Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Analýza zabezpečení sítí na L2

Diplomová práce

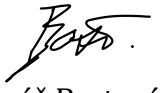
Autor: Bc. Tomáš Bartoníček
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Josef Horálek, Ph.D.

Hradec Králové Duben 2020

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.



V Hradci Králové dne 30.4.2020 Tomáš Bartoníček

Poděkování:

Děkuji vedoucímu diplomové práce Mgr. Josefu Horálkovi, Ph.D. za metodické vedení práce, své rodině a pracovnímu kolektivu za podporu během celého studia.

Anotace

Diplomová práce popisuje funkčnost L2 sítí, jejich protokoly, známé útoky, které se v tomto typu sítí vyskytují a také technologie pro obranu proti těmto útokům. Teoretická část se zabývá výkladem informací o nejpoužívanějších protokolech L2 – Ethernet, ARP, 802.1Q, STP a CDP a navazuje popisem známých síťových útoků ARP spoofing, ARP cache poisoning, MAC spoofing, CAM table overflow, DHCP starvation, VLAN hopping, STP root bridge change. Následně jsou uvedeny technologie zabraňující použití těchto útoků v počítačové síti (Dynamic ARP inspection, Port-security, DHCP snooping, STP Root Guard, STP BPDU Guard).

V praktické části je vypracován nástroj, který po spuštění dokáže pomocí počítače připojeného k LAN síti komplexně otestovat úroveň zabezpečení sítě proti zmíněným útokům na L2. Cílem je tedy vyvinout aplikaci umožňující penetrační testování sítě pomocí útoků na protokoly spojové vrstvy a následně ověřit její funkčnost na modelu s reálnými síťovými zařízeními.

Annotation

Title: Analysis of network security on L2

Diploma thesis describes functionality of L2 networks, their protocols, known attacks which occur in this type of networks and ways of defense against these attacks.

Theoretical part contains information about most used L2 protocols – Ethernet, ARP, 802.1Q, STP and CRP, continues with description of known network attacks such as ARP spoofing, ARP cache poisoning, MAC spoofing, CAM table overflow, DHCP starvation, VLAN hopping, STP root bridge change. Thereafter technologies which prevents using of these attacks in computer network are described (Dynamic ARP inspection, Port-security, DHCP snooping, STP Root Guard, STP BPDU Guard).

In practical part, author creates tool which is able to test network security comprehensively after start them on a computer connected to network. The aim is develop application enabling penetration testing of networks against L2 protocol attacks and verify its functionality on network model with real network devices.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Oblast počítačových sítí zaznamenala, stejně jako celý obor informačních technologií, za posledních několik let znatelný pokrok charakteristický vyšší rychlostí přenosu dat, masovým rozšířením do zařízení domácí a nositelné elektroniky, nízkou cenou komponent, důrazem na uživatelskou přívětivost a také zvýšenými nároky na zabezpečení přenosu. V době, kdy firemní struktura běžně zahrnuje až několik tisíc zařízení, vzájemné propojení poboček, vzdáleně pracující obchodní zástupce, mobilní zařízení návštěvníků či datové centrum s nepřetržitým provozem, je zcela nezbytné řešit obranu před různými typy síťových útoků, které se dnes vyskytují. Zatímco automaticky pracující skenery a boti působí převážně z prostředí internetu na veřejně publikované služby – typicky webový nebo e-mailový server, záškodník uvnitř sítě získává velké pole působnosti a může napáchat mnohem více škody. Z těchto důvodů je nutné nepodcenit vrstvu zabezpečení už při samotném procesu připojení zařízení do sítě a inicializaci komunikace.

Pomineme-li bezpečnost na úrovni fyzické kabeláže a elektrických signálů, dostáváme se na druhou, spojovou vrstvu OSI modelu, která pracuje s množstvím známých protokolů. Právě zde nastává ideální možnost chránit síť před zapojením nežádoucích zařízení a zachytit komunikaci záškodníka. S pomocí správně nastavených bezpečnostních technologií na prvcích sítě lze detekovat podezřelé chování některého koncového bodu a včas zabránit větším škodám.

Tato diplomová práce se zaměřuje na protokoly využívané na druhé vrstvě OSI modelu a zejména pak na jejich bezpečnostní slabiny a obranu proti potenciálním útokům. Teoretická část popisuje princip a funkčnost protokolů Ethernet, ARP, 802.1Q, STP a CDP, jež patří v současné době mezi nejvyužívanější. Dále jsou vysvětleny útoky ARP spoofing, ARP cache poisoning, MAC spoofing, CAM table overflow, DHCP starvation, VLAN hopping a STP Root bridge change včetně obranných technologií dostupných na switchích. Praktická část představuje nástroj určený ke komplexnímu otestování úrovně zabezpečení libovolné počítačové sítě. V závěru jsou vyhodnoceny výsledky testování této aplikace na reálných síťových modelech. Z důvodu vyšší přehlednosti textu je odborná terminologie psána originálními anglickými názvy.

2 Cíl práce

Cílem práce je navrhnout komplexní nástroj pro analýzu a testování zabezpečení počítačových sítí na L2.

V teoretické části autor představí funkcionality a protokoly L2 vrstvy dle architektury TCP/IP včetně analýzy bezpečnostních rizik a útoků. Dále provede analýzu dostupných nástrojů pro testování zabezpečení L2 vrstvy.

V praktické části pak autor vytvoří aplikaci, sloužící pro analýzu a testování zabezpečení L2 s možností využití dostupných nástrojů na základě provedené analýzy.

3 Rešerše tématu

Klíčová slova: *Network, security, analysis, automation, link, layer, attack, vulnerability, simulation, penetration, testing, tool.*

Cílem této práce je především analyzovat bezpečnostní nastavení uvnitř počítačové sítě a reálně sesbírané informace použít jako parametr dále pro porovnání. V praktické části autor vyvine nástroj schopný provést několik typů neznámějších síťových útoků na linkové vrstvě a tím otestovat případné zranitelnosti.

Řada vědeckých článků a jiných odborných publikací o tomto tématu pojednává a do určité míry problematiku řeší. Zejména se jedná o následující odborné práce a články:

Anomaly detection system for enterprise network security
https://patentimages.storage.googleapis.com/3e/65/62/1b8cdf42d951c1/US9112895.pdf

Patent s označením US 9,112,895 B1, vydaný v roce 2015, zahrnuje popis procesu sběru vzorků z reálných dat, detekci neobvyklých událostí v podnikové síti, proces tvorby senzorů a modelu založeného na historických datech.

Navazuje na již existující řešení SIEM, řízených předem definovanými pravidly. Řeší jejich nedostatečnou adaptabilitu na reálný provoz v síti a zmiňuje nevýhodu vysoké míry „false-positive“ nálezů u těchto systémů. Jako optimální je uvedeno řešení sledující anomálie v interní síti (například počet neúspěšných přihlášení, počet otevřených síťových spojení z konkrétní IP adresy, přihlášení z nové geografické oblasti atd.) a odchylku jejich množství od normálu.

Ve své podstatě se jedná o systém pro detekci neobvyklých událostí v síti a jejich sběr.

Zdroj: (1)

Network security testing tools for SMEs (small and medium enterprises)
https://ieeexplore.ieee.org/abstract/document/8394272

Tento článek pojednává o nutnosti zabezpečení podnikových sítí a nedostatku expertů v této oblasti. Zaměřuje se na vývoj nástroje pro menší a střední firmy, který bude schopen provést základní testování bezpečnosti – např. kontrolu konfigurace firewallu, klasifikace zranitelností či simulace útoků DoS. Správcům sítě také nabídne možnosti administrace jako vyhledávání zařízení nebo správu registrů. Stejně jako většina řešení tohoto typu bude schopen generovat reporty o nalezených kritických zranitelnostech.
Autor tedy vyvíjí řešení pro penetrační testování, podle dostupných informací je zaměřeno zejména na zabezpečení aplikační vrstvy.
Zdroj: (2)

PENTOS: Penetration testing tool for Internet of Thing devices
https://ieeexplore.ieee.org/abstract/document/8228241

Aktuálnímu trendu IoT zařízení se věnuje tento článek. Zdůrazňuje, že z důvodu práce s citlivými informacemi a nedostatečného povědomí uživatelů, představují IoT zařízení mnoho potenciálních rizik a stávají se tak novým cílem útočníků.
Skupina autorů popisuje vývoj nástroje PENTOS pro tento typ zařízení založeného na grafickém rozhraní OS Kali Linux. PENTOS získává automaticky informace o zařízeních prostřednictvím bezdrátové komunikace Wi-Fi a Bluetooth a umožňuje provedení různých typů útoku s cílem získání přístupu – cracking hesla, exploatace webového rozhraní nebo zachycení bezdrátové komunikace. Následně nástroj sumarizuje výsledky všech modulů pro testování útoků a navrhne doporučení pro zmírnění nalezených rizik.
Kromě toho poskytuje PENTOS základní bezpečnostní postupy podle žebříčku OWASP Top 10 IoT Vulnerabilites pro edukaci uživatelů a zvýšení povědomí o bezpečnosti.
Jedná se o řešení pro IoT zaměření, z velké části zaměřené spíše aplikačně. Ukazuje možnosti vývoje nástrojů na bázi OS Kali Linux, speciálně určeného pro etický hacking.
Zdroj: (3)

Simulated Penetration Testing and Mitigation Analysis
https://arxiv.org/pdf/1705.05088.pdf

Autoři práce poukazují na přínos penetračního testování u bezpečnostních auditů jako praktického konceptu pro identifikaci potenciálně zranitelných slabých míst důležitých komponent sítě.

Využívají simulaci penetračního testování s automatickým nalezením útoku, založené na síťovém modelu se sadou změn pro prevenci útoku, jakými jsou aplikace změn v topologii sítě, aktualizace systému, změny konfigurace atd. Zjišťují optimální kombinaci, která minimalizuje úspěšnost útočníka.

V praktické části je popsán přístup automatizace v získání modelu sítě pomocí výstupu programů pro skenování síťové topologie, aktuálního nastavení síťových prvků, repozitáře zranitelností NVD a bezpečnostního scanneru Nessus. Autoři následně z těchto vstupů vytvoří sadu potenciálních rizik přímo pro konkrétní síť, provedou analýzu zneužití těchto rizik a vyhodnotí výsledek.

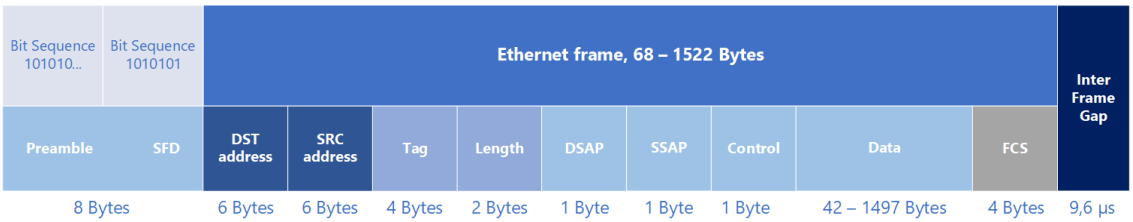
Jde tedy o nástroj s velkou účinností penetračního testování a vysokou mírou adaptability přímo na dané síťové prostředí.

Zdroj: (4)

4 Protokoly spojové vrstvy

4.1 Ethernet

Ethernet je protokol specifikovaný normou IEEE 802.3, udává původní přístup ke sdílenému médiu metodou CSMA/CD a také obecný adresovací formát a přístup k síti.
Původně se pro přenos využíval koaxiální kabel, ke kterému byla připojena veškerá zařízení. Toto řešení s sebou však neslo riziko vzniku kolize při vysílání signálu vícero body v jeden okamžik. Díky metodě CSMA/CD je vznik kolizí ošetřen.
V dnešní době se sdílené médium využívá jen zřídka (s výjimkou bezdrátových sítí), moderní počítačové sítě jsou segmentovány až na úroveň jednotlivých zařízení pomocí switchů, které odesílají data dalším koncovým bodům.



Obrázek 1 - Ethernetový rámec podle normy IEEE 802.3
Zdroj: vlastní tvorba

4.1.1 Metoda CSMA/CD

Tato metoda určuje pravidla pro vysílání na sběrnicové topologii s vícenásobným přístupem k médiu. Řeší již nastalé kolize, avšak nepředchází jim.
Alternativou k CSMA/CD je metoda CSMA/CA. Ta se využívá typicky v prostředí bezdrátových sítí, kde z povahy jejich fungování není možné data přijímat a zároveň vysílat.

4.1.2 Metoda CSMA/CA

Tento způsob komunikace využívá princip rezervace sdíleného média před započetím vysílání některého zařízení.

Na počátku tedy odesílatel naslouchá na médiu, dokud není volné a neprobíhá žádná jiná komunikace. Jakmile se médium uvolní, zvolí se náhodné časové okno. Během tohoto časového úseku stanice naslouchá a monitoruje stav média. Jestliže je médium stále volné, začne vysílat.

(5)
Problematiku metod CSMA a způsoby komunikace na sdílených médiích popisuje podrobněji například Jerry D. Gibson v knize The Communications Handbook. (6)
Způsob komunikace lze dělit podle toho, zda je možné vysílat i přijímat data v jeden časový okamžik.

- **Half-duplex** – v jeden okamžik může být aktivní komunikace pouze jedním směrem
- **Full-duplex** – v jeden okamžik může být komunikace aktivní oběma směry. Při tomto způsobu komunikace nemůže dojít ke kolizi

4.1.3 Popis spojové vrstvy OSI modelu

Subvrstva LLC (Logical Link Control)

Vrstva LLC přímo spolupracuje se třetí, síťovou vrstvou OSI modelu a jejím úkolem je provádění multiplexingu/demultiplexingu, tedy konverze rámců na správný formát paketů pro L3 protokoly (IP, ARP, …) a naopak.

LLC je implementována čistě softwarově v ovladači síťového adaptéru. Protokolům vyšších vrstev předává informace o pozici bufferu v paměti, kam může adaptér uložit datový rámec.

„Tato subvrstva je používána pro bezestavová spojení starším protokolem Ethernet II/DIX. Hlavička LLC má velikost 3 bajty a je umístěna v jednom z datových polí hlavičky Ethernetového rámce. Pro spojově-orientovaná data (SNA, NETBEUI a jiná) je používaná 4bajtová LLC hlavička, zahrnující další dvě pole SAP s hexadecimálním kódem 0x0404 pro SNA a 0xFOFO pro NETBEUI.“

(7)

Popis datové jednotky LLC (LPDU)

„LPDU pole mohou obsahovat buď příkazy a instrukce pro ovládání funkcí spojové vrstvy nebo odpovědi protistrany na již zaslané příkazy a informaci o čísle sekvence. Pole „Control“ (1-2 bajty) udává, zda se jedná o „supervisory frame“ nebo „information frame“ a zda tento rámec obsahuje přijímaná sekvenční čísla používaná pro potvrzení přijetí dat, zotavení z chyb a kontrolu toku provozu. Pole LLC control má délku 1 byte pro bezestavová spojení a 2 byty pro spojově-orientované služby.“

Jak uvádí dále Chwan-Hwa, mezi instrukce lze zařadit ***Receive Ready*** *(RR) a* ***Receive not ready*** *(RNR), vyjadřující aktuální stav, zda zařízení na druhé straně je nebo není připraveno přijímat další data. Typicky se situace vyskytuje u pomalejších zařízení jako jsou tiskárny během jejich zaneprázdnění. V tu dobu vysílají právě tyto RNR a RR rámce. Jedná se o typickou vlastnost služeb spojové síťové komunikace, kdy je po přijetí příkazu odesláno protistranou potvrzení ACK a zároveň kontrolován sled pořadových čísel příkazů* ***NS*** *(Transmitter Send Sequence Number) a* ***NR*** *(Transmitter Receive Sequence Number).*

(8)

Obrázek 2 – LLC PDU (LPDU)
Zdroj: vlastní tvorba

Příklady nejčastěji používaných adres SAP:

- **IP** – 06H
- **Novell IPX** – E0H
- **NetBIOS** – F0H
- **SNAP** – AAH

Popis polí LLC hlavičky

DSAP (1B) – udává pozici ukazatele v paměti přijímající stanice, resp. předává informaci o předpokládaném umístění dat přijímajícímu síťovému adaptéru. Pokud se jedná o rámec SNAP, hodnota DSAP je **AA** nebo **BB**

SSAP (1B) – opačný údaj k DSAP, pozice ukazatele v paměti odesílající stanice

Control (1B) – specifikuje typ LLC rámce, hodnota **0x03** reprezentuje typ SNAP

OC (3B) – identifikátor výrobce NIC zdrojové stanice, obvykle obsahuje první 3 bajty zdrojové MAC adresy rámce

EthType (2 B) – specifikace protokolu zapouzdřeného uvnitř rámce zajišťuje zpětnou kompatibilitu s rámci typu Ethernet II/DIX

Data (38-1492 B) – skládá se z hlaviček vyšších vrstev (například TCP/IP), po kterých následují samotná uživatelská data

(7)

Pole Control obsahuje příkazy spojové vrstvě a odpovědi na ně. Zahrnuje také informaci, zda datový obsah reprezentuje řídící rámec (supervisory frame) nebo informační rámec (information frame). V závislosti na typu rámce pak obsahuje příslušná pořadová čísla (sequence numbers) pro řízení toku dat, zotavení z chyb a potvrzení přijetí dat. Jeho velikost je 1B u nespojových protokolů a 2B při spojové komunikaci.

Typy operací mezi SAP

LLC podvrstva provádí mezi servisními body následující 3 typy operací (označované jako LLC1, LLC2 a LLC3):

1) Prvním typem je komunikace nespojová. Místní SAP odesílá a přijímá informace ze vzdáleného SAP bodu, který také používá nespojový způsob komunikace. Ta neposkytuje žádné možnosti řízení pomocí příkazů. Pole Control v hlavičce rámce je v tomto případě dlouhé **1B**. Pořadí PDU jednotek není nijak číslováno – jsou pouze odesílány, o jejich doručení není zpětně předávána žádná informace. U tohoto typu komunikace také není prováděno řízení toku dat ani zotavení z chyb. Z důvodu kompenzace spolehlivosti doručení protokoly vyšší vrstvy (TCP) je typ LLC1 běžně využíván u vysílání multicast či broadcast.
2) Druhým typem je spojová komunikace. Na rozdíl od prvního typu je každý účastník zodpovědný za udržování navázaného spojení. Řízení provozu mezi

zdrojovou a cílovou LLC vrstvou může být ovlivněno schématem číslování PDU (pořadí se cyklicky opakuje pomocí operace modulo). Jiné nezávislé číslování použité pro detekci ztráty PDU nebo chybové rámce páruje zdrojové a cílové LLC vrstvy. Každý takový pár je logickým point-to-point spojením mezi SAP body linkové vrstvy a bere v úvahu adresaci DA a SA jako části podvrstvy MAC. Funkce potvrzování doručení PDU (ACK) spoléhá na předání informace od cílové LLC směrem ke zdrojové LLC vrstvě o dalším očekávaném pořadovém čísle.

3) LLC3 je nespojová potvrzovaná komunikace. Mezi účastníky tedy není navázáno trvalé spojení, avšak každá stanice musí pro každý pár SSAP-DA udržovat jiné pořadové číslo pro odesílání a jiné pro příjem dat. Po každé řídící PDU jednotce pak následuje potvrzení o jejím přijetí. Přestože zdrojová LLC vrstva může požádat o opětovné odeslání řídící PDU, nová datová jednotka s daným SSAP a DSAP není odeslána, protože cílová LLC stále očekává pořadové číslo předchozí PDU jednotky se stejnou adresou a prioritou.

(8)

Podvrstva LLC-SNAP

*„Důvodem pro zavedení SNAP je ten, že v době, kdy byl LLC navrhován, se většina vědců domnívala, že kapacita jediného bajtu, který by mohlo být použit pro specifikaci až 256 hodnot, bude dostačující pro registraci všech hodnot protokolu. Bohužel, jakmile začaly být hodnoty registrovány, organizace IEEE si uvědomila brzké vyčerpání kapacity LLC hlavičky. Tím pádem hexadecimální hodnoty **0xAA** a **0xBB** byly rezervovány a byla vyvinuta SNAP podhlavička.“*

LLC spolu se SNAP podvrstvou omezují svou velikostí rozsah datové části rámce na pozice 38-1492 bajtů.

(7)

Obrázek 3 – Vložená subvrstva SNAP
Zdroj: vlastní tvorba

Subvrstva MAC (Media Access Control)

Na rozdíl od vrstvy LLC je MAC implementována přímo na hardwaru síťové karty. Plní funkci zapouzdření dat do rámců a řízení přístupu k médiu (například pomocí metody CSMA/CD).

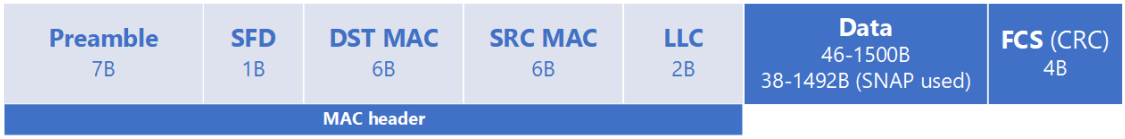
„Datová jednotka MAC protokolu (MPDU) je PDU od MAC adresy k FCS poli. Servisní datová jednotka MAC protokolu (MSDU) je datová jednotka, která je přijímána z podvrstvy LLC, nacházející se nad podvrstvou MAC v zásobníku protokolu. Pokud je MPDU větší než MSDU, jednotka MPDU může obsahovat více MSDU jako výsledek agregace paketů. Pokud je MPDU menší než MSDU, pak jediná MSDU jednotka může generovat více MPDU jako výsledek segmentace paketů.“(8)

Popis operací na MAC subvrstvě

Primárně poskytuje své služby vyšší vrstvě LLC – datové jednotky LPDU jsou odeslány do MAC podvrstvy pro přenos skrze sdílené médium vícenásobného přístupu. Oproti tomu MPDU přijaté na médiu a cílené do LLC jsou přenášeny přes LLC podvrstvu jako servisní datová jednotka LLC-SDU. Podkladová jednotka LLC-SDU v sobě zahrnuje informace o zdrojové a cílové adrese, vlastní data rámce a parametry třídy služby a QoS. Dále tato vrstva poskytuje služby výměny dat mezi CU entitami pro přímo připojené CU uživatele (channel users) izochronním i neizochronním způsobem. Po úvodní inicializaci spojení je uživatel CU schopen přímo přistupovat do komunikačního kanálu zprostředkovaného MAC vrstvou. CU generuje a přijímá datové jednotky přes existující spojení MAC vrstvy na izochronním nebo neizochronním základu. Takováto servisní datová jednotka CU-SDU obsahuje samotná data i parametry QoS. Jakmile je spojení navázáno, nejsou již potřeba informace o adresaci.

MAC subvrstva vyžaduje služby z fyzické vrstvy, které jí poskytují přenos a příjem bitů s informacemi. V předložených rámcích do fyzické vrstvy tak MAC vrstva implementuje algoritmus pro řízení přístupu k médiu, jenž poskytuje svým klientům (LLC nebo vyšším vrstvám) přístup k tomuto sdílenému médiu. V informaci přijaté z fyzické vrstvy využívá MAC subvrstva tento implementovaný algoritmus pro výběr MAC rámců mířených směrem do ní a následně k jejich poskytování protokolům vyšších vrstev. Adresa MAC vrstvy je pak použita k identifikaci cíle rámce.

(6)



Obrázek 4 – Struktura MAC rámce
Zdroj: vlastní tvorba

Popis polí hlavičky MAC rámce

Preamble – slouží pro synchronizaci odesílatele a příjemce rámců, obsahuje hodnoty 0 a 1,

SFD (Start Frame Delimiter) – indikuje počátek datové části rámce,

Destination MAC – MAC adresa síťového adaptéru příjemce rámce,

Source MAC – MAC adresa síťového adaptéru odesílatele rámce,

LLC – obsahuje datovou jednotku LPDU vyšší podvrstvy LLC,

Data – obsahuje data vyšší vrstvy (zde L3 protokolu),

FCS (Frame Check Sequence) – obsahuje CRC (Cyclic redundancy checksum), které poskytuje příjemci kontrolu integrity přijatého rámce a detekci chyb,

(9)

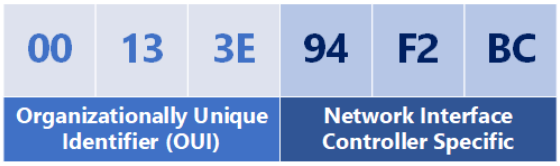
4.1.4 Adresace rámců na spojové vrstvě

Každý ethernetový rámec obsahuje v hlavičce dvě pole potřebná pro správné odeslání a doručení rámce do cíle. Těmi jsou MAC adresy zdrojového a cílového zařízení. Podle těchto údajů je následně protokolem ARP zjištěna skutečná IP adresa zařízení, po které bude komunikace realizována.

4.1.5 MAC adresa

Media Access Control (MAC) adresa je identifikátorem každého síťového adaptéru s podporou protokolu Ethernet, je unikátní, neměnná a přidělená výrobcem NIC. Tato pravidla ustanovuje organizace IEEE a přiděluje také výrobcům síťového hardware OUI identifikátory.

Samotná fyzická adresa je 48bitový hexadecimální kód složený ze šesti skupin po dvou znacích. *Prvních 24 bitů (6 znaků) MAC adresy udává tzv. OUI, což je identifikátor výrobce zařízení. Např. OUI* ***00-90-4B*** *patří společnosti Gemtek Technology Co.* (10)



Obrázek 5 - Části MAC adresy
Zdroj: vlastní tvorba

MAC adresa síťového adaptéru je uložena v ROM paměti, tudíž ji nelze změnit, avšak pomocí softwarových nástrojů lze systém nastavit tak, aby komunikoval prostřednictvím jiné MAC adresy (ethernetový rámec tedy bude mít v hlavičce jinou zdrojovou adresu). Tato technika je v oblasti kybernetické bezpečnosti známá jako MAC spoofing. Z důvodu závislosti funkčnosti některých dalších protokolů na MAC adrese (DHCP, ARP) může být MAC spoofing právě pro ně nebezpečný.
V praxi se lze setkat s následujícími typy MAC adres:

Unicast MAC *– využívá se pro komunikaci s jedním konkrétním zařízením. Poslední bit prvního bajtu adresy má vždy hodnotu 0.*

Multicast MAC *– využívá se pro komunikaci se skupinou zařízení. Poslední bit prvního bajtu adresy má vždy hodnotu 1.*

Broadcast MAC *– využívá se pro komunikaci se všemi zařízeními v dané síti (broadcastové doméně).* Na broadcastovém vysílání závisí funkčnost celé řady služeb (DHCP, ARP, …). *Tvar této adresy je vždy* ***FF-FF-FF-FF-FF-FF.***

(11)

4.1.6 IEEE 802.3 standardy protokolu Ethernet

Postupem času vznikaly různé standardy protokolu, které se liší především rychlostí přenosu dat a použitým médiem. Základní verze 802.3 (známá také pod označením 10BASE5) vyvinutá roku 1983 využívala k přenosu koaxiální kabel a podporovala maximální rychlost přenosu 10 Mbps. Tabulka níže uvádí další nejvýznamnější specifikace protokolu Ethernet.

Standard IEEE	Vznik	Označení	Max. rychlost	Médium
802.3a	1985	10BASE2	10 Mbps	Tenký koaxiální kabel (maximálně 200 m)
802.3j	1993	10BASE-F	10 Mbps	Optický kabel (maximální délka 2 km)
802.3u	1995	100BASE-TX	100 Mbps	Metalický kabel (kroucená dvoulinka, maximální délka 100 m)
802.3z	1998	1000BASE-X	1 Gbps	Optický kabel
802.3ab	1999	1000BASE-T	1 Gbps	Metalický kabel (kroucená dvoulinka)
802.3ae	2003	10GBASE-LR	10 Gbps	Optický kabel
802.3ak	2004	10GBASE-CX4	10 Gbps	Koaxiální kabel
802.3an	2006	10GBASE-T	10 Gbps	Metalický kabel (kroucená dvoulinka)
802.3bq	2013	40GBASE-T	40 Gbps	Metalický kabel (kroucená dvoulinka)
802.3bm	2013		100 Gbps	Optický kabel

Tabulka 1 **- Specifikace protokolu Ethernet,** Zdroj: (12)

Nejznámější specifikace jsou také často uváděny pod názvy:

- **Fast Ethernet** – přenosová rychlost 100 Mbps
- **Gigabit Ethernet** – přenosová rychlost 1000 Mbps
- **Ten Gigabit Ethernet** – přenosová rychlost 10 000 Mbps

Fast Ethernet

Standard Fast Ethernet nahradil v roce 1995 původní standard s rychlostí 10 Mbps. Ten již nedostačoval svým výkonem pro provoz sítí většího rozsahu, které museli administrátoři segmentovat do menších částí. Z těchto důvodů organizace IEEE rozhodla o nutnosti vývoje rychlejšího standardu 802.3u s přenosovou rychlostí 100 Mbps.

Existují 3 varianty média pro Fast Ethernet a to 100Base-T4, 100Base-TX a 100Base-FX. Poslední jmenovaný pracuje s optickými kabely o délce až 2 km, ostatní se standardním krouceným metalickým kabelem maximální délky 100 m. Frekvence přenosu je 25 MHz.

(13)

Gigabit Ethernet

„Stálé zvyšování množství síťového provozu způsobeného narůstajícím výkonem počítačů a požadavky na aplikace bylo motivací pro vznik rychlejší verze Ethernetu. Ta je známá pod názvem Gigabit Ethernet, s označením 802.3z byla schválena v roce 1998 a operuje s rychlostí 1000 Mbps (1 Gbps). Zapojení Gigabit Ethernet existuje jak přímo mezi dvěma stanicemi nebo obvykleji – v síťové topologii star (hvězda) s hubem nebo switchem v jejím středu.“ (13)

Gigabit Ethernet přenáší data po optické kabeláži typu 1000BASE-SX a 1000BASE-LX, měděné kabeláži 1000BASE-CX nebo metalické variantě 1000BASE-T frekvencí 125 MHz.

(13)

Ten Gigabit Ethernet

„10GBASE-T reprezentuje standard IEEE 802.3an-2006, vydaný v roce 2006, který poskytuje připojení rychlosti 10 Gbps skrze nestíněné nebo stíněné páry kabeláže s délkou do 100 m. Jedna z klíčových výhod spojená s 10GBASE-T je možnost použití struktury kabelů pro standard 1000BASE-T umožňující postupnou inovaci pomocí funkce auto-negotiation pro volbu rychlosti.“

(14)

Stejně jako u Gigabit Ethernetu, i zde se využívá síťová topologie star, režim Full-duplex, čímž je dosaženo bezkolizního prostředí. V tomto případě subvrstva MAC pracuje se čtyřmi paralelními vlákny dat z důvodu urychlení zpracování signálu. Využívá frekvenci 322,27 MHz.

(13)

Pro přenos dat je použita kabeláž typů 10GBASE-SR/-LR/-ER/-SW/-LW/-EW (optická), 10GBASE-CX4 (koaxiální) a 10GBASE-T (měděná).

(12)

4.2 ARP

ARP (Address Resolution Procotol) je dalším z nejpoužívanějších protokolů linkové vrstvy. Plní úlohu vyhledání neznámé fyzické adresy zařízení (MAC adresy) ke známé logické adrese (IP adrese) a je definován standardem RFC 826.

Síťová zařízení si ve své operační paměti udržují ARP tabulku s dvojicemi MAC adresa-IP adresa ostatních zařízení ve stejné síti. Pokud tedy potřebuje některá stanice komunikovat s jinou, použije pro zjištění cílové MAC adresy právě ARP tabulku, protože běžné aplikace pro komunikaci používají IP adresy.

(15)

Některá zařízení však protokol ARP nevyužívají z důvodu přímého použití MAC adres pro doručení dat. Jde o zařízení L2 switche, bridge a huby. L3 switche již zahrnují funkce směrování paketů, a tudíž mají i svou ARP tabulku.

(16)

4.2.1 Základní pojmy problematiky ARP protokolu

ARP request – broadcastová žádost o zjištění hardwarové adresy síťového adaptéru stanice.

ARP response – unicastová odpověď na přijatý ARP request paket. Pro jeden ARP request paket by měla být odeslána nejvýše jedna odpověď ARP response paket. Opak by totiž znamenal dvě aktivní stanice v jedné společné síti se stejnou IP adresou, došlo by ke kolizi IP adres.

ARP cache – místo pro ukládání ARP tabulky (v RAM paměti).

ARP cache timeout – časový údaj, po jehož uplynutí daný ARP záznam zmizí z dočasného úložiště (cache) a pro další komunikaci se stanicí bude třeba vyslat ARP request paket pro opětovné zjištění IP adresy.

Statický ARP záznam *– přetrvává v ARP cache i po restartu tohoto zařízení.*

Dynamický ARP záznam *– přetrvává v ARP cache pouze několik minut (doba závisí v závislosti na operačním systému), není-li využíván, je odstraněn.*

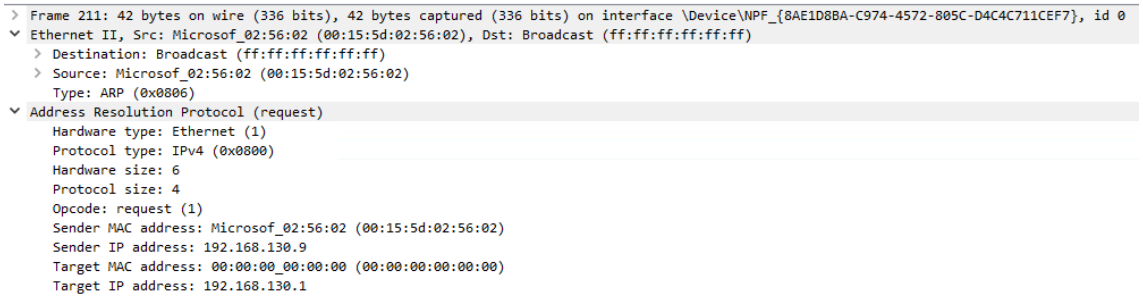
Statické ARP záznamy je možné využívat pouze ve velmi malých počítačových sítích, v jakémkoli větším prostředí by jejich vytváření a následná aktualizace způsobila enormní časovou náročnost správy.
(17)

4.2.2 Proces sestavení ARP tabulky

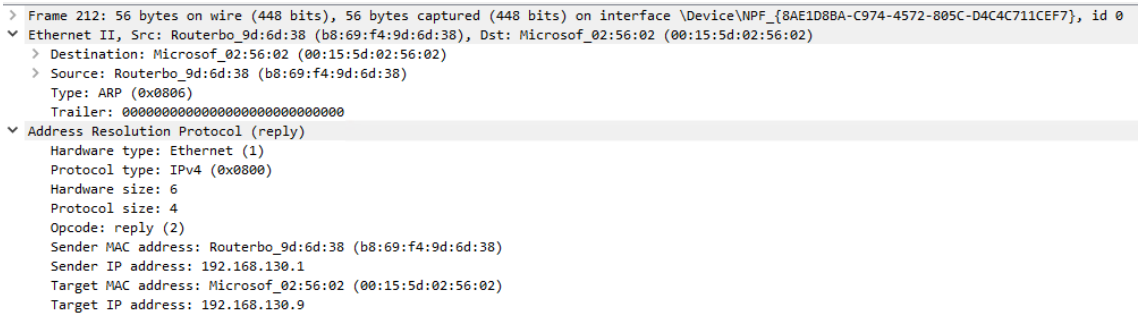
Po zapnutí síťového zařízení je jeho ARP tabulka prázdná. Následně se během doby provozu zařízení provádí analýza provozu ve stejné podsíti a pro každou novou IP adresu, s níž potřebuje stanice komunikovat, je dohledána MAC adresa a doplněna do ARP tabulky.

„Někdy zařízení potřebuje odeslat data stanici, jejíž MAC adresa není v jeho ARP tabulce. V tomto případě je třeba vyslat ***ARP request****. Jedná se o paket distribuovaný broadcastovým typem vysílání. Obsahuje hardwarovou Ethernet adresu a také její IP adresu. Dále je obsažena IP adresa cíle. Všechna ostatní zařízení musí paket přijmout a vyhodnotit jej. Pokud jedno z těchto zařízení zjistí shodu cílové IP adresy se svou vlastní IP adresou, odpoví zdrojové stanici informací o MAC adrese jeho síťového adaptéru. Jakmile žadatel obdrží odpověď, může provést zapouzdření svého IP datagramu do Ethernet rámce a odeslat jej.“*

(15)
Zbývající stanice, které ARP request paket obdrží, jej zahodí, protože není určen přímo pro ně.



Obrázek 6 - Zachycený rámec ARP request
Zdroj: vlastní tvorba



Obrázek 7 - Zachycený rámec ARP reply
Zdroj: vlastní tvorba

Popis atributů ARP rámců

Hardware address type *– údaj specifikující hardwarový standard použitý pro odesílání ARP dat*

Protocol address type *– adresa použitého protokolu vrstvy Internet Layer (např. IP protokol)*

HW address length *(Hardware size) – délka fyzické MAC adresy (u Ehternet protokolu standardně 6 bajtů)*

Protocol address length *(Protocol size) – délka logické IP adresy (standardně 4 bajty v případě IPv4 nebo 16 bajtů v případě IPv6)*

Operation *(Opcode) – udává typ zprávy ARP (request=1; reply=2)*

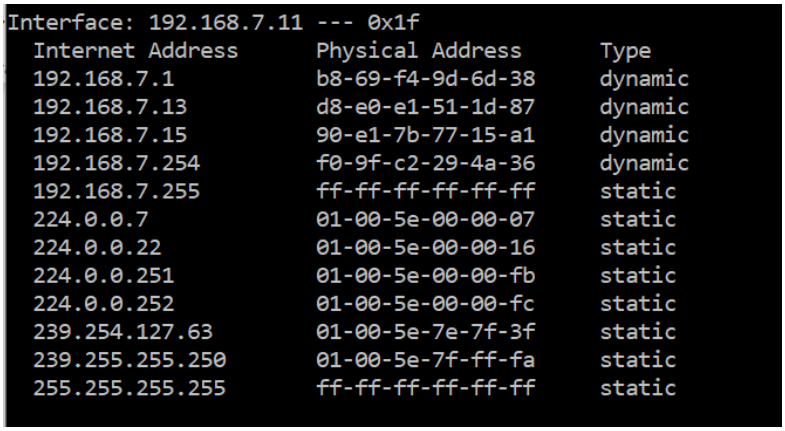
Sender HW address *– MAC adresa odesílatele*

Target HW address *– MAC adresa cílové strany*

Sender protocol address *– použitý protokol na straně odesílatele*

Target protocol addresss *– protokol na straně příjemce*

(18)



```
Interface: 192.168.7.11 --- 0x1f
  Internet Address      Physical Address      Type
  192.168.7.1           b8-69-f4-9d-6d-38     dynamic
  192.168.7.13          d8-e0-e1-51-1d-87     dynamic
  192.168.7.15          90-e1-7b-77-15-a1     dynamic
  192.168.7.254         f0-9f-c2-29-4a-36     dynamic
  192.168.7.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.7             01-00-5e-00-00-07     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.254.127.63        01-00-5e-7e-7f-3f     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Obrázek 8 - ARP tabulka v operačním systému Windows
Zdroj: vlastní tvorba

4.2.3 Procesy v ARP cache

*Proces ARP cache (uchovávání již použitých výsledků ARP requests) má kromě jiného implementaci „bezdůvodný ARP“ (**gratuitous ARP**), která zajišťuje distribuci lokálně uložených párů MAC-IP adres k ostatním stanicím ve stejné síti pomocí broadcast vysílání. V takovém případě si poté mohou všechna zařízení upravovat navzájem své tabulky ARP cache. Pokud se však naopak ARP response vrátí zpět, může být ukončena s chybou o duplicitní MAC adrese. (19)*

Nevalidní záznamy v ARP cache

„Síťové problémy zapříčiněné ARP nejsou vždy pro uživatele jednoduché na trasování tím, že komunikace ARP probíhá transparentně (není registrována uživatelem). Validace ARP záznamu v ARP cache vyžaduje většinu času. Pokud je vadný, neprobíhá žádná komunikace s poškozenou stanicí, dokud je její záznam přítomen. Nekorektní záznamy mohou být následkem například přetíženého počítače s již neaktuální IP adresou. Z důvodu přetížení je tato stanice poslední v odpovědi na ARP request a může potenciálně přepsat případné korektní záznamy v ARP cache.

Cílená manipulace s ARP cache s falšováním IP adres je nazývána jako ARP spoofing a představuje vážný bezpečnostní problém.“

(19)

4.2.4 Nadstavby ARP protokolu

Protokol ARP zahrnuje další rozšíření za účelem pokrytí potřeb při práci se síťovým provozem.

Proxy ARP

Proxy ARP je rozšíření pro standardní verzi protokolu, umožňuje odpovídat na ARP request pakety v zastoupení jiného síťového zařízení. To je užitečné v situacích, kdy potřebujeme přeposílat ARP rámce na jiná zařízení. Tato funkce někdy bývá označována jako tzv. „ARP hack".

Pokud stanice zašle ARP request do sítě, router ve výchozím stavu tento typ komunikace zahodí, protože se jedná o broadcastový typ vysílání na vrstvě L2, kde router nepracuje. Má však svá rozhraní v obou sítích, mezi kterými má komunikace probíhat. Může také odpovídat na ARP request pakety jako další běžná stanice v dané síti s MAC adresou svého rozhraní. Původní zařízení si následně uloží do své ARP tabulky MAC adresu síťového rozhraní routeru a se zařízením v jiné síti může komunikovat.

(14)

Reverse ARP

V porovnání se standardní verzí ARP protokolu je princip fungování této nadstavby přesně opačný – neslouží ke zjištění MAC adresy, nýbrž IP adresy zařízení.

Reverse ARP (RARP) protokol, definovaný standardem RC 903, nacházel v minulosti uplatnění především na bezdiskových stanicích, kde nebylo možné uchovávat údaj o IP adrese zařízení a nebyl používán protokol DHCP pro automatické přidělování TCP/IP konfigurace. Jediným známým identifikátorem tedy zůstávala MAC adresa, pevně uložená výrobcem v hardware síťového adaptéru. Pro správnou funkčnost bylo třeba mít ve stejném síťovém segmentu spuštěný RARP server, jenž odesílal odpovědi s IP adresami koncových zařízení. (16)

Jistou nevýhodou byla nutnost mít samostatný RARP server v každém síťovém segmentu a také náročnost údržby tabulky s mapováním každé MAC adresy k IP adrese. Kromě toho RARP server nepřiděloval další nastavení jako síťovou masku či IP adresu výchozí brány. Formát rámce RARP je stejný jako u protokolu ARP. Podstatný rozdíl je však v předvyplněných hodnotách. Zatímco rámec ARP nemá vyplněnou cílovou MAC adresu, u RARP rámce je prázdné pole IP adresy. (10)

*Další odlišností jsou rozdílné hodnoty v poli Operation – u RARP protokolu obsahuje hodnotu **3** (request) nebo **4** (reply). (18)*

Alternativou k RARP byl později protokol BOOTP, nahrazený dnes velmi rozšířeným protokolem DHCP.

Inverzní ARP

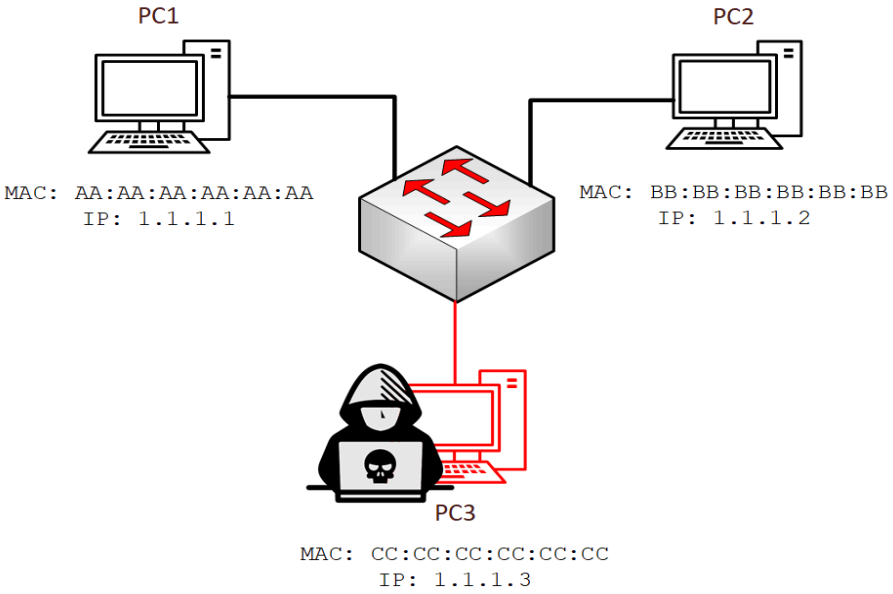
„Inverzní ARP (InARP) je dalším podpůrným protokolem, méně rozšířeným než ARP a RARP. InARP je používaný v sítích typu Frame-Relay k poskytování funkce ARP na Frame-Relay rozhraních".

WAN technologie Frame-Relay pracuje s identifikátory virtuálních okruhů (DLCI) namísto zdrojové a cílové adresy. Čísla DLCI označují virtuální okruh, připojující zdrojový uzel k cílovému. Hlavním účelem InARP je tedy v tomto případě mapování mezi logickou adresou a virtuálním okruhem ve Frame-Relay sítích. (18)

4.2.5 Bezpečnost ARP protokolu

Zejména jednoduchý princip ARP protokolu vytváří dvě hlavní zranitelnosti a činí jeho použití v jistých ohledech nebezpečné.

Při odeslání ARP request a následném vyčkávání na odpověď může dojít k situaci, kdy je cílová stanice vypnutá nebo nedostupná. V takovém případě má libovolná jiná stanice ve stejné síti příležitost vygenerovat si ARP reply s vlastní fyzickou adresou a odeslat ji dotazující se stanici.



Obrázek 9 - Model pokusu o přesměrování provozu pomocí ARP záznamu
Zdroj: vlastní tvorba

To má za následek cílené přesměrování síťového provozu ze zdrojové stanice PC1 do zařízení útočníka PC3. Odesílatel PC1 se však stále domnívá, že komunikuje s legitimním zařízením PC2. Přesměrování je aktivní do doby vypršení záznamu v ARP cache, následně stanice provede další ARP request.

„V tomto případě MAC adresa v návratové zprávě je hardwarová adresa PC2. Tato zpráva přesvědčuje PC1, že PC2 je dostupný a PC1 tedy pokračuje ve vysílání směrem k zamýšlenému cíli PC2 skrze Ethernet. Počítač PC2 je nedostupný, zprávu tedy nepřijme, avšak útočníkův PC3 v promiskuitním režimu může tuto zprávu přijmout.“ (20)

4.3 802.1Q

Standard 802.1Q definuje pravidla pro používání virtuálních sítí (VLAN) v prostředí Ethernetu. Jde o proces tzv. tagování L2 rámců, jenž umožňuje logicky rozdělit jeden fyzický síťový segment na několik menších v závislosti na hodnotě **VLAN ID** – numerického identifikátoru, podle kterého síťové prvky s rámcem dále pracují. Tato hodnota je umístěna v hlavičce rámce. Síťové prvky musí standard značkování rámců 802.1Q podporovat, aby bylo možné této funkcionality využívat.
Užití virtuálních sítí má dva hlavní důvody:

Posílení bezpečnosti sítě *– zařízení ve výchozím stavu nemohou komunikovat mezi virtuálními sítěmi. To lze provádět pouze při použití routeru, který dokáže také filtrovat provoz na síťové vrstvě a zajistí tak směrování pouze pro nás potřebných spojení.*

Snížení provozu v síti *– vyšší průtok broadcastového provozu v síti spotřebovává hardwarové zdroje prvků. Kromě toho zaplňuje větší šířku pásma. Tím, že se jedná o broadcastový provoz, je nutné, aby ho zpracovaly všechny stanice v daném segmentu.*

(21)

4.3.1 Struktura rámce tagovaného podle 802.1Q

L2 rámce prostředí, ve kterém se 802.1Q využívá, mají ve své hlavičce přidanou další část o velikosti 4 bajtů. Ta obsahuje údaje TPID, PRI, CFI a VID (viz Obrázek 11 - Složení části hlavičky rámce pro 802.1Q).

Pokud switch přijme rámec bez značky (untagged frame), zvětší hlavičku rámce o část 802.1Q a do polí doplní potřebné hodnoty.

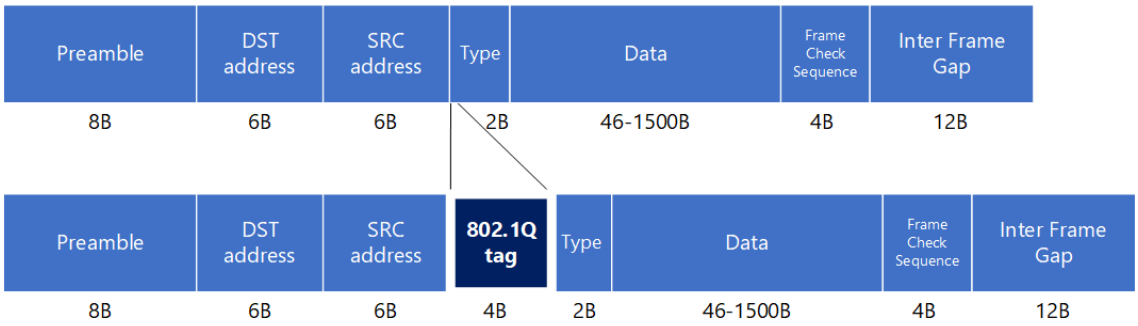
TPID (Tag Protocol ID) *– udává, zda je rámec opatřen značkou, či nikoliv. Hodnota 0x8100 indikuje tagovaný rámec, ostatní hodnoty pak rámec netagovaný.*

PRI (Priority) *– priorita rámce v rozmezí hodnot 0-7. Pokud je switch přetížen, upřednostňuje rámce s vyšší prioritou.*

CFI (Canonical Format Indicator) *– značí, zda je VLAN identifikátor VID kompatibilní s protokolem Ethernet, či nikoliv. Pokud ano, pak je hodnota rovna 0. Nekompatibilními typy jsou například sítě Token Ring.*

VID (VLAN identificator) *– 12bitová číselná hodnota identifikátoru konkrétní virtuální sítě. Může nabývat hodnot 1-4094.*

(21)



Obrázek 10 - Vložení 4bajtové části 802.1Q do hlavičky rámce
Zdroj: vlastní tvorba

Obrázek 11 - Složení části hlavičky rámce pro 802.1Q
Zdroj: vlastní tvorba

4.3.2 Praktické užití virtuálních sítí VLAN

Jak již bylo zmíněno v úvodu této kapitoly, síťové prvky musí podporovat standard 802.1Q, aby byly schopny s tagovanými rámci správně manipulovat.

Protože se jedná o protokol spojové vrstvy, majoritní část funkcionality je závislá na nastavení switchů.

Režimy fyzických portů pro 802.1Q

Každý jednotlivý port switche musí mít nastavenou hodnotu **PVID** (Port VLAN ID). Ta je přiřazována všem neznačkovaným rámcům přijatých na daném portu. Výchozí hodnota PVID je **1**.

Port switche může v závislosti na zamýšleném účelu použití pracovat v jednom z následujících režimů:

1) ***Access port*** *– slouží pro připojení koncového zařízení, které nevysílá již tagované rámce. Switch provede otagování hodnotou uvedenou v parametru Port VLAN ID. Pokud je rámec již otagovaný a hodnota VID rámce nesouhlasí s hodnotou PVID, rámec je zahozen.*

2) ***Trunk port*** *– slouží pro propojení s jiným síťovým zařízením, podporujícím protokol VLAN. Na každém takovém portu musí být nastaven seznam povolených VID, které se zde mohou vyskytnout. Zároveň je i v tomto případě nastavena hodnota PVID. Ta slouží pro označení netagovaného rámce. Pokud hodnota PVID není mezi povolenými VID, rámec je zahozen. V situaci, kdy switch obdrží rámec tagovaný stejným VID jako PVID, je tag odebrán a rámec je dále zpracováván jako netagovaný.*

(21)

Směrování mezi VLAN sítěmi

Z hlediska funkčnosti se VLAN chová jako klasická L2 síť, zařízení tedy mohou ve výchozím stavu komunikovat pouze v rámci jedné takové virtuální sítě. Ve většině případů je však přesto žádoucí, aby byla zařízení schopna komunikovat alespoň určitými druhy provozu i mimo svou VLAN (tento proces je označován jako Inter-VLAN routing). Existuje několik možných scénářů, kterými lze provoz mezi virtuálními sítěmi směrovat na síťové vrstvě:

a) ***Multi-armed router***

Na switchi je pro každou VLAN vyčleněn jeden další fyzický port typu access, jenž je přímo propojen s fyzickým rozhraním routeru. Skrze něj se přenáší provoz konkrétní sítě VLAN – router tedy nijak nepracuje s 802.1Q tagem v hlavičce rámce, proces tagování probíhá pouze na switchi. Nevýhodou je nutnost více fyzických propojení a fyzických rozhraní routeru při vyšším počtu VLAN v síti.

b) One-armed router

Tento způsob propojení síťových prvků pracuje s rozdělením jednoho fyzického rozhraní na routeru na více virtuálních logických rozhraní (subinterfaces), korespondujících s identifikátory VLAN a majících stejnou MAC adresu jako fyzické rozhraní, kterému náleží. IP adresa tohoto virtuálního rozhraní je výchozí branou dané VLAN, do níž virtuální rozhraní patří.

Router a switch jsou v tomto případě propojeny portem typu trunk. Z toho vyplývá, že router musí pracovat s VLAN tagem v hlavičce rámce. Po jeho přijetí na fyzické rozhraní (tedy podle VID) rámec odešle na příslušné logické virtuální rozhraní.

Konfigurace One-armed router je efektivní k využívání hardwarových zdrojů (fyzická kabeláž, obsazené porty zařízení).

c) L3 switch

„Použití One-Armed způsobu propojení má však také nevýhody – pokud je v síti používáno více VLAN s vysokým průtokem síťového provozu mezi VLAN, šířka pásma poskytovaný jedním spojem může být nedostatečná. Kromě toho, dojde-li na takovém spoji k poruše, žádná VLAN nebude moci komunikovat.“

Zmíněné nevýhody lze řešit tzv. L3 switchem. Jedná se o síťový prvek, který v sobě kombinuje funkce klasického switche, a navíc přidává schopnost statického routování paketů. Takto lze provádět inter-VLAN L3 komunikaci ekonomicky, snadno a spolehlivě.

(22)

4.3.3 Typy VLAN z hlediska nasazení

Port-based VLAN – Nastavení je staticky přiřazeno na jednotlivé fyzické porty switche. Výhodou je vyšší bezpečnost, po případném přesunu PC se změní i VLAN, do které je zařazen.

MAC-based VLAN – V paměti switche je uložena tabulka přiřazení MAC adresy zařízení k VID, které je nastaveno na daném portu.

Tento způsob zařazování je sice flexibilnější, nicméně nebezpečný. Potenciálnímu útočníkovi stačí zjistit MAC adresu některé z legitimních stanic, tu zfalšovat a použít u svého zařízení pro získání přístupu do důvěryhodných VLAN sítí.

Protocol-based VLAN – V paměti switche je uložena tabulka přiřazení protokolu k VID, které je nastaveno na daném portu. Jedná se tedy o způsob tagování podle použitého L2 protokolu (IP, IPX, IPv6, ARP, …). V dnešní době se příliš nevyužívá.

4.3.4 GVRP protokol

Jednou z nevýhod použití sítí VLAN je nutnost jejich vytvoření manuálně na každém jednotlivém switchi zvlášť (pokud není využíváno centralizované řízení síťových prvků – controller).

„Switche by měly být schopny registrovat množinu VLAN procházejících skrze specifický spoj bez potřeby manuální konfigurace na každém switchi. V 802.1Q existuje Generic VLAN Registration Protocol (GVRP). Využívá dynamického vytváření a prostupování VLAN na 802.1Q trunk portech. Switche, které protokol GVRP podporují mohou vyměňovat s ostatními switchi (připojenými skrze 802.1Q trunk porty) informace o konfiguraci VLAN, dynamicky je vytvářet, spravovat a tím omezit nepotřebné broadcast pakety a neznámé unicast pakety.“

Pokud je tedy nezbytné například začít provozovat na infrastruktuře novou virtuální síť, postačí ji nastavit na prvním switchi spolu s porty pro koncová zařízení (access porty), které budou do sítě zařazeny. GVRP protokol již zajistí automatickou propagaci informací a správnou konfiguraci VLAN pro průchod Trunk porty, jimiž jsou switche propojeny. Dalším protokolem je MVRP, jenž je efektivnější při zakládání nebo rušení většího množství virtuálních sítí v konfiguraci prvků.

(8)
Pro dynamické šíření informací o VLAN slouží také protokol VTP, který je popsán v následující podkapitole.

4.3.5 VTP protokol

Jedná se o proprietární protokol společnosti Cisco určený pro dynamickou propagaci informací o VLAN sítích mezi jednotlivými switchi. Stejně jako u protokolu GVRP je hlavním účelem VTP usnadnění správy sítě s vyšším počtem switchů při zavádění nových VLAN nebo úpravě stávajících, a také snížení rizika výskytu lidské chyby při konfiguraci. Princip funkčnosti je založen na modelu server-klienti, kdy jeden ze switchů pracuje v režimu tzv. VTP serveru a ostatní switche, které změny přijímají, fungují v roli VTP klienta.

Tento proces je nazýván také jako VTP advertisement a probíhá každých 5 minut. Klientské switche vždy před aplikováním změn porovnají číslo revize konfigurace a pokud je číslo propagované revize vyšší, než číslo lokálně uložené revize, je použita nová konfigurace.

VLAN je možné upravovat i na VTP klientech, tyto změny se však v síti nikam nepropagují a projeví se pouze lokálně na konkrétním switchi.

Dalším typem switchů mohou být tzv. VTP transparent prvky. Ty využívají pouze lokálně nakonfigurovaných VLAN a případné VTP propagační rámce pouze přeposílají dále, na sebe je neaplikují.

Informace o uložených VLAN jsou v případě VTP serveru uchovány v paměti NVRAM (nevolatilní), oproti VTP klientům, kteří pro tyto účely využívají RAM paměť (volatilní).

Pro zvýšení bezpečnosti tohoto protokolu je vhodné chránit proces VTP advertisement heslem. Všichni účastníci procesu jej poté musí mít uložené v konfiguraci, aby došlo k synchronizaci informací.

(23)

4.3.6 Dynamické přiřazování VLAN pomocí protokolu 802.1x

V rozsáhlejších firemních sítích, které navíc vyžadují zařazování stanic do různých sítí v závislosti na atributech, kontrolovatelných pomocí spojení s adresářovými službami (například LDAP), lze efektivně využít ověřování AAA protokoly, typicky RADIUS nebo TACACS.
V takovém případě switch přímo komunikuje s ověřovacím serverem (je v roli RADIUS klienta) a port je zařazen do VLAN tehdy, když ověřovací server předá switchi informaci o úspěšné autentizaci zařízení a o VID, které má být portu přiřazeno.

Častým scénářem také bývá zařazení do veřejné sítě (tzv. Guest VLAN), pokud se nepodaří zařízení úspěšně ověřit.

Ověření počítače nebo uživatele je realizováno pomocí některého atributu oproti adresářovým službám (nejčastěji Microsoft Active Directory). Může se tedy jednat o MAC adresu síťového adaptéru, uživatelské heslo, SMART kartu, certifikát stanice apod.
Způsob zabezpečení přístupu do sítě pomocí 802.1x protokolu je vysoce bezpečný zejména za použití autentizačních prostředků, jenž nelze snadno zfalšovat nebo odcizit. Zařazování zařízení do správné VLAN znamená kromě zvýšené bezpečnosti i absenci potřeby manuálního přiřazování VID každému portu s koncovou stanicí.

Protokol EAPOL

„EAPOL protokol zajišťuje výměnu informací mezi žadatelem a autentizátorem. Iniciuje oznámení identity žadatele a kapacit každého konce. To zajišťuje transport zpráv EAP/EAP-Method, které dovolují autentizaci žadatele a případně i autentizaci serveru.“

Při procesu autentizace se využívají následující základní typy zpráv:

***EAPOL-Start** – slouží k iniciaci 802.1x ověřovacího procesu,*

***EAPOL-Logoff** – slouží k ukončení 802.1x ověřovacího procesu,*

***EAPOL-Key** – slouží k navázání spojení a vytvoření šifrovacích klíčů odvozených z hlavního klíče. V těle zprávy jsou obsaženy informace potřebné k sestavení klíče.*

(24, 25 s. 642)

4.3.7 Virtualizace VLAN sítí (V2LAN)

S rozmachem služeb cloud computing (Microsoft Azure, Amazon AWS, …) a virtualizace serverů, storage i sítí úzce souvisí i nutnost zabezpečení přístupu k takto hostovaným službám. Z důvodu provozu několika instancí na stejném hardware/síti je potřeba zajistit efektivní a bezpečné oddělení prostoru pro jednotlivé klienty.

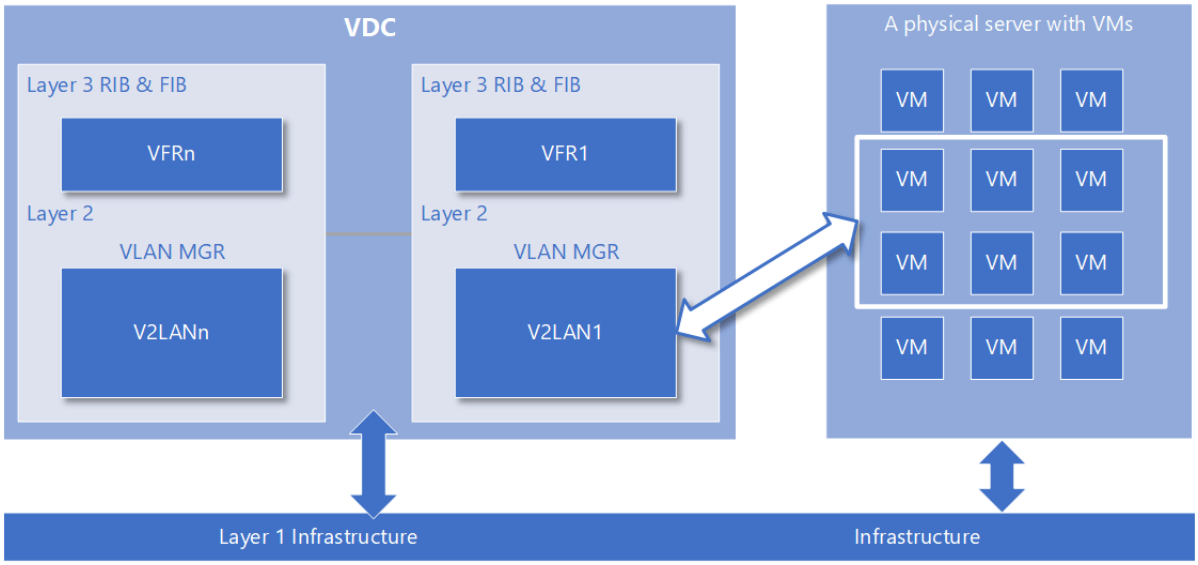
Klasické VLAN síť definuje broadcastovou doménu, v rámci které si mohou zařízení libovolně vyměňovat informace.

Switche s podporou cloudových funkcí pro virtualizaci zařízení na obou vrstvách (L2 i L3) jsou často označovány jako Layer2.5 a umožňují nastavení V2LAN pomocí VDC. Každý z těchto VDC uzlů je oddělenou logickou jednotkou poskytující switching, zabezpečení a další služby pro provoz v cloudovém prostředí. Je integrován se servery, úložišti a orchestrační platformou pro dosažení lepší škálovatelnosti a efektivnější správy a je schopen kompletního L3 routingu provozu skrze technologii VFR.

„Tyto typy switchů jsou často prezentovány jako Layer 2.5 protokol. Například série Cisco Nexus 7000 Series může být konfigurována pro podporu VLAN i V2LAN. V nich pracuje oddělená logická jednotka zahrnující switching, bezpečnost a služby navržené pro fyzická, virtuální a cloudová prostředí. Unikátně se integruje se servery, storage a platformami pro

orchestraci pro vyšší efektivitu operací a lepší škálovatelnost skrze virtual route forwarding (VFR)".

(26)



Obrázek 12 - Integrace virtuálních VLAN
Zdroj: vlastní tvorba

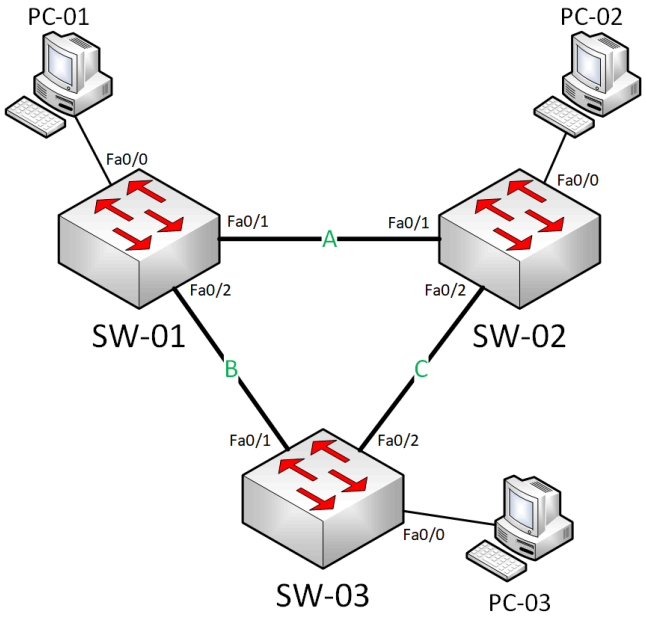
4.4 STP

Protokol STP (Spanning Tree Protocol) slouží k blokování redundantních cest v síti a tím i zabránění vzniku smyček a broadcastových bouří. Jeho využití je tedy určeno především pro topologie typu ring nebo mesh, v běžné praxi ho lze uplatnit i v často používaných topologiích star s nutností řízení redundance vícenásobných propojení mezi klíčovými prvky sítě v distribuční nebo core vrstvě.

U těchto propojení zajišťujících vysokou dostupnost je potřeba zajistit blokaci nadbytečných cest z důvodu prevence vzniku smyček a naopak aktivaci některé záložní trasy v případě výpadku trasy primární.

4.4.1 Smyčky

Smyčkou se rozumí proces, kdy dojde k zacyklení toku paketů v síti bez doručení do cílového bodu. Výsledkem zacyklení je vznik broadcastových bouří a „flappingu" MAC adres. (27)



Obrázek 13 - Model sítě obsahující smyčku
Zdroj: vlastní tvorba

Broadcastová bouře

Při vzniku smyčky v síti a zahájení broadcastového vysílání některým koncovým zařízením dojde k nekonečnému oběhu paketů v síti a postupnému nárůstu jejich objemu.

Proces vzniku broadcastové bouře (popsáno pro Obrázek 13) při zahájení broadcastového vysílání ze stanice PC-01:

1) Stanice PC-01 vyšle broadcastový paket
2) Switch SW-01 přijme paket na rozhraní Fa0/0 a odešle jej na rozhraní Fa0/1 a Fa0/2
3)
    a. Switch SW-02 přijme paket na rozhraní Fa0/1 a odešle jej na rozhraní Fa0/0 a Fa0/2
    b. Stanice PC-02 přijme paket
    c. Switch SW-03 přijme paket na rozhraní Fa0/1 a odešle jej na rozhraní Fa0/0 a Fa0/2
    d. Stanice PC-03 přijme paket
4)
    a. Switch SW-03 přijme paket na rozhraní Fa0/2 a odešle jej na rozhraní Fa0/0 a Fa0/1
    b. Stanice PC-03 přijme paket
    c. Switch SW-02 přijme paket na rozhraní Fa0/2 a odešle jej na rozhraní Fa0/0 a Fa0/1
    d. Stanice PC-02 přijme paket
5)
    a. Switch SW-01 přijme paket na rozhraní Fa0/2 a odešle jej na rozhraní Fa0/0 a Fa0/1
    b. Stanice PC-01 přijme paket
    c. Switch SW-01 přijme paket na rozhraní Fa0/1 a odešle jej na rozhraní Fa0/0 a Fa0/2
    d. Stanice PC-01 přijme paket

Proces rozesílání paketů by takto postupoval dále, dokud by nedošlo k rozpojení některé z linek A, B, C nebo k vypnutí některého ze switchů.

„Flapping“ MAC adres

Tímto jevem se rozumí nekonečné přepisování záznamů v CAM tabulce switche s informací o portu, kam je konkrétního zařízení připojeno.
Při situaci popsané v předchozí podkapitole je v krocích **3a** a **4c** zřejmé, že záznam v CAM tabulce switche SW-02 o pozici PC-01 se bude neustále měnit mezi rozhraními Fa0/1 a Fa0/2. To bude mít za následek omezení primárních funkcí switche, což především v sítích většího rozsahu způsobí znatelné výpadky.

Dalším následkem vzniku smyčky je také nutnost vícenásobného zpracování každého paketu všemi prvky sítě, což způsobí zpravidla jejich přetížení, protože takový nárůst síťového provozu nejsou schopny zpracovat. (27)

Typy portů switche v RSTP protokolu

Typ portu	Popis
Root port	Port spojený s root bridgem buď přímo nebo optimální trasou
Designated port	Port připojující další část sítě, je členem STP stromu
Blocking	Port připojující alternativní trasu k root bridgi, blokovaný STP protokolem

Tabulka 2 - Typy portů RSTP protokolu

Zdroj: (8)

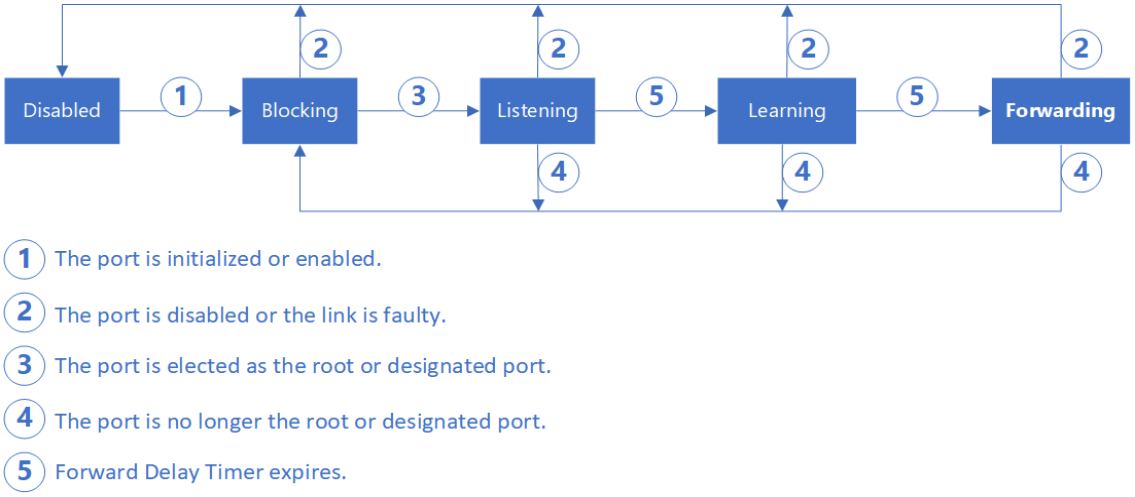
Stavy portů switche v STP protokolu

STP definuje dohromady pět následujících stavů portů závisejících na schopnosti portu odesílat a přijímat STP rámce a uživatelská data.

Stav	Popis
Blocking	Port může pouze přijímat STP rámce, nemůže odesílat STP ani předávat uživatelská data
Listening	Port může odesílat a přijímat STP rámce, avšak nemůže se učit MAC adresy ani předávat uživatelská data
Learning	Port může odesílat a přijímat STP rámce a učit se MAC adresy, avšak nemůže předávat uživatelská data
Forwarding	Port může odesílat a přijímat veškerý provoz i učit se MAC adresy

Tabulka 3 - Stavy portů STP protokolu

Zdroj: (8)



Obrázek 14 - Přechody portů v STP mezi stavy
Zdroj: Tomáš Bartoníček, zpracováno dle (27)

PortFast

Doplňková funkce Spanning Tree urychlující proces přechodu portu mezi STP stavy. Port po připojení vynechá stavy Listening a Learning a přejde přímo do stavu Forwarding. (28) Je doporučeno zapnout tuto funkci na všech koncových (access) portech, aby zapnutí zařízení s rychlou startovací sekvencí nebylo zbytečně zdržováno nedostupným síťovým připojením.

4.4.2 Procesy STP protokolu

V přepínané síti obsahující fyzické smyčky switche spustí STP pro automatické generování bezsmyčkové topologie, která je nazývána jako STP tree. Takovýto strom může obsahovat pouze jeden root bridge. Každý z ostatních bridgů se připojí k root bridge skrze jednu aktivní optimální cestu. Během procesu generování STP tree je nejdříve zvolen root bridge, následně jsou určeny role portů (root, designated) a jako poslední část probíhá blokování portů pro alternativní cesty. Jakmile dojde znovu ke změně v topologii, STP protokol automaticky aktualizuje STP tree odpovídajícím způsobem.“ (27)

Volba root bridge

Samotnému sestavení STP stromu předchází volba hlavního switche – root bridge. *Tím se stane prvek s nejnižší hodnotou BID, kterou může administrátor ovlivnit ruční změnou priority a určit tím tak výběr root bridge (vhodné použít nejvýkonnější switch v infrastruktuře). Pokud má více switchů stejnou prioritu, porovnává se MAC adresa, což je druhá část BID. Jeho vytvoření z MAC adresy switche zajišťuje, že budou BID rozdílné. (3)*

„Po zapnutí všechny switche podporující STP protokol se domnívají, že ony samy jsou root bridgem. Oznámí tedy tento předpoklad v BPDU zprávě, kterou zašlou ostatním switchům. Po přijetí informace v BPDU zprávě switch porovná své vlastní BID s root bridgem specifikovaným v BPDU. Switche poté pokračují ve výměně a porovnávání BPDU, dokud není switch s nejnižším BID zvolen jako root bridge.“

(27)

Volba root portů

Mimo root bridge má každý běžný switch v infrastruktuře právě jednu aktivní cestu k root bridgi – optimální trasu. Fyzický port switche, který tuto trasu připojuje, se nazývá ***root port*** *a je vybrán na základě nejnižšího součtu z cen všech tras vedoucích od tohoto portu do root bridge.* (27)

4.4.3 Výpočet optimální trasy v topologii

Pro výpočet celkové ceny trasy je nejprve třeba všechny její segmenty ohodnotit. Cena linky závisí na její rychlosti a použité verzi STP protokolu.

Rychlost linky	Cena linky (STP)	Cena linky (RSTP)
10 Mbps	100	2 000 000
100 Mbps	19	200 000
1 Gbps	4	20 000
10 Gbps	2	2 000

Tabulka 4 - Přehled cen jednotlivých typů STP linek

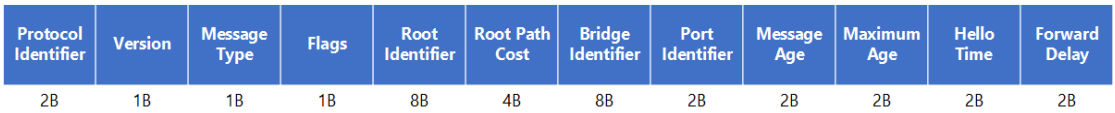
Zdroj: (29)

4.4.4 STP BPDU rámec

Pomocí BPDU rámců probíhá veškerá komunikace mezi switchi související s generováním STP stromové hierarchie. Po jejím sestavením generuje a odesílá BPDU konfigurační zprávy pouze zvolený root bridge.

Ostatní switche BPDU zprávy periodicky přijímají z jejich root portů, zároveň generují vlastní konfigurační BPDU, které odesílají skrze designated porty.

Následuje popis polí BPDU rámce:



Obrázek 15 - Složení STP BPDU rámce

Zdroj: vlastní tvorba

Protocol Identifier *– hodnota vždy 0x0000*

Version *– hodnota vždy 0x00*

Message Type *– udává typ zprávy (BPDU = 0x00, TCN BPDU = 0x80)*

Flags *– identifikátor změny topologie (LSB = TC flag, MSB = TCA flag)*

Root Identifier *– BID identifikátor aktuálního root bridge*

Root Path Cost *– cena optimální trasy k root bridge*

Bridge Identifier *– BID identifikátor switche odesílajícího BPDU*

Port Identifier *– PID identifikátor portu odesílajícího BPDU*

Message Age *– celkový čas potřebný pro odeslání určité konfigurační BPDU z root bridge do aktuálního switche včetně zpoždění při odesílání. Při odeslání z root bridge je hodnota 0.*

Maximum Age *– maximální životnost konfigurační BPDU. Hodnotu určuje root bridge a ve výchozím stavu je nastavena na 20 vteřin. Jakmile switch přijme BPDU, porovná hodnoty Message Age a Maximum Age. Pokud je Message Age menší nebo rovna Max Age, BPDU vyvolá vygenerování a odeslání nové BPDU, v opačném případě je BPDU zpráva ignorována.*

Hello Time *– Časový interval, ve kterém root bridge i ostatní switche odesílají BPDU zprávy. Výchozí hodnota je 2 vteřiny. Jakmile je sestaven STP tree, všechny prvky používají hodnotu Hello Time určenou root bridgem.*

Forward Delay *– Doba, kdy je port ve stavech listening a learning. Jde o pozdržení přechodu portu do stavu stavu forwarding. Po tuto dobu je sestavován STP tree. Během tohoto procesu se stav portu mění. Pokud je zvolen nový root bridge a začne odesílat rámce s daty, může to vést k dočasné smyčce. Z těchto důvodů je zaveden Forward Delay mechanismus – pouze dvakrát lze nově vybraný root port a designated port přepnout do stavu forwarding pro odesílání datových rámců. Tím je garantována topologie bez smyček.*

(27)

4.4.5 TCN BPDU

„Z hlediska struktury a obsahu jsou TCN BPDU rámce jednoduché. Obsahují pouze pole protocol identifier, version number a type (první tři pole standardní BPDU).“

*„Jestliže způsobí výpadek linky změnu síťové topologie, pak switch umístěný v bodě poruchy toto detekuje prostřednictvím informace o stavu portu. Tu však ostatní switche nemají možnost získat. Proto switch umístěný u poruchy začne konstantně vysílat TCN BPDU skrze své root porty v intervalu Hello Time, dokud nepřijme konfigurační BPDU (s TCA příznakem nastaveným na hodnotu **1**), odeslanou tímto nadřazeným switchem. Po přijetí TCN BPDU tento nadřazený switch odpoví konfigurační BPDU zprávou skrze své designated porty a konstantně odesílá TCN BPDU skrze root port. Celý proces se opakuje tak dlouho, dokud root bridge nepřijme TCN BPDU. Jakmile k tomu dojde, root bridge odešle konfigurační BPDU (**TC flag = 1**), kterou upozorní všechny ostatní switche v síti na změnu topologie.“*

Po změně topologie sítě může být obsah CAM tabulky nevalidní. Kvůli tomu switche sníží interval aging time na hodnotu Forward Delay, což zrychlí stárnutí původních záznamů v CAM tabulce.

(27)

4.4.6 Teorie grafů v STP

Přepínanou síť typu Ethernet si lze představit jako graf – stanice a switche reprezentují jeho vrcholy a obousměrná spojení hrany.

Uvažujme neorientovaný graf $G = (V, E)$, kde V je množina vrcholů velikosti n (vertices) a E je množina hran velikosti m (edges). Hrana $e = \{i, j\} \in E$ reprezentuje neorientovanou hranu určenou dvěma vrcholy, $i \in V$ a $j \in V$. (30)

Stupeň vrcholu $deg_G v$ je počet hran, které z daného vrcholu vychází.

Definice problému TE-MSTPP pomocí teorie grafů

V případě definice problému nalezení optimálního návrhu přepínané sítě implementací MSTP lze uvažovat následovně:

Mějme neorientovaný graf $G = (V, E)$, kde každá jeho hrana $e = \{i, j\}$ má definovanou svoji symetrickou kapacitu označenou jako C_e, která limituje celkové množství síťového provozu procházejícího v obou směrech (hrany $\{i, j\}$ a $\{j, i\}$). Dále uvažujme S jako množinu VLAN v ethernetové síti. Pro každou VLAN $s \in S$ reprezentuje $d_s(u, v)$ provoz mezi uzly $u \in V$ a $v \in V$. Pro zjednodušení předpokládejme, že $u < v$ a $d_s(u, v)$ znamená celkové množství odeslaného provozu po hranách $\{u, v\}$ i $\{v, u\}$.

TE-MSTPP definice se skládá z nalezení návrhu všech VLAN $s \in S$ s minimalizací hodnoty worst-case link utilization (poměr mezi celkovým vytížením linky a její kapacitou). Kromě toho musí uvažovaná množina sítí VLAN splňovat následující podmínky:

- *Topologie každé VLAN je spanning tree*
- *Všechen požadovaný provoz v dané VLAN je routovaný*
- *Celkové množství provozu procházejícího linkou nepřesáhne její kapacitu*

(30)

Z hlediska teorie složitosti jde o NP-těžký, tedy nedeterministicky polynomiální problém. Existuje takový nedeterministický algoritmus, který řeší problém v polynomiálním čase.

4.4.7 Verze STP

802.1D – původní verze předpokládá jedinou instanci STP pro celou síť nezávisle na počtu virtuálních sítí VLAN. Díky tomu jsou zde nároky na využívání CPU a operační paměti nižší než u ostatních verzí. Provoz pro všechny VLAN prochází přes stejnou trasu (je zde pouze jeden root bridge a jeden STP tree) a kvůli omezením standardu 802.1D je nevýhodou pomalá konvergence. Tato verze je často označována jako původní „STP“ verze.

PVST+ - jde o proprietární verzi protokolu vyvinutou společností Cisco. Vytváří zvláštní instanci STP pro každou VLAN v síti a podporuje funkce jako PortFast, UplinkFast, BackboneFast, bezpečnostní funkce BPDU guard, BPDU filter, root guard a loop guard. PVST+ umožňuje optimalizaci STP pro síťový provoz každé VLAN.

RSTP – poskytuje rychlejší, i když problémovou konvergenci než původní verze 802.1D. Pracuje stále s jedinou instancí STP. Nároky na CPU a paměť jsou mírně vyšší než u CST, avšak nižší než RSTP+.

Rapid PVST+ – proprietární protokol společnosti Cisco kombinuje výhody RSTP a PVST+. Pro každou VLAN vytváří separátní instanci a používá PortFast, BPDU guard, BPDU filter, root guard a loop guard. Má nejvyšší nároky na využití HW prostředků a problémy s konvergencí a neoptimálními trasami síťového provozu.

MSTP – standard 802.1s. Provádí mapování VLAN se stejnými požadavky na tok provozu a pro každou takovou skupinu vytváří zvláštní instanci STP. Tím optimalizuje využití prostředků.

MST – implementace MSTP protokolu společnosti Cisco podporuje až 16 instancí RSTP a kombinuje více VLAN se stejnou fyzickou i logickou topologií do stejné RSTP instance. Každá taková instance může používat PortFast, BPDU guard, apod. HW nároky jsou nižší než u Rapid PVST+, ale vyšší než u RSTP.

(31)

4.5 CDP

Cisco Discovery Protocol je proprietárním protokolem společnosti Cisco sloužící ke zjišťování informací o ostatních Cisco zařízeních a jejich připojení do sítě. CDP verze 2 je jeho nejnovějším vydáním a poskytuje inteligentní sledování zařízení.

Ve výchozím nastavení je CDP funkce povolena na každém podporovaném rozhraní Cisco zařízení (routery, switche). Fyzické médium musí podporovat zapouzdření SNAP.

Protokol funguje na spojové vrstvě OSI modelu a povoluje dva systémy, které podporují různé protokoly síťové vrstvy pro komunikaci.

4.5.1 Procesy CDP protokolu

„Cisco Discovery Protocol pracuje na bázi „hello" packetů. Všechna zařízení s běžícím CDP periodicky propagují své atributy směrem k sousedním uzlům prostřednictvím multicast adres. CDP pakety propagují také informaci o hodnotě „hold time" (udávaný v sekundách), indikující čas, po jehož vypršení je paket zahozen."

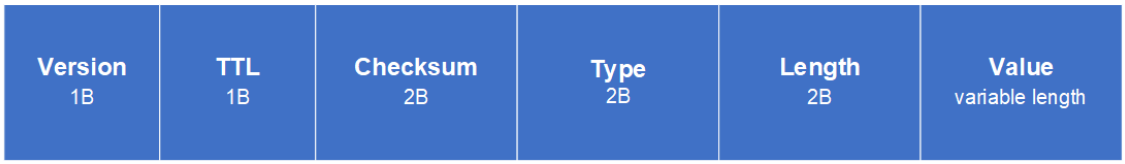
„Cisco zařízení odesílají CDP pakety s nenulovou hodnotou, v případě, že je rozhraní povoleno. Vypnutí síťového rozhraní znamená nulový „hold time". Odeslání CDP paketu s atributem „hold time" = 0 dovoluje danému zařízení rychle prohledat ztracené sousední uzly."

Pokud dojde od posledního odeslání CDP ke změně nějakého z atributů, zařízení uloží nové informace do cache a zahodí předchozí (neaktuální) informaci, přestože její hodnota „hold time" ještě nevypršela. (32)

4.5.2 Bezpečnost použití

CDP protokol lze vypnout vždy buď na konkrétním rozhraní či na celém zařízení. Z důvodu přenášení informací o síťových prvcích je doporučeno nepoužívat tuto funkci na rozhraních bez připojených Cisco zařízení a také tam, kde není prohledávání žádoucí – například rozhraní připojení k internetu. Blokaci lze provést pomocí SNMP protokolu. Cisco zařízení nikdy pakety CDP neroutují. (32)

4.5.3 Popis struktury CDP paketu



Obrázek 16 - Rámec CDP
Zdroj: vlastní tvorba

Version – udává verzi použitého CDP protokolu (hodnota je vždy 0x01)

TTL – zbývající čas (ve vteřinách), po který příjemce udržuje informace obsažené v paketu

Checksum – kontrolní součet IP paketu

Type – CDP typ atributu (Device ID, Address, Port ID,...)

Length – celková délka polí type, length a value (v bytech)

Value – možné hodnoty tohoto pole jsou popsány v tabulce níže (- Přehled atributů)

Atributy přenášené CDP protokolem

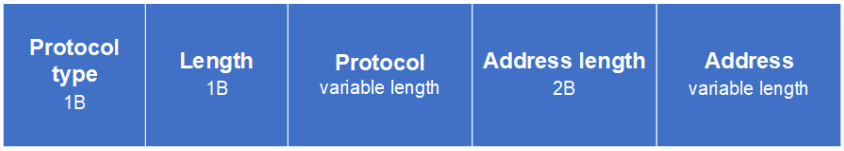
Název	Popis
Device ID	Název sousedního zařízení a jeho MAC adresa nebo sériové číslo
Local interface	Rozhraní spojené se zařízením
Hold time	Zbývající čas (v sekundách), po který bude CDP paket držen odesílajícím routerem
Capability	Kód typu připojeného zařízení
Platform	Produktové číslo zařízení
Port ID	Číslo portu na nalezeném sousedním zařízení
Address	Identifikace všech adres protokolu síťové vrstvy, které jsou nastaveny na daném rozhraní

Tabulka 5 - Přehled atributů protokolu CDP
Zdroj: (32)

Popis pole Address

„Pole typu/délky/hodnoty adresy obsahuje číslo, indikující, kolik adres je obsaženo v paketu, následované jedním záznamem pro každou propagovanou adresu. Ty jsou poprvé přiřazeny k rozhraním, na které jsou odesílány pakety CDP.“

Zařízení může propagovat všechny adresy pro danou sadu protokolů a volitelně také jednu či více IP adres typu loopback. Pokud má zařízení možnost správy SNMP protokolem, první záznam v polích typu/délky/hodnoty adresy je právě ta, kde zařízení přijímá SNMP zprávy. (33)



Obrázek 17 - Pole atributu Address v CDP paketu
Zdroj: vlastní tvorba

Protocol type – typ použitého protokolu (1=NLPID, 2=802.2)

Length – délka pole protokolu (u 802.2 závislá na použití SNAP)

Protocol – kód protokolu vyšší vrstvy (0xCC=IP)

Address length – délka pole adresy

Address – adresa rozhraní, resp. systémová adresa, pokud žádné rozhraní nemá adresu přiřazenou

5 Útoky ve spojové vrstvě

Tato kapitola popisuje nejčastější útoky cílené na protokoly spojové vrstvy. U každé podkapitoly je vysvětlen cíl útoku a také jeho průběh.

5.1 Útoky na ARP protokol

Jak již bylo zmíněno v teoretické kapitole věnované protokolu ARP, jde o bezestavový protokol, jenž využívá ke své funkci dočasné uchovávání všech informací, které přijme od ostatních, ARP cache.
Absence jakéhokoliv mechanismu pro ověření činí protokol ARP relativně nebezpečným a náchylným k různým útokům. Další problematickou částí je využívání broadcastu pro prvotní komunikaci. Popisovanými útoky jsou ARP spoofing a ARP cache poisoning.

5.1.1 Cíl útoků

Cílem je změna, resp. přidání zfalšovaného záznamu do ARP cache zařízení. Následně mohou útočníci přesměrovat provoz určený pro toto zařízení na svůj PC a získávat tak přístup k příchozím paketům.

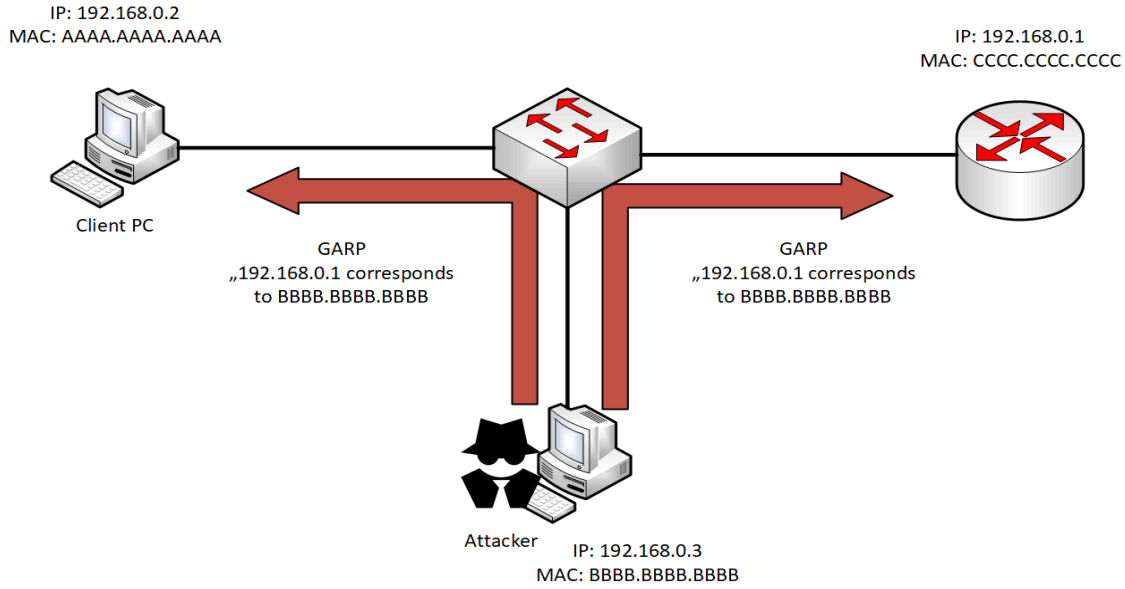
Útočníci obecně používají ARP spoofing pro krádež informací, modifikaci přenosu dat nebo zastavení síťového provozu v LAN. Usnadňuje také ostatní typy útoků jako například DDoS, session hijacking nebo MITM. Je proveditelný pouze v LAN sítích využívajících ARP protokol. (34)

5.1.2 Popis

ARP spoofing je proces falšování paketů ARP za účelem schopnosti impersonifikace jiné stanice v síti. Obvykle útočník odesílá oběti podvržený ARP response periodicky v časovém intervalu několikanásobně nižším, než je vypršení timeoutu ARP cache v konkrétním operačním systému oběti. Tím je zajištěno, že oběť nikdy nebude požadovat pro tuto stanici odeslat ARP request a tím zjistit její skutečnou MAC adresu. (35)

ARP cache poisoning je taktéž škodlivý proces mapování falešné IP adresy k již existující MAC adrese v ARP cache. K této manipulaci může dojít přímo v ARP cache napadeného PC nezávisle na odeslaných ARP zprávách tohoto PC. (36)

5.1.3 Průběh útoku ARP spoofing



Obrázek 18 - Model útoku na ARP cache klientského PC
Zdroj: Tomáš Bartoníček, zpracováno dle (37)

1) Stanice Client PC s IP adresou 192.168.0.2 (dále jen „Stanice“) potřebuje přistupovat do prostředí jiné sítě – využije tedy router 192.168.0.1 jako svoji výchozí bránu
2) Útočník u Attacker's PC (dále jen „Útočník“) periodicky odesílá do sítě zprávy typu ARP response s informací o své MAC adrese (BBBB.BBBB.BBBB) patřící IP adresám 192.168.0.1, resp. 192.168.0.2.
3) Tyto zfalšované informace se uloží do ARP cache zařízení.
4) Stanice začíná komunikovat s routerem - podle záznamu ve své ARP tabulce tedy odešle rámec do cíle s MAC adresou BBBB.BBBB.BBBB (útočník).
5) Pokud by router inicioval komunikaci se stanicí, bude výsledek stejný. Router podle záznamu ve své ARP tabulce odešle rámec do cíle s MAC adresou BBBB.BBBB.BBBB (útočník).
6) Falešné záznamy zůstanou v ARP tabulkách zařízení, dokud nevyprší limit ARP cache timeout.

5.2 MAC spoofing

Jedná se o cílenou softwarovou změnu MAC adresy síťového zařízení. Standardně je fyzická adresa přiřazena výrobcem, speciální softwarové nástroje však umožňují její změnu. Zařízení se poté v síti prezentuje jako jiný typ od odlišného výrobce.

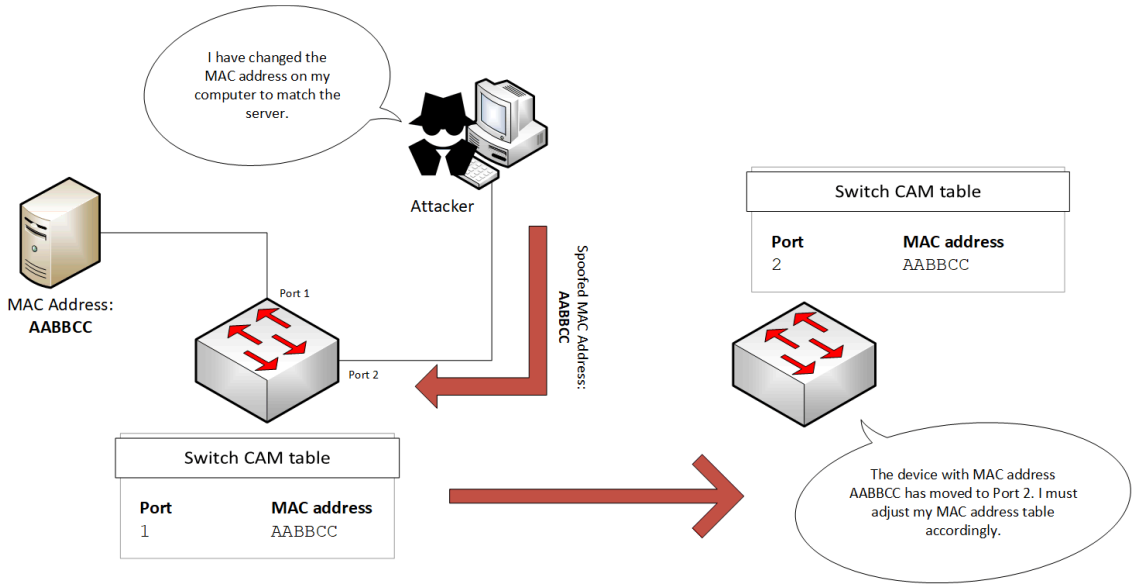
5.2.1 Cíl útoku

Často je používán za účelem obcházení ochrany omezení přístupu pouze pro určité MAC adresy (například ve Wi-Fi sítích), dále impersonifikace, přijímání a odesílání dat za cizí zařízení.

5.2.2 Popis

Odesláním jediného rámce obsahující zfalšovanou HW adresu v síti je přepsán záznam v CAM tabulce switche. Následně switch přeposílá data určená pro stanici k útočníkovi. (38)

5.2.3 Průběh útoku v bezdrátovém prostředí



Obrázek 19 - Model útoku MAC spoofing
Zdroj: Tomáš Bartoníček, zpracováno dle (39)

1) *Útočník provede skenování MAC adres klientů pomocí Wi-Fi*
2) *Změní svoji MAC adresu na adresu jednoho ze skenovaných klientů*
3) *Odesíláním De-Auth paketů začne blokovat bezdrátovou komunikaci klienta, jehož HW adresa byla použita*
4) *Útočník komunikuje s Wi-Fi AP se zfalšovanou MAC adresou*
5) *Falešnou adresu používá i pro interní autentizaci do sítě*
6) *Vytváří SoftAP falešnou Wi-Fi síť pomocí ostatních bezdrátových Wi-Fi adaptérů (virtuální router)*
7) *Další potencionální klienti, kteří se připojí skrze SoftAP již komunikují přes zařízení útočníka = riziko*

Zdroj: (40)
Z výše uvedeného vyplývá, že zabezpečení bezdrátové sítě pouze pomocí filtru MAC adres je značně nedostatečné a k síti se tedy mohou připojit i nechtění klienti.

5.2.4 MAC piggy-backing

„Jedním z míst, kde je útok MAC spoofing stále úspěšně využíván, jsou veřejně dostupné sítě. Útok MAC piggy-backing se používá především k obcházení přihlašovacích captive portálů. Útočník se nezkouší do sítě prolomit za účelem krádeže dat, ale raději využije tímto způsobem captive portálu pro získání volného přístupu k internetu zdarma. Řešení autentizace portálem jsou často jediným bezpečnostním prvkem při poskytování veřejného přístupu k internetu.“

Pokud je stanice připojena k hotspotu a získá IP adresu, uživatel je v rámci webového prohlížeče automaticky přesměrován k přihlášení. Jakmile je po vyplnění určitých údajů ověřen, access point povolí přístup k internetu zařazením MAC adresy klienta na seznam povolených. Tím proces autentizace končí. Útočník zachytí pakety v bezdrátové síti (používá se sdílené vysílání CSMA/CA) ke zjištění MAC adres úspěšně ověřených stanic a některou z nich nastaví na své síťové kartě. Nemluvíme tedy přímo o napadení či krádeži dat, lze však

uvažovat například o zneužití cizí veřejné IP adresy pro provozování nelegálních aktivit, krádež služby placeného přístupu atd. (41)

5.3 CAM table overflow

Princip tohoto útoku spočívá v přepnutí funkce switche na funkci hubu z důvodu přeplnění CAM tabulky a tím i zaplnění operační paměti prvku.

5.3.1 Cíl útoku

Útočník může vidět všechen síťový provoz odeslaný ze stanice oběti směrem k jinému PC v lokální síti bez záznamu v CAM tabulce switche.

Pokud dojde k zastavení zaplavování nevalidními zdrojovými MAC adresami, začnou po vypršení „aging time" mizet také tyto falešné adresy, switch začne opět plnit CAM tabulku legitimními záznamy. (42)

5.3.2 Popis

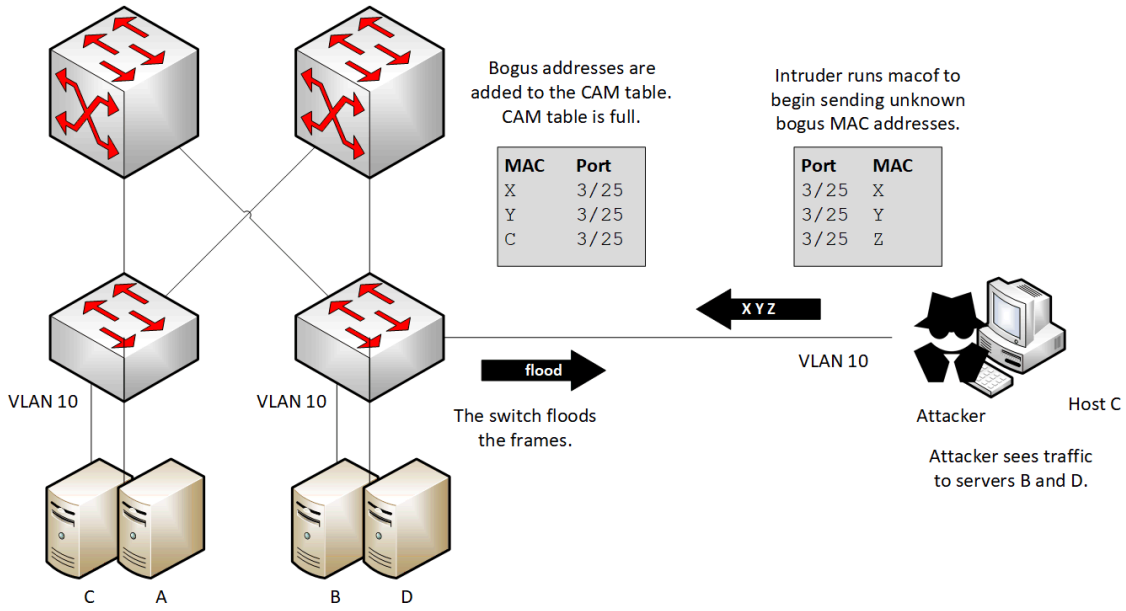
Útok je založen na omezené kapacitě CAM tabulky. Pokud je tedy přidáno dostatečné množství záznamů předtím než ostatní záznamy expirují, tabulka se zaplní a nepřijímá žádné nové záznamy.

„Při útoku CAM table overflow zaplaví útočník switch velkým množstvím nevalidních zdrojových MAC adres, dokud není CAM tabulka plná. Pokud k této situaci dojde, switch začne zaplavovat veškerým příchozím provozem všechny porty, protože již nemá v CAM tabulce prostor pro učení nových legitimních MAC adres. Switch tedy pak jedná stejně jako hub."

*Pro praktické použití je vhodný například software **macof**, jenž zaplaví switch pakety obsahující náhodně vygenerovanou zdrojovou a cílovou MAC adresu a IP adresu. Po dobu, kdy program macof zůstává spuštěn, zůstává plná i CAM tabulka, tím pádem dochází k rozesílání přijatých rámců na všechny porty switche.*

(38)

5.3.3 Průběh útoku



Obrázek 20 - Model útoku CAM table overflow
Zdroj: Tomáš Bartoníček, zpracováno dle (42)

1) Útočník po přístupu do sítě spustí generátor vysokého množství rámců s podvrženými zdrojovými MAC adresami
2) Falešné adresy se postupně přidávají do CAM tabulky switche
3) Jakmile je tabulka zcela zaplněná, switch začne předávat příchozí provoz na všechny své porty
4) Útočník může zachytávat síťový provoz určený pro ostatní zařízení v síti

5.4 Útoky na DHCP protokol

Vzhledem k funkci automatického přidělování TCP/IP nastavení síťovým klientům je protokol DHCP důležitý především po stránce bezpečnosti. Bohužel během procesu přidělování nastavení se ve výchozím stavu žádná autentizace klientů neprovádí.
Popisovaný útok DHCP starvation je charakterem DDoS – nově připojená zařízení nebudou mít k dispozici funkční připojení k síti z důvodu vyřazení DHCP serveru z provozu. Na tento proces navazuje také DHCP spoofing, jenž principem odpovídá kategorii útoků MITM.

5.4.1 Cíl útoku

Útočníci se snaží vyřadit z provozu legitimní DHCP server infrastruktury. Cílem může být zabránění klientům v připojení nebo umožnění běhu záškodnického DHCP serveru, jenž bude přidělovat TCP/IP nastavení výhodná pro útočníka (upravená IP adresa výchozí brány, DNS server…).

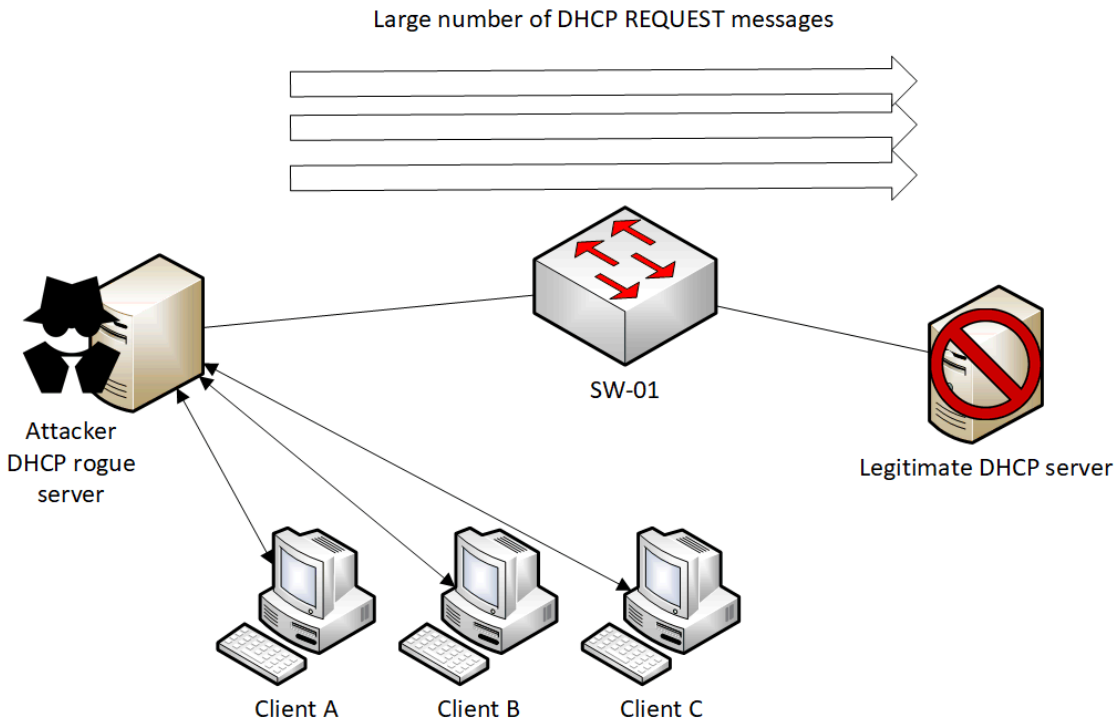
5.4.2 Popis

Princip útoku DHCP starvation spočívá ve vyčerpání všech volných IP adres v DHCP poolu.

„Každý DHCP request s jinou MAC adresou je DHCP serverem považován za request nového síťového klienta. Pokud útočník odešle dostatečný počet požadavků, DHCP pool serveru bude pokaždé vyčerpán. Proto budou noví síťoví klienti DHCP službou odmítnuti a z toho důvodu nebudou schopni se k síti připojit.“

„Kromě toho autoři poukazují, že zprávy DHCPDECLINE (oznamuje použití stejné IP adresy jiným klientem) mohou být použity k implementaci útoku DHCP starvation, přesněji k oklamání DHCP serveru falešnou informací o použití stejné IP adresy jiným klientem.“

DHCP starvation často slouží jako přípravný krok před nasazením cizího DHCP serveru (útok DHCP spoofing). (43)



Obrázek 21 - Model útoků DHCP starvation a DHCP spoofing
Zdroj: Tomáš Bartoníček, zpracováno dle (44)

Po jeho zapnutí v síti může útočník přesměrovat část síťového provozu na svůj připravený server/stanici. Nejčastěji jde o IP adresu výchozí brány (přeposílání veškerého provozu z koncového zařízení) či adresu DNS serveru (přeposílání DNS dotazů).

Pro účely penetračního testování je při simulaci popisovaných útoků často používán nástroj **Yersinia** (využitý také v praktické části této práce). (44)

Indukovaná verze DHCP starvation

V porovnání s klasickou verzí je tento typ útoku méně náročný na počet zaslaných zpráv a čerpání zdrojů ze strany útočníka.

Pro popis uvažujme síťový model s jednou uživatelskou stanicí, DHCP serverem a stanicí útočníka. Průběh indukované verze DHCP starvation je následující:

1) *Uživatelská stanice po zapnutí odešle pomocí všesměrového vysílání žádost DHCP discover pro zjištění dostupných DHCP serverů v síti*

2) *Server zašle stanici DHCP offer s nabídkou volné IP adresy*

3) *Stanice odesílá DHCP request, žádá tedy o nabídnutou IP adresu. Tato zpráva obsahující zdrojovou MAC adresu stanice je odeslána broadcastem, přijme ji tedy také útočník a uloží si MAC adresu stanice*

4) *Server zasílá stanici DHCP acknowledgement – potvrzení o přidělení dohodnuté IP adresy*

5) *Stanice s IP adresou odešle broadcastem ARP request pro zjištění, zda některé zařízení ve stejné síti nepoužívá přidělenou IP adresu. Zdrojová adresa ARP requestu je 0.0.0.0*

6) *Odpověď ARP reply od útočníka. Zdrojová IP adresa je přidělená adresa pro stanici, cílová pak 0.0.0.0.*

7) *Odeslání DHCP decline stanicí – jakmile klient přijme odpověď ARP reply se známkou podvrhnuté zprávy, předpokládá, některá jiná zařízení v síti. Následně zašle uživatelská stanice informaci DHCP serveru (DHCP decline) o odmítnutí využívání přidělené IP adresy z důvodu prevence konfliktu adres.*

Jakmile DHCP server přijme zprávu DHCP decline, označí navrácenou IP adresu jako nedostupnou po nastavený časový interval lease-time. Opakováním výše uvedeného postupu je nakonec DHCP pool vyčerpán a nelze dále přidělovat IP adresy nově připojeným zařízením. (45)

5.5 VLAN hopping

Virtuální síť VLAN slouží pro logické členění L2 sítě do jednotlivých segmentů (viz kapitola 802.1Q) a tím i izolaci některých zařízení od jiných.
Jak již název napovídá VLAN hopping je útokem, spočívající v možnosti přecházení mezi těmito virtuálními sítěmi, což může mít řádu následků.

5.5.1 Cíl útoku

Útočníci se snaží docílit získání přístupu k síťovému provozu v jiné VLAN, která standardně není přístupná. Toho lze využít ke kompromitaci a ovládnutí dalších zařízení v důvěryhodné VLAN s dostupností administračních rozhraní síťových prvků (tzv. Management VLAN) nebo serverů či stanic s důvěrnými daty atd.

5.5.2 Popis

„VLAN hopping dovoluje provozu z jedné VLAN nahlížet do jiné bez nutnosti průchodu přes router. Za jistých okolností mohou útočníci zachytávat pakety a z nich poté získávat hesla nebo jiná citlivá data. Velmi často využívají nesprávně nastaveného portu typu trunk. Ve výchozím stavu má přístup do všech VLAN sítí a lze tedy skrze něj rozesílat provoz i mezi switchi.“ (42)

5.5.3 Rozdělení druhů VLAN hoppingu

1) **Spoofing zpráv DTP** – útočník může odesílat již tagované zprávy s VLAN ID cílové sítě
2) **Zprovoznění falešného switche** – útočník může přistupovat do všech VLAN na switchi oběti

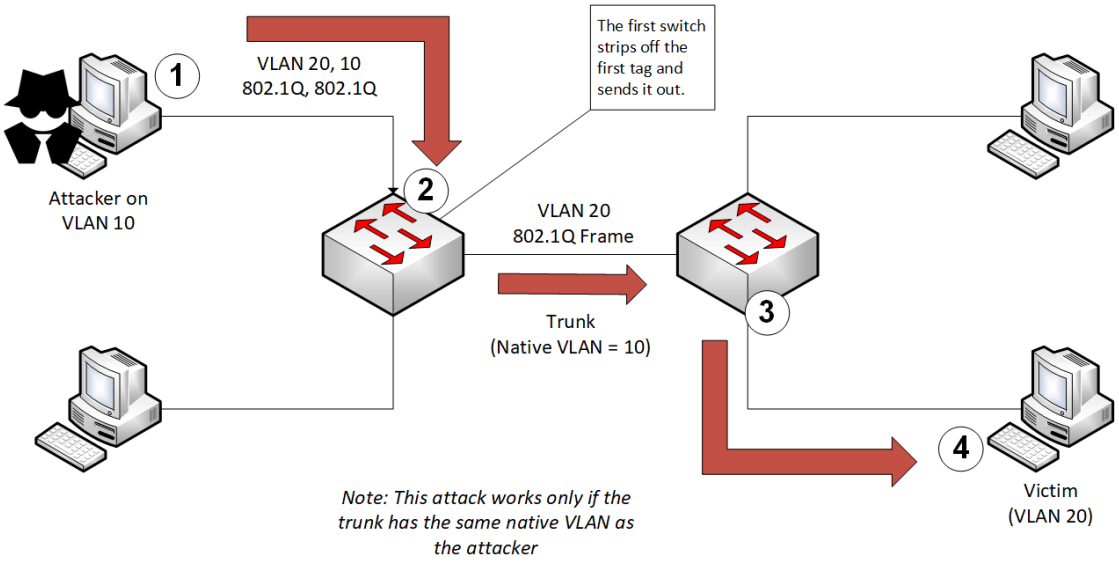
5.5.4 Double-tagging (double-encapsulated)

Tento způsob zneužívá vlastnosti jednoúrovňového zapouzdření 802.1Q u většiny modelů switchů. Dovoluje tak útočníkům vložit další, skrytý 802.1Q tag dovnitř rámce a tím tento rámec nasměrovat do skryté VLAN. Double-tagging útok je funkční dokonce i bez použití trunk portů. (42)

5.5.5 Rogue trunk

Toto je nejjednodušší forma VLAN hopping útoku. Využívá vlastnost automatické konfigurace trunk portů na většině switchů (protokol DTP). Stanice útočníka musí mít k dispozici funkci spoofingu, být schopna emulovat 802.1Q nebo ISL trunk a odesílat komunikační zprávy DTP protokolu.
Útočník poté vysílá signály DTP směrem ke switchi. Následkem toho switch v domnění, že komunikaci iniciuje jiný switch, který žádá o vytvoření trunk propoje, provede přidání VLAN na daný port.
Pro úspěšné dokončení je nezbytné mít nakonfigurovaný režim „auto“ v nastavení DTP protokolu. Následně je útočník již členem všech VLAN sítí provozovaných na trunk portech switche a stačí pouze provést další krok do některé z těchto VLAN.

Postup útoku typu Double-tagging



Obrázek 22 - Model útoku VLAN hopping
Zdroj: Tomáš Bartoníček, zpracováno dle (42)

1) *Útočník odešle dvojitě tagovaný rámec 802.1Q na switch. Vnější tag (zde VLAN 10) pochází od útočníka a odpovídá hodnotě nativní VLAN na trunk portu.*

2) *Switch odešle rámec do VLAN 10. Žádné další značkování není provedeno, protože se jedná o nativní VLAN.*

3) *Rámec dorazí na druhý switch, který však nemá informaci o svém zařazení do VLAN 10.*

4) *Druhý switch zjistí informace z 802.1Q tagu, zjistí, že rámec je určen pro VLAN 20. Odešle jej tedy na konkrétní port odpovídající stanici.*

Je třeba zmínit, že výše popsaný útok je jednosměrný a funguje pouze v případě, že zařízení útočníka a trunk port mají stejnou nativní VLAN. Zabránění typu double-tagging je složitější než zastavení běžných VLAN hopping útoků. Pro trunk porty switche se doporučuje používat nativní VLAN odlišnou od VLAN na portech pro koncová zařízení. (42)

5.6 STP root bridge change

Automatická správa redundantních spojení v síti je podmíněna vytvořením STP topology tree – stromové topologie sítě při spuštění zařízení, která podporují tento protokol. Po stanovení hlavního switche (root bridge) je však nutné zajistit, aby případný útočník nemohl tuto roli převzít.

5.6.1 Cíl útoku

Primárním cílem je převzetí role root bridge v instanci STP protokolu z důvodu ovlivnění chování síťové infrastruktury. Takto tedy lze například přesměrovat veškerý provoz přes cizí switch nebo vyloučit určitý prvek z topologie.

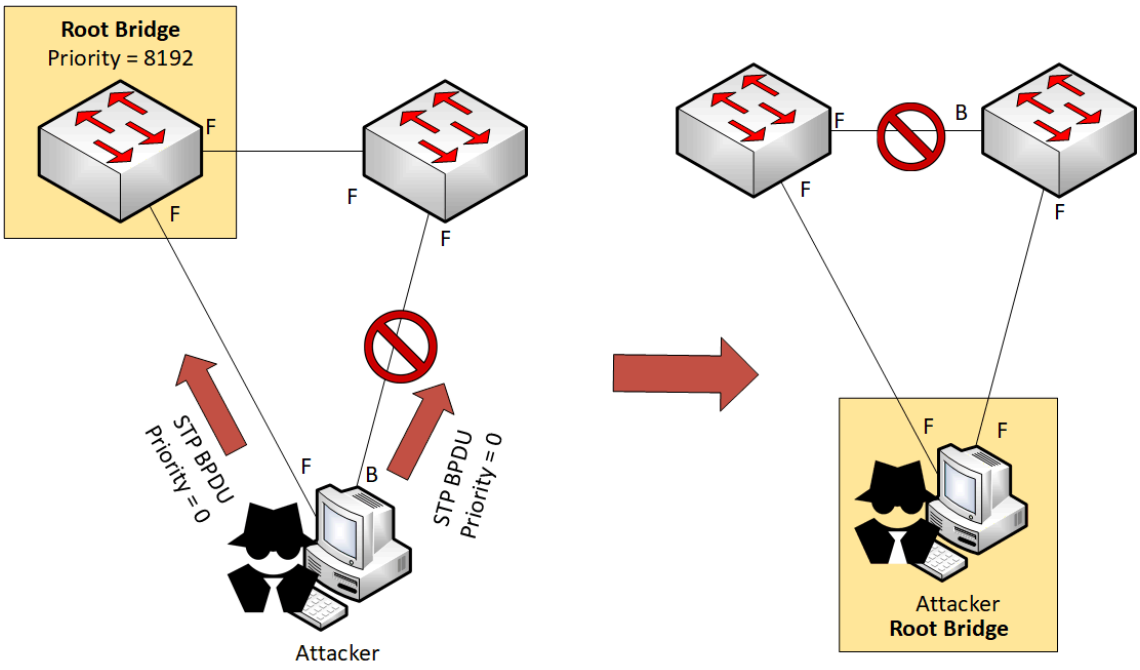
5.6.2 Popis

Útoky na STP protokol cílí na prostředí beze smyček, kde se switche dorozumívají pomocí BPDU zpráv, na kterých je protokol založen. Jsou tedy využívány během volby STP root bridge a následně k rozhodování, jaké porty switche zůstanou ve stavu forwarding a jaké budou zablokovány (stav blocked). Zprávy BPDU jsou také nezbytné při náhlých změnách v infrastruktuře a následném sestavení nové bezsmyčkové topologie s užitím redundantních spojů.

Změnu způsobí problémy se spojením mezi prvky, stejně jako zařazení nového switche, jenž má vyšší hodnotu bridge priority (tzv. ***superior BPDU****), než dosavadní root bridge a z toho důvodu tedy tuto roli přebírá.*

„Pokud útočník zavede do přepínané sítě cizí switch a tento switch odešle superior BPDU, pak dojde k převzetí role root bridge útočníkem. Topologie přepínané sítě závisí na pozici root bridge a relativní pozici ostatních switchů k root bridge, díky tomu změna topologie útočníkem nebude mít vliv na výkon sítě, avšak může všechen provoz přesměrovat přes switch útočníka. (46)

5.6.3 Průběh útoku



5.6.4

Obrázek 23 - Model útoku STP root bridge change
Zdroj: Tomáš Bartoníček, zpracováno dle (46)

1) Útočník zapojí do sítě vlastní switch s podporou STP protokolu
2) Po zjištění změny v topologii je prováděno opětovné sestavení STP tree, všechny switche vysílají BPDU zprávy, probíhá volba root bridge
3) Útočníkův switch vysílá BPDU s hodnotou priority 0, tím vyhrává volbu a stává se novým root bridgem

5.7 Další útoky na STP protokol

Kromě uvedeného STP root bridge change útoku je na Spanning Tree protokol cílena i řada dalších, o kterých je nutné při návrhu zabezpečení sítě uvažovat.

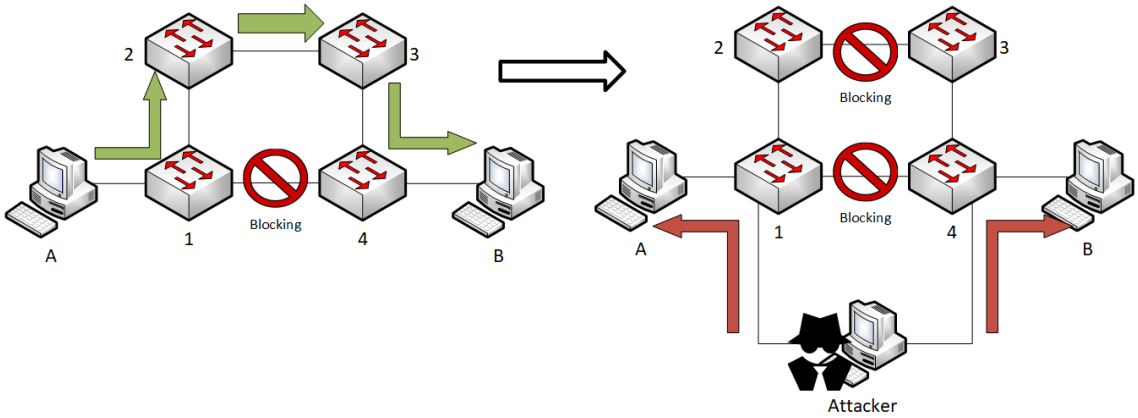
5.7.1 Flood of Config BPDUs

„Jak je doporučeno specifikací 802.1w, STP není určen pro odbavování tisíce příchozích BPDU zpráv. Zařízení se snaží zpracovat tohoto provozu co možná nejvíce, dokud není vyčerpána jeho výpočetní kapacita. Následkem je vysoké vytížení CPU a extrémně vysoký nárůst počtu přijatých BPDU zpráv na jednom portu.“

5.7.2 Simulating Dual-homed switch

Jedná se o útok přesměrování síťového provozu přes zařízení útočníka. Ten se v síti nejprve připojí mezi dva switche, převezme roli root, vytvoří novou topologii a následně přes ni vynutí průchod veškerého provozu. Může také vynutit na switchích vyjednávání o vytvoření trunk portu pro umožnění zachytávání provozu z více sítí VLAN.

Zdroj: (47)



Obrázek 24 - Model útoku Simulating Dual-Homed Switch
Zdroj: Tomáš Bartoníček, zpracováno dle (47)

6 Ochrana proti síťovým útokům na L2

Proti výše zmíněným útokům lze síťovou infrastrukturu bránit pomocí bezpečnostních technologií, jimiž jsou switche vybaveny. Ty dokáží útokům buď zabránit (prevence) nebo alespoň snížit jejich následky.
Pořadí následujících podkapitol koresponduje s pořadím jednotlivých síťových útoků popsaných v kapitole předchozí.
Z důvodu odlišných názvů bezpečnostních funkcí u každého výrobce síťových prvků jsou v této práci používána pojmenování společnosti Cisco Systems, Inc. Stejně tak ukázky nastavení pomocí příkazů budou přizpůsobeny pro switche tohoto výrobce.

6.1 Dynamic ARP inspection

6.1.1 Účel technologie

Hlavním účelem této nekryptografické funkce je zamezení útokům typu ARP spoofing a ARP cache poisoning. *Dokáže také detekovat a zabránit DoS útokům, které by způsobeným zahlcením vysokého množství ARP rámců (tzv. rate-limiting). Pokud jejich počet překročí určený limit za jednotku času, fyzický port se přepne do stavu error-disabled.* (48)

Nevýhodou jsou však vysoké náklady na údržbu a také nemožnost nasazení na bezdrátových sítích, kde se MAC adresa koncových zařízení může přesouvat mezi fyzickými porty (Wireless AP roaming).(49)

6.1.2 Princip

„Dynamic ARP inspection analyzuje v reálném čase ARP pakety, zahazuje nevalidní nebo podvržené. Taková analýza používá databázi validních IP-MAC vazeb, která může být sestavena buď manuálně správcem sítě nebo dynamicky pomocí funkce DHCP snooping.“ (49, 50)

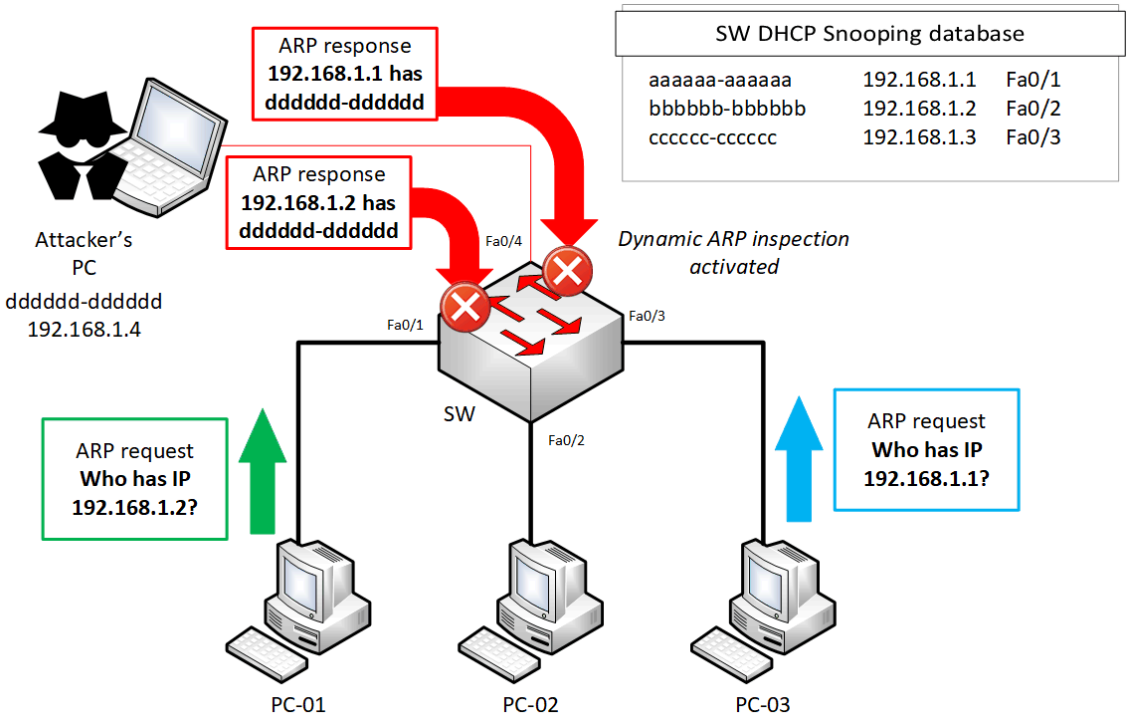
Po připojení nového koncového zařízení switch detekuje DHCP Acknowledgement pakety od DHCP serveru směrem ke klientovi a ty si přidává do své DHCP snooping databáze. Díky tomu má switch informace o MAC adresách i IP adresách těchto zařízení na každém svém portu a může tedy provádět validaci nejen MAC adres v těle ARP rámců, ale také IP adres.
DAI funguje zvlášť pro každou VLAN.

6.1.3 Nastavení

Při návrhu konfigurace funkce Dynamic ARP inspection je nutné promyslet, na kterých fyzických portech budou připojeny další síťové prvky infrastruktury.

*Technologie používá dva stavy portů – **trusted** (tyto porty jsou z procesu inspekce vyloučeny) a **untrusted** (zde se kontrolují všechny ARP rámce).*

Doporučuje se ochranu aktivovat pouze na přístupových portech s koncovými zařízeními. Dále je nutné do ACL seznamu doplnit zařízení se statickou IP adresou.(48)



Obrázek 25 - Ukázka funkce Dynamic ARP inspection
Zdroj: Tomáš Bartoníček, zpracováno dle (51)

Konfigurace v prostředí CLI

Zapnutí ARP inspekce pro konkrétní VLAN

```
SW(config)# ip arp inspection vlan <vlan_id>
```

Definování parametrů, které budou kontrolovány ARP inspekcí

```
SW(config)# ip arp inspection validate [src-mac, dst-mac, ip]
```

Definice výjimky pro ARP pakety konkrétního zařízení

```
SW(config)# permit ip host <sender-ip> mac host <sender-mac> log
```

Vyloučení konkrétního rozhraní z ARP inspekce

```
SW(config)# ip arp inspection trust <interface>
```

Zdroj: (52)

6.2 Port-security

Ve výchozím nastavení sítě lze získat přístup pouhým zapojením zařízení do zásuvky ve zdi, což je samo o sobě rizikem, pokud uvažujeme, že se ve stejném síťovém segmentu nacházejí důležité prvky jako jsou servery, datová úložiště nebo uložené zálohy.
Pro zabezpečení samotného přístupu při zapojení do sítě slouží funkce Port-Security.

6.2.1 Účel technologie

Technologie je vhodná především pro nasazení v interních korporátních sítích. Slouží k zamezení útokům typu CAM table overflow.

„Pomáhá v regulaci síťového provozu ve větších sítích během špiček. Port-Security je aplikovatelná na switche u zdrojů provozu a povoluje pouze konkrétní počet MAC adres pro vstup do sítě. Blokuje vstup dat na všech typech Ethernet portu, jestliže se MAC adresa zařízení, pokoušejícího se k síti připojit, liší od povoleného seznamu adres, definovaných pro konkrétní port.“ (53)

6.2.2 Princip

To znamená, že přístup je umožněn pouze autorizovaným zařízením. V případě pokusu nepovolené stanice o přístup switch buď začne tento síťový provoz zahazovat nebo celý fyzický port vypne.

Port-security může být nastavována automaticky nebo manuálně, definicí maximálního povoleného počtu MAC adres pro určitý časový interval. Jakmile je dosaženo limitu, port-security vykoná předem nastavenou akci při porušení podmínek (tzv. security-violation action). (28, 53)

Lze také implementovat omezení na každý port v podobě staticky konfigurované MAC adresy nebo dynamicky naučené adresy. Tímto krokem zúžíme použití portu switche pouze na definované zařízení. Pokud tak učiníme, nebude port předávat příchozí provoz, pokud je jeho zdrojová adresa mimo seznam povolených.

6.2.3 Nastavení

Stejně, jako v případě Dynamic ARP inspection je i zde vhodné aktivovat ochranu pouze na portech pro koncová zařízení (access porty).

Jakmile definujeme maximální počet MAC adres na portu, lze tyto adresy vkládat do tabulky switche několika způsoby:

- *Manuální konfigurací všech MAC adres na konkrétním rozhraní použitím příkazu switchport port-security mac-address macAddr*
- *Povolením dynamického vložení adres připojených zařízení k portu (tzv. „sticky learning")*
- *Manuální konfigurací počtu adres a povolením dynamického vložení*

V případě dynamického doplňování adres zůstává seznam adres po startu switche prázdný, dokud není detekován příchozí provoz.

Dojde-li k překročení maximálního počtu MAC adres na portu, případně je přijat rámec s nepovolenou zdrojovou adresou, switch provede jednu z následujících akcí:

- ***Protect** – pakety s neautorizovanou zdrojovou MAC adresou jsou zahazovány, dokud není snížen počet MAC adres na portu*
- ***Restrict** – pakety s neautorizovanou zdrojovou MAC adresou jsou zahazovány, dokud není snížen počet MAC adres na portu. Zároveň je inkrementována hodnota security-violation counteru.*
- ***Shutdown** – okamžitě přepne rozhraní do stavu error-disabled a odešle SNMP trap o bezpečnostním incidentu na portu*

Zdroj: (28)

Konfigurace v prostředí CLI

Zapnutí funkce Port-Security na konkrétním rozhraní

```
SW(config-if)# switchport port-security
```

Definování maximálního povoleného počtu MAC adres na jednom rozhraní

```
SW(config-if)# switchport port-security maximum <mac_limit>
```

Definování akce při přijetí rámce s nepovolenou MAC adresou/překročení limitu

```
SW(config-if)# switchport port-security violation <action>
```

Switch si uloží všechny aktuální adresy na portu do seznamu povolených MAC adresu (režim „sticky learning “)

```
SW(config-if)# switchport port-security mac-address sticky
```

Povolení konkrétní MAC adresy

```
SW(config-if)# switchport port-security mac-address <mac_addr>
```

Definování času (v sekundách) pro “stárnutí” MAC adres v seznamu

```
SW(config-if)# switchport port-security aging time <time_value>
```

Zdroj: (42)

6.3 DHCP snooping

6.3.1 Účel technologie

Existuje několik způsobů obrany proti útokům DHCP Starvation se zapojením cizího serveru. Jedním z nich je DHCP snooping. Účelem je zabránit, aby se případný útočníkův DHCP server mohl účastnit legitimních procesů při získávání TCP/IP konfigurace pro koncová zařízení.
Pro eliminaci vyčerpání DHCP poolu IP adres legitimního serveru slouží limit maximálního počtu vazeb (bindings) pro každý port. Stejně jako Dynamic ARP inspection i DHCP snooping usnadňuje obranu rate-limiting proti DoS útoku zaplavení DHCP requesty.

6.3.2 Princip

DHCP snooping provádí filtrování nedůvěryhodných DHCP požadavků na základě sestavování tabulky vazeb obsahující informace o MAC adrese, portu switche, VLAN, přidělené IP adrese a času lease-time. Těchto dat využívá také funkce Dynamic ARP inspection při validaci obsahu ARP rámců.

Jednotlivé porty jsou označeny jako důvěryhodné (trusted) nebo nedůvěryhodné (untrusted). Všechny typy DHCP zpráv mohou přicházet pouze na důvěryhodné porty, kde je umístěn například DHCP server. Na nedůvěryhodných portech jsou akceptovány pouze zprávy typu DHCP discover nebo DHCP request, všechny ostatní jsou zahazovány. (54)

DHCP snooping také nabízí doplňkovou funkci Option-82. Pokud je zapnutá, subscriber je identifikován MAC adresou a portem switche, kterým jej připojen do sítě.

6.3.3 Nastavení

Jak již bylo zmíněno v předchozí podkapitole, před vlastním nasazením technologie je nezbytné určit, ke kterým portům jsou připojeny DHCP servery a budou tedy nastaveny jako důvěryhodné. Namísto portu je možné definovat IP adresy legitimních DHCP serverů.

„Doplňkovou funkcí k DHCP Snoopingu je IP Source Guard. Ten skládá informace přijaté z DHCP offer paketů a pro zvýšení bezpečnosti efektivně sestavuje dynamický ACL list pro každý fyzický port pro povolení síťového provozu pouze z konkrétních zdrojů. Adresy těchto zdrojových zařízení jsou předem získány z DHCP offers.“ (55, 56)

Na nedůvěryhodných portech používá DHCP snooping následující rozhodovací logiku:

1) *filtruje všechny zprávy odeslané DHCP serverem,*
2) *switch kontroluje zprávy typu DHCP release a DHCP decline oproti tabulce vazeb. Pokud zde není IP adresa těchto zpráv zařazena, zpráva se zahazuje,*
3) *volitelně může být porovnávána MAC adresa klienta ve zprávě DHCP request se zdrojovou MAC adresou uvnitř Ethernetového rámce.*

První krok algoritmu chrání síť proti MITM útoku DHCP serveru útočníka, druhý krok brání záškodníkovi v převzetí již přidělené IP adresy legitimní stanice, následnému pokusu o opětovnou žádost o adresu a získání stejné IP adresy z DHCP serveru v síti. Tím by útočník převzal všechna existující spojení vytvořené původní stanicí. Posledním krokem je pak obrana před DoS útokem, tedy pokusem o alokaci všech IP adres z DHCP poolu. (56)

Konfigurace v prostředí CLI

Globální zapnutí funkce DHCP snooping

```
SW(config)# ip dhcp snooping
```

Zapnutí funkce pro konkrétní VLAN

```
SW(config)# ip dhcp snooping VLAN <VLAN_ID>
```

Označení konkrétního rozhraní za důvěryhodné

```
SW(config-if)# ip dhcp snooping trust
```

Zapnutí DHCP option 82 (pakety DHCP request budou obsahovat informaci o tom, z jakého rozhraní switche byly odeslány)

```
SW(config-if)# ip dhcp snooping information option
```

Definování maximálního počtu přijatých DHCP paketů za vteřinu na konkrétním rozhraní

```
SW(config-if)# ip dhcp snooping limit rate <max_limit>
```

Zdroj: (57)

6.4 Bezpečnostní funkce STP protokolu

Pro efektivní udržování STP topologie by mělo být umístění root bridge předvídatelné. Hlavní výkonný switch by měl plnit tuto roli, další stanovený switch pak roli secondary root. Pokud však potencionální útočník zapojí do sítě nový neznámý switch se záměrně nastavenou prioritou na nulovou hodnotu, logicky převezme roli root bridge.(58)

Ochranu před touto situací nabízejí funkce STP Root Guard a STP BPDU Guard.

6.4.1 STP Root Guard

„Switche vždy očekávají viditelnost root bridge na svém root portu a alternativním portu, protože se jedná o optimální trasu.“

Předpokládáme, že každý další switch je do sítě zapojen s hodnotou bridge priority, která je nižší než priorita aktuálního root bridge. Nový switch se následně stává root bridgem a STP topologie se rekonverguje do nového tvaru.

„Toto chování je normální, neboť switch s nejnižším BID vždy vyhraje volbu root bridge, není však vždy vhodné pro administrátory sítě – nově vzniklá STP topologie může být sestavena nesprávně (nechtěně), navíc každý proces rekonvergence znamená výpadek sítě.“ (58)

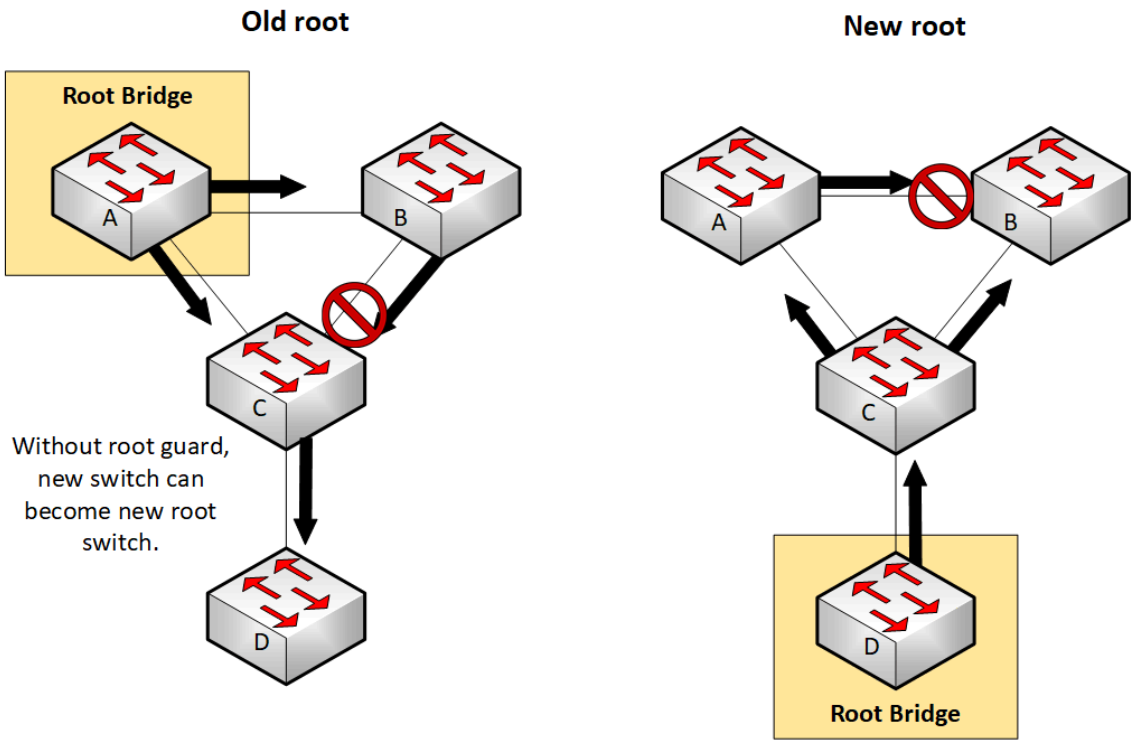
Účel technologie

Root Guard zabraňuje nechtěnému převzetí role root bridge v STP topologii. Nově připojený switch se nestane root bridgem ani při nastavené hodnotě **BID = 0**.

Princip

Pro zamezení převzetí role root bridge jinými switchi připojenými do stávající infrastruktury slouží funkce Root Guard.

Je-li tato technologie zapnuta na portu a po přepočítání STP tree tento port získá roli „root port“, přepne jej Root Guard do stavu „root inconsistent“, což je stav ekvivalentní ke stavu blokace. Jakmile je na tomto portu opět přijata zpráva BPDU s vyšším identifikátorem BID, switch jej obnoví z chybového stavu a port je znovu aktivní. (28)



Obrázek 26 - Model sítě s vypnutou funkcí Root Guard
Zdroj: Tomáš Bartoníček, zpracováno dle (57)

Nastavení

Doporučuje se zapínat funkci Root Guard na portech pro koncová zařízení (access porty), zde se nepředpokládá připojení jiných síťových prvků, které by měly převzít roli root bridge.

Konfigurace v prostředí CLI

Zapnutí funkce Root Guard na konkrétním rozhraní

```
SW(config-if)# spanning-tree guard root
```

Zdroj: (59)

6.4.2 STP BPDU Guard

Účel technologie

Další z bezpečnostních funkcí STP protokolu zabraňuje smyčkám deaktivací koncového (access) portu, kde bylo zjištěno nežádoucí zařízení, například switch útočníka.

Princip

Jakmile switch na portu s aktivovanou funkcí BPDU Guard zaznamená jakoukoliv BPDU zprávu, přepne okamžitě port do stavu Err-Disabled. Port zůstává vypnutý tak dlouho, dokud jej administrátor manuálně nezapne nebo nevyprší časový interval pro obnovu z chybového stavu. (58)

Nastavení

„Ve výchozím stavu je tato bezpečnostní funkce vypnutá na všech portech switche. Lze ji konfigurovat globálně a automaticky ji nechat zapnout na rozhraních s aktivovanou funkcí PortFast.“

To je doporučeno zejména z důvodu prevence před nechtěným či nežádoucím zapojením cizího switche do interní sítě.

Typická aplikace BPDU Guard je na access portech pro koncová uživatelská zařízení, kde není očekáván příjem BPDU zpráv.

„Na uplink spojích prvků infrastruktury s root bridgem by se naopak neměla nikdy zapínat. Má-li switch více takových uplink spojů, musí na nich být schopen přijímat BPDU odeslané z root bridge, a to i v případě, že tyto porty jsou ve stavu Blocking. Jakmile by zde byl BPDU Guard povolen, následovala by téměř okamžitě blokace portu a tím i výpadek důležité propojovací linky.“ (58)

Konfigurace v prostředí CLI

Zapnutí BPDU Guard pro všechna rozhraní switche se zapnutou funkcí PortFast

```
SW(config)# spanning-tree portfast bpduguard default
```

Zapnutí funkce pro konkrétní rozhraní

```
SW(config-if)# spanning-tree bpduguard enable
```

Zdroj: (59)

7 Vývoj vlastního řešení

Praktickou část této diplomové práce tvoří aplikace **netSecurityAnalyser,** která komplexně ověřuje zabezpečení konkrétní sítě a přítomnost bezpečnostních funkcí na L2.

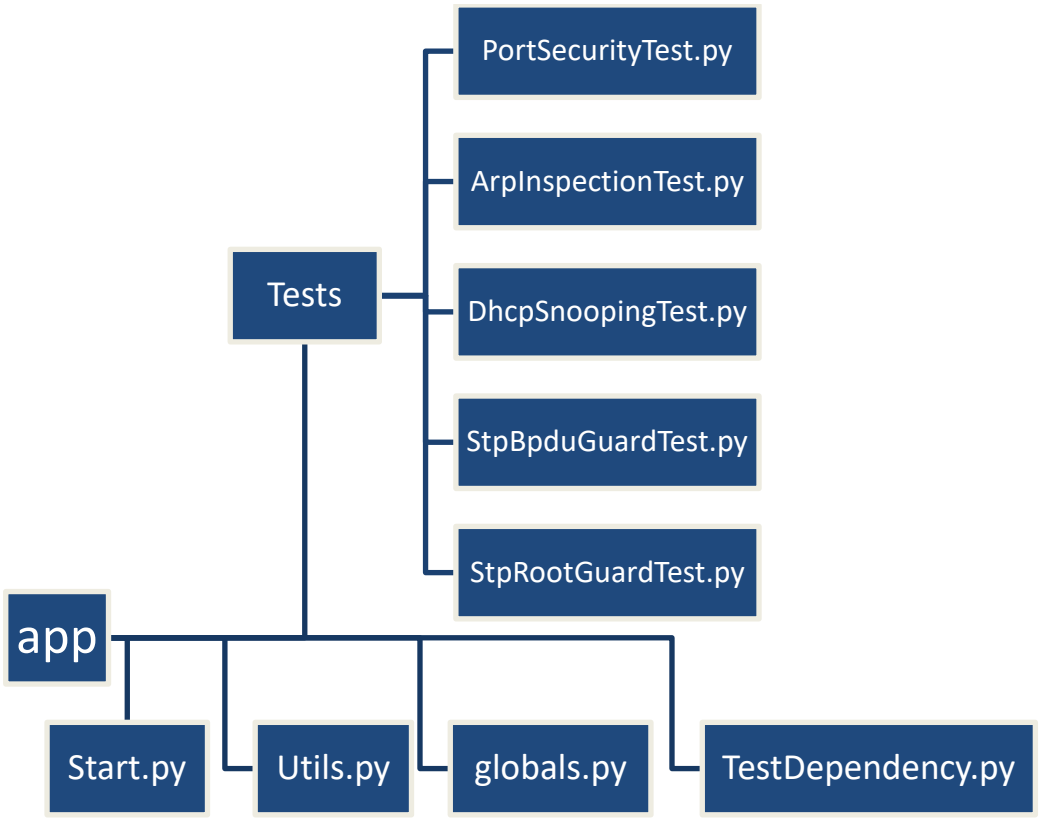
7.1 Základní popis aplikace

Celý program se skládá z hlavní aplikační třídy zodpovědné za kontrolu všech požadavků instalovaných modulů, deklaraci proměnných apod. Dále zahrnuje třídy s celkem pěti bezpečnostními testy. Ty se vždy skládají ze dvou hlavních částí – útok a ověření jeho úspěšnosti.
Nástroj provádí ověření přítomnosti těchto funkcí:

- Port security
- Dynamic ARP inspection
- DHCP snooping
- STP BPDU Guard
- STP Root Guard

Aplikace je vytvořena v jazyce Python.
Jde o interpretovaný, vysokoúrovňový programovací jazyk podporující různé programovací přístupy – procedurální, objektově-orientované a funkcionální. Jeho první verze pochází z roku 1994 a jejím autorem byl Guido van Rossum. Vývojový tým jazyka uvedl Python ve verzi 2.0 v říjnu roku 2000, jejíž oficiální podpora trvala až do letošního roku (2020). Nejnovější verzí jazyka je Python 3.8.2 (k 30. 4. 2020).
Mezi hlavní důvody výběru tohoto programovacího jazyka pro implementaci praktické části práce patří především jednoduchost, dostupnost velkého množství modulů a knihoven rozšiřujících základní funkcionalitu a také použitelnost i pro platformy IoT.

7.2 Struktura aplikace



Aplikační balíček se skládá ze souboru Start.py, hlavní aplikační třídy, jejíž metoda main() je volána v nastavení entry point jako počáteční. Utils.py obsahuje pomocné a doplňkové funkce, využívané v rámci celé aplikace, například zjišťování aktuálního stavu síťových rozhraní, odchycení specifického paketu, vyprázdnění ARP cache apod. V této úrovni struktury lze také nalézt soubor globals.py s definicí globálních proměnných sdílených napříč moduly v rámci celé aplikace a definici třídy TestDependency. Ten je využíván při vytváření pravidel závislostí mezi určitými bezpečnostními testy.

Podadresář Tests již obsahuje samotné soubory dílčích bezpečnostních testů. Jejich koncepce je z velké části podobná, zahrnuje následující atributy a metody:

- `__name` – název bezpečnostního testu
- `__testManual` – instrukce pro použití daného testu, definice požadavků ke spuštění, zapojení a konfiguraci sítě
- `__partialResults` – pole pro ukládání výsledků jednotlivých pokusů testu (hodnoty datového typu Boolean)
- `__init__()` – konstruktor třídy
- `start()` – spouští samotný test

- `testPassed()` – vrací konečný výsledek testu (hodnota datového typu Boolean)
- `getTestName()` – vrací název testu (hodnota datového typu String)
- `getTestManual()` – vrací instrukce pro použití daného testu

Aplikace dále využívá tyto externí moduly a knihovny dostupné v otevřených databázích pro vývojáře:

- ipaddress, zdroj: (60)
- netifaces, zdroj: (61)
- pyersinia, zdroj: (62)
- ping3, zdroj: (63)
- scapy, zdroj: (64)
- psutil, zdroj: (65)
- python-iptables, zdroj: (66)
- randmac, zdroj: (67)

7.3 Uživatelské rozhraní

Obrázek 27 - Úvodní nabídka s výběrem testu ke spuštění
Zdroj: Tomáš Bartoníček

```
>>>>>>>>> ARP inspection test <<<<<<<<<
Please connect both network interfaces to general access ports for endpoint devices (e.g. computer).

[PROMPT] Press ENTER to start the ARP inspection test...
ARP inspection test starting...
ARP inspection test completed. Result: False
=====================================
===== netSecurity Analyser REPORT =====
=====================================
[FAILED] ARP inspection test failed. This protection is not enabled.
```

Obrázek 28 - Spuštění a vyhodnocení testu samostatně
Zdroj: Tomáš Bartoníček

```
=======================================
===== netSecurity Analyser REPORT =====
=======================================
[FAILED] DHCP snooping test failed. This protection is not enabled.
[FAILED] ARP inspection test failed. This protection is not enabled.
[PASSED] Port Security test passed. This protection is enabled.
[FAILED] STP BPDU guard test failed. This protection is not enabled.
[FAILED] STP Root Guard test failed. This protection is not enabled.
tomas@ubuntu18-04tls:~/PycharmProjects/netsecurity-analyser/app$
```

Obrázek 29 - Kompletní vyhodnocení po dokončení všech testů
Zdroj: Tomáš Bartoníček

7.4 Použití

Aplikace je určena zejména administrátorům sítí, kteří potřebují zjistit úroveň zabezpečení, případně může být použita jako jedna z více částí kompletního penetračního testu.

Pro běh aplikace je vyžadován operační systém Linux (standardní distribuce Debian/Ubuntu), prostředí Python verze 3 nebo vyšší a sada povinných Python modulů uvedených v souboru **requirements.txt** (součástí repozitáře v příloze práce). Kontrola jejich přítomnosti je prováděna jednak při instalaci samotné aplikace a také vždy po spuštění. Současně je vyžadována instalace vybraných balíčků v systému.

Kvůli specifickým potřebám některých testů a procesu zpětné kontroly úspěšnosti je třeba, aby počítač, kde bude program spuštěn, byl do sítě LAN připojen alespoň dvěma síťovými kartami. Ve většině případů tedy dojde k použití externího síťového adaptéru (například RJ-45 – USB).

Po spuštění je uživatel vyzván k vyplnění několika nezbytných informací – určení dvojice síťových rozhraní, kterých bude aplikace využívat, a také to, zda mají být spuštěny za sebou všechny testy či pouze jeden vybraný.

Program v průběhu testování informuje vždy o výsledku dílčích testů, na konci je zobrazen souhrnný přehled.

7.5 Popis postupů testování

Tato podkapitola podrobně popisuje postupy použité pro testování jednotlivých bezpečnostních technologií na L2.

Stejně jako v předchozích kapitolách, i zde je primárně uvažována implementace funkcí na síťových prvcích výrobce Cisco Systems, Inc.

7.5.1 Úvodní příprava aplikace

Po spuštění programu je provedena řada kontrol ověřujících způsobilost systému a síťového prostředí pro další běh bezpečnostních testů.

Aplikace kontroluje:

- Spuštění pod správcovským uživatelem root
- Nainstalovanou sadu potřebných balíčků v operačním systému
- Nainstalovanou sadu potřebných Python modulů

Následně uživatel zvolí, které z dostupných síťových rozhraní si přeje použít pro útok a které pro ověření úspěšnosti útoku.
Poté:

- Obě rozhraní musí být zapnuta a aktivní s přidělenou IP adresou
- Musí být nalezena výchozí brána
- Je provedena kontrola přítomnosti STP protokolu v síti zachycením síťového provozu

7.5.2 Port security

Technologie, která při správném nasazení nabízí následující možnosti ochrany před zapojením cizího zařízení do sítě:

- Povolení přístupu pouze pro definované MAC adresy
- Povolení přístupu pouze pro konkrétní počet MAC adres

Právě na druhý typ ochrany je test v aplikaci zaměřen.
Je nutné brát v úvahu rozdílné implementace jednotlivých výrobců síťových prvků, což znamená rozdílnou reakci switche na porušení nastaveného pravidla Port security. Dojde-li k úplnému zablokování (vypnutí) rozhraní, není příliš složité takovou situaci identifikovat. V některých případech je však i po překročení limitního počtu MAC adres stále povolen provoz z prvních *x* adres. Tento jev je na simulaci již náročnější.
Uvažujme potřebu simulovat souběžnou komunikaci z celkem n zařízení.
V analyzátoru je takový postup implementován pomocí vytvoření $n-1$ síťových subrozhraní s náhodně zvolenou MAC adresou. Zbývajícím rozhraním je samotná fyzická síťová karta, na které jsou ostatní subrozhraní navázána. Následně jsou pomocí DHCP klienta přiděleny rozhraním IP adresy ze stejného síťového segmentu.
Skrze každé jednotlivé rozhraní je poté spuštěn ICMP ping test na předem určenou IP adresu (testovací bod). V případě úspěšného prvotního testu zahájí aplikace simulaci souvislého ICMP provozu z tohoto subrozhraní.
Tento postup se opakuje $(n-1)$krát. V případě selhání prvotního ICMP ping testu je celý proces ukončen – jedná se o známku dosažení limitu maximálního povoleného počtu MAC adres na rozhraní switche. Test je následně vyhodnocen jako pozitivní – ochrana Port security je aktivní.

Zdrojový kód testovacího algoritmu

```
testPingIp = gatewayIp
pingTimeout = 1
maxNumberOfSimulatedDevices = 4


for i in range(0, maxNumberOfSimulatedDevices):
  vInterfaceName = 'veth' + str(i)
  randomMacAddr = str(randmac.RandMac('00:00:00:00:00:00', True))
  Utils.createVirtualNetworkIf(vInterfaceName, attackingIf, randomMacAddr,
'macvlan')
  virtualInterfaces.append(vInterfaceName)


  if not Utils.isNetworkIfUp(attackingIf):
      self.__partialResults.append(False)
      break


  Utils.renewIpFromDhcp(vInterfaceName)



  if not Utils.hasNetworkIfIpAddr(vInterfaceName):
      atttackingIfIp =
netifaces.ifaddresses(attackingIf)[netifaces.AF_INET][0]['addr']
      netmask = netifaces.ifaddresses(attackingIf)[netifaces.AF_INET][0]['netmask']
      currentNetworkCIDR = ipaddress.ip_network(atttackingIfIp + '/' + netmask,
False)
      chosenIp = ipaddress.ip_address(list(currentNetworkCIDR.hosts())[RandNum(1,
254)])


      while Utils.isIpReachableFromNetworkIf(chosenIp, attackingIf, 3) and
Utils.isIpInArpCache(attackingIf, chosenIp):
          chosenIp =
ipaddress.ip_address(list(currentNetworkCIDR.hosts())[RandNum(1, 254)])


      Utils.bindIpToNetIf(vInterfaceName, chosenIp)


  if self.testPingFromInterface(testPingIp, virtualInterfaces[i]):
      self.__partialResults.append(True)
      testPingProceeses.append(Process(target=Utils.startIcmpTrafficSimulation,
name="Test ping process",
                                       args=(vInterfaceName, testPingIp)))
      testPingProceeses[-1].start()
  else:
      self.__partialResults.append(False)
      break
```

Cyklus pro vytvoření jednotlivých simulovaných zařízení.

Vytvoření virtuálního rozhraní s náhodnou MAC adresou.

Zachycení situace zablokování portu switchem.

Přiřazení volné IP adresy ze stejného síťového rozsahu, není-li přidělena z DHCP serveru.

Je-li z vytvořeného virtuálního rozhraní možné komunikovat, zaznamená se výsledek a spustí simulace komunikace.

V opačném případě je celý cyklus ukončen a zaznamenán negativní výsledek.

7.5.3 Dynamic ARP inspection

Funkce inspekce rámců ARP protokolu brání síť před útokem ARP cache poisoning a tedy před rizikem přesměrování legitimního provozu směrem k útočníkovi pomocí falešných záznamů v ARP tabulce.

Technologie DAI využívá znalosti vazeb IP-MAC adres uložených v databázi DHCP snoopingu a switch tak propouští pouze provoz, jehož zdrojové adresy odpovídají hodnotám těchto vazeb.

Toho lze efektivně využít při testování, zda je tato funkce na switchi zapnutá.

Popisovaná aplikace tedy nastavuje na síťové rozhraní klienta různé statické IP adresy a snaží se skrze ně komunikovat.

Nejprve je nutné zjistit v jakém IP subnetu se zařízení právě nachází, to lze odvodit z aktuálně přidělené adresy DHCP serverem. Algoritmus tedy zvolí IP adresu ze stejného rozsahu, otestuje pomocí ICMP ping, zda není aktivní. Zkontroluje také dynamické ARP záznamy pro případ, že by na nějakém zařízení byl ICMP protokol blokován firewallem (například OS Windows s veřejným síťovým profilem). Pokud není vybraná adresa přidělena, aplikace odebere všechny dosavadní IP adresy z rozhraní a nastaví sem zvolenou adresu.

Po provedení změny je spuštěn ICMP ping test na předem určenou IP adresu (testovací bod). Negativní výsledek testu (nedostupný ICMP ping) indikuje nastavenou funkci Dynamic ARP inspection.

Pro zvýšení relevance výsledků je postup přidělení statické adresy opakován n-krát (ve výchozím stavu čtyřikrát).

Zdrojový kód testovacího algoritmu

```
attemptCount = 3
hostsCount = 30

atttackingIfIp = netifaces.ifaddresses(attackingIf)[netifaces.AF_INET][0]['addr']
netmask = netifaces.ifaddresses(attackingIf)[netifaces.AF_INET][0]['netmask']
currentNetwork = ipaddress.ip_network(atttackingIfIp + '/' + netmask, False)
availableIpList = list(currentNetwork.hosts())[:hostsCount]

Utils.bringNetworkIfDown(networkIf)
Utils.flushArpCache()

forexp = (i for i in range(0, attemptCount))

for i in forexp:
    ip = availableIpList.__getitem__(int(RandNum(0, len(availableIpList))))
    testedIpWithPrefix = (str(ip) + "/" + str(currentNetwork.prefixlen))

    if not Utils.isIpInArpCache(attackingIf, str(ip)) and not
Utils.isIpReachableFromNetworkIf(ip, attackingIf, 5)

        netIfIp = ipaddress.ip_address(
            netifaces.ifaddresses(attackingIf)[netifaces.AF_INET][0]['addr'])
        Utils.unbindIpFromNetIf(attackingIf, netIfIp)
        time.sleep(2)
        Utils.bindIpToNetIf(attackingIf, testedIpWithPrefix)

        if Utils.isIpReachableFromNetworkIf(testIp, attackingIf, 10):
            self.__partialResults.append(True)
        else:
            self.__partialResults.append(False)
```

Definice IP rozsahu pro staticky přidělované IP adresy.

Pokud není IP adresa dosažitelná ani není nalezena v ARP cache, je přiřazena na síťové rozhraní.

Testování, zda je síťová komunikace skrze přiřazenou IP adresu funkční.

7.5.4 DHCP Snooping

DHCP Snooping poskytuje ochranu sítě před provozem záškodnického DHCP serveru. Ten může záměrně přidělovat upravené TCP/IP nastavení, tím je usnadněno provedení dalších útoků (například ARP cache poisoning, Man-In-The-Middle, DNS spoofing apod.). Funkce DHCP Snooping provádí filtraci DHCP offer zpráv jdoucích od serveru směrem ke klientovi, který o TCP/IP nastavení žádá. Tyto typy zpráv mohou následně přicházet buď pouze z předem definovaného seznamu IP adres legitimních DHCP serverů či z důvěryhodných portů, do kterých jsou tyto servery připojeny. Pokud není vyhověno žádné z podmínek, switch zprávu DHCP offer zahodí.

V případě uváděného řešení **netSecurityAnalyser** je pro simulaci cizího DHCP serveru použit Python modul Pyersinia a metody třídy dhcp_rogue.py.

Přípravná část útoku zahrnuje deklaraci parametrů simulovaného serveru, tedy síťový rozsah, ze kterého budou přidělovány falešné IP adresy, síťové rozhraní použité pro útok, adresu výchozí brány a DNS serveru. Je také nutné zajistit, aby ověřovací rozhraní nemohlo TCP/IP nastavení získat z legitimních serverů v síti. Z tohoto důvodu provede aplikace vložení nového pravidla do firewallové tabulky, které zablokuje příchozí DHCP provoz ze všech IP adres, vyjma adresy rozhraní použitého pro útok. K tomuto účelu slouží Python modul python-iptables poskytující knihovny pro efektivní práci s linuxovým firewallem iptables.

Připravený DHCP server je následně spuštěn v samostatném procesu na pozadí.

Následuje tedy část ověření úspěšnosti útoku. Na druhém síťovém rozhraní je provedena změny MAC adresy za náhodně zvolenou, DHCP klient zažádá o přidělení nového nastavení. Po určité časové odmlce aplikace porovná aktuální IP adresu na ověřovacím síťovém rozhraní s IP rozsahem definovaném při konfiguraci falešného DHCP serveru. Pokud adresa rozhraní náleží do rozsahu, útok je považován za úspěšný a funkce DHCP snooping tedy není nastavena (případně je rozhraní použité pro útok zapojeno do důvěryhodného portu switche).

Ukázka firewallového pravidla použité pro blokování DHCP provozu

```
iptables -I INPUT ! -s [attackIfIpAddr] -p udp --sport 67 --dport 68
-j DROP
```

Zdrojový kód testovacího algoritmu

```
rogueDhcpServerNetworkCIDR = "5.5.5.0/24"
attackingIfIp = netifaces.ifaddresses(attackingIf)[netifaces.AF_INET][0]['addr']


processRogueDhcpServer = Process(target=self.startDhcpRogueServer,
name='rogueDhcpServerThread', args=(networkIf, attackingIf,
rogueDhcpServerNetworkCIDR))


processAttackVerification = Process(target=self.startAttackVerification,
name='attackVerificationThread', args=(networkIf, rogueDhcpServerNetworkCIDR,
self.__partialResults))


fwDhcpBlockRule = self.getAuxiliaryFirewallRule(networkIf, attackingIfIp)


self.insertAuxiliaryFirewallRule(fwDhcpBlockRule)


processRogueDhcpServer.start()


processAttackVerification.start()
processAttackVerification.join()


processRogueDhcpServer.terminate()
self.removeAuxiliaryFirewallRule(fwDhcpBlockRule)
```

Deklarace parametrů falešného DHCP serveru.

Vytvoření a přidání potřebného firewallového pravidla.

Spuštění DHCP serveru a následné spuštění procesu ověření útoku.

Zdrojový kód ověřovacího algoritmu

```
Utils.falseNetworkIfMacAddr(primaryNetworkIf)
    try:
        interfaceIpAsNetwork =
ipaddress.ip_network((netifaces.ifaddresses(primaryNetworkIf)[netifaces.AF_INET][0]
['addr']), strict=False).network_address


        if interfaceIpAsNetwork in
list(ipaddress.ip_network(rogueDhcpServerNetworkCIDR).hosts()):
            resultList.append(True)
        else:
            resultList.append(False)
    except:
        resultList.append(False)
```

Pokud aktuální IP adresa ověřovacího rozhraní patří do IP rozsahu falešného serveru, útok byl úspěšný.

7.5.5 STP BPDU Guard

Jde o poměrně jednoduchý, avšak důležitý bezpečnostní doplněk STP protokolu určený k použití na portech switche s koncovými zařízeními (tzv. access portech). Pokud na některém z těchto portů dojde k přijetí rámce typu BPDU, switch jej vypne do stavu error-disabled.

Jednoduchý je tedy i proces ověření, zda je ochrana aktivní.

Také v tomto případě jsou využity funkce modulu Pyersinia, konkrétně metody třídy stp_bpdu_conf.py.

Na počátku jsou deklarovány parametry útoku. Zde je to pouze síťové rozhraní, skrze které budou BPDU zprávy odesílány, samotný proces Pyersinia je poté spuštěn na pozadí. Reakce switche je takřka okamžitá, pro vyhodnocení postačí zjistit, zdali je síťové rozhraní aktivní či nikoliv.

Pokud nebylo switchem vypnuto, ochrana STP BPDU Guard není na daném portu aktivní.

Zdrojový kód testovacího algoritmu

```
testPingIp = gatewayIp


attackConfig = pyersinia_lib.api.GlobalParameters()
attackConfig.attack = ['stp_conf']
attackConfig.interface = [attackingIf]


attackProcess = Process(target=self.startAttack, name='stpBpduAttackProcess',
args=[attackConfig])
attackVerification = Process(target=self.startAttackVerification,
name='stpBpduAttackVerificationProcess', args=[attackingIf, self.__partialResults])


attackProcess.start()


attackVerification.start()
attackVerification.join()


attackProcess.terminate()
```

Definice parametrů útoku

Spuštění procesů pro simulaci i ověření úspěšnosti útoku

Zdrojový kód ověřovacího algoritmu

```
if Utils.isNetworkIfUp(attackingInterface):
            resultList.append(True)
        else:
            resultList.append(False)
```

Ověření, zda je síťové rozhraní stále aktivní

7.5.6 STP Root Guard

I STP Root Guard je ochranný mechanismus STP protokolu. Dokáže zabránit převzetí role root bridge jiným switchem a tím i nežádoucím změnám v L2 topologii.

Při přijetí tzv. superior BPDU zprávy na portu se zapnutou ochranou jej switch přepne do stavu „root-inconsistent“ (ekvivalentní ke STP stavu portu listening), žádný provoz tedy není skrze tento port přeposílán.

Simulace procesu převzetí role root bridge je v aplikaci opět zajištěna Python modulem Pyersinia, konkrétně částí stp_root_role.py.

Ta zajistí sniffing části STP provozu, z něj extrahuje údaj o MAC adrese aktuálního root bridge, kterou poníží o 1. Tentýž úkon provede také s MAC adresou bridge, z něhož STP paket pochází.

Dále sestaví nový STP paket s vypočítanou MAC adresou bridge a root bridge, hodnoty rootID a bridgeID jsou převzaty z původního STP paketu. Cílovou MAC adresou je multicast adresa používaná STP protokolem (hodnota 01:80:c2:00:00:00).

Připravený paket je poté odesílán skrze určené rozhraní do sítě.

Verifikace úspěšnosti útoku je v tomto případě založena na znalosti principu generování STP BPDU paketů použitých pro převzetí root bridge role.

Před započetím samotného útoku i po jeho startu provede aplikace zachycení části STP provozu, ze kterého zjistí hodnotu Root Bridge ID. Tyto hodnoty v samostatném procesu porovná a pokud se hodnota Root BID změní po započetí útoku o 1, jak je očekáváno, lze považovat útok za úspěšný. V případě zapnuté bezpečnostní funkce STP Root Guard na daném portu switche se adresa Root BID nezmění.

Zdrojový kód testovacího algoritmu

```
        attackConfig = pyersinia_lib.api.GlobalParameters()
        attackConfig.attack = ['stp_root_role']
        attackConfig.interface = [attackingIf]

        verificationProcesses = []
        numberOfVerifProcesses = 3

        attackProcess = Process(target=self.startAttack,
name='stpRootAttackProcess', args=[attackConfig])

        for i in range(numberOfVerifProcesses):
            verificationProcesses.append(
                Process(target=self.startAttackVerification,
name='stpRootAttackVerificationProcess',
                        args=[networkIf, self.__partialResults,
Utils.getSniffedStpPacket(networkIf)]))

        attackProcess.start()

        for i in range(numberOfVerifProcesses):
            verificationProcesses[i].start()
            verificationProcesses[i].join()

        attackProcess.terminate()
```

Definice parametrů útoku a počtu ověření

Vytvoření procesů útoku a ověření

Spuštění procesů útoku a ověření

Zdrojový kód ověřovacího algoritmu

```
newStpPacket = Utils.getSniffedStpPacket(networkIf=networkIf)

# Author of the source code below for new Root bridge MAC calculation:
# Nottingham Prisa Team.
# URL:
https://github.com/nottinghamprisateam/pyersinia/blob/master/pyersinia_lib/libs/plu
gins/stp_root_role.py
rootMAC = stpPacket.rootmac
rootMAC = rootMAC[::-1]
newMAC = ''
aux = False
```

Zachycení STP packetu

Proces výpočtu předpokládané nové hodnoty Root bridge MAC address z STP packetu získaného před testováním

```
for x in range(len(rootMAC)):
    if (rootMAC[x] in '123456789abcdefABCDEF') and not aux:
        n = int(rootMAC[x], 16)
        n -= 1
        n = format(n, 'x')
        newMAC += n
        aux = True
    else:
        newMAC += rootMAC[x]
expectedNewRootMac = newMAC[::-1]

if newStpPacket.rootmac == expectedNewRootMac:
    resultList.append(True)
else:
    resultList.append(False)
```

Porovnání hodnot Root bridge MAC address a zápis výsledků testu

7.6 Řešené problémy v průběhu implementace

V průběhu samotné implementace, tvorby aplikace a následného testování se naskytla celá řada problémů, které bylo nutné vyřešit či implementovat odlišnou cestou. Předchozí postup nebyl dostatečně spolehlivý a univerzální, případně způsoboval další problémy. Níže jsou popsány nejzajímavější z nich.

Simulace několika paralelně otevřených síťových spojení z různých MAC adres na stejném fyzickém portu

Jednou z možných reakcí bezpečnostní funkce port-security je pouhá blokace provozu z nově připojených zařízení, pokud byl přesažen limit maximálního počtu zařízení namísto úplného vypnutí fyzického portu switche.

Bylo tedy nutné vymyslet způsob simulace paralelní komunikace z několika zařízení v jeden časový okamžik. Jako nejlepší řešení se ukázalo vytvoření virtuálních L2 rozhraní typu „macvlan“ s náhodně zvolenou MAC adresou, navázaných na jedno fyzické síťové rozhraní. Z každého virtuálního rozhraní je poté v samostatném procesu spuštěna kontinuální ICMP komunikace.

Po skončení celého procesu testování je nutné ukončit otevřené procesy s ICMP komunikací odesláním signálu SIGSTOP, případně SIGKILL.

Vhodný způsob ověření úspěšnosti STP Root Guard útoku

Po provedení útoku s převzetím role STP Root Bridge bylo třeba zajistit relevantní ověření úspěšnosti útoku. Jako nejjednodušší se ukázalo zachycení STP provozu před a po útoku a porovnání hodnot Root Bridge ID v paketech. Hodnota po provedení útoku by měla být o jednotku nižší než v paketu získaném před útokem (závisí na volbě Root BID hodnoty použité pro útok).

Problémy s kompatibilitou balíčků převzatých od jiných autorů

Testovací aplikace využívá Python modul Pyersinia pro simulaci útoků DHCP rogue server, STP BPDU attack a STP Root attack. Bohužel v průběhu testování se objevily potíže s některými částmi zdrojového kódu, který bylo třeba pro zajištění správné funkčnosti drobně syntakticky upravit. Zmíněný modul Pyersinia je tedy při instalaci aplikace netSecurityAnalyser instalován z GitHub repozitáře autora této práce. Vzhledem k datu

poslední úpravy modulu původními autory provedené přibližně před pěti lety není jisté, zda a kdy lze očekávat oficiální opravu chyby.

Změna hodnoty již existujících proměnných v nově vzniklých procesech

Jak již bylo zmíněno v podkapitole Struktura aplikace, algoritmy ověření úspěšnosti provedeného útoku ukládají dílčí výsledky do proměnné **partialResults**. Verifikace se však v určitých případech provádí v nových procesech, které nemají dříve vytvořené proměnné k dispozici.

Tento jev lze řešit pomocí synchronizace objektů mezi spuštěnými procesy využitím třídy **multiprocessing.managers.SyncManager**, která vytvoří a vrátí tzv. ProxyObjects pro běžně používané datové typy určených k synchronizaci.

Ukončování pozůstalých procesů po testování

Během samotné simulace síťových útoků či verifikaci její úspěšnosti dochází k častým manipulacím se síťovými rozhraními testovací stanice (zapínání a vypínání fyzických rozhraní, falšování MAC adres, vytváření a mazání virtuálních rozhraní atd.) v relativně krátkém časovém horizontu. To má za následek přetrvávání pozůstalých procesů v systému, jenž způsobují komplikace při běhu aplikace.

V případě popisovaného analyzátoru šlo především o problém pozůstalých procesů klienta DHCP na síťových rozhraních, které prodlužovaly dobu získání nové IP adresy.

Jako optimální řešení se ukázalo zaznamenání časového razítka při spuštění bezpečnostního testu a použití Python modulu psutil pro vypsání seznamu procesů, které byly vytvořeny až po spuštění testu. Tyto procesy pak byly po dokončení testu jednoduše ukončeny odesláním signálu SIGSTOP, popřípadě SIGKILL.

```
def stopProcessesFromTime(processesNamesArray, fromTimeStamp):
    for p in psutil.process_iter():
        if p.name() in processesNamesArray and time.strftime("%H:%M:%S",
time.localtime(p.create_time())) > fromTimeStamp:
            if globals.debugMode:
                print('[DEBUG] Try to stop process', p.name())


            os.kill(p.pid, signal.SIGSTOP)
```

7.7 Postup a parametry testování

7.7.1 Metodika testování

Cílem testování bylo především zjistit, zda bude vyvinutá aplikace podávat relevantní informace při použití switchů různých výrobců, neboť většina z nich provádí v implementaci technologie s drobnými odlišnostmi, což se může následně projevit na celkovém chování aplikace. Vzhledem k tomu, že rychlost testování závisí na několika vedlejších faktorech – nastavení limitních hodnot bezpečnostních funkcí, výkonu testovací stanice apod., není použita jako srovnávací kritérium.

Při testování byly dále využity různé síťové topologie (star, ring), které mají v souvislosti s STP protokolem a zapojenými redundantními cestami odlišné chování. U zapojení byly uvažovány různé varianty propojení testovací stanice (například situace, kdy je každá síťová karta zapojena do jiného switche v topologii). STP protokol byl testován v různých verzích dostupných na konkrétních switchích (MST, Rapid-PVST).

Pro lepší přiblížení testovacího prostředí reálné počítačové síti byl použit i model sítě s kombinací několika zařízení odlišných výrobců.

7.7.2 Použitá zařízení

Testování bylo prováděno na několika modelech obsahujících síťové prvky různých výrobců. U každého použitého typu je rovněž uveden také seznam podporovaných bezpečnostních funkcí

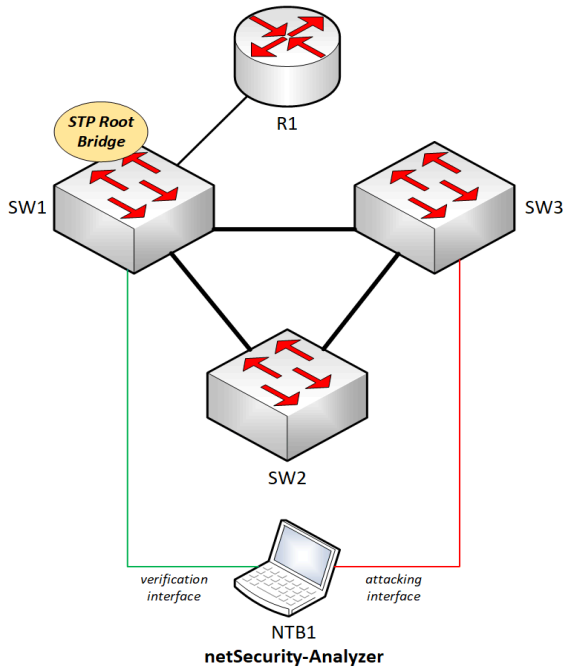
- **HP OfficeConnect 1920S** – „Small office - Home office“ řada
    - Max MAC count (port-security) – *není podporováno aplikací*
    - DHCP snooping
    - ARP attack protection (Dynamic ARP inspection)
    - BPDU protection (STP BPDU Guard)
    - Root protection (STP Root Guard)
- **HP ProCurve 2510** – Small Business řada
    - Port-security
    - BPDU protection (STP BPDU Guard)
    - STP Root guard

- **Fortinet FortiSwitch 224E/248D** – Small Business řada
  - Learning-limit (port-security)
  - DHCP snooping
  - Dynamic ARP inspection
  - STP BPDU guard
  - STP root guard
- **Cisco Catalyst 3560 v2 Series** – Enterprise řada
  - Port-security
  - DHCP snooping
  - Dynamic ARP inspection
  - STP BPDU guard
  - STP root guard

Dále byl použit router MikroTik hAP Lite s jednoduchou konfigurací brány do Internetu, DNS a DHCP serveru pro zajištění základní funkčnosti sítě během testování.

7.8 Průběh a výsledky testování

7.8.1 Topologie Ring složená ze 3 switchů Cisco Catalyst 3560 v2 Series



Obrázek 30 - Topologie Ring – switche Cisco
Zdroj: Tomáš Bartoníček

Výsledky testování

Popis situace, nastavení	Úspěšný útok	Správná detekce (ne)úspěšnosti	Poznámky
DHCP snooping enabled	NE	ANO	
DHCP snooping disabled	ANO	ANO	Pokud je tato funkce vypnuta, následující test ARP inspection se již neprovádí.
ARP inspection enabled	NE	ANO	
ARP inspection disabled	ANO	ANO	
Port-security, max. 3 zařízení, akce „Protect“	NE	ANO	Před provedením testu je vhodné promazat CAM tabulku switche.
Port-security, max. 6 zařízení, akce „Shutdown“	NE	ANO	Před provedením testu je vhodné promazat CAM tabulku switche. Po provedení testu je nutné, aby správce manuálně povolil zablokovaný port na switchi.
STP BPDU guard enabled*	NE	ANO	Po provedení testu je nutné, aby správce manuálně povolil zablokovaný port na switchi. Je-li funkce zapnuta, test STP root guard se již neprovádí.
STP BPDU guard disabled*	ANO (např. TCN)	ANO	
STP root guard enabled*	NE	ANO	Po provedení testu bylo nutné déle čekat na přidělení IP adresy DHCP serverem z důvodu přechodu rozhraní ze stavu listening a vypnuté funkce PortFast.
STP root guard disabled*	ANO	ANO	

Pro otestování komunikace mezi switchi jsou síťová rozhraní testovací stanice zapojena do odlišných switchů v kruhové topologii.

* Testy STP protokolu byly provedeny s variantami RSTP i MST, výsledky byly totožné.

Ukázky z testování

```
%DHCP_SNOOPING-5-DHCP_SNOOPING_MATCH_MAC_FAIL: DHCP_SNOOPING drop message because the chaddr doesn't match source mac, message type: DH
5a.b63c.af43, MAC sa: 00e0.4c68.0212
%DHCP_SNOOPING-5-DHCP_SNOOPING_UNTRUSTED_PORT: DHCP_SNOOPING drop message on untrusted port, message type: DHCPACK, MAC sa: 00e0.4c68.0

%SYS-5-CONFIG_I: Configured from console by admin on vty0 (192.168.90.244)
%DHCP_SNOOPING-5-DHCP_SNOOPING_MATCH_MAC_FAIL: DHCP_SNOOPING drop message because the chaddr doesn't match source mac, message type: DH
5a.b63c.af43, MAC sa: 00e0.4c68.0212
%DHCP_SNOOPING-5-DHCP_SNOOPING_MATCH_MAC_FAIL: DHCP_SNOOPING drop message because the chaddr doesn't match source mac, message type: DH
5a.b63c.af43, MAC sa: 00e0.4c68.0212
%DHCP_SNOOPING-5-DHCP_SNOOPING_UNTRUSTED_PORT: DHCP_SNOOPING drop message on untrusted port, message type: DHCPACK, MAC sa: 00e0.4c68.0

%DHCP_SNOOPING-5-DHCP_SNOOPING_MATCH_MAC_FAIL: DHCP_SNOOPING drop message because the chaddr doesn't match source mac, message type: DH
```

Obrázek 31 - Činnost funkce DHCP snooping
Zdroj: Tomáš Bartoníček

```
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([0015.5d02.9011/192.168.90.244/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([68f7.2869.34d1/192.168.90.248/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([68f7.2869.34d1/192.168.90.248/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([0015.5d02.9011/192.168.90.244/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([68f7.2869.34d1/192.168.90.248/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([0015.5d02.9011/192.168.90.244/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([68f7.2869.34d1/192.168.90.248/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([68f7.2869.34d1/192.168.90.248/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([0015.5d02.9011/192.168.90.244/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([0015.5d02.9011/192.168.90.244/0000.0000.0000/192.168.90.1/02:16:
)
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/13, vlan 1.([68f7.2869.34d1/192.168.90.248/0000.0000.0000/192.168.90.1/02:16:
```

Obrázek 32 - Činnost funkce ARP inspection
Zdroj: Tomáš Bartoníček

```
%SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port Fa0/2 with BPDU Guard enabled. Disabling port.
%PM-4-ERR_DISABLE: bpduguard error detected on Fa0/2, putting Fa0/2 in err-disable state
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to down
%LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to down
```

Obrázek 33 - Činnost funkce STP BPDU guard
Zdroj: Tomáš Bartoníček

```
SW1#show spanning-tree root

                                        Root    Hello Max Fwd
MST Instance            Root ID         Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
MST0             24576 e8ba.7090.a380         0    2   20  15
SW1#show spanning-tree root

                                        Root    Hello Max Fwd
MST Instance            Root ID         Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
MST0             24576 e8ba.7090.a370    200000    2   20  15  Fa0/23
```

Obrázek 34 - Převzetí role STP root bridge útočníkem
Zdroj: Tomáš Bartoníček

```
SW3#show spanning-tree root

                                Root    Hello Max Fwd
Vlan                Root ID     Cost    Time  Age Dly  Root Port
---------------- ------------------ --------- ----- --- ---  ------------
VLAN0001         24577 e8ba.7090.a380       19    2   20  15  Fa0/24
SW3#
*Mar  1 01:47:32.106: %SPANTREE-2-ROOTGUARD_BLOCK: Root guard blocking port FastEthernet0/2 on VLAN0001.show spanning-tree root
```

Obrázek 35 - Činnost funkce STP root guard
Zdroj: Tomáš Bartoníček

7.8.2 Topologie Ring složená ze 3 switchů Fortinet FortiSwitch 224E/248D

Výsledky testování

Popis situace, nastavení	Úspěšný útok	Správná detekce (ne)úspěšnosti	Poznámky
DHCP snooping enabled	NE	ANO	
DHCP snooping disabled	ANO	ANO	Pokud je tato funkce vypnuta, následující test ARP inspection se již neprovádí.
ARP inspection enabled	NE	ANO	
ARP inspection disabled	ANO	ANO	
Learning-limit, max. 3 zařízení	NE	ANO	Před provedením testu je vhodné promazat CAM tabulku switche.
STP BPDU guard enabled	NE	ANO	Po provedení testu je nutné, aby správce manuálně povolil zablokovaný port na switchi. Je-li funkce zapnuta, test STP root guard se již neprovádí.
STP BPDU guard disabled	ANO (např. TCN)	ANO	
STP root guard enabled	NE	ANO	
STP root guard disabled	ANO	ANO	

Pro otestování komunikace mezi switchi jsou síťová rozhraní testovací stanice zapojena do odlišných switchů v kruhové topologii.

Ukázky z testování

Timestamp	Subtype	Level	User	Action	Message
1 second ago	Switch	Warning	arp-inspection		Arp-Insp-Drop: Source interface mismatch {mac=0:e0:4c:68:2:12, vlan=1, ip= 192:168:90:17}.
2 seconds ago	Switch	Warning	arp-inspection		Arp-Insp-Drop: Source interface mismatch {mac=0:e0:4c:68:2:12, vlan=1, ip= 192:168:90:17}.
2 seconds ago	Switch	Warning	arp-inspection		Arp-Insp-Drop: Source interface mismatch {mac=0:e0:4c:68:2:12, vlan=1, ip= 192:168:90:17}.
2 seconds ago	Switch	Warning	arp-inspection		Arp-Insp-Drop: Source interface mismatch {mac=0:e0:4c:68:2:12, vlan=1, ip= 192:168:90:17}.
3 seconds ago	Switch	Warning	arp-inspection		Arp-Insp-Drop: Source interface mismatch {mac=0:e0:4c:68:2:12, vlan=1, ip= 192:168:90:17}.
3 seconds ago	Switch	Warning	arp-inspection		Arp-Insp-Drop: Source interface mismatch {mac=0:e0:4c:68:2:12, vlan=1, ip= 192:168:90:17}.
3 seconds ago	Switch	Warning	arp-inspection		Arp-Insp-Drop: Source interface mismatch {mac=0:e0:4c:68:2:12, vlan=1, ip= 192:168:90:17}.

Obrázek 36 - Činnost funkce Dynamic ARP inspection
Zdroj: Tomáš Bartoníček

Timestamp	Subtype	Level	User	Action	Message
24 seconds ago	Spanning Tree	notice	stp_daemon	state-change	primary port port2 instance 0 changed state from forwarding to discarding
24 seconds ago	Spanning Tree	notice	stp_daemon	role-change	primary port port2 instance 0 changed role from designated to disabled
24 seconds ago	Link	information			primary switch port port2 has gone down
24 seconds ago	Spanning Tree	notice	stpd		BPDU Guard: BPDU detected on port2. Shutting down port2.
1 minute ago	Spanning Tree	notice	stp_daemon	state-change	primary port port2 instance 0 changed state from discarding to forwarding
1 minute ago	Spanning Tree	notice	stp_daemon	state-change	primary port port2 instance 0 changed state from forwarding to discarding

Obrázek 37 - Činnost funkce STP BPDU guard
Zdroj: Tomáš Bartoníček

Timestamp	Subtype	Level	User	Action	Message
1 second ago	Spanning Tree	notice	stp_daemon	state-change	primary port port2 instance 0 changed state from forwarding to discarding
1 second ago	Spanning Tree	notice	stp_daemon	state-change	primary port port47 instance 0 changed state from forwarding to discarding
1 second ago	Spanning Tree	notice	stp_daemon	role-change	primary port port47 instance 0 changed role from root to designated
1 second ago	Spanning Tree	notice	stp_daemon	role-change	primary port port48 instance 0 changed role from designated to root
1 second ago	Spanning Tree	notice	stp_daemon	role-change	primary port port47 instance 0 changed role from designated to root
1 second ago	Spanning Tree	notice	stp_daemon	role-change	primary port port2 instance 0 changed role from root to designated

ID	Priority	Root MAC	Root Priority	Root Path Cost	Regional Root MAC	Regional Priority	Regional Path Cost	Remaining Hops	Bridge MAC	Root
0	16384	e8:1c:ba:d1:82:75	16384	20,000	90:6c:ac:ad:b8:c8	32768	20,000	18	e8:1c:ba:d1:82:76	–

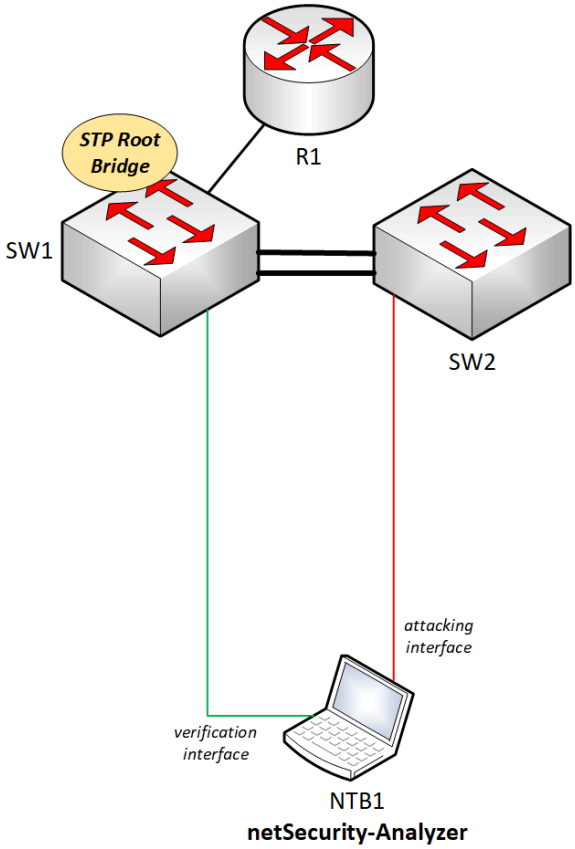
Obrázek 38 - Převzetí role STP root bridge útočníkem
Zdroj: Tomáš Bartoníček

Timestamp	Subtype	Level	User	Action	Message
13 seconds ago	Spanning Tree	notice	stp_daemon	role-change	primary port port2 instance 0 changed role from alternative to designated
13 seconds ago	Spanning Tree	notice	stp_daemon	state-change	STP Root Guard: Un-blocking Port2 (STP instance 0)
27 seconds ago	Spanning Tree	notice	stp_daemon	role-change	primary port port2 instance 0 changed role from designated to alternative
27 seconds ago	Spanning Tree	notice	stp_daemon	state-change	STP Root Guard: Superior BPDUs received on Port2 (STP instance 0)
3 minutes ago	Spanning Tree	notice	stp_daemon	state-change	primary port port2 instance 0 changed state from discarding to forwarding

ID	Priority	Root MAC	Root Priority	Root Path Cost	Regional Root MAC	Regional Priority	Regional Path Cost	Remaining Hops	Bridge MAC	Root
0	16384	e8:1c:ba:d1:82:76	16384	0	e8:1c:ba:d1:82:76	16384	0	20	e8:1c:ba:d1:82:76	✔

Obrázek 39 - Činnost funkce STP root guard
Zdroj: Tomáš Bartoníček

7.8.3 Topologie Ring složená ze 2 switchů HP ProCurve 2510



Obrázek 40 - Topologie Ring – switche HP
Zdroj: Tomáš Bartoníček

Výsledky testování

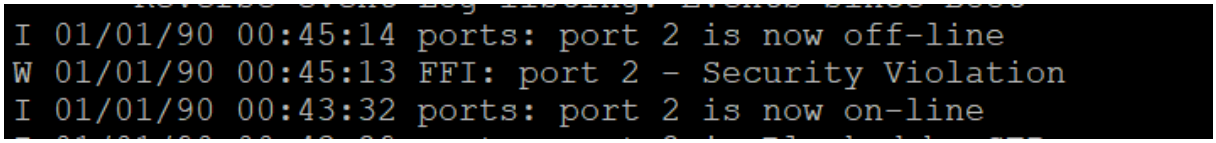
Popis situace, nastavení	Úspěšný útok	Správná detekce (ne)úspěšnosti	Poznámky
DHCP snooping enabled	-	-	Switch tuto funkci nepodporuje.
DHCP snooping disabled	-	-	Switch tuto funkci nepodporuje.
ARP inspection enabled	-	-	Switch tuto funkci nepodporuje.
ARP inspection disabled	-	-	Switch tuto funkci nepodporuje.
Port-security, max. 8 zařízení	NE	ANO	Před provedením testu je vhodné promazat CAM tabulku switche.

STP BPDU protection enabled*	NE	ANO	Po provedení testu je nutné, aby správce manuálně povolil zablokovaný port na switchi. Je-li funkce zapnuta, test STP root guard se již neprovádí.
STP BPDU protection disabled*	ANO (např. TCN)	ANO	
STP root guard enabled*	NE	ANO	
STP root guard disabled*	ANO	ANO	

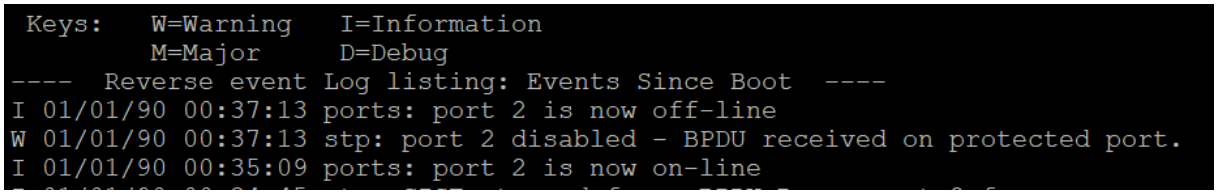
Pro otestování komunikace mezi switchi jsou síťová rozhraní testovací stanice zapojena do odlišných switchů v kruhové topologii.

* Testy STP protokolu byly provedeny s variantami RSTP i MST, výsledky byly totožné.

Ukázky z testování



Obrázek 41 - Činnost funkce Port-security
Zdroj: Tomáš Bartoníček



Obrázek 42 - Činnost funkce STP BPDU protection
Zdroj: Tomáš Bartoníček

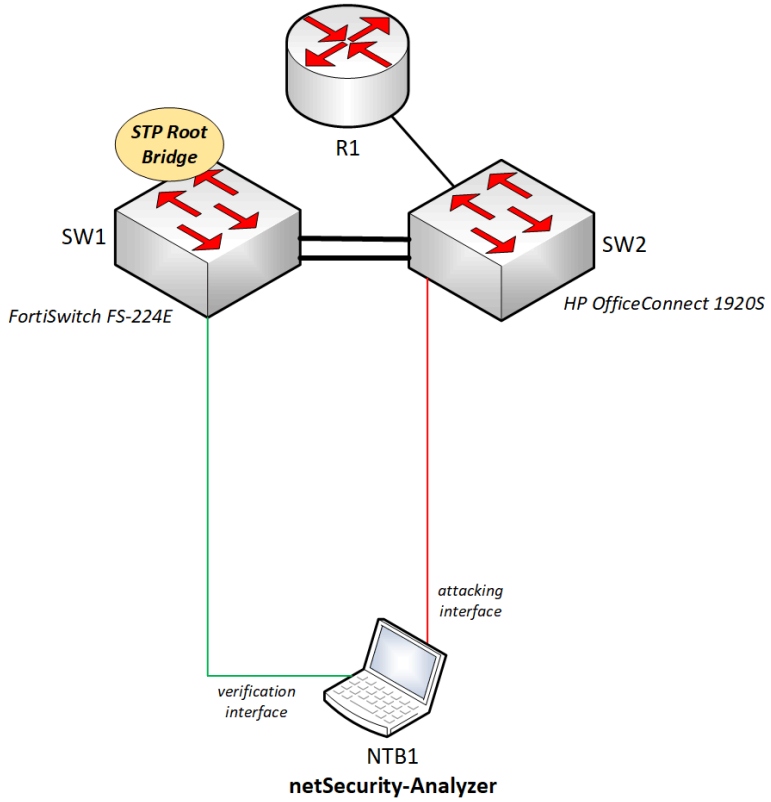
```
Multiple Spanning Tree (MST) Information

 STP Enabled   : Yes
 Force Version : RSTP-operation
 IST Mapped VLANs : 1-4094
 Switch MAC Address : 009c02-e56b20
 Switch Priority    : 32768
 Max Age  : 20
 Max Hops : 20
 Forward Delay : 15

 Topology Change Count  : 19
 Time Since Last Change : 2 secs

 CST Root MAC Address : 009c02-e56b10
 CST Root Priority    : 32768
 CST Root Path Cost   : 200000
 CST Root Port        : 2

 IST Regional Root MAC Address : 009c02-e56b20
 IST Regional Root Priority    : 32768
 IST Regional Root Path Cost   : 0
 IST Remaining Hops            : 20

 Root Guard Ports :
 TCN Guard Ports  :
 Protected Ports :
 Filtered Ports :

                |           Prio              | Designated    Hello
 Port Type      | Cost      rity  State       | Bridge        Time  PtP Edge
 ---- --------- + --------- ----- ----------- + ------------- ----- --- ----
 1    10/100TX  | 200000    128   Forwarding  | 009c02-e56b20 2     Yes Yes
 2    10/100TX  | 200000    128   Forwarding  | 009c02-e56b10 2     Yes No
 3    10/100TX  | Auto      128   Disabled    |
 4    10/100TX  | Auto      128   Disabled    |
 5    10/100TX  | Auto      128   Disabled    |
 6    10/100TX  | Auto      128   Disabled    |
```

Obrázek 43 - Převzetí role STP root bridge útočníkem
Zdroj: Tomáš Bartoníček

```
I 01/01/90 00:24:16 ports: port 2 is now on-line
I 01/01/90 00:23:52 stp: CIST starved for a BPDU Rx on port 2 from
           32768:009c02-e56b10
I 01/01/90 00:23:34 ports: port 2 is Blocked by STP
I 01/01/90 00:22:31 ports: port 11 is now on-line
I 01/01/90 00:22:31 ports: port 2 is now on-line
```

Obrázek 44 - Činnost funkce STP Root guard
Zdroj: Tomáš Bartoníček

7.8.4 Kombinovaná topologie HP OfficeConnect 1920S + FortiSwitch 224E



Obrázek 45 - Kombinovaná testovací topologie HP + FortiSwitch
Zdroj: Tomáš Bartoníček

Výsledky testování

Popis situace, nastavení	Úspěšný útok	Správná detekce (ne)úspěšnosti	Poznámky
DHCP snooping enabled	NE	ANO	
DHCP snooping disabled	ANO	ANO	Pokud je tato funkce vypnuta, následující test ARP inspection se již neprovádí.
ARP attack protection enabled	NE	ANO	
ARP attack protection disabled	ANO	ANO	
Max MAC count, max. 5 zařízení	NE	**NE**	Před provedením testu je vhodné promazat CAM tabulku switche.

STP BPDU protection enabled	NE	ANO	Po provedení testu je nutné, aby správce manuálně povolil zablokovaný port na switchi. Je-li funkce zapnuta, test STP root guard se již neprovádí.
STP BPDU protection disabled	ANO (např. TCN)	ANO	
STP root protection enabled	NE	ANO	
STP root protection disabled	ANO	ANO	

Pro otestování komunikace mezi switchi jsou síťová rozhraní testovací stanice zapojena do odlišných switchů v kruhové topologii.

Rozhraní používané pro generování útoků bylo v tomto testovacím scénáři zapojeno záměrně do switche HP OfficeConnect 1920S, aby byly otestovány bezpečnostní funkce také u tohoto modelu switche.

Ukázky z testování

IFNET	Error	LINK_UPDOWN	GigabitEthernet1/0/48 link status is DOWN.
MSTP	Warning	MSTP_BPDU_PROTECTION	BPDU-Protection port GigabitEthernet1/0/48 received BPDUs.
MSTP	Information	MSTP_FORWARDING	Instance 0's port GigabitEthernet1/0/48 has been set to forwarding state.

Obrázek 46 - Činnost funkce STP BPDU protection
Zdroj: Tomáš Bartoníček

Source	Level	Digest	Description
MSTP	Information	MSTP_NOTIFIED_TC	Instance 0's port GigabitEthernet1/0/2 was notified of a topology change.
MSTP	Information	MSTP_NOTIFIED_TC	Instance 0's port GigabitEthernet1/0/2 was notified of a topology change.
MSTP	Information	MSTP_NOTIFIED_TC	Instance 0's port GigabitEthernet1/0/2 was notified of a topology change.
MSTP	Information	MSTP_NOTIFIED_TC	Instance 0's port GigabitEthernet1/0/2 was notified of a topology change.

Obrázek 47 - Převzetí role STP root bridge útočníkem
Zdroj: Tomáš Bartoníček

Source	Level	Digest	Description
MSTP	Warning	MSTP_ROOT_PROTECTION	Instance 0's ROOT-Protection port GigabitEthernet1/0/2 received superior BPDUs.
MSTP	Information	MSTP_DISCARDING	Instance 0's port GigabitEthernet1/0/2 has been set to discarding state.

Obrázek 48 - Činnost funkce STP root protection
Zdroj: Tomáš Bartoníček

8 Závěry a doporučení

Jak je zřejmé z výsledků získaných v předchozí kapitole, aplikace poskytuje pomocí kontrolních algoritmů relevantní informace o přítomnosti ochranných funkcí pro L2 vrstvu OSI modelu.

Ačkoli bylo testování provedeno na zařízeních tří známých světových výrobců, problém s funkčností by neměl nastat ani ve spojení s jinými značkami síťového hardwaru. Jak ukázaly výsledky testů, bezpečnostní funkce jsou implementovány, až na drobné odchylky, velmi podobnými způsoby. Zvláště pak vyšší modelové řady, umožňující plnou konfiguraci zařízení v příkazové řádce CLI, mají princip funkčnosti takřka stejný.

Vytvořenou aplikaci lze vhodně použít především v korporátních počítačových sítích, kde je vyžadována přítomnost bezpečnostních funkcí na L2 z důvodu eliminace rizika útoku. Nabízí se také možnost integrace ve formě Python modulu do již existujícího většího řešení, kde analyzátor nebude podávat přímý výstup informací uživateli, ale například pouze předávat data o úspěšnosti testů zpět do hlavního programu, který provede další vyhodnocení (viz například koncepce nástroje *Simulated Penetration Testing and Mitigation Analysis* uvedeného v rešerši práce).

Popisovanou aplikaci netSecurityAnalyser lze mimo jiné využít také k různým simulačním účelům při plánování nových síťových struktur či k edukaci během výuky nastavování bezpečnosti síťových prvků.

Další vývoj aplikace by měl zcela logicky směřovat k vytvoření grafického rozhraní, vylepšení interakce s uživatelem a v neposlední řadě také k vytvoření verze pro operační systémy Windows. Vzhledem k vysokému nárůstu míry automatizace procesů při nasazování a provozování IT technologií je vhodné i v případě tohoto řešení uvažovat o implementaci konceptu klient-server. Jednotlivé testovací stanice či například jednoúčelové minipočítače by pak periodicky získávaly data o nastavení bezpečnosti sítě a tyto informace nahrávaly na centrální server. Zde by byly dále zpracovávány, vyhodnocovány a ukládány pro potřeby archivace.

Účelem této diplomové práce je vysvětlení základní teorie L2 protokolů v počítačových sítích, objasnění rizik spojených s útoky na spojové vrstvě a možnosti jejich obrany. Praktická část představuje řešení automatické analýzy přítomnosti ochranných funkcí v reálné síťové infrastruktuře.

9 Seznam použité literatury

1. LIN, Derek. Anomaly detection system for enterprise network security. US9112895B1.

2. B. IYAMUREMYE a H. SHIMA. Network security testing tools for SMEs (small and medium enterprises). In: *2018 IEEE International Conference on Applied System Invention (ICASI)*: *2018 IEEE International Conference on Applied System Invention (ICASI)* [online]. 2018, s. 414–417. ISBN null. Dostupné z: doi:10.1109/ICASI.2018.8394272

3. VISOOTTIVISETH, Vasaka, Phuripat AKARASIRIWONG, Siravitch CHAIYASART a Siravit CHOTIVATUNYU. PENTOS: Penetration testing tool for Internet of Thing devices. In: *TENCON 2017 - 2017 IEEE Region 10 Conference*: *TENCON 2017 - 2017 IEEE Region 10 Conference* [online]. 2017, s. 2279–2284. ISSN 2159-3450. Dostupné z: doi:10.1109/TENCON.2017.8228241

4. BACKES, Michael, Jörg HOFFMANN, Robert KÜNNEMANN, Patrick SPEICHER a Marcel STEINMETZ. Simulated Penetration Testing and Mitigation Analysis. *ArXiv*. 2017, **abs/1705.05088**.

5. DARGIE, W. a C. POELLABAUER. *Fundamentals of Wireless Sensor Networks: Theory and Practice* [online]. B.m.: Wiley, 2010. Wireless Communications and Mobile Computing. ISBN 978-0-470-97568-8. Dostupné z: https://books.google.cz/books?id=8c6k0EVr6rMC

6. GIBSON, J.D. *The Communications Handbook* [online]. B.m.: CRC Press, 2018. Electrical engineering handbook series. ISBN 978-1-4200-4116-3. Dostupné z: https://books.google.cz/books?id=Tokk5bZxB0MC

7. HELD, G. *Carrier Ethernet: Providing the Need for Speed* [online]. B.m.: CRC Press, 2008. Auerbach publications. ISBN 978-1-4200-6040-9. Dostupné z: https://books.google.cz/books?id=Xbbq68wGz64C

8. WU, C.H. a J.D. IRWIN. *Introduction to Computer Networks and Cybersecurity* [online]. B.m.: CRC Press, 2016. ISBN 978-1-4987-6013-3. Dostupné z: https://books.google.cz/books?id=JAAZCwAAQBAJ

9. ODOM, W. *CCNA Routing and Switching 200-120 Official Cert Guide Library* [online]. B.m.: Pearson Education, 2013. Official Cert Guide. ISBN 978-0-13-347989-8. Dostupné z: https://books.google.cz/books?id=36k5UBScR_QC

10. CLARKE, G.E. *CompTIA Network+ Certification Study Guide, Seventh Edition (Exam N10-007)* [online]. B.m.: McGraw-Hill Education, 2018. ISBN 978-1-260-12205-3. Dostupné z: https://books.google.cz/books?id=QgxlDwAAQBAJ

11. HUAWEI TECHNOLOGIES CO., L. *HCNA Networking Study Guide* [online]. B.m.: Springer Singapore, 2016. ISBN 978-981-10-1554-0. Dostupné z: https://books.google.cz/books?id=fRyfDAAAQBAJ

12. MEINEL, Christoph a Harald SACK. Network Access Layer (1): Wired LAN Technologies. In: *Internetworking: Technological Foundations and Applications* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, s. 131–259. ISBN 978-3-642-35392-5. Dostupné z: doi:10.1007/978-3-642-35392-5_4

13. ROBERTAZZI, T. *Basics of Computer Networking* [online]. B.m.: Springer New York, 2011. SpringerBriefs in Electrical and Computer Engineering. ISBN 978-1-4614-2104-7. Dostupné z: https://books.google.cz/books?id=Fnz2qDpofqEC

14. HELD, G. *Windows Networking Tools: The Complete Guide to Management, Troubleshooting, and Security* [online]. B.m.: CRC Press, 2016. ISBN 978-1-4665-1107-1. Dostupné z: https://books.google.cz/books?id=gDzSBQAAQBAJ

15. COWLEY, J. *Communications and Networking: An Introduction* [online]. B.m.: Springer London, 2012. Undergraduate Topics in Computer Science. ISBN 978-1-4471-4356-7. Dostupné z: https://books.google.cz/books?id=orFwKxpKZxgC

16. CISCO, Systems. *IP Addressing: ARP Configuration Guide, Cisco IOS Release 15M&T*.

17. FILIPE, J. a M.S. OBAIDAT. *E-Business and Telecommunication Networks: Third International Conference, ICETE 2006, Setúbal, Portugal, August 7-10, 2006, Selected Papers* [online]. B.m.: Springer Berlin Heidelberg, 2008. Communications in Computer and Information Science. ISBN 978-3-540-70760-8. Dostupné z: https://books.google.cz/books?id=CkzDgho9qVkC

18. ALANI, Mohammed M. TCP/IP Model. In: Mohammed M. ALANI, ed. *Guide to OSI and TCP/IP Models* [online]. Cham: Springer International Publishing, 2014, s. 19–50. ISBN 978-3-319-05152-9. Dostupné z: doi:10.1007/978-3-319-05152-9_3

19. MEINEL, Christoph a Harald SACK. Internet Layer. In: Christoph MEINEL a Harald SACK, ed. *Internetworking: Technological Foundations and Applications* [online]. Berlin, Heidelberg: Springer, 2013 [vid. 2020-02-20], X.media.publishing, s. 455–590. ISBN 978-3-642-35392-5. Dostupné z: doi:10.1007/978-3-642-35392-5_7

20. GOUDA, Mohamed G. a Chin-Tser HUANG. A secure address resolution protocol. *Computer Networks* [online]. 2003, **41**(1), 57–71. ISSN 1389-1286. Dostupné z: doi:https://doi.org/10.1016/S1389-1286(02)00326-2

21. HUAWEI TECHNOLOGIES CO., Ltd., ed. VLAN. In: Ltd. HUAWEI TECHNOLOGIES CO., ed. *HCNA Networking Study Guide* [online]. Singapore: Springer Singapore, 2016, s. 119–145. ISBN 978-981-10-1554-0. Dostupné z: doi:10.1007/978-981-10-1554-0_5

22. HUAWEI TECHNOLOGIES CO., Ltd., ed. Inter-VLAN Layer 3 Communication. In: Ltd. HUAWEI TECHNOLOGIES CO., ed. *HCNA Networking Study Guide* [online]. Singapore: Springer Singapore, 2016, s. 229–244. ISBN 978-981-10-1554-0. Dostupné z: doi:10.1007/978-981-10-1554-0_9

23. DA SILVA, M.M. *Cable and Wireless Networks: Theory and Practice* [online]. B.m.: CRC Press, 2018. ISBN 978-1-4987-5154-4. Dostupné z: https://books.google.cz/books?id=A6x-DwAAQBAJ

24. PÉREZ, A. *Network Security* [online]. B.m.: Wiley, 2014. Networks and Telecommunications Series. ISBN 978-1-84821-758-4. Dostupné z: https://books.google.cz/books?id=gTtcCwAAQBAJ

25. WILKINS, S. a T. SMITH. *CCNP Security Secure 642-637 Official Cert Guide: CCNP Se Secu 642-637 ePub_1* [online]. B.m.: Pearson Education, 2011. Official Cert Guide. ISBN 978-0-13-237856-7. Dostupné z: https://books.google.cz/books?id=1Xj95IP8AosC

26. SHIRAZI, Farid a Alexander KRASNOV. Cloud Security: A Virtualized VLAN (V2LAN) Implementation. In: Masaaki KUROSU, ed. *Human-Computer Interaction. Theory, Design, Development and Practice*. Cham: Springer International Publishing, 2016, s. 610–621. ISBN 978-3-319-39510-4.

27. HUAWEI TECHNOLOGIES CO., Ltd., ed. STP. In: Ltd. HUAWEI TECHNOLOGIES CO., ed. *HCNA Networking Study Guide* [online]. Singapore: Springer Singapore, 2016, s. 99–117. ISBN 978-981-10-1554-0. Dostupné z: doi:10.1007/978-981-10-1554-0_4

28. RANJBAR, A. *Troubleshooting and Maintaining Cisco IP Networks (TSHOOT) Foundation Learning Guide: (CCNP TSHOOT 300-135)* [online]. B.m.: Pearson Education, 2014. Foundation Learning Guides. ISBN 978-0-13-396594-0. Dostupné z: https://books.google.cz/books?id=M63TBQAAQBAJ

29. SOLOMON, M.G., D. KIM a J.L. CARRELL. *Fundamentals of Communications and Networking* [online]. B.m.: Jones & Bartlett Learning, 2014. Jones & Bartlett Learning information systems security & assurance series. ISBN 978-1-284-06015-7. Dostupné z: https://books.google.cz/books?id=LVktBAAAQBAJ

30. FORTZ, Bernard, Luís GOUVEIA a Martim JOYCE-MONIZ. Optimal design of switched Ethernet networks implementing the Multiple Spanning Tree Protocol. *Special Issue on the Ninth International Colloquium on Graphs and Optimization (GO IX), 2014* [online]. 2018, **234**, 114–130. ISSN 0166-218X. Dostupné z: doi:10.1016/j.dam.2016.07.015

31. ACADEMY, Cisco Networking. *Switched Networks Companion Guide: Switch Networ Compan Gui*. B.m.: Cisco Press, 2014. ISBN 978-0-13-347644-6.

32. WILKINS, S. *Designing for Cisco Internetwork Solutions (DESGN) Foundation Learning Guide: (CCDA DESGN 640-864)* [online]. B.m.: Pearson Education, 2011. Foundation Learning Guides. ISBN 978-0-13-258242-1. Dostupné z: https://books.google.cz/books?id=x9CG5Wz6trsC

33. *Frame Formats* [online]. [vid. 2020-03-01]. Dostupné z: https://docstore.mik.ua/univercd/cc/td/doc/product/lan/trsrb/frames.htm#xtocid12

34. G., D.K. *Network Security Attacks and Countermeasures* [online]. B.m.: IGI Global, 2016. Advances in Information Security, Privacy, and Ethics (1948-9730). ISBN 978-1-4666-8762-2. Dostupné z: https://books.google.cz/books?id=hOJmCwAAQBAJ

35. RAMACHANDRAN, Vivek a Sukumar NANDI. Detecting ARP Spoofing: An Active Technique. In: Sushil JAJODIA a Chandan MAZUMDAR, ed. *Information Systems Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, s. 239–250. ISBN 978-3-540-32422-5.

36. TRABELSI, Zouheir a Khaled SHUAIB. Spoofed ARP Packets Detection in Switched LAN Networks. In: Joaquim FILIPE a Mohammad S. OBAIDAT, ed. *E-Business and Telecommunication Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, s. 81–91. ISBN 978-3-540-70760-8.

37. RICHARDSON, Stephen. Using Dynamic ARP Inspection - Configuration Mode. *Cisco Certified Expert* [online]. 26. srpen 2019 [vid. 2020-03-20]. Dostupné z: https://www.ccexpert.us/configuration-mode/using-dynamic-arp-inspection.html

38. LOBO, L. a U. LAKSHMAN. *CCIE Security v4.0 Quick Reference: Cisc CCIE Secu v4.0 Qui ePub_3* [online]. B.m.: Pearson Education, 2014. Quick Reference. ISBN 978-0-13-385511-1. Dostupné z: https://books.google.cz/books?id=ajVcBAAAQBAJ

39. MAC ADDRESS SPOOFING. *TECHNOCATE* [online]. [vid. 2020-03-20]. Dostupné z: http://technocate.weebly.com/12/post/2014/06/mac-address-spoofing.html

40. JUNG, Sungmo, Jong Hyun KIM a Seoksoo KIM. A Study on MAC Address Spoofing Attack Detection Structure in Wireless Sensor Network Environment. In: Tai-hoon KIM, Hojjat ADELI, Rosslin John ROBLES a Maricel BALITANAS, ed. *Advanced Communication and Networking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, s. 31–35. ISBN 978-3-642-23312-8.

41. COLEMAN, D.D., D.A. WESTCOTT, B.E. HARKINS a S.M. JACKMAN. *CWSP Certified Wireless Security Professional Official Study Guide: Exam PW0-204* [online]. B.m.: Wiley, 2011. ISBN 978-0-470-61964-3. Dostupné z: https://books.google.cz/books?id=0ZWLn57EdpsC

42. PAQUET, C. *Implementing Cisco IOS Network Security (IINS 640-554) Foundation Learning Guide: Imp Cisco IOS Netw Sec F_c2* [online]. B.m.: Pearson Education, 2012. Foundation Learning Guides. ISBN 978-0-13-298331-0. Dostupné z: https://books.google.cz/books?id=oTKEX9stxooC

43. MUKHTAR, Husameldin, Khaled SALAH a Youssef IRAQI. Mitigation of DHCP starvation attack. *Computers & Electrical Engineering* [online]. 2012, **38**(5), 1115–1128. ISSN 0045-7906. Dostupné z: doi:https://doi.org/10.1016/j.compeleceng.2012.06.005

44. SANTOS, O. a R. TAYLOR. *CompTIA PenTest+ PT0-001 Cert Guide* [online]. B.m.: Pearson Education, 2018. Certification Guide. ISBN 978-0-13-522618-6. Dostupné z: https://books.google.cz/books?id=lQJ6DwAAQBAJ

45. HUBBALLI, Neminath a Nikhil TRIPATHI. A closer look into DHCP starvation attack in wireless networks. *Computers & Security* [online]. 2017, **65**, 387–404. ISSN 0167-4048. Dostupné z: doi:10.1016/j.cose.2016.10.002

46. MCMILLAN, T. *CCNA Security Study Guide: Exam 210-260* [online]. B.m.: Wiley, 2018. ISBN 978-1-119-40988-5. Dostupné z: https://books.google.cz/books?id=5tJKDwAAQBAJ

47. VYNCKE, E. a C. PAGGEN. *LAN Switch Security: What Hackers Know About Your Switches* [online]. B.m.: Pearson Education, 2007. Networking Technology: Security. ISBN 978-0-13-443360-8. Dostupné z: https://books.google.cz/books?id=0yraCgAAQBAJ

48. BOUSKAP@SAMURAJ-CZ.COM, Petr Bouška-Samuraj; e-mail: Běžné útoky na switche, Cisco Dynamic ARP Inspection < články -> SAMURAJ-cz.com. *SAMURAJ-cz.com* [online]. [vid. 2020-03-08]. Dostupné z: https://www.samuraj-cz.com/clanek/bezne-utoky-na-switche-cisco-dynamic-arp-inspection/

49. ENCK, William. ARP Spoofing. In: Henk C. A. VAN TILBORG a Sushil JAJODIA, ed. *Encyclopedia of Cryptography and Security* [online]. Boston, MA: Springer US, 2011, s. 48–49. ISBN 978-1-4419-5906-5. Dostupné z: doi:10.1007/978-1-4419-5906-5_100

50. MCQUERRY, S., D. JANSEN a D. HUCABY. *Cisco LAN Switching Configuration Handbook: Cisc Cata Swit Conf Hand_2* [online]. B.m.: Pearson Education, 2009. Networking Technology. ISBN 978-1-58714-063-1. Dostupné z: https://books.google.cz/books?id=97xAymxB2e8C

51. The Importance of Dynamic ARP Inspection in a Network Infrastructure. *My Network Dojo* [online]. 7. prosinec 2019 [vid. 2020-03-20]. Dostupné z: https://www.mynetworkdojo.com/the-importance-of-dynamic-arp-inspection-in-a-network-infrastructure/

52. VACHON, B. *CCNA Security (210-260) Portable Command Guide: Exam 54 Porta Comma ePub _2* [online]. B.m.: Pearson Education, 2016. Portable Command Guide. ISBN 978-0-13-430745-9. Dostupné z: https://books.google.cz/books?id=qB3NCwAAQBAJ

53. DHAKA, Shivali. Traffic Management and Security in Wired Network. In: Sonajharia MINZ, Sushanta KARMAKAR a Latika KHARB, ed. *Information, Communication and Computing Technology*. Singapore: Springer Singapore, 2019, s. 17–30. ISBN 978-981-13-5992-7.

54. TANCESKA, Biljana, Mitko BOGDANOSKI a Aleksandar RISTESKI. Simulation Analysis of DoS, MITM and CDP Security Attacks and Countermeasures. In: Vladimir ATANASOVSKI a Alberto LEON-GARCIA, ed. *Future Access Enablers for Ubiquitous and Intelligent Infrastructures* [online]. Cham: Springer International Publishing, 2015, s. 197–203. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. ISBN 978-3-319-27072-2. Dostupné z: doi:10.1007/978-3-319-27072-2_25

55. DUGGAN, M. *Cisco CCIE Routing and Switching v5.0 Configuration Practice Labs: Cisc CCIE Rou Seit5.0 ePub_3* [online]. B.m.: Pearson Education, 2014. Practical Studies.

ISBN 978-0-13-378673-6. Dostupné z: https://books.google.cz/books?id=JR7qAgAAQBAJ

56. KOCHARIANS, N. a T. VINSON. *CCIE Routing and Switching v5.0 Official Cert Guide, Volume 2: Exa 21 Of Cer Gui ePub_5* [online]. B.m.: Pearson Education, 2014. ISBN 978-0-13-359106-4. Dostupné z: https://books.google.cz/books?id=OfA8BQAAQBAJ

57. FROOM, R., B. SIVASUBRAMANIAN a E. FRAHIM. *Implementing Cisco IP Switched Networks (SWITCH) Foundation Learning Guide: Foundation learning for SWITCH 642-813* [online]. B.m.: Pearson Education, 2010. Foundation Learning Guides. ISBN 978-1-58714-165-2. Dostupné z: https://books.google.cz/books?id=dkDsJmnsejEC

58. HUCABY, D. *CCNP Routing and Switching SWITCH 300-115 Official Cert Guide: Exam 38 Cert Guide* [online]. B.m.: Pearson Education, 2014. ISBN 978-0-13-341430-1. Dostupné z: https://books.google.cz/books?id=FLtoBQAAQBAJ

59. PAQUET, C. *Implementing Cisco IOS Network Security (IINS): (CCNA Security exam 640-553) (Authorized Self-Study Guide)* [online]. B.m.: Pearson Education, 2009. Self-Study Guide. ISBN 978-1-58705-883-7. Dostupné z: https://books.google.cz/books?id=0ses_ButxJUC

60. MOORE, Nicolai. *ipaddress* [online]. Python. 2020. Dostupné z: https://github.com/python/cpython/blob/3.8/Lib/ipaddress.py

61. HOUGHTON, Alastair. *Netifaces 0.10.8* [online]. Python. 2019. Dostupné z: https://github.com/al45tair/netifaces

62. *Pyersinia* [online]. Python. 2016. Dostupné z: https://github.com/nottinghamprisateam/pyersinia

63. KYAN001. *Ping3* [online]. Python. 2020. Dostupné z: https://github.com/kyan001/ping3

64. BIONDI, Philippe. *Scapy* [online]. Python. 2020. Dostupné z: https://scapy.net/

65. RODOLA, Giampaolo. *PSutil* [online]. 2020. Dostupné z: https://github.com/giampaolo/psutil

66. NEBEHAJ, Vilmos. *Python-iptables* [online]. 2017. Dostupné z: https://github.com/ldx/python-iptables

67. SCHMELZLE, Josh. *RandMac* [online]. 2019. Dostupné z: https://github.com/joshschmelzle/randmac

10 Seznam použitých zkratek

- L2 – Layer 2 (spojová vrstva ISO/OSI modelu)
- ARP – Address Resolution Protocol
- STP – Spanning Tree Protocol
- CDP – Cisco Discovery Protocol
- MAC – Media Access Control
- CAM – Content Addressable Memory
- DHCP – Dynamic Host Configuration Protocol
- VLAN – Virtual Local Area Network
- BID – Bridge Identificator
- BPDU – Bridge Protocol Data Unit
- LAN – Local Area Network
- OSI – Open Systems Interconnection
- IEEE – Institute Of Electrical And Electronics Engineers
- CSMA/CD – Carrier Sense Multiple Access/Collision Detection
- CSMA/CA – Carrier Sense Multiple Access/Collision Avoidance
- IP – Internet Protocol
- LLC – Logical Link Control
- SSAP – Source Service Access Point
- DSAP – Destination Service Access Point
- IPX – Internetwork Packet Exchange
- ROM – Read Only Memory
- OUI – Organizational Unique Identifier
- GVRP – Generic VLAN Registration Protocol
- GARP – Generic Attribute Registration Protocol
- MVRP – Multiple VLAN Registration Protocol
- VTP – VLAN Trunking Protocol
- EAPOL – Extensible Authentication Protocol over LAN
- RPC – Root Path Cost
- BOOTP – Bootstrap Protocol
- RARP – Reverse Address Resolution Protocol
- DLCI – Data Link Connection Identifier
- VDC – Virtual Device Context
- VFR – Virtual Route Forwarding
- DDOS – Distributed Denial Of Service
- MITM – Man In The Middle
- SNAP – Subnetwork Access Protocol
- DAI – Dynamic ARP inspection
- ACL – Access Control List
- SNMP – Simple Network Management Protocol
- FCS – Frame Check Sequence
- PDU – Protocol Data Unit
- SIEM – Security Information and Event Management
- IOT – Internet of Things
- GUI – Graphical User Interface

- OWASP – Open Web Application Security Project

11 Přílohy

1) ZIP archiv s kompletními zdrojovými kódy aplikace **netSecurityAnalyser**
2) ZIP archiv s instalačními soubory aplikace **netSecurityAnalyser**

Příloha č. 1

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor:	**Bc. Tomáš Bartoníček**
Studium:	I1800740
Studijní program:	N1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název diplomové práce:	**Analýza zabezpečení sítí na L2**
Název diplomové práce AJ:	Analysis of network security on L2

Cíl, metody, literatura, předpoklady:

Cílem práce je navrhnout komplexní nástroj pro analýzu a testování zabezpečení počítačových sítí na L2. V teoretické části autor představí funkcionality a protokoly L2 vrstvy dle architektury TCP/IP včetně analýzy bezpečnostních rizik a útoků. Dále provede analýzu dostupných nástrojů pro testování zabezpečení L2 vrstvy. V praktické části pak autor vytvoří aplikaci, sloužící pro analýzu a testování zabezpečení L2 s možností využití dostupných nástrojů na základě provedené analýzy.

BRYANT, Chris. *Chris Bryant's CCNP Switch 300-115 Study Guide. 1. UK: Createspace Independent Publishing Platform, 2015. ISBN 9781517351229.*

LACOSTE, Raymond. CCNP routing and switching TSHOOT 300-135 official cert guide. Indianapolis, IN: Cisco Press, [2015]. ISBN 9781587205613.

Garantující pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Mgr. Josef Horálek, Ph.D.
Datum zadání závěrečné práce:	21.10.2014