 po sobě jdoucí dotazy |
| *Odezva* | 14:25 | 14:31:28 | Průměrná hodnota za 5 minut |
| *Ztráta paketů* | 14:40 | 14:45:28 | Průměrná hodnota za 5 minut |
| *Restart* | 21:30 | 21:34:40 | |

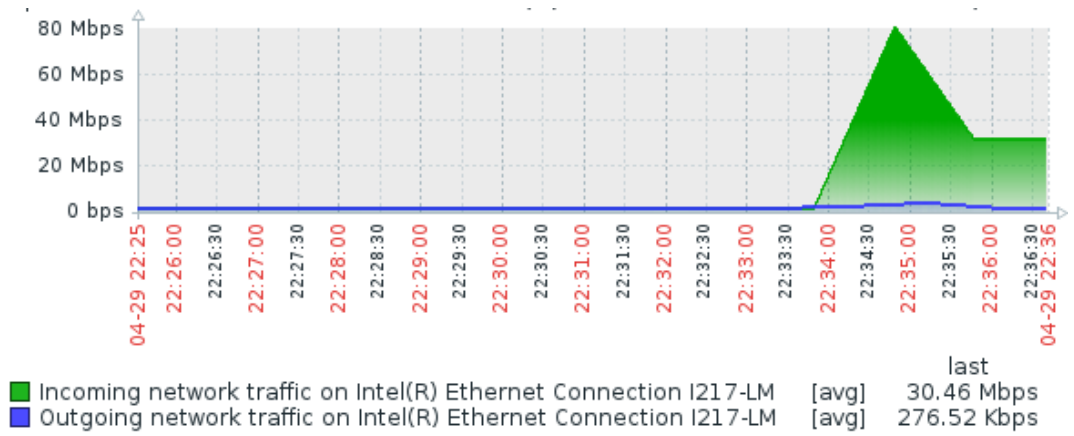
*Tabulka 19 – Tabulka výsledů testů dostupnosti, zdroj: autor*

2) **Využití zdrojů** (Paměť RAM, HDD, CPU)

a. Paměť
    i. RAM – Spuštěn zátěžový test [47]
    ii. SWAP – Snížení prahové hodnoty
b. HDD – Snížení prahové hodnoty
c. CPU
    i. CPU load – spuštěn zátěžový test [47]
    ii. Využití CPU – spuštěn zátěžový test [47]
e. Provoz na síťovém rozhraní – Kopie souboru po sítí

| Požadavek | Simulace události | Vznik a detekce události | Poznámka |
|---|---|---|---|
| *SWAP* | 15:15 | 15:16:37 | |
| *RAM* | 16:20 | 15:26:42 | Průměrná hodnota za 5 minut |
| *HDD* | 22:05 | 22:07:32 | |
| *CPU load* | 22:10 | 22:17:35 | Průměrná hodnota za 5 minut |
| *CPU Util* | 22:10 | 22:27:32 | Průměrná hodnota za 15 minut |
| *Provoz na síťovém rozhraní* | 22:33 | 22:33:45 | *Viz obrázek č. 12* |

*Tabulka 20 – Tabulka výsledků testů využití zdrojů, zdroj: autor*




*Obrázek 12 - Graf provozu na síťovém rozhraní, zdroj: autor*

**3)** Monitoring Windows služeb – služba byla ručně zastavena.

| Požadavek | Simulace události | Vznik a detekce události | Poznámka |
|---|---|---|---|
| *Windows služba* | 15:50 | 15:50:52 | Odeslán první alert |
| | | 15:55:54 | Odeslán Druhý alert |
| | | 15:55:54 | Spuštěn vzdálený příkaz |
| | | 16:56:54 | Detekce spuštěné služby |

*Tabulka 21 - Tabulka výsledků testu monitorování windows služby, zdroj: autor*

**4)** Eskalace – Čas pro eskalaci byl snížen na 1 minutu a byla vyvolána major událost.

| Požadavek | Simulace události | Vznik a detekce události | Poznámka |
|---|---|---|---|
| *Eskalace* | 18:03 | 18:04:38 | Odeslán první alert |
| | | 18:05:39 | Odeslání alertu do ServiceDesku |
| | | 18:06:26 | Založen ticket v ServiceDesku |

*Tabulka 22 - Tabulka výsledků testu eskalace, zdroj: autor*

## 6.7 Operativní požadavky

Během provozu Zabbixu vznikly další doplňující požadavky na monitorování a bylo tak do konfigurováno:

1) Monitorování vybraných služeb např.: http a https pomocí tzv.: *simple checks.* Stav služby je vyhodnocen na základě toho, zdali přijímá TCP nebo UDP spojení na definovaném portu.

*2)* Rozsáhlá vizualizace dat v podobě tzv.: screenů pro podporu řešení problémů s terminálovými Citrix servery. Jednotlivé screeny byly vytvořeny tak, aby vždy zahrnovaly grafy jednotlivých CI pro všechny Citrix servery např.: byl vytvořen screen *ABC Citrix CPU Util,* který obsahuje 5 grafů s využitím CPU pro každý z Citrix serverů.

*3)* Rozsáhlá vizualizace dat v podobě screenů pro podporu řešení problémů s využitím šířky pásma linky, přes kterou probíhá replikace hlavní lokality do DR lokality.

## 6.8 Vyhodnocení konfigurace

I přesto, že byly stanoveny pouze základní požadavky na monitoring, bylo potřeba provést rozsáhlou analýzu nejen prostředí ale i jednotlivých serverů. Na základě této analýzy proběhlo rozdělení IT operátorů a monitorovaných hostů do logických skupin. Byly tak zohledněny různorodé požadavky na zdroje i na monitorování jednotlivých hostů a bylo tím dosaženo i odfiltrování nerelevantních informací pro jednotlivé operátory. Jako základ pro monitorování byly využity oficiální šablony pro OS Windows, Linux a WMware. Tyto šablony však nebyly dostatečně vhodné pro potřeby organizace. Jako jejich největší nevýhodu vidím nejednotnou jmennou konvenci, která je značně nepřehledná, monitorování CI na základě nevhodných metrik a nevhodně definované prahové hodnoty. Tyto nedostatky bylo potřeba postupně napravit doplněním vhodných metrik, upravením prahových hodnot a přejmenováním triggerů viz kapitola 6.3. Aby nedocházelo k alertování false positive události, bylo zamezeno flappingu pomocí hystereze. Alerting byl dále doplněn o možnosti eskalace významných událostí do interního ServiceDesku. Eskalace bylo také využito pro automatické spuštění vybraných Windows služeb.

Na základě provedeného otestování konfigurace hodnotím celkovou implementaci za zdařilou a odpovídající stanoveným požadavkům. Tato základní konfigurace e byla dále rozvedena, tak aby poskytovala podporu při řešení současných i budoucích problémů. Zabbix je stabilně využíván při operativním řešení incidentů i dlouhodobých problémů. Díky možnosti rozsáhlé vizualizace je také využíván pro zhodnocení a plánování kapacit. Celkově je monitorováno 61 hostů viz *tabulka č. 23* a byly tak pokryty stanovené požadavky na monitoring vybraných produkčních serverů i zbylé doplňující požadavky.

| **Host group** | **Počet Hostů** |
|---|---|
| *ABC Application Servers* | 8 |
| *ABC Citrix Servers* | 5 |
| *ABC Exchange Servers* | 2 |
| *ABC SQL Servers* | 2 |
| *ABC Windows Servers* | 18 |
| *ABC Linux Servers* | 1 |
| *ABC Oracle Servers* | 9 |
| *ABC VMware Infrastructure* | 7 |
| *ABC Network Nodes* | 4 |
| *Virtual Machines* | 5 |
| *Celkem* | 61 |

*Tabulka 23 - Počet monitorovaných hostů, zdroj: autor*

# 7 Doporučený postup implementace libovolného monitorovacího nástroje

V této kapitole jsou shrnuty jednotlivé kroky a myšlenkové pochody výše zmíněného procesu implementace a konfigurace Zabbixu tak, aby je bylo možné aplikovat na implementaci libovolného monitorovacího nástroje. Přihlédnuto bylo také k nasbíraným zkušenostem autora během tohoto procesu a poučení se z vlastních chyb.

#### 1) Analýza prostředí

Zásadní krok pro určení požadavků na monitoring, zároveň také pro výběr vhodného monitorovacího nástroje. Cílem analýzy prostředí je zejména získat přehled o tom, jaké množství a jaké prvky je potřeba monitorovat. Přihlédnout je také vhodné k potenciální expanzi.

#### 2) Analýza současného stavu

Při analýze současného stavu je potřeba se ujistit, zdali je již pokryta některá z forem monitoringu či nikoliv – monitoring provozu, bezpečnosti nebo některé úrovně (tzn.: businessu, služby nebo zdrojů *viz kapitola 2.1*) Dále je také potřeba přihlédnout k tomu, zdali budou pro účely implementace, konfigurace a provozu monitorovacího nástroje alokovány vlastní personální zdroje.

#### 3) Stanovení požadavků

Na základě proběhlé analýzy prostředí a současného stavu by mělo dojít ke stanovení požadavků jednak na monitoring, tak i na samotný monitorovací nástroj. V tomto kroku je tedy potřeba specifikovat jaké CI a jakou úroveň je potřeba monitorovat. Dále je nutné stanovit základní požadavky na monitorovací nástroj (např.: podporované platformy a cena)

#### 4) Upřesnění požadavků

Definované požadavky v kroku č. 3 je dále potřeba upřesnit a konkretizovat na potřeby daného prostředí. Otázkou kroku č. 3 je: „*Co monitorovat?*“. Otázkou Kroku č. 4 je naopak: „*Jak monitorovat?*“ tzn: jakými technologiemi je možné dané CI monitorovat. Tímto se také konkretizují a upřesní požadavky na monitorovací nástroj. (např.: podpora SNMP, WMI atp.)

#### 5) Výběr vhodného monitorovacího nástroje

Vhodný monitorovací nástroj by měl splňovat veškeré požadavky stanovené v kroku č. 3 a 4. Je také vhodné přihlédnout k cenové politice nástroje a dostupnosti technické podpory.

#### 6) Nasazení nástroje a komunitních řešení v pasivním režimu

Pro většinu nástrojů je k dispozici spoustu oficiálních nebo komunitních řešení ve formě šablon, pluginů apod. Tyto řešení nejsou však vždy vhodné pro konkrétní potřeby. Proto, jako v případě *NBA*, vidím vhodné nejdříve pasivní nasazení nástroje spolu s těmito řešeními. Účelem tohoto kroku je stanovit běžný profil chování.

#### 7) Logické seskupení

Prvky, které je možné monitorovat stejným způsobem, by měly být seskupeny tak, aby bylo možné provést konfiguraci hromadným způsobem. Stejně tak by měli mít operátoři IT přístup pouze k relevantním informacím a být alterování pouze o, pro ně, relevantních událostech.

#### 8) Úprava konfigurace nástroje ke konkrétním potřebám

Na základě profilu chování z kroku č. 6 je posléze potřeba zhodnotit vhodnost monitorovaných CI, stanovených prahových hodnot, triggerů, alertingu a následná úprava všech aspektů, které nejsou vyhovující. Takto dojde k eliminaci záplavy alertů v důsledku nevhodné konfigurace. Jako vhodné také vidím odstranění CI, které není potřeba monitorovat.

#### 9) Testování na omezeném vzorku

Krok č. 7 a 8. je potřeba otestovat, a to zejména z pohledu triggeringu a alertingu tedy, zdali dochází k filtraci nevýznamných událostí, alertování významných událostí určeným operátorům, automatickému řešení událostí či k eskalaci.

#### 10) Vyhodnocení

Pokud je na základě kroku č. 9 vyhodnocena konfigurace jako nevyhovující, je potřeba se vrátit do kroku č. 7, respektive č. 8. Pokud je konfigurace vyhovující, je možné přejít do kroku č. 11.

#### 11) Aktivní provoz monitoringu v plném rozsahu

I když je konfigurace uznána jako vyhovující, je potřeba monitoring rozvíjet spolu s rozvojem IT infrastruktury a novými požadavky. Za tímto účelem může být někdy potřeba se vrátit znovu do některého z předcházejících kroků.

Celý výše uvedený postup je shrnut v *tabulce č. 24*.

| **č.** | **Název kroku** | **Na co brát ohled?** |
|---|---|---|
| 1. | Analýza prostředí | Jak rozsáhlé je prostředí? nN jakých technologiích je založeno? Předpokládá se výrazná expanze? |
| 2. | Analýza současného stavu | Je v současné době pokryta některá vrstva monitoringu? (tzn.: bussinessu, služby, infrastruktury). Je přítomný monitoring provozu/bezpečnosti?<br>jsou dostatečné personální kapacity? |
| 3. | Stanovení požadavků | Jaké prvky a CI je potřeba monitorovat? Co by měl monitorovací nástroj splňovat? |
| 4. | Upřesnění požadavků | Jakým způsobem je možné dosáhnout bodu č.3? |
| 5. | Výběr vhodného nástroje | Je daný nástroj dostatečný pro stanovené požadavky? Je jeho cena a cenová politika přijatelná? Je dostupná technická podpora? |
| 6. | Nasazení nástroje a komunitních řešení v pasivním režimu | Jaký je běžný profil chování? |
| 7. | Logické seskupení | Je možné podobné prvky monitorovat stejným způsobem? Je možné alertovat operátory hromadně? |
| 8. | Úprav konfigurace nástroje ke konkrétním potřebám | Jsou pokryty všechny potřebné CI? vyhovují stanovené prahové hodnoty, triggery, alerting? |
| 9. | Testování na omezeném vzorku | Dochází správně k filtraci, korelaci, eskalaci, řešení událostí? |
| 10. | Vyhodnocení | Je na základě korku č. 9 konfigurace vyhovující?<br>- NE -> návrat do kroku č.7, respektive č.8<br>- ANO -> posun do kroku č. 11 |
| 11. | Aktivní provoz monitoringu v plném rozsahu | Je konfigurace stále vyhovující?<br>- NE -> návrat libovolného předešlého kroku<br>- ANO -> pasivní údržba, rozvoj s prostředím |

*Tabulka 24 – Doporučený postup implementace libovolného monitorovacího nástroje, zdroj: autor*

# 8 Závěr

Bakalářská práce se zabývá monitoringem IT infrastruktury. Nejprve je v druhé kapitole tato problematiky představena spolu s principy. Dále jsou také popsány aspekty, které je potřeba brát v úvahu při implementaci libovolného monitoringu. Jmenovitě se jedná o metriky, prahové hodnoty, triggering a alerting. Uveden je jak monitoring z pohledu provozu služeb dle ITIL, tak i bezpečnostní monitoring. Jmenovitě je uvedena behaviorální analýza síťového provozu, Log managementu a SIEM.

V třetí kapitole je představen aktivní, pasivní, proaktivní a reaktivní způsob monitorování. Dále je popsán monitoringu pomocí agenta i bez agenta. V druhém případě jsou představeny protokoly SNMP, IPMI, ICMP, WMI a NetFlow. V následující, čtvrté, kapitole jsou představeny monitorovací nástroje Nagios Core, Nagios XI, Zabbix a SolarWinds SAM. Každý nástroj je přestaven z pohledu technologické stránky, podporovaných platforem, logiky monitorování a cenové politiky. Srovnání nástrojů je poté vykonáno na základě autorem definované metodiky a autorem vypracované tabulky srovnání funkcionalit jednotlivých nástrojů. Na závěr srovnání jsou stanovena doporučení pro výběr jednotlivých nástrojů.

Samotná implementace nástroje Zabbix do produkčního prostředí reálné společnosti je uvedena v páté kapitole. Nejprve je uvedena autorem provedená analýza prostředí a současného stavu. Dále jsou zde uvedeny požadavky na monitoring, které byly posléze autorem upřesněny a konkretizovány včetně způsobu monitorování. Při konfiguraci Zabbixu byl brán ohled na informace uvedené v druhé kapitole. Konfigurace byla proto provedena a je popsána s důrazem na pochopení monitorovaných CI, jejich metrik a správného určení prahových hodnot. Při konfiguraci triggerů byl brán ohled na význam a závažnost jednotlivých událostí s přihlédnutím k eliminaci false positive událostí. Dále byla popsána konfigurace alertingu včetně eskalace a monitorování Windows služeb. Závěrem kapitoly je provedeno otestování a vyhodnocení popsané konfigurace. Na základě tohoto procesu bylo autorem sestaveno doporučení pro implementaci libovolného monitorovacího nástroje.

Hlavním přínosem práce je implementovaný monitorovací nástroj v prostředí reálné společnosti nakonfigurovaný tak, aby byly pokryty stanovené požadavky, a který bude dále sloužit jako podpora operativního řešení incidentů i dlouhodobých problémů.

Hlavním přínosem práce pro čtenáře je pak komplexní pohled na problematiku monitoringu a srovnání vybraných monitorovacích nástrojů. Nejen díky tomuto srovnání, ale i díky autorem sestavenému doporučení pro implementaci libovolného monitorovacího nástroje, může být práce dále využita společnostmi při výběru a implementaci monitorovacího nástroje do vlastního prostředí.

# Seznam použitých zkratek

| | |
|---|---|
| API | Application Programming Interface |
| BMC | Baseboard Management Controller |
| bps | bits per second |
| CI | Configuration item |
| CPU | Central Processing Unit |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name Systém |
| FTP | File Transfer Protocol |
| GDPR | General Data Protection Regulation |
| HDD | Hard Disk Drive |
| HTTP | Hypertext Transfer Protocol |
| HW | Hardware |
| ICMP | Isnternet Control Message Protocol |
| IS/IT | Information System / Information Technology |
| IOPS | Input/output operations per second |
| IP | Internet Protocol |
| IPFIX | IP Flow Information Export |
| IPMI | Intelligent Platform Management Interface |
| IT | Information technology |
| ITIL | Information Technology Infrastructure Library |
| MIB | Management Information Base |
| ms | millisecond |
| NAS | Network Attached Storage |
| NBA | Network Behavior Analysis |
| OID | Object identifier |
| OS | Operating Systém |
| RAM | Random Access Memory |
| RFC | Request For Comments |
| SAM | Server & Application Monitor |
| SAN | Storage Area Network |
| SIEM | Security Information and Event Management |
| SLA | Service-level agreement |
| SMTP | Simple Mail Transfer Protocol |
| SNMP | Simple Network Management Protocol |
| SQL | Structured Query Language |
| SW | Software |
| WMI | Windows Managemen Instrumentation |

# Seznam literatury


[1] *ITIL service operation*. 2nd ed. London: TSO, 2011. Best Management Practice. ISBN 9780113313075.

[2] OGBONNA, Dominic. *A-Z of Capacity Management: Practical Guide for Implementing Enterprise It Monitoring* ... Florida: Booklocker.com, 2017. Kindle, ISBN 9781634927598.

[3] FATEMA, Kaniz, Vincent EMEAKAROHA, Philip HEALY, John MORRISON a Theo LYNN. *A Survey of Cloud Monitoring Tools: Taxonomy, Capabilities and Objectives* [online]. 2014 [cit. 2018-03-01]. Dostupné z: https://www.researchgate.net/publication/263962818_JPDC-13-315R1

[4] KETTNER, Mathias. *Distributed Monitoring* [online]. [cit. 2018-03-24]. Dostupné z: https://mathias-kettner.com/cms_distributed_monitoring.html

[5] Metriky (Metrics). *Management Mania* [online]. [cit. 2018-03-20]. Dostupné z: https://managementmania.com/cs/metriky

[6] ANDREA DALLE VACCHE a STEFANO KEWAN LEE. *Mastering zabbix*. S.l.: Packt Publishing Limited, 2014. ISBN 9781783283507.

[7] *ITIL – A guide to event management*. UCISA [online]. [cit. 2018-03-18]. Dostupné z: https://www.ucisa.ac.uk/-/media/files/members/activities/itil/service_operation/eventm_management/itil_a%20guide%20to%20event%20management%20pdf.ashx?la=en

[8] *Zabbix*.com, Zabbix Documentation 3.0. [online]. [cit. 2018-03-20]. Dostupné z: https://www.zabbix.com/documentation/3.0/

[9] Event Handlers. Asset *Nagios* [online]. [cit. 2018-03-20]. Dostupné z: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/eventhandlers.html

[10] Monitor, alert and restart a Windows service from SAM. Support *Solarwinds* [online]. [cit. 2018-03-20]. Dostupné z: https://support.solarwinds.com/Success_Center/Server_Application_Monitor_(SAM)/Monitor_alert_and_restart_a_Windows_Service_from_SAM

[11] ROGERS, Joe. Monitoring What's Important – Event Severity and Notifications. [online]. 2014 [cit. 2018-03-18]. Dostupné z: https://www.netiq.com/communities/cool-solutions/netiq-views/monitoring-whats-important-event-severity-and-notifications/

[12] LIGUS, Slawek. *Effective monitoring and alerting*. Farnham: O'Reilly, 2013. ISBN 978-1-449-33352-2.

[13] KEMPTER, Stefan a Andrea KEMPTER. ITIL Service Operation. In: *ITIL Wiki* [online]. [cit. 2018-03-21]. Dostupné z: https://wiki.en.it-processmaps.com/index.php/ITIL_Service_Operation

[14] *ITIL – A guide to problem management*. UCISA [online]. [cit. 2018-03-21]. Dostupné z: https://www.ucisa.ac.uk/~/media/Files/members/activities/ITIL/service_operation/problem_management/ITIL_a%20guide%20to%20problem%20management%20pdf

[15] What is Security Monitoring?. *Hewlett Packard Enterprise* [online]. [cit. 2018-04-30]. Dostupné z: https://www.hpe.com/us/en/what-is/security-monitoring.html

[16] *ISO/IEC 27000:2016: Information technology — Security techniques — Information security management systems — Overview and vocabulary*. Fourth edition. Switzerland, 2016.

[17] ŠKORNIČKOVÁ, Eva. *Jaké zásadní změny GDPR přinese* [online]. [cit. 2018-03-22]. Dostupné z: https://www.gdpr.cz/gdpr/zmeny/

[18] YOUSSEF, Ahmed a Ahmed EMAM. *Network Intrusion Detection Using Data Mining and Network Behaviour Analysis* [online]. 2011 [cit. 2018-03-22]. Dostupné z: https://www.researchgate.net/publication/266167734_Network_Intrusion_Detection_Using_Data_Mining_and_Network_Behaviour_Analysis

[19] ANTON A. CHUVAKIN, KEVIN J. SCHMIDT, CHRISTOPHER PHILLIPS, Anton A. Chuvakin, Kevin J. Schmidt, Christopher Phillips a technical editor. PATRICIA MOULDER. *Logging and log management the authoritative guide to understanding the concepts surrounding logging and log management*. Online-Ausg. Waltham, Mass: Syngress, 2013. ISBN 9781597496360

[20] SWIFT, David. *Successful SIEM and Log Management Strategies for Audit and Compliance* [online]. 2010 [cit. 2018-03-22]. Dostupné z: https://www.sans.org/reading-room/whitepapers/auditing/successful-siem-log-management-strategies-audit-compliance-33528

[21] DEVANA, Ramkumar. When do you go for agent-based and when do you go for agent-less monitoring? [online]. 2013 [cit. 2018-03-24]. Dostupné z: https://community.saas.hpe.com/t5/IT-Operations-Management-ITOM/When-do-you-go-for-agent-based-and-when-do-you-go-for-agent-less/ba-p/303551#.WTGRDmjyjDd

[22] SHERWOOD, Jeremy. Agent vs. Agentless Monitoring [online]. 2011 [cit. 2018-03-24]. Dostupné z: https://www.sciencelogic.com/blog/agent-vs-agentless-monitoring

[23] BOUŠKA, Petr. Začínáme s monitoringem sítě [online]. 2009 [cit. 2018-03-24]. Dostupné z: http://www.samuraj-cz.com/clanek/zaciname-s-monitoringem-site/

[24] CASE, Jeffrey, Mark FEDOR a Martin SCHOFFSTALL. *A Simple Network Management Protocol (SNMP)* [online]. 1990 [cit. 2018-03-24]. Dostupné z: https://www.ietf.org/rfc/rfc1157.txt

[25] MAURO, Douglas R. a Kevin J. SCHMIDT. *Essential SNMP*. 2nd ed. Sebastopol, CA: O'Reilly, 2005. ISBN 9780596008406.

[26] FISCHER, Werner. *IPMI Basics* [online]. [cit. 2018-03-24]. Dostupné z: https://www.thomas-krenn.com/en/wiki/IPMI_Basics#Intelligent_Platform_Management_Bus_.28IPMB.29

[27] POSTEL, J. *INTERNET CONTROL MESSAGE PROTOCOL* [online]. 1981 [cit. 2018-03-23]. Dostupné z: https://tools.ietf.org/html/rfc792

*[28] Microsoft: Introduction to WMI* [online]. [cit. 2018-04-24]. Dostupné z: https://technet.microsoft.com/en-us/library/cc181125.aspx

[29] BLAWAT, Brenton J.W. *Mastering Windows PowerShell Scripting*. Packt Publishing, 2015. ISBN 9781782173557.

[30] DASS, Anthony. *Developing WMI Hi-Perf Provider using C++* [online]. 2005 [cit. 2018-04-24]. Dostupné z: https://www.codeproject.com/Articles/11969/Developing-WMI-Hi-Perf-Provider-using-C

[31] SANTOS, Omar. *Network Security with NetFlow and IPFIX: Big Data Analytics for Information Security*. Cisco Press, 2015. ISBN 9780134033549.

[32] *Flowmon.com: Flowmon sonda* [online]. [cit. 2018-03-25]. Dostupné z: https://www.flowmon.com/getattachment/fd92a24b-17ca-4dbb-bd0f-aa65981a9275/Flowmon-Probe-Product-Brief.aspx

[33] *Nagios.com: Nagios XI* [online]. [cit. 2018-03-27]. Dostupné z: https://www.nagios.com/products/nagios-xi/?__hstc=118811158.f56a2d0cd59f2e31ecfae61012dfff19.1520864518966.1523987982728.1524605279623.20&__hssc=118811158.1.1524605279623&__hsfp=354899117#systemreqs

*[34]* BELTOWSKI, Piotr a Wojciech KOCJAN. *Learning Nagios* [online]. 3rd Edition. [cit. 2018-04-24]. ISBN 9781785885952.

[35]*Nagios.com: NRPE* [online]. [cit. 2018-03-27]. Dostupné z: https://assets.nagios.com/downloads/nagioscore/docs/nrpe/NRPE.pdf

*[36] Nagios.com: Nagios XI // Pricing* [online]. [cit. 2018-03-27]. Dostupné z: https://assets.nagios.com/handouts/nagiosxi/Nagios-XI-Pricing-Information.pdf

*[37] Zabbix.com: Zabbix Life Cycle & Release Policy*[online]. [cit. 2018-03-27]. Dostupné z: https://www.zabbix.com/life_cycle_and_release_policy

*[38] Zabbix.com: Features* [online]. [cit. 2018-03-27]. Dostupné z: https://www.zabbix.com/features

*[39] Zabbix.com: Technical Support* [online]. [cit. 2018-03-27]. Dostupné z: https://www.zabbix.com/support

*[40] Solarwinds.com: SolarWinds SAM system requirements* [online]. [cit. 2018-03-28]. Dostupné z: https://support.solarwinds.com/Success_Center/Server_Application_Monitor_(SAM)/SAM_Administrator_Guide/020_SAM_Requirements/010_SolarWinds_SAM_hardware_requirements

*[41] Solarwinds.com: SOLARWINDS SERVER & APPLICATION MONITOR* [online]. [cit. 2018-03-28]. Dostupné z: https://www.solarwinds.com/-/media/solarwinds/swdc/resources/datasheets/sam/sam_evaluators_guide.ashx

*[42] Solarwinds.com: Online Quote* [online]. [cit. 2018-03-28]. Dostupné z: https://www.solarwinds.com/onlinequotes/#/onlineQuote

[43] *Nagios.com: Documentation* [online]. [cit. 2018-03-25]. Dostupné z: https://www.nagios.org/documentation/

[44] *Solarwinds.com: SAM DOCUMENTATION* [online]. [cit. 2018-03-25]. Dostupné z: https://support.solarwinds.com/Success_Center/Server_Application_Monitor_(SAM)/SAM_Documentation

[45] Understanding Linux CPU Load - when should you be worried?. *Scoutapp* [online]. [cit. 2018-04-15]. Dostupné z: http://blog.scoutapp.com/articles/2009/07/31/understanding-load-averages

[46] clumsy 0.2. *jagt.github.io* [online]. [cit. 2018-04-29]. Dostupné z: http://jagt.github.io/clumsy/index.html

[47] HeavyLoad. *Jam-software.com* [online]. [cit. 2018-04-29]. Dostupné z: https://www.jam-software.com/heavyload/

[48] PAPEŽ, Tomáš. *Monitoring počítačové sítě*. Brno, 2014. Bakalářská práce. Mendelova univerzita v Brně. [online]. [cit. 2018-02-12]. Dostupné z: http://is.mendelu.cz/lide/clovek.pl?zalozka=13;id=398;studium=50179;zp=40888;download_prace=1;lang=sk

[49] PLESKOT, Vít. *Dohledové systémy pro počítačové sítě*. Pardubice, 2012. Bakalářská práce. UNIVERZITA PARDUBICE. [online]. [cit. 2018-02-12]. Dostupné z: http://dspace.upce.cz/bitstream/handle/10195/48433/PleskotV_DohledoveSystemy_JH_2012.pdf?sequence=2

[50] SZABO, Gábor. *Monitoring lokálnej siete prostredníctvom Nagiosu*. Brno, 2012. Bakalářská práce. MASARYKOVA UNIVERZITA. [online]. [cit. 2018-02-12]. Dostupné z: https://is.muni.cz/th/me35p/bcpraca.pdf

[51] HABRMAN, Zdeněk. *Monitorování aktivních prvků počítačové sítě*. Zlín, 2013. Diplomová práce. Univerzita Tomáše Bati ve Zlíně. [online]. [cit. 2018-02-12]. Dostupné z: http://digilib.k.utb.cz/bitstream/handle/10563/25097/habrman_2013_dp.pdf?sequence=1&isAllowed=y

[52] KOCOURKOVÁ, Lucie. *Udržitelnost a rozvoj monitorovacích systémů v komerční společnosti*. Brno, 2013. Diplomová práce. MASARYKOVA UNIVERZITA. . [online]. [cit. 2018-02-12]. Dostupné z: https://is.muni.cz/th/262486/fi_m/diplomova_prace.pdf

[53] Google trends. *Google.com* [online]. [cit. 2018-03-11]. Dostupné z:

https://trends.google.com/trends/

# Seznam obrázků a tabulek

## Seznam obrázků

*Obrázek 1 - Zjednodušená architektura monitorovacích nástrojů, Zdroj: [3]* ..... 11
*Obrázek 2 – Logika triggeringu monitorovacího nástroje Zabbix, zdroj: [3]* ..... 14
*Obrázek 3 – Struktura MIB, Zdroj: [25]* ..... 23
*Obrázek 4 – Dlouhodobé využití RAM databázového serveru, zdroj: autor* ..... 44
*Obrázek 5 - Graf využití RAM serveru, zdroj: autor* ..... 50
*Obrázek 6 - Graf využití SWAP serveru, zdroj: autor* ..... 50
*Obrázek 7 - Graf využití CPU serveru, zdroj: autor* ..... 51
*Obrázek 8 - historický trend vývoje využití paměti RAM, zdroj: autor* ..... 51
*Obrázek 9 - Konfigurace alertu, zdroj: autor* ..... 53
*Obrázek 10 - Eskalace major událostí do ServiceDesku, zdroj: autor* ..... 54
*Obrázek 11 - Automatické spuštění zastavené Windows služby, zdroj: autor* ..... 56
*Obrázek 12 - Graf provozu na síťovém rozhraní, zdroj: autor* ..... 58

## Seznam tabulek

*Tabulka 1 – Metriky monitorování IT infrastruktury, zdroj: [2], upraveno* ..... 13
*Tabulka 2 – Ceny licencí Nagios XI, zdroj: [36], vlastní zpracování* ..... 27
*Tabulka 3 – Ceny technické podpory Nagiosu, zdroj: [36], vlastní zpracování* ..... 27
*Tabulka 4: Stupně technické podpory, zdroj: [39], vlastní zpracování* ..... 28
*Tabulka 5 – Licence SolarWinds SAM, zdroj: [42], vlastní zpracování* ..... 29
*Tabulka 6 - Kategorie srovnání nástrojů, zdroj: autor* ..... 30
*Tabulka 7 - Srovnání Nagios Core, Nagios XI, Zabbix, SAM, zdroj: autor, [3][35][36][37][38][39][40][41][42][43][44][44]* ..... 31
*Tabulka 8 – Výsledek srovnání Nagios Core, Nagios XI, Zabbix a SAM, zdroj: autor* ..... 32
*Tabulka 9 – Vážený výsledek srovnání Nagios Core, Nagios XI, Zabbix a SAM, zdroj: autor* ... 32
*Tabulka 10 - Host groups, zdroj: autor* ..... 38
*Tabulka 11 - Uživatelské skupiny, zdroj: autor* ..... 39
*Tabulka 12 - Seznam definovaných šablon, zdroj: autor* ..... 40
*Tabulka 13 - Seznam monitorovaných CI u standardních Windows serverů, zdroj: autor* ..... 42
*Tabulka 14 - Seznam definovaných triggerů pro standardní MS Windows, zdroj: autor* ..... 46
*Tabulka 15 - Seznam definovaných triggerů pro MS Windows, zdroj: autor* ..... 47
*Tabulka 16 - Seznam definovaných triggerů pro DB Linux Oracle, zdroj: autor* ..... 48
*Tabulka 17 - Seznam definovaných triggerů pro servery monitorované přes VMware API, zdroj: autor* ..... 49
*Tabulka 18 - Seznam definovaných alertů, zdroj: autor* ..... 52
*Tabulka 19 – Tabulka výsledů testů dostupnosti, zdroj: autor* ..... 57
*Tabulka 20 – Tabulka výsledků testů využití zdrojů, zdroj: autor* ..... 58
*Tabulka 21 - Tabulka výsledků testu monitorování windows služby, zdroj: autor* ..... 58
*Tabulka 22 - Tabulka výsledků testu eskalace, zdroj: autor* ..... 58
*Tabulka 23 - Počet monitorovaných hostů, zdroj: autor* ..... 60
*Tabulka 24 – Doporučený postup implementace libovolného mon. nástroje, zdroj: autor* ..... 63

# Příloha 1 – Rešerše zdrojů a stávajících prací

Pro účely této práce byly jako hlavní zdroje informací využity následující knižní publikace.

*A-Z OF CAPACITY MANAGEMENT: practical guide for implementing enterprise it monitoring &... capacity planning.* [2] Kniha je zaměřena na monitoring v souvislosti s capacity managementem. Cenný zdroj informací představovaly kapitoly věnované monitoringu, ve kterých kniha zmiňuje, jaká data sbírat a metody a technologie pro sběr dat. Dále jsou uvedeny způsoby, jak stanovit vhodné metriky a způsoby nastavení prahových hodnot a alertingu.

Pro účely doplnění vztahu monitoringu na provoz služeb bylo zejména čerpáno z *ITIL service operation.* [1] Kniha pojednává o provozu IT služeb s ohledem na efektivitu, účinnost a udržení bezchybného stavu. Také zmiňuje procesy, jak těchto cílů dosáhnout.

Ke konkretizace obecných poznatků o monitoringu na specifický nástroj sloužila kniha *Mastering Zabbix monitor your large IT environment efficiently with Zabbix*, [6] která nejprve představuje Zabbix a jeho architekturu, dále se zaměřuje na sbíraná data, způsob jejich sběru a jejich zpracování/analýzu. Další část knihy je zaměřena na konfiguraci nástroje a jeho správu. Zmiňuje také vhodný výběr a nastavení triggerů, alertingu, správu šablon apod. Poslední část je věnována rozšíření systému Zabbix a jeho integrací s jinými systémy.

Kniha *Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management*, [19] posloužila jako zdroj informací v případě bezpečnostního monitoringu a obecného seznámení s problematikou log managmentu. Kniha nejdříve pojednává obecně o logu, zmiňuje zdroje logů a způsoby uchovávání logů. Další část je zaměřena na práci s logy – zisk relevantních informací z logů – analýza, filtrování, korelace, vizualizace atd. Další část zmiňuje vyhodnocování logů, dostupné nástroje a příklady reakcí/procesů.

Za účelem stanovení směru a cílů této práce bylo také nahlédnuto do níže uvedených stávajících kvalifikačních prací.

Bakalářská práce *Monitoring počítačové sítě* [48] pojednává o monitoringu počítačové sítě z pohledu dostupnosti prvků. V teoretické části je popsán způsob monitorování

s agentem, bez agenta a protokol SNMP a NetFlow. V praktické části autor postupně implementoval vybrané nástroje, které postupně otestoval. Jako vhodný nástroj byl zvolen Zabbix.

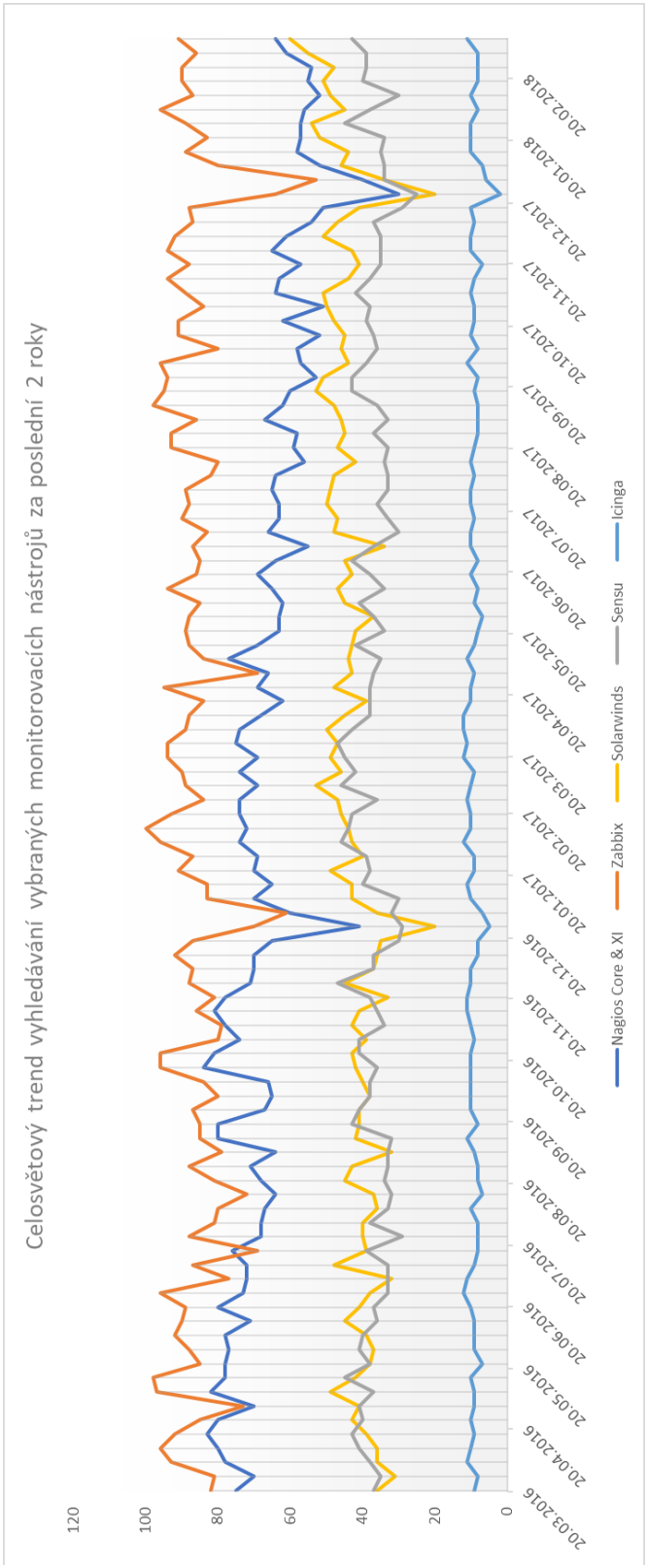
Bakalářská práce *Dohledové systémy pro počítačové sítě* [49] nejprve seznamuje čtenáře s problematikou počítačové sítě a monitorovacích nástrojů, které kategorizuje dle jejich funkcionalit na Alerting, Application monitoring, Cloud monitoring apod. Z protokolů popisuje ping, SNMP a netstat. Praktická část je defacto návod instalace a konfigurace nástroje OpsView

Autor bakalářské práce *Monitoring lokálnej siete prostredníctvom Nagiosu* [50] se zaměřil specificky na nástroj Nagios a protokol SNMP, na kterém vytvořil a implementoval plugin pro monitorování tiskárny.

V diplomové práci *Monitorování aktivních prvků počítačové sítě* [51] se autor zabývá principy dohledových systémů, které kategorizuje na lokální, síťové, komplexní, profesionální, hardwarové a speciální. Z protokolů se detailně zaměřil na SNMP a ICMP. Výsledkem je implementace nástroje Cacti v prostředí univerzitní sítě.

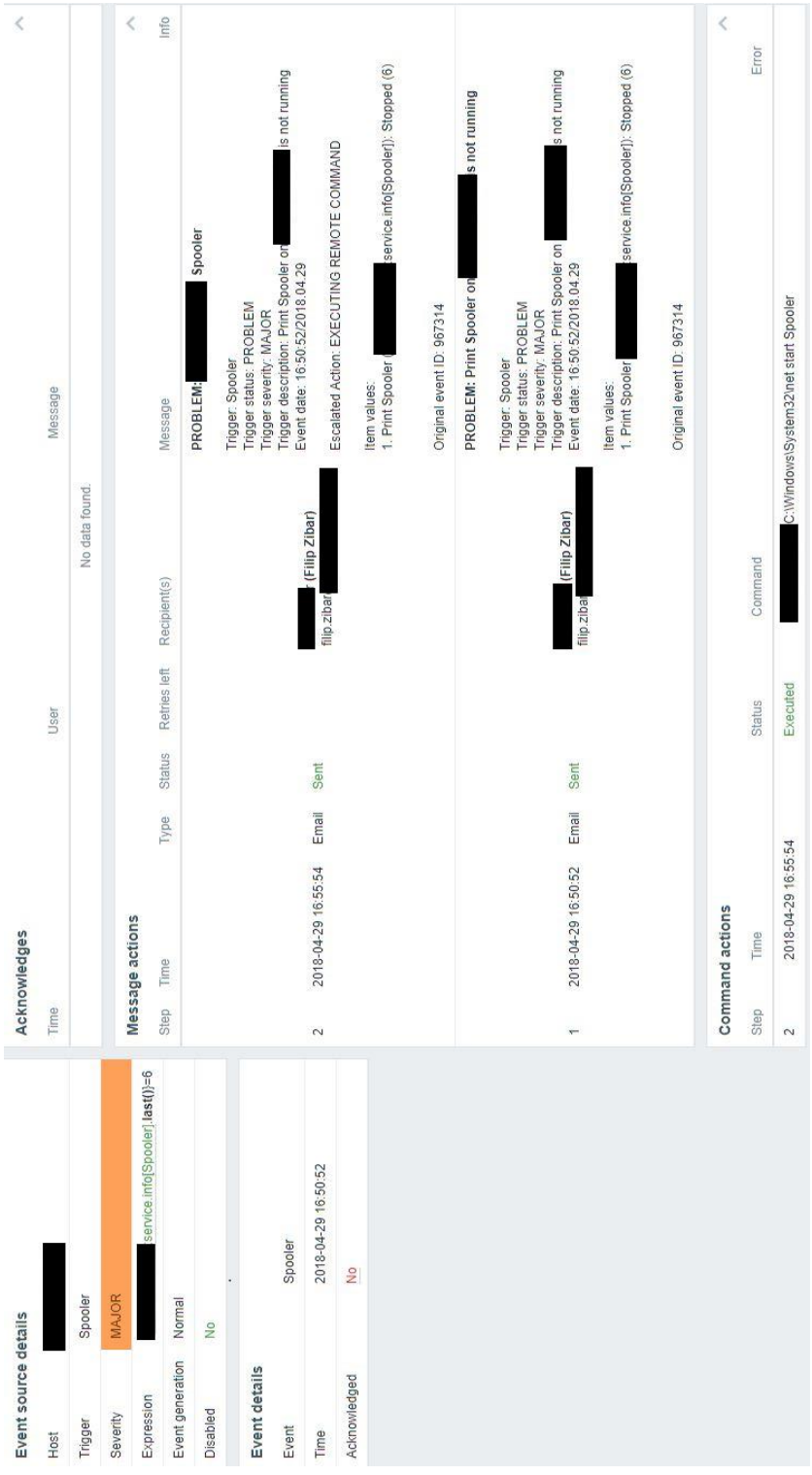
Autorka diplomové práce *Udržitelnost a rozvoj monitorovacích systémů v komerční společnosti* [52] se zaměřila na srovnání dostupných monitorovacích nástrojů jako je Zabbix, Observium a OpenNMS s interně vyvinutým monitorovacím nástrojem společnosti – MonitorFM. Závěrem práce bylo uznání stávajícího nástroje dostačujícím a navržení několika změn.

# Příloha 2 – Trend vyhledávání Nagios Core & XI, Zabbix a SAM




*zroj: [53]*

# Příloha 3 – Test eskalačního pravidla pro spuštění zastavené tiskové služby




zdroj: autor