VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ARCHITEKTURA A SPRÁVA ZABEZPEČENÝCH SÍTÍ
ARCHITECTURE AND MANAGEMENT OF SECURE NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Jan Štangler**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Vlastimil Člupek, Ph.D.**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Jan Štangler ***ID:*** 184487
Ročník: 2 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Architektura a správa zabezpečených sítí

POKYNY PRO VYPRACOVÁNÍ:

V diplomové práci vytvořte metodiku pro návrh zabezpečené malé až středně velké sítě s centrální správou. Navrhněte zabezpečenou síť pro konkrétní vybrané zaměření. Síť bude obsahovat servery a uživatelské počítače, bude centrálně spravována a bude využívat software s otevřeným zdrojovým kódem. Proveďte nasazení Vámi navržené zabezpečené sítě s centrální správou. Otestujte konektivitu jednotlivých prvků v síti a ověřte jejich funkčnost. Proveďte automatizaci zabezpečené centrální správy sítě. Simulujte modelové situace provozu v síti a útoky na síť, jenž budou centrálním bodem detekovány a vyhodnoceny z hlediska závažnosti dopadu na bezpečnost sítě. Výsledky detekcí útoků na síť přehledně zpracujte a doporučte další postup.

DOPORUČENÁ LITERATURA:

[1] ALI, Q.; ALABADY, S. Design and implementation of a secured remotely administrated network. In: proceedings international arab conference on information technology, ACIT. 2007.

[2] CHOI, Hyunsang; LEE, Heejo; KIM, Hyogon. Fast detection and visualization of network attacks on parallel coordinates. computers & security, 2009, 28.5: 276-288.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 1.6.2020

Vedoucí práce: Ing. Vlastimil Člupek, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato práce je zaměřena na bezpečnost malých až středně velkých sítí s centrální správou, zejména pak na vytvoření metodiky pro návrh zabezpečené sítě. Popsán je návrh zabezpečené sítě pro začínající IT firmu s využitím open-source softwaru. Je provedeno nasazení navržené zabezpečené sítě s centrální správou a otestována konektivita prvků. V práci jsou simulovány modelové situace provozu a útoky na síť pomocí technik penetračního testování. Z hlediska závažnosti dopadu na bezpečnost sítě jsou zachycené útoky vyhodnoceny a okamžitě oznámeny zodpovědným osobám. Na závěr jsou výsledky zachycených útoků zpracovány a je doporučen další postup.

KLÍČOVÁ SLOVA

Ansible, architektura sítě, automatizace, bezpečnost, detekce útoku na síť, metodika návrhu zabezpečené sítě, open-source, síťový útok, správa systémů.

ABSTRACT

This work is focused on the security of small to medium-sized networks with central administration, especially on the creation of a methodology for secure network design.The design of a secure network for a start-up IT company, using open-source software, is described. Deployment of the designed secure network, with central management, is performed and the connectivity of network elements are tested. The model simulates network traffic situations and network attacks using penetration testing techniques. In terms of the severity of the impact on network security, intercepted attacks are evaluated and immediately reported to responsible persons. Finally, the results of the intercepted attacks are processed and further actions are recommended.

KEYWORDS

Ansible, automation, network architecture, network attack detection, open-source, methodology for secure network design, network attack, security, system administration.

ŠTANGLER, Jan. *Architektura a správa zabezpečených sítí*. Brno, Rok, 122 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Vlastimil Člupek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Architektura a správa zabezpečených sítí“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno ....

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

V důsledku snižování nákladů společnosti dávají přednost službám založených na open-source softwaru. Vystavují se však riziku ztráty podpory nebo vůbec žádné podpory. Totéž přichází i po bezpečnostní stránce, jelikož vydávání bezpečnostních záplat je závislé od komunity, která software spravuje a vyvíjí. Nedostatečné zajištění bezpečnosti ve firemní síti může vést k úniku citlivých dat a informací. Za důležité považuji zmínit *obecné nařízení o ochraně údajů – General Data Protection Regulation* (GDPR), jenž stanovuje práva a povinnosti občanů, firem a on-line služeb za účelem zlepšit ochranu osobních údajů. V případě, že firma je správcem nebo zpracovatelem osobních údajů, je nezbytné tyto údaje patřičně chránit. Nedostatečnou úrovní bezpečnosti se firmy vystavují riziku, jehož důsledky mohou být likvidační.

Jeden z problémů open-source softwaru je dokumentace, která nemusí být dostatečně zpracována nebo pravidelně udržována, což pak ztěžuje celý proces nasazení a konfiguraci. Samotná konfigurace může být zdlouhavá, a proto se často zanedbává. Přitom optimalizace nastavení služby může vést k odstranění nedostatků, zvýšení výkonu nebo zajištění lepší bezpečnosti. Pro snížení rizika je také vhodné pečlivě zvážit a porovnat alternativy služeb, které chceme v rámci sítě využívat. Zásadní výhodou služeb postavených na open-source je možnost integrace do jiných řešení díky dostupnosti *rozhraní pro programování aplikací – Application Programming Interface* (API) rozhraní. V porovnání s proprietárním softwarem se lze vyhnout dalším nákladům, které vznikají za jakoukoliv modifikaci nebo vyžádání modulu do dané služby. Navíc podmínky open-source licencí jsou dost otevřené, což umožňuje i jejich další modifikaci a distribuci. V takovém případě se nelze obejít bez zkušeného IT pracovníka nebo vývojáře.

Nároky na správu a monitoring zabezpečené sítě jsou závislé od její velikosti. Pro zajištění kvalitního přehledu o síti je potřeba monitorovat vytížení jednotlivých prvků v síti, výpadky, nestandardní chování a samozřejmě jakékoliv události zachycené systémy prevence průniku. V ideálním případě za co nejefektivnějšího využití hardwarových prostředků a včasného varování ze strany centrálních prvků.

Zpočátku se práce věnuje architektuře počítačových sítí, základům správy a monitoringu sítí. Ve druhé kapitole jsou popsány hrozby a útoky v oblasti počítačových sítí. Uvedeny jsou také základní způsoby zabezpečení sítí.

Třetí kapitola se věnuje tvorbě metodiky pro návrh zabezpečené sítě. Nejprve řeší architekturu a hierarchii sítě, doporučuje konfiguraci pro jednotlivé síťové prvky a služby, popisuje základní principy bezpečnostních politik a co by měly obsahovat.

Čtvrtá kapitola popisuje návrh malé počítačové sítě a její architekturu na základě vytvořené metodiky. Zahrnuje úvod do vybraných služeb a nástrojů. Důraz je kladen na použití open-source řešení.

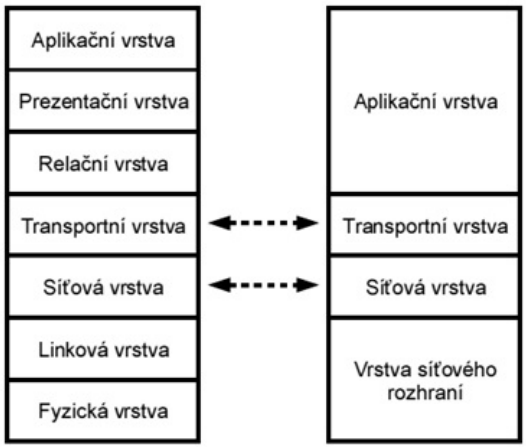
V páté kapitole dochází k implementaci laboratorní sítě, jejíž architektura se odvíjí od vytvořeného návrhu. Na základě vybraného softwaru dochází k implementaci automatizačních technik pro centrální správu, monitoring a zavedení detekčních mechanismů pro zajištění bezpečnosti sítě.

Poslední kapitola se věnuje simulaci provozu v síti a útokům na ni pomocí penetračního testování. Testují se detekční mechanismy a na závěr je doporučen další postup.

1 Architektura počítačových sítí

Za největší počítačovou síť považujeme Internet, jež propojuje stovky milionů zařízení po celém světě. Zpočátku se jednalo o tradiční stolní počítače a servery. Postupně se sítě rozrostly o další zařízení jako jsou mobilní telefony, tablety, notebooky a další. Vlivem pokroku v oblasti chytrých sítí a technologií dochází k exponenciálnímu nárůstu zařízení připojených do sítě Internet. Dle odhadů by měl *Internet věcí – Internet of Things* (IoT) po roce 2020 přesáhnout 30 miliard zařízení, což také vysvětluje obavy z rizika kybernetických útoků [1, 2].

Počítačovou sítí se obecně myslí propojení dvou a více počítačů, které spolu mohou komunikovat. Nezávisle na tom, jaký používají operační systém nebo hardware. Všechna zařízení však musí rozumět stejným protokolům. Aby to bylo možné, tak došlo k zavedení síťových referenčních modelů ISO/OSI a TCP/IP, neboli konceptu síťové architektury. Vyjadřují představu o tom, jak by měli sítě fungovat a řeší problematiku vzájemného propojování. Na obrázku 1.1 můžeme vidět, že jednotlivé modely se liší především počtem vrstev. Model ISO/OSI popisuje až 7 vrstev z nichž na každé zajišťuje spojitost a spolehlivost přenosu. Ve srovnání s modelem TCP/IP je spolehlivost zajišťována pouze na vyžádání a to jen u vyšších vrstev. Vycházelo se z předpokladu, že zajištění spolehlivosti je problémem až koncových účastníků. Může tedy docházet ke ztrátám přenášených paketů bez snahy o nápravu, tudíž jde o nezaručenou službu tzv. best effort. Každá z vrstev zastupuje určitou funkci, která je poskytována vyšší sousední vrstvě. Model TCP/IP je praktickou realizací modelu ISO/OSI [2].

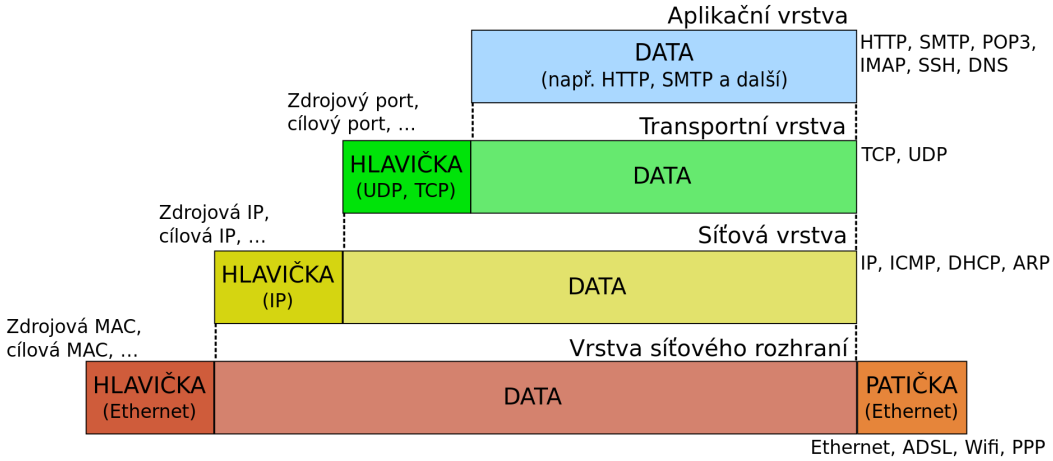


Obr. 1.1: Srovnání modelu ISO/OSI a TCP/IP [3].

V rámci obou modelů probíhá zapouzdření dat. Každá z vrstev je schopna zapouzdřit data z vrstvy vyšší a naopak. Takto se pak data tzv. *protokolová datová jednotka – Protocol Data Unit* (PDU) pohybují mezi vrstvami. Účelem je umožnit

přepravu služeb mezi vrstvami a tím i přepravu na stejné vrstvě jiného uzlu. Proces kdy nižší vrstva rozděluje PDU vyšší vrstvy do několika dalších PDU je nazýván segmentace nebo fragmentace. Naopak proces kdy je spojováno více PDU do jednoho se nazývá agregace nebo blokování.

Na obrázku 1.2 je ukázka principu zapouzdření dat v referenčním modelu TCP/IP. Na každé vrstvě mají bloky dat specifické označení. Na transportní vrstvě se komunikuje protokolem *spojově orientovaný protokol – Transmission Control Protocol* (TCP) nebo *nezávisle orientovaný protokol – User Datagram Protocol* (UDP). Pokud je blok předán modulu pro TCP, vytváří se segmenty. V případě modulu UDP se vytváří datagramy. Blok pak směřuje na vyšší vrstvu. Síťová vrstva přidává IP hlavičky, čímž vytváří IP paket. Nakonec blok putuje na linkovou vrstvu, kde se přidáním hlavičky a patičky vytváří rámec, jenž je pomocí fyzické vrstvy odeslán do sítě.



Obr. 1.2: Zapouzdření dat v TCP/IP [6].

V zásadě počítačové sítě rozlišujeme na místní sítě LAN a globální počítačovou síť Internet, označovanou jako WAN. Dělení je však rozmanité např. dle topologie, role jednotlivých stanic, způsobu uchovávání dat nebo jejich přenosu.

1.1 Síťové prvky

Počítačová síť se skládá z řady zařízení, které umožňují přenos mezi koncovými body. Taková zařízení řadíme mezi síťové prvky. Pozornost bude věnována vybraným zařízením, jež se primárně vyskytují v sítích standardů IEEE 802.3 a IEEE 802.11. Konkrétně standard IEEE 802.3 popisuje specifikace fyzické a linkové vrstvy Ethernetu v lokálních sítích, jejichž uzly jsou fyzicky propojené. Přenosové rychlosti se pohybují mezi 10 Mbit/s až 400 Gbit/s. Naopak IEEE 802.11 je standard pro Wi-Fi,

tedy lokální bezdrátové sítě. V obou případech existuje řada modulací, které upřesňují jejich účel a přenosové vlastnosti. Přenosové rychlosti začínaly na 2 Mbit/s a v případě nového standardu 802.11ax mohou dosahovat i více jak 10 Gbit/s.

Síťové prvky se rozdělují na aktivní a pasivní. Za pasivní síťové prvky považujeme přenosová média jako je koaxiální kabel, kroucená dvoulinka, optické vlákno nebo vzduch. Jejich použití závisí na našich možnostech a požadavcích pro danou síť. Aktivní prvky kromě přenosu pracují se signály v síti – modifikují a vyhodnocují je. Zpravidla jde o konkrétně umístěná zařízení v uzlech sítě. Některá zařízení mohou zastávat funkci více aktivních prvků současně. Níže jsou popsány vybrané síťové prvky [2].

Směrovač

Směrovač (router) je síťové zařízení, které propojuje alespoň dvě různé sítě. Primárním úkolem je přeposílat (směrovat) pakety k jejich cíli. Pracuje na síťové vrstvě, 3. vrstvě ISO/OSI, 2. vrstvě TCP/IP. Jde o klíčový prvek sítí. Obvykle disponuje více jak jedním rozhraním, dokáže však přeposílat pakety i mezi *virtuální lokální síť – Virtual Local Area Network* (VLAN) v rámci jednoho rozhraní. Rozděluje kolizní domény, filtruje a blokuje všesměrové vysílání, zjišťuje optimální trasu pro směrování paketů k cíli. Router jako zařízení může zastávat více funkcí – síťový most[1] (bridge), brána[2] (gateway), firewall a v malých sítích také přístupový bod (access point).

Přepínač

Přepínač (switch) je víceportový prvek, který spojuje je jednotlivé části sítě. Pracuje na linkové vrstvě, L2 ISO/OSI. Je taktéž považován za centrální prvek sítě. Rozlišujeme L2 a L3 přepínače dle toho, zda jsou schopny analyzovat hlavičku síťového protokolu. Existují také vícevrstvé přepínače, ale ty nejsou tak používané. Na standardním L2 přepínači dochází k analýze rámců linkové vrstvy a segmentaci sítě na kolizní domény, takže každý port využívá celou šířku pásma síťového média. Přepínače pomáhají snížit nadbytečné toky v síti. Jde o efektivní zařízení, které vytváří a uchovává tabulku MAC adres připojených zařízení dle portů. Zpočátku je tabulka prázdná a přepínač se chová jako rozbočovač[3]. Po naplnění tabulky již rámce posílá odpovídajícím portem. VLAN vzniká logickým rozdělením prostředků jediné fyzické sítě Ethernet. Přepínač lze nakonfigurovat tak, aby přepínal rámce pouze mezi určitými porty. Aby bylo možné sítě VLAN definovat i na sítích s mnoha přepínači,

[1] propojuje segmenty počítačové sítě na linkové vrstvě

[2] uzel propojující sítě s odlišnými protokoly

[3] příchozí provoz z jednoho portu se kopíruje na ostatní porty

tak byl rámec Ethernet rozšířen o pole „tag“, jenž obsahuje identifikátor sítě VLAN. Spoj, kterým předáváme „tagované“ VLANy mezi přepínači, se nazývá trunk [2, 8].

Přístupový bod

Přístupový bod (Access Point – AP) je základní prvek bezdrátových sítí zajišťující připojení k datové síti. Svou činností je prvek velmi podobný přepínači. Komunikace mezi AP a koncovým uživatelem má charakter poloviční duplexu[4]. Existuje technologie *frekvenční skokové rozpětí spektra – Frequency Hopping Spread Spectrum* (FHSS), která umožňuje definovat oblast pokrytí a tím vymezit prostor, ve kterém je možné se připojit. Přístupové body lze provozovat v režimu root nebo bridge. Nejběžnější je režim root, kdy se koncové zařízení připojuje k přístupovému bodu AP. Přístupových bodů může být více a ty spolu mohou komunikovat prostřednictvím lokální sítě. V režimu bridge zařízení pracuje jako síťový most, který propojuje oddělené části pevné lokální sítě. U některých zařízení můžeme narazit na podporu opakovacího módu – repeater. Jde o režim tzv. bezdrátového mostu[5]. V takovém režimu dochází k příjmu zkresleného signálu zatíženého šumem z jiného AP, který je následně obnoven na původní signál a vyslán dál. Tato varianta se příliš nedoporučuje, jelikož může dojít ke snížení celkové propustnosti kanálu. V oblasti bezdrátových sítích se ještě můžeme setkat s pojmem roaming. Roaming umožňuje automatické předání připojených klientů mezi jednotlivými AP v závislosti na síle signálu. Aby roaming fungoval správně, je nutné aby měla koncová stanice implementována standardy 802.11r, 802.11k nebo 802.11v. V případě, že AP funkci roaming nepodporují, problém se řeší nastavením stejných SSID na různé frekvenci [9].

Firewall

Účelem firewallu je řídit a zabezpečit síťový provoz mezi sítěmi na základě definovaných pravidel. Pracuje obvykle mezi transportní a aplikační vrstvou, L4-L7 ISO/OSI. Umisťují se na hranice sítí. Díky firewallu lze koncentrovat služby pro zabezpečení do jediného zařízení a tím odlehčit síťové prostředky a zdroje v lokální síti. Některé firewallu mohou současně zastávat funkci *hloubková inspekce paketů – Deep Packet Inspection* (DPI), *Systém detekce průniku/Systém prevence průniku – Intrusion Detection System/Intrusion Prevention System* (IDS/IPS). V praxi rozlišujeme několik typů – paketové filtry, stavové firewally, proxy firewally, *webový aplikační firewall – Web Application Firewall* (WAF), *firewally nové generace – New Generation Firewall* (NGFW).

Paketové filtry pracují v základní podobě na síťové vrstvě, L3 ISO/OSI. Řídí se pakety na základě zdrojové a cílové IP adresy nebo podle TCP a UDP portu.

[4] v jeden okamžik se může vysílat pouze jedním směrem

[5] obdoba síťového mostu

Filtrování však neřeší problém zabezpečení samotného připojení, jelikož odchozí provoz obsahuje IP adresy počítačů umístěných za filtrem.

Problém paketového filtru řeší stavový firewall pomocí stavové inspekce. Úroveň bezpečnosti je vyšší díky přiřazování paketů k příslušnému spojení a hlídání jeho stavu. Dochází k překladu síťových adres (NAT), díky němuž zůstává identita interních počítačů skryta. Funkce NAT umožňuje převod IP adres počítačů z vnitřní sítě na venkovní adresu hraničního směrovače a naopak. V praxi to znamená, že provoz z interní sítě je schován za jednu IP adresu.

Mezi pokročilejší firewally se řadí aplikační firewall. Řeší bezpečnost na aplikační vrstvě L7 ISO/OSI a je schopen zasahovat do provozu na úrovni protokolů – HTTP, SMTP a další. Potencionální nevýhodou je snížení výkonu sítě, jelikož dochází k aktivní analýze a manipulaci s přenášeným provozem, který aplikačním firewallem prochází. Pro zabezpečení webových serverů se používá WAF, jehož cílem je na základě předdefinovaných pravidel detekovat a zabránit případným útokům na webový server. V poslední době se začínají objevovat firewally nové generace NGFW nebo taky *jednotné řízení hrozeb – Unified Threat Management* (UTM) firewally, které mohou provádět inspekci šifrovaného provozu nebo spolupráci s DHCP, DNS a AD [10].

1.2 Síťové protokoly

Pro komunikaci v počítačové síti je používána rodina protokolů TCP/IP. Název pochází ze dvou hlavních protokolů – TCP (Transmission Control Protocol), který je zodpovědný za doručení dat, dále pak IP (Internet Protocol) a ten rozděluje data do paketů. Jde o hlavní protokolovou sadu sítě Internet. Architektura je členěna do čtyř vrstev – linková, síťová, transportní a aplikační. Každá vrstva využívá služeb vrstvy nižší a poskytuje své služby vrstvě vyšší. Pro účel této práce byly vybrány jen některé protokoly [2, 11].

Vrstva síťového rozhraní
Nejnižší vrstva umožňující přístup k fyzickému přenosovému médiu.
Ethernet
Protokol je určen k přenosu dat prostřednictvím spojů typu Ethernet. K určení stanice v rámci sítě Ethernet používáme MAC adresu, která je adresou rozhraní stanice. MAC adresa má délku 6 bajtů a může mít podobný formát 51:64:00:3D:19:08. První tři bajty identifikují výrobce rozhraní a poslední tři bajty reprezentují výrobní číslo.

PPP

Point-to-Point Protocol je založený na dvoubodovém spoji. Využití nachází především u DSL, ADSL technologií, které se používají pro zavedení Internetu do domácností prostřednictvím pevné linky. Jde o linkový protokol zajišťující multiplexovaný přenos různých protokolů. Umožňuje autentizaci komunikujících stran, dynamické nastavení síťových zařízení, kompresi a šifrování dat.

Síťová vrstva

Zajišťuje mechanismus pro síťovou adresaci a směrování. Jde o jednu z nejvíce komplexních vrstev. Na této úrovni se používá IP adresa zařízení, která nabízí verzi IPv4 nebo IPv6.

IP

Používají se dvě verze tohoto protokolu – IPv4 a IPv6. Základní jednotkou je blok bajtů, který je nazýván paket. V současné době se IPv4 postupně nahrazuje verzí IPv6. V záhlaví obou těchto protokolů se nachází informace, které identifikují zdroj a příjemce – zdrojová a cílová IP adresa. Protokol verze IPv4 obsahuje IP adresy s délkou 32 bitů, což umožňuje adresovat pouze 2^{32} stanic. Ve srovnání IPv6 může obsahovat IP adresy s délkou 128 bitů, takže počet adresovatelných stanic je až 2^{128} [12].

Celková délka paketu může být maximálně 65535 bajtů, v praxi je však maximální doporučená generovaná délka paketu 576 bajtů. Velikost se obecně označuje jako *maximální přenosová jednotka – Maximum Transmission Unit* (MTU) a standardní hodnotou je 1500 bajtů. Pokud je paket u rámce Ethernet delší jak 1500 bajtů, tak dochází k jejich fragmentaci. Aby se tomuto předešlo, technologie MTU discovery prověřuje spoj mezi koncovými body a nastaví se maximální možná velikost MTU. U každého paketu definujeme *doba života – Time To Live* (TTL), který ve skutečnosti určuje maximální počet směrovačů, kterými paket může projít. Hlavička paketu obsahuje *druh služby – Type of Service* (ToS) určující typ služby a její prioritu. Pakety obsahují i další příznaky a hlavičky, které například identifikují paket, takže není nutné, aby byly do cíle doručeny ve správném pořadí.

Protokol IPv6 má ve srovnání s IPv4 řadu předností a nezatěžuje tak například uzly sítě fragmentací paketů, protože fragmentaci paketu povoluje pouze u odesílatele. Zásadní výhodou verze IPv6 je zjednodušení záhlaví, možnost integrace bezpečnostních funkcí a podpora multimediálních přenosů.

ICMP

Protokol ICMP slouží pro diagnostiku a řízení na úrovní síťové vrstvy. Každá ICMP zpráva je zapouzdřená v jediném datagramu. Protokol dokáže analyzovat vybrané stavy a řídit některé procesy spojené s konkrétním zařízením. Mezi nejznámější patří – Echo Request, Echo Reply, Destination Unreachable, Time Exceeded. Je součástí

řady nástrojů sloužící k diagnostice v síti. V souvislosti s protokolem ICMP známe nástroj „ping“, který se používá k testování dosažitelnosti hostitele v síti. Dochází k posílání IMCP zpráv Echo Request a Echo Reply. Nevýhodou ICMP je možnost zneužití k útokům např. odepření služby (DoS).

ARP

ARP vznikl potřebou používat protokol IP v sítích Ethernet. Protokol slouží k překladu IP adresy stanice na její MAC adresu. Protokol je přenášen v rámcích. Základem je mezipaměť ARP, kterou má každé zařízení v paměti a uchovává v něm tabulku s IP adresami, které jsou přiřazeny ke konkrétním MAC adresám. Slabou stránkou protokolu ARP je fakt, že jde o bezestavový protokol, takže zúčastněné strany nemají přehled o vykonaných akcích. Jako útok na protokol ARP se používá technika zvaná APR spoofing, při níž dochází k podvržení odpovědi na ARP dotaz.

IPsec

IPsec je kryptografické rozšíření protokolu IP. Internet Protocol Security. Zahrnuje ověření autentičnosti a důvěrnosti přenášených paketů. Bezpečnostní vylepšení pro IPv6 přinesl hlavně IPsec, který je u dané verze povinný. Pro zvýšení bezpečnosti IPsec využívá záhlaví *ověřovací hlavička – Authentication Header* (AH) a *zapouzdření dat šifrováním – Encapsulating Security Payload* (ESP). AH záhlaví se stará o autentičnost paketů a ESP záhlaví řeší jejich důvěrnost. Použití ESP je povinné, záhlaví AH je volitelné. Celý proces stojí na databázi bezpečnostní politiky *databáze zásad zabezpečení – Security Policy Database* (SPD), což je soubor pravidel pro práci s datagramem. Protokol IPsec lze provozovat v transportním nebo tunelovacím režimu. Transportní režim zabezpečuje komunikaci mezi dvojicí počítačů tzv. šifrováním mezi koncovými body (end-to-end). Naopak tunelovací mód řeší zabezpečení paketů pouze mezi dvojicí počítačových sítí, ve kterých se počítače nachází. V praxi je možné využít i dynamické vytváření bezpečnostních asociací podle aplikačního protokolu IKEv2[6] [11].

Transportní vrstva

Vrstva zodpovědná za spojení mezi klienty a servery. Umožňuje multiplexování[7] služeb. Poskytuje transportní služby protokoly TCP (spolehlivé spojení) a UDP (nespolehlivé spojení). Port má na transportní vrstvě abstraktní podobu a jedná se o rozhraní aplikace. Existuje 65536 různých portů a dělí se do tří rozsahů – systémové (0 až 1023), uživatelské (1024 až 49151) a dynamické (49152 až 65535).

TCP

TCP je spojově orientovaný protokol zajišťující spolehlivé doručení, správnou posloupnost a také bezchybnost přenášených dat. Základní jednotkou je TCP paket.

[6] používá se k ustanovení šifrovacích klíčů

[7] dělení společného zdroje mezi více uživateli

Před samotným přenosem se vždy navazuje spojení. Hlavička segmentu především obsahuje zdrojový a cílový port, oznamovací číslo (sekvenční), potvrzovací číslo (AN) a příznak. Mezi používané příznaky patří:

- URG – pro naléhavé data
- ACK – signalizace platného AN
- PSH – signalizace pro okamžité předání dat aplikaci
- RST – pro odmítnutí nabízeného spojení
- SYN – pro nabídku spojení
- FIN – pro jednostranné ukončení spojení

Jde o nejčastěji využívaný protokol. Používají jej aplikační protokoly jako FTP, HTTP, SSH a další.

UDP

UDP je nespojově orientovaný protokol, který je označován za nespolehlivý a nezaručuje správnou posloupnost doručených dat. Základní jednotkou je UDP datagram. Spojení se předem nenavazuje, ale data se posílají přímo. Hlavička datagramu především obsahuje zdrojový a cílový port. Žádné příznaky nebo další identifikační prvky nejsou použity. Používá se u aplikací, které nemají velké nároky na spolehlivost, ale přitom preferují rychlé doručení. Např. aplikační protokol TFTP, DNS, SNMP a DHCP využívá UDP protokol. Protokol může být použit k útoku na dostupnost služeb (DoS).

TLS

TLS je nejznámější kryptografický protokol k zajištění bezpečné komunikace mezi aplikacemi. Protokol má předejít odposlouchávání nebo falšování komunikace. Nahradil původní protokol SSL. Komunikace je typu klient-server. Přenášená data jsou šifrována a strany komunikace autentizovány. Protokol TLS zahrnuje celkem tři fáze. V první fázi se obě strany dohodnou na podporovaných šifrách a protokolech. Ve druhé fázi pomocí kryptografie s veřejným klíčem (RSA) dochází k výměně šifrovacího klíče nebo jeho ustanovení pomocí protokolů (např. DH, ECDH). V poslední fázi je zahájeno šifrování některou ze symetrických šifer (např. AES, Camellia). Protokol TLS se využívá u řady aplikačních protokolů jako je HTTPS, SMTP, IMAP nebo POP3. Protokol má aktuálně čtyři verze – 1.0, 1.1, 1.2 a 1.3. V roce 2020 končí u velké části služeb podpora verze 1.0 a 1.1.

Existuje obecné povědomí o možnostech inspekce TLS komunikace, jejímž účelem je získat přístup k indikátorům pro IDS/IPS zařízení a firewally. V základním principu jde o útok muž uprostřed (Man-In-The-Middle) prostřednictvím proxy, kdy na proxy dochází k dešifrování a opětovnému šifrování provozu a přeposílání k cíli. Vše za předpokladu, že certifikát proxy je na straně klienta důvěryhodný [13, 14].

NAT

Technika NAT umožňuje přepis jedné nebo více IP adres na jednu globální IP adresu. Kromě přepisu IP adresy může docházet také k přepisu portů. NAT standardně pracuje v rámci firewallu nebo routeru. Zvyšuje bezpečnost vnitřní sítě a šetří adresní prostor. Rozlišujeme tři typy:

- DNAT – dynamický NAT, přístup do vnější sítě je umožněn přes několik globálních IP adres.
- SNAT – statický NAT, provádí statický překlad IP adres tj. překlad jedné IP adresy vždy na tutéž IP adresu, např. SRCNAT nebo DSTNAT.
- PAT (port forwarding) – přesměrování portů, dochází k překladu specifických portů z globální IP adresy na porty vnitřní IP adresy.

Aplikační vrstva

Vrstva pro aplikace, programy a procesy, které využívají přenos dat po síti. Protokoly aplikační vrstvy vždy využívají některou z transportních služeb TCP nebo UDP. Jednotlivá spojení určují porty. Většina protokolů aplikačních vrstvy je založena na modelu klient-server.

HTTP

Protokol HTTP je bezstavový a pracuje způsobem dotaz-odpověď. Je velmi populární a jednoduchý na implementaci. Umožňuje přenášet libovolné soubory. Původně podporoval pouze *hypertextový značkovací jazyk – Hypertext Markup Language* (HTML). Standardně používá protokol TCP na portu 80. Šifrovaná varianta HTTPS používá port 443 a pracuje s protokolem TLS. Přenášené soubory v rámci protokolu jsou kódovány technikou *víceúčelová rozšíření elektronické pošty – Multipurpose Internet Mail Extensions* (MIME), která umožňuje přenos jakýchkoliv souborů. Součástí hlavičky v odpovědi serveru na klientský požadavek je vždy nějaký stavový kód, který upřesňuje zpracování požadavku. Mezi nejznámější stavové kódy patří:

- 200 OK – standardní odpověď na úspěšný požadavek.
- 301 Moved Permanently, 302 Found – nejčastější metody pro přesměrování, obvykle pro přesměrování z HTTP na HTTPS.
- 400 Bad Request – server nerozumí požadavku.
- 403 Forbidden – požadavek nemá přístup k souboru nebo adresáři.
- 404 Not Found – stránka nenalezena, server nenašel danou adresu URL.
- 500 Internal Server Error – došlo k neočekávané chybě na straně serveru.
- 503 Service Unavailable – server nemůže nebo nechce zpracovat požadavek, např. při přetížení nebo údržbě.

Protokol nemá žádné zvláštní bezpečnostní mechanismy, a proto může útočník skrytě útočit na cílový počítač. Mezi útoky na webové servery patří především odepření služby (DoS) a útoky na webové aplikace (XSS[8], SQLi[9] a další) [15].

DHCP

Protokol DHCP slouží k automatickému přidělování síťových parametrů zařízením v sítí. Standardně se jedná o IP adresu, masku sítě, IP adresu brány a DNS server. Existují však desítky další parametrů, kterou mohou být pomocí DHCP doručeny. K přenosu používá protokol UDP. Server poslouchá na portu 67 a klient se připojuje z portu 68. DHCP klienta má v sobě většinou každé síťové zařízení. Komunikaci začíná DHCP klient, který vyšle paket s DHCP zprávou DISCOVER na globální všesměrovou IP adresu 255.255.255.255. Každý DHCP server odpovídá paketem s DHCP zprávou OFFER v němž nabízí IP adresu z přiděleného rozsahu. Klient vybere jednu z nabízených IP adres a pošle DHCP zprávu REQUEST s danou IP adresou. Jako odpověď dostává klient potvrzení se zprávou ACK, po kterém již může IP adresu začít používat. V síti se může nacházet i více DHCP serverů. Protokol DHCP bohužel nemá žádné bezpečnostní mechanismy, takže je vystaven řadě útoků. Útočník může vystupovat jako více klientských stanic a jednoduše celý rozsah přidělených IP adres vyčerpat. Další variantou je, že si útočník vytvoří vlastní DHCP server, který bude vysílat neplatné údaje a provoz bude směřovat na vlastní bránu, a pak může provést útok mužem uprostřed (Man-In-The-Middle).

DNS

Protokol DNS slouží k překladu doménových jmen na IP adresy. Opačný proces se nazývá reverzní DNS. Pro předávání DNS zpráv se používá port 53 a protokol TCP nebo UDP. Každé doménové jméno je jedinečné označení počítače. Ke spojení s konkrétním síťovým zařízením (počítačem) je nutná znalost jeho IP adresy. Myšlenkou DNS je, že doménové jméno se snadněji pamatuje než konkrétní IP adresa.

Systém doménových jmen připomíná uspořádaný strom. Každému uzlu kromě kořene je přiděleno nějaké unikátní jméno, jejichž zřetězením získáváme globální doménové jméno serveru. Překlad může vypadat následovně – feec.vutbr.cz na IP adresu 147.229.71.30. Každá doména provozuje alespoň jeden DNS server, který zná jména v doméně, případně zná IP adresu DNS serveru jeho bezprostředně podřízené domény. Jednotlivé úrovně se oddělují tečkou. V 1. úrovni je uzel se jménem „cz“. Ve 2. úrovni se nachází uzel „vutbr“ a ve 3. úrovni je uzel „feec“, který obsahuje a záznam pro IP adresu. Počet samotných úrovní a uzlů v každé z úrovní je závislý na konkrétní doméně.

V případě protokolu DNS se využívají ještě tzv. DNS záznamy, které slouží k základnímu nasměrování v rámci domény [7].

[8] metoda narušení WWW stránek využitím bezpečnostních chyb ve skriptech

[9] technika napadení databázové vrstvy programu injekcí kódu

Nejčastější DNS záznamy:

- A – obsahuje IPv4 adresu, pomáhá při určení hostingu webových stránek
- AAAA – obsahuje IPv6 pokud je podporována
- CNAME – obsahuje doménové jméno pro subdoménu – nižší úroveň
- MX – obsahuje název e-mailového serveru, kam chodí elektronická pošta

Samotný DNS protokol neprovádí žádnou autentizaci komunikujících stanic, takže přenášená data nejsou nijak chráněna. Existuje však zabezpečená varianta protokolu s názvem DNSSEC, jejímž základem je digitální podpis, takže po přijetí dat lze zjistit zda nedošlo k podvržení IP adresy [16].

SSH

Protokol SSH je aplikační protokol určený k bezpečnému vzdálenému přístupu pro uživatele a automatizované procesy. Jde o bezpečnou alternativu k zastaralých protokolům Telnet, Rlogin. Pro přenos používá protokol TCP na portu 22. Klient zahajuje připojení k SSH serveru a pomocí veřejných klíčů proběhne ověření serveru a klienta mezi sebou. Pak následuje ustanovení klíčů pro zajištění symetrického šifrování. Protokol kromě důvěrnosti zajišťuje také integritu pomocí hashovacích algoritmů a umožňuje předávání libovolných TCP portů modelu TCP/IP včetně možností komprese [18].

SMTP, POP3/IMAP

Protokoly slouží k přenosu elektronické pošty. Poštovní server se stará o příjem a odesílání pošty, která patří pod jeho doménu. Využívá se zde MX záznamů DNS systému. Používají se následující poštovní protokoly:

- SMTP – protokol aplikační vrstvy pro přenos elektronické pošty mezi stanicemi. Aktualizován byl standardem RFC 5321. Pro přenos používá protokol TCP na portu 25. Preferuje se samozřejmě zabezpečená varianta přes protokol TLS, která funguje na portu 465, označována jako SMTPS. Někde můžeme narazit ještě na zabezpečení pomocí STARTTLS na portu 587. Protokol má nižší efektivitu a nemá implementovány žádné bezpečnostní mechanismy. Obecně se doporučuje zprávy zabezpečovat pomocí S/MIME[10].
- POP3 – protokol aplikační vrstvy pro stahování e-mailových zpráv z poštovního serveru do poštovního klienta. Pro přenos používá protokol TCP na portu 110. Zajišťuje autentizaci klienta pomocí hesla. Ve výchozím režimu není spojení šifrované, a proto se používá spolu s kryptografickým protokolem TLS na portu 995, označována jako POP3S.

[10]standard pro veřejný klíč šifrování a podepisování

- IMAP – protokol aplikační vrstvy pro online práci s e-mailovými zprávami, které jsou umístěné na poštovním serveru. Narozdíl od protokolu POP3 vyžaduje trvalé připojení se serverem. V případě některých poštovních klientů lze načíst všechny zprávy do mezipaměti a s poštou pak lze pracovat i v režimu offline. Pro přenos se používá protokol TCP na portu 143. Zajišťuje autentizaci klienta pomocí hesla. Šifrovaná varianta pro přenosy zpráv funguje na portu 993 díky protokolu TLS, označována jako IMAPS.

SNMP

Protokol SNMP je aplikační protokol zajišťující komunikaci mezi zařízeními v síti. Umožňuje průběžné získávání cenných dat ze zařízení pro účely správy sítě. Je to nejpoužívanější protokol pro správu sítí. Komunikace probíhá mezi tzv. SNMP agentem a SNMP manažerem. Existují dvě varianty komunikace – manažer zasílá dotaz a agent odpovídá, nebo agent informuje server dle nějaké události. SNMP manažer používá protokol UDP a naslouchá na portu 161. SNMP agent naslouchá taktéž přes protokol UDP na portu 161, ale události se pro SNMP manažera posílají přes port 162.

Jsou dosud známy tři verze protokolu: SNMPv1 – špatné zabezpečení, autentizace prostřednictvím komunitního názvu. SNMPv2 – lepší výkon a bezpečnost než u předchozí verze, obsahuje autentizaci a navíc se posílá potvrzení o přijetí události. SNMPv3 – přidáno šifrování a zajištění integrity přenášených dat.

Ač je poslední verze nejlepším východiskem pro správu sítí, přesto nemá ještě širokou podporu na straně síťových zařízení. Nejvíce převládá protokol SNMPv2, pak protokol SNMPv1, jehož výchozí komunitní název je „public“ na každém zařízení, což představuje zásadní bezpečnostní slabinu, které se snažíme předcházet alespoň změnou výchozí hodnoty. U protokolu SNMPv3 se pro zabezpečený přenos používají porty 10161 a 10162 [11, 17].

RADIUS

Protokol RADIUS je aplikační protokol určený k přenosu autentizačních, autorizačních, konfiguračních a případně evidenčních informací mezi RADIUS klientem a serverem. Tento protokol nachází široké využití v bezdrátových sítích, jde ale o obecný bezpečnostní rámec pro LAN. RADIUS protokol je používán jako autentizační protokol v bezpečnostním standardu 802.1x, který je založen na protokolu EAP[11]. Obdobou protokolu RADIUS je proprietární protokol TACACS+ od firmy Cisco, který ale není u řady zařízení podporován. Protokol RADIUS a TACACS+ patří mezi tzv. AAA[12] protokoly.

[11] protokol zajišťující transportní mechanismus pro ověřování

[12] authentication, authorization and accounting

RADIUS server standardně používá protokol UDP a naslouchá na portu 1812. K dispozici je rozšíření pro zasílání zúčtovacích informací nazývané RADIUS „accounting“, které komunikuje přes port 1813. V rámci RADIUS serveru může přímo fungovat centrální správa uživatelských účtů.

Jako bezpečnostní nedostatek je, že RADIUS zprávy nepodléhají žádné ochraně na nižších vrstvách, takže může dojít k podvržení IP adresy. Za nástupce protokolu RADIUS je považován protokol Diameter, který by měl řešit jeho nejdůležitější nedostatky [19].

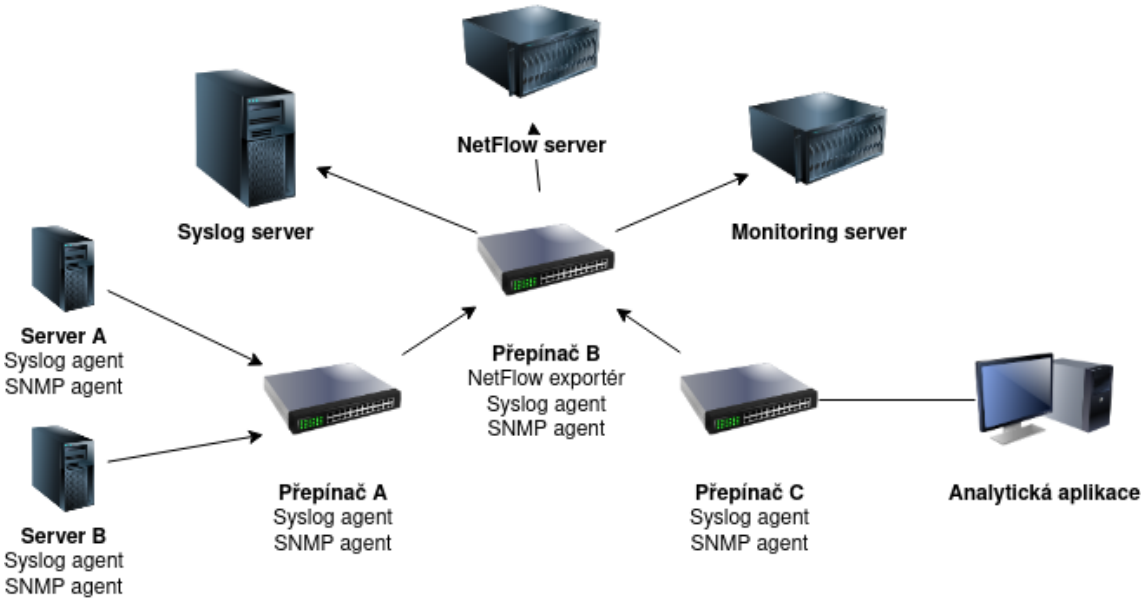
Kerberos

Protokol Kerberos je aplikační protokol určený k autentizaci a autorizaci uživatelů v rámci počítačové sítě. Předpokládá se, že prostředí sítě pro přenos není důvěryhodné a může dojít k užití služby nebo odposlechu cizí osobou. Protokol Kerberos je postaven na použití tzv. lístků, které slouží pro přístup k určité službě. Hlavním prvkem je *centrum distribuce klíčů – Key Distribution Center* (KDC), který se označuje za důvěryhodnou třetí stranu pro bezpečnou autentizaci a autorizaci přístupu. Tato služba současně uchovává data o uživatelích. KDC se skládá ze služeb *ověřovací server – Authentication Server* (AS) (autentizační server) a *řídicí server – Ticket Granting Server* (TGS). Autentizační server odpovídá za autentizaci a autorizaci uživatelů. TGS řeší přidělování lístků k použití dané služby, ke které se klient autentizuje. Uživatel provádí autentizace pouze jednou a po obdržení základního lístku *tiket opravňující uživatele ke komunikaci s řídícím serverem – Ticket Granting Ticket* (TGT) je s daným lístek schopen získat další lístky [19]. Slabou stránkou protokolu je, že neřeší útoky *útok odepření služby – Denial of Service* (DoS) ani útoky hrubou silou, které mohou vést k získání hesla. Aby se předešlo riziku získání uživatelských dat, doporučuje se volit dostatečně silná hesla.

1.3 Monitoring

Monitoring znamená sledovat určité informace v časovém období, na základě kterých se pak můžou odvíjet konkrétní akce. Účelem monitoringu je získat kompletní přehled o dění v síti. V mnoha případech lze pomocí monitoringu řešit většinu provozních problémů v síti včetně její optimalizace. S rostoucí infrastrukturou rostou také požadavky na sledování systémů. Pokud plánujeme zavedení monitoringu, tak je nezbytné určit oblasti, které pro nás budou z pohledu monitoringu zásadní. Vždy však bude záviset na konkrétním zaměření dané firmy nebo jednotlivce. Většinou se chceme zajímat o dostupnost služeb, dostupnost serverů, detekci nestandardního chování v síti nebo bezpečnostní incidenty. Pro někoho může být užitečná také kvalita poskytovaných služeb. Monitorovací systém můžeme postavit na vlastních

skriptech nebo některých dostupných řešení. Nejlepší a univerzální cestou jsou bezplatné řešení, které dokáží vyhovět našim požadavkům. Musíme však předpokládat, že to bude vyžadovat hlubší znalosti bez nichž může být zprovoznění daného řešení velmi náročné [21]. Příklad monitoring sítě a zařízení je možné vidět na obrázku 1.3.



Obr. 1.3: Příklad monitoringu sítě a zařízení.

Stěžejní částí jsou výstupy a upozornění ze strany monitoringu. Nejlepší variantou pro přehledný monitoring je jeho grafická reprezentace. Např. vytížení linky nás zajímá v určité časové periodě a vhodné zobrazení je grafem. Upozorňování může probíhat prostřednictvím e-mailu, SMS nebo služby pro okamžité zasílání zpráv.

Monitoring síťových zařízení a služeb

Ať už se jedná o aktivní síťové prvky nebo servery tak ve všech případech lze daná zařízení nějakým způsobem monitorovat – hlídat jejich dostupnost, vytížení systémových zdrojů nebo stav jednotlivých komponent. Standardně se dostupnost zařízení zjišťuje prostřednictvím protokolu ICMP se zprávami „echo request“ a „echo reply“. Tímto způsobem lze ověřit nejen dostupnost zařízení, ale také dobu odezvy (latency). Může nastat situace, kdy zařízení protokol ICMP s danými zprávami nepřijímá nebo jsou zakázány. Pokud je na zařízení přístupná jiná služba, tak se může dostupnost monitorovat podle ní. Pro zjištění dostupnosti služby jako je například FTP, HTTP, SSH, SMTP je potřeba využít protokol TCP a pokusit se o navázání spojení na daný port. Úspěšné navázání spojení však nemusí vždycky znamenat, že služba funguje správně. V případě například HTTP je vhodné provést aplikační test,

který kontroluje zda webový server vrací v hlavičce stavový kód 200.

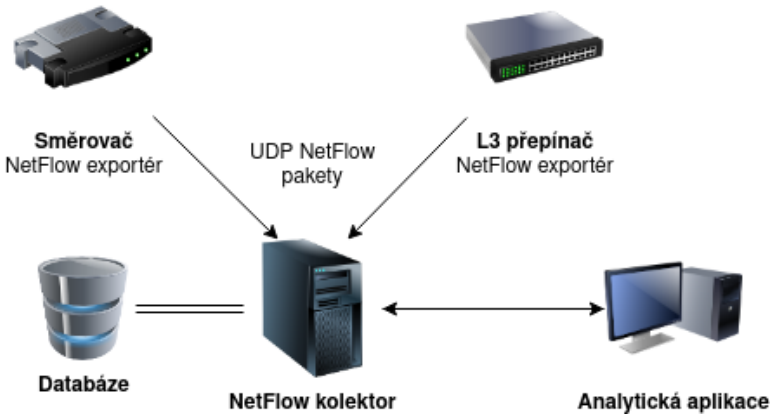
U aktivních síťových prvků se často setkáváme s protokolem SNMP, jenž nám umožňuje monitoring vybraných parametrů. Mezi takovými parametry se může objevit vytížení systémových zdrojů, aktuální provoz na jednotlivých rozhraních nebo počet chybně přijatých paketů. V případě serverů a počítačů nebývá přítomnost protokolu SNMP obvyklá a může být potřeba službu nastavit a spustit. Vzhledem k tomu, že v rámci těchto zařízení můžeme většinou zasahovat do systému, tak se objevuje možnost použití alternativních nástrojů pro monitoring a není nutné se vázat pouze na protokol SNMP.

Události ze síťových zařízení lze monitorovat také pomocí tzv. syslogu. Syslog je protokol pro příjem a odesílání zpráv z logů po síti. Umožňuje uchovávání logů z jednotlivých zařízení na jednom centrálním místě. Syslog standardně používá protokol UDP a poslouchá na portu 514. Podporuje většinu operačních systémů včetně systémů Linux, Unix a MacOS. U systémů Windows je potřeba instalace některého z nástrojů třetích stran, který je schopen tzv. „Event Log“ převést na syslog. Zásadní výhodou syslogu je, že umožňuje uchovávat dlouho historii, protože některé zařízení umožňují ukládat pouze omezené množství logů. Navíc lze k logům přistupovat nezávislé na dostupnosti zařízení odkud pocházejí [22].

Monitoring síťového provozu

Smyslem monitoringu síťového provozu je vytvářet agregované statistiky o všech datových přenosech v síti. Pro sběr provozních dat můžeme použít protokoly jako NetFlow nebo IPFIX. Prostřednictvím uvedených protokolů obvykle sledujeme zdrojové a cílové IP adresy, zdrojové a cílové porty, objemy dat, čas, protokoly a případně další technické parametry TCP/IP komunikace. Získávání provozních dat probíhá prostřednictvím aktivních síťových prvků jako jsou směrovače nebo přepínače. Z kopie datového provozu jsou vytvářeny přesné a detailní statistiky.

Protokol NetFlow je proprietární protokol vytvořený společností Cisco a jeho původním účelem bylo umožnit snadné účtování objemu přenášených dat u poskytovatelů Internetu. Nejvíce je rozšířena NetFlow verze 5 a 9. NetFlow v9 odstranila nedostatky verze 5, která nepodporovala IPv6 provoz, informace o VLAN nebo MAC adresy. V současné době je preferován protokol IPFIX, který je obecně uznávaný jako IETF standard. Zásadní výhodou IPFIX oproti NetFlow je jeho flexibilita. Obecným přínosem sběru provozních dat je možnost sledování zátěže sítě v čase kvůli případnému navyšování kapacity a současně lze provozní data použít pro analýzu bezpečnostních incidentů.



Obr. 1.4: Příklad architektury sítě pro sběr provozních dat

Zařízení, které provozní data sbírá, ukládá a zobrazuje označujeme jako NetFlow kolektor. Všechny zmíněné protokoly používají protokol UDP a nastavení portu pro kolektor je individuální. Existuje ale zaužívaný port pro NetFlow 2055, 2056, případně pro IPFIX je to port 4739. Výběr portu může být individuální [20].

Pro rozpoznávání provozních dat na aplikační vrstvě se používá DPI (hloubková analýza paketů). Kombinuje funkci IDS/IPS spolu s tradičním stavovým firewallem. Identifikace aplikací pomocí DPI může probíhat na základě používaných portů, specifických signatur nebo pomocí heuristické a behaviorální analýzy. Dochází k vyhodnocování obsahu paketů. DPI určuje, jak bude s pakety naloženo [24].

Monitorování sítě z bezpečnostního hlediska lze provádět pomocí NBA (behaviorální analýza sítě). Pomocí této technologie lze získat úplně nový pohled na provoz v síti. Myšlenkou NBA je detekce obecné anomálie v provozu sítě, která by mohla vést k odhalení infikovaných stanic nebo dosud neznámých hrozeb a útoků tzv. „Zero Day“ útoků [20].

Pro monitoring průniku do sítě používáme systémy IDS/IPS. Tyto systémy zajišťují detekční nebo prevenční mechanismy pro podezřelé aktivity v síti. Mezi detekční metody patří detekce signatur, odchylek nebo anomálií. Rozlišujeme pasivní a aktivní IDS. Pasivní varianta v případě odhalení průniku nezasahuje do síťového provozu. Aktivní variantu označujeme jako IPS, která reaguje na podezřelé aktivity přerušením spojení nebo provedením úprav na firewallu [28].

1.4 Centrální správa sítí

V současné době sítě obsahují desítky až stovky zařízení, z nichž každé může používat jiný operační systém a nabízet různé služby. Správa takových sítí je velmi komplexní a kromě samotných zařízení je potřeba mít přehled nad monitoringem, konfigurací a zabezpečením celé sítě. Předpokladem pro ideální správu sítě je její bezproblémový chod. V praxi se však vždycky nějaké problémy naskytnou, a proto se musíme snažit předcházet neočekávaným situacím a rizikům, které mohou nastat. Odhalení slabých míst hned v počátku může výrazně pomoci celému procesu v budoucnu. Největší problém správy sítí je, když se neřeší problémy, které sice ještě nenastaly, ale ví se o nich.

Samotné systémy používané pro správu nejsou součástí žádného ze síťových modelů. Při správě sítě si samozřejmě můžeme pomoci některými protokoly určenými pro správu jako je protokol SNMP, RMON nebo DMI, ale zdaleka nám nebudou schopny poskytnout plnohodnotné řešení [8].

V praxi nám mohou být oporou základní funkce síťového managementu, které můžeme najít v ISO/IEC 7498-4:1989 v části „Management framework“:

- **správa poruch** – nejdůležitější část pro identifikaci problémů. Vzniklé problémy jsou předávány zodpovědným osobám prostřednictvím komunikačního kanálu ve formě notifikací. Doporučuje se mít také efektivní způsob pro informování běžných uživatelů, ať už formou automaticky generované zprávy nebo formou reportu. Správa poruch by měla mít přístup ke všem zdrojům v síti. Důvodem jsou doplňující procesy, které mohou provádět diagnostiku případně další analýzu.
- **správa konfigurace** – účelem je dohled na stav sítě a její konfiguraci. Jakékoliv změny mohou mít vliv na její funkčnost, výkon a stabilitu. Zásahy do konfigurace by měly být minimální pokud to pro situaci není žádoucí. Může

dojít k nějaké poruše nebo konkrétnímu požadavku ze strany koncových uživatelů. Z praktického hlediska se může jednat o správu adresního prostoru, VLAN nebo management síťových služeb jako je DHCP, DNS.

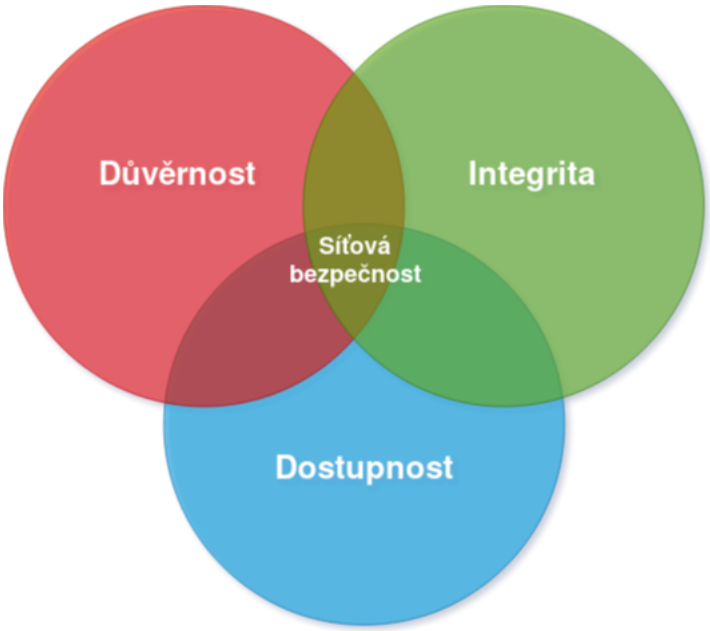
- **správa účtování** – úkolem je sběr dat, které souvisejí s využíváním síťových prostředků. Využití takový dat nalezneme při výpočtu skutečných nákladů u zákazníků. Zdrojem dat pro tento účel může být například protokol RADIUS.
- **správa výkonnosti** – provádí dohled a následné posouzení výkonnosti sítě a jejich prvků. Dlouhodobé statistiky se mohou stát podkladem pro optimalizaci, která může vést ke zlepšení výkonu.
- **správa bezpečnosti** – slouží k zabezpečení síťových prostředků pomocí autorizace a autentizace. Chrání před útoky a provádí záznam bezpečnostních incidentů. Vhodným příkladem je zabezpečení standardem IEEE 802.1X.

Cílem centrální správy sítě je přinést unifikované prostředí, které umožňuje zjednodušit a zrychlit správu samotnou. Existuje řada komerčních řešení, které sjednocují správu a monitoring do jednoho centrálního bodu. V našem případě se tomu pokusíme alespoň přiblížit. Hlavní přínosem centrální správy by měla být vyšší míra bezpečnosti a provozní spolehlivost základních síťových služeb [26].

2 Zabezpečení počítačových sítí

Se zabezpečení počítačových sítí souvisí především datová a síťová bezpečnost. Datová bezpečnost řeší zabezpečení dat. Vzhledem k tomu, že nemůžeme spoléhat pouze na šifrovací algoritmy, tak se zabýváme také síťovou bezpečností. Je potřeba zajistit bezpečnost také na úrovni komunikační kanálů a síťových prvků, kterými data procházejí.

Při vytváření zabezpečené sítě bychom měli zachovat tzv. CIA[1] triádu, viz obrázek 2.1. Definuje model pro vývoj bezpečnostních politik spolu s nezbytnými řešeními v oblasti informační bezpečnosti [27].



Obr. 2.1: CIA triáda

Důvěrnost

V současné době je velmi důležité, aby lidé chránili citlivé údaje. Důvěrnost má zajistit, že informace jsou přístupné nebo sděleny pouze oprávněným osobám. V komerční sféře se nejčastěji využívají následující klasifikační stupně pro práci s informacemi:

- důvěrné informace – nejvyšší stupeň, zpřístupnění může mít zničující dopad pro daný subjekt (know-how)
- soukromé informace – zpřístupnění může mít negativní dopad pro daný subjekt (osobní data zaměstnanců nebo klientů)
- citlivé informace – zpřístupnění může mít negativní dopad pro daný subjekt (plánované změny nebo informace o projektech)

[1] Confidentiality, Integrity, Availability

- veřejné informace – nejnižší stupeň, zpřístupnění je bez dopadu pro daný subjekt, veřejně známé informace (e-maily, telefonní čísla, zaměstnanci)

Ochrana důvěrnosti dat závisí na vynucení dané úrovně přístupu k informacím. To by nás mělo nutit data rozdělovat na základě přístupu k nim. Současně bychom měli být schopni odhadnout výši škody při porušení důvěrnosti dat. Současně bychom měli být schopni řešit případnou výši škody při porušení důvěrnosti dat. Nejběžnější způsob zajištění důvěrnosti je šifrování dat. Přijetím opatření jako jsou silná hesla, dvoufázové ověření, biometrické ověření a různé bezpečnostní tokeny se důvěrnost navyšuje [29, 25].

Integrita

Integrita zajišťuje správnost a úplnost informace. Chrání před vymazáním nebo úpravou neoprávněnou osobou. Zajišťuje také, aby v případě změny bylo možné vrátit původní stav. Aby tohle bylo zaručeno, tak se přijímají patřičné opatření. Některé data zahrnují kontrolní součty nebo kryptografické kontrolní součty pomocí hashů pro ověření integrity. Abychom však mohli postižená data obnovit do správného stavu, tak musíme mít správně nastavený proces pro zálohování dat. Pro zajištění integrity se obvykle dělají různé kontrolní součty [31].

Dostupnost

Cílem je zajistit dostupnost dat. Součástí jsou mechanismy pro ověřování, přístupové kanály a systémy, které musí být funkční. Vysokou dostupnost by měly mít systémy, které jsou navrženy pro zlepšení dostupnosti. Při návrhu takových systémů je potřeba brát v potaz selhání hardwaru, neúspěšný upgrade nebo výpadky napájení.

U systémů se často uvádí jejich dostupnost v procentech. Daleko praktičtější je uvádění RTO (Recovery Time Objective) a RPO (Recovery Point Objective). RTO určuje jak dlouhý výpadek může být tolerován. Při zcela redundantní infrastruktuře by RTO mohlo být rovno 0. RPO popisuje jaké množství dat může být ztraceno. Například záloha se provádí v 0:00 a ve 12:00, těsně před polednem dojde k havárii diskového pole. Tím jsme ztratili data za posledních skoro 12 hodin a jejich obnova potrvá asi 2 hodiny. RPO je dáno dobou výpadku dokud není obnovena poslední záloha a daná část infrastruktury plně funkční, takže v našem případě může být RPO 14h. Pro tyto situace je vhodné zavedení tzv. Disaster Recovery (obnovení po havárii), které zajišťuje obnovení IT služeb po havárii.

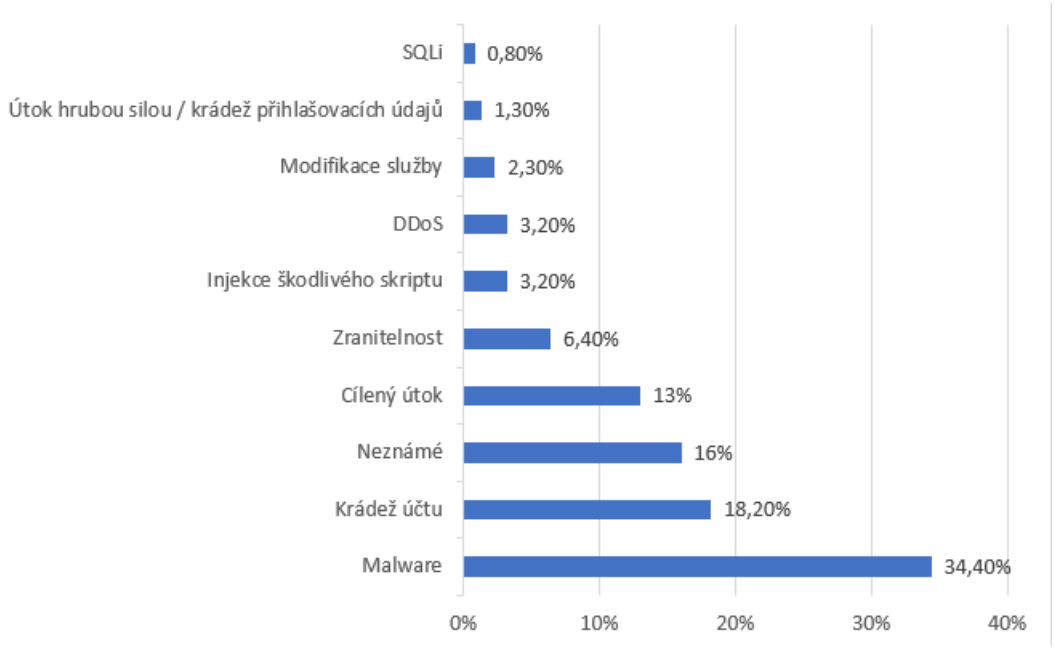
Dostupnost může být nejlépe zajištěna pečlivou údržbou – hardware, software, aktualizace. Zmírnit škody při výpadku může redundantní zapojení prvků, geogra-

fické zálohy nebo dostatek zapojených disků v RAID[2] [30]. Pořád však platí, že čím více disků, tím větší pravděpodobnost selhání. Je vhodné kombinovat disky různých výrobců nebo jiné série, kvůli případným výrobním vadám.

Triáda je považována za klíčový faktor pro zabezpečení IT a počítačových sítí. Přesto poskytuje pouze omezený pohled na zabezpečení a ignoruje některé další důležité faktory. Zabezpečení informací samo o sobě nezaručuje, že někdo nevyužije systémové prostředky bez oprávnění. Triáda má pomoci při implementaci kvalitní bezpečnostní politiky a vysvětlit principy za ní. Důležité je také pochopit omezení, které představuje.

2.1 Rizika a hrozby

V případě, že má dojít k připojení sítě k Internetu, tak je vhodné vyhodnotit všechny možné hrozby a rizika. Riziko je potencionální možnost, že nějaká hrozba využije zranitelnosti. Rizika uvnitř sítí se podceňují, přitom nejvíce útoků pochází právě odtud.



Obr. 2.2: Nejčastějších 10 útoků za rok 2018 [32].

V tomto případě se hrozby rozdělují podle toho, kdo by mohl být jejich původcem. Hrozby jsou pojaty z obecného hlediska a popisují konkrétní příklady. Následkem může být úspěšný kybernetický útok nebo únik dat.

[2]způsob práce se dvěma nebo více pevnými disky jako s jednou logickou datovou jednotkou

Hrozby z administrátorského hlediska

- **výchozí přístupové údaje** – na aktivních síťových prvcích zůstaly výchozí přihlašovací údaje (admin/admin, root/root) nebo nejsou zabezpečeny vůbec.
- **chybějící aktualizace** – na serverech funguje zastaralá verze BIOS, kvůli které je systém zranitelný na chybu Meltdown[3]. Na přístupovém bodu pracuje zastaralý firmware, který trpí na zranitelnost, díky které lze obejít autentizaci do administrátorského rozhraní.
- **nevyhovující síťové prvky** – v síťové infrastruktuře je přepínač, který nepodporuje management, žádné bezpečnostní funkce a ani nenabízí možnost monitoringu. Přístupový bod pro zabezpečení nabízí pouze zastaralé protokoly WEP a WPA.
- **nevyužité služby a aplikace** – na veřejné IP adrese je otevřené NTP a DNS, i když nemají žádné využití. V interní síti se nachází služba.
- **zranitelné nebo zastaralé služby** – pro přístup k síťovým prvkům je používán protokol Telnet a pro přenos konfiguračních souborů protokol TFTP. Přístup k webovému konfiguračnímu rozhraní podporuje pouze HTTP. Síťové zařízení poskytovatele Internetu jsou spravovány protokolem SMTP v1. Pro vzdálený přístup se používá protokol PPTP nebo se obecně používá slabé zabezpečení.
- **chybná konfigurace** – na straně poskytovatele Internetu je špatná konfigurace firewallu, která umožňuje přistupovat zákazníkům do jejich vnitřní sítě (špatné ACL mezi VLANy). Klienti bezdrátové sítě pro hosty dostávají IP adresy z interní podsítě, ve které jsou současně interní služby. Špatná konfigurace webového serveru nevynucuje připojení přes HTTPS, i když je tato funkce podporována. Špatná konfigurace sdílených složek umožňuje přístup neoprávněným lidem.
- **nezodpovědnost administrátora** – nedostatečné logování a monitoring sítě. Nedostatečné zálohování nebo zálohování v nevhodnou dobu. Používání zastaralých řešení a konceptů. Nedostatečná dokumentace sítě. Špatně nastavené procesy pro řešení problémů. Nedostatečná úroveň zabezpečení sítě a uživatelských počítačů. Neexistence bezpečnostní politiky. Nepoužívání dvoufázového ověření.

Hrozby z uživatelského hlediska

- **slabé nebo žádné heslo** – uživatel k přihlášení do počítače nepoužívá žádné heslo a na emailovou schránku se přihlašuje pomocí svého data narození.

[3]dovoluje uživatelským procesům číst data z libovolné části fyzické paměti

- **chybějící aktualizace** – uživatel vypnul automatické aktualizace systému, protože jej vyrušovali při práci. Vzniká riziko výskytu zranitelnosti v daném systému.
- **nedostatečná ochrana hesla** – uživatel má heslo umístěné na monitoru nebo pod klávesnicí. Všechny přístupové údaje uživatel ukládá v textovém dokumentu na ploše svého systému.
- **úniky dat** – uživateli byl ukraden počítač s cennými daty, úložiště však nebylo zašifrované nebo bylo použito slabé heslo.
- **automatizace vzdáleného přístupu** – uživatel má nastavené automatické připojení k VPN, aniž by musel zadávat heslo. V horším případě nemá nastavené heslo pro samotný přístup k počítači.
- **nezodpovědný přístup** – otevírání odkazů a příloh z emailu neznámého původu. Návštěva nevhodných webových stránek.

Ať už chceme nebo ne, tak největším zdrojem hrozeb je člověk. K útokům může docházet přímo (hackerem) nebo nepřímo (provozem IT systémů). Je vhodné rozlišovat hrozby pocházející zevnitř nebo zvenku. Externí hrozby se dají omezit prostřednictvím bezpečnostních prvků na perimetru sítě. V případě hrozby přicházející zevnitř je tento ochranný perimetr překonán. Zdrojem takové hrozby může být některý z uživatelů tzv. insider, který obvykle má podrobné informace o fungování systémů [36].

2.2 Vnitřní útoky na síť

Obecně vedou k odposlechu síťové komunikace nebo její modifikaci či ovládnutí samotné sítě. Většina útoků na sítě pochází od vnitřních uživatelů se špatnými úmysly případně od útočníků v jejím okolí. Existují obranné mechanismy, kterými lze některým útokům předcházet. [33]

- **IP spoofing** – technika, která umožňuje podvržení zdrojové IP adresy. Používá se především pro DoS útoky nebo pokud chceme utajit původ paketů. Využití nachází také u vnějších útoků na síť.
- **Wi-Fi hacking** – předpokladem jsou zařízení, které mezi sebou komunikují na bezdrátové síti. Útočníci mohou snadno sledovat provoz na základě SSID dané bezdrátové sítě. Můžou se pokusit o prolomení zabezpečení hrubou silou (brute force) nebo využít nějakého zranitelného protokolu. Další útok může probíhat pomocí falešného AP, které se tváří jako důvěryhodné. Pokud je signál od falešného AP silnější, klienti se přepojí na něj a útočníkům již ke sledování provozu nebrání téměř nic. Navíc existují útoky na protokoly používané pro

zabezpečení bezdrátových sítí, konkrétně na protokoly WEP, WPA a některé varianty WPA2 [34].

- **DHCP starvation** – obvykle předchází útoku DHCP spoofing. Útočníkův počítač se snaží vystupovat jako více klientských počítačů. Cílem je vyčerpat IP adresy přidělené oficiálním DHCP serverem.
- **DHCP spoofing** – po konfiguraci vlastního DHCP serveru připojeného k napadené síti dojde k distribuci IP adresy s podvrženým DNS serverem nebo výchozí branou, přes kterou pak následně proudí veškerý provoz připojených klientů. Vzniká tak útok mužem uprostřed (Man-in-the-middle).
- **ARP spoofing** – předchází útoku MiTM, umožňuje neoprávněnou modifikaci ARP tabulky. Vede k posílání provozu na MAC adresu útočníka, který pak může provoz přeposílat na cílovou stanici. Jde o jeden z útoků mužem uprostřed (Man-in-the-middle). [35]
- **Session hijacking** – cílem je získat kontrolu nad relací mezi klientem a serverem. Všechny nešifrované spojení jsou zranitelné na tento typ útoků. Dochází ke zneužití HTTP cookie. Předpokladem je, že jsme úspěšně provedli útok mužem uprostřed (man-in-the-middle). Tento typ útoku se kombinuje s technikou SSL stripping, která se zaměřuje na šifrovaná spojení. V praxi to funguje tak, že útočník ustanoví legitimní spojení s cílovým serverem přes HTTPS, ale provoz k oběti se downgraduje na HTTP a přitom se snaží vystupovat jako legitimní server. Teoreticky by bylo možné útočit i bez SSL strippingu za předpokladu, že na straně oběti importujeme CA našeho vlastního certifikátu, který používáme pro simulaci cílového serveru pomocí HTTPS. Tento typ útoku se označuje jako HTTPS spoofing.
- **MAC flooding** – cílem je přetečení paměti přepínače, což způsobí zasílání provozu na všechny porty. Přepínač se začne chovat jako rozbočovač. Toho lze snadno využít pro poslech veškeré komunikace.

2.3 Vnější útoky na síť

Útoky, které vedou k odepření služby nebo přístupu k ní z veřejné sítě Internet. Některým hrozbám lze zabránit účinným firewallem nebo antivirovým programem. Obvykle jsou tyto útoky doprovázeny formou průzkumu jako je skenování otevřených portů, analýzou služeb a dostupných zdrojů.

- **DNS spoofing** (DNS cache poisoning) – technika, jejímž výsledkem je podvržení IP adresy na úrovni DNS. Útočník tak může vytvářet falešné DNS záznamy, které mohou obsahovat škodlivý obsah. Tento útok je velmi podobný

útoku na ARP. V praxi to funguje tak, že uživatel zadá adresu www.vutbr.cz, ale je přesměrován na jiný podvodný web.

- **Phishing** – útok, kterým se útočník snaží najít nebo získat citlivé informace prostřednictvím elektronické komunikace. Obvykle se provádí automatizovaně od související a důvěryhodné domény.
- **E-mail spoofing** – tento útok umožňuje podvržení e-mailové adresy odesílatele z nějaké známé domény, takže si příjemce myslí, že tuto osobu zná a e-mail otevře. Obvykle se v e-mailu vyskytují podezřelé odkazy nebo přílohy.
- **URL spoofing** – cílem tohoto útoku je podvržení cílové URL. Kliknutí na takový odkaz může způsobit infekci počítače nějakým malwarem nebo provedení nechtěné akce. Pro vytvoření takového odkazu se používají podobné domény, URL zkracovače nebo hyperlinky pro skrytí skutečného odkazu. V jiném případě může útočník využít důvěryhodné stránky, která má v sobě zranitelnost otevřené přesměrování (Open Redirect) a využít ji pro svůj účel. Bývá součástí phishingových e-mailů.
- **Sociální inženýrství** – snaha zmanipulovat uživatele za účelem provedení akce, která se po něm žádá. Obvykle prostřednictvím phishing útoku.
- **Malware** – infekce malwarem je jeden z možných způsobů jak ovládnout počítač a data v něm. Mezi příklady malware patří viry, trojské koně, rootkity, spyware, ransomware, počítačové červy a další. Existuje velká škála škodlivých kódů, které mohou otevřít zadní vrátka do systému nebo zašifrovat celý počítač.
- **APT** (Advanced Persistent Threat) – představují sofistikované hackerské techniky zaměřené na konkrétní cíl. Útok může být zaměřen na organizace, instituce nebo státy. Obsahují malware, který je přímo určen pro konkrétní cíl. Využívají se dosud neobjevená zranitelná místa. Součástí mohou být cílené techniky sociálního inženýrství a phishingu. [41]
- **DoS, DDoS** – účelem útoku je pokus o nedostupnost služby online nebo webu jejím přetížením z generovaného provozu. Při útoku odepření služby DoS obvykle figuruje pouze jeden počítač k zaplavení cílového počítače. Distribuované odepření služby DDoS používá mnoho počítačů současně, často se jedná o tzv. botnety. Takový útok ve velkém měřítku může generovat provoz v desítkách až stovkách gigabitů za sekundu, což běžná síť nemůže zvládnout. [40]
- **Prohlížečové útoky** – převážně útoky na webové aplikace nebo webové prohlížeče prostřednictvím webových zranitelností jako je XSS, SQLi, Broken Authentication.

2.4 Způsoby zabezpečení sítí

Existuje řada způsobů jak zajistit uspokojivou úroveň zabezpečení v počítačové síti. Je nutné zmínit, že bezpečnost není stav, ale neustálý proces, jehož cílem je daný stav udržovat.

Pro zabezpečení sítí je velmi důležitý bezpečný přenos dat. Zajištění bezpečné komunikace v síti je rozděleno následovně:

- **autentizace** – zastupuje ověření samotného zařízení, že je za co se vydává. Předmětem je uživatelské jméno nebo certifikát.
- **autorizace** – představuje poskytnutí práv ke zdrojům uživateli nebo zařízení. Cílem je zjistit zda ověřený uživatel má práva pro používání systému. Např. někteří uživatelé mohou mít práva pouze pro čtení nebo mají přístup jen do určité sekce.
- **soukromí** – je zajištěno šifrováním komunikace, které pomáhá chránit data před jejich zneužitím. Šifrování může být náročný proces pro systémové zdroje. V praxi dosahujeme soukromí pro uživatele pomocí metod jako VPN nebo protokol HTTPS. [39]

Vnější zabezpečení sítě

Vnější zabezpečení sítě se vztahuje k rozhraní mezi vnitřní a vnější sítí, obvykle sítí Internet. Pro ochranu tohoto rozhraní používáme:

- směrovač (brána)
- firewall (NGFW, aplikační firewall)
- systémy IDS/IPS

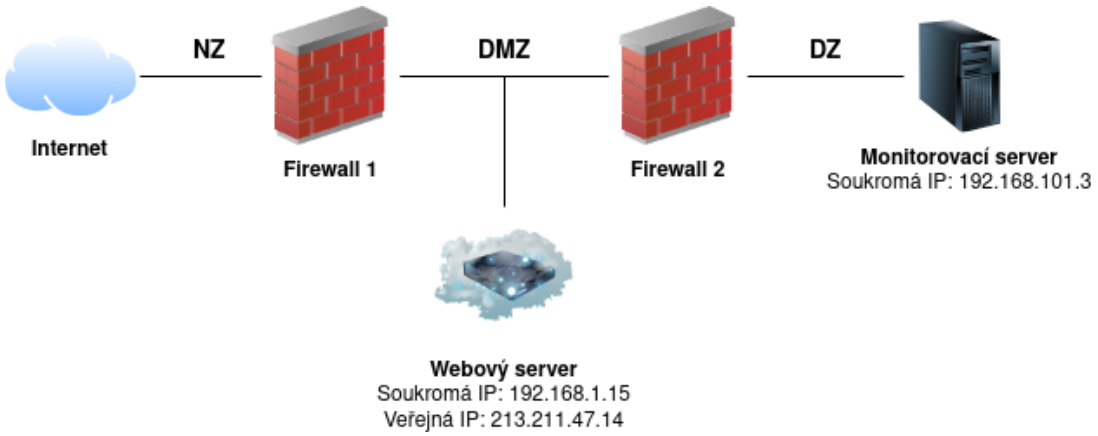
Pro vzdálený přístup prostřednictvím vnější sítě se obvykle využívá zařízení Virtual Private Network (VPN), které zajišťuje bezpečnou komunikaci mezi vzdálenou koncovou stanicí a vnitřní sítí.

Vnitřní zabezpečení sítě

Vnitřní zabezpečení sítě představuje zabezpečení prostředků, které se nasazují ve vnitřní počítačové síti. V síti mohou existovat zařízení, kterým můžeme přesně určit s kým mohou a s kým nemohou komunikovat. V závislosti na dané hierarchii sítě lze provést rozdělení na následující celky:

- **důvěryhodná zóna** (DZ) – nastavení filtrace provozu omezuje provoz dovnitř a ven z takových sítí. V takových sítích nacházíme kritické služby infrastruktury jako je IDS/IPS senzor, server pro zálohy a monitoring.

- **demilitarizovaná zóna** (DMZ) – označuje část sítě, kterou sice máme pod kontrolou, ale nelze u ní plně kontrolovat bezpečnost. Očekáváme, že v této síti mohou vznikat bezpečnostní problémy. Obvykle jde o veřejně poskytované služby. Jde o logickou podsíť, která je obvykle izolovaná a nemá přístup do DZ. Setkáváme se zde s webovým serverem, e-mailovým serverem, DNS případně nějakou další aplikací.
- **nedůvěryhodná zóna** (NZ) – vše co je mimo naši kontrolu považujeme za nedůvěryhodné. V našem případě to je síť Internet, jelikož je potenciálně nebezpečná a v případě vzniku nějakých bezpečnostních problémů nejsme schopni v místě jejich vzniku s nimi cokoliv dělat.



Obr. 2.3: Příklad sítě s DMZ a dvojitým firewallem

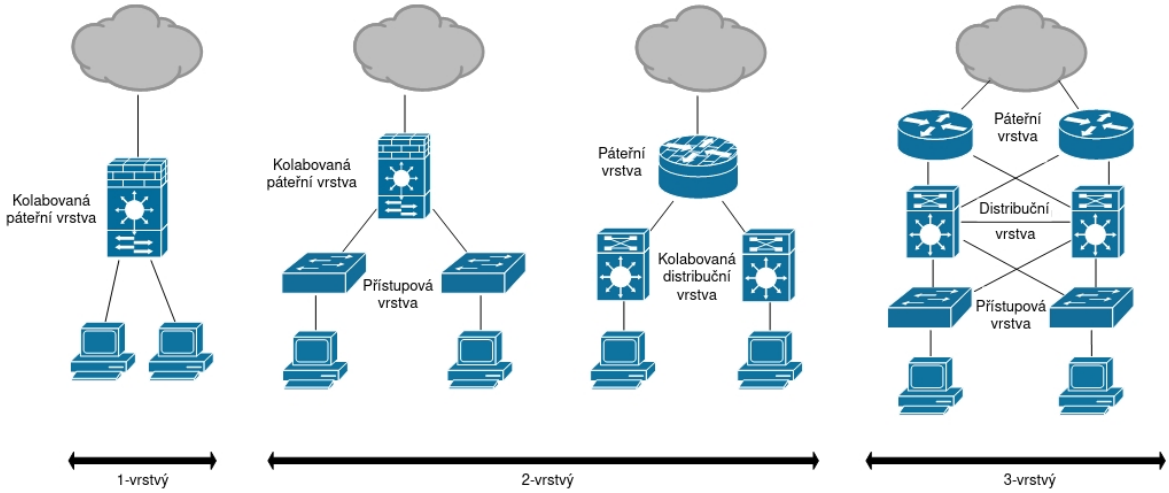
Mobilní zařízení

Při zabezpečování sítí je potřeba věnovat pozornost také mobilním zařízením, které se připojují do počítačové sítě. Za mobilní zařízení považujeme například notebooky, mobilní telefony nebo tablety. Řada z nich může obsahovat citlivé údaje, takže je vhodné předem rozhodnout, jakým způsobem budou chráněna. Jeden z účinných prostředků je šifrování. Další možností je umožnit vzdálený výmaz zařízení (wipe). Cílem je zamezit zneužití zařízení v případě ztráty nebo krádeže. [36]

Nároky na bezpečnost zvyšuje trend zvaný Bring Your Own Device (BYOD), který je založen na tom, že uživatelé používají své vlastní zařízení a těžko se pak na ně uplatňují nastavené bezpečnostní politiky. Pro vynucení specifické bezpečnostní politiky pro mobilní zařízení se uplatňuje Mobile Device Management (MDM), který podporuje plnou kontrolu nad zařízením nebo nad jeho vybranou částí. [37]

3 Metodika návrhu zabezpečené sítě

Tato metodika se věnuje doporučením a zásadám při návrhu zabezpečené počítačové sítě. U návrhu sítí se setkáváme s hierarchickým modelem pro vícevrstvé sítě, jenž se obvykle skládá ze sítě přístupové, distribuční a páteřní. V praxi se však u malých a středně velkých sítí setkáváme spíše s nejvýše dvouvrstvým modelem. První vrstva (páteřní) se stará o rozdělení sítě do VLAN, směrování mezi VLAN, kontrola provozu pomocí ACL nebo vzájemné propojení přepínačů z druhé vrstvy. Druhá vrstva (přístupová) připojuje zařízení do sítě a zajišťuje bezpečnost na portech. V případě jednovrstvého modelu jsou všechny uvedené funkce sloučeny do jedné vrstvy.



Obr. 3.1: Srovnání hierarchických modelů [45]

Hierarchické modely se používají v závislosti na počtu uživatelů v dané síti. U jednovrstvého modelu se předpokládá nejvýše 50 uživatelů. V rámci dvouvrstvého modelu se počítá něco mezi 50 až 100 uživateli a třívrstvý model je vhodný pro více jak 100 uživatelů. [44]

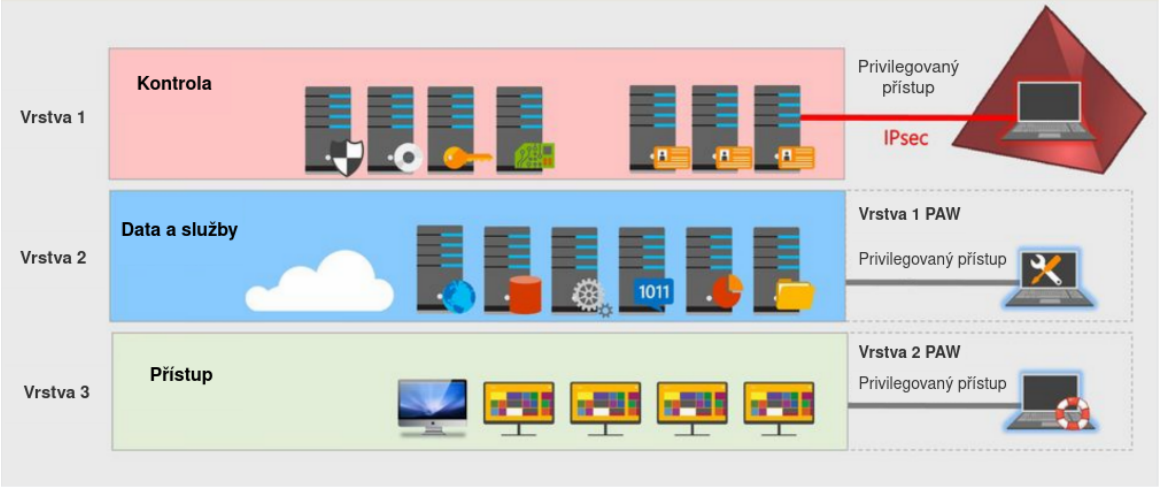
Pokud máme být schopni čelit hrozbám v oblasti počítačových sítí, tak je potřeba mít zavedené následující prostředky:

- **správa systémů** – pravidelné aktualizace minimalizují rozsah možných útoků a centrální správa usnadní proces obnovy.
- **filtrování** – použití vhodných firewallů může zastavit malware, detekovat znaky podezřelého chování a hrozeb.
- **detekce průniku** – aplikace pro monitoring sítě a systémy detekce průniku mohou zavčas odhalit škodlivé nebo podezřelé aktivity.

- **šifrování** – používání kryptografických protokolů jako je TLS a SSH může chránit konkrétní části sítě před některými síťovými útoky. [4]

3.1 Architektura sítě

Pojmem architektura sítě je v této části myšlen celkový design a struktura prvků v rámci sítě. Pro úpravu přístupu mezi službami můžeme použít model vrstev pro dělení administrativních oprávnění (PAM – Privileged Access Management) spolu s privilegovanými přístupovými stanicí (PAW – Privileged Access Workstations). Cílem PAM je chránit před zvýšením oprávnění oddělením aktivit s vysokými oprávněními od vysoce rizikových oblastí. PAW představuje počítače, které mají přístup do příslušné vrstvy a mohou spravovat zařízení v ní. Pro všechny počítače, které spravují vyšší vrstvy, by měli platit přísné restrikce. Model byl definován z bezpečnostních důvodů společností Microsoft a týká se dělení administrativních oprávnění ve strukturách Active Directory (AD). Tento model lze však aplikovat na jakoukoliv počítačovou síť. Očekává se, že přístup mezi vrstvami je ve směru dolů omezen a ve směru nahoru zakázán. Myšlenkou je, že pokud dojde ke kompromitaci vrstvy nižší, tak z ní pak nemůže dojít ke kompromitaci vrstvy vyšší. Každý koncept zabezpečené sítě by měl částečně vycházet z uvedeného modelu. Na obrázku 3.2 je cílem rozdělit a omezit přístupy mezi službami pro kontrolu, úložišti pro data a samotnými uživateli v síti. Tento způsob vrstvení sítě nepřímo souvisí s její segmentací. Segmentace již přímo řeší rozdělení sítě do VLAN a uplatnění ACL mezi nimi. Vytvoření jednotlivých vrstev po vzoru uvedeného modelu pak může mnohonásobně usnadnit celý proces segmentace [46, 47].



Obr. 3.2: Model vrstev prostředí PAM [52].

Pravidla rozdělení sítě do vrstev:

- **rozdělovat do vrstev na základě důvěryhodnosti** – kompromitace jedné vrstvy může znamenat kompromitaci všech vrstev pod ní. Například není vhodné umístit počítače uživatelů do vrstvy vyšší než síť pro management.
- **přihlašovací údaje musí být zadávány pouze v rámci dané vrstvy nebo z vyšší vrstvy do nižší** – například ani přes SSH bychom neměli být schopni přistupovat z klientské sítě do vrstvy managementu na konfiguraci routeru.
- **zařízení z nižší vrstvy nesmí nikdy spravovat zařízení z vyšší vrstvy** – například služba pro centrální správu umístěná do vrstvy služeb nesmí spravovat síťové zařízení ve vrstvě managementu [50].

V mnoha počítačových sítích spoléhají pouze na funkci DMZ. Zapomínají však na to, kolik takových sítí již bylo kompromitováno. Základní stavebním kamenem bezpečné sítě je její segmentace. Ta zajišťuje zásadní ochranu i před hrozbami, které jsou již uvnitř. Vzniklé bezpečnostní incidenty navíc ukazují, že je důležité mít pečlivě implementovanou a udržovanou segmentaci sítě.
Provedení efektivní segmentace sítě lze provést v následujících krocích:

- **pochopení struktury a jejich procesů** – je nutné vědět, co je potřeba chránit, jak vstupují například finance nebo data do organizace. Jaké komponenty se používají v popředí a co je základní funkcí v pozadí. Jaká aktiva, data nebo personál je nezbytný pro fungování dané organizace.
- **vytvoření plánu** – návrh klasifikace, izolace a ochrany pro nejdůležitější zařízení v síti. Související položky by se měly seskupit do jedné VLAN. Jednou skupinou může být infrastruktura, která se skládá ze směrovače, přepínače a VPN serveru. V další skupině mohou být prvky pro zajištění bezpečnosti – firewall, IDS/IPS. Servery provozující činnost důvěrné povahy by měli mít také vlastní VLAN. To stejné platí pro stanice uživatelů.
- **mapování přístupů** – vytvoření přehledu o přístupech uživatelů k jednotlivým systémům. Pokud neexistuje potřeba, tak by neměl existovat ani přístup. Přístup může být regulován například ke kamerovým systémům, serverům monitoringu nebo ke správě sítě. Vše závisí na konkrétních systémech a uživatelích. Pokud organizace funguje pouze na místní úrovni, tak je vhodná blokace zeměpisně vzdálených regionů na základě IP adresy. Standardně se doporučuje provést blokaci pro každou VLAN jako výchozí a povolit pouze přístupy, které jsou žádoucí.
- **zavedení segmentace** – v některých sítích je segmentace sítě dlouhodobým projektem a může vyžadovat značnou dávku úsilí. Každý takový zásah ale zvyšuje úroveň zabezpečení [48].

3.2 Síťové služby a zařízení

Pro vytvoření následujících doporučení byly použity bezpečnostní technické implementační příručky (Security Technical Implementation Guide – STIG). Jedná se o konfigurační standardy vytvořené obrannou komunikačních agenturou (Defence Information Systems Agency – DISA) pro Ministerstvo obrany Spojených států amerických (Department of Defense – DoD). STIG obsahuje technické pokyny pro uzamčení informací, systémů a softwaru, které by mohly být ohroženy útokem. [51]

Doporučení pro správu sítě

- Pro správu sítě a síťových prvků se výhradně používají protokoly SSH, SSL/TLS, HTTPS nebo SNMPv3.
- Při ochraně perimetru musí být firewall umístěn mezi vnitřní sítí, hraničním směrovačem a DMZ.
- Zprávy SNMP se musí ukládat po dobu minimálně 30 dní a poté se archivují.
- Šifrovat se musí všechny uložené konfigurace síťových zařízení.
- Protokol DHCP (Dynamic Host Configuration Protocol) musí zaznamenávat názvy hostitelů a MAC adresy, které se ukládají online po dobu třiceti dnů a offline po dobu jednoho roku.
- Syslog server je kritickou částí síťové bezpečnosti.
- Syslog server by měl shromažďovat zprávy syslog z úrovní 0 až 6.
- NTP server by měl být připojen pouze do sítě managementu.
- v síti pro správu musí být nasazeny dva servery NTP (Network Time Protocol).
- Pro přenos konfigurace na zařízení se používá výhradně SCP nebo vyměnitelné médium.
- Server TFTP používaný k ukládání konfigurací a obrazů síťových zařízení musí být připojen pouze k síti pro správu.
- Všechny síťové zařízení vystavené do Internetu musí být umístěné do DMZ.
- Všechny síťové zařízení musí být umístěna do zabezpečené místnosti s omezeným přístupem.
- Při ochraně perimetru sítě se musí implementovat DPI.
- Implementace dvoufaktorové autentizace na všechny síťové prvky. [54, 53]

Doporučení pro síťové zařízení

- Síťová zařízení musí vynutit šifrování hesla při přenosu.
- Síťová zařízení nesmí mít žádná výchozí hesla výrobce.
- Síťové zařízení musí zakázat opakované použití hesla po dobu nejméně pěti opakování.
- Síťová zařízení musí být chráněna heslem s určitou složitostí.

- Síťové zařízení musí vynutit minimální délku hesla 15 znaků.
- Síťové zařízení nesmí mít povolené nadbytečné služby a funkce.
- Síťové zařízení musí vyžadovat ověření pro přístup z konzole.
- Síťová zařízení musí odepřít přístup k portu konzoly při nečinnosti větší jak 10 minut.
- Síťové zařízení nesmí používat skupinové účty.
- Síťové zařízení nesmí umožnit připojení použitím SSH verze 1.
- Síťové zařízení musí mít nakonfigurován maximální počet neúspěšných pokusů o přihlášení přes SSH na 3 pokusy.
- Síťová zařízení musí být nakonfigurována na vypršení časového limitu po 60 sekundách nebo méně pro neúplné nebo přerušené relace SSH.
- Síťová zařízení musí přijímat připojení pro správu pouze od hostitelů sídlících v síti pro správu.
- Logy síťových zařízení musí být časově označeny.
- Síťové zařízení musí logovat všechny zprávy kromě ladění (debug) a odesílat všechna data protokolu na syslog server.
- Logy síťových zařízení musí zahrnovat zdrojovou adresu, cílovou adresu, port, použitý protokol a provedenou akci.
- Síťové zařízení musí mít administrátorské rozhraní přes HTTP vypnuté.
- Síťové zařízení musí používat protokol SNMP verze 3. Předchozí verze 1 a 2 se nepovažují za bezpečné.
- Síťové zařízení nesmí používat veřejné nebo soukromé výchozí nebo známé řetězce komunity SNMP.
- Síťová zařízení musí umožňovat přístup přes SNMP pouze pro čtení.
- Pro snížení rizika u protokolu SNMPv3 by měly síťové prvky používat autentizaci pomocí HMAC-SHA a šifrovat obsah komunikace pomocí AES s největší délkou klíče jaká je podporována.
- Na síťovém zařízení musí být deaktivována služba nebo funkce, které komunikuje s výrobcem zařízení.
- Síťové zařízení musí zajišťovat autentizaci, ochranu integrity a důvěrnosti pomocí kryptografických mechanismů pro externí správu a diagnostiku.
- Pro hranici mezi sítí správy a spravovanou sítí se musí používat ACL.
- Všechna fyzická, logická a virtuální rozhraní musí být nakonfigurovány s ACL, aby se zamezilo neoprávněnému provozu z jedné sítě do druhé.
- Síťové zařízení musí logovat všechny pokusy o navázání připojení pro přístup do své správy.
- Síťové zařízení musí logovat všechny případy odepření přístupu pro ACL.
- Síťové zařízení musí ve výchozím nastavení zahazovat provoz na všech rozhraních perimetru sítě a povolovat pouze vyjímky pro konkrétní síťový provoz

- Síťové zařízení musí bránit přístupu do interních sítí, s výjimkou případů, kdy je to výslovně povoleno a řízeno pomocí zařízení pro ochranu perimetru.
- Síťové zařízení musí využívat automatizované mechanismy k centrálnímu ověření nastavení konfigurace.
- Síťové zařízení nesmí dovolit uživatelům zavést vyměnitelná média do informačního systému.
- Síťové zařízení musí neaktivní účty automaticky deaktivovat po 35-denní nečinnosti.
- Síťové zařízení musí používat interní systémové hodiny pro generování časových razítek pro záznamy auditu.
- Síťové zařízení musí vynutit 24 hodin jako minimální životnost hesla a 60 dní jako maximální životnost hesla. [58, 55, 57]

Doporučení pro hraniční síťové prvky a směrovače

- Funkce pro úpravu směrování zdrojem musí být zakázána (source routing).
- FTP servery musí být vypnuty.
- Relace L2TPv3 musí být ověřeny před přenosem provozu.
- Výchozí směrovací trasy (default route) nesmí být směrovány do VPN tunelu.
- Správce musí zajistit, aby byl na všech externích rozhraních zablokován protokol SNMP.
- Síťové zařízení musí mít zakázané SNMP na externích rozhraních.
- Příchozí ICMP zprávy z externích nedůvěryhodných sítí musí být blokovány.
- Odchozí ICMP zprávy do externích nedůvěryhodných sítí musí být blokovány.
- Příchozí IP pakety s adresou zpětné smyčky hostitele (127.0.0.0/8) musí být na obvodovém zařízení blokovány, zamítnuty nebo zrušeny.
- Pakety z VPN tunelu musí být filtrovány v místě výstupu.
- Síťová zařízení musí mít definovány servery DNS, pokud jsou nakonfigurovány jako klientský DNS.
- Odchozí zprávy ICMP o překročení času (Time Exceeded) musí být zablokovány, aby se zabránilo objevení sítě neoprávněnými uživateli. [59, 60]

Doporučení pro přepínače

- Přepínač L2 musí mít všechny trunk spoje aktivované staticky.
- Přepínač L2 musí implementovat Rapid STP pokud VLAN pokrývají více přepínačů s redundantními linkami.
- Přepínač L2 musí povolit maximálně jednu registrovanou MAC adresu na každý port pro uživatele. (Port Security)
- Přepínač L2 musí umožňovat oprávněným uživatelům vzdáleně zobrazit veškerý obsah související se zavedenou relací uživatele v rámci daného portu.

- Přepínač L2 musí mít deaktivované nepoužívané služby a funkce.
- Přepínač L2 musí ověřovat všechny zprávy VTP Trunk Protocol (VTP) pomocí hashovací funkce a pomocí nejbezpečnějšího dostupného kryptografického algoritmu.
- VLAN Trunk Protocol (VTP) musí být autentizován pomocí hashe a nejbezpečnějšího dostupného kryptografického algoritmu.
- Přepínač L2 musí řídit nadměrnou šířku pásma, aby se omezil účinky útoků typu DoS.
- Přepínač L2 nesmí mít žádný přístup k portům přiřazeným do nativní sítě VLAN.
- Přepínač L2 musí mít nativní VLAN přiřazenu k jinému ID, než k výchozí VLAN pro všechny trunk spojení 802.1q (VLAN 1).
- Přepínač L2 musí mít na všech uživatelských VLAN povolenou Inspekci protokolu ARP (Dynamic Address Resolution Protocol) (DAI).
- Přepínač L2 musí mít aktivovanou ochranu zdroje IP (IP Source Guard) na všech uživatelských nebo nedůvěryhodných přístupových portech.
- Přepínač L2 musí mít DHCP snooping pro všechny uživatelské VLAN, aby ověřil DHCP zprávy z nedůvěryhodných zdrojů.
- Přepínač L2 musí mít povoleno Unknown Flood Blocking (UUFB).
- Přepínač L2 musí ověřit všechna koncová zařízení před navázáním síťového připojení pomocí obousměrného ověřování, které je založeno na kryptografii. (802.1X)
- Přepínač L2 musí mít na všech portech, kde by se kořenový most (root bridge) neměl objevit, povolen Root Guard.
- Přepínač L2 musí mít aktivovanou ochranu STP Loop Guard na všech neurčených portech přepínače STP.
- Přepínač L2 musí mít aktivovanou ochranu BPDU na všech uživatelských nebo nedůvěryhodných přístupových portech.
- Přepínač L2 musí mít nativní VLAN odebranou ze všech trunk portů, které to nevyžadují.
- Přepínač L2 nesmí používat výchozí VLAN pro provoz síťové správy.
- Přepínač L2 musí mít všechny deaktivované porty přiřazeny k nepoužívané VLAN.
- Přepínač L2 nesmí mít výchozí VLAN přiřazená k žádným portům orientovaných na hosta.
- Přepínač L2 musí mít všechny porty orientované na uživatele nebo nedůvěryhodné hosty nakonfigurované jako přístupové porty. [61, 62]

Doporučení pro firewally

- Firewall nesmí naslouchat službě Telnet.
- Firewall musí používat filtry, které zabraňují nebo omezují účinky všech typů běžně známých útoků typu DoS (Denial-of-Service), včetně záplavových útoků a neoprávněného skenování portů.
- Firewall musí být nakonfigurován tak, aby používal filtry, které pracují se záhlavím paketů a atributy paketů, včetně zdrojových a cílových IP adres a portů, aby se zabránilo toku neoprávněného nebo podezřelého přenosu mezi propojenými sítěmi s různými bezpečnostními politikami (včetně firewallů na perimetru a serverech v sítích VLAN).
- Firewall musí být nakonfigurován tak, aby odesílal logy provozu na centrální monitorovací server. (NetFlow)
- Firewall umístěný za hraničním směrovačem musí být nakonfigurován tak, aby blokoval veškerý odchozí provoz ze sítě (VLAN) pro správu.
- Firewall nesmí využívat žádné služby nebo funkce, které nejsou nutné pro správu brány firewall.
- Firewall musí deaktivovat nebo odstranit nadbytečné síťové služby a funkce, které se nepoužívají jako součást pro roli v dané architektuře.
- Firewall musí být nakonfigurován tak, aby zasílal výstrahu v reálném čase zodpovědným uživatelům v případě selhání.
- Hraniční firewall musí být nakonfigurován pro redundanci služeb, vyvažování zátěže nebo jiná ochranná opatření definovaná organizací, aby se omezily účinky útoků typu DoS (Denial-of-Service) na síť.
- Firewall musí generovat logy, pokud je provoz odmítnut, omezen nebo zahozen.
- Firewall musí chránit logy před neoprávněným přístupem ke čtení na místním úložišti.
- Firewall musí chránit logy před neoprávněnou úpravou.
- Firewall musí blokovat nebo omezit příchozí IP pakety určené do řídící roviny samotného firewallu.
- Firewall musí použít vstupní filtry na provoz přicházející do sítě prostřednictvím jakéhokoli aktivního externího rozhraní.
- Hraniční firewall musí filtrovat provoz určený do interních sítí v souladu se specifickým provozem, který je povolen a definován v seznamu pravidel.
- Firewall musí blokovat odchozí IP pakety, které obsahují nelegitimní atributy včetně neplatné zdrojové adresy nebo pakety, které nesplňují podmínky minimální délky (TCP, UDP, data IP) a nedefinují číslo portu.
- Brána firewall musí být nakonfigurována tak, aby umožňovala oprávněným uživatelům zaznamenávat síťový provoz spojený s relací nějakého uživatele.
- Firewall musí blokovat odchozí provoz obsahující útoky typu DoS (Denial-

of-Service), aby byla zajištěna ochrana před použitím interních informačních systémů k zahájení útoků DoS proti jiným sítím nebo koncovým bodům.
- Firewall musí omezit provoz vstupující do VPN tunelu pouze na autorizované pakety založené na cílové adrese.
- Firewall musí filtrovat provoz z bezdrátových přístupových bodů na základě definovaných pravidel organizace a monitorovat provoz samotný.
- Firewall musí filtrovat provoz přicházející z VPN tunelu na základě definovaných pravidel organizace a monitorovat provoz samotný. [63, 64]

Doporučení pro přístupové body a bezdrátové sítě

- Časový limit pro neaktivní relaci WLAN musí být nastaven na 30 minut nebo méně.
- Heslo nakonfigurované v přístupovém bodu WLAN pro generování klíčů a přístup klientů musí mít délku minimálně 14 znaků a dostatečnou složitost.
- Přístupový bod WLAN musí být nakonfigurován pro zabezpečení protokolem WPA2 (CCMP).
- WLAN musí používat protokol EAP-TLS.
- Sítě WLAN SSID musí být změněny z výchozího na pseudonáhodné slovo, které přímo neidentifikuje uživatele, organizaci apd.
- Signál WLAN nesmí být zachycen mimo povolenou oblast pro přístup k WLAN.
- Sítě WLAN musí být nepřetržitě hlídány prostřednictvím bezdrátového IDS.
- Všechna bezdrátová síťová zařízení, jako je bezdrátový systém detekce narušení (IDS) a bezdrátové směrovače, přístupové body, brány a řadiče, musí být umístěny v zabezpečené místnosti s omezeným přístupem nebo jiným zabezpečením, aby se zabránilo neoprávněné manipulaci nebo krádeži. [65, 66, 67]

Doporučení pro správu přístupu

- Implementace AAA protokolů
- Z AAA protokolů by se měli používat pouze standardizované protokoly RADIUS, TACACS+ nebo Kerberos.
- AAA servery se starají o přístup uživatelů. Na zařízeních by se neměli používat žádné lokální uživatelské účty, je povolen pouze jeden administrátorský účet pro nouzové účely
- Server AAA musí být nakonfigurován s jedinečným klíčem, který má být použit pro komunikaci (tj. RADIUS, TACACS +) s jakýmkoli klientem požadujícím autentizační služby.
- Na serveru AAA by měl být implementován HIDS
- Standard 802.1x by měl být implementováno pomocí zabezpečeného EAP, jako je EAP-TLS, EAP-TTLS nebo PEAP.

- Autorizovaným uživatelům musí být přidělena nejnižší úroveň oprávnění nezbytná k plnění přidělených povinností.

Doporučení pro log server management

Log management by měl zahrnovat:

- Centrální agregaci logů shromážděných od zařízení, které dokáží používat syslog protokol.
- Centrální agregaci logů z databází, serverů, které obvykle nepoužívají syslog protokol. (aplikační servery, OS serverů)
- Použití technik pro redukci událostí a normalizaci logů pro následnou analýzu a report.
- Definování a dokumentování rozsahu pokrytí pro každý syslog, management bezpečnostních informací a události (SIEM) a agregační servery. Dokumentování toho, které logy zařízení a hostů jsou jednotlivými servery shromažďovány.
- Definování a dokumentování požadavků na uchovávání logů z každého zařízení, a poté konfigurace centrálního log serveru tak, aby vyhovoval požadované době uchování. (data retention)
- Využitím robustních automatizačních nástrojů pro management bezpečnostních informací a událostí (SIEM) by mělo dojít k automatizaci oznámení a pravidelné analýze logů na případné ukazatele kompromitace.
- Konfiguraci funkcí na automatizováné vytváření tiketů pro jednotlivé události, aby bylo zajištěno provedení příslušných opatření a zaznamenávání provedených akcí. [68]

Doporučení pro centrální log server

- Prostředky serveru pro sběr logů musí být nakonfigurovány tak, aby používaly protokol syslog nebo jiný průmyslový standardní formát (např. protokol událostí systému Windows), který lze použít s běžnými analytickými nástroji.
- Server, který agreguje záznamy protokolu z hostitelů a zařízení, musí být nakonfigurován tak, aby používal TCP pro přenos a bylo tak zaručeno doručení.
- Server musí být nakonfigurován tak, aby vynucoval minimální délku hesla 15 znaků.
- Server musí být nakonfigurován tak, aby si zachoval identitu původního zdrojového hostitele nebo zařízení, kde k události došlo v podle logu.
- Součástí serveru by měly být také funkce pro analýzu, prohlížení a indexování.
- Pokud je v síti více log serverů, každý z nich by měl být nakonfigurován tak, aby agregoval logy do centrálního serveru.
- Server musí být nakonfigurován tak, aby generoval reporty, které nemění původní obsah ani časové řazení záznamů logu.

- Server musí být nakonfigurován tak, aby používal systémové hodiny ke generování časových razítek pro záznamy logu.
- Server musí být nakonfigurován tak, aby zálohoval úložiště logů nejméně každých 7 dní na jiný systém nebo zařízení, než je spravovaný systém nebo zařízení.
- Server musí být nakonfigurován tak, aby odeslal okamžité varování administrátorovi systému (SA), když přidělený objem úložiště logů dosáhne 75 % maximální kapacity úložiště.
- Zálohy serveru musí být uloženy na médiu schopném zaručit integritu dat po dobu nejméně 5 let.
- Po přijetí logů od hostitelů a zařízení musí být centrální logovací server nakonfigurován tak, aby zaznamenával časová razítka času přijatých logů a ty pak mapoval na koordinovaný světový čas (UTC).
- Při používání autentizace založené na PKI musí server ověřovat certifikáty.
- Pro účty využívající ověřování pomocí hesla musí server použít SHA-1 nebo novější, aby byla chráněna integrita procesu ověřování.
- Server musí být nakonfigurován tak, aby jednoznačně identifikoval a ověřoval uživatele organizace (nebo procesy jednající jménem uživatelů organizace).
- Pro účty využívající ověřování pomocí hesla musí být server nakonfigurován tak, aby se ukládala pouze kryptografická reprezentace hesla (hash).
- Server musí být nakonfigurován tak, aby chránil důvěrnost a integritu přenášených informací.
- Server musí deaktivovat účty (jednotlivci, skupiny, role a zařízením) po 35 dnech nečinnosti.
- Server musí používat více faktorové ověřování pro místní přístup k privilegovaným uživatelským účtům. Jeden z faktorů by měl být oddělen od systémů, ke kterému přístup získáváme.
- Server musí uchovávat záznamy logů na základě úrovně kritičnosti, typu události a nebo doby uchovávání. [69]

Doporučení pro linuxové servery na distribuci RHEL / CentOS

- Systém musí být nakonfigurován tak, aby uživatel root byl jediným účtem s neomezeným přístupem do systému.
- Systém musí používat samostatný souborový systém pro adresáře /var, /home, /tmp.
- Systém musí používat samostatný souborový systém pro adresář /var.
- Výchozí řetězce komunity SNMP v systému se musí změnit.
- Systém nesmí umožňovat neomezený počet pokusů o přihlášení.
- Systém nesmí umožňovat bezobslužné nebo automatické přihlášení k systému

pomocí grafického uživatelského rozhraní.

- Systémy používající Basic Input Output System (BIOS) nebo Unified Extensible Firmware Interface (UEFI) musí vyžadovat ověření po zavedení režimu pro jednoho uživatele nebo režimu údržby.
- Systém nesmí mít nainstalován balíček ypserv, rsh-server, telnet-server.
- Systém nesmí mít nainstalován balíček pro Trivial File Transfer Protocol (TFTP), pokud to není vyžadováno pro správu.
- Systém musí být nakonfigurován tak. aby vzdálená připojení pro interaktivní uživatele byla šifrována.
- Systém musí povolit SELinux a zapnout jej v módu „targeted“.
- Systém musí být nakonfigurován tak, aby oprávnění souborů, vlastnictví a členství ve skupinách systémových souborů a příkazů odpovídaly výchozím hodnotám.
- Systém musí zabránit instalaci softwaru, oprav, aktualizací Service Pack, ovladačů zařízení nebo součástí operačního systému bez ověření, že byly digitálně podepsány pomocí certifikátu vydaného certifikační autoritou (CA), která je uznána a schválena danou organizací.
- Systém musí být nakonfigurován tak, aby byla zakázána sekvence kláves Ctrl-Alt-Delete.
- Systém nesmí mít nainstalován balíček pro File Transfer Protocol (FTP), pokud to není nutné.
- Systém musí používat antivirový program.
- Systém musí používat alespoň dva DNS.
- SSH démon musí být nakonfigurován tak, aby používal pouze protokol SSHv2.
- Systém musí být nakonfigurován tak, aby démon SSH neumožňoval autentizaci pomocí prázdného hesla.
- Systém musí být nakonfigurován tak, aby démon SSH nepovolil autentizaci generického zabezpečení aplikačního programového rozhraní (GSSAPI), pokud to není nutné.
- Systém musí být nakonfigurován tak, aby soubory soukromých klíčů SSH měly souborové práva 0640 nebo striktnější.
- Systém musí být nakonfigurován tak, aby soubory veřejných klíčů SSH měly souborové práva 0644 nebo striktnější.
- Systém musí být nakonfigurován tak, aby démon SSH neumožňoval autentizaci pomocí autentizace RSA rhosts.
- Systém nesmí povolit přímé přihlášení na uživatele root pomocí SSH.
- Systém nesmí dovolit uživatelům přepsat proměnné v konfiguraci pro SSH.
- Systém musí být nakonfigurován tak, aby démon SSH nepovolil autentizaci Kerberos, pokud to není nutné.

- Systém musí být nakonfigurován tak, aby všechna síťová připojení spojená s provozem SSH byla ukončena na konci relace nebo po 10 minutách nečinnosti.
- Systém musí být nakonfigurován tak, aby démon SSH neumožňoval kompresi bez úspěšné autentizaci.
- Systém musí být nakonfigurován tak, aby otisky systémových souborů a příkazů odpovídalo výchozím hodnotám.
- Systém nesmí obsahovat soubory shosts.equiv nebo .shosts.
- Systém nesmí mít uživatelské účty s prázdnými nebo žádnými hesly.
- Systém musí mít nainstalované požadované balíčky pro vícefaktorové ověřování.
- Systém musí být nakonfigurován tak, aby logování vytvářelo záznamy obsahující informace o typu události, kde k události došlo, její zdroj a výsledek. Tyto auditní záznamy musí také identifikovat jednotlivé uživatele skupinových účtů.
- Systém musí logovat všechna použití příkazů sudo, crontab, chown, chownat, fchown, lchown, chmod, fchmod, passwd, setseboo, ltruncate, ftruncate, chcon, lsetxattr, removexattr, setxattr, fsetxattr, fremovexattr, kmod, rename, newgrp nebo ssh-keysign.
- Systém nesmí mít nepoužívané nebo nadbytečné uživatelské účty.
- Systém musí být nakonfigurován tak, aby všechny soubory a adresáře měly platného vlastníka.
- Systém musí být nakonfigurován tak, aby síťový systém souborů (NFS) povoloval použití RPCSEC_GSS.
- Systém musí být nakonfigurován tak, aby udělil nebo zakázal systémový přístup konkrétním hostitelům a službám.
- Systém musí implementovat vícefaktorové ověřování pro přístup k privilegovaným účtům prostřednictvím zásuvných autentizačních modulů (PAM).
- Systém musí být nakonfigurován tak, aby stávající hesla byla omezena životností maximálně 60 dní.
- Systém musí být nakonfigurován tak, aby všichni místní interaktivní uživatelé měli v souboru /etc/passwd přiřazen domovský adresář.
- Systém musí být nakonfigurován tak, aby všechny místní interaktivní uživatelské domovské adresáře měly režim 0750 nebo méně přípustný.
- Systém musí být nakonfigurován tak, aby při změně hesel nebo zavedení nových hesel bylo nutné použít pwquality.
- Systém nesmí provádět přeposílání paketů, pokud není systémem router.
- Systém musí být nakonfigurován tak, aby byly deaktivovány všechny bezdrátové síťové adaptéry.
- Systém musí být nakonfigurován tak, že pokud je vyžadován server Trivial

File Transfer Protocol (TFTP), je démon TFTP nakonfigurován pro práci v zabezpečeném režimu.
- Systém musí být nakonfigurován tak, aby zakázal velkokapacitní paměť USB.
- Systém musí být nakonfigurován tak, aby používal soubor shadow k ukládání pouze šifrovaných reprezentací hesel.
- Systém musí deaktivovat automounter systému souborů, pokud to není vyžadováno.
- Systém musí deaktivovat výpisy Kernel core, pokud to není nutné.
- Systém musí generovat logy pro všechna vytvoření účtu, úpravy, deaktivace a ukončení událostí, které ovlivňují soubor /etc/passwd.
- Systém musí instalovat bezpečnostní záplaty a aktualizace systému.
- Systém musí být nakonfigurován tak, aby hesla měla minimálně 15 znaků.
- Systém nesmí odpovídat na echo s protokolem ICMP (Internet Control Message Protocol) ICMP (Internet Protocol verze 4) odeslanou na adresu původu.
- Systém musí odeslat výstup rsyslogu na server pro agregaci logů.
- Systém musí zabránit tomu, aby se soubory setuid a setgid spouštěly v souborových systémech, které jsou importovány přes Network File System (NFS).
- Systém musí zabránit spuštění binárních souborů v souborových systémech, které jsou importovány přes Network File System (NFS).
- Systém musí být nakonfigurován tak, aby všechny lokální inicializační soubory měly režim 0740 nebo striktnější.
- Systém musí logovat všechna použití souboru sudoers a všech souborů v adresáři /etc/sudoers.d/.
- Systém musí povolit aplikační firewall, pokud je k dispozici. [70]

Doporučení pro systémy detekce průniku
- IDS/IPS musí monitorovat veškerý provoz v síti
- IDS/IPS systémy musí implementovat detekci únosů TCP spojení spolu s výchozími pravidly pro detekci.
- IDS/IPS systémy použité k ochraně serverů nebo DMZ musí poskytovat ochranu před útoky DoS SYN Flood vyřazením poloviny aktivních TCP relací.
- IDS/IPS systémy musí být chráněny před malware a neočekávaným provozem s příznaky TCP reset.
- Všechny systémy IDS/IPS musí být nainstalovány a zprovozněny utajeně – žádná adresa IP na rozhraní toku dat.
- IDS/IPS systémy, správa a databázové servery musí být umístěny v síti pro správu.
- IDS/IPS systémy musí mít implementovány základní pravidla a detekční signatury k ochraně sítě.

- IDS/IPS systémy musí mít být nakonfigurovány tak, aby upozorňovaly na potencionální útoky, incidenty a neautorizované síťové služby příslušné osoby.
- Přenos dat pro systémy IDS/IPS musí procházet vyhrazenou VLAN
- IDS/IPS musí chránit nebo omezit účinky známých i neznámých typů útoků odepření služby Denial of Service (DoS) pomocí behaviorální analýzy a prevence na základě rychlosti útoku.
- IDS/IPS musí nepřetržitě kontrolovat provoz na neobvyklé / neautorizované činnosti nebo události.
- v případě, že dojde k zaplnění místa IDS/IPS záznamy, musí se začít přepisovat nejstarší záznamy.
- IDS/IPS musí ověřovat integritu získaných aktualizací.
- IDP/IPS musí blokovat škodlivé ICMP pakety.
- IDS/IPS musí blokovat odchozí zprávy ICMP Destination Unreachable, Redirect a Address Mask.
- IDS/IPS musí provádět monitorování souborů z externích zdrojů na vstupních a výstupních bodech sítě.
- IDS/IPS musí nepřetržitě monitorovat příchozí provoz na neobvyklé / neautorizované činnosti nebo události.
- IDS/IPS musí blokovat odchozí provoz obsahující známé a neznámé útoky na odepření služby (DoS)
- IDS/IPS musí být schopné detekovat útoky na injekci kódu a SQL injekci.
- IDS/IPS musí na odchozí provoz aplikovat odpovídající bezpečnostní zásady, signatury, pravidla a techniky pro detekci anomálií.
- IDS/IPS musí přeposílat logy na centralizovaný log server.
- IDS/IPS musí fragmentované pakety blokovat nebo je před přeposlání sestavit a zkontrolovat. [71, 72]

3.3 Bezpečnostní politika

Bezpečnostní politika je správně definovaný soubor pravidel, které pomáhají chránit bezpečnost dat a systémů. Základ bezpečnostní politiky by měl být postavený na triádě CIA. Bezpečnostní politika musí odpovědět na následující otázky:

- Co je potřeba chránit?
- Proč je to potřeba chránit?
- Jakým způsobem se to má chránit?
- Jak lze ověřit, že ochrana funguje?
- Co se bude dít, pokud dojde k narušení ochrany? [74]

Uživatelé si musí být vědomi své zodpovědnosti a musí nést následky při porušení takové politiky. Pro jistotu je vhodné, aby politika zahrnovala jednotlivé pracovní pozice a jejich odpovědnost. Bezpečnostní politika obvykle zahrnuje následující dokumenty:

- **Politika** – stanovuje očekávání na vysoké úrovni, slouží k vedení rozhodnutí a dosahování výsledků.
- **Postupy** – dokument, který obsahuje procedurální kroky, jak se něco musí dělat.
- **Normy** – stanovuje požadavky, které je potřeba dodržovat.
- **Směrnice** – popisují doporučený postup.
- **Osvědčené postupy** – postup, který se má implementovat. Můžou být stanovené pro jednotlivé role. Např. webové servery by měly mít osvědčené postupy pro nejlepší zabezpečení přímo od vývojářů před nasazením do ostrého provozu.

Všechny uvedené body by měly být synchronizovány a spravovány. Nejprve je potřeba vytvořit celý bezpečnostní program pro celou organizaci. Cíle pro kontrolu zabezpečení organizace navrhuje publikace NIST 800-53[1].

Nejdůležitějších bodem je vzdělávání koncových uživatelů, protože neznalý uživatel může způsobit obrovské škody. Problém je v tom, že mnoho uživatelů může pro přístup používat vlastní zařízení (BYOD). Svou nezodpovědnou aktivitou jsou pak snadným cílem hackerů, kteří to pak nemají daleko k údajům o společnosti.

Dalším důležitým bodem je vynucení dané politiky, které probíhá za použití různých technologií a prostředků dle možností společnosti. Ideální je, když máme dokumentaci sítě a víme jaký je proud dat v síti. Kdo by měl mít kam přístup a jaké jsou vstupní body. Řada organizací při vynucování bezpečnostních politik selže, jelikož je vynucují pouze na serverech a koncových zařízení. Je důležité zohledňovat také další zařízení v síti.

Za vhodné považuji zmínit normu ISO 27001, která funguje jako hlavní norma pro Systém řízení bezpečnosti informací (ISMS). Obecně je vhodná pro organizace, které pracují s informacemi. Zaručuje soulad s aktuálními legislativními požadavky. Norma pracuje s modelem Plánuj-Dělej-Kontroluj-Jednej (Plan-Do-Check-Act – PDCA). Součástí fáze plánování je také vytvoření bezpečnostní politiky. [76] Jako základ pro tento standard funguje norma ISO 27000, která definuje pojmy a terminologický slovník pro všechny normy z dané kategorie. [75]

[1] `https://nvd.nist.gov/800-53`

4 Návrh zabezpečené sítě

V návrhu je popsána malá počítačová síť, její architektura a správa. Je kladen důraz na použití open-source softwaru. Mezi hlavní cíle návrhu patří snaha centralizovat a automatizovat správu služeb v rámci počítačové sítě. Pro zajištění vyšší míry bezpečnosti se návrh řídí vytvořenou metodikou.

4.1 Popis a požadavky sítě

Návrh sítě je vytvořen pro začínající IT firmu s velkými požadavky na automatizaci a bezpečnost. Firma má v tuto chvíli pouze 5 zaměstnanců, ale plánuje se rozrůst. Uplatňuje se zde politika BYOD, takže uživatelé jsou odkázáni na vlastní zařízení. Převážná část zaměstnanců pracuje na některé z distribucí systému Linux, pak MacOS nebo Windows. Zavádění doménového řadiče v takovém případě postrádá smysl, takže prostředí musí zůstat univerzální a současně dobře zabezpečené.

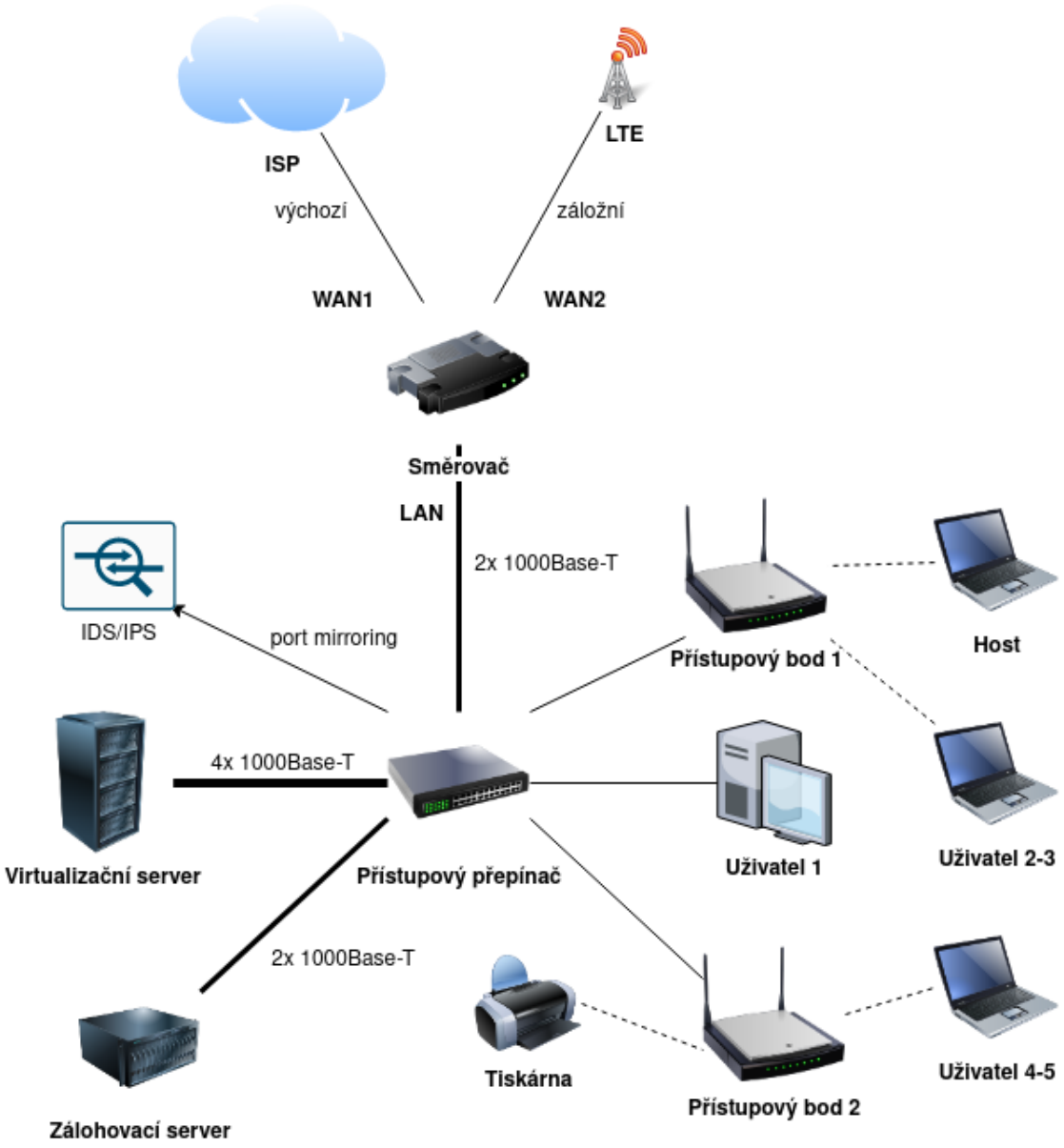
Zaměstnanci pracují převážně z prostor kanceláře, ale může se stát, že budou potřebovat vzdálený přístup, takže VPN je nutnost. V rámci kanceláře je potřeba bezdrátové připojení jak pro zaměstnance, tak i pro klienty. K dispozici by mělo být i standardní pevné připojení. Hlavní preference mají lokálně provozované služby, kromě elektronické pošty, která je již zajištěna prostřednictvím externího poskytovatele. Firma provozuje webové stránky prostřednictvím soukromé VPS, ale ty mohou být přesunuty do DMZ firemní sítě. Webové stránky nejsou nijak propojeny se systémy uvnitř firmy a obsahují pouze kontaktní údaje a nabízené služby. Všechny objednávky jsou řešeny e-mailem, ale firma by uvítala tiketovací systém. Vzhledem k počtu zaměstnanců není pro tuto chvíli nutné řešit tiskový server nebo DNS, ale bude vhodné jejich přítomnost do budoucna zvážit. K dispozici je jedna tiskárna konfigurovatelná na bezdrátovou síť, která by měla být přístupná pouze zaměstnancům. Stejně tak je potřeba zřídit místní úložiště pro efektivní spolupráci na projektech a sdílení dat mezi zaměstnanci. Vyžaduje se, aby v rámci místního úložiště bylo možné delegovat přístupy k datům pro jednotlivé zaměstnance. Velkým přínosem by byl také server pro správu hesel, aby se zamezilo používání přihlašovacích údajů se slabým heslem nebo vzájemnému sdílení hesel mezi zaměstnanci prostřednictvím nevhodných komunikačních kanálů. Posledním požadavkem je přítomnost služby pro vytváření přehledné dokumentace.

4.2 Architektura

Pro tento návrh se použil 2-vrstvý hierarchický model sítě, který má jednu kolabovanou páteřní vrstvu a jednu přístupovou vrstvu. Firma má v tuto chvíli pouze 5

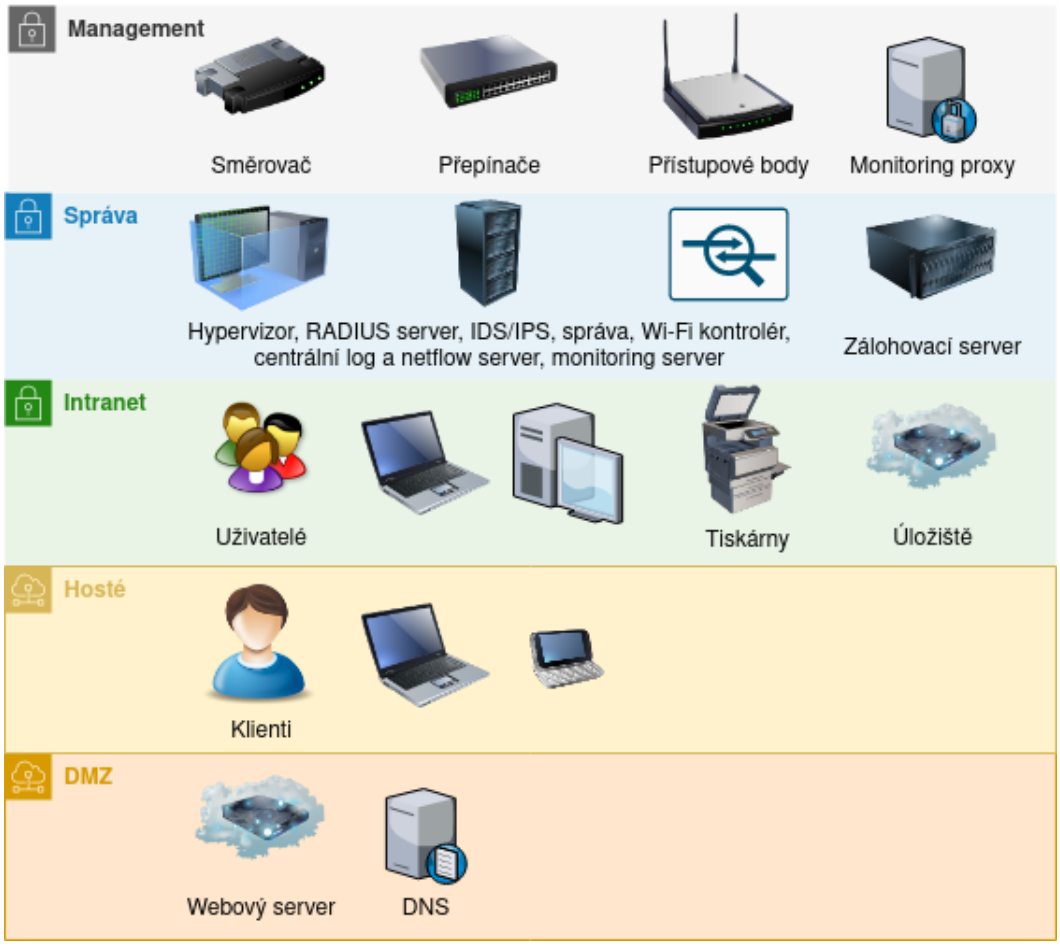
zaměstnanců, ale je vhodné síť předimenzovat kvůli případné expanzi. Centrálním a současně hraničním prvkem je v síti směrovač, který zastává hned několik funkcí – brána, firewall a VPN server. V rámci konektivity od ISP jsou k dispozici 3 veřejné IP adresy. Záložní konektivita může být zajištěna mobilní sítí LTE. V sítí se nachází tři přístupové přepínače, které vzájemně propojují všechny prvky a zařízení v síti. Pro zvýšení přenosové kapacity se mezi některými zařízeními používá agregace linek[1], což umožňuje automatické přepnutí na redundantní nebo záložní linku v případě, že dojde k výpadku aktivní linky (failover). Přenosové kapacity portů na všech směrovačích a přepínačích splňují Gigabit Ethernet známý jako 1000BASE-T. Propojení všech zařízení a prvků zajišťuje kabeláž FTP CAT6. Infrastruktura je vybavena dvěma fyzickými servery a jedním záložním. Výkonnější server pracuje jako hypervizor na linuxové distribuci CentOS/RHEL 7. Virtualizace je postavena na KVM[2], který k emulaci hardwaru využívá QEMU[3]. Většina služeb sítě je provozována na tomto serveru. Jako úložiště se používají 4 disky o velikosti 4 TB v poli RAID 10[4], čímž vytváří výslednou kapacitu úložiště 8 TB. Touto konfigurací došlo k navýšení rychlosti čtení až 4x a rychlost zápisu se zvýšila 2x. Selhat může pouze jeden z disků. Druhý server zastává funkcí zálohovací a disponuje úložištěm se 4 disky o velikosti 4 TB v poli RAID 6[5]. Tato varianta nabízí 2x vyšší rychlost čtení, rychlost zápisu není ovlivněna, ale jako nevýhoda se projevuje vyšší náročnost na výpočetní výkon. Zásadní předností je, že může dojít k selhání až 2 disků. Bezpečnost dat na úložištích je zajištěna šifrovacím algoritmem AES s 256 bitů dlouhým klíčem. Bezdrátovou síť vytváří dva přístupové body AP spravované Wi-Fi kontrolérem podporující funkci roaming[6]. Ukázku topologie sítě je možné vidět na obrázku 4.1.

[1] kombinuje dvě či více fyzických Ethernetových linek do jedné logické
[2] virtualizační řešení systému Linux pro procesory vybavené technologií Intel VT nebo AMD-V
[3] open-source emulátor procesoru pro různé architektury
[4] kombinace RAID 1 a 0, má dobrý poměr mezi odolností a výkonem
[5] obdoba pole RAID 5 navýšená o jeden paritní blok
[6] funkce pro automatické předávání klientských stanic mezi přístupovými body



Obr. 4.1: Ukázka topologie sítě.

Na základě vzniklé metodiky a modelu vrstev prostředí PAM byl vytvořen koncept vrstev sítě, viz obrázek 4.2. Tento koncept výrazně pomůže při segmentaci sítě a konfiguraci seznamu pro řízení přístupu ACL mezi jednotlivými vrstvami.



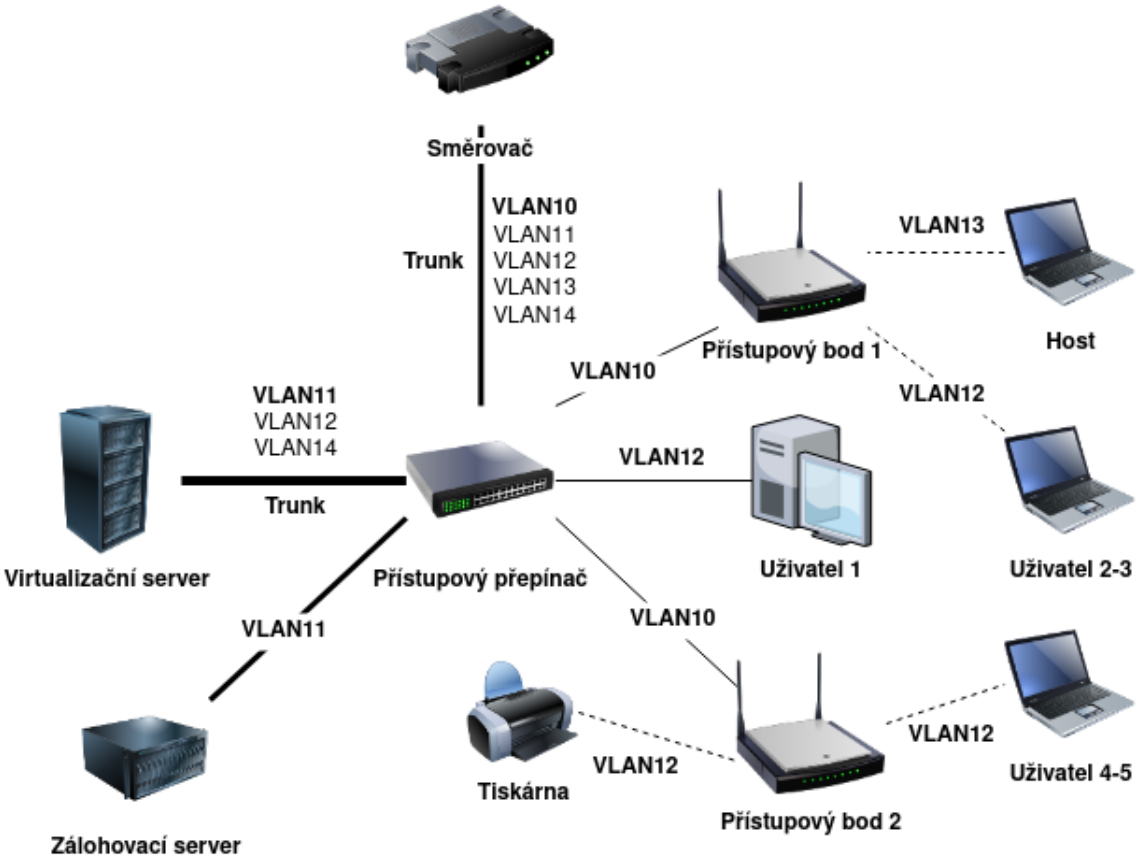
Obr. 4.2: Ukázka vrstev sítě.

Model PAM aplikovaný na síť zajistí, že zařízení ve vyšší vrstvě mají omezený přístup k zařízením v nižších vrstvách a naopak ze zařízení z nižších vrstev do vyšších nemají přístup vůbec nebo jen velmi omezeně. Rozdělení do dalších podsítí VLAN je přiblíženo v tabulce 4.1.

Tab. 4.1: Tabulka podsítí.

Název	VLAN ID	Podsíť
Management	10	192.168.140.0/24
Správa	11	192.168.141.0/24
Intranet	12	192.168.142.0/24
Hosté	13	192.168.143.0/24
DMZ	14	192.168.144.0/24

Pro lepší pochopení segmentace sítě a předávání definovaných VLAN vznikl obrázek 4.3. Zvýrazněná VLAN značí, do které podsítě se koncové zařízení na dané lince přiřadí.



Obr. 4.3: Ukázka topologie podsítí.

4.3 Služby

V následující kapitole jsou popsány vybrané služby a aplikace, ze kterých se vytvořený návrh skládá. Výběr byl závislý na možnostech integrace s dalšími systémy a dostupnosti open-source verze. V případě, že se nenalezne vhodné a efektivní řešení pro snadnou integraci, proběhne implementace jednoduchého skriptu v programovacím jazyce Bash nebo Python. Pro každou službu bude vytvořena Ansible[7] role, která umožní automatizované nasazení aplikace nebo jejich částí na libovolného hosta linuxové distribuce CentOS/RHEL 7. Automatizace dalších procesů a úloh

[7]konfigurační a orchestrační nástroj vytvářený firmou Red Hat Inc.

bude zajištěna také nástrojem Ansible pomocí tzv. playbooků, díky čemuž lze zavézt centrální správu sítě. Služby intranetu vyjma aplikace pro místního úložiště budou nasazeny pomocí prostředí Docker[8], aby se dosáhlo větší efektivity a snazší správy těchto aplikací.

Dle požadavků byl navržen následující volně dostupný software a open-source aplikace:

- **KVM/QEMU** – open-source hypervizor a emulátor poskytující hardwarovou i softwarovou virtualizaci. Sloužit bude pro virtualizaci serverů celé infrastruktury.
- **Zabbix** – open-source řešení pro monitorování sítí, serverů, služeb a aplikací. Pro návrh byla zvolena databáze PostgreSQL, která má při správné konfiguraci vyšší výkon než databáze MySQL.
- **Elasticsearch** – open-source vyhledávací a analytických nástroj vyznačující se vysokou dostupností, rychlostí a škálovatelností. Elasticsearch je bezschémová databáze, jejíž struktura se vytváří až na základě vložených dat. Účelem databáze v tomto návrhu bude ukládat provozní a lokalizační údaje, systémové logy, logy ze systémů detekce průniku a další.
- **Kibana** – open-source uživatelské rozhraní určené k vizualizaci a procházení dat z databáze Elasticsearch. Elasticsearch a Kibana jsou primárně klasifikovány jako nástroje tzv. search as a service – vyhledávání jako služba. V navržené síti umožní především pracovat s daty v databázi a definovat jejich životní cyklus. Kibana nabízí užitečný vizualizační nástroj SIEM (Security Information and Event Management), který nabízí řízení bezpečnostních operací a vyhledávání hrozeb ze získaných logů.
- **Filebeat** – volně dostupná a nenáročná aplikace pro zasílání logů z koncových stanic do databáze Elasticsearch. Obsahuje řadu užitečných modulů pro syslog, auditd, iptables, NetFlow, suricatu, webové servery NGINX nebo Apache, databáze a další. Díky čemuž se stane jednou ze základních služeb.
- **Suricata** – open-source aplikace pro rychlou a masivní detekci síťových hrozeb. Aplikace je dostupná v rámci linuxové distribuce CentOS/RHEL pod repositářem EPEL[9]. V navržené síti bude pracovat jako hlavní IDS/IPS.
- **Telegram** – open-source aplikace pro zasílání zpráv se zaměřením na rychlost a bezpečnost. Aplikaci lze používat na chytrých telefonech, tabletech a dokonce i na počítačích. Tato aplikace bude sloužit pro zasílání notifikací zodpovědným osobám o vzniklých incidentech nebo problémech v síti.

[8]standardizované zapouzdřené prostředí, které spouští aplikace v tzv. kontejnerech

[9]znamená Extra Packages for Enterprise Linux. Jde o bezplatný a open-source repositář dostupný všem

- **AlertBot**[10] – open-source nástroj pro filtrování logů aplikace Snort nebo Suricata, který následně řídí zasílání upozornění prostřednictvím různých komunikačních kanálů včetně Telegramu. Detekované události lze navíc filtrovat.
- **Arpwatch** – open-source nástroj pro monitorování ARP paketů za účelem sledování párů MAC a IP adres. Účelem aplikace je detekovat pokus o ARP spoofing nebo jakékoliv další modifikace. Aplikace je běžně dostupná v rámci výchozích repositářů linuxových distribucí. Její nasazení proběhne u kritických částí sítě. K nástroji proběhne implementace jednoduché aplikace pro filtrování a zasílání notifikací na Telegram.
- **Cowrie**[11] – open-source aplikace pracující jako SSH a Telnet honeypot[12]. Aplikace je napsána v jazyce Python. Zajišťuje včasnou detekci škodlivého softwaru nebo přítomnosti potencionálních útočníků v rámci sítě. V souvislosti s aplikací proběhne implementace nástroje pro zasílání notifikací na Telegram.
- **TelNot**[13] – open-source aplikace napsaná v jazyce Python určená pro jednoduché zasílání notifikací přes HTTP. Pomocí této varianty proběhne integrace aplikace Telegram do monitorovacího systému Zabbix. Využití však nalezne i v dalších procesech.
- **Apache** – standardně používán jako webový server. Bude zastávat funkci reverzní SSL/TLS proxy pro kontejnery Docker, které nedokáží sami zprostředkovat šifrovanou komunikaci přes HTTPS. Webový server Apache je běžně dostupný ve výchozích repositářích linuxových distribucí.
- **Docker** – open-source nástroj k snadnému vytváření a spouštění aplikací pomocí kontejnerů. Aplikace mohou být se všemi závislostmi zabaleny do jediného kontejneru. Nasazení je pak možné na kterékoliv linuxové distribuci. Pro vybrané služby budou použity definice pro více kontejnerové aplikace pomocí docker-compose.yml. Definice jsou napsány v jazyce YAML[14] a umožňují velmi snadné nasazení dané aplikace pomocí jediného příkazu. V rámci návrhu bude tato metoda preferována pro aplikace intranetu, které mohou pracovat na jednom virtuálním hostiteli. Nástroj je taktéž běžně dostupný ve výchozích linuxových distribucích. Pro instalaci nejnovější verze je nutné přidat repositář pro Docker-CE[15].
- **UniFi Controller** – volně dostupné softwarové řešení pro správu bezdrátové sítě od společnosti Ubiquiti Networks prostřednictvím webového prohlížeče.

[10] `https://github.com/nockstarr/alertBot`
[11] `https://github.com/cowrie/cowrie`
[12] aplikace simulující reálné zařízení za účelem nalákat a zaznamenat aktivitu útočníků
[13] `https://github.com/fkolacek/TelNot`
[14] formát pro serializaci strukturovaných dat
[15] `https://docs.docker.com/engine/install/centos/`

Pro správnou funkčnost jsou potřeba přístupové body značky Ubiquiti[16]. Pro nasazení aplikace se použije Docker. Je doporučeno, aby se spravované zařízení nacházely ve stejné podsíti jako kontrolér.

- **AWX** – open-source aplikace pro automatizaci pomocí orchestračního nástroje Ansible. K dispozici jsou rozsáhlé funkce a možnosti konfigurace včetně správy přihlašovacích údajů. Aplikace může být nasazena pomocí Docker. Další informace k centrální správě se nachází v části centrální správa.
- **DokuWiki** – open-source aplikace pro snadné a univerzální použití. Nepoužívá databázi, zálohování a integrace dalších funkcí je snadná.
- **Bitwarden**[17] – open-source aplikace pro jednoduché a bezpečné ukládání, sdílení a synchronizaci citlivých dat mezi jednotlivci a týmy. Nasazení aplikace bude provedeno pomocí Docker. Tato aplikace je vyžadována v rámci intranetu a bude dostupná pouze pro zaměstnance.
- **Nextcloud**[18] – open-source aplikace klient-server pro poskytování cloudového úložiště. Funkcionality jsou podobné jako v případě Office 365 nebo Google Drive. Disponuje klientskými aplikacemi pro systémy Windows, Linux, MacOS a také Android. Podporuje také WebDAV[19] protokol. Službu je možné provozovat také v kontejneru pomocí Docker. V rámci sítě zastane při nejmenším funkci místního úložiště a může být zvážen přesunu do DMZ.
- **Zammad**[20] – open-source řešení pro poskytování přehledného helpdesku a tiketovacího systému. Nabízí mnoho funkcí pro správu komunikace se zákazníky prostřednictvím několika kanálů. Aplikace má vlastní API, díky kterému je možné provádět integraci s dalšími procesy. K dispozici je také kontejner pro Docker, ten je však vhodný pouze pro testovací účely. Tato aplikace nabízí výkonné řešení na požadovaný tiketovací systém.

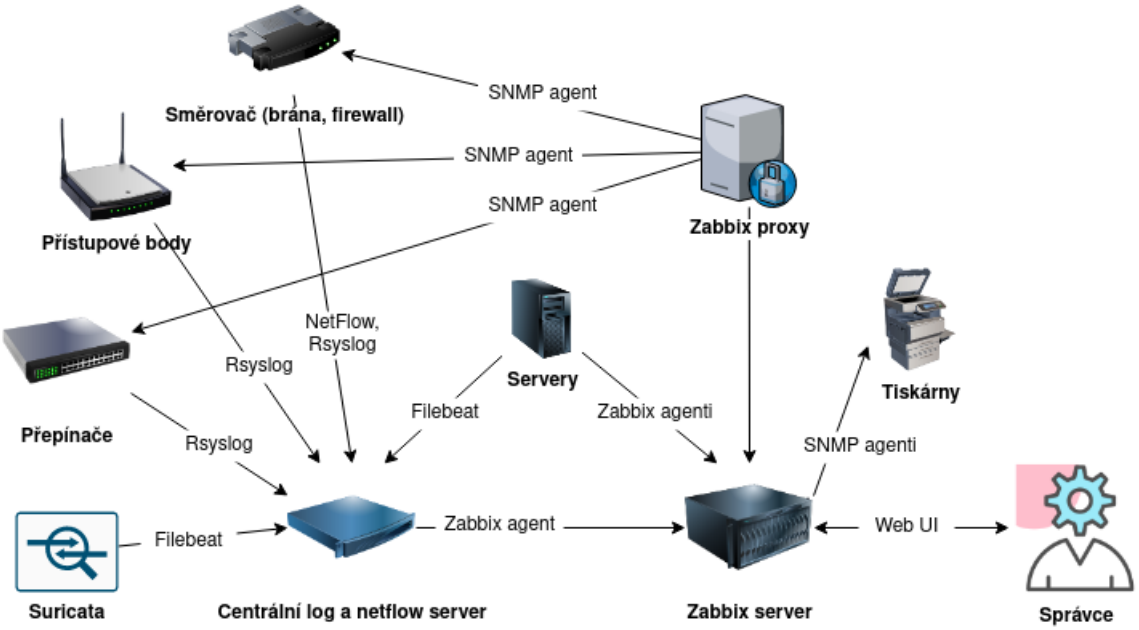
Monitoring a správa logů

Pro monitoring sítě, síťových zařízení a služeb byl vybrán Zabbix. Nabízí shromažďování dat do databáze, vytváření statistik a podporuje různé metody komunikace – SNMP, Zabbix agent, SSH, IPMI[21] nebo vlastní skripty. Umožňuje práci s různými šablonami a definovat spouštěče na konkrétní události. Dokáže přehledně vykreslovat grafy a automaticky objevovat připojené stroje z konkrétních podsítí. Velkou výhodou je rozšiřitelnost celého systému. Zabbix používá vlastní proxy, které se mohou větvit se Zabbix servery. V případě, že by se firma rozrostla o další pobočku,

[16] `https://www.ui.com/`
[17] `https://bitwarden.com/`
[18] `https://nextcloud.com/`
[19] rozšíření protokolu HTTP pro správu a operaci se soubory na vzdáleném úložišti
[20] `https://zammad.org/`
[21] rozhraní pro správu serverů

tak může využít stávající server a do nové pobočky přidat pouze Zabbix proxy. Prostřednictvím této proxy pak může monitorovat onu vzdálenou síť. Veškerá kontrola však zůstává na serveru a všechno jde snadno nakonfigurovat v přehledném webovém rozhraní. Dokumentace je přehledně zpracována. Komunikaci Zabbix serveru se Zabbix agenty nebo proxy je možné šifrovat SSL/TLS prostřednictvím vlastních certifikátů nebo před sdíleného klíče. [81]

Pro názornou ukázku byl vytvořen obrázek 4.4. Vzhledem k tomu, že Zabbix server se nachází v nižší vrstvě než aktivní síťové prvky, tak bylo nutné do sítě managementu nasadit Zabbix proxy, která je schopna aktivně komunikovat se serverem a tím zprostředkovat monitoring pro prvky z vyšších vrstev. U zařízení v nižších vrstvách jsou využívány pasivní Zabbix agenti, jelikož spojení může za těchto podmínek iniciovat pouze server. Pokud je to za daných okolností možné, upřednostňuje se aktivní režim před pasivním z důvodů nižších nároků na vytížení sítě.

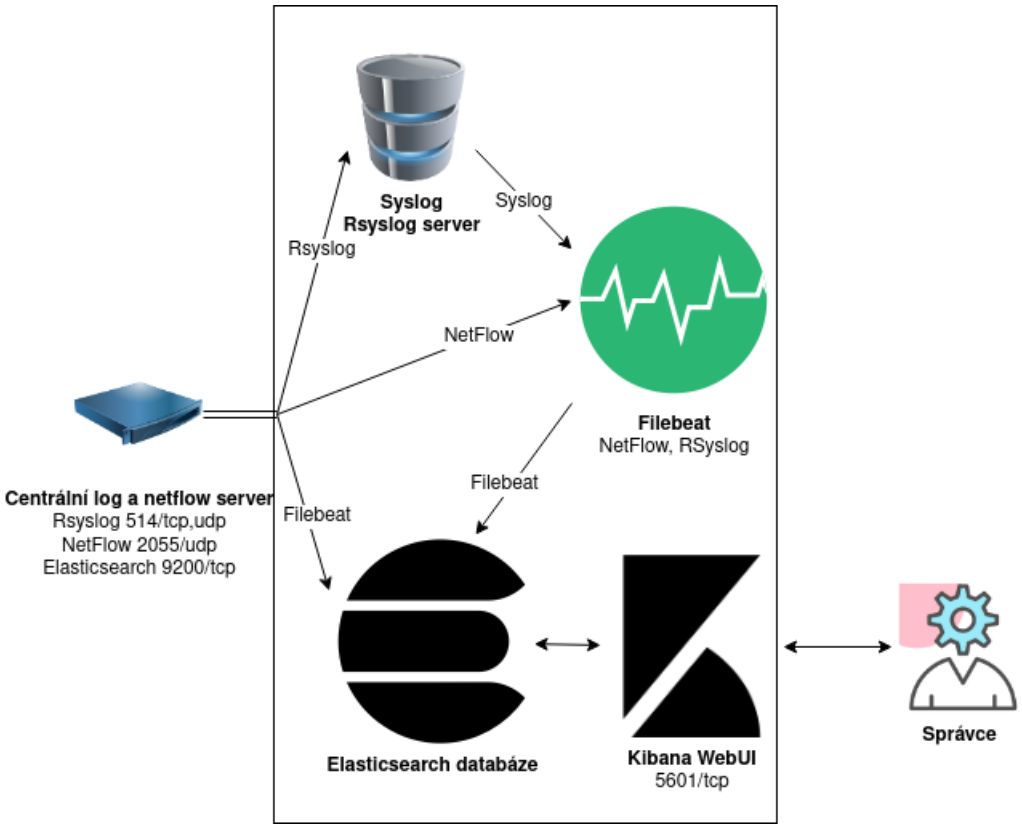


Obr. 4.4: Ukázka architektury pro monitoring sítě.

Součástí monitoringu je také centrální log a NetFlow server. Kromě sběru logů a provozních dat se tu shromažďují logy z IDS/IPS a dalších detekčních mechanismů. Centrální log server je pro většinu hostů přístupný. V případě služeb umístěných v DMZ nebo nižších úrovních sítě je potřeba provádět dotazování na logy tzv. polling. Nebo řešit agregaci logů zvlášť v rámci nižších vrstev.

Pro lepší porozumění funkce samotného serveru pro agregaci logů a NetFlow je zde obrázek 4.5. Jako úložiště používáme databázi Elasticsearch. Pro vizualizaci a práci s databází zde nachází uplatnění webová aplikace Kibana. Aplikační logy a IDS/IPS logy se sbírají pomocí modulu nástroje Filebeat, který data agreguje

a posílá přes šifrovaný kanál přímo do databáze Elasticsearch. U zařízení, na kterých není možné provést instalaci Filebeat je nutné zasílat logy na centrální syslog server, který naslouchá na portu 514 pro protokoly TCP i UDP. Pro zprovoznění rsyslogu u hostů mimo důvěryhodnou síť (např. DMZ), bude provedena konfigurace proxy na každé vrstvě pro předávání logů do centrálního log serveru.



Obr. 4.5: Ukázka architektury pro centrální log a NetFlow server.

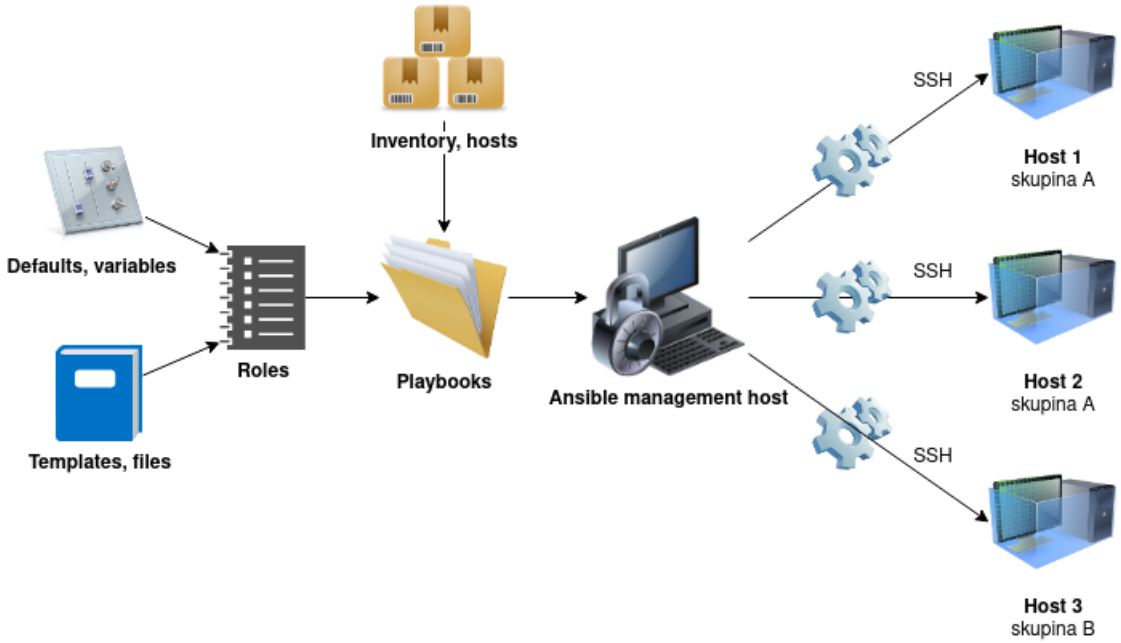
Myšlenkou této části návrhu je zavést centrální prvek pro uchování logů a provozních dat, které jsou zásadní pro behaviorální analýzu nebo zpětnou analýzu incidentů. Aplikace Kibana nabízí již zmíněnou funkci SIEM, jejímž prostřednictvím lze definovat spouštěče na konkrétní události. Variantou pro vylepšení tohoto procesu by mohla být integrace spouštěčů přímo s monitorovacím systémem Zabbix nebo vytváření tiketů v rámci helpdesku.

Centrální správa

Centrální správa má umožnit správu sítě a služeb z jednoho centrálního bodu. V případě implementovaných rolí a playbooků pro nástroj Ansible se nabízí příbuzná aplikace pracující ve webové rozhraní, a tím je projekt AWX[22]. Jedná se o neplacenou verzi produktu Ansible Tower. Nabízí vytvoření automatizačních procesů pro

[22] `https://github.com/ansible/awx`

různé akce, jejich delegaci mezi vybranou skupinou lidí a monitorovat jejich stavy a výsledky. Samotná role představuje soubor předem definovaných úkolů a proměnných, které instalují a konfigurují aplikace nebo provádí jen nějaký proces. Playbook umožňuje spuštění více takových rolí současně na skupinu hostů a přitom stále nabízí možnost úpravy definovaných parametrů pro jednotlivé role. Jednotlivé hosty a jejich skupiny mohou být definovány pomocí souboru hosts. Bližší vysvětlení přináší obrázek 4.6.



Obr. 4.6: Ukázka funkce nástroje Ansible.

V praxi to funguje tak, že veřejný SSH klíč AWX serveru se importuje pod oprávněného uživatele do systémů, které mají být pod správou. Tím dojde k získání neomezeného přístupu pro AWX server. Pro automatizační nástroj Ansible se připraví role pro jednotlivé služby nebo procesy, které se mají na hosty aplikovat. Spuštěním playbooku je možné provádět zásah do více systémů současně a tím výrazně ušetřit čas. V případě potřeby je možné provést implementaci role pro různé distribuce a systémy.

Bezpečnost

Bezpečnost návrhu je zajištěna na několika úrovních. Kromě firewallu na hraničním směrovači je zavedena patřičná segmentace sítě a včetně ACL mezi jednotlivými vrstvami sítě. V případě potřeby je možné zavést některý z aplikačních firewallů. Jako IDS/IPS byla vybrána Suricata, na kterou se zrcadlí síťový provoz. Částečnou funkci bezdrátového IDS systému zajišťuje UniFi Controller, který je schopen neob-

vyklé aktivity nebo falešné AP detekovat. Navíc nabízí DPI, což může být dalším přínosem. Monitoring ARP paketů zajišťuje nástroj Arpwatch. Pro případ, že by došlo k průniku do sítě, jsou v síti zavedeny honeypoty pro včasnou detekci průniku. Všechny události a incidenty vyhodnocené jako vážné jsou okamžitě hlášeny zodpovědným osobám prostřednictví komunikačního kanálu aplikace Telegram. Stejný systém hlášení bude integrován pro Zabbix monitoring. V síti pro zaměstnance jsou vystaveny dva webové servery, Bitwarden pro správu hesel a Nextcloud jako místní úložiště. Obě řešení podporují dvoufaktorovou autorizaci. Do DMZ se obvykle umisťují servery, které jsou přístupné zvenčí pomocí veřejné IP adresy a DNS záznamu. Pro tiketovací systém Zammad je vybrán webový server Apache, který nabízí modul mod_security. Modul je schopen poskytnout podobnou ochranu jako WAF. Tentýž modul je možné použít pro Nextcloud v případě, že se zváží jeho přesun do DMZ.

Přihlášení k síti je možné pouze pomocí uživatelského jména, hesla a certifikátu CA. Autentizace a autorizace se provádí přes RADIUS server na základě standardu IEEE 802.1X. Stejným způsobem se autentizuje vůči OpenVPN serveru. Všechny virtuální stroje a úložiště jsou zabezpečeny šifrovacím algoritmem AES s 256-bitů dlouhým . Pro přístup ke kritickým systémům se pravidelně každý měsíc mění hesla. Hesla jsou distribuovány pouze mezi administrátory a nikdo jiný k nim nemá přístup. Hesla splňují požadavky na složitost a minimální délku 15 znaků. Na všech serverech jsou automaticky instalovány důležité a kritické aktualizace systému. Přítomnost místního NTP nebo DNS bude zvážena časem. Při konfiguraci sítě a zařízení se očekává naplnění maxima možných doporučení z vytvořené metodiky.

Zálohování

Základem správného zálohování je jeho spolehlivost. Obnova ze záloh by měla trvat co nejkratší dobu, aby se minimalizovaly ztráty způsobené výpadkem. Dostupnost se obvykle vyvažuje redundancí nebo pomocí serverů v clusteru[23].

Zálohy mohou být prováděny nástrojem Rclone[24]. K nástroji je dostupná také oficiální Ansible role, kterou je možné aplikovat pro centrální správu. Aplikace je postavena na protokolu Rsync[25]. Výhodou je schopnost replikace dat na úrovni bloků. Alternativou je implementace vlastního zálohovacího skriptu pro rsync přes SSH. Pro případ, že by došlo k selhání zálohy, je možné provést integraci přes nástroj TelNot a zasílat upozornění při nezdařilé záloze do komunikačního kanálu aplikace Telegram. Provádí se rozdílové zálohy každých 12 hodin. Implementaci rolí a playbooků pro centrální správu lze provézt na takové úrovni, aby byla možná okamžitá replikace dat ze záloh na hlavní nebo záložní virtualizaci.

[23]spojení dvou a více serverů pro zvýšení jejich efektivity

[24]`https://rclone.org/`

[25]síťový protokol používaný pro synchronizování souborů a složek z jednoho umístění na druhé

5 Implementace zabezpečené sítě

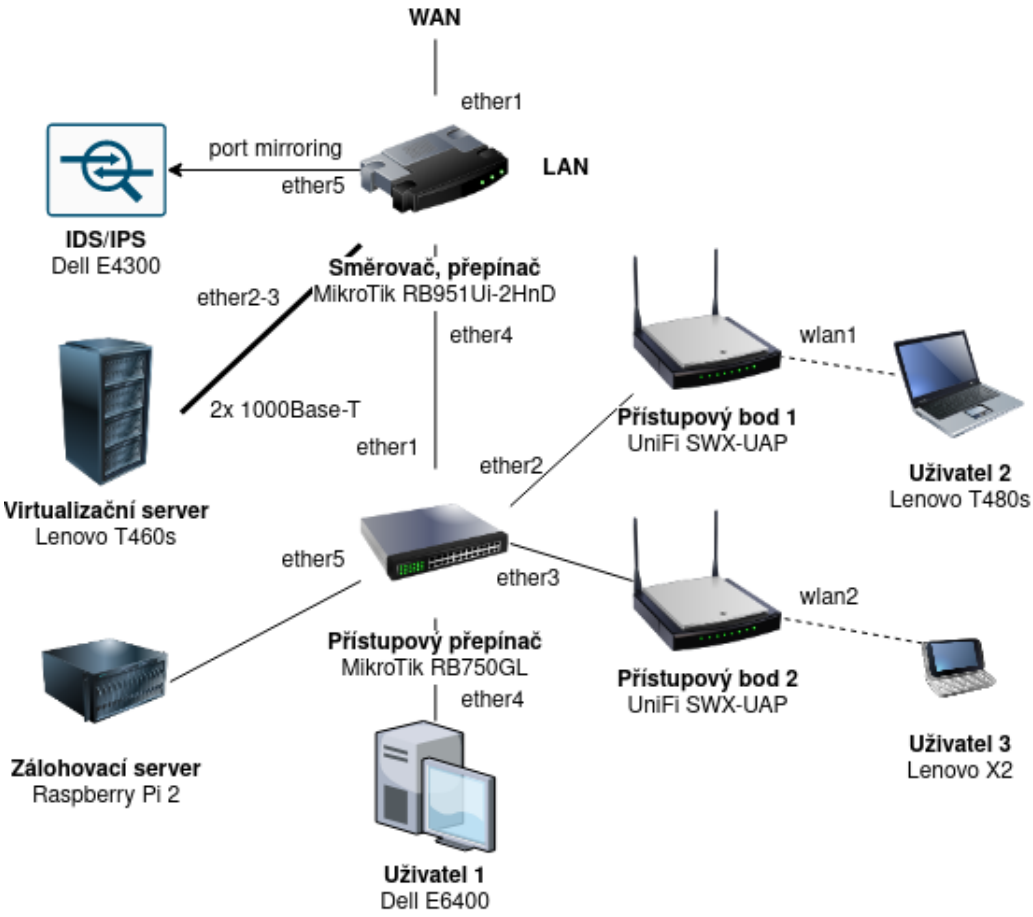
V této kapitole dochází k implementaci laboratorní sítě, která je simulací pro uvedený návrh v předchozí kapitole. Přístup k odpovídajícímu hardwaru návrhu nebyl v důsledku celosvětové pandemie způsobené Covid-19 možný, a proto proběhla implementace pouze v laboratorní podobě pro účely této práce, kterými je mimo vytvořených rolí a playbooků pro centrální správu také následné vyhodnocení detekčních mechanismů penetračním testování sítě.

5.1 Prvky sítě a jejich zapojení

Pro laboratorní síť byly použity následující prvky a zařízení:

- **MikroTik RB951Ui-2HnD** – funkce směrovače, přepínače a OpenVPN serveru.
- **MikroTik RB750GL** – funkce přístupového přepínače.
- **2x UBNT UniFi SWX-UAP, 2,4 GHz** – přístupové body pro vytvoření bezdrátových sítí.
- **Lenovo ThinkPad T460s** – hypervizor pro virtualizaci většiny služeb sítě. Určeno pro monitoring Zabbix, centrální log a NetFlow server (Elasticsearch, Kibana, Filebeat), aplikace AWX pro centrální správu a automatizaci, Zammad pro helpdesk, Nextcloud pro místní úložiště a nakonec host vyhrazený pro Docker kontejnery (TelNot, Cowrie, Bitwarden a UniFi Controller).
- **Dell Latitude E4300** – host vyhrazen pro systémy detekce průniku, zrcadlení veškerého provozu na síťové rozhraní. Určeno pro aplikace Suricata, arpwatch a alertBot.
- **Raspberry Pi 2 Model B** – pro simulaci zařízení na ukládání záloh.
- **Lenovo ThinkPad T480s** – pro simulaci uživatele systému Linux.
- **Dell Latitude E6400** – pro simulaci uživatele systému Windows
- **Lenovo Vibe X2** – mobilní telefon pro simulaci uživatele systému Android

Topologii laboratorní sítě je možné vidět na obrázku 5.1. Rozdělení do podsítí je totožné jako v návrhu.



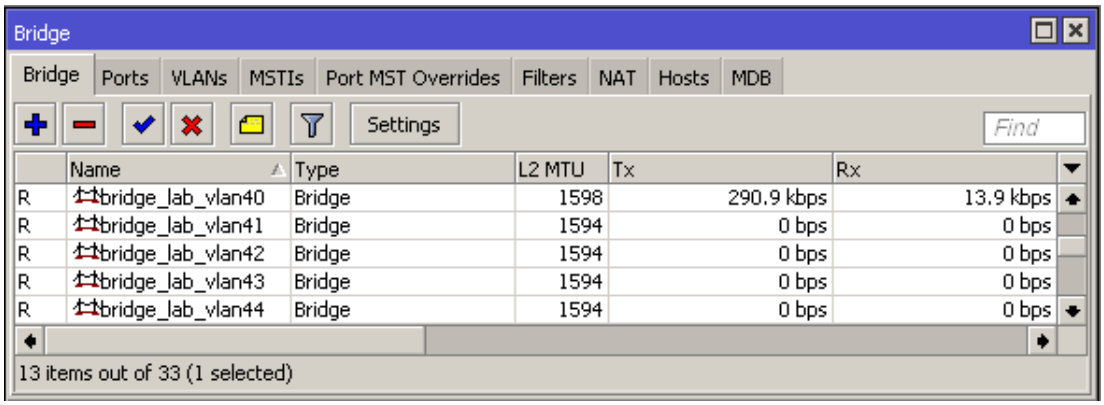
Obr. 5.1: Topologie laboratorní sítě.

5.2 Implementace sítě a služeb

Konfigurace směrovače

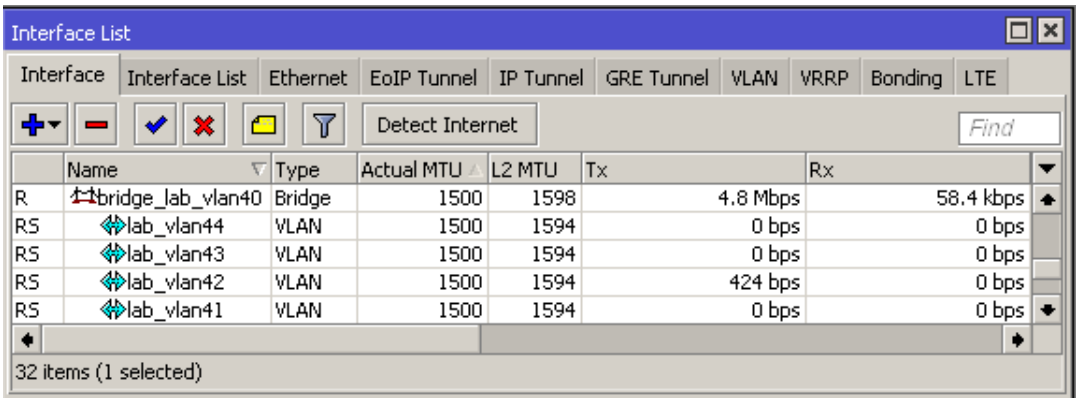
Funkci směrovače, přepínače a OpenVPN serveru v laboratorní síti zastává MikroTik RB951Ui-2HnD. Konfigurace se provádí pomocí utility Winbox.

Nejprve proběhne konfigurace rozhraní WAN přidáním DHCP klienta na rozhraní ether1 „IP –> DHCP Client –> New DHCP Client“. Tím se přiřadí IP adresa pro WAN rozhraní, dynamicky se přidělí výchozí brána a DNS. V dalším kroku je potřeba nastavit maškarádu pro celou síť přes ether1 „IP –> Firewall –> NAT –> New NAT Rule“. Nové pravidlo „Chain: srcnat, Out. Interface: ether1“. V dalším kroku se vytvoří síťové mosty pro další podsítě dle obrázku 5.2.



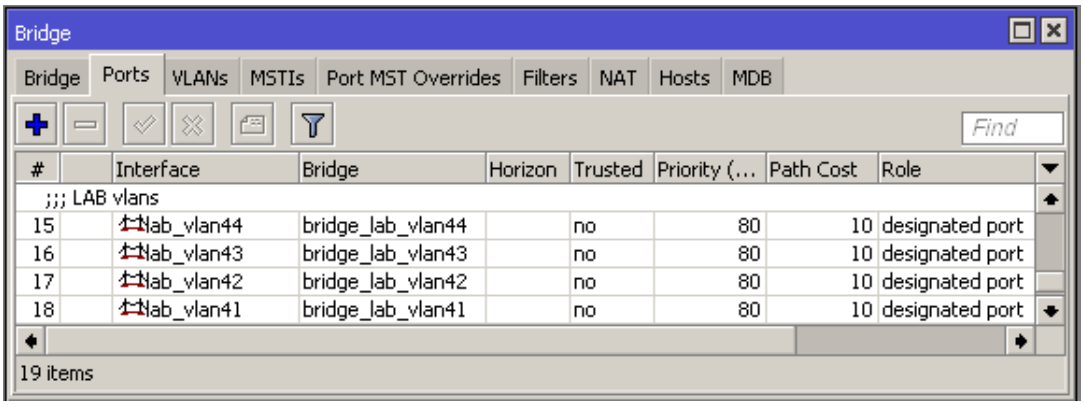
Obr. 5.2: Konfigurace síťových mostů.

Síťový most *bridge_lab_vlan40* určený pro management pracuje současně jako trunk. Na tento síťový most se přiřadí VLAN pro další podsítě „Interfaces –> New Interface –> VLAN“. Dle obrázku 5.3 se pro každou z VLAN 41 až 44 nastaví rozhraní *bridge_lab_vlan40*.



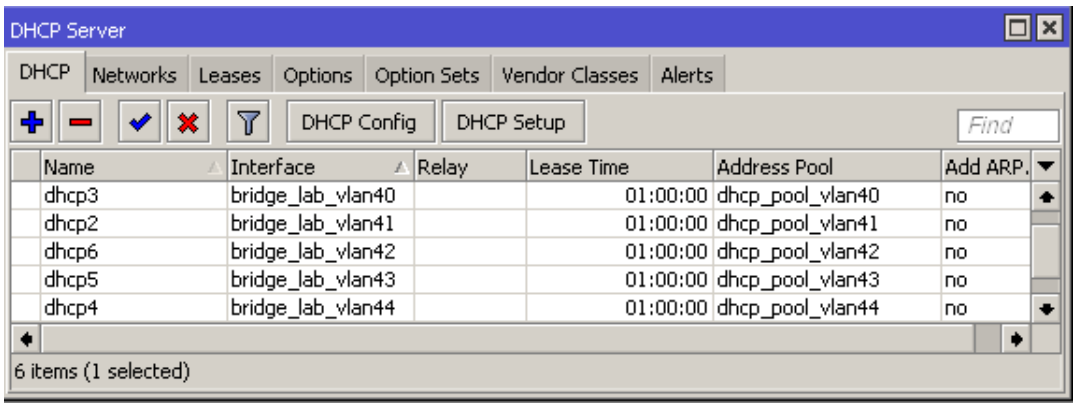
Obr. 5.3: Konfigurace VLAN.

Nyní se VLAN přiřadí k již vytvořeným síťovým mostům „Bridge –> Ports –> New Bridge Port“ podle obrázku 5.4.



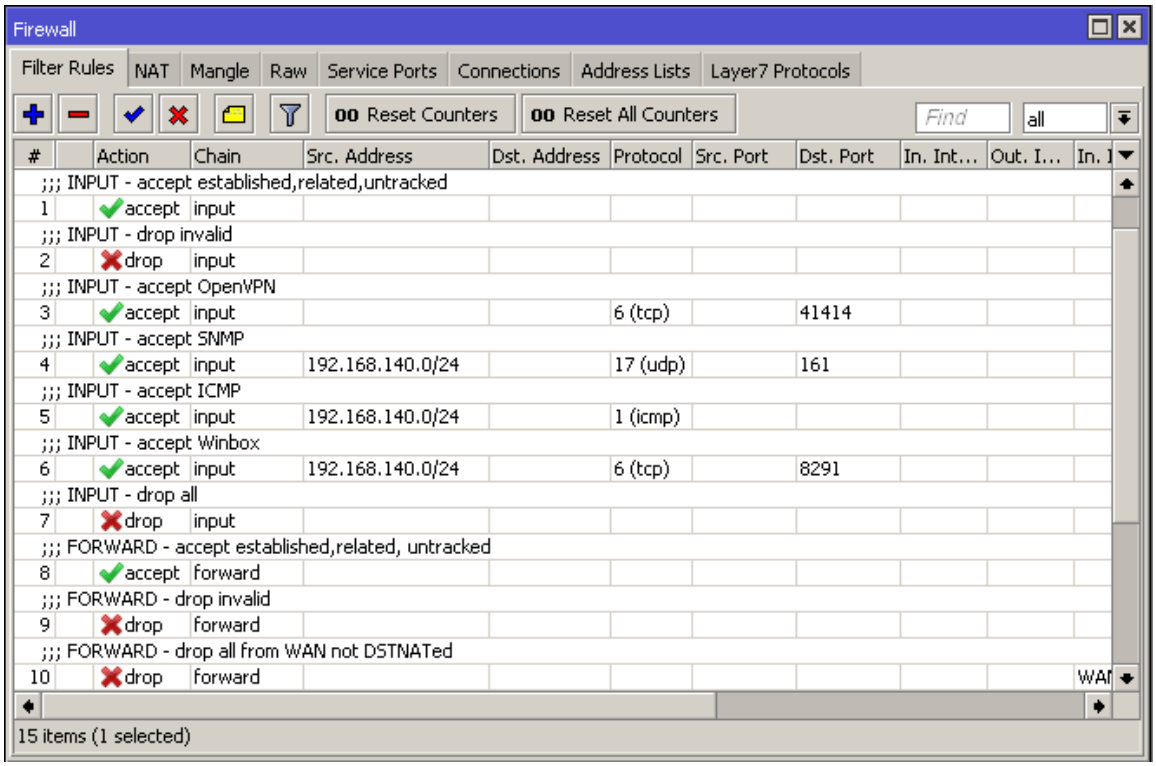
Obr. 5.4: Přiřazení VLAN k síťovým mostům.

Na základě tabulky podsítí z návrhu sítě se nakonfigurují DHCP servery „IP –> DHCP Server –> DHCP Setup“. Konfigurace se provede pro každou síť a rozhraní podobně jako na obrázku 5.5. Konfigurace DHCP je individuální a není nutné zabírat celý rozsah IP adres. Je vhodné ponechat některé počáteční adresy pro statické přiřazování IP adres.



Obr. 5.5: Konfigurace DHCP serverů.

V tuto chvíli již stačí přiřadit některý z portů *ether2* až *ether5* k jednomu z cílových sítových mostů, např. *bridge_lab_vlan42* by přiřadil IP adresu z rozsahu 192.168.142.0/24 pro intranet. Nejprve je ale vhodné provést alespoň základní konfiguraci firewallu „IP –> Firewall“. Počáteční pravidla firewallu by mohla vypadat podobně jako na obrázku 5.6.



Obr. 5.6: Ukázka počáteční konfigurace firewallu.

Konfiguraci je potřeba dále rozšířit tak, aby nebylo možné přistupovat z nižších vrstev sítě do vyšších, např. z podsítě 192.168.144.0/24 by nemělo jít přistoupit do žádné z vnitřních sítí, jelikož se jedná o DMZ. Výjimku může dostat proxy pro logy z webových serverů, která by měla vlastní pravidlo s přístupem k centrálnímu log serveru. Konfigurace firewallu bude při dokončení sítě daleko komplexnější.

Konfigurace trunk portu pro přístupový přepínač

Pokud byla úspěšně provedena konfigurace VLAN dle předchozího postupu, vytvoření trunk portu pro další přepínač už bude snadné. Všechny definované VLAN jsou svázané se síťovým mostem *bridge_lab_vlan40*, takže pro konfiguraci trunk portu stačí přiřadit dané rozhraní k síťovému mostu *bridge_lab_vlan40* „Bridge –> Ports –> New Bridge Port“. Přístupový přepínač je připojen na rozhraní *ether4*, takže konfigurace bude „Interface: ether4, Bridge: bridge_lab_vlan40“.

Konfigurace OpenVPN serveru

Implementace OpenVPN serveru u zařízení MikroTik není stále optimální a má řadu nedostatků, takže je doporučeno provést konfiguraci takového serveru na samostatném virtualizovaném systému. Pro laboratorní síť proběhne konfigurace na zařízení MikroTik. Pro funkčnost OpenVPN serveru je nezbytné vytvořit CA spolu s certifikátem pro server a klienty. Vytvoření certifikátů můžu být provedeno v terminálu.

Výpis 5.1: Generování CA pro OpenVPN

```
/certificate add name=CA country="CZ" state="Czech Republic" organization="LAB" unit="IT" common-name=lab.loc key-usage=key-cert-sign,crl-sign days-valid=3650 key-size=4096
/certificate sign CA ca-crl-host=192.168.0.151 name=CA
/certificate export-certificate CA
/certificate set CA trusted=yes
```

Z bezpečnostních důvodů byl vybrán veřejný RSA klíč v délce 4096 bitů. Parametr *ca-crl-host* by měl obsahovat veřejnou IP adresu směrovače. Pro případ laboratorní sítě se použila IP adresa 192.168.0.151. Proces podpisu může trvat o něco déle v závislosti na délce RSA klíče. V dalším kroku se vytvoří certifikát pro OpenVPN server.

Výpis 5.2: Generování certifikátu pro OpenVPN server

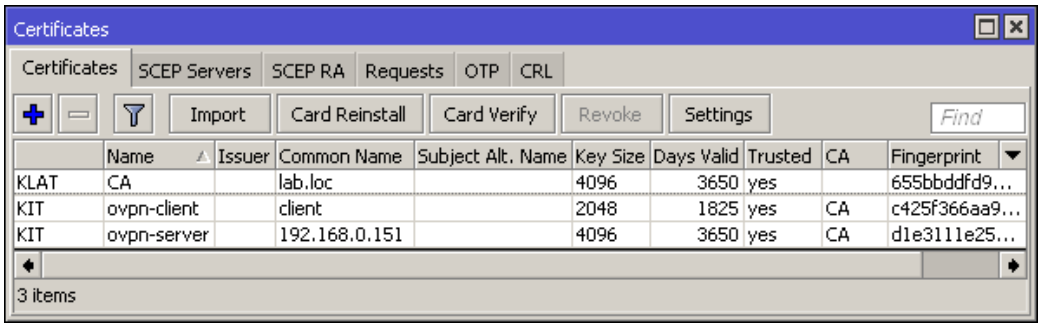
```
/certificate add name=ovpn-server country="CZ" state="Czech Republic" organization="LAB" unit="IT" common-name="192.168.0.151" key-usage=digital-signature,key-encipherment,tls-server days-valid=3650 key-size=4096
/certificate sign ovpn-server ca=CA name=ovpn-server
/certificate set ovpn-server trusted=yes
```

V dalším kroku se vytvoří certifikát pro OpenVPN klienta a provede se jeho export s heslem například *TohleJeHeslo!*. Exportované certifikáty s zašifrovaným klíčem lze stáhnout v položce *Files*.

Výpis 5.3: Generování certifikátu pro OpenVPN klienta

```
/certificate add name=ovpn-client country="CZ" state="Czech Republic" organization="LAB" unit="IT" common-name="client" key-usage=digital-signature,key-encipherment,tls-server days-valid=1825 key-size=2048
/certificate sign ovpn-client ca=CA name=ovpn-client
/certificate set ovpn-server trusted=yes
/certificate export-certificate ovpn-client export-passphrase=TohleJeHeslo!
```

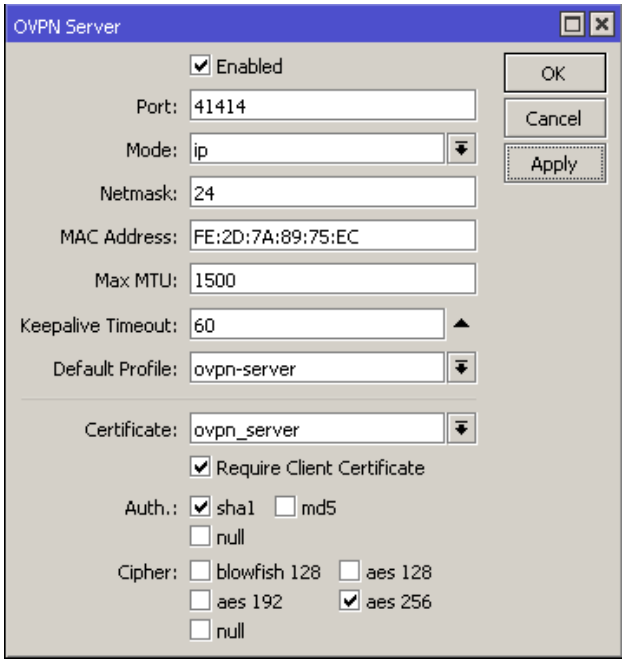
Fáze pro vytvoření certifikátů je tímto dokončena. Přehled vytvořených certifikátů je možné vidět na obrázku 5.7.



Obr. 5.7: Konfigurace certifikátů pro OpenVPN.

Pro OpenVPN klienty se vytvoří *IP Pool*, ze kterého budou dostávat IP adresy „IP Pool –> New IP Pool –> Name: ovpn-pool, Addresses: 192.168.100.2-192.168.100.10 –> OK“. Vybrán byl rozsah z nové podsítě 192.168.100.0/24.

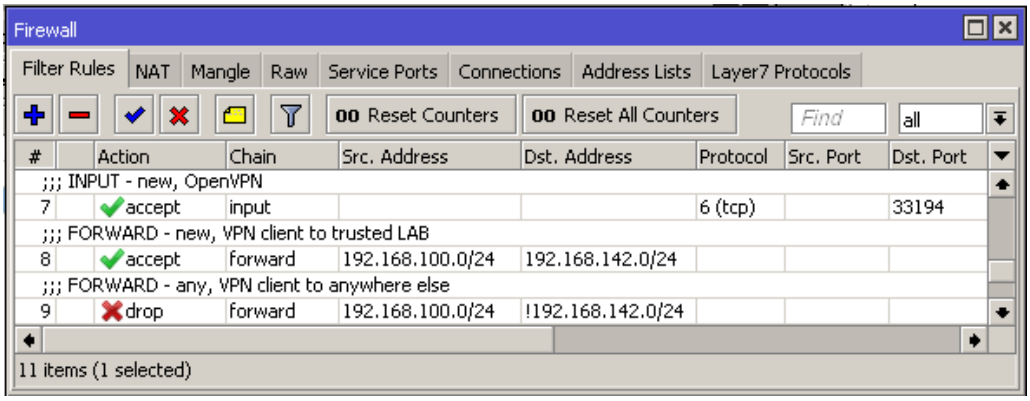
Konfigurace samotného OpenVPN serveru počíná konfigurací PPP profilu „PPP –> Profiles –> New PPP Profile“ s parametry „Name: ovpn-server, Local Address: 192.168.100.1, Remote Address: ovpn-pool“. V záložce *Protocols* se upraví následující parametry „Use MPLS: no, Use Compression: no, Use Encryption: yes“. V záložce *Interface* se otevře položka *OVPN Server*. Konfigurace je vidět na obrázku 5.8.



Obr. 5.8: Konfigurace OpenVPN serveru.

Z bezpečnostních důvodů byl vybrán TCP port 41414 namísto výchozího portu 1194. Zásadní nevýhodou OpenVPN na zařízeních MikroTik je nedostatečná bez-

pečnostní úroveň autentizačních mechanismů – MD5, SHA1, ale to by se mělo brzy zlepšit. Pro dostupnost OpenVPN serveru je potřeba přidat ještě nějaká pravidla do firewallu včetně omezení přístupu pouze do zaměstnanecké sítě VLAN42. Pravidla ke konfiguraci jsou vidět na obrázku 5.9.



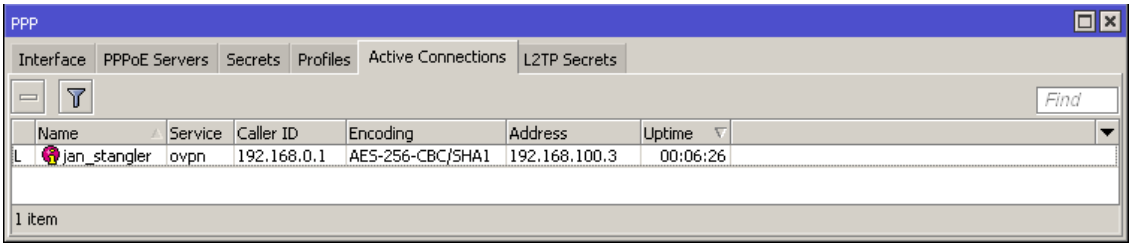
Obr. 5.9: Konfigurace firewallu pro OpenVPN.

Ještě zbývá přidat uživatele, kteří se budou moci připojit „PPP –> Secrets –> New PPP Secret“. Zde je důležité zadat následující parametry „Name: jan_stangler, Password: TohleJeHeslo!, Service: ovpn, Profile: ovpn-server“. Konfigurace pro OpenVPN klienty je k nahlédnutí ve výpise 5.4.

Výpis 5.4: Konfigurace OpenVPN klienta

```
client                                                                  1
remote 192.168.0.151 41414                                              2
proto tcp                                                               3
auth-user-pass                                                          4
cipher AES-256-CBC                                                      5
dev tun                                                                 6
auth SHA1                                                               7
resolv-retry infinite                                                   8
nobind                                                                  9
route-delay 4                                                           10
auth-nocache                                                            11
script-security 2                                                       12
persist-key                                                             13
persist-tun                                                             14
verb 3                                                                  15
reneg-sec 0                                                             16
verify-x509-name "C=CZ, ST=Czech Republic, O=LAB, OU=IT,                17
   CN=192.168.0.151"
route 192.168.142.0 255.255.255.0 192.168.100.1                         18
                                                                        19
ca '/home/jstangle/.cert/lab-CA.crt'                                    20
cert '/home/jstangle/.cert/lab-ovpn.crt'                                21
key '/home/jstangle/.cert/lab-ovpn.key'                                 22
```

Tento konfigurační soubor lze snadno importovat do téměř všech distribucí systému Linux a s programy třetích stran je funkční také na systému Windows. MikroTik běžně exportuje privátní klíč šifrovaný heslem. Dešifrování klíče je možné pomocí příkazu „openssl rsa -in ovpn-client.key -out ovpn-client.dec.key“. Úspěšný pokus o připojení k OpenVPN je vidět na obrázku 5.10.



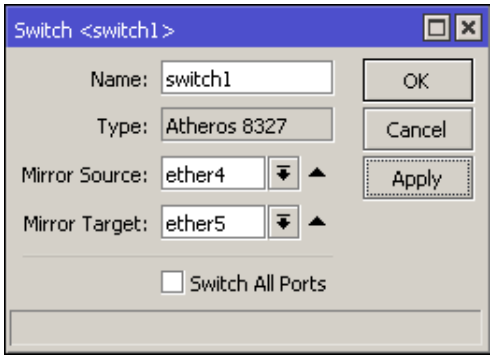
Obr. 5.10: Připojení klienta k OpenVPN.

Konfigurace DHCP pro detekci AP v kontroléru

Přístupové body se připojují do sítě spravované Wi-Fi kontolérem, v tomto případě k UniFi Controlleru pracujícím v Docker kontejneru na adrese 192.168.141.9. Aby byla zaručena detekce přístupových bodů kontrolérem, je potřeba provést některá vylepšení DHCP serveru. Konfigurace je možná v těchto krocích „IP –> DHCP Server“. Aby bylo možné přidat *DHCP option*, nejprve se musí převést adresa kontroléru 192.168.141.9 na hexadecimální hodnotu, tj. 0xC0A88D09: „Option –> New DHCP Option –> Name: unifi, Code: 43, Value: 0xC0A88D09 –> OK“. V záložce *Networks* se upraví nastavení pro podsíť 192.168.140.0/24, což je management pro laboratorní síť, ve které jsou síťové prvky včetně přístupových bodů. V položce *DHCP options* zvolíme vytvořenou volbu *unifi*. Díky této konfiguraci by se přístupové body měly automaticky hlásit Wi-Fi kontroléru po obdržení IP adresy od DHCP serveru. V případě, že by tato konfigurace nefungovala správně, je možné se přihlásit k přístupovým bodům pomocí SSH a informovat kontrolér o přístupovém bodu příkazem „set-inform http://192.168.141.9:8080/inform“.

Konfigurace zrcadlení provozu pro IDS/IPS

Vzhledem k nedostatku síťových zařízení není možné vytvořit pouze jeden trunk port ze směrovače, protože by nebylo možné přivést trunk port na požadovaný počet zařízení. Provoz, který se bude zrcadlit na IDS není kompletní a bude souviset pouze se zařízeními připojenými na přístupovém přepínači. MikroTik nabízí rozšířenou funkcionalitu *switch chiping* umožňující port switching, port mirroring a další. Pro Mikrotik se nastaví tzv. *port mirroring* mezi rozhraním ether4 a ether5: „Switch –> switch1 –> Mirror Source: ether4, Mirror Target: ether5“. Na rozhraní ether4 je trunk port, který vede k přístupovému přepínači a na rozhraní ether5 je připojeno zařízení pro IDS – Suricata. Ta slouží jako hlavní detekční mechanismus podezřelých aktivit v síti. Na obrázku 5.11 je vidět konfigurace pro zrcadlení portů.



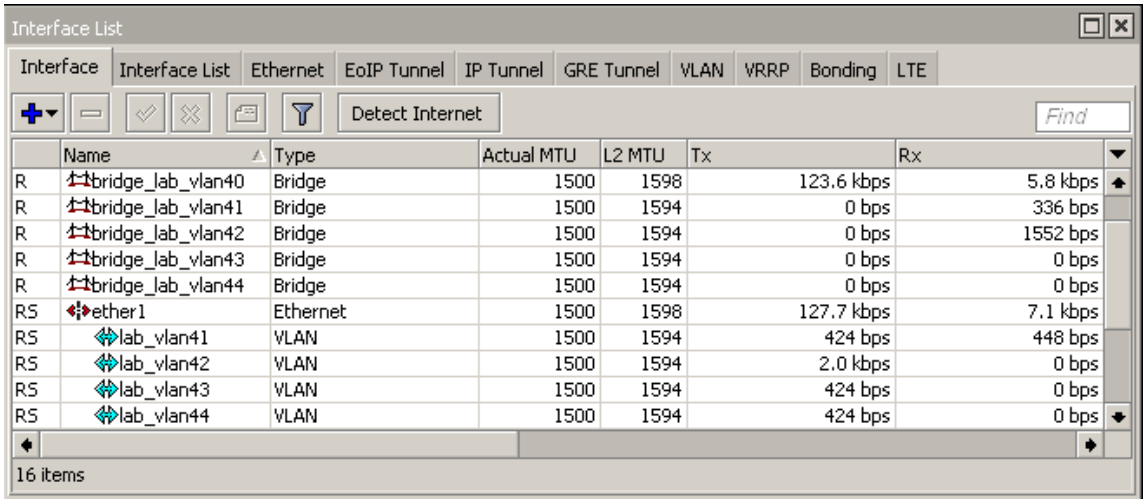
Obr. 5.11: Konfigurace port mirroringu pro MikroTik.

Agregace linky pro hlavní hypervizor

Zvýšení propustnosti je možné docílit agregací linek. Hlavní hypervizor je připojen přes dvě rozhraní splňující 1000BASE-T k zařízení MikroTik RB951Ui-2HnD. Konfigurace se provede následovně: „Interfaces –> New Interface –> Bonding“. V záložce *Bonding* je potřeba definovat tzv. *slaves*, kterými je *ether2* a *ether3* – Mode: 802.3ad, MII Interval: 100 ms. Jako mód agregace je preferován standard IEEE 802.3ad s protokolem LACP, ovšem záleží na podpoře u daných zařízení. MikroTik a linuxové distribuce tento standard podporují.

Konfigurace přístupového přepínače

Přístupový přepínač v síti zastupuje zařízení MikroTik RB750GL. Na rozhraní *ether1* je potřeba pro příchozí trunk nastavit VLANy dle obrázku 5.12. Zrovna se přidají vyhrazené síťové mosty pro příslušné VLAN. Dokončení konfigurace probíhá přes „Bridge –> Ports –> New Bridge Port“. Zde je potřeba opět propojit vytvořená VLAN rozhraní s danými síťovými mosty. Spojením rozhraní k libovolnému síťovém VLAN mostu dojde k přiřazení IP adresy z daného rozsahu. Ověření funkčnosti je možné přidáním DHCP klienta na rozhraní síťového mostu.

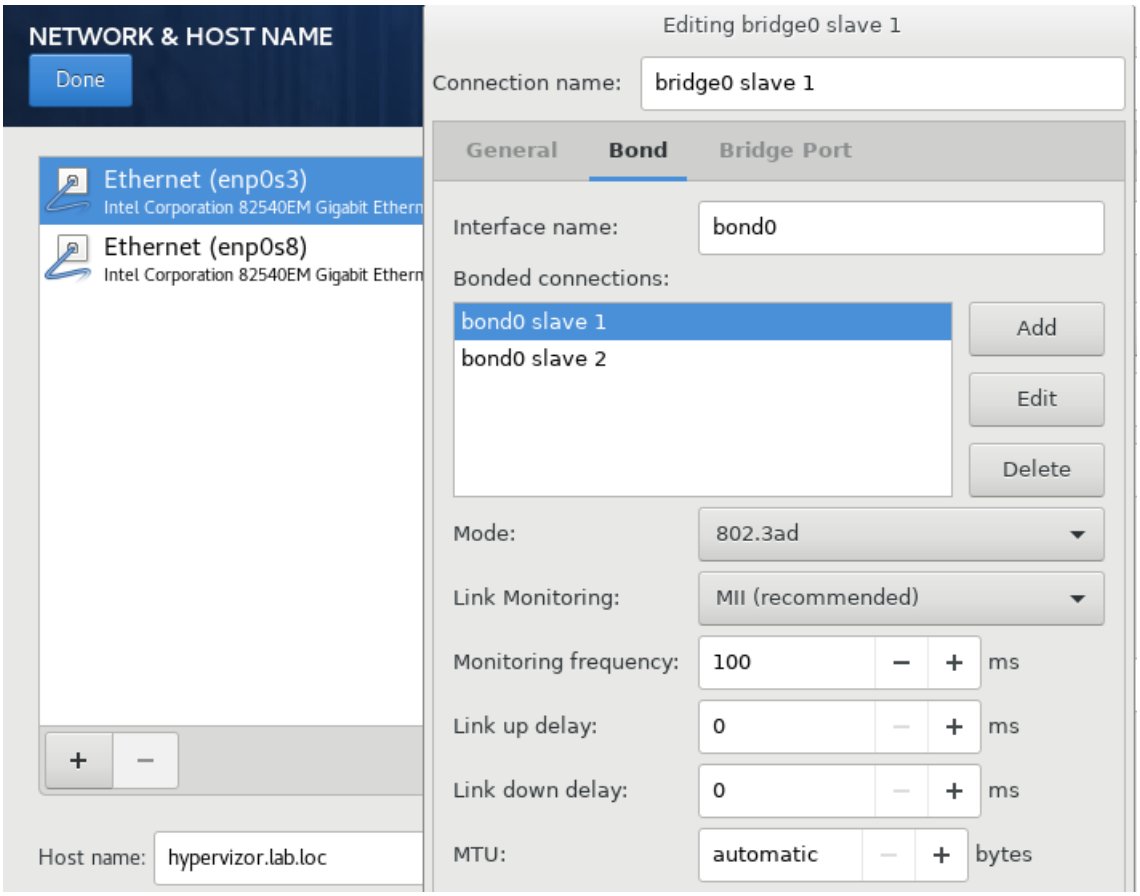


Obr. 5.12: Konfigurace přístupového přepínače.

Instalace a konfigurace systému pro hypervizor

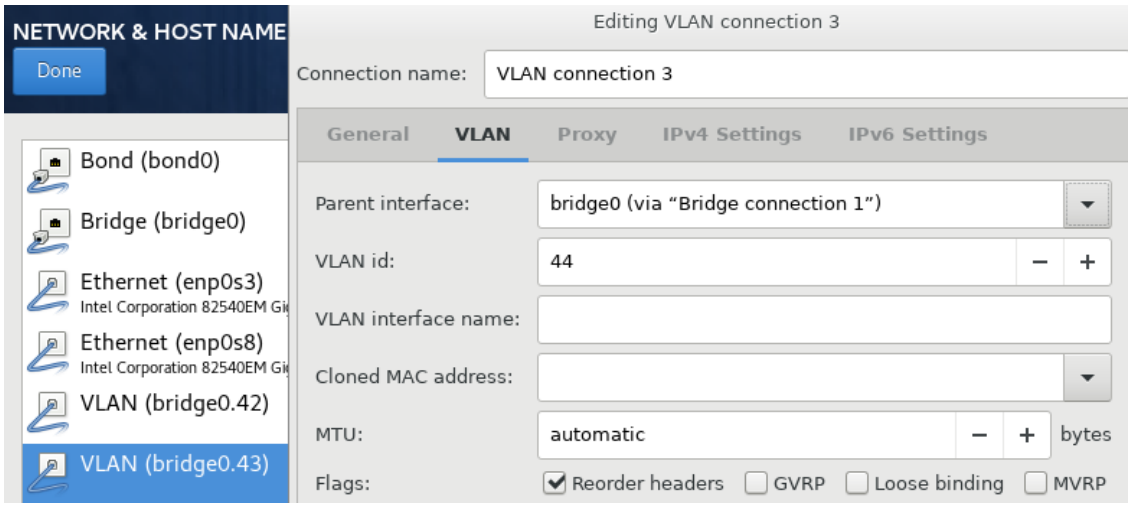
V tuto chvíli se lze přesunout k přípravě hypervizoru na notebooku Lenovo ThinkPad T460s. Nejprve je potřeba stáhnout instalační minimální ISO pro CentOS 7 např. ze stránek `http://merlin.fit.vutbr.cz/mirrors/centos/7.8.2003/isos/x86_64/` a připravit instalační médium. Po zahájení instalace je potřeba nastavit úložiště a síťové rozhraní. Úložiště se konfiguruje v sekci „SYSTEM – INSTALLATION DESTINATION“. Dále se provede „I will configure partioning + Encrypt my data –> Done –> Click here to create them automatically“. V případě potřeby je možné

provést úpravu velikosti jednotlivých diskových oddílů. V dalším kroku proběhne nastavení šifrovacího klíče „Done –> Passphrase: TohleJeHeslo!, Confirm: TohleJeHeslo! –> Save Passphrase“. Konfigurace síťového rozhraní je o něco složitější, jelikož je nejprve potřeba vytvořit síťový most a teprve pak řešit agregaci linek – „NETWORK & HOST NAME –> Add device –> Bridge –> Bridged connections –> Add –> Bond“. Pomocí tlačítka *Add* je potřeba přidat *slave* rozhraní pro agregaci linky dle obrázku 5.13.



Obr. 5.13: Konfigurace agregace linky na síťovém mostu.

Na rozhraní *bridge0* by se měla přiřadit IP adresy z rozsahu 192.168.141.0/24. Vzhledem k přítomnosti trunk portu na tomto zařízení, přidají se také VLAN pro jednotlivé podsítě. Rodičovským rozhraním bude v tomto případě vytvořený *bridge0*. Ukázku konfigurace je možné vidět na obrázku 5.14.



Obr. 5.14: Konfigurace síťového rozhraní pro hypervizor.

Jelikož není žádoucí vystavovat hypervizor na všech sítích, je vhodné deaktivovat nastavení IPv4 na *disabled* a IPv6 na *ignore* v příslušných záložkách. V poslední fázi je potřeba nastavit heslo pro uživatele root a instalace může být dokončena.

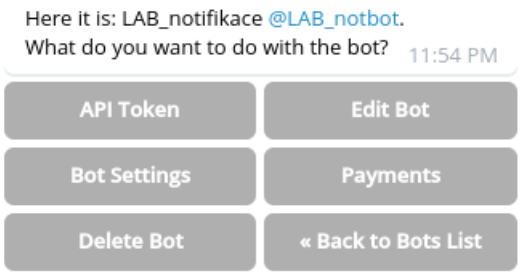
Vytvoření Telegram bota pro zasílání notifikací

Předpokladem pro vytvoření bota je uživatelský účet v aplikaci Telegram. Pomocí něj je možné zasílat libovolné notifikace. Pro instalaci a zprovoznění je potřeba telefonní číslo. Kromě mobilní aplikace je k dispozici také desktopová na Linux nebo Windows. Pro práci s vlastními boty se musí vyhledat bot s názvem *BotFather* a zahájit s ním konverzaci. Tento bot umožňuje vytvářet, konfigurovat a spravovat vlastní boty v rámci aplikace Telegram.

Vytvoření nového bota:

1. /newbot
2. LAB_notifikace (jméno)
3. LAB_notbot (uživatelské jméno)
4. Obdržení HTTP API tokenu: 1105302101:AAGETZ_pvNfB8GksLxFbeMQ0a113HdVyPzA

Obdržený token slouží pro operace s botem. Pro zobrazení všech botů se zadá příkaz */mybots*. Po kliknutí na vytvořeného bota se jménem *LAB_notifikace* je možné provádět jeho správu viz obrázek 5.15.



Obr. 5.15: Správa bota v aplikaci Telegramu.

Každého bota lze konfigurovat pro různé účely. Ve výchozím nastavení bot zachovává soukromí a nedokáže číst zprávy v rámci skupiny, do které je přidán.

Vytvořený bot se vyhledá a zahájí se s ním konverzace pomocí tlačítka *start*. Aby bylo možné dostávat notifikace pouze do určené konverzace, je potřeba zjistit identifikační číslo cílové konverzace. To lze udělat zasláním libovolné zprávy do konverzace s botem a následně použít příkaz CURL[1] s URL v následujícím tvaru *https://api.telegram.org/bot<přístupový-API-token>/getUpdates?offset=0*.

Výpis 5.5: Požadavek pro získání CHAT ID.

```
[jstangle@host ~]# curl https://api.telegram.org/
   bot1105302101:AAGETZ\_pvNfB8GksLxFbeMQ0a113HdVyPzA/
   getUpdates?offset=0}
```

Odpověď je ve formátu JSON a ukazuje poslední aktivitu bota.

[1] HTTP klient

Výpis 5.6: Odpověď Telegram API.

```
{
  "ok":true,
  "result":[
    {
      "update_id":947710051,
      "message":{
        "message_id":6,
        "from":{
          "id":293667243,
          "is_bot":false,
          "first_name":"Jan",
          "username":"jstangler",
          "language_code":"en"
        },
        "chat":{
          "id":293667243,
          "first_name":"Jan",
          "username":"jstangler",
          "type":"private"
        },
        "date":1590012972,
        "text":"test"
      }
    }
  ]
}
```

V poli *chat* a *id* se nachází jedinečný identifikátor zahájené konverzace, který je *293667243*. Další akce spojené s Telegram botem probíhají až v rámci kapitoly automatizace a centrální správa při instalaci aplikace TelNot a AlertBot.

5.3 Automatizace a centrální správa

Automatizace a centrální správa laboratorní sítě je založena na návrhu z předešlé kapitoly. Pro nástroj Ansible proběhla implementace rolí a playbooků, které je možné spouštět prostřednictvím příkazové řádky nebo po importu ve webovém rozhraní automatizační platformy AWX. Popsány jsou pouze stěžejní body implementace.

V rámci přílohy lze nalézt implementaci rolí pro nasazení a správu následujících služeb nebo prostředků, v závorce jsou názvy rolí:

- **KVM/QEMU** – příprava hypervizoru, instalace libovolné virtualizace, záloha (kvm-install, kvm-vm, kvm-snapshot)
- **Zabbix** – instalace a konfigurace Zabbix serveru, Zabbix agentů pro šifrovanou komunikaci. (zabbix-server, zabbix-agent)
- **Elasticsearch, Kibana, Filebeat, Rsyslog** – instalace a konfigurace hosta pro sběr provozních dat, instalace a konfigurace klientských stanic (data-retention, filebeat-shipper, rsyslog-node)
- **Suricata** – instalace a konfigurace včetně možnosti zasílání notifikací na Telegram pomocí AlertBot (suricata, suricata-alerter)
- **Docker** – instalace a konfigurace hosta včetně přípravy úložiště a reverzní proxy (docker-host)
- **Arpwatch** – instalace a konfigurace hosta s vlatním skriptem pro notifikace přes Telegram (arpwatch)
- **UniFi Controller, AWX** – instalace a konfigurace Docker kontejnerů (docker-unifi, docker-awx)
- **Bitwarden, Nextcloud, DokuWiki** – instalace Docker kontejnerů (docker-bitwarden, docker-nextcloud, docker-dokuwiki)
- **Cowrie** – instalace a konfigurace honeypotu v Docker kontejneru včetně vlastního skriptu pro notifikace na Telegram (docker-cowrie).
- **TelNot** – instalace a konfigurace pro zasílání libovolných notifikací přes Telegram (docker-telnot, telnot-node)
- **Konfigurace pro CentOS/RHEL 7** – souhrn vylepšení pro systém CentOS/RHEL 7 související se službami auditd, fail2ban, firewalld, chronyd, selinux, sshd a další (hardening).
- **Další operace spojené se správou systémů** – aktualizace, zálohy, instalace vybraných repozitářů, rozšíření diskového oddílu (update, rsync-ssh, centos-vm, extend-lvm-part, basic-tools, centos-vm)

Výpis 5.7: Instalace nástroje Ansible.

```
# CentOS/RHEL
yum install -y ansible
# Fedora
dnf install -y ansible
# Ubuntu, Debian
apt install -y ansible
```

Pro jakoukoliv relaci s Ansible playbooky je potřeba seznam hostů a jejich skupin v souboru *hosts*. V příloze je soubor *hosts* uložen ve složce *inventory*.

Výpis 5.8: Příklad souboru hosts

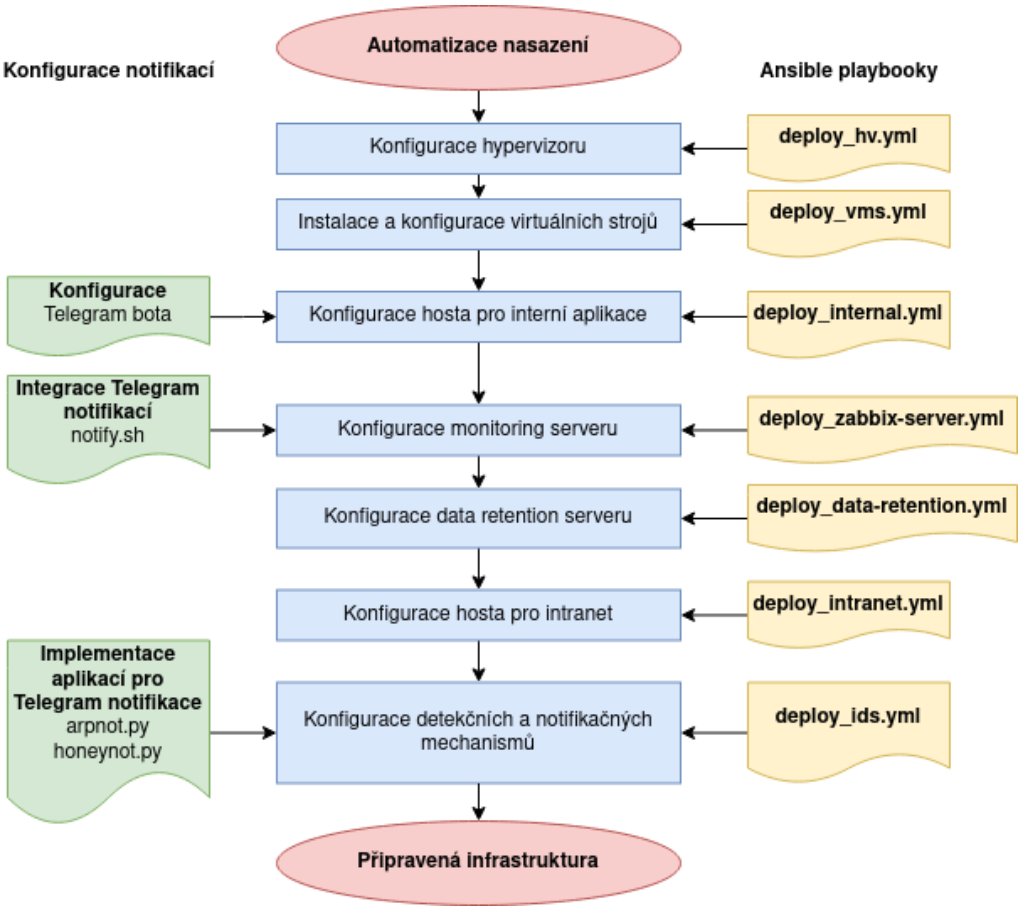
```
[libvirt]
192.168.141.20
[ids]
192.168.141.12
[L1]
192.168.141.12
192.168.141.20
```

Spuštění libovolného playbooku pomocí nástroje Ansible probíhá příkazem:

Výpis 5.9: Příklad spuštění playbooku.

```
[root@host asami]# ansible-playbook playbooks/update_all.yml
```

Na obrázku 5.16 je zpracován diagram automatizace nasazení na laboratorní síť zahrnující stěžejní playbooky a náležitosti pro systém notifikací. Diagram znázorňuje proces celé automatizace a její dílčí kroky. V pravé části se nachází nezbytné kroky, které souviseli s realizací systému notifikací pro aplikaci Telegram. V levé části se nachází názvy playbooků, které můžou být pro daný krok použity.



Obr. 5.16: Diagram automatizace nasazení.

Instalace a konfigurace hypervizoru

Pro instalaci a konfiguraci hypervizoru se spouští playbook *deploy_hv.yml*, který je ve výpise 5.10. Kromě instalace KVM/QEMU zahrnuje také standardní aktualizaci, instalaci základních nástrojů pro správu, instalaci bezpečnostních vylepšení a po zprovoznění serveru Zabbix a databáze Elasticsearch budou správně pracovat také zbylé úkony, kterými je instalace Zabbix agenta pro monitoring a Filebeat pro zasílání logů do centrálního log serveru.

Výpis 5.10: Playbook pro KVM hypervizor.

```
---
- hosts: libvirt
  gather_facts: yes
  roles:
    - { role: common/update }
    - { role: common/basic-tools }
    - { role: common/hardening }
    - { role: kvm-install }
    - { role: common/zabbix-agent, ZABBIX_SERVER:
       192.168.141.6, ZABBIX_CONNECT_ACCEPT: psk,
        ZABBIX_TLS_PSK_IDENTITY: PSK_LAB,
        ZABBIX_TLS_PSK: 8e64ad20456d480bdba115161646b3cd3839
           a7f886bb821f2f03fd5376c2bce4 }
    - { role: common/filebeat-node, FILEBEAT_SHIPPER_NAME: "
       hypervisor", FILEBEAT_MODULES: [ 'system' ], ES_URL: "
       https://192.168.141.5:9200"}
```

Automatizovaná instalace virtuálních strojů

Pro automatizovanou instalaci virtuálních strojů je nutné definovat tzv. kickstart[2]. Ukázka vytvořeného souboru kickstart upraveného na šablonu pro Ansible je možné vidět v příloze ve výpisu A.3.

V kickstartu jsou definovány základní úkony pro instalaci, parametry pro síťovou konfiguraci, nastavení diskových oddílů, heslo pro uživatele root, nastavení služeb, firewallu a přidání veřejných SSH klíčů pro vzdálenou správu. Playbook *deploy_vms.yml* pro automatizovanou instalaci virtuálních hostů v KVM je možné vidět ve výpise 5.11.

[2]umožňuje provádět bezobslužnou instalaci systému

Výpis 5.11: Playbook automatizované instalaci virtuálních hostů

```
---
- hosts: hypervizor
  gather_facts: yes
  become: True
  roles:
    - { role: kvm-vm, vm_name: "Data-Retention", vcpus: 2,
       memory_mb: 4096, disk_size_gb: 50, disk_name: dr,
        bridge: bridge0.41, ip_address: 192.168.141.5,
           net_gateway: 192.168.141.1, hostname: dr.lab.loc }
    - { role: kvm-vm, vm_name: "Zabbix", vcpus: 2, memory_mb:
       3078, disk_size_gb: 30, disk_name: zabbix,
        hostname: zabbix.lab.loc, bridge: bridge0.41,
           ip_address: 192.168.141.6, net_gateway:
           192.168.141.1 }
    - { role: kvm-vm, vm_name: "Zammad", vcpus: 2, memory_mb:
       4096, disk_size_gb: 20, disk_name: zammad,
        bridge: bridge0.44, ip_address: 192.168.144.4,
           net_gateway: 192.168.144.1, hostname: zammad.lab.
           loc }
    - { role: kvm-vm, vm_name: "Docker-intranet", vcpus: 2,
       memory_mb: 4096, disk_size_gb: 30, disk_name: intranet,
        bridge: bridge0.41, ip_address: 192.168.142.2,
           net_gateway: 192.168.142.1, hostname: intranet.lab.
           loc }
    - { role: kvm-vm, vm_name: "Docker-internal", vcpus: 2,
       memory_mb: 4096, disk_size_gb: 20, disk_name: internal,
        bridge: bridge0.41, ip_address: 192.168.141.9,
           net_gateway: 192.168.141.1, hostname: internal.lab.
           loc}
```

Role *kvm-vm* nabízí řadu dalších parametrů k úpravě. Standardně probíhá instalace postupně, host po hostu, ale dodáním parametru *autoconsole: false* lze provést instalaci najednou. Dostupné parametry jsou k nahlédnutí v příloze */roles/kvm-vm/defaults/main.yml*. Patří mezi ně cesta k instalačnímu obrazu ISO, výchozí úložiště pro disk nebo výchozí root heslo. Uvedeným playbookem proběhla pouze samotná příprava virtuálních hostů a aktualizace systému. V uvedeném výpisu je vidět parametry pro virtuální hosty – počet virtuálních procesorů, operační paměť, úložiště, síťová konfigurace a další. Pro konfiguraci jednotlivých systémů je nutné spustit příslušné role nebo playbooky jim určené.

Automatizovaná instalace a konfigurace interních aplikací

Playbook *deploy_internal.yml* zahrnuje instalaci a konfiguraci hosta a interních aplikací v Docker kontejnerech, mezi které patří UniFi Controller, TelNot, AWX a DokuWiki. V případě aplikace TelNot je potřeba vygenerovat libovolný token pro parametr *TELNOT_TOKEN*, který slouží k autentizaci při zasílání notifikací na Telegram.

Výpis 5.12: Playbook hosta pro interní aplikace.

```
---
- hosts: internal
  gather_facts: yes
  roles:
    - { role: common/update }
    - { role: common/centos-vm }
    - { role: common/basic-tools }
    - { role: common/hardening }
    - { role: common/zabbix-agent, ZABBIX_SERVER:
       192.168.141.6, ZABBIX_CONNECT_ACCEPT: psk,
        ZABBIX_TLS_PSK_IDENTITY: PSK_LAB,
        ZABBIX_TLS_PSK: 8e64ad20456d480bdba115161646b3cd3839a7
            f886bb821f2f03fd5376c2bce4 }
    - { role: common/filebeat-shipper, FILEBEAT_SHIPPER_NAME:
       "intranet", FILEBEAT_MODULES: [ 'system' ], ES_URL: "
       https://192.168.141.5:9200"}
    - { role: docker-host }
    - { role: docker-telnot, TEL_CHAT_ID: 293667243,
        TEL_TOKEN: 1105302101
           :AAGETZ_pvNfB8GksLxFbeMQ0a113HdVyPzA,
        TELNOT_TOKEN: 4rFVYuGJV3BgV4jE4Oloary6Zs1rNVvx}
    - { role: docker-unifi }
    - { role: docker-awx }
    - { role: docker-dokuwiki }
```

Při úspěšném nasazení UniFi Controlleru je možné dokončit konfiguraci bezdrátových sítí. Webové rozhraní kontroléru je dostupné na adrese *https://192.168.141.8:8443*. Vzhledem k tomu, že aplikace je dostatečně intuitivní, tak nebude dalšímu postupu konfigurace věnována pozornost. Aplikace AWX vyžaduje hlubší znalosti systémového administrátora, takže je vhodné nasazení této služby zvážit.

Automatizovaná instalace a konfigurace Zabbix serveru

Instalace a konfigurace Zabbix serveru probíhá v rámci role *zabbix-server*, která je součástí playbooku *deploy_zabbix-server.yml*. Účelem Zabbix serveru je monitorovat síť a síťové zařízení. K dispozici je velké množství konfigurovatelných parametrů, které lze nalézt v příloze */roles/zabbix-server/defaults/main.yml*.

Výpis 5.13: Playbook Zabbix serveru.

```
---
- hosts: zabbix
  gather_facts: yes
  become: True
  roles:
    - { role: common/update }
    - { role: common/centos-vm }
    - { role: common/basic-tools }
    - { role: common/hardening }
    - { role: zabbix-server, SERVER_NAME: 192.168.141.6,
       ZABBIX_DB_PASS: KTrRZ7szV2eFOB77}
    - { role: common/telnot-node, TELNOT_TOKEN: 4
       rFVYuGJV3BgV4jE4Oloary6Zs1rNVvx, TELNOT_URL: https
       ://192.168.141.9:5001/notify, ZABBIX_SERVER: true }
    - { role: common/filebeat-shipper, FILEBEAT_SHIPPER_NAME:
       "zabbix", FILEBEAT_MODULES: [ 'system', 'apache' ],
              ES_URL: "https://192.168.141.5:9200"}
```

Výstupem je téměř funkční server pro monitoring. Po otevření webového rozhraní na adrese *https://192.168.141.6* je potřeba provést počáteční konfiguraci – název serveru, přihlašovací údaje nebo údaje pro přihlášení k databázi. Následuje konfigurace monitoringu jednotlivých zařízení a import vhodných šablon.

Součástí playbooku je role *common/telnot-node*, která do Zabbix přidá skript na zasílání notifikací přes aplikaci Telegram. Skript využívá aplikaci TelNot, která byla nasazena na hosta pro interní aplikace.

Automatizovaná instalace centrálního log a NetFlow serveru

Instalace a konfigurace Elasticsearch, Kibany, Filebeat a Rsyslog probíhá v rámci role *data-retention*, která je součástí playbooku *deploy_data-retention.yml*.

Výpis 5.14: Playbook log a NetFlow serveru.

```
---
- hosts: dr
  gather_facts: yes
  roles:
    - { role: common/update }
    - { role: common/centos-vm }
    - { role: common/basic-tools }
    - { role: common/hardening }
    - { role: common/zabbix-agent, ZABBIX_SERVER:
       192.168.141.6, ZABBIX_CONNECT_ACCEPT: psk,
        ZABBIX_TLS_PSK_IDENTITY: PSK_LAB,
        ZABBIX_TLS_PSK: 8e64ad20456d480bdba115161646b3cd3839a7
            f886bb821f2f03fd5376c2bce4 }
    - { role: data-retention }
```

Kromě instalace *data-retention* playbook zahrnuje také standardní aktualizaci, instalaci základních modulů pro virtuálního hosta, základní nástroje pro správu systému a instalaci bezpečnostních vylepšení. Samozřejmostí je také monitoring prostřednictvím Zabbix agenta.

Automatizovaná instalace a konfigurace intranetu

Playbook *deploy_intranet.yml* provádí instalaci a konfiguraci hosta a aplikací intranetu v Docker konejnerech, mezi které patří aplikace Bitwarden a Nextcloud.

Výpis 5.15: Playbook pro intranet.

```
---
- hosts: intranet
  gather_facts: yes
  roles:
    - { role: common/update }
    - { role: common/basic-tools }
    - { role: common/hardening }
    - { role: common/zabbix-agent, ZABBIX_SERVER:
       192.168.141.6, ZABBIX_CONNECT_ACCEPT: psk,
        ZABBIX_TLS_PSK_IDENTITY: PSK_LAB, ZABBIX_TLS_PSK: 8
           e64ad20456d480bdba115161646b3cd3839a7
           f886bb821f2f03fd5376c2bce4 }
    - { role: common/filebeat-shipper, FILEBEAT_SHIPPER_NAME:
       "intranet", FILEBEAT_MODULES: [ 'system' ], ES_URL: "
       https://192.168.141.5:9200"}
    - { role: docker-host }
    - { role: docker-bitwarden }
    - { role: docker-nextcloud }
```

Pro každý Docker kontejner nebo soustavu kontejnerů je vytvořen soubor docker-compose.yml[3] pro snadné nasazení na libovolné distribuci. Kromě standardních úkonů je v playbooku role *docker-host*. Tato role instaluje hlavní závislosti pro Docker, přichystá úložiště a konfiguraci reverzní proxy. V další fázi již probíhá instalace samotných kontejnerů pro správu hesel a místní úložiště.

Automatizovaná instalace a konfigurace detekčních mechanismů

Instalace a konfigurace detekčních mechanismů sítě. Playbook *deploy_ids.yml* zajišťuje instalaci aplikací Arpwatch a Suricata. Součástí je nástroj AlertBot, který zasílá notifikace přímo přes Telegram API, když Suricata detekuje neobvyklou aktivitu. V roli *arpwatch* se nachází aplikace v jazyce Python, která byla implementována, aby bylo možné zasílat notifikace na vzniklé události do aplikace Telegram.

[3]definici převážně pro vícekontejnerové aplikace

Výpis 5.16: Playbook pro IDS mechanismy.

```
---
- hosts: intranet
  gather_facts: yes
  roles:
    - { role: docker-telnot, TEL_CHAT_ID: 293667243,
        TEL_TOKEN: 1105302101
          :AAGETZ_pvNfB8GksLxFbeMQ0a113HdVyPzA,
        TELNOT_TOKEN: 4rFVYuGJV3BgV4jE4Oloary6Zs1rNVvx}
    - { role: docker-cowrie, HOSTNAME: data01, SSH_ENABLED:
       true, SSH_PORT: 222,
        TELNOT_URL: http://localhost:5000/notify, TELNOT_BOT:
          bot1, TELNOT_TOKEN: 4
          rFVYuGJV3BgV4jE4Oloary6Zs1rNVvx}

- hosts: ids
  gather_facts: yes
  become: True
  roles:
    - { role: common/update }
    - { role: common/basic-tools }
    - { role: common/hardening }
    - { role: common/zabbix-agent, ZABBIX_SERVER:
       192.168.141.6, ZABBIX_CONNECT_ACCEPT: psk,
        ZABBIX_TLS_PSK_IDENTITY: PSK_LAB,
        ZABBIX_TLS_PSK: 8e64ad20456d480bdba115161646b3cd3839a7
            f886bb821f2f03fd5376c2bce4 }
    - { role: common/filebeat-shipper, FILEBEAT_SHIPPER_NAME:
       "intranet", FILEBEAT_MODULES: [ 'system' ], ES_URL: "
       https://192.168.141.5:9200"}
    - { role: arpwatch, ARPWATCH_INTERFACE: eth1, TELNOT_URL:
       https://192.168.141.9:5001/notify,
        TELNOT_TOKEN: 4rFVYuGJV3BgV4jE4Oloary6Zs1rNVvx}
    - { role: suricata, INTERFACE: eth1, UPDATE_RULES: daily }
    - { role: suricata-alerter, TEL_TOKEN: 1105302101
       :AAGETZ_pvNfB8GksLxFbeMQ0a113HdVyPzA, CHAT_ID:
       293667243 }
```

Mezi vstupními parametry role *suricata-alerter* se nachází token získaný při vytvoření bota v aplikaci Telegram. V tomto případě dochází k přímé interakci s Telegram API. Ve většině případů je však použita forma lokální proxy – TelNot.

Parametr *TELNOT_TOKEN* představuje individuálně definovanou hodnotu pro ověření klientů zasílacích notifikace na Telegram.

Role *docker-cowrie* instaluje a konfiguruje honeypoty, které slouží k detekci průniku a průzkumu v síti. Pro honeypoty Cowrie existuje integrace do platformy MHN[4], ale v tomto případě jsou použity samostatně. Aby bylo možné průnik zachytit včas, byla vytvořena aplikace v jazyce Python, jenž kontroluje logy na specifické události a zasílá notifikace na Telegram viz výpis v příloze A.5.

[4]systém určený pro správu a monitoring honeypotů

6 Penetrační testování zabezpečené sítě

Penetrační testování označuje testování síťové bezpečnosti na úrovni počítačů, systémů a aplikací. Účelem je odhalit známé zranitelnosti a zvýšit úroveň zabezpečení. Probíhá vždy za souhlasu vlastníka. Testování se obvykle dělí do několika fází – průzkum, mapování cíle, analýza zranitelností, zneužití zranitelností a report.

Existuje několik aspektů, podle kterých se penetrační testování rozlišuje – znalost systému, pozice vůči testované straně a způsobu provedení. Dle znalosti systému se rozlišuje:

- **white-box testování** – tester má úplnou znalost testovaného prostředí (zdrojové a konfigurační kódy)
- **black-box testování** – tester nemá žádné informace o testovaném prostředí

Dle strany odkud se testuje:

- **externí** – z veřejné sítě (Internet)
- **interní** – z vnitřní sítě (síť pro zaměstnance)

Způsob provedení se obvykle kombinuje automatizovanými a poloautomatizovanými nástroji včetně manuálního testování. [83]

Pro penetrační testování byly zváženy celkem 3 různé scénáře průniku do sítě:

- průnik do intranetu – prostřednictvím uživatele s napadeným systémem (malware), prolomení zabezpečení Wi-Fi nebo fyzický přístup k síti
- průnik do sítě DMZ (prostřednictvím služeb vystavených do sítě Internet)
- průnik prostřednictvím sítě pro hosty – nepravděpodobný vzhledem k architektuře sítě

Vytvořená laboratorní síť bude vystavena penetračnímu testování, čímž současně proběhne simulace útoků na síť a její systémy. Půjde o black-box testování z vnitřní sítě. Rozsah je omezen na intranet tzn. 192.168.142.0/24. Přístup pro testování bude poskytnut prostřednictvím již nakonfigurované VPN. Účelem testování je odhalit slabá místa v síti, rizika vzniklá při průniku do sítě a otestovat detekční mechanismy. Do sítě bylo úmyslně přidáno zranitelné zařízení, aby bylo možné snáze detekovat některé útoky. Jednalo se o zařízení MikroTik 133C3 se zastaralým a zranitelným RouterOS verze 3.30 s adresou 192.168.142.12. Služby v síti a na uživatelských stanicích byly nakonfigurovány tak, aby co nejvíce simulovaly danou síť.

6.1 Penetrační testování

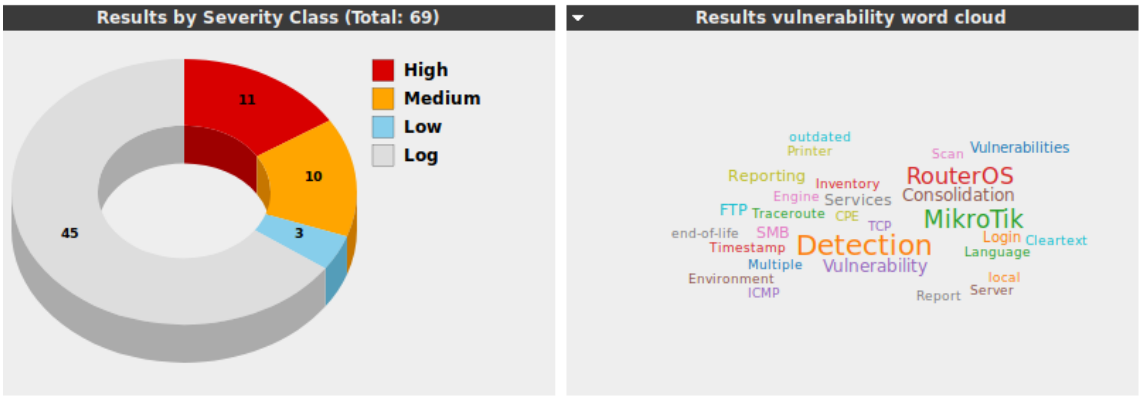
V první fázi proběhl průzkum za pomoci nástroje Nmap[1]. Hledaly se otevřené porty a související aplikace.

[1] bezpečnostní skener k hledání hostitelských počítačů

Výpis 6.1: Příkaz použitý pro průzkum sítě.

```
nmap -sV -sC -sS 192.168.142.0/24
```

Proběhlo mapování nalezených zařízení a byl spuštěn sken pomocí přes OpenVAS[2].



Obr. 6.1: Ukázka výsledků z OpenVAS.

Skener nalezl velké množství zranitelností. Ukázalo se však, že většinou jde o falešně pozitivní (false positive) nálezy. Na zařízení s IP adresou 192.128.142.12 byla otestována zranitelnost CVE-2018-7445, ale tento pokus se nezdařil, jelikož požadovaný port pro uskutečnění nebyl dostupný, což neumožnilo tuto chybu zneužít.

Došlo k útoku hrubou silou na přihlášení do několika služeb a byl získán přístup k SSH na systémy s IP adresou 192.168.142.1, 192.168.142.12 a 192.168.142.18.

Výpis 6.2: Příkaz použitý pro brute-force útok.

```
hydra -l root -P /usr/share/wordlists/rockyou.txt ssh
   ://192.168.142.18 -t 4 na bruteforce
```

Útok hrubou silou na přihlášení k SSH byl úspešný. U IP adres 192.168.142.1 a 192.168.142.18 se nalezlo heslo pro uživatele *root:thisisfine*. Pro IP adresu 192.168.142.12 fungovala kombinace admin:ThisIsFine.

Na portu 443 adresy 192.168.142.18 byl objeven portál pro přihlášení do aplikace Nextcloud a na portu 8443 aplikace pro správu hesel – Bitwarden. Bylo vyzkoušeno založení nového účtu v aplikaci Bitwarden, což aplikace umožňila, ale nedošlo k získání žádných citlivých údajů. V rámci aplikace Nextcloud bylo možné provést jen reset hesla. Výsledky skenování pro adresu 192.168.142.18 jsou vidět ve výpise 6.2.

[2]skener zranitelností

Výpis 6.3: Výsledky skenování z nástroje Nmap pro adresu 192.168.142.18.

```
PORT     STATE  SERVICE          VERSION                                  1
22/tcp   open   ssh              OpenSSH 6.0p1 Debian 4+deb7u2            2
   (protocol 2.0)
| ssh-hostkey:                                                            3
|   1024 51:af:e9:cb:36:80:1a:15:f2:dd:4a:c4:62:1c:bb:2f (DSA)            4
|_  2048 91:5b:7e:4f:74:ec:9b:d7:14:25:c8:f4:e4:13:d1:f7 (RSA)            5
222/tcp   open   ssh              OpenSSH 7.4 (protocol 2.0)               6
| ssh-hostkey:                                                            7
|   2048 05:6f:f5:46:2e:59:ff:6f:71:df:66:1c:3d:1b:69:65 (RSA)            8
|   256 1a:1e:7d:e9:bb:78:8a:ad:7a:a6:78:6b:c2:15:69:62 (ECDSA)           9
|_  256 af:ed:60:36:4b:c0:03:25:4c:2c:24:3b:23:18:3d:d4 (ED25519          10
   )
443/tcp  open   ssl/http         Microsoft IIS httpd 7.5                  11
|_http-server-header: Microsoft-IIS/7.5                                   12
|_http-title: 503 Service Unavailable                                     13
| ssl-cert: Subject: commonName=192.168.142.18/                           14
   organizationName=LAB/countryName=CZ
| Not valid before: 2020-04-26T09:38:08                                   15
|_Not valid after:  2030-05-24T09:38:08                                   16
8443/tcp open   ssl/http         Microsoft IIS httpd 7.5                  17
|_http-server-header: Rocket                                              18
|_http-title: Bitwarden Web Vault                                         19
| ssl-cert: Subject: commonName=192.168.142.18/                           20
   organizationName=LAB/countryName=CZ
| Not valid before: 2020-04-26T09:38:08                                   21
|_Not valid after:  2030-05-24T09:38:08                                   22
MAC Address: 52:54:00:C8:9F:57 (QEMU virtual NIC)                         23
```

Na základě výsledků Nmap skenu pro adresu 192.168.142.18 vznikla domněnka, že se jedná o Windows 7 nebo Windows Server 2008, jelikož hlavička *http-server-header* je Microsoft-IIS/7.5.

Na adrese 192.168.142.15 byla nalezena služba SMB a SSH. Dle průzkumu se jedná o Raspberry Pi s operačním systémem Raspbian. Pomocí nástroje enum4linux byly zjištěny podrobnosti o sdílených složkách protokolu SMB.

Výpis 6.4: Výstup z nástroje enum4linux.

```
 ============================================
|    Share Enumeration on 192.168.142.15    |
 ============================================

  Sharename       Type      Comment
  ---------       ----      -------
  print$          Disk      Printer Drivers
  share           Disk      public shared folder
  private         Disk      public shared folder
  IPC$            IPC       IPC Service (Samba 4.9.5-Debian)
SMB1 disabled -- no workgroup available

[+] Attempting to map shares on 192.168.142.15
//192.168.142.15/print$ Mapping: DENIED, Listing: N/A
//192.168.142.15/share  Mapping: OK, Listing: OK
//192.168.142.15/private  Mapping: DENIED, Listing: N/A
//192.168.142.15/IPC$ [E] Can't understand response:
NT\_STATUS\_OBJECT\_NAME\_NOT\_FOUND listing \*
```

Z uvedeného výpisu je vidět, že server 192.168.142.15 přes protokol SAMBA sdílí složky – *print*, *share* a *private*. Z toho složka *share* je dostupná pod anonymním přihlášením. Uvnitř této sdílené složky se nenacházelo nic zajímavého, ovšem přítomnost složky s názvem *jack-nextcloud-bcp* naznačuje, že na instanci Nextcloud bude pravděpodobně nějaký uživatel *jack*.

Webová rozhraní byla otestována nástroji Nikto, Arachni a ZAP, ale nebyly nalezeny žádné zranitelnosti ani chyby v konfigurace bezpečnostních hlaviček.

Shrnutí nalezených hostů v síti:

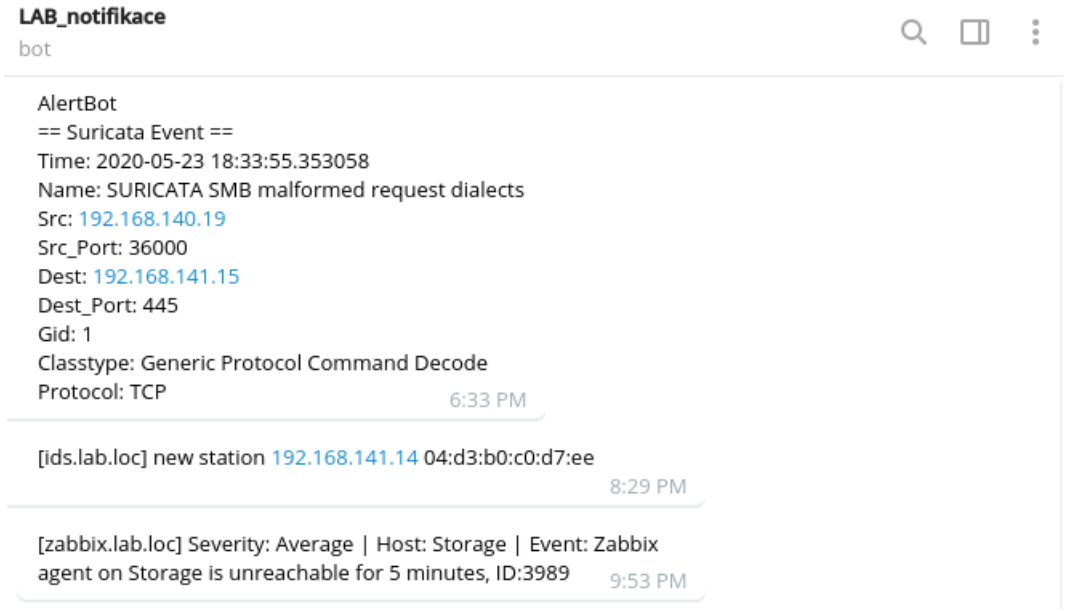
- 192.168.142.8 – otevřený RDP port, dle verze RDP se jedná o Windows 7
- 192.168.142.18 – dle hlavičky webového serveru se jedná o systém Windows Server 2008 (IIS server 7.5)
- 192.168.142.15 – Raspberry Pi se systémem Raspbian, nalezeno několik složek sdílených přes SMB, ale přístupná byla pouze složka *share*
- 192.168.142.12 – MikroTik s verzí RouterOS 3.30, velmi zastaralý software, vyzkoušeno RCE pro CVE-2018-7445, ale SMB port 139 byl zavřený, nalezeno nakonec heslo k webovému rozhraní *admin:ThisIsFine*, podezření na honeypot
- 192.168.142.1 – Debian, jednoduché heslo pro uživatele *root:thisisfine*, dostupný protokol Telnet

Ostatní zařízení nebyli z pohledu penetračního testu zajímavé.

6.2 Vyhodnocení detekčních mechanismů

V průběhu penetračního testu bylo ověřeno, že detekční mechanismy fungují správně a notifikace o událostech přichází na Telegram dle očekávání.

Na obrázku 6.2 je možné vidět ukázku notifikací zaslaných botem na Telegram. První notifikace ukazuje událost zachycenou aplikací Suricata. Většina systémů vytváří notifikace ve formátu „[název hostitele] událost“. Druhá událost značí, že nástroj Arpwatch detekoval nového hosta v síti, kterého ještě nikdy neviděl. Poslední notifikace je ze systému Zabbix pro monitoring sítě a síťových zařízení. Kromě zasílaných notifikací se všechny logy standardně ukládají na centrálním log a NetFlow server.



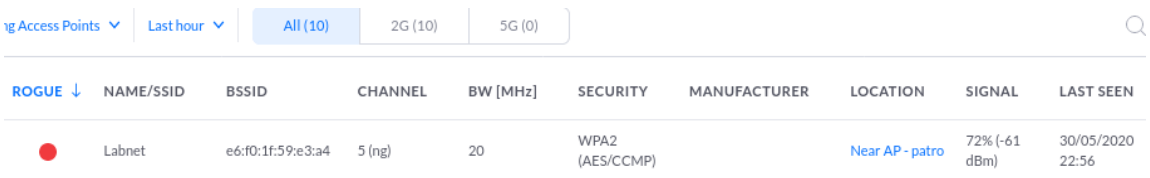
Obr. 6.2: Ukázka notifikací na Telegramu

Penetrační test odhalil většinu zařízení v laboratorní síti. Modifikace hlaviček webového serveru s konfigurací A.4 splnila svůj účel a podařilo se útočníky zmást, na jakém systému webový server opravdu pracuje. U zařízení s IP adresou 192.168.142.12 vzniklo podezření, že se jedná o honeypot. Ve skutečnosti se jednalo o jediné opravdu zranitelné zařízení v síti. Další zmocněné systémy s adresou 192.168.142.1 a 192.168.142.18 byly skutečnými honeypoty a veškerou aktivitu zaznamenaly a oznámily na Telegram. Ukázka logů z centrálního log serveru pro Cowrie honeypoty je na obrázku 6.3.

Obr. 6.3: Záznamy z Cowrie na centrálním log a NetFlow serveru

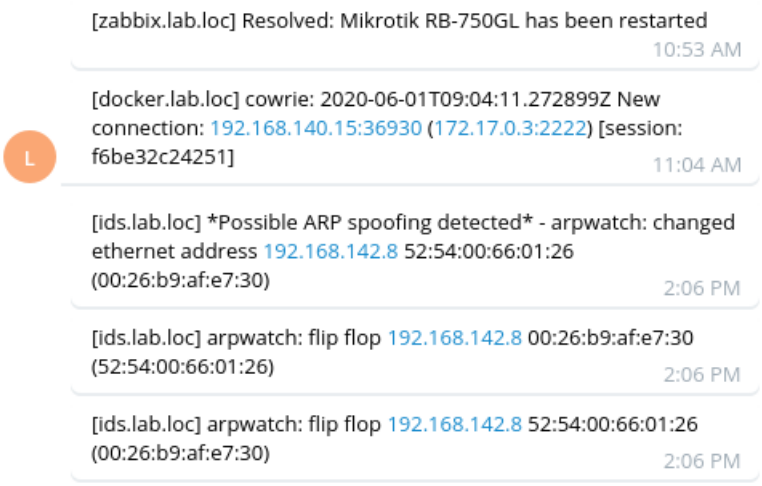
Zpočátku je v záznamech vidět neúspěšné pokusy o přihlášení, při třetím pokusu již bylo zadáno správné heslo a útočník byl vpuštěn do emulace systému Debian, která se chová jako plnohodnotné zařízení, ovšem ve skutečnosti s ním nelze dělat nic, co by mohlo ohrozit ostatní systémy v síti.

Pro ověření detekce falešných AP byl vytvořen v mobilním telefonu Wi-Fi hotspot se stejným názvem SSID jako je laboratorní síť – *Labnet*. Kromě emailového upozornění se také zobrazila notifikace v UniFi Controlleru. Ukázka detekovaného tzv. rogue AP je vidět na obrázku 6.4.



Obr. 6.4: Detekce falešného AP na Unifi Controlleru.

Na závěr proběhl pokus o ARP spoofing síťového zařízení s adresou 192.168.142.8, na kterém se nachází uživatel systému Windows. Arpwatch tuto aktivitu odhalil a zaslal notikaci na Telegram, jak jde vidět na obrázku 6.5. Pojem *flip flop* obecně označuje změnu již známé MAC adresy na jinou známou MAC adresu.



Obr. 6.5: Notifikace ARP spoofing na Telegram

Výsledky penetračního testu ukázaly, že díky segmentaci sítě, bezpečné konfiguraci a navržených detekčních mechanismů nedošlo k přístupu útočníků k citlivým údajům nebo systémům sítě.

6.3 Další postup

V případě, že by některý z útoků probíhal v reálném čase, je potřeba okamžitě reagovat a ověřit, jaký je původ útoku a zda se nejedná o falešný poplach (false positive). V dalším kroku by proběhly úpravy v zabezpečení sítě v místech, kde k útoku došlo.

V síti by měla zlepšit filtrace zachycených událostí pro nástroj Suricata, aby se snížil počet falešných poplachů. Velkým přínosem by bylo nasazení aplikace Bro (Zeek), která by zvýšila pokrytí detekce. Monitoring lze rozšířit o kontrolu integrace souborů na jednotlivých serverech pomocí nástroje Auditbeat[3]. Pro ztížení práce útočníkům by bylo vhodné podvrhnout identitu systémů dalšími technikami, s čímž by mohl pomoci nástroj OSfooler-ng[4]. Pro notifikace je zapotřebí zřídit ještě jednoho Telegram bota, aby se rozlišily události související s monitoringem systémů a bezpečností. Dodatečně by mohla být implementována integrace pro zasílání notifikací z UniFi Controlleru.

Pokud má dojít ke snížení rizika ze strany uživatelských počítačů, je potřeba jistým způsobem kontrolovat nebo vynutit konfiguraci pro uživatelské systémy, ač se jedná o síť zastávající BYOD. Aby se minimalizovalo riziko zranitelných zařízení

[3]umožňuje sběr rámcových dat pro audit ze systémů Linux
[4]`https://github.com/segofensiva/OSfooler-ng`

v síti, je nutné vynutit určité zásady (např. pomocí bezpečnostní politiky) na straně uživatelů.

Závěr

Hlavním cílem práce bylo vytvořit metodiku pro návrh zabezpečené sítě s centrální správou. Na začátku se práce věnovala základům počítačových sítí, jejich správě a monitoringu. Další kapitola se zaměřila na zabezpečení, rizika a hrozby pro počítačové sítě. Byly popsány vybrané útoky na počítačové sítě a uvedeny základní způsoby zabezpečení sítí.

Ve třetí kapitole se práce zaměřila na vytvoření metodiky pro návrh zabezpečené sítě. Nejprve proběhlo seznámení s hierarchickými modely a základními prostředky sítí. Pak byly popsány modely pro architekturu sítě a pravidla pro jejich vytvoření. Pro síťové služby a zařízení vznikla metodika doporučené konfigurace na základě příruček STIG. Nakonec byly popsány základy bezpečnostních politik, jejich náležitosti a prostředky.

Čtvrtá kapitola popisuje návrh sítě pro začínající IT firmu podle vytvořené metodiky. Byly popsány vybrané služby a jejich účel. Při výběru služeb byl kladen důraz na použití open-source řešení.

Pátá kapitola představuje implementaci laboratorní sítě, která je simulací pro návrh sítě z předchozí kapitoly. Přístup k odpovídajícímu hardwaru návrhu nebyl v důsledku celosvětové pandemie způsobené Covid-19 možný, a proto proběhla její implementace pouze v laboratorní podobě pro účely této práce, kterými je mimo vytvořených rolí a playbooků pro centrální správu také následné vyhodnocení detekčních mechanismů penetračním testování sítě. Služby jsou spravovány prostřednictvím implementovaných rolí v Ansible, což umožňuje jejich nasazení na libovolnou infrastrukturu. Všechny vytvořené role jsou zaměřeny na použití na systémů CentOS/RHEL, které jsou dodávány s technologií SELinux. Tato technologie poskytuje další vrstvu bezpečnosti na základě konceptu povinného řízení přístupu (Mandatory Access Control). Ukázalo se, že implementované detekční mechanismy a segmentace sítě výrazně zvyšuje úroveň bezpečnosti v síti.

Výstupem práce je metodika pro návrh zabezpečené malé až středně velké sítě s centrální správou, návrh zabezpečené sítě, rozsáhlá Ansible konfigurace zajišťující automatizované nasazení zásadních částí infrastruktury, simulace provozu a útoků na laboratorní síť, detekce útoků s vyhodnocením závažnosti dopadu útoků na bezpečnost sítě.

Literatura

[1] Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025. *Statista* [online]. [cit. 2019-12-01]. Dostupné z: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] KUROSE, James F. a Keith W. ROSS. *Computer networking: a top-down approach*. 6th ed. Boston: Pearson, 2013. ISBN 978-0-13-285620-1.

[3] *Srovnání modelů ISO/OSI a TCP/IP* [online]. In: . [cit. 2019-11-28]. Dostupné z: http://ijs.8u.cz/images/Vrstvy2.jpg

[4] ANDERSON, Ross. *Security engineering: a guide to building dependable distributed systems*. 2nd ed. Indianapolis, IN: Wiley Pub., c2008. ISBN 978-0-470-06852-6.

[5] FORSHAW, James. *Attacking network protocols: a hacker's guide to capture, analysis, and exploitation*. San Francisco: No Starch Press, [2017]. ISBN 1593277504.

[6] Zapouzdření dat v síti TCP/IP. In: *WIKIMEDIA COMMONS* [online]. [cit. 2019-12-12]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c0/Tcpip_zapouzdreni.svg/1920px-Tcpip_zapouzdreni.svg.png

[7] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008. ISBN 978-80-251-2236-5.

[8] KOTON, Jaroslav. *Aktivní prvky datových sítí prointegrovanou výuku VUT a VŠB-TUO* [online]. Brno: elektronicky, 2014 [cit. 2019-11-21]. ISBN 978-80-214-5066-0. Dostupné z: https://vut-vsb.cz/home/getfile?file=417&amp%3Bportal=Portal2

[9] LABUDA, Adam. *Roaming ve WiFi sítích*. Brno, 2018. Diplomová práce. Vysoké učení tech-nické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Ondřej Krajsa, Ph.D.

[10] About firewalls. *INDIANA UNIVERSITY: Knowledge Base* [online]. [cit. 2019-12-03]. Dostupné z: https://kb.iu.edu/d/aoru

[11] BURDA, Karel. *Návrh, správa a bezpečnost počítačových sítí* [online]. Brno: elektronicky, 2014 [cit. 2019-12-21]. ISBN 978-80-214-5155-1. Dostupné z: https://www.vutbr.cz/www_base/priloha.php?dpid=91322

[12] V čem se IPv6 liší a na co bychom si při jeho implementaci měli dát pozor. *CSIRT.CZ* [online]. [cit. 2019-12-21]. Dostupné z: https://www.csirt.cz/page/3099/v-cem-se-ipv6-lisi-a-na-co-bychom-si-pri-jeho-implementaci-meli-dat-pozor/

[13] THOMSON, Martin. Removing Old Versions of TLS. *Mozilla Security Blog* [online]. [cit. 2019-12-15]. Dostupné z: https://blog.mozilla.org/security/2018/10/15/removing-old-versions-of-tls/

[14] MANAGING RISK FROM TRANSPORT LAYER SECURITY INSPECTION. *National Security Agency: Cybersecurity Information* [online]. [cit. 2019-12-21]. Dostupné z: https://media.defense.gov/2019/Nov/18/2002212783/-1/-1/0/MANAGING%20RISK%20FROM%20TLS%20INSPECTION_20191106.PDF

[15] Stavové kódy a hlášení v odpovědi protokolu HTTP. *Interval.cz* [online]. [cit. 2019-12-13]. Dostupné z: https://www.interval.cz/clanky/stavove-kody-a-hlaseni-v-odpovedi-protokolu-http/

[16] Nastavení DNS záznamů (A, AAAA, MX, CNAME, TXT...). *Active.24: Centrum ná* [online]. [cit. 2019-12-01]. Dostupné z: https://faq.active24.com/cz/866013-Nastaven%C3%AD-DNS-z%C3%A1znam%C5%AF-A-AAAA-MX-CNAME-TXT?l=cs

[17] Exploiting SNMPv1 for Reconnaissance. *Hackers-Arise* [online]. [cit. 2019-12-21]. Dostupné z: https://www.hackers-arise.com/post/2016/06/07/exploiting-snmpv1-for-reconnaissance

[18] SSH, Secure Shell. *Network Sorcery* [online]. [cit. 2019-12-02]. Dostupné z: http://www.networksorcery.com/enp/protocol/ssh.htm

[19] MILFAJT, Jiří. *Bezpečnostní protokoly v praxi*. Brno, 2008. Bakalářská práce. Vysoké učení technické v Brně. Fa-kulta elektrotechniky a komunikačních technologií. Ústav telekomunikací. Vedoucí práce Ing. Tomáš Pelka.

[20] Pokročilá analýza provozu datových sítí. *SystemOnLine* [online]. [cit. 2019-12-21]. Dostupné z: https://www.systemonline.cz/it-security/pokrocila-analyza-provozu-datovych-siti.htm

[21] Monitoring. *IT SLOVNIK.cz* [online]. [cit. 2019-12-21]. Dostupné z: https://it-slovnik.cz/pojem/monitoring

[22] Začínáme s monitoringem sítě. *Www.SAMURAJ-cz.com* [online]. [cit. 2019-12-21]. Dostupné z: https://www.samuraj-cz.com/clanek/zaciname-s-monitoringem-site/

[23] Syslog Tutorial. *Stackify* [online]. [cit. 2019-12-21]. Dostupné z: https://stackify.com/syslog-101/

[24] Deep Packet Inspection. *PC & Network Downloads* [online]. [cit. 2019-12-21]. Dostupné z: https://www.pcwdld.com/deep-packet-inspection

[25] Confidentiality, integrity, and availability (CIA triad). *Whatls.com* [online]. [cit. 2019-12-21]. Dostupné z: https://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA

[26] ISO/IEC 7498-4:1989. *ISO* [online]. [cit. 2019-12-13]. Dostupné z: https://www.iso.org/standard/14258.html

[27] KARTALOPOULOS, S. V. Differentiating Data Security and Network Security. In: *2008 IEEE International Conference on Communications* [online]. IEEE, 2008, 2008, s. 1469-1473 [cit. 2019-12-21]. DOI: 10.1109/ICC.2008.284. ISBN 978-1-4244-2075-9. Dostupné z: http://ieeexplore.ieee.org/document/4533320/

[28] HOCK, Filip a Peter KORTIS. Commercial and open-source based Intrusion Detection System and Intrusion Prevention System (IDS/IPS) design for an IP networks. In: *2015 13th International Conference on Emerging eLearning Technologies and Applications (ICETA)* [online]. IEEE, 2015, 2015, s. 1-4 [cit. 2019-12-21]. DOI: 10.1109/ICETA.2015.7558466. ISBN 978-1-4673-8534-3. Dostupné z: http://ieeexplore.ieee.org/document/7558466/

[29] CIA: Důvěrnost. *CLEVERANDSMART* [online]. [cit. 2019-12-16]. Dostupné z: https://www.cleverandsmart.cz/duvernost/

[30] CIA: Dostupnost. *CLEVERANDSMART* [online]. [cit. 2019-12-15]. Dostupné z: https://www.cleverandsmart.cz/dostupnost/

[31] CIA: Integrita. *CLEVERANDSMART* [online]. [cit. 2019-12-15]. Dostupné z: https://www.cleverandsmart.cz/integrita/

[32] 2018: A Year of Cyber Attacks. *HACKMAGEDDON* [online]. [cit. 2019-12-16]. Dostupné z: https://www.hackmageddon.com/2019/01/15/2018-a-year-of-cyber-attacks/

[33] Basic Network Attacks in Computer Network. *GeeksforGeeks* [online]. [cit. 2019-12-17]. Dostupné z: https://www.geeksforgeeks.org/basic-network-attacks-in-computer-network/

[34] Ethical Hacking - ARP Poisoning. *Tutorialspoint* [online]. [cit. 2019-12-17]. Dostupné z: https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_arp_poisoning.htm

[35] DHCP Starvation attacks and DHCP spoofing attacks. *OmniSecu.com* [online]. [cit. 2019-12-18]. Dostupné z: http://www.omnisecu.com/ccna-security/dhcp-starvation-attacks-and-dhcp-spoofing-attacks.php

[36] *Počítačové sítě a ochrana dat* [online]. Ostrava: elektronicky, 2015 [cit. 2019-12-19]. Dostupné z: https://fbiweb.vsb.cz/ sen76/data/uploads/skripta/poitaove-sit-a-ochrana-dat.pdf

[37] VIGNESH, U. a S. ASHA. Modifying Security Policies Towards BYOD. *Procedia Computer Science* [online]. 2015, **50**, 511-516 [cit. 2019-12-21]. DOI: 10.1016/j.procs.2015.04.023. ISSN 18770509. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S1877050915005244

[38] BURDA, Karel. *Bezpečnost informačních systémů* [online]. Brno: elektronicky, 2013 [cit. 2019-12-21]. ISBN 978-80-214-4890-2. Dostupné z: https://www.vutbr.cz/www_base/priloha.php?dpid=78973

[39] Zabezpečení sítě. *Netcam.cz* [online]. [cit. 2019-12-18]. Dostupné z: https://netcam.cz/encyklopedie-ip-zabezpeceni/zabezpeceni-site.php

[40] Ethical Hacking - DDOS Attacks. *Tutorialspoint* [online]. [cit. 2019-12-17]. Dostupné z: https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_ddos_attacks.htm

[41] Advanced Persistent Threat (APT). *AEC* [online]. [cit. 2019-12-19]. Dostupné z: https://www.aec.cz/cz/Documents/Files/AEC-Advanced-Persistent-Thread.pdf

[42] Síťová bezpečnost. *Extranet.kr-vysocina.cz* [online]. [cit. 2019-12-20]. Dostupné z: https://extranet.kr-vysocina.cz/download/odbor_informatiky/ecrime/Bezpecnost-CS4.pdf

[43] Výstavba moderní sítě. *Www.SAMURAJ-cz.com* [online]. [cit. 2019-12-16]. Dostupné z: https://www.samuraj-cz.com/clanek/vystavba-moderni-site/

[44] Cisco Design Models compilation. *Burningnode* [online]. [cit. 2019-12-18]. Dostupné z: https://www.burningnode.com/2012/12/20/cisco-design-models-compilation/

[45] Hierarchical models. In: *Burningnode* [online]. [cit. 2019-12-18]. Dostupné z: http://www.burningnode.com/images/2012/11/hierarchical-models.jpg

[46] Active Directory administrative tier model. *Microsoft Docs* [online]. [cit. 2019-12-19]. Dostupné z: https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/securing-privileged-access-reference-material

[47] Privileged Access Workstations. *Microsoft docs* [online]. [cit. 2019-12-19]. Dostupné z: https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/privileged-access-workstations

[48] Chraňte si síť pomocí chytré segmentace. *Computerworld* [online]. [cit. 2019-12-20]. Dostupné z: https://computerworld.cz/internet-a-komunikace/chrante-si-sit-pomoci-chytre-segmentace-52532

[49] Security Technical Implementation Guide (STIG) (1): Úvodní seznámení. *Michal Zobec: Blog* [online]. [cit. 2019-12-18]. Dostupné z: https://www.michalzobec.cz/security-technical-implementation-guide-stig-1-uvodni-seznameni-5684

[50] Zabezpečení sítě: Tierování a PAW. *Martin Haller* [online]. [cit. 2019-12-18]. Dostupné z: https://martinhaller.cz/know-how/zabezpeceni-site-tierovani-a-paw/

[51] STIG compliance. *IBM Knowledge Center* [online]. [cit. 2019-12-21]. Dostupné z: https://www.ibm.com/support/knowledgecenter/en/SSHRBY/com.ibm.swg.im.iias.admin.doc/doc/appl_stig_about.html

[52] TIER MODEL FOR PRIVILEGED ACCESS. *SlidePlayer* [online]. [cit. 2019-12-19]. Dostupné z: https://slideplayer.com/slide/13057984/

[53] *NETWORK MANAGEMENT SECURITY GUIDANCE AT-A-GLANCE* [online]. [cit. 2019-12-21]. Dostupné z: https://dl.dod.cyber.mil/wp-content/uploads/stigs/pdf/U_Network_Management_Security_Guidance_At-a-Glance_V9R1.pdf

[54] Network Infrastructure Policy Security. *STIG Viewer* [online]. [cit. 2019-12-20]. Dostupné z: https://www.stigviewer.com/stig/network_infrastructure_policy/

[55] Network Security Requirements Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/network_security_requirements_guide/

[56] Network Device Management Security Requirements Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/network_device_management_security_requirements_guide/

[57] *NETWORK DEVICE MANAGEMENT* [online]. [cit. 2019-12-19]. Dostupné z: https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/U_NDM_V2R15_SRG.zip

[58] Network Devices Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/network_devices/

[59] Infrastructure Router Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/infrastructure_router/

[60] Perimeter Router Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/perimeter_router/

[61] Layer 2 Switch Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/layer_2_switch/

[62] Layer 2 Switch Security Requirements Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/layer_2_switch_security_requirements_guide/

[63] Firewall Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/firewall/

[64] Firewall Security Requirements Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/firewall_security_requirements_guide/

[65] WLAN Authentication Server Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/wlan_authentication_server/

[66] WLAN Access Point Policy Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/wlan_access_point_policy/

[67] *WLAN Access Point (Internet Gateway Only Connection) Security Technical Implementation Guide* [online]. [cit. 2019-12-19]. Dostupné z: https://www.stigviewer.com/stig/wlan_access_point_internet_gateway_only_connection/

[68] *CENTRAL LOG SERVER SECURITY REQUIREMENTS GUIDE* [online]. [cit. 2019-12-20]. Dostupné z: https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/U_Central_Log_Server_V1R2_SRG.zip

[69] Central Log Server Security Requirements Guide. *STIG Viewer* [online]. [cit. 2019-12-20]. Dostupné z: https://www.stigviewer.com/stig/central_log_server_security_requirements_guide/

[70] Red Hat Enterprise Linux 7 Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-20]. Dostupné z: https://www.stigviewer.com/stig/red_hat_enterprise_linux_7/

[71] Intrusion Detection and Prevention Systems (IDPS) Security Requirements Guide. *STIG Viewer* [online]. [cit. 2019-12-20]. Dostupné z: https://www.stigviewer.com/stig/intrusion_detection_and_prevention_systems_idps_security_requirements_guide/

[72] IDS/IPS Security Technical Implementation Guide. *STIG Viewer* [online]. [cit. 2019-12-20]. Dostupné z: https://www.stigviewer.com/stig/idsips/

[73] Tři způsoby zabezpečení firemní sítě v rámci politiky BOYD. *Computerworld* [online]. [cit. 2019-12-16]. Dostupné z: https://computerworld.cz/technologie/tri-zpusoby-zabezpeceni-firemni-site-v-ramci-politiky-boyd-49129

[74] HAVELKOVÁ, Jitka. *Bezpečnostní politika firmy*. Pardubice, 2007. Bakalářská práce. UNIVERZITA PARDUBICE, FAKULTA EKONOMICKO-SPRÁVNÍ, ÚSTAV SYSTÉMOVÉHO INŽENÝRSTVÍ A INFORMATIKY. Vedoucí práce Ing. Renáta Bílková.

[75] Publikované normy. *KYBEZ* [online]. [cit. 2019-12-20]. Dostupné z: https://www.kybez.cz/bezpecnost/serie-norem-iso-27000

[76] ISO/IEC 27001:2013. *RiskAnalysisConsultants* [online]. [cit. 2019-12-20]. Dostupné z: https://www.rac.cz/rac/homepage.nsf/CZ/27001

[77] Koncept Aktivní bezpečnost sítě. *Novicom* [online]. [cit. 2019-12-20]. Dostupné z: https://www.novicom.cz/koncept_aktivni_bezpecnosti_site

[78] Koncept Centrálního monitoringu a IP správy sítě. *Novicom* [online]. [cit. 2019-12-20]. Dostupné z: https://www.novicom.cz/data/183/popis-konceptu.pdf

[79] Meraki Reference Architecture - Small Office Business. *Meraki Documentation* [online]. [cit. 2019-12-20]. Dostupné z: https://documentation.meraki.com/Architectures_and_Best_Practices/Meraki_Reference_Architecture_-_Small_Office_Business

[80] Zabbix Documentation 4.0. *ZABBIX* [online]. [cit. 2019-12-20]. Dostupné z: https://www.zabbix.com/documentation/4.0/manual/introduction

[81] Zabbix — Dohled nad IT infrastrukturou. *ITBLOCK* [online]. [cit. 2019-12-20]. Dostupné z: http://www.itblock.cz/2018/10/24/zabbix-dohled-nad-it-infrastrukturou/

[82] AWX: otevřená varianta Ansible Tower pro automatizaci správy. *Root.cz* [online]. [cit. 2019-12-20]. Dostupné z: https://www.root.cz/clanky/awx-otevrena-varianta-ansible-tower-pro-automatizaci-spravy/

[83] ŠTANGLER, Jan.Integrace nástrojů pro skenování zranitelností. Brno, 2018, 66 s.Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Tomáš Lieskovan

Seznam symbolů, veličin a zkratek

AH	ověřovací hlavička – Authentication Header
API	rozhraní pro programování aplikací – Application Programming Interface
AS	ověřovací server – Authentication Server
DoS	útok odepření služby – Denial of Service
DPI	hloubková inspekce paketů – Deep Packet Inspection
ESP	zapouzdření dat šifrováním – Encapsulating Security Payload
FHSS	frekvenční skokové rozpětí spektra – Frequency Hopping Spread Spectrum
GDPR	obecné nařízení o ochraně údajů – General Data Protection Regulation
HTML	hypertextový značkovací jazyk – Hypertext Markup Language
IDS/IPS	Systém detekce průniku/Systém prevence průniku – Intrusion Detection System/Intrusion Prevention System
IoT	Internet věcí – Internet of Things
KDC	centrum distribuce klíčů – Key Distribution Center
MIME	víceúčelová rozšíření elektronické pošty – Multipurpose Internet Mail Extensions
MTU	maximální přenosová jednotka – Maximum Transmission Unit
NGFW	firewally nové generace – New Generation Firewall
PDU	protokolová datová jednotka – Protocol Data Unit
SPD	databáze zásad zabezpečení – Security Policy Database
TCP	spojově orientovaný protokol – Transmission Control Protocol
TGS	řídicí server – Ticket Granting Server
TGT	tiket opravňující uživatele ke komunikaci s řídícím serverem – Ticket Granting Ticket
ToS	druh služby – Type of Service
TTL	doba života – Time To Live
UDP	nezávisle orientovaný protokol – User Datagram Protocol
UTM	jednotné řízení hrozeb – Unified Threat Management
VLAN	virtuální lokální síť – Virtual Local Area Network
WAF	webový aplikační firewall – Web Application Firewall

Seznam příloh

A Zdrojové kódy

Příloha obsahuje několik programů, skriptů nebo konfiguračních souborů, které byly vytvořeny pro účely této práce. Většina je použita jako šablona pro Ansible role.

Výpis A.1: Bash skript pro posílání notifikací.

```
#!/bin/sh

MESSAGE="message=[$HOSTNAME] $1"
curl -X POST --insecure -d 'token=4
   rFVYuGJV3BgV4jE4Oloary6Zs1rNVvx' -d 'bot=bot1' -d "${
   MESSAGE}" '<telnot_url>'
```

Výpis A.2: Honeynot služba pro systemd.

```
[Unit]
Description=Alert bot for cowrie
After=network.target

[Service]
Type=simple
ExecStart=/usr/bin/python3 /opt/honeynot.py

[Install]
WantedBy=default.target
```

Výpis A.3: Kickstart šablona pro Ansible.

```
# System language
lang {{ system_language }}
# Keyboard layouts
keyboard --vckeymap={{ keyboard_layouts }} --xlayouts='{{
   keyboard_layouts }}'
# System authorization information
auth --enableshadow --passalgo=sha512
# Accept Eula
eula --agreed
# Use CDROM installation media
cdrom
# Use text mode install
text
skipx
# Network information
network --bootproto={{ bootproto }} --device={{ device }} --
   gateway={{ net_gateway }} --ip={{ ip_address }} --
   nameserver={{ nameserver }} --netmask={{ netmask }} --
   noipv6 --activate
network --hostname={{ hostname }}
# Root password
rootpw --iscrypted {{ encrypted_root_password }}
# System services
services --enabled="chronyd"
selinux --enforcing
firewall --ssh
# System timezone
timezone {{ timezone }} --isUtc
# Use only vda disk
ignoredisk --only-use=vda
# System bootloader configuration
zerombr
bootloader --append=" crashkernel=auto" --location=mbr --boot
   -drive=vda
autopart --nohome --type=lvm
# Partition clearing information
clearpart --none --initlabel
```

```
# Don't run the Setup Agent on first boot
firstboot --disable
# Reboot after installing
reboot
%packages
@^minimal
@core
chrony
kexec-tools
%end
%addon com_redhat_kdump --enable --reserve-mb='auto'
%end
%post --log=/root/ks-post.log
#!/bin/bash
mkdir /root/.ssh
cat << xxEOFxx >> /root/.ssh/authorized_keys
{{ ansible_ssh_key }}
{{ admin_ssh_key }}
xxEOFxx
yum update -y
# swapoff -a
%end
```

Výpis A.4: Konfigurační soubor pro server Apache.

```
# Sec
SecRuleEngine on
ServerTokens Full
SecServerSignature "Microsoft-IIS/7.5"

# SSL/TLS config
SSLCertificateFile /etc/pki/tls/certs/docker.crt
SSLCertificateKeyFile /etc/pki/tls/private/docker.key
# intermediate configuration
SSLProtocol             all -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite          ECDHE-ECDSA-AES128-GCM-SHA256:
   ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
   SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
   CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-
   AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
SSLHonorCipherOrder     off

# Nextcloud
Listen 443 https
<VirtualHost *:443>
    SSLEngine On
    Header unset Server
    Header unset X-Powered-By
    SecRuleEngine on
    ProxyPass / http://localhost:8080/
    ProxyPassReverse / http://localhost:8080/
</VirtualHost>

# Bitwarden
Listen 8443 https
<VirtualHost *:8443>
    SSLEngine On
    SecRuleEngine on
    Header unset X-Powered-By
    ProxyPass / http://localhost:88/
    ProxyPassReverse / http://localhost:88/
</VirtualHost>
```

Výpis A.5: Program v jazyce Python pro notifikace z Cowrie na Telegram.

```
import time
import socket
import requests
from requests.packages.urllib3.exceptions import
   InsecureRequestWarning

requests.packages.urllib3.disable_warnings(
   InsecureRequestWarning)

f = open('/root/.docker/cowrie/log/audit.log', 'r')
f.seek(0, 2)
last_source = ''

while True:
    line = ''
    while len(line) == 0 or line[-1] != '\n':
        tail = f.readline()
        if tail == '':
            time.sleep(10)
            continue
        line += tail
    if 'New connection:' in line:
        split_line = line.split(' ', 2)
        source = split_line[2].split(':')[1].strip()
        if last_source == source:
            continue
        alert = '[{}] cowrie: {} {}'.format(socket.
           gethostname(), split_line[0], split_line[2])
        data = {
            'token': '4rFVYuGJV3BgV4jE4Oloary6Zs1rNVvx',
            'bot': 'bot1',
            'message': alert
        }
        response = requests.post('http://localhost:5000/
           notify', data=data, verify=False)
        if response.text == "Success":
            print('alert sent')
            last_source = source
        else:
            print('alert failed')
        time.sleep(3)
```

Výpis A.6: Program v jazyce Python pro notifikace z Arpwatch na Telegram.

```
import time
import socket
import requests
from requests.packages.urllib3.exceptions import
   InsecureRequestWarning

requests.packages.urllib3.disable_warnings(
   InsecureRequestWarning)

f = open('/var/log/messages', 'r')
f.seek(0, 2)
report_messages = ['new activity', 'new station', 'flip flop'
   , 'changed ethernet address']

while True:
    line = ''
    while len(line) == 0 or line[-1] != '\n':
        tail = f.readline()
        if tail == '':
                time.sleep(10)
                continue
        line += tail
    if 'arpwatch:' in line:
        split_line = line.split(' ', 5)[5]
        if not any(message in split_line for message in
           report_messages):
            continue
        if 'changed ethernet address' in split_line:
            split_line = '*Possible ARP spoofing detected* -
               {}'.format(split_line)
        alert = '[{}] {}'.format(socket.gethostname(),
           split_line)
        data = {
            'token': '4rFVYuGJV3BgV4jE4Oloary6Zs1rNVvx',
            'bot': 'bot1',
            'message': alert
        }
        response = requests.post('https://192.168.141.9:5001/
           notify', data=data, verify=False)
        if response.text == "Success":
            print('alert sent')
        else:
            print('alert failed')
        time.sleep(3)
```

B Obsah přiloženého CD

Médium obsahuje soubory pro automatizované nasazení aplikací a služeb pro laboratorní síť přizpůsobenou původnímu návrhu. Součástí jsou také detekční a notifikační mechanismy. Přiložená skladba playbooků byla testována v Ansible verzi 2.9.9 a Python verzi 3.7.7.

```
ansible
├── playbooks
│   ├── deploy_zabbix-agent.yml
│   ├── deploy_hv.yml
│   ├── deploy_vms.yml
│   ├── test.yml
│   ├── deploy_data-retention.yml
│   ├── deploy_filebeat-shipper.yml
│   ├── deploy_intranet.yml
│   ├── deploy_ids.yml
│   ├── deploy_zabbix-server.yml
│   ├── deploy_arpwatch.yml
│   ├── run_hardening.yml
│   ├── deploy_internal.yml
│   ├── deploy_zabbix-server_host.yml
│   ├── update_all.yml
│   └── deploy_vm.yml
├── ansible.cfg
└── roles
    ├── kvm-install
    │   ├── handlers
    │   │   └── main.yml
    │   └── tasks
    │       └── main.yml
    └── common
        ├── centos-vm
        │   ├── handlers
        │   │   └── main.yml
        │   └── tasks
        │       └── main.yml
        ├── basic-tools
        │   ├── tasks
        │   │   └── main.yml
        └── extend-lvm-part
            ├── defaults
            │   └── main.yml
            └── tasks
                └── main.yml
```

```
├── update
│   ├── tasks
│   │   └── main.yml
├── filebeat-shipper
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   └── filebeat.yml.j2
│   ├── handlers
│   │   └── main.yml
│   ├── tasks
│   │   └── main.yml
│   ├── files
│   │   └── elasticsearch.repo
├── telnot-node
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   └── notify.j2
│   ├── tasks
│   │   └── main.yml
├── zabbix-agent
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   └── zabbix_agentd.conf.j2
│   ├── handlers
│   │   └── main.yml
│   ├── tasks
│   │   └── main.yml
├── hardening
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   ├── sshd_config.j2
│   │   ├── sshd.local.j2
│   │   ├── selinux.j2
│   │   └── banner.j2
│   ├── handlers
│   │   └── main.yml
│   ├── tasks
│   │   ├── firewall.yml
│   │   ├── packages.yml
│   │   ├── selinux.yml
│   │   ├── gui.yml
│   │   └── main.yml
```

```
        ├── fail2ban.yml
        ├── ssh.yml
        ├── auditd.yml
        ├── ntp.yml
        └── policy.yml
├── suricata
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   ├── suricata.j2
│   │   ├── suricata.yml.j2
│   │   └── promisc.service.j2
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       └── main.yml
├── docker-telnot
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   ├── config.ini.j2
│   │   └── docker-compose.yml.j2
│   └── tasks
│       └── main.yml
├── docker-host
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   └── vhosts.conf.j2
│   └── tasks
│       └── main.yml
└── zabbix-server
    ├── defaults
    │   └── main.yml
    ├── templates
    │   ├── zabbix_server.conf.j2
    │   └── zabbix.conf.j2
    ├── handlers
    │   └── main.yml
    ├── tasks
    │   ├── postgresql.yml
    │   ├── main.yml
    │   └── zabbix.yml
    └── files
        ├── pg_hba.conf
        └── zabbixalerter.pp
```

```
├── data-retention
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   ├── handlers
│   │   └── main.yml
│   ├── tasks
│   │   ├── elastic.yml
│   │   ├── dataretention.yml
│   │   ├── main.yml
│   │   └── kibana.yml
│   └── files
│       └── elasticsearch.repo
├── docker-awx
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   └── docker-compose.yml.j2
│   └── tasks
│       └── main.yml
├── kvm-vm
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   ├── pool.xml.j2
│   │   └── kickstart-centos7.cfg.j2
│   └── tasks
│       ├── centos7.yml
│       └── main.yml
├── docker-bitwarden
│   ├── defaults
│   │   └── main.yml
│   ├── templates
│   │   └── docker-compose.yml.j2
│   └── tasks
│       └── main.yml
├── docker-unifi
│   ├── tasks
│   │   └── main.yml
│   └── files
│       └── docker-compose.yml
└── docker-cowrie
    ├── defaults
    │   └── main.yml
    └── templates
        └── cowrie.cfg
```

```
        │
        │
        │   │
        │   │   │
        │   │   ├── honeynot.py.j2
        │   │   └── docker-compose.yml
        │   ├── tasks
        │   │   └── main.yml
        │   └── files
        │       └── honeynot.service
        ├── arpwatch
        │   ├── defaults
        │   │   └── main.yml
        │   ├── templates
        │   │   ├── arpnot.py.j2
        │   │   └── arpwatch.j2
        │   ├── handlers
        │   │   └── main.yml
        │   ├── tasks
        │   │   └── main.yml
        │   └── files
        │       └── arpnot.service
        ├── suricata-alerter
        │   ├── defaults
        │   │   └── main.yml
        │   ├── templates
        │   │   └── config.json.j2
        │   ├── tasks
        │   │   └── main.yml
        │   └── files
        │       └── alertbot.service
        └── docker-dokuwiki
            ├── defaults
            │   └── main.yml
            ├── templates
            │   └── docker-compose.yml.j2
            ├── tasks
            │   └── dokuwiki.yml
    ├── inventory
        ├── group_vars
        │   ├── all
        │   └── ssh
        └── hosts
```