

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2025

Ing. Anna Voskárová



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## POČÍTAČOVÁ CVIČENÍ PRO PŘEDMĚT NÁVRH, SPRÁVA A BEZPEČNOST POČÍTAČOVÝCH SÍTÍ

COMPUTER EXERCISES FOR THE COURSE DESIGN, ADMINISTRATION AND SECURITY OF COMPUTER NETWORKS

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Ing. Anna Voskářová

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Karel Burda, CSc.

BRNO 2025

# Diplomová práce

magisterský navazující studijní program **Telekomunikační a informační technika**

Ústav telekomunikací

**Studentka:** Ing. Anna Voskářová

**ID:** 211326

**Ročník:** 2

**Akademický rok:** 2024/25

**NÁZEV TÉMATU:**

**Počítačová cvičení pro předmět Návrh, správa a bezpečnost počítačových sítí**

## POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je navrhnout soubor 11 témat počítačových cvičení pro předmět Návrh, správa a bezpečnost počítačových sítí. Následně pro čtyři z nich pak připravit softwarovou podporu a zpracovat návody pro studenty a potřebnou dokumentaci pro vyučujícího. Součástí každého návodu musí být vysvětlení základů problematiky dané úlohy. Návody pro studenty se požadují v listinné podobě i v podobě webových stránek. Úlohy musí být realizovatelné na jediném počítači s několika virtuálními stroji typu VMware. U všech úloh se požaduje jednotná metodika a software. Značnou pozornost je zapotřebí věnovat pedagogickým aspektům úloh, přičemž doba trvání každé úlohy musí být 90 minut.

## DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

**Termín zadání:** 10.2.2025

**Termín odevzdání:** 27.5.2025

**Vedoucí práce:** doc. Ing. Karel Burda, CSc.

**prof. Ing. Jiří Mišurec, CSc.**  
předseda rady studijního programu

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## Abstrakt

Diplomová práca sa venuje návrhu súboru laboratórnych úloh pre počítačové cvičenia predmetu magisterských študijných programov *Návrh, správa a bezpečnosť počítačových sietí*. Cieľom práce je navrhnúť zoznam laboratórnych úloh vhodných pre praktické overenie nadobudnutých teoretických poznatkov a k vybraným úlohám následne vytvoriť podrobné návody pre študentov, ktoré môžu slúžiť ako študijný materiál pre účely praktického vyučovania predmetu, a to jednak v textovej podobe a tiež v podobe HTML stránok. Rovnako tak bude pre každú z vybraných úloh vytvorená dokumentácia pre vyučujúceho.

## Kľúčové slová

laboratórne úlohy, bezpečnosť počítačových sietí, sieťové útoky, dátová komunikácia, dôvernosť, dostupnosť, integrita, autentizácia, anonymizačné siete, Kali Linux, VMware Player, ARP spoofing, IP spoofing, IPsec, RADIUS, EAP, ToR, HTML

## Abstract

The diploma thesis focuses on the design of a set of laboratory assignments for practical exercises in the master's degree course *Design, Management and Security of Computer Networks*. The aim of the thesis is to propose a list of laboratory tasks suitable for the practical verification of acquired theoretical knowledge and, for selected tasks, to create detailed instructions for students. These instructions may serve as study material for the practical teaching of the subject, both in textual format and as HTML pages. In addition, a separate set of documentation will be prepared for instructors for each selected task.

## Keywords

laboratory assignments, computer network security, network attacks, data communication, confidentiality, availability, integrity, authentication, anonymization networks, Kali Linux, VMware Player, ARP spoofing, IP spoofing, IPsec, RADIUS, EAP, Tor, HTML



## **Bibliografická citácia**

VOSKÁROVÁ, Anna. *Počítačová cvičení pro předmět Návrh, správa a bezpečnost počítačových sítí*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2025. Vedúci práce: doc. Ing. Karel Burda, CSc.

# Vyhlásenie autora o pôvodnosti diela

|                                  |                                                                                |
|----------------------------------|--------------------------------------------------------------------------------|
| <b>Meno a priezvisko autora:</b> | Ing. Anna Voskárová                                                            |
| <b>VUT ID autora:</b>            | 211326                                                                         |
| <b>Typ práce:</b>                | Diplomová práca                                                                |
| <b>Akademický rok:</b>           | 2024/25                                                                        |
| <b>Téma záverečnej práce:</b>    | Počítačová cvičení pro předmět Návrh,<br>správa a bezpečnost počítačových sítí |

Vyhlasujem, že svoju záverečnú prácu som vypracovala samostatne pod vedením vedúceho záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autorka uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušila autorské práva tretích osôb, najmä som nezasiahla nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomá následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno: 27. mája 2025

-----  
podpis autorky

## **Pod'akovanie**

Rada by som vyjadrila svoje pod'akovanie vedúcemu tejto diplomovej práce pánovi doc. Ing. Karlovi Burdovi, CSc. za odbornú pomoc a usmernenie pri písaní mojej práce, za jeho cenné rady a poznatky, užitočné pripomienky, inšpiratívne nápady, trpezlivosť a predovšetkým za čas, ktorý mi pri príprave tejto záverečnej práce venoval.

Brno: 27. mája 2025

-----  
podpis autorky

# OBSAH

|                                                                    |           |
|--------------------------------------------------------------------|-----------|
| <b>ZOZNAM OBRÁZKOV .....</b>                                       | <b>8</b>  |
| <b>ZOZNAM TABULIEK .....</b>                                       | <b>10</b> |
| <b>ÚVOD .....</b>                                                  | <b>11</b> |
| <b>1. NÁVRH LABORATÓRNYCH ÚLOH .....</b>                           | <b>13</b> |
| 1.1 POŽIADAVKY NA VYTVORENÉ LABORATÓRNE ÚLOHY .....                | 13        |
| 1.2 CIELE LABORATÓRNYCH ÚLOH .....                                 | 13        |
| 1.3 OSNOVA PREDMETU MPC-NSB .....                                  | 14        |
| 1.4 NAVRHOVANÝ SÚBOR 11 POČÍTAČOVÝCH CVIČENÍ .....                 | 15        |
| 1.5 ŠTRUKTÚRA POČÍTAČOVÝCH CVIČENÍ .....                           | 21        |
| 1.6 POUŽITÉ NÁSTROJE .....                                         | 21        |
| <b>2. POPIS LABORATÓRNYCH ÚLOH .....</b>                           | <b>23</b> |
| 2.1 LABORATÓRNA ÚLOHA Č. 1 .....                                   | 23        |
| 2.2 LABORATÓRNA ÚLOHA Č. 2 .....                                   | 24        |
| 2.3 LABORATÓRNA ÚLOHA Č. 3 .....                                   | 25        |
| 2.4 LABORATÓRNA ÚLOHA Č. 4 .....                                   | 25        |
| 2.5 LABORATÓRNA ÚLOHA Č. 5 .....                                   | 26        |
| 2.6 LABORATÓRNA ÚLOHA Č. 6 .....                                   | 27        |
| 2.7 LABORATÓRNA ÚLOHA Č. 7 .....                                   | 28        |
| 2.8 LABORATÓRNA ÚLOHA Č. 8 .....                                   | 28        |
| 2.9 LABORATÓRNA ÚLOHA Č. 9 .....                                   | 29        |
| 2.10 LABORATÓRNA ÚLOHA Č. 10 .....                                 | 30        |
| 2.11 LABORATÓRNA ÚLOHA Č. 11 .....                                 | 31        |
| <b>3. POUŽITÉ TECHNOLOGIE A SOFTWARE .....</b>                     | <b>33</b> |
| 3.1 VMWARE WORKSTATION .....                                       | 33        |
| 3.2 KALI LINUX .....                                               | 34        |
| <b>4. PRÍPRAVA VIRTUÁLNYCH STROJOV PRE LABORATÓRNE ÚLOHY .....</b> | <b>41</b> |
| 4.1 STIAHNUTIE A INŠTALÁCIA KALI LINUX .....                       | 41        |
| <b>5. DOKUMENTÁCIA PRE POTREBY VÝUKY .....</b>                     | <b>44</b> |
| 5.1 TVORBA NÁVODOV PRE VYBRANÉ LABORATÓRNE ÚLOHY .....             | 44        |
| 5.2 FORMA A ZVEREJNENIE VYTVORENÝCH NÁVODOV .....                  | 47        |
| <b>6. ZÁVER .....</b>                                              | <b>53</b> |
| <b>LITERATÚRA .....</b>                                            | <b>56</b> |
| <b>ZOZNAM SYMBOLOV SKRATIEK .....</b>                              | <b>57</b> |
| <b>ZOZNAM PRÍLOH .....</b>                                         | <b>59</b> |

## ZOZNAM OBRÁZKOV

|     |                                                                                          |    |
|-----|------------------------------------------------------------------------------------------|----|
| 3.1 | Ukážka zoznamu niekoľkých integrovaných nástrojov vo vytvorenom VM s Kali Linux. ....    | 35 |
| 4.1 | Úvodná obrazovka pre prihlásenie užívateľa po spustení vytvoreného VM s Kali Linux. .... | 43 |
| 5.1 | Ukážka (1) vytvorenej webovej stránky: kontrolný test. ....                              | 50 |
| 5.2 | Ukážka (2) vytvorenej webovej stránky: teoretický úvod. ....                             | 51 |
| 5.3 | Ukážka (3) vytvorenej webovej stránky: praktická časť (postup). ....                     | 52 |

### ZOZNAM OBRÁZKOV – PRÍLOHA A (LABORATÓRNA ÚLOHA Č. 3)

|     |                                                                                         |    |
|-----|-----------------------------------------------------------------------------------------|----|
| 1.1 | Všeobecná štruktúra správy ARP protokolu a jej zapuzdrenie do ethernetového rámca. .... | 64 |
| 1.2 | Schematické znázornenie výmeny správ ARP protokolu. ....                                | 65 |
| 1.3 | Schematické znázornenie priebehu ARP <i>spoofing</i> útoku. ....                        | 66 |
| 2.1 | Topológia siete laboratórnej úlohy. ....                                                | 73 |
| 2.2 | Zmeny v ARP tabuľke klienta. ....                                                       | 78 |
| 2.3 | Ukážka zachytenej komunikácie v prostredí nástroja Wireshark. ....                      | 79 |
| 2.4 | Nástroj Ettercap: úvodné grafické rozhranie. ....                                       | 80 |
| 2.5 | Nástroj Ettercap: výpis nájdených zariadení v sieti. ....                               | 81 |

### ZOZNAM OBRÁZKOV – PRÍLOHA C (LABORATÓRNA ÚLOHA Č. 4)

|     |                                                                                                        |     |
|-----|--------------------------------------------------------------------------------------------------------|-----|
| 1.1 | IPsec: transportný a tunelový režim. ....                                                              | 94  |
| 1.2 | Schematické znázornenie zapuzdrovania IPsec paketu. ....                                               | 95  |
| 2.1 | Topológia siete laboratórnej úlohy. ....                                                               | 99  |
| 2.2 | Spustenie IP spoofing útoku v termináli Kali Linux. ....                                               | 101 |
| 2.3 | Ukážka zachytenej komunikácie po spustení útoku IP Spoofing (server). ....                             | 103 |
| 2.4 | Overenie vytvorenia IPsec spojenia (klient). ....                                                      | 105 |
| 2.5 | Výpis nástroja tcpdump: sledovanie komunikácie prostredníctvom vytvoreného IPsec tunela (server). .... | 106 |
| 2.6 | Ukážka zachytenej komunikácie – prenos cez šifrovaný IPsec tunel (server). ....                        | 107 |

### ZOZNAM OBRÁZKOV – PRÍLOHA E (LABORATÓRNA ÚLOHA Č. 8)

|     |                                                                                                             |     |
|-----|-------------------------------------------------------------------------------------------------------------|-----|
| 1.1 | Základné komponenty 802.1X a ich vzájomné prepojenie. ....                                                  | 119 |
| 1.2 | Schematické znázornenie štruktúry správy protokolu EAP. ....                                                | 121 |
| 1.3 | Schematické znázornenie výmeny správ medzi klientom a autentizačným serverom pri EAP-MD5 autentizácii. .... | 123 |
| 1.4 | Schematické znázornenie zapuzdrenia EAPoL správy do rámca technológie Ethernet. ....                        | 125 |
| 1.5 | Komponenty siete s autentizáciou EAP-MD5 a ich vzájomná komunikácia. ....                                   | 126 |
| 1.6 | Schematické znázornenie autentizácie EAP-MD5 podľa štandardu IEEE 802.1X. ....                              | 127 |
| 2.1 | Topológia siete laboratórnej úlohy. ....                                                                    | 131 |
| 2.2 | Ukážka zachytenej komunikácie – výmena správ v priebehu autentizácie EAP-MD5. ....                          | 139 |

### ZOZNAM OBRÁZKOV – PRÍLOHA G (LABORATÓRNA ÚLOHA Č. 11)

|     |                                                             |     |
|-----|-------------------------------------------------------------|-----|
| 1.1 | Vrstvová architektúra siete Tor. ....                       | 153 |
| 1.2 | Schematické znázornenie štruktúry Tor bunky. ....           | 154 |
| 1.3 | Komponenty Tor siete a znázornenie vytvoreného okruhu. .... | 155 |

|     |                                                                                                             |     |
|-----|-------------------------------------------------------------------------------------------------------------|-----|
| 1.4 | Schematické znázornenie vrstveného šifrovania v Tor sieti.....                                              | 157 |
| 2.1 | Topológia siete laboratórnej úlohy.....                                                                     | 159 |
| 2.2 | Pripojenie k prehliadaču Tor Browser (a), načítanie úvodnej domovskej stránky (b). ....                     | 161 |
| 2.3 | Úvodná stránka – potvrdenie úspešného pripojenia. ....                                                      | 162 |
| 2.4 | Ukážka zachytenej komunikácie: využitie Tor Browser pre anonymné prehliadanie. ....                         | 163 |
| 2.5 | <i>Flow Graph</i> komunikácie medzi Tor klientom a vzdialeným serverom. ....                                | 164 |
| 2.6 | Tor klient: ukážka výstupu po prístupe na webové stránky whatismyipaddress.com (pridelenie IP adresy). .... | 164 |
| 2.7 | Porovnanie výsledkov meraní prenosových parametrov: klient bez Tor (hore) vs. klient s Tor (dole). ....     | 166 |

# ZOZNAM TABULIEK

|                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------|----|
| 1.1 Zdôvodnenie výberu a zaradenia jednotlivých úloh do finálneho zoznamu pre počítačové cvičenia ... | 17 |
|-------------------------------------------------------------------------------------------------------|----|

## ZOZNAM TABULIEK – PRÍLOHA A (LABORATÓRNA ÚLOHA Č. 3)

|                                                   |    |
|---------------------------------------------------|----|
| 1.1 Porovnanie nástrojov arpspoof a ettercap..... | 71 |
|---------------------------------------------------|----|

## ZOZNAM TABULIEK – PRÍLOHA C (LABORATÓRNA ÚLOHA Č. 4)

|                                                            |    |
|------------------------------------------------------------|----|
| 1.1 Wireshark: príklady použitých filtrov komunikácie..... | 97 |
|------------------------------------------------------------|----|

## ZOZNAM TABULIEK – PRÍLOHA E (LABORATÓRNA ÚLOHA Č. 8)

|                                                           |     |
|-----------------------------------------------------------|-----|
| 1.1 Prehľad typov EAP správ.....                          | 121 |
| 1.2 Správy odosielané v procese autentizácie EAP-MD5..... | 123 |
| 1.3 Prehľad typov EAPoL správ.....                        | 124 |

# ÚVOD

Bezpečnosť počítačových systémov a sietí predstavuje v dnešnej modernej, už takmer plne digitalizovanej dobe zásadnú otázku. Hľadanie vhodných spôsobov a riešení pre dostatočné zabezpečenie sieťovej komunikácie v medziach celého internetu patrí celosvetovo k najviac diskutovaným témam 21. storočia. Nielen znalosť, informovanosť dostatočné povedomie o tejto problematike, ale i schopnosť správnej orientácie v nej predstavuje pre bežných užívateľov počítačov a internetu nezanedbateľný benefit. Vďaka vzdelaniu a získavaniu praktických skúseností sa z pozície užívateľa stáva pochopenie možných hrozieb a podniknutie vhodných krokov pre realizáciu adekvátnych opatrení vedúcich k ich odstráneniu oveľa jednoduchším. Táto práca sa preto zaoberá návrhom a vytvorením vhodných laboratórnych úloh, aby sa vysokoškolskí študenti už v priebehu svojho univerzitného vzdelávania mohli oboznámiť s vhodnými postupmi pre návrh, správu a bezpečnosť počítačových sietí.

Úvodná časť semestrálnej práce je venovaná návrhu zoznamu laboratórnych úloh vhodných pre praktickú realizáciu na počítačových cvičeniach vyučovaného predmetu s názvom *Návrh, správa a bezpečnosť počítačových sietí*. V tejto časti bude uvedený nielen samotný návrh súboru 11 laboratórnych úloh, ktoré budú predstavovať hlavú výučbovú náplň počítačových cvičení, ciele jednotlivých úloh, predstavy o ich realizácii a ich stručná charakteristika, ale tiež bude výber úloh a ich zaradenie do osnovy počítačových cvičení patrične odôvodnený. V závere budú popísané používané nástroje, software a ďalšie programové vybavenie potrebné pre realizáciu navrhnutých laboratórnych úloh.

Nadväzujúca praktická časť prinesie výber štyroch konkrétnych úloh, pre ktoré budú následne spracované podrobné návody pre praktické uskutočnenie danej úlohy predstavujúce základný výučbový materiál a oporu pre študentov v rámci počítačových cvičení. Súčasťou návodov pre študentov budú nielen základné teoretické východiská potrebné pre úspešné zvládnutie úlohy, no jeho podstatnú zložku bude predstavovať najmä samotný návod, resp. presný a podrobný postup pre praktické vypracovanie úlohy, zadanie samostatnej úlohy a tiež záverečné kontrolné otázky slúžiace jednak pre overenie správnosti samotného riešenia zadanej praktickej úlohy, a taktiež celkového pochopenia diskutovanej problematiky zo strany študentov. Ďalej bude ku každej z vybraných úloh vypracovaná vzorová dokumentácia pre vyučujúceho obsahujúca základné fakty týkajúce sa praktickej časti úlohy, dôležité poznatky a očakávané výstupy, na základe ktorých bude možné následne overiť správnosť riešenia samotnej praktickej časti úlohy, a tiež odpovede na kontrolné otázky odzrkadľujúce mieru porozumenia príslušnej problematike zo strany študentov.

Vytvorené návody budú poskytnuté jednak v podobe výstupu štandardného textového editora, a to vo formáte .docx, no z dôvodu ich praktickej využiteľnosti pre potreby výuky predmetu budú pre všetky návody k laboratórnym úlohám vytvorené aj jednoduché webové stránky, ktoré môžu slúžiť nielen ako výuková pomôcka, ale tiež ako vhodný



študijný materiál. Webové stránky budú vytvárané na základe prevodu textu návodov do podoby html kódu s následnou úpravou vzhľadu a celkovej vizuálnej podoby, aby v čo najväčšej možnej miere reflektovali podobu vytvorených textových návodov.

Pre potreby neskoršieho uplatnenia vytvorených návodov priamo vo výuke budú v priebehu testovania vytvorených návodov plnohodnotne nakonfigurované virtuálne stanice so systémom Kali Linux tak, aby na nich bolo možné realizovať vybrané úlohy. Takto pripravené virtuálne stroje budú poskytnuté vedúcemu práce už s kompletnou konfiguráciou tak, aby boli priamo využiteľné pre uskutočnenie popísaných úloh.

# 1. NÁVRH LABORATÓRNYCH ÚLOH

Jedným z hlavných cieľov tejto práce je vytvorenie kompletného zoznamu spolu celkom 11 laboratórnych úloh, ktoré budú predmetnou náplňou praktických počítačových cvičení príslušných k vyučovanému magisterskému predmetu *Návrh, správa a počítačových bezpečnosť sítí* (MPC-NSB).

Táto kapitola preto ďalej uvádza návrh osnovy laboratórnych cvičení, ktoré svojou obsahovou náplňou pokryjú vyučovanie v priebehu celého jedného semestra, počas ktorého je predmet vyučovaný, a následne tiež prehľad navrhovaných úloh spoločne s odôvodnením ich zaradenia do výučby s prihliadnutím na celkovú koncepciu tohto predmetu a obsahovú náplň prednášok.

## 1.1 Požiadavky na vytvorené laboratórne úlohy

Pri návrhu a vytváraní jednotlivých laboratórnych úloh je potrebné dodržať nasledovné požiadavky:

- vytvorené (resp. navrhnuté) laboratórne úlohy majú byť realizované vo forme počítačových cvičení;
- náplň vytvorených laboratórnych úloh musí byť v súlade s celkovou obsahovou náplňou predmetu a vhodným spôsobom nadväzovať na teoretickú problematiku diskutovanú v rámci prednášok;
- cieľom laboratórnych úloh je praktické overenie a upevnenie nadobudnutých znalostí;
- všetky úlohy musia byť realizovateľné na jedinom počítači s využitím jedného alebo viacerých virtuálnych strojov (typu VMware), pričom je pre všetky úlohy potrebné vytvoriť jednotnú metodiku a využiť spoločné programové vybavenie;
- doba nutná k vypracovaniu jednej úlohy musí odpovedať časovému intervalu v trvaní približne 90 minút;
- k navrhnutým úlohám musí byť spracovaný podporný materiál pre študentov v podobe návodu, ktorý okrem samotného postupu pre vypracovanie úlohy obsahovať ďalej tiež základnú teóriu súvisiacu s konkrétnou úlohou a potrebnú k jej vypracovaniu a záverečné testovacie otázky.

## 1.2 Ciele laboratórnych úloh

Vytvorený súbor laboratórnych úloh, ktoré budú študenti počas semestra postupne realizovať v rámci počítačových cvičení predmetu MPC-NSB, má za cieľ študentom ešte viac priblížiť, prehĺbiť a upevniť nadobudnuté znalosti týkajúce sa problematiky spadajúcej pod obsahovú náplň vyučovaných prednášok tohto predmetu a zároveň im

poskytnúť možnosť nadobudnúť praktické zručnosti v oblasti bezpečnosti sietí. Študenti si tak budú môcť vďaka praktickému nahliadnutiu do všetkých oblastí zahrnutých do problematiky návrhu, správy a bezpečnosti počítačových sietí upevniť svoje vedomosti a preniesť jednotlivé poznatky z teoretickej roviny do vlastnej praxe.

Pri vypracovávaní jednotlivých zadaných úloh by si mali študenti taktiež osvojiť základy práce s používanými nástrojmi, akými sú napr. softwarová virtualizačná platforma VMware či Linuxová distribúcia operačného systému Kali Linux (viď ďalej v kap. 3), a taktiež si ozrejmiť a vo väčšej miere prehĺbiť nadobudnuté teoretické poznatky súvisiace s riešenou problematikou týkajúcou sa bezpečnosti počítačových sietí.

### 1.3 Osnova predmetu MPC-NSB

Laboratórne úlohy, ktoré budú realizované v priebehu praktických počítačových cvičení, sú navrhované a štruktúrované s cieľom poskytnúť študentom priestor pre praktické overenie znalostí nadobudnutých počas prednášok. Cieľom vytvorených praktických úloh je poskytnúť študentom možnosť nahliadnuť na diskutovanú problematiku z inej perspektívy, prakticky aplikovať nadobudnuté poznatky v praxi a tým si zároveň ozrejmiť niektoré dôležité súvislosti v oblasti bezpečnosti sietí.

Nižšie je uvedená osnova predmetu<sup>1</sup> *Návrh, správa a bezpečnosť počítačových sietí*, ktorá uvádza prehľad tém (oblastí), ktorých problematika bude náplňou prednášok:

1. Kryptografia v počítačových sieťach
2. Fyzická vrstva a jej zabezpečenie
3. Spojová vrstva a jej zabezpečenie
4. Sieťová vrstva a jej zabezpečenie – protokoly
5. Sieťová vrstva a jej zabezpečenie – smerovanie
6. Transportná vrstva a jej zabezpečenie
7. Relačná vrstva a jej zabezpečenie
8. Aplikačná vrstva a jej zabezpečenie
9. Siete Wi-Fi a ich zabezpečenie
10. Anonymizačné siete
11. Návrh siete LAN a WAN
12. Perspektívne koncepty<sup>2</sup>
13. Rezerva

---

<sup>1</sup> Osnova je prebratá z karty predmetu *Návrh, správa a bezpečnosť počítačových sietí* dostupnej v rámci informačného systému Studis VUT, ktorá je aktuálna pre letný semester akademického roku 2024/25.

<sup>2</sup> Cieľom v poradí 12. prednášky je študentom predstaviť problematiku nových, aktuálnych trendov v oblasti bezpečnosti sietí, jej obsah preto vopred nie je jasne stanovený a môže sa v priebehu jednotlivých akademických rokov meniť. Ako príklad predošlých diskutovaných tém možno uviesť napr. protokol IPv6 alebo otázku smerovania v rozsiahlych transportných sieťach založených na protokole Ethernet apod.

Pri vytváraní kompletného súboru laboratórnych úloh pre počítačové cvičenia bude jedným z hlavných cieľov zachovať konzistentnosť obsahu prednášok a nadväzujúcich praktických počítačových cvičení.

## **1.4 Navrhovaný súbor 11 počítačových cvičení**

V nadväznosti na vyššie uvedenú osnovu predmetu bol navrhnutý nasledovný súbor laboratórnych úloh, ktoré pokrývajú jednotlivé témy vyplývajúce z obsahovej náplne predmetu:

### **1. Kryptografické metódy a ich implementácia v sieťach**

Cieľ: Implementácia a analýza symetrických a asymetrických šifrovacích metód (napr. AES, RSA). Študenti budú generovať kľúče pre kryptosystém a následne šifrovať/dešifrovať komunikáciu medzi dvoma virtuálnymi strojmi.

Nástroje: OpenSSL, GPG

### **2. Zabezpečenie fyzickej vrstvy siete**

Cieľ: Simulácia zraniteľností a ochrany fyzickej vrstvy, napríklad ochrana pred útokmi typu *wiretapping* (odpočúvanie). Ukážka možností zabezpečenia na úrovni fyzickej vrstvy, akými sú napr. vhodné mechanizmy pre fyzické zabezpečenie a kontrola prístupu k zariadeniam apod.

Nástroje: Wireshark, tcpdump, fyzické vrstvy protokolov

### **3. Zabezpečenie spojovej vrstvy siete**

Cieľ: Ochrana pred útokmi typu ARP spoofing, implementácia zabezpečenia na úrovni MAC adresácií a VLAN.

Nástroje: arpspoof, ettercap, Wireshark

### **4. Zabezpečenie sieťovej vrstvy – analýza IP protokolov**

Cieľ: Analýza a ochrana IP komunikácie, detekcia a prevencia útokov typu IP spoofing, konfigurovanie IPsec.

Nástroje: hping3, Wireshark, strongSwan, tcpdump

### **5. Zabezpečenie dynamických smerovacích protokolov**

Cieľ: Simulácia útokov na dynamické smerovacie protokoly (napr. RIP, OSPF) a implementácia zabezpečenia v rámci smerovacích protokolov.

Nástroje: Quagga, ospfd, Wireshark

### **6. Pokročilé filtrovanie a analýza sieťovej komunikácie pomocou stavového a NGFW firewallu**

Cieľ: Základná i pokročilá konfigurácia firewallu pre filtrovanie komunikácie pomocou nástroja iptables a simulácia moderného aplikačného NGFW.

Nástroje: iptables, Suricata

## **7. Zabezpečenie transportnej vrstvy**

Cieľ: Ochrana transportnej vrstvy pred útokmi ako TCP SYN flood, analýza SSL/TLS a ich implementácia.

Nástroje: OpenSSL, Wireshark, hping3

## **8. Autentizácia pomocou EAP a RADIUS**

Cieľ: Implementácia autentizačného RADIUS servera a jeho konfigurácia pre vytvorenie zabezpečeného pripojenia k sieti.

Nástroje: FreeRADIUS, OpenSSL

## **9. Zabezpečenie aplikačnej vrstvy**

Cieľ: Skúmanie zraniteľností aplikačných protokolov (napr. DNS, HTTP apod.) a ochrana proti útokom typu SQL Injection a XSS.

Nástroje: Burp Suite, Nikto, OWASP ZAP

## **10. Pokročilé zabezpečenie Wi-Fi sietí**

Cieľ: Zabezpečenie bezdrôtových sietí pred útokmi na autentizačné protokoly ako napr. WPA/WPA2 *cracking*, implementácia pokročilých konceptov pre zabezpečenie Wi-Fi sietí, analýza WPS zraniteľnosti.

Nástroje: aircrack-ng, Wireshark, wifite, aircrack-ng

## **11. Anonymizačné siete**

Cieľ: Práca s anonymizačnými sieťami, analýza fungovania Tor siete, výhody a nevýhody anonymizácie.

Nástroje: Tor, Wireshark, Onion services

Podrobnejší popis cieľov jednotlivých úloh a nevyhnutných teoretických podkladov pre vypracovanie a úplnú praktickú realizáciu danej úlohy bude prehľadne uvedený v samostatných oddieloch zvlášť pre každú z úloh v rámci nasledujúcej kap 2.

K laboratórnym úlohám č. 3, 4, 8 a 11 bude následne vytvorený kompletný, detailne rozpracovaný a podrobný návod poskytujúci študentom plnú oporu pri ich praktickej realizácii. Jednotlivé návody k týmto úlohám budú následne doplnené samostatne ako prílohy k tejto záverečnej práci. Zároveň bude ku každej z vybraných úloh vytvorená aj dokumentácia pre vyučujúceho popisujúca nielen ciele samotnej úlohy, ale najmä očakávané výstupy samostatnej práce študentov. Výber týchto konkrétnych úloh bol motivovaný viacerými faktormi:

- pokrývajú rôzne oblasti zabezpečenia počítačových sietí (spojová a sieťová vrstva, problematika autentizácie klienta, využitie anonymizačných sietí, ...),
- sú technologicky a didakticky rôznorodé a umožňujú praktické využitie rôznych nástrojov a techník,
- poskytujú študentom dostatočný priestor na pochopenie podstaty sieťových útokov a zároveň na osvojenie si rôznych metód ochrany.

Zámerom bude pripraviť metodicky ucelené a didakticky hodnotné podklady, ktoré poslúžia nielen ako sprievodca pre prácu priamo vo vyučovacím procese, ale aj ako samostatný študijný materiál, s dôrazom na pochopenie kontextu a nadobudnutie praktických zručností. O vytvorených návodoch bude viac písané neskôr v kap. 5.

#### 1.4.1 Výber laboratórnych úloh

Nižšie uvedená tabuľka 1.1 uvádza pre jednotlivé úlohy stručné vyjadrenie k dôvodom, prečo boli dané úlohy zaradené do konečného zoznamu vytváraného pre počítačové cvičenia predmetu MPC-NSB.

Tabuľka 1.1 Zdôvodnenie výberu a zaradenia jednotlivých úloh do finálneho zoznamu pre počítačové cvičenia

| Názov úlohy                                                             | Zdôvodnenie výberu                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Kryptografické metódy a ich implementácia v počítačových sieťach</b> | <ul style="list-style-type: none"> <li>• oboznámenie študentov so základnými kryptografickými mechanizmami pre zabezpečenie prenosov dát proti odposluchu</li> <li>• uvedenie zásadných rozdielov medzi symetrickou a asymetrickou kryptografiou a z nich prameniace oblasti praktického využitia jednotlivých mechanizmov v dnešných počítačových sieťach</li> </ul>                                                                                                                          |
| <b>Zabezpečenie fyzickej vrstvy siete</b>                               | <ul style="list-style-type: none"> <li>• študenti si osvoja základy práce s populárnym a široko využívaným nástrojom pre odposluch, zachytávanie a analýzu dátovej komunikácie Wireshark</li> </ul>                                                                                                                                                                                                                                                                                            |
| <b>Zabezpečenie spojovej vrstvy siete</b>                               | <ul style="list-style-type: none"> <li>• ozrejmienie poznatkov z oblasti problematiky adresovania na úrovni spojovej a sieťovej vrstvy RM ISO/OSI a vysvetlenie princípov fungovania protokolu ARP používaného k prekladu medzi adresami druhej a tretej úrovne</li> <li>• praktická laboratórna úloha umožní študentom realizovať simulovaný útok z pohľadu útočníka a tým i demonštrovať hrozby prameniace z možnosti manipulácie záznamov obsiahnutých v ARP tabuľkách zariadení</li> </ul> |
| <b>Analýza sieťových IP protokolov</b>                                  | <ul style="list-style-type: none"> <li>• oboznámenie študentov s možnými hrozbami na úrovni sieťovej vrstvy RM ISO/OSI a následná praktická simulácia vybraných druhov útokov</li> <li>• predstavenie a podrobný popis bezpečnostného rozšírenia IPsec široko využívaného pre zabezpečenie dátových prenosov v dnešných počítačových sieťach,</li> </ul>                                                                                                                                       |

|                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                       | <p>vysvetlenie princípov jeho fungovania a možností využitia</p> <ul style="list-style-type: none"> <li>• osvojenie praktických zručností pre prácu s knižnicami strongSwan a/alebo OpenSwan pre implementáciu a konfiguráciu IPsec-u</li> </ul>                                                                                                                                                                                                                                                                                                        |
| <b>Zabezpečenie dynamických smerovacích protokolov</b>                                | <ul style="list-style-type: none"> <li>• študenti budú oboznámení so základnými princípmi dynamických smerovacích protokolov používaných v počítačových sieťach pre účely smerovania paketových prenosov</li> <li>• v praktickej časti budú demonštrované riziká manipulácie s obsahom informácií v smerovacích tabuľkách, vďaka čomu si študenti ozrejmi dôležitosť správnosti týchto záznamov pre bezproblémové fungovanie procesu smerovania</li> </ul>                                                                                              |
| <b>Pokročilé filtrovanie a analýza komunikácie pomocou stavového a NGFW firewallu</b> | <ul style="list-style-type: none"> <li>• študenti si osvoja poznatky v oblasti filtrovania nežiadúcej komunikácie</li> <li>• praktická časť bude zameraná na základnú i pokročilejšiu konfiguráciu pravidiel umožňujúcich filtrovanie dátových tokov najmä s využitím nástroja iptables, kde budú mať študenti možnosť realizovať rôzne kombinácie nastavení s cieľom dosiahnuť požadované fungovanie firewallu</li> <li>• študenti si ďalej vyskúšajú i pokročilejšie metódy filtrácie, a to vďaka simulácii NGFW pomocou nástroja Suricata</li> </ul> |
| <b>Zabezpečenie transportnej vrstvy</b>                                               | <ul style="list-style-type: none"> <li>• oboznámenie študentov s rizikami na úrovni transportnej vrstvy RM ISO/OSI s následnou praktickou simuláciou typických útokov na transportné prenosové protokoly</li> <li>• študenti opakovane využijú nadobudnuté praktické zručnosti pri využití sieťového analyzátoru Wireshark, kde budú môcť aplikovať i jeho pokročilejšie funkcionality</li> <li>• študenti si taktiež upevnia znalosti z predchádzajúcej úlohy venovanej konfigurácii pravidiel v iptables</li> </ul>                                   |
| <b>Autentizácia s využitím protokolu EAP a RADIUS</b>                                 | <ul style="list-style-type: none"> <li>• rozšírenie teoretických poznatkov v oblasti autentizácie pomocou protokolu RADIUS využívaného v značnej miere v dnešných sieťach</li> </ul>                                                                                                                                                                                                                                                                                                                                                                    |

|                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                       | <ul style="list-style-type: none"> <li>• študenti budú môcť samostatne implementovať vlastný RADIUS server a s jeho využitím následne prakticky odsledovať priebeh autentizácie pomocou protokolu EAP-TLS</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Zabezpečenie aplikačnej vrstvy</b> | <ul style="list-style-type: none"> <li>• študenti nadobudnú znalosti v oblasti najznámejších zraniteľností webových aplikácií</li> <li>• teoretické znalosti budú upevnené vďaka praktickej realizácii vybraných typov útokov, čo študentom poskytne priestor k lepšiemu pochopeniu princípov ich uskutočnenia</li> <li>• študenti budú môcť využiť taktiež známy nástroj Burp Suite vhodný pre penetračné testovanie s cieľom odhaliť významné zraniteľnosti vytvoreného webového servera, resp. aplikácie</li> </ul>                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Zabezpečenie Wi-Fi sietí</b>       | <ul style="list-style-type: none"> <li>• oboznámenie študentov s existujúcimi možnosťami zabezpečenia bezdrôtových Wi-Fi sietí s dôrazom na upozornenie na zásadné nedostatky jednotlivých používaných protokolov (WEP, WPA, WPA2, ... ) a vysvetlenie rozdielov medzi nimi</li> <li>• cieľom praktickej činnosti študentov bude nielen samotná realizácia útokov (simulovaný útok s cieľom prelomenia prístupového hesla k Wi-Fi sieti), ale následne aj ich mitigácia implementáciou vhodných ochranných opatrení, ako je aktivácia 802.11w, vypnutie WPS, filtrovanie MAC adries alebo skrytie SSID, s následnou analýzou dosiahnutých výsledkov zabezpečenia a návrhom ďalších opatrení k zvýšeniu bezpečnosti</li> <li>• ďalej bude opäť využitý i sieťový analyzátor Wireshark k precvičeniu už predtým využívaných funkcionalít za účelom analýzy zachytenej dátovej komunikácie</li> </ul> |
| <b>Anonymizačné siete</b>             | <ul style="list-style-type: none"> <li>• študenti získajú teoretické znalosti v oblasti využívania a správy anonymizačných sietí</li> <li>• nadobudnuté teoretické poznatky si budú môcť študenti prakticky overiť pri vlastnej inštalácii a konfigurácii siete ToR</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |



### 1.4.2 Doplnenie zoznamu úloh s možnosťou budúceho využitia

Nakoľko problematika bezpečnosti počítačových sietí je pomerne rozsiahla a zahŕňa niekoľko síce nezávislých, no pritom vzájomne súvisiacich oblastí, výber vyššie uvedených 11 laboratórnych úloh nie je ani zďaleka konečný. Jedná sa o úlohy, ktoré boli vybraté pre počítačové cvičenia keďže svojou náplňou, či už v zmysle teoretického pochopenia problematiky, ale aj praktického overovania nadobudnutých poznatkov, dokážu plne pokryť tematické celky obsiahnuté v osnove predmetu MPC-NSB. Avšak celkový počet možných laboratórnych úloh, ktoré by dokázali riešenie problematiky a jej pochopenie prakticky overovať, nekončí na hodnote 11, a preto určite stojí za zmienku, že tento uvedený výber úloh je možné na základe potreby a/alebo preferencií v budúcnosti modifikovať, resp. rozšíriť.

Nižšie bude preto nasledovať ešte predstavenie ďalších možných laboratórnych úloh<sup>3</sup> zameraných na overenie teoretických znalostí z oblasti bezpečnosti počítačových sietí, ktoré by bolo z praktického hľadiska možné realizovať a s pomocou ktorých by bolo tak isto možné veľmi vhodným spôsobom ilustrovať diskutovanú problematiku. Avšak pre účely tejto práce nakoniec do finálneho súboru jedenástich laboratórnych úloh pre počítačové cvičenia predmetu MPC-NSB nakoniec zahrnuté neboli.

- **Úloha: Vzdialená správa zariadení s využitím protokolu SSH**

Cieľ: Inštalácia SSH servera, oboznámenie sa s možnosťami základnej konfigurácie, a to najmä so zameraním na nastavenie kryptografického zabezpečenia a autentizácie. Vzdialená správa zariadenia a diskusia nad výhodami využitia protokolu SSH v porovnaní s protokolom Telnet.

Nástroje: TCP Wrappers, Fail2Ban

- **Úloha: Zabezpečenie relačnej vrstvy**

Cieľ: Analýza protokolov na relačnej vrstve, zabezpečenie pri reláciách HTTP a FTP (napr. SSL/TLS).

Nástroje: OpenSSL, Wireshark, Burp Suite

- **Úloha: Návrh a simulácia siete LAN a WAN sietí**

Cieľ: Vytvorenie vlastného návrhu malej LAN siete s VLAN, a WAN spojení, implementácia základných bezpečnostných prvkov (firewall, VPN).

Nástroje: GNS3, OpenVPN, iptables

---

<sup>3</sup> Uvedené úlohy môžu byť v budúcnosti využité v prípade potreby aktualizácie, resp. doplnenia či obmeny vytvoreného návrhu osnovy počítačových cvičení predmetu MPC-NSB. Prípadne by tieto úlohy mohli predstavovať tiež „voliteľné“, resp. doplnujúce úlohy, ktoré by prípadní záujemcovia z radov študentov mohli realizovať vo vlastnom záujme nad rámec požadovanej obsahovej náplne počítačových cvičení s cieľom ďalšieho rozšírenia svojich znalostí a praktických zručností nad rámec požiadaviek kladených v priebehu vyučovacieho procesu.

- **Úloha: Súbor úloh CTF**

Cieľ: Riešenie praktických úloh typu CTF (*Capture the Flag*) zameraných na problematiku bezpečnosti sietí, ktoré môžu spočívať napr. v prelomení hesla bezdrôtovej WiFi siete po predložení záznamu výmeny EAPOL správ v priebehu *4-way handshake* pri autentizácii klienta, získaní a/alebo modifikácii súboru na vzdialenom serveri alebo cielenej zmene jeho konfigurácie apod.

Nástroje: v závislosti na type a zameraní konkrétnej CTF úlohy

## 1.5 Štruktúra počítačových cvičení

Pre všetky vytvorené laboratórne úlohy riešené postupne počas celého semestra bude použitá jednotná metodika. Táto metodika bude zahŕňať nielen postup vypracovania samotnej úlohy, ale tiež teoretický úvod poskytujúci študentom možnosť bližšieho zoznámenia sa s problematikou. Súčasťou budú tiež samostatné úlohy pre študentov a záverečné otázky podmieňujúce zamyslenie sa a prípadnú diskusiu. Preto každá laboratórna úloha bude obsahovať nasledovné:

- zadanie problému, resp. stručné uvedenie problematiky, ktorej sa príslušná laboratórna úloha venuje,
- teoretické východiská, resp. stručný teoretický úvod ozrejmujúci základné fakty týkajúce sa riešenej problematiky, ktorých pochopenie je nevyhnutné k bezproblémovej a úspešnej realizácii úlohy,
- postup praktickej úlohy, príp. vytvorenia simulácie,
- analýza riešenia, vypracovanie samostatných úloh,
- kontrolné otázky a diskusia k riešeniu,
- záver (zhrnutie úlohy, prípadne doplňujúce odporúčania, návrhy k zaisteniu bezpečnosti).

## 1.6 Použité nástroje

Všetky vyššie uvedené laboratórne úlohy budú počas semestra v rámci počítačových cvičení predmetu MPC-NSB realizované **s využitím virtuálnych strojov**. Pre účely vytvorenia virtualizovaného prostredia bude využívaná **platforma VMware**, pričom konečný počet virtuálnych strojov potrebných pre kompletnú a úspešnú realizáciu laboratórnej úlohy sa môže pre jednotlivé úlohy líšiť<sup>4</sup>. Virtuálne stroje budú využívať OS Linux, a to konkrétne **systém Kali Linux**. Táto Linuxová distribúcia bola zvolená z dôvodu jej vhodnosti pre realizáciu väčšiny navrhnutých laboratórnych úloh, nakoľko je určená pre penetračné testovanie, etický hacking, vykonávanie reverzného inžinierstva

---

<sup>4</sup> Typicky bude však potrebné využiť min. 2, v niektorých prípadoch i viac (typicky 3) VMs.

a pokročilého testovania sieťovej bezpečnosti, ako uvádzajú i oficiálne stránky vývojárskej spoločnosti tohto produktu *Offensive-security* [1].

Distribúcia Kali Linux už samotná obsahuje niekoľko integrovaných nástrojov, balíčkov a ďalších rozšírení, ktoré je možné vhodným spôsobom využiť pri realizácii jednotlivých úloh. Podrobnejší popis vybraných použitých nástrojov a potrebného programového vybavenia (*software*) je uvedený v kap. 3.

## 2. POPIS LABORATÓRNYCH ÚLOH

V tejto kapitole bude uvedený popis jednotlivých navrhnutých laboratórnych úloh, ktorých prehľad bol predstavený v kap. 1.4. Pre každú z navrhnutých úloh bude objasnený a presne špecifikovaný jej zámer, resp. cieľ, ktorý má byť dosiahnutý, ďalej nutné teoretické východiská, ktoré by mali byť študentom objasnené, aby získali znalosti potrebné pre následnú praktickú realizáciu danej úlohy. Taktiež bude stručne popísaný priebeh praktickej časti, napr. popis inštalácie a konfigurácie zariadení, simulácie sieťových útokov apod. Uvedené budú i používané nástroje..

Takto vytvorený popis vytvára predbežnú predstavu o priebehu počítačového cvičenia venovanému riešeniu danej úlohy. Zatiaľ sa však jedná len o predbežné návrhy, ktoré budú ďalej špecificky upresnené pri vytváraní kompletných, podrobných návodov k príslušným laboratórnym úlohám.

Ďalej uvedené návrhy sú zostavené na základe vlastných praktických skúseností autorky tejto práce, ktoré nadobudla počas absolvovania niekoľkých predmetov zameraných na problematiku bezpečnosti sietí a komunikačných technológií počas svojich vysokoškolských štúdií v univerzitnom prostredí.

### 2.1 Laboratórna úloha č. 1

#### Kryptografické metódy a ich implementácia v sieťach

V rámci laboratórnej úlohy budú študenti oboznámení so základnými kryptografickými technikami zabezpečujúcimi dôvernosť, integritu a autentickosť prenášaných dát, resp. celej dátovej komunikácie v počítačových sieťach.

Teoretický úvod úlohy bude venovaný vysvetleniu základným princípom šifrovania, a to konkrétne vysvetleniu zásadných rozdielov medzi symetrickými a asymetrickými šifrovacími technikami, princípu ich fungovania a zoznámeniu s hlavnými predstaviteľmi, a to asymetrickým kryptosystémom RSA a štandardom symetrickej blokovej šifry AES. Pozornosť bude venovaná i súvislosti miery bezpečnosti v závislosti na vhodne zvolenej dĺžke šifrovacieho, resp. dešifrovacieho kľúča. Spomenuté budú aj aktuálne požiadavky na vhodnú dĺžku používaných šifrovacích kľúčov pre dosiahnutie dostatočnej miery kryptografického zabezpečenia. Súčasťou teoretického úvodu bude taktiež vysvetlenie problematiky digitálneho podpisu.

V praktickej časti si študenti vyskúšajú implementáciu uvedených symetrických i asymetrických kryptografických mechanizmov s využitím knižnice umožňujúcej použitie kryptografických nástrojov OpenSSL<sup>5</sup>. Využívané budú dva virtuálne stroje s OS Kali Linux vo virtualizačnej platforme VMware. Vhodným rozšírením úlohy by

---

<sup>5</sup> OpenSSL je *open-source* knižnica pre prácu s protokolmi TLS/SSL vhodná pre implementáciu zabezpečenia internetovej komunikácie. Umožňuje generovanie a správu kryptografických kľúčov (privátnych a verejných), šifrovanie a dešifrovanie súborov a vytváranie certifikátov

mohlo byť porovnanie implementácie šifrovania s využitím OpenSSL a iných dostupných knižníc, napr. GPG<sup>6</sup>. Overenie správnosti použitého šifrovania v rámci prebiehajúcej komunikácie môže byť realizované napr. s využitím nástroja Wireshark s následnou analýzou obsahu jednotlivých zachytených dátových jednotiek (paketov) a ich porovnaním pred a po aplikácii kryptografického zabezpečenia.

V rámci záverečnej diskusie je vhodné kontrolné otázky smerovať k vysvetleniu a správnej interpretácii logov zachytenej komunikácie, ďalej k vysvetleniu podmienok, resp. požiadaviek pre použitie kľúčov a tiež k objasneniu dôvodov praktickej realizácie dnešných komplexných kryptosystémov založených na využití vhodnej kombinácie symetrických i asymetrických algoritmov (napr. TLS/SSL).

## 2.2 Laboratórna úloha č. 2

### Zabezpečenie fyzickej vrstvy siete

Cieľom tejto úlohy je demonštrovať študentom bezpečnostné riziká, ktoré sa môžu vyskytnúť na úrovni fyzickej vrstvy počítačových sietí. Jedná sa najmä o útoky, ktoré cielia na narušenie dôvernosti prebiehajúcej komunikácie, t. j. odposluch (*wiretapping*) prenosu na fyzickom prenosovom médiu.

Teoretický úvod bude venovaný možnostiam realizácie fyzickej prenosovej vrstvy v dnešných prenosových sieťach, a to konkrétne rôznym druhom používaných prenosových médií. Taktiež bude značná časť pozornosti venovaná hrozbám na úrovni fyzickej vrstvy.

V praktickej časti laboratórnej úlohy si študenti vyskúšajú simuláciu odposluchu komunikácie. Využitie budú celkom tri virtuálne stroje -- komunikácia bude prebiehať v prostredí vytvorenej virtuálnej siete medzi dvoma strojmi (A a B), tretí stroj (C) bude slúžiť k simulácii útočnickovho zariadenia, ktorého cieľom je narušiť najmä dôvernosť prebiehajúcej dátovej komunikácie medzi legitímnymi zariadeniami. Na virtuálnom stroji útočníka bude možné s využitím sieťového analytického nástroja `tcpdump` zachytiť odosielané dátové jednotky, ktoré budú následne v prostredí nástroja Wireshark analyzované z hľadiska ich obsahu.

Súčasťou tejto laboratórnej úlohy bude tiež záverečná diskusia, ktorej cieľom bude aplikácia nadobudnutých znalostí pri riešení otázky návrhu vhodného fyzického zabezpečenia objektov či rôznych typov zariadení.

---

<sup>6</sup> GPG je nástroj umožňujúci šifrovanie správ a súborov pomocou verejných a privátnych kľúčov, založený na štandarde OpenPGP. Používa sa najmä na ochranu e-mailovej komunikácie a overovanie integrity a pôvodu odosielaných správ. GPG umožňuje okrem samotného šifrovania súborov a správ taktiež vytváranie digitálnych podpisov.

## 2.3 Laboratórna úloha č. 3

### Zabezpečenie spojovej vrstvy siete

Laboratórna úloha bude venovaná problematike identifikácie a adekvátnej ochrany proti útokom realizovaným na úrovni spojovej vrstvy so zameraním na útok cielený na podvrhnutie fyzickej adresy zariadenia a s následným presmerovaním komunikácie na zariadenie útočníka, tzv. **ARP spoofing**.

Teoretický úvod objasní študentom princípy fungovania prepínača (*switch*) ako základného prepojovacieho prvku pracujúceho na úrovni spojovej vrstvy prenosových sietí, priblíži princípy adresácie zariadení pomocou fyzických adries v lokálnej sieti, pričom bude dôraz kladený primárne na použitie MAC adries v prenosových sieťach založených na protokole Ethernet, a nakoniec študentom vysvetlí základný princíp a možnosti realizácie útoku *ARP spoofing* a možnosti obrany proti tomuto typu útoku.

V praktickej časti úlohy študenti realizujú sieťový útok *ARP spoofing* s využitím nástrojov *arp spoof* a *ettercap* dostupných v používanom OS Kali Linux. Za týmto účelom budú využité opäť tri virtuálne stroje – komunikácia bude prebiehať v prostredí vytvorenej virtuálnej siete medzi dvoma virtuálnymi strojmi (A a B), tretí virtuálny stroj (C) bude slúžiť k simulácii útočnickovho zariadenia, prostredníctvom ktorého bude realizovaný samotný útok. V prvej časti úlohy bude vysvetlené, ako postupovať pri „otrave“ ARP tabuľky mapujúcej pripojené zariadenia (resp. ich MAC adresy) na konkrétne porty zariadenia, tzv. *ARP cache poisoning*. Nadväzujúca samostatná úloha bude spočívať vo využití sieťového nástroja *ettercap* k realizácii útoku *ARP spoofing*. Po úspešnom vykonaní útoku bude následne na stroji „útočníka“ využitý program Wireshark k zachyteniu paketov komunikácie medzi legítimnými účastníkmi komunikácie (stroje A a B), ktorá v prípade úspešného útoku prechádza i cez útočnickovo zariadenie (C).

Záverečná diskusia bude zameraná na možnosti obrany proti uvedenému typu sieťového útoku. Študenti dostanú nakoniec priestor i k vysvetleniu ďalších útokov hroziacich na úrovni spojovej vrstvy, ako napr. *MAC flooding* apod.

## 2.4 Laboratórna úloha č. 4

### Zabezpečenie sieťovej vrstvy – analýza IP protokolov

Študenti sa v rámci tejto laboratórnej úlohy oboznámia s bezpečnostnými hrozbami na sieťovej vrstve, akou je napr. útok *IP spoofing*, a tiež sa naučia implementovať základné ochranné mechanizmy, kedy spomedzi najpoužívanejších bude venovaná pozornosti najmä bezpečnostnému rozšíreniu IP protokolu známemu ako IPsec.

Teoretický úvod priblíži študentom jednak možné bezpečnostné riziká na úrovni sieťovej vrstvy, no taktiež študentov oboznámi s bezpečnostným rozšírením IPsec zaisťujúcim autentizáciu komunikujúcich strán, ustanovenie kľúčov pre šifrovanie

a pečatenie správ prenosu a zaistenie dôvernosti (resp. autenticity) prenášaných dátových jednotiek, ozrejmi základný princíp fungovania, jeho jednotlivé súčasti (protokoly), spôsoby realizácie a možnosti implementácie.

V praktickej časti študenti realizujú na jednom virtuálnom stroji s využitím nástroja **hping3** simuláciu útoku typu IP *spoofing*. Avšak hlavnou náplňou počítačového cvičenia bude implementácia protokolu IPsec v transportnom režime medzi dvoma virtuálnymi strojmi s OS Kali Linux s cieľom zaistiť šifrovanie a autentizáciu komunikácie na sieťovej vrstve. Pre konfiguráciu protokolu IPsec môžu byť využité dostupné nástroje knižnice **strongSwan** alebo **OpenSwan**. Následne bude využitý sieťový analytický nástroj Wireshark k zachyteniu komunikácie (pred a po konfigurácii zabezpečeného spojenia prostredníctvom IPsec) medzi virtuálnymi strojmi a analýze a porovnaniu zachytených paketov so šifrovaním a bez použitia šifrovania.

Záverečná diskusia bude vychádzať z vykonanej analýzy zachytenej komunikácie a môže byť ďalej rozšírená o otázky súvisiace so zhodnotením výhod a nevýhod, príp. určitých nedostatkov či obmedzení pri použití protokolu IPsec.

## 2.5 Laboratórna úloha č. 5

### Zabezpečenie dynamických smerovacích (*routing*) protokolov

Cieľom laboratórnej úlohy bude objasniť výskyt možných zraniteľností smerovacích protokolov (napr. RIP a OSPF) a následne implementovať vhodné ochranné mechanizmy proti útokom na smerovanie.

Teoretický základ priblíži študentom základný princíp fungovania dynamických smerovacích protokolov a ďalej študentov oboznámi s existujúcimi zraniteľnosťami. Vysvetlený bude napr. útok Route Hijacking či ďalšie bezpečnostné hrozby, na ktoré sú zmienené smerovacie protokoly zraniteľné (*spoofing*, t. j. podvrhnutie adresy, , tzv. *replay* útoky známe tiež ako opakovaním správ, DoS útoky narušujúce dostupnosť sieťových zariadení atď.).

V praktickej časti študenti využijú nástroj **Quagga** pre vytvorenie simulácie útoku na smerovanie. Cieľom vytvorenej simulácie je demonštrovať, akým spôsobom môže útočník manipulovať s informáciami uloženými v smerovacích tabuľkách sieťových prvkov (smerovačov), a to na základe rozposielania nesprávnych, falošných smerovacích informácií medzi týmito zariadeniami. Po realizácii samotného útoku študenti sledujú zmeny záznamov obsiahnutých v smerovacích tabuľkách jednotlivých smerovačov. Samostatná úloha študentov bude spočívať v implementácii vhodných mechanizmov umožňujúcich ochranu dynamických smerovacích protokolov pred uvedeným typom útokom (napr. konfigurácia autentizácie v rámci protokolu OSPF pomocou MD5).

Záverečná diskusia a kontrolné otázky bude nadväzovať na riešenie samostatnej časti úlohy, diskutovaná bude samotná autentizácia v rámci implementácie dynamických smerovacích protokolov a tiež limity uvedených autentizačných mechanizmov.

## 2.6 Laboratórna úloha č. 6

### **Pokročilé filtrovanie komunikácie pomocou stavového a NGFW firewallu**

Laboratórna úloha bude zameraná na základnú konfiguráciu nastavení firewallu na úrovni sieťovej vrstvy pomocou nástroja **iptables** a na pokročilé možnosti filtrovania komunikácie s využitím **aplikačného NGFW**.

Teoretický úvod priblíži študentom dôvody použitia a možnosti konfigurácie firewallu pre filtrovanie prenášanej komunikácie, a to súčasne s uvedením rôznych typov firewallu, objasnením princípov ich fungovania a vzájomným porovnaním. V ďalšej časti bude nasledovať vysvetlenie práce s nástrojom **iptables**, ktoré je nevyhnutné pre úspešné praktické absolvovanie laboratórnej úlohy. Vysvetlené bude použitie reťazcov (INPUT, OUTPUT, FORWARD) a tiež možných akcií (ACCEPT, DROP, REJECT, LOG). V rámci teórie bude tiež stručne zhrnutý princíp použitia techniky prekladu adries NAT, resp. DNAT a SNAT. Zmienené bude rovnako tak i použitie modernejších NGFW, ktorých nasadenie v rámci počítačových sietí umožňuje realizovať i pokročilejšie metódy filtrácie, a to až na úrovni aplikačnej vrstvy. Uvedené budú ich hlavné výhody v porovnaní so stavovým či paketovým typom FW.

V praktickej časti študenti aplikujú nadobudnuté teoretické znalosti a vyskúšajú si rôzne nastavenia stavového firewallu s použitím nástroja **iptables**, napr. blokovanie komunikácie na konkrétnom porte (resp. portoch), povolenie či naopak zamietnutie prístupu len z určitej IP adresy (resp. adries), nastavenie pravidiel pre limitovanie počtu nových pripojení v stanovenom časovom intervale apod. Následne sa zamerajú na možnosti pokročilej filtrácie dátovej komunikácie, kde pomocou nástroja Suricata, simulujú funkcionality NGFW.

Samostatná úloha bude spočívať v konfigurácii pravidiel na stavovom FW, ktoré povolia iba komunikáciu cez zabezpečené HTTPS (na porte 443) z konkrétnej IP adresy. Následne v rámci simulácie funkčnosti NGFW študenti vytvoria pravidlá pre detekciu prístupu k zakázaným webovým stránkam (nastavenie napr. blokovania URL adries obsahujúcich určitý textový reťazec).

Jednotlivé aplikované nastavenia firewallu študenti priebežne monitorujú a kontrolujú pomocou vhodného zobrazenia aktuálne platných pravidiel v prostredí systémovej konzoly (terminálu). Súčasťou zadanej samostatnej úlohy môže byť i overenie výkonnosti oboch použitých typov FW (stavový FW **iptables** a NGFW v prostredí nástroja Suricata) počas generovania veľkého množstva paketov pomocou nástroja **hping3**, s ktorým sa študenti prakticky oboznámili v laboratórnej úlohe č. 4.

V rámci diskusie študenti demonštrujú a vysvetlia použité pravidlá nastavenia firewallu a v príp. potreby realizujú ich vhodné úpravy na základe dodatočných pokynov vyučujúceho. Kontrolné otázky môžu byť zamerané na uvedenie a nevýhod použitia **iptables** v porovnaní s hardvérovým firewallom.



## 2.7 Laboratórna úloha č. 7

### Zabezpečenie transportnej vrstvy

Cieľom úlohy je oboznámiť študentov s možnými útokmi na transportnej vrstve a s rôznymi možnosťami implementácie vhodných bezpečnostných opatrení proti týmto útokom. Študenti sa naučia zabezpečiť komunikáciu na transportnej vrstve pomocou protokolu TLS.

Úvodnej teoretická časť študentom prehľadne zhrnie základné znalosti týkajúce sa transportných protokolov TCP a UDP a tiež možnosti ich zabezpečenia. Väčšia pozornosť bude venovaná protokolu TCP a najmä možným útokom cieľovým na tento transportný protokol (napr. SYN flood). Súčasťou teoretického úvodu bude taktiež vysvetlenie zabezpečenia prenosu na úrovni transportnej vrstvy s využitím protokolu TLS.

Úlohou študentov v praktickej časti laboratórnej úlohy bude vhodným spôsobom nakonfigurovať webový server za účelom realizácie šifrovanej komunikácie a následne overiť úspešné zabezpečenie. Pre analýzu priebehu zachytenej šifrovanej komunikácie bude využitý sieťový analyzátor Wireshark. V ďalšej časti bude realizovaná simulácia **SYN flood** útoku s využitím nástroja hping. Samostatná úloha študentov bude následne spočívať v aplikácii ochranných opatrení proti tomuto typu útoku na strane servera, a to napr. správnou konfiguráciou nastavení pre použitie TCP SYN cookies alebo vhodnou úpravou pravidiel v iptables.

Záverečné kontrolné otázky budú spočívať primárne vo vysvetlení fungovania použitých obranných mechanizmov a k porovnaniu výsledkov simulácie so stavom bez ich použitia..

## 2.8 Laboratórna úloha č. 8

### Autentizácia pomocou EAP a RADIUS

Laboratórna úloha bude venovaná pokročilým možnostiam autentizácie s využitím frameworku EAP a autentizačného protokolu RADIUS. Študenti v rámci tejto úlohy nadobudnú praktické skúsenosti s implementáciou servera RADIUS pre zaistenie zabezpečeného pripojenia v sieti.

Teoretický úvod k laboratórnej úlohe poskytne študentom základnú charakteristiku frameworku autentizačných protokolov EAP a prehľad rôznych autentizačných metód (EAP-TLS, EAP-TTLS, EAP-MD5, EAP-PSK, PEAP atď.) a taktiež jeho porovnanie s protokolom CHAP, príp. PAP. Zároveň ich uvedie do problematiky riadenia prístupu, pričom bude ako hlavný predstaviteľ AAA protokolov<sup>7</sup> uvedený protokol RADIUS.

---

<sup>7</sup> AAA protokoly sú protokoly používané za účelom riadenia prístupu v počítačových sieťach. Skratka AAA (z angl. *authentication, authorization and accounting*) zastupuje tri základné funkcie, a to postupne autentizáciu, autorizáciu a účtovanie.

V praktickej časti študenti nainštalujú a vhodne konfigurujú RADIUS server. Pre tento účel je možné v OS Kali Linux využiť integrovaný server **FreeRADIUS**. Po úspešnej inštalácii servera bude nasledovať požadovaná konfigurácia. Následne bude virtuálny stroj zastupujúci RADIUS server využitý pre zaistenie autentizácie používateľov (resp. ďalších VMs) v sieti<sup>8</sup>. Študenti si ďalej overia použitie **EAP-TLS** k autentizácii používateľov pomocou certifikátov, a to s použitím knižnice OpenSSL.

Diskusia po vypracovaní praktických úloh bude zameraná na zhodnotenie výhod centralizovanej autentizácie užívateľov, resp. zariadení (pomocou servera RADIUS) v porovnaní so spôsobmi lokálnej autentizácie. Ďalšie doplňujúce kontrolné otázky môžu spočívať vo vysvetlení možných bezpečnostných rizík súvisiacich s použitím EAP a RADIUS v otvorených sieťach alebo napr. vo vysvetlení hlavných pilierov bezpečnosti EAP-TLS a možných nevýhod tejto autentizačnej metódy.

## 2.9 Laboratórna úloha č. 9

### Zabezpečenie aplikačnej vrstvy

Cieľom laboratórnej úlohy bude oboznámiť študentov so zraniteľnosťami rôznych aplikačných protokolov a následne i s možnosťami ich vhodného zabezpečenia pred rozličnými typmi útokov. Úloha poskytne študentom praktický náhľad na testovanie zraniteľností a zabezpečenie webových aplikácií, pričom kombinuje simuláciu útokov s implementáciou ochranných opatrení.

Vysvetlenie obsiahnuté v teoretickom úvode bude zamerané na predstavenie aplikačných protokolov (HTTP, DHCP, DNS a i.) a stručnému popisu ich fungovania. Prehľadne budú zhrnuté i poznatky súvisiace s problematikou ich bezpečnosti a možných hrozieb. V súvislosti s protokolom DNS nesmie byť opomenuté jeho rozšírenie DNSSEC, ďalej v súvislosti s protokolom HTTP možnosť zabezpečenia komunikácie pomocou transportného protokolu TLS. Ďalej bude uvedený aj prehľad aktuálnych bezpečnostných hrozieb v aplikačných protokoloch a nakoniec bude nemalá pozornosť venovaná tiež vysvetleniu základných princípov realizácie útokov na aplikačné protokoly, medzi ktoré patria napr. *SQL Injection*, *Cross-Site Scripting (XSS)* a *Cross-Site Request Forgery (CSRF)*.

Študenti v praktickej časti najskôr realizujú testovanie zraniteľností webovej aplikácie bežiackej na webovom serveri (Apache/Nginx), ktorá bude zraniteľná voči vyššie zmieneným útokom *SQL Injection* a XSS. K otestovaniu zraniteľností aplikácie bude využitý nástroj **Burp Suite**, príp. OWASP ZAP. Následne študenti realizujú simulácie vyššie uvedených útokov, konkrétne *SQL Injection*, XSS a CSRF. Po vykonaní jednotlivých útokov samostatne implementujú vhodné opatrenia voči jednotlivým

---

<sup>8</sup> Úloha môže byť rozšírená o vytvorenie Wi-Fi siete s prístupovým bodom, ktorý by zároveň vystupoval v pozícii servera RADIUS.

útokom. Za účelom overenia správnosti implementácie ochranných opatrení môže byť opäť využitý nástroj Burp Suite.

Diskusia môže byť zameraná na objasnenie dôvodov vzniku uvedených zraniteľností aplikačnej vrstvy a tiež na otázky ochrany súkromia a riziko zneužitia osobných informácií, a to najmä v súvislosti s útokom XSS. Študenti ďalej vysvetlia, aký je význam validácie a ošetrovania vstupov pri zabezpečení webových aplikácií a taktiež navrhnu ďalšie kroky pre komplexnú ochranu webových aplikácií.

## 2.10 Laboratórna úloha č. 10

### Zabezpečenie Wi-Fi sietí

Laboratórna úloha umožní študentom realizovať simuláciu útokov na bezdrôtové siete s dôrazom na následnú implementáciu moderných bezpečnostných opatrení. Študenti sa v rámci tejto úlohy zoznámia s možnosťami zabezpečenia bezdrôtových sietí pred útokmi na protokoly WPA a WPA2, ďalej preskúmajú možnosti kryptografickej ochrany, a to najmä šifrovanie, používané v týchto sieťach a v neposlednom rade budú mať možnosť analyzovať dátový tok prenášaný v rámci bezdrôtovej siete.

Teoretická časť k úlohe bude predstavovať úvod do problematiky bezpečnostných protokolov používaných v bezdrôtových sieťach. Uvedený bude prehľad historických a súčasných protokolov používaných za účelom šifrovania bezdrôtovej komunikácie (WEP, WPA, WPA2, WPA3) a ich vzájomné porovnanie. Dôraz bude kladený najmä na vysvetlenie bezpečnostných slabín protokolov WEP a WPA. Pozornosť bude tiež venovaná možným hrozbám a útokom na Wi-Fi, za zmienku nepochybne stojí útok na prelomenie šifrovacích kľúčov známy ako WPA/WPA2 *cracking* či iné.

Pred zahájením samotnej praktickej časti bude najskôr potrebné na fyzickom alebo virtuálnom prístupovom bode vytvoriť testovaciu Wi-Fi sieť s protokolom WPA2-PSK, ktorú budú študenti používať na simulované útoky. Pomocou nástrojov *airodump-ng* a *aircrack-ng* študenti realizujú útok WPA/WPA2 *cracking*, kedy po zámernej deautentizácii pripojeného klienta a jeho následnom pokuse o opakované pripojenie vo vybranej bezdrôtovej sieti so šifrovaním WPA2 odchytiť *handshake* medzi prístupovým bodom a daným klientom. Táto komunikácia by mohla byť zneužitá k prelomeniu hesla danej bezdrôtovej siete realizáciou slovníkového útoku pri zvolení vhodného slovníka. Študenti následne na VM „prístupového bodu“ bezdrôtovej siete aktivujú protokol 802.11w a pomocou vhodných úprav v konfigurácii AP sa pokúsia o mitigáciu deautentizačných útokov.

Ďalej s využitím nástroja *airbase-ng* vytvorí na VM „útočníka“ tzv. *Rogue Access Point*, ktorý predstavuje falošný prístupový bod s cieľom zachytiť komunikáciu a údaje klientov. Posledná časť bude zameraná na analýzu WPS zraniteľnosti, v rámci ktorej si študenti prakticky overia, akým spôsobom môže útočník využiť zraniteľnosti WPS na

získanie prístupu do siete a ako je možné útok eliminovať vypnutím tejto funkcie na prístupovom bode.

Samostatné úlohy budú spočívať v zachytení a následnej analýze zaznamenaných dátových paketov šifrovaného prenosu v prostredí sieťového analyzátora Wireshark. Študenti taktiež implementujú rôzne metódy zabezpečenia bezdrôtovej siete, kedy vyskúšajú použitie komplexného, dostatočne silného hesla odolného voči slovníkovému útoku, implementáciu mechanizmov pre filtrovanie adries (tzv. *MAC address filtering*) na AP, vhodnú konfiguráciu WPS apod. Po implementácii uvedených bezpečnostných opatrení opäť otestujú sieť použitím nástrojov `wifite` a `aircrack-ng` k overeniu správnosti ich implementácie za účelom odstránenia identifikovaných zraniteľností siete. Kontrolou vhodnosti a dostatočnej sily nového zvoleného hesla WPA2 overia, či je sieť odolnejšia voči nežiadúcej deautentizácii klienta a *crackingu* prístupových údajov.

V rámci záverečnej diskusie študenti vysvetlia použitie jednotlivých bezpečnostných mechanizmov pre odstránenie zraniteľností bezdrôtovej siete. Taktiež vhodne interpretujú skutočnosti týkajúce sa šifrovaného prenosu zistené z vykonanej analýzy zachyteného dátového toku. Kontrolné otázky môžu spočívať napríklad vo vysvetlení nedostatkov, resp. slabín šifrovania WPA2 a zamyslení o možnostiach využitia a prínose z hľadiska bezpečnosti pri použití WPA3.

## 2.11 Laboratórna úloha č. 11

### Anonymizačné siete

V tejto laboratórnej úlohe sa študenti zoznámia s anonymizačnými sieťami, pričom hlavným zameraním bude práca s **anonymizačnou sieťou Tor**. Vyskúšajú si inštaláciu a konfiguráciu Tor, možnosti anonymného prehliadania, analyzujú tok dát vo vytvorenej anonymizačnej sieti, porovnávajú výhody a nevýhody anonymizačných sietí a diskutujú o rizikách spojených s ich využívaním.

Teoretický základ poskytne študentom úvod do problematiky anonymizačných sietí poskytujúcich používateľom možnosť anonymného prístupu do verejných sietí (resp. internetu) a utajenie ich identity. Predstavené budú rôzne anonymizačné siete ako Tor, I2P a Freenet. Hlavná pozornosť bude venovaná anonymizačnej sieti Tor, kedy budú vysvetlené základné princípy vrstveného šifrovania (tzv. *onion routing*) a tiež nebudú opomenuté známe slabiny a možné nevýhody používania Tor.

V praktickej časti sa študenti po inštalácii a konfigurácii Tor na virtuálnom stroji s OS Kali Linux zoznámia s prehliadačom **Tor Browser**, ktorý predstavuje upravenú verziu prehliadača Firefox prispôbenu pre použitie cez Tor. Študenti si prostredníctvom tohto prehliadača môžu vyskúšať anonymné prehliadanie rôznych webových stránok<sup>9</sup>.

---

<sup>9</sup> Pri dodržaní zásad etického rozsahu použitia majú študenti možnosť otestovania pripojenia i k nelegálnym webovým stránkam dostupným cez Darknet. Dôležité je ale upozorniť ich na možné riziká a zdôrazniť vhodnosť používania **primárne legálnych stránok** pre účely testovania.

S použitím sieťového analyzátora Wireshark študenti odsledujú šifrované prenosy smerované na rôzne uzly vo vytvorenej sieti Tor.

Diskusia po absolvovaní praktickej časti úlohy bude zameraná na interpretáciu informácií získaných analyzovaním zachyteného šifrovaného dátového toku prenášaného v sieti Tor. Doplnujúce kontrolné otázky môžu spočívať vo vysvetlení princípov „cibuľového“ (t. j. vrstveného) šifrovania, pomocou ktorého Tor zabezpečuje anonymitu komunikujúcich uzlov, ďalej v uvedení výhod a nevýhod použitia siete Tor či v uvedení iných alternatív anonymizačných sietí a možností ich prípadného využitia.

## 3. POUŽITÉ TECHNOLOGIE A SOFTWARE

### 3.1 VMware Workstation

VMware Player je popredná platforma pre virtualizáciu, ktorá umožňuje vytvárať a prevádzkovať virtuálne stroje<sup>10</sup> na fyzických počítačoch. Týmto spôsobom je možné na jednom počítači prevádzkovať viacero operačných systémov súčasne, bez nutnosti využitia ďalšieho hardvéru. Využitie vhodne zvolených VM bude bezpodmienečne pre kompletnú realizáciu laboratórnych úloh.

Nástroj VMware Player bol zvolený ako virtualizačná platforma, ktorá bude použitá pre virtualizáciu VM s operačným systémom Kali Linux. Tento nástroj umožňuje virtualizovať i niekoľko rôznych operačných systémov súčasne a spĺňa všetky požadované parametre potrebné pre úspešnú realizáciu jednotlivých laboratórnych úloh, na základe čoho bol zvolený ako najvhodnejší kandidát spomedzi iných dostupných nástrojov<sup>11</sup>. Ďalšou veľkou prednosťou tohto nástroja a zároveň i dôvodom, prečo bol zvolený ako najvhodnejší, je skutočnosť, že daný software je na hostiteľských zariadeniach (typicky osobných počítačoch) s OS Windows umiestnených a používaných v špecializovaných laboratórnych učebniach, už nainštalovaný. Medzi popredné funkcie nástroja VMware patrí najmä:

- umožňuje efektívne využívanie zdrojov hostiteľského systému (CPU, pamäť, kapacita disku),
- podporuje široké spektrum hostovaných operačných systémov,
- ponúka jednoduché nástroje pre správu a monitorovanie prevádzkovaných virtuálnych strojov.

Produkt VMware Workstation pozostáva z dvoch vzájomne oddelených riešení: Workstation Pro a Workstation Player. Oba produkty sú vhodné pre jednotlivcov i vývojárov, ktorí potrebujú jednoduchým spôsobom vytvárať, spúšťať a spravovať virtuálne stroje. Tieto produkty sú vhodné pre účely vzdelávania, testovania a simulácie bezpečnostných cvičení. Pre účely realizácie laboratórnych úloh bude využívaný produkt **VMware Workstation Player**.

Nástroj VMware Workstation Player umožňuje realizovať virtualizáciu:

- desktopových systémov, a to v podobe **VMware Workstation**
- a tiež serverových riešení **VMware Server**. [2]

Využitie virtuálneho počítača (resp. operačného systému) a jeho logické oddelenie od hostiteľského operačného systému je ideálnym riešením nielen pre účely vyučovania,

---

<sup>10</sup> V ďalších častiach práce bude pojem „virtuálny stroj“ nahradený skratkou VM stanovenou na základe anglického prekladu tohto termínu: *virtual machine*. V prípade označenia viacerých virtuálnych strojov bude použitá skratka **VMs**.

<sup>11</sup> Medzi ďalšie známe, často používané virtualizačné platformy patrí napr. Oracle VirtualBox, Microsoft Hyper-V ai.

nakoľko ponúka priestor pre praktické overenie aj takých situácií, ktoré by z hľadiska bezpečnosti mohli predstavovať pre hostiteľský počítač určité bezpečnostné riziko, akými môžu byť napr. zložitejšie a rozsiahlejšie úpravy v konfigurácii operačného systému zariadení alebo tiež penetračné testovanie neznámych aplikácií. Pre všetky tieto uvedené, ale i ďalšie obdobné činnosti je vhodné využiť práve možnosti virtualizácie.

## 3.2 Kali Linux

Kali Linux je špecializovaná distribúcia OS Linux, založená na distribúcii Debian, navrhnutá jeho tvorcami najmä pre etický hacking a penetračné testovanie. Dôvodom výberu Kali Linux pre počítačové cvičenia predmetu MPC-NSB bola najmä široká škála nástrojov pre vykonávanie bezpečnostnej analýzy a ďalších súvisiacich činností, ktoré sú v ňom už integrované. Táto distribúcia OS Linux poskytuje všetky nástroje potrebné pre vypracovanie jednotlivých laboratórnych úloh, ako napr. Wireshark, Burp Suite, Aircrack-ng a mnoho ďalších<sup>12</sup>, vďaka čomu je veľmi vhodnou alternatívou pre potreby počítačových cvičení. Integrácia Kali Linux do platformy VMware umožňuje simuláciu a testovanie rôznych scenárov na viacerých VMs, ktoré simulujú sieťové prostredie bez potreby vytvárania skutočnej fyzickej infraštruktúry. [3]

### 3.2.1 Nástroje Kali Linux

Práca súvisiaca s návrhom laboratórnych úloh, tvorbou praktických návodov a ich realizáciou bola podmienená vhodným výberom nástrojov, prostredí a technických prostriedkov. V tejto kapitole budú preto predstavené kľúčové technológie, ktoré boli v rámci práce používané, a ktorých uplatnenie zohráva nezanedbateľný význam pri praktickom testovaní a uskutočňovaní laboratórnych úloh, ktorých podrobnému predstaveniu bola venovaná kap. 2. Tieto nástroje a ďalšie softvérové vybavenie študentom uľahčia plnenie jednotlivých krokov pri realizácii vybraných úloh. Spravidla sa jedná o rôzne nástroje, balíčky alebo ďalší software, ktoré sú priamo integrované v systéme Kali Linux, a tak sú už priamo pripravené k okamžitému použitiu bez potreby inštalácie či ďalšej dodatočnej konfigurácie. Pre prípad nutnosti ich inštalácie (resp. aktualizácie) je do vytvoreného návodu k laboratórnej úlohe zakomponovaný aj postup pre inštaláciu príslušného nástroja, ktorá je realizovateľná typicky v prostredí terminálu s použitím niekoľkých jednoduchých príkazov. Ukážku niektorých spomedzi dostupných nástrojov je možné vidieť na obr. 3., kde je zobrazené priamo vyhľadávacie menu v prostredí Kali Linux.

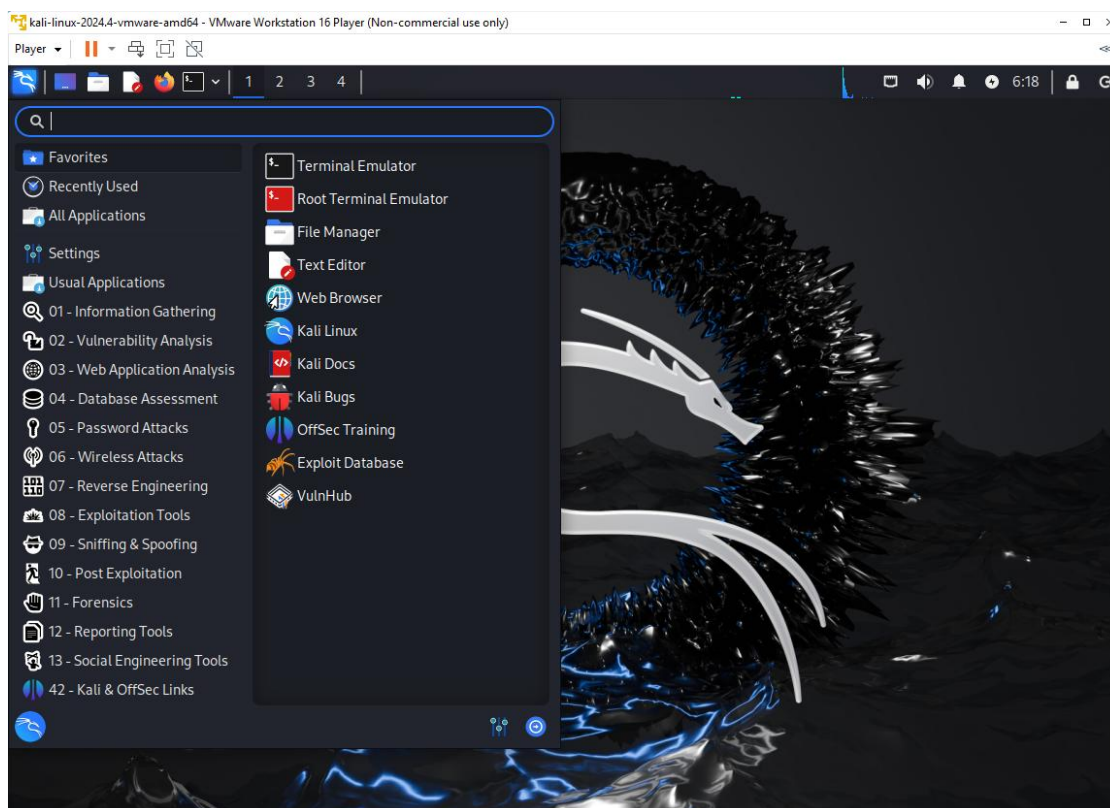
Kali Linux vo svojej základnej verzii obsahuje viac než 600 nástrojov (tzv. *tools*) pre penetračné testovanie a bezpečnostný audit. Nižšie bude uvedený ich stručný prehľad<sup>13</sup>. Podrobnejšie oboznámenie s danými nástrojmi bude pre študentov pripravené

---

<sup>12</sup> Súhrnný prehľad používaných nástrojov bude ďalej nasledovať.

<sup>13</sup> Je dôležité podotknúť, že sa nejedná o kompletný zoznam všetkých existujúcich nástrojov dostupných v prostredí systému Kali Linux. Vybraných bolo len niekoľko reprezentantov, a to na základe ich dôležitosti

neskôr ako súčasť materiálov a samotných praktických návodov vytvorených pre účely vyučovania u konkrétnych laboratórnych úloh. [5], [6]



Obrázok 3.1 Ukážka zoznamu niekoľkých integrovaných nástrojov vo vytvorenom VM s Kali Linux.

## Wireshark

Nástroj Wireshark patrí medzi najznámejšie a najpoužívanejšie sieťové analyzátory. Umožňuje odchytiť komunikáciu prebiehajúcu na rozhraniach daného zariadenia a následne analýzu takto zaznamenaných dátových jednotiek. V rámci analýzy umožňuje nielen skúmať podrobné informácie o zachytených dátových jednotkách, ale poskytuje tiež rôzne možnosti pokročilejšieho spracovania zaznamenaných dát v podobe vytvorenia schematických znázornení priebehu komunikácie, analýzy strátovosti a úspešnosti prenosu dátových jednotiek, apod. Wireshark dokáže spracovať rôzne typy zachytených formátov súborov, vrátane Microsoft Network Monitor, Pcap NG, Siffer Pro a ďalších. Wireshark disponuje pestrú škálou rozličných funkcionalít, no medzi kľúčové funkcie tohto nástroja patria najmä:

---

pre účely praktickej realizácie príslušných laboratórnych úloh, ktorých náplň a podrobnejšie predstavenie bude nasledovať v ďalších častiach tejto práce.



- **Zachytávanie paketov ( dátovej komunikácie)**
  - Wireshark umožňuje zaznamenávať dátové jednotky v rámci sledovanej prebiehajúcej prevádzky v reálnom čase.
  - Získané dáta sú ukladané do súborov (vo formáte .pcap), ktoré je možné následne analyzovať offline.
- **Podpora mnohých protokolov**
  - Wireshark podporuje veľké množstvo protokolov, vrátane TCP/IP, HTTP, DNS, ARP, SSL/TLS a ďalších.
- **Filtrácia zachytených paketov**
  - Podrobný triediaci systém umožňuje filtrovať pakety na základe rôznych stanovených kritérií, a to napr. podľa protokolov, IP adries, portov, typu prenášaných dát a ďalších iných parametrov.
- **Analýza šifrovanej komunikácie**
  - Samotný nástroj Wireshark síce neumožňuje realizovať priamo dešifrovanie kryptograficky zabezpečených dát, no s jeho využitím je možno zachytené šifrované dátové jednotky analyzovať z hľadiska v nich obsiahnutých metadát, ktoré môžu zahŕňať napr. zdrojovú a/alebo cieľovú IP adresu komunikujúcich koncových zariadení, veľkosť obsahu prenášaných dát, atď.
- **Grafická vizualizácia**
  - Pre zachytenú dátovú komunikáciu je možné vytvoriť a graficky znázorniť napr. prenosovú šírku pásma zachytenej komunikácie, priebeh toku dát, latenciu alebo graf stratovosti či vyťaženia siete apod., čo môže byť veľmi vhodné a užitočné pre riešenie prípadných otázok súvisiacich s diagnostikou siete a identifikáciou vzniknutých problémov. [4]

## Metasploit Framework

Metasploit Framework patrí medzi významné nástroje integrované v systéme Kali Linux. Táto modulárna platforma vhodná pre penetračné testovanie umožňuje podľa potreby rôznym spôsobom manipulovať s testovacím kódom (tzv. *exploitom*) – vytvárať ho, editovať, testovať, spúšťať. Používatelia môžu v rámci frameworku využiť množstvo nástrojov podporujúcich rozličné funkcie, ako je prehliadanie siete, maskovanie prenosov a/alebo obsahu za účelom vyhnutia sa detekcii pri distribúcii kódu, testovanie bezpečnostných zraniteľností, vykonávanie útokov a mnohé ďalšie.

## **Burp Suite**

Jedná sa o sadu nástrojov pre testovanie bezpečnosti webových aplikácií, Jednotlivé súčasti (nástroje) umožňujú zautomatizovať opakujúce sa kroky reprezentujúce jednotlivé úlohy v rámci testovania. Burp Suite ponúka možnosti pre praktické otestovanie 10 najzávažnejších zraniteľností webových aplikácií (tzv. OWASP TOP 10)<sup>14</sup> a zároveň poskytuje najnovšie hackerské techniky. Výhodou tohto nástroja je dostupnosť jednoduchšej dokumentácie a možnosť prehľadnej tvorby správ, komunikácie a zdieľania medzi užívateľmi.

## **Aircrack-ng**

Aircrack-ng predstavuje balíček nástrojov vhodných pre testovanie zabezpečenia bezdrôtových WiFi sietí. Umožňuje testovanie odolnosti, resp. sily použitého hesla, sledovanie komunikácie, no najčastejšie sa používa za účelom prelomenia použitého zabezpečenia, a to konkrétne hesiel príslušných bezdrôtových sietí zabezpečených pomocou WEP alebo WPA-PSK s využitím napríklad slovníkového útoku. Všetky nástroje v balíku je možné aplikovať prostredníctvom príkazového riadku (v termináli Kali Linux).

## **Nmap (Network Mapper)**

Network Manager, skrátene označovaný ako Nmap je bezplatný *open-source* nástroj s otvoreným zdrojovým kódom, ktorý možno využívať pre vykonávanie úloh zameraných na testovanie sieťovej bezpečnosti, akými môžu byť napríklad správa plánov aktualizácie služieb, monitorovanie dostupnosti služieb, správa a dohľad nad jednotlivými zariadeniami a sieťovými prvkami v sieti apod. Funkcionalita tohto nástroja je založená na sledovaní sieťového toku IP paketov. V rámci výstupov skenovania siete sú získané výsledky zahŕňajúce prehľad dostupných hostiteľských zariadení, serverov a iných súčastí v analyzovanej sieti, dostupné služby, ich operačné systémy, aktuálne používané verzie atď.

---

<sup>14</sup> OWASP TOP 10 predstavuje zoznam najvýznamnejších bezpečnostných rizík a zraniteľností, ktoré ohrozujú webové aplikácie. V pravidelných intervaloch ho zostavuje OWASP, nezisková organizácia zameraná na zvyšovanie povedomia o kybernetickej bezpečnosti, v rámci svojich prebiehajúcich projektov v oblasti bezpečnosti webových aplikácií. Tento zoznam je pravidelne aktualizovaný na základe údajov z reálneho sveta, vrátane analýzy bezpečnostných incidentov, trendov a príspevkov komunity. Aktuálna verzia tohto zoznamu bola vyhotovená v r. 2021 – pre viac informácií viď [7]. Momentálne sa pracuje na aktualizácii zoznamu, pričom oficiálne zverejnenie OWASP Top 10:2025 je plánované v prvej polovici kalendárneho roka 2025.

## Nástroje využité pre vybrané laboratórne úlohy

V rámci tejto diplomovej práce budú spracované podrobné návody<sup>15</sup> popisujúce kroky pre praktické uskutočnenie štyroch vybraných laboratórnych úloh. Pripravené virtuálne stroje, ktoré budú poskytnuté ako praktický výstup k tejto práci, majú uskutočnené nastavenie a potrebnú konfiguráciu, aby bolo s ich využitím uskutočniť vybrané úlohy<sup>16</sup>. Nižšie bude uvedený stručný prehľad nástrojov, ktorých aplikované nastavenia, funkčnosť a praktické uplatnenie v prostredí poskytnutých VMs boli overené počas praktického testovania vytvorených návodov:

- **Wireshark**: pokročilý sieťový analyzátor, umožňujúci monitorovanie, zaznamenávanie, zobrazenie a filtrovanie komunikácie, resp. prenášaných dátových jednotiek. Predstavuje kľúčový nástroj pri analýze protokolov ako ARP, IP, TCP, Tor a iné.
- **arpspoof**: nástroj vhodný pre realizáciu ARP *spoofing* útokov, ktorý umožňuje útočníkovi generovať podvrhnuté odpovede na žiadosti ARP protokolu a docieľiť tak prepísanie informácií v ARP tabuľke cieľového zariadenia.
- **Etercap**: komplexný nástroj pre *Man-in-the-Middle* útoky s dostupným grafickým užívateľským rozhraním, ktorý podporuje útoky ako ARP *poisoning*, odposluch komunikácie (*sniffing*), DNS *spoofing*, ai.
- **hping3**: generátor paketov pre testovanie siete, umožňujúci tiež simuláciu IP *spoofingu*, skenovania portov, merania latencie alebo DOS útokov.
- **Tcpdump**: konzolový nástroj na zachytávanie paketov, ktorý je možné využiť najmä za účelom rýchleho overovania komunikácie priamo v termináli.
- **ip**: nástroj na konfiguráciu sieťových rozhraní zariadení, smerovacích tabuliek a ďalších základných sieťových operácií.
- **hostapd** + **FreeRADIUS**: uvedená kombinácia nástrojov využívaná na simuláciu bezdrôtového prístupového bodu a autentizačného servera pri úlohách s EAP autentizáciou.
- **Tor**: klient anonymizačnej siete používaný k anonymnému prehliadaniu v kombinácii so špeciálne prispôbeným prehliadačom **Tor Browser** pre analýzu rozdielov medzi bežným a anonymným pripojením cez Tor sieť.

Tieto nástroje spolu s potrebnými konfiguráciami a prípadnými ďalšími analytickými pomôckami boli kľúčovým prvkom pre tvorbu praktickej časti každej z vybraných laboratórnych úloh. Podrobný popis práce s jednotlivými nástrojmi je uvedený v príslušnom návode ku konkrétnej úlohe.

---

<sup>15</sup> Popisu návodov vytváraných za účelom ich neskoršieho využitia pre potreby výuky počítačových cvičení predmetu MPC-NSB je venovaná kap. 5.

<sup>16</sup> Konkrétne sa jedná o vybrané úlohy č. 3: Bezpečnosť spojenej vrstvy, č. 4: Bezpečnosť sieťovej vrstvy, č. 8: Autentizácia pomocou EAP a RADIUS a č. 11: Anonymizačné siete.

## Ďalšie použité nástroje

Pri riešení jednotlivých laboratórnych úloh z celého zoznamu celkom jedenástich úloh budú študenti v priebehu celého semestra využívať pre dosiahnutie stanovených cieľov počítačových cvičení okrem vyššie uvedených nástrojov i niekoľko ďalších, medzi ktorými možno ešte spomenúť napr.: John the Ripper, netcat, Nikto, OWASP ZAP, sqlmap, sslstrip atď.

Podrobnejšie informácie, popis a charakteristika jednotlivých použitých nástrojov budú vždy súčasťou vytvorených návodných postupov a ďalšej dokumentácie k príslušným laboratórnym úlohám. Viac podrobností o vyššie uvedených nástrojoch je možné dočítať sa napr. v literatúre [5], [6].

### 3.2.2 Konfigurácia pracovného prostredia a virtuálnych strojov

Na realizáciu jednotlivých laboratórnych úloh bolo využité virtualizačné prostredie s viacerými samostatnými inštanciami operačného systému Kali Linux. Tento typ pracovného prostredia bol zvolený z dôvodu, že študentom umožní bezpečne testovať reálne útoky, uskutočniť ich simulácie a taktiež implementovať ochranné mechanizmy proti nežiadúcim hrozbám a útokom, a to všetko v izolovanom prostredí, bez rizika ohrozenia vonkajšej siete alebo iných systémov.

Praktické cvičenia sú navrhnuté tak, aby simulovali reálne sieťové prostredia, v ktorých vystupujú minimálne dve až tri rôzne zariadenia. Ich funkcionality, resp. role sa líšia v závislosti na problematike, ktorej praktickému rozboru sa príslušná laboratórna úloha venuje, no typicky sa jedná o koncového klienta, server (napr. autentizačný alebo koncový) a posledný VM slúži k zastúpeniu role útočníka alebo určitého sprostredkovateľa komunikácie medzi klientom a serverom (napr. prístupový bod). Preto každá úloha predpokladá súbežné spustenie, spravidla, troch nezávislých virtuálnych strojov v rámci jedného hostiteľského počítača, čo umožňuje plnohodnotné testovanie bezpečnostných mechanizmov a sieťovej komunikácie.

Virtuálne stroje boli vytvárané ako klony základného predinštalovaného systému Kali Linux<sup>17</sup>. Tento prístup umožňuje jednoduché obnovenie východiskového stavu, úsporu času pri ich nastavovaní a potrebnej dodatočnej konfigurácii a zaručuje konzistenciu konfigurácie medzi študentmi. Každý z VM mal priradenú statickú IP adresu v rámci preddefinovaného privátneho rozsahu (napr. 192.168.126.0/24), čo zjednodušuje adresovanie a zabezpečuje jednoznačnú identifikáciu jednotlivých účastníkov komunikácie.

Komunikácia medzi virtuálnymi strojmi prebiehala cez internú (host-only) alebo NAT sieť, v závislosti od potreby pripojenia do internetu pri riešení konkrétnej praktickej úlohy. V niektorých úlohách bolo nevyhnutné zvoliť režim NAT, aby študenti mohli testovať externé služby (napr. výstup na web cez Tor, zisťovanie IP adresy na externom

---

<sup>17</sup> Podrobný popis inštalácie je pre prípadných záujemcov popísaný v kap. 4.

serveri či komunikáciu so vzdialeným WHOIS serverom<sup>18</sup>). Sieťové adaptéry jednotlivých VM boli nastavené v súlade s požiadavkami príslušnej úlohy, pričom bola zabezpečená vzájomná konektivita a možnosť monitorovať prevádzku prostredníctvom nástrojov vhodných pre záznam a analýzu sieťovej komunikácie ako Wireshark.

Pre bezproblémové fungovanie odporúčame overiť aktuálnosť systémových balíčkov, dostupnosť siete z každého VM a zabezpečiť, že každý stroj má nainštalované základné nástroje<sup>19</sup> popísané v príslušnej časti návodu k úlohe, kde je vždy uvedený prehľad použitých nástrojov.

---

<sup>18</sup> Vymenované príklady sa týkajú konkrétne úlohy č. 11 venovanej problematike anonymizačných sietí.

<sup>19</sup> Pre potreby inštalácie či prípadnej aktualizácie používaných nástrojov sú vždy v návodoch uvedené konkrétne príkazy pre prácu v terminálovom okne Kali Linux.

## 4. PRÍPRAVA VIRTUÁLNYCH STROJOV PRE LABORATÓRNE ÚLOHY

V rámci tejto kapitoly bude rozpísaný postup pre vytvorenie a inštaláciu potrebných virtuálnych strojov, pomocou ktorých budú realizované všetky uvedené laboratórne úlohy v rámci laboratórnych cvičení. Pre inštaláciu a konfiguráciu, prístup k vytvoreným virtuálnym strojom a následnú prácu s nimi bude využitá **platforma VMware** umožňujúca virtualizáciu jedného či viacerých virtuálnych počítačov, resp. operačných systémov súčasne na jednom hositeľskom zariadení.

### 4.1 Stiahnutie a inštalácia Kali Linux

V priestoroch špecializovanej laboratórnej, resp. počítačovej učebne používanej pre potreby výučby predmetu MPC-NSB, kde budú jednotlivé laboratórne úlohy riešené, bude potrebný software už vopred pripravený. Avšak pre záujemcov, ktorí by pre vlastné potreby zvažovali nutnosť využitia virtuálnych strojov s Kali Linux, bude ďalej uvedený postup pre stiahnutie a inštaláciu Kali Linux vo VMware.

#### 4.1.1 Stiahnutie ISO obrazu Kali Linux

Pre účely využitia Kali Linux je v súčasnej dobe k dispozícii už vytvorený kompletný ISO obraz Kali Linux. Ten je možné získať priamo z oficiálnych stránok 56[1]:

- Prejdite na stránku **Download Kali Linux** <sup>20</sup>.
- Vyberte možnosť **Installer Images** a zvolte konkrétny **ISO** obraz vhodný pre vaše potreby (32-bit alebo 64-bit podľa vami používaného hosťujúceho operačného systému).
- Kliknutím na symbol reprezentujúci možnosť download stiahnete vybraný ISO obraz na svoj lokálny disk.

#### 4.1.2 Inštalácia Kali Linux vo VMware

Pre vytvorenie nového virtuálneho stroja v prostredí nástroja VMware je potrebné postupovať nasledovne:

1. **Otvorte VMware Workstation Player alebo Pro**
  - V prípade, ak ešte nemáte nainštalovaný VMware na svojom hosťujúcom zariadení, stiahnite a nainštalujte ho z [oficiálnej stránky](https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion) <sup>21</sup>.
2. **Vytvorenie nového virtuálneho stroja**
  - Kliknite na **Create a New Virtual Machine**.

---

<sup>20</sup> Konkrétne: <https://www.kali.org/get-kali/#kali-platforms>.

<sup>21</sup> Konkrétne: <https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion>.

- Vyberte možnosť **Installer disc image file (ISO)** a vyberte ISO obraz Kali Linux stiahnutý podľa postupu uvedeného v kap. 4.1.1.
- V ďalšom kroku **Guest Operating System** zvolte operačný systém "**Linux**" a jeho verziu, ktorou je **najnovšia** možná verzia **distribúcie Debian**.

### 3. Konfigurácia nastavení virtuálneho stroja

- Zvoľte vhodný názov (meno) virtuálneho stroja, napr. **Kali Linux**.
- Vyberte vhodné miesto na disku hostujúceho OS, kde má byť vytvorený VM uložený.
- Nastavte veľkosť virtuálneho disku, napr. **20 GB** (minimálne odporúčené pre Kali Linux).
- Priradte vytváranému VM dostatočné množstvo pamäti (RAM) hostujúceho zariadenia, napr. **2 GB** (odporúča sa 4 GB alebo viac).

### 4. Pridanie sieťových rozhraní

- Ako úvodné nastavenie zvolte konfiguráciu sieťového rozhrania v režime **NAT**, aby mal vytvorený VM prístup k internetu, prípadne **Bridged**, ak požadujete, aby bol VM súčasťou lokálnej siete.

### 5. Spustenie inštalácie

- Kliknite na **Finish** a spustíte vytvorený virtuálny stroj.
- Virtuálny stroj sa spustí z ISO obrazu Kali Linux uloženého na disku hostujúceho zariadenia.

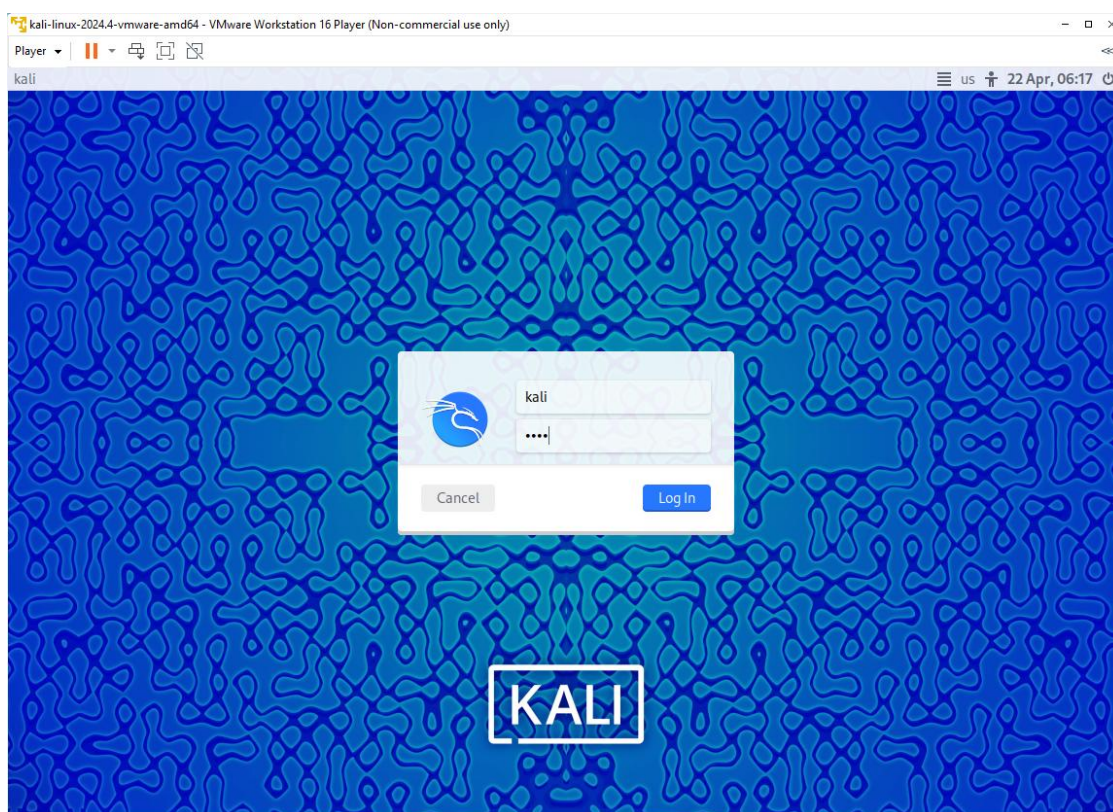
### 6. Inštalácia Kali Linux

- Po spustení nového VM postupujte ďalej podľa pokynov zobrazených na obrazovke pre inštaláciu Kali Linux (výber jazyka, voľba časového pásma, používateľského mena a hesla, rozdelenie disku, ...).
- Po úspešnej inštalácii sa VM reštartuje a na úvodnej obrazovke (obr. 4.) sa zobrazí možnosť pre zadanie údajov pre prihlásenie užívateľa.
- Vytvorený Kali Linux tak bude pripravený pre ďalšie použitie.

#### 4.1.3 Spustenie a používanie Kali Linux vo VMware

Po úspešnej inštalácii a následnom spustení vášho Kali Linux môžete spustiť a používať predinštalované bezpečnostné nástroje a utility, ako sú Wireshark, Metasploit, Burp Suite, Aircrack-ng atď.

VMware umožňuje vytváranie snímok (tzv. *snapshots*) aktuálneho stavu, resp. obrazu vášho virtuálneho stroja. Jedná sa o užitočnú funkcionality, ktorú je možné využívať pre uloženie aktuálneho stavu VM, ku ktorému sa môžete neskôr v budúcnosti vrátiť a pokračovať v jeho ďalšom používaní



Obrázok 4.1 Úvodná obrazovka pre prihlásenie užívateľa po spustení vytvoreného VM s Kali Linux.



## 5. DOKUMENTÁCIA PRE POTREBY VÝUKY

Súčasťou tejto kapitoly je popis vytvárania štruktúrovaných návodov pre účely výuky predmetu MPC-NSB. V kap. 1.4 bol predstavený kompletný zoznam obsahujúci spolu celkom jedenásť praktických laboratórnych úloh, neskôr boli v rámci kap. 2 jednotlivé úlohy stručne charakterizované. Z navrhnutého zoznamu boli vybrané štyri laboratórne úlohy, pre ktoré boli ako súčasť praktického výstupu k tejto práci vytvorené podrobné návody obsahujúce podrobné informácie nutné k plnohodnotnej a úspešnej realizácii príslušnej úlohy. Táto kapitola bližšie popisuje vytvorené návody, a to jednak z hľadiska ich štruktúry a obsahu, a tiež z pohľadu ich publikácie pre následné praktické použitie.

### 5.1 Tvorba návodov pre vybrané laboratórne úlohy

V rámci tejto diplomovej práce boli z celého zoznamu jedenástich navrhnutých laboratórnych úloh (viď kap. 1.4) vybraté štyri, pre ktoré boli vytvorené rozsiahle, podrobné a didakticky štruktúrne spracované návody<sup>22</sup>. Samotný výber týchto úloh nebol náhodný, ale opieral sa o dôkladnú úvahu o ich zameraní, praktickej prenositeľnosti, aktuálnosti a pedagogickom prínose. Vybrané úlohy pokrývajú kľúčové oblasti sieťovej bezpečnosti, akými sú napríklad ARP *spoofing*, IP *spoofing* a zabezpečenie prenosov pomocou technológie IPsec, uplatnenie autentizačnej metódy EAP-MD5 v procese autentizácie klienta v spolupráci s autentizačným serverom RADIUS, ako aj prostriedky anonymizácie s cieľom ochrany súkromia klienta v sieti prostredníctvom siete Tor. Každá z vybraných úloh tak reprezentuje odlišný aspekt sieťovej komunikácie a jej zabezpečenia. Konkrétne sa jedná o nižšie uvedené laboratórne úlohy:

- úloha č. 3: **Bezpečnosť spojenej vrstvy,**
- úloha č. 4: **Bezpečnosť sieťovej vrstvy,**
- úloha č. 8: **Autentizácia pomocou EAP a RADIUS,**
- úloha č. 11: **Anonymizačné siete.**

Cieľom podrobného spracovania týchto návodov bolo vytvoriť prehľadne štruktúru, ktorá by bola pre študentov ľahko čitateľná a pochopiteľná, no zároveň dostatočne podrobná pre zrozumiteľné vysvetlenie diskutovanej problematiky. A taktiež vo veľkej miere názorná, aby študentom uľahčila praktické spracovanie zadaných problémov. Každý návod je spracovaný ako samostatný dokument a má jednotnú a prehľadnú štruktúru, ktorá sa v jednotlivých úlohách opakuje. Táto štruktúra bola navrhnutá s dôrazom na používateľskú prístupnosť, pedagogickú hodnotu a praktickú využiteľnosť. Každý návod preto obsahuje nasledujúce časti:

---

<sup>22</sup> Podrobne **spracované návody k vybraným laboratórnym úlohám**, ktoré boli vytvárané za účelom ich budúceho vyučovania v rámci vyučovania praktických počítačových cvičení k predmetu MPC-NSB, sú súčasťou tejto diplomovej práce vo forme príloh.

- **Úvod:** predstavuje stručné oboznámenie s cieľom úlohy a naznačuje problematiku, ktorou sa príslušná laboratórna úloha zaoberá.
- **Teoretický úvod:** oboznamuje študentov so základnými pojmami, protokolmi, typmi útokov a vysvetľuje všetky podstatné súvislosti týkajúce sa riešenej problematiky v rámci danej úlohy, ktorých objasnenie a pochopenie je nevyhnutné pre nadväzujúcu praktickú časť, a následne tiež s technológiami, ktoré sa v praktickej časti úlohy využívajú. Slúži tak ako vhodný teoretický podklad pre nadobudnutie základných znalostí ohľadom riešenej problematiky .
- **Popis použitých nástrojov:** špecifikácia softvérových nástrojov a ich funkcií s uvedením základných príkladov pre ich praktickú aplikáciu, čo študentom pomáha lepšie porozumieť ich účelu a spôsobu použitia. Zahŕnutie tejto časti do návodu je potrebné pre zoznámenie študentov s konkrétnymi nástrojmi, ich možnosťami a praktickým využitím v kontexte danej konkrétnej úlohy. Súčasťou sú často aj ukážky príkazov alebo výstupov v termináli.
- **Praktický návod:** najdôležitejšia časť predstavujúca samotný návod, kde je presne popísaný postup riešenia praktickej časti úlohy, ktorá môže spočívať napr. v simulácii útoku, aplikácii a konfigurácii vhodných mechanizmov pre zabezpečenie dátovej komunikácie apod., a to vrátane konfigurácií, príkazov, príkladu očakávaných výstupov, terminálových záznamov, snímok obrazovky a vizualizácií. Súčasťou tejto časti je aj schematické znázornenie topológie vytvorenej virtuálnej siete medzi používanými virtuálnymi strojmi, nakoľko sa pre riešenie úloh využíva viacero (typicky dva až tri) VMs Kali Linux spustených na jednom hostiteľskom počítači v laboratórnej učebni.
- **Zadanie samostatnej úlohy:** nadväzuje na predchádzajúcu praktickú prácu študentov a jej cieľom je študenta podnietiť k samostatnému mysleniu a aplikácii nadobudnutých znalostí mimo predpísaného návodu. Samostatná úloha tak poskytuje študentom priestor na overenie získaných vedomostí a samostatné uplatnenie získaných zručností v podobnej situácii alebo pri riešení modifikovaného zadania.
- **Kontrolné otázky:** slúžia na overenie porozumenia kľúčovým konceptom z teoretickej a praktickej časti úlohy. Ich cieľom je pomôcť študentom upevniť si nadobudnuté poznatky, pripraviť sa na prípadné ústne obhajoby riešených úloh a zároveň poskytnúť vyučujúcemu nástroj na jednoduché overenie porozumenia prebranej problematiky. Otázky sú zadané vo forme testových otázok s výberom z ponúkaných možností (typ ABCD), pričom u niektorých otázok môže byť správna viac než len jedna odpoveď.
- **Záver:** stručné zhrnutie úlohy, ktoré uvádza súhrnný prehľad toho, čo bolo cieľom úlohy, akým spôsobom sa daný cieľ naplňal a aké boli očakávané výstupy praktického riešenia. Zároveň poskytuje tiež prehľad kľúčových

teoretických a praktických poznatky, ktoré mali študenti absolvovaním laboratórnej úlohy nadobudnúť a ktoré budú môcť využiť pri riešení ďalších, náročnejších úloh v oblasti sieťovej bezpečnosti.

Súčasťou návodov sú aj snímky obrazovky, ukážky výstupov príkazov, konfiguračné súbory, sieťové diagramy alebo iné vizuálne ukážky, ktoré študentom pomáhajú lepšie pochopiť situáciu a ľahšie sa orientovať v postupe, alebo im môžu v určitých bodoch postupu overiť si správnosť svojho doterajšieho riešenia.

Z didaktického hľadiska je dôležité spomenúť aj to, že stratégia tvorby jednotlivých návodov reflektovala aj potrebu postupného zvyšovania samostatnosti študentov. Preto bol kladený dôraz na to, aby prvé návody boli detailné, čo najviac podrobné a aby študentom poskytli úplnú oporu v rámci praktickej časti úlohy v podobe postupu uvedeného „krok za krokom“. Následne, s postupne rastúcou zložitou a počtom absolvovaných úloh, sa niektoré rutinné úpravy (napr. spúšťanie a použitie nástroja Wireshark pre monitorovanie a následnú analýzu prebiehajúcej komunikácie) opakovali už len v podobe skráteného, jednoduchého pokynu alebo formou odkazu na predchádzajúce úlohy, aby sa predišlo zbytočnej duplicite a zvýšila sa efektivita práce. Predpokladá sa totiž, že študenti v priebehu semestra postupne nadobudnú základné praktické zručnosti, ktoré budú vedieť aplikovať neskôr samostatne v praxi, a zároveň sa tým podporuje i ich schopnosť udržať si konzistentný pracovný postup a vhodným spôsobom prepájať nadobudnuté znalosti a zručnosti.

### 5.1.1 Dokumentácia pre vyučujúceho

Každý z vytvorených návodov pre príslušnú laboratórnu úlohu je navyše doplnený o samostatnú **dokumentáciu pre vyučujúceho**. Tento dokument obsahuje:

- základné informácie k laboratórnej úlohe popisujúce jej cieľ a účel,
- predpokladané výstupy práce študentov v praktickej časti úlohy,
- popis očakávaného riešenia samostatnej úlohy študentov,
- odpovede na zadané testové otázky,
- doplňujúce kontrolné otázky.

Cieľom pripravenej dokumentácie je uľahčiť prácu pedagógovi pri kontrole práce študentov, zjednotiť hodnotenie a poskytnúť oporu pri objasňovaní riešenia a v prípade výskytu možných problémov, ktoré by sa mohli objaviť pri práci študentov počas ich praktickej činnosti. Dokumentácia obsahuje tiež doplňujúce otázky, ktoré vyžadujú formuláciu vlastnej plnohodnotnej odpovede na zadanú otázku a môžu byť využité pri kontrole samostatnej práce študentom. Vyučujúcemu tak táto dokumentácia môže slúžiť ako pomocný podklad pre jednoduché overenie, či študenti pochopili podstatu úlohy a zvládli jej riešenie.

## 5.2 Forma a zverejnenie vytvorených návodov

Vyššie popísané návody k vybraným štyrom úlohám boli písané v slovenskom jazyku, a v tejto podobe sú zakomponované ako súčasť tejto diplomovej práce ako záverečné prílohy. Rovnako sú však súčasťou odovzdanej elektronickej prílohy k tejto práci ako samostatné súbory vo formáte `.docx`, a to jednak jednotlivé návody, ale tak isto aj k nim prislúchajúce texty pre vyučujúcich (dokumentácie).

Aby mohli byť vytvorené návody využité aj v rámci výuky a mohli slúžiť študentom *Vysokého učení technického v Brně* ako učebný materiál, boli preložené aj do českého jazyka. V tejto podobe boli odovzdané len ako súčasť elektronickej prílohy, rovnako tak ako aj české verzie dokumentácie pre vyučujúceho ku každému vytvorenému návodu.

### 5.2.1 HTML verzia návodov

Za účelom využitia návodov k vybraným laboratórnym úlohám aj priamo v procese výuky počítačových cvičení k predmetu MPC-NSB boli na základe ich textovej podoby, ktorá je súčasťou tejto diplomovej práce vo forme príloh, pre každý zo štyroch vytvorených návodov vytvorené aj interaktívne HTML stránky, ktoré slúžia ako multimediálna a responzívna alternatíva ku klasickým textovým návodom vo formáte `.docx`. Tieto webové stránky boli vytvárané s cieľom zabezpečiť efektívnu použiteľnosť vytvorených návodov pre potreby výuky, umožniť jednoduché zobrazenie na rôznych typoch zariadení, a zároveň poskytnúť nástroje na samostatné overovanie získaných vedomostí prostredníctvom testových otázok. HTML podoba návodov zabezpečuje:

- dostupnosť z akéhokoľvek zariadenia s prehliadačom, čo zabezpečí, že ich môžu mať študenti kedykoľvek k dispozícii,
- možnosť interaktívneho zobrazenia, preklikávania medzi jednotlivými kapitolami (teoretický úvod, praktický návod apod.),
- responzivitu na mobilných zariadeniach,
- možnosť jednoduchej aktualizácie pripraveného obsahu (t. j. jednotlivých súčastí návodu k laboratórnej úlohe).

### Použité technológie

Pre tvorbu HTML stránok pre textové návody k laboratórnym úlohám boli použité štandardné webové technológie:

- **značkovací jazyk HTML5** – vytvorenie základnej kostry webovej stránky, definícia, vhodné formátovanie a členenie štruktúry obsahu (sekcie, nadpisy, odstavce, tabuľky, obrázky, ukážky kódu, ...),
- **CSS (*Cascading Style Sheets*)** – použité za účelom dotvorenia vzhľadu a celkovej vizuálnej podoby webových stránok,
- **JavaScript** – zabezpečuje interaktívne prvky stránky, ako napr. zobrazovanie obrázkov a dynamické generovanie číslovania, plynulé prechody medzi

jednotlivými sekciami, no najmä umožňuje implementovať potrebné funkcionality pre automatické vyhodnocovanie kontrolných testov.

### Štruktúra webovej stránky

Štruktúra vytvorených webových stránok odzrkadľuje štruktúru samotného návodu a snaží sa v čo najväčšej možnej miere reflektovať všetky prvky textových verzií návodov (rozdelenie na jednotlivé sekcie, formáty písma, nadpisy rôznych úrovní, zoznamy, bloky s príkazmi, výpisy, ukážky apod.) a ich výhodou je aj možnosť prepojenia s externými odkazmi, ako aj vloženie obrazových ukážok v podobe schém alebo názorných snímok obrazovky. Každý návod je prevedený do HTML pomocou jednoduchej štruktúry, ktorá zahŕňa nasledujúce:

- **záhlavie stránky** – obsahuje názov úlohy, prípadne logo školy,
- **navigácia** – ponúka rýchly prístup k jednotlivým sekciam stránky, ktoré kopírujú logickú štruktúru návodu (teória, nástroje, praktický postup...),
- **telo stránky** – samotný obsah návodu k úlohe (textová časť), rozdelený do samostatných, prehľadných blokov, často doplnený o obrázky, tabuľky, farebné zvýraznenie dôležitých príkazov či bloky kódu,
- **záver alebo záložky na stiahnutie** – odkazy na PDF verziu, zdroje použitej literatúry a prípadne ďalšie súbory na stiahnutie.

HTML stránky (resp. ich kód) môže byť vytváraný buď ručne, alebo pomocou jednoduchých nástrojov a editorov (napr. Visual Studio Code, Notepad++). Stránky sú responzívne, čo umožňuje automatické prispôbenie ich veľkosti a rozlíšenia obrazovke zariadenia, na ktorom sú zobrazené, a pripravené na publikovanie na webovom serveri školy alebo LMS systéme<sup>23</sup> (napr. Moodle). Takto vytvorené HTML návody sú zároveň vhodné aj na dlhodobú archiváciu a prípadné aktualizácie. V prípade potreby je možné jednoducho doplniť alebo upraviť jednotlivé sekcie bez zásahu do celej štruktúry dokumentu.

*Pozn.: ukážka podoby webovej stránky pre jednu z úloh je zobrazená na priložených obrázkoch na konci tejto kapitoly (viď obr. 5.1, 5.2 a 5.3).*

### Vyhodnocovanie kontrolných testov

Za jednu z významných predností HTML podoby vytvorených návodov je možnosť automatického vyhodnocovania odpovedí študentov na zadané kontrolné otázky týkajúce sa riešenej problematiky, ktoré sú taktiež súčasťou návodov. Cieľom kontrolných testov je overiť mieru porozumenia problematiky a celkového zvládnutia praktickej realizácie laboratórnej úlohy, pričom možnosť ich vyhodnocovania priamo na webovej stránke

---

<sup>23</sup> LMS (*Learning Management System*) predstavuje systém vhodný pre podporu vzdelávania a výučbového procesu vo virtuálnom online prostredí. Umožňuje učiteľom jednoducho vytvárať a spravovať rôzne kurzy, zadávať úlohy, testy a publikovať študijné materiály. Zároveň poskytuje priestor na komunikáciu medzi vyučujúcimi a študentmi. Medzi najpoužívanejšie LMS systémy patria napríklad Moodle, Google Classroom alebo Microsoft Teams for Education.

počas samostatnej práce študentov bez nutnosti bezprostrednej interakcie s vyučujúcim prináša veľký benefit z pohľadu časovej náročnosti, a tak môže vo významnej miere pozitívne ovplyvniť celkový priebeh praktických počítačových cvičení

**Interaktívny kontrolný test** predstavuje jednu z kľúčových častí webovej stránky. Umožňuje študentom preveriť svoje vedomosti nadobudnuté po splnení praktickej časti úlohy. Test je zobrazený v prehľadnom formáte a obsahuje niekoľko testových otázok, pre ktoré existuje možnosť výberu správnych odpovedí z ponúkaných možností.

Pri štandardnom prístupe k vyhodnocovaniu len na základe konkrétnych zvolených správnych odpovedí v interaktívnom teste by bolo možné v HTML kóde webovej stránky alebo v implementovanej vyhodnocovacej funkcii explicitne uviesť, ktoré odpovede sú správne (napr. vo forme označenia príslušného prvku **checkbox** pre danú možnosť odpovede). Tento prístup by však bol nevyhovujúci, keďže by umožňoval študentom relatívne jednoducho získať správne odpovede ešte pred samotným vyhodnotením testu. Aby bolo možné eliminovať možnosť zistenia správnych odpovedí jednoduchým zobrazením HTML kódu stránky v prípade vyhodnocovania správnosti len na základe zvolených odpovedí, bol zvolený bezpečnejší mechanizmus. Vyhodnocovanie správnosti odpovedí v kontrolnom teste je realizované na základe porovnávania hashov reťazcov reprezentujúcich pre každú otázku kombináciu správnych možností. Za týmto účelom bol vytvorený externý JavaScript súbor s názvom **test-eval.js**, ktorého úlohou je:

- zaznamenať označené odpovede pre každú otázku,
- z označených možností vytvoriť pre príslušnú otázku reťazec v tvare: **q<cislo\_otazky><spravne\_odpovede>**, napr. pre otázku č. 3 s označenými odpoveďami A a B by mal výsledný reťazec podobu: **q3ab**,
- vypočítať SHA-256 hash tohto reťazca, k čomu je využitá externá javascriptová knižnica **sha256.js**,
- porovnať výsledný hash stanovený týmto postupom na základe zadaných odpovedí študenta s preddefinovaným správnym, a teda očakávaným hashom reťazca správnych odpovedí pre príslušnú otázku.

### Generátor hashov pre správne odpovede

Pre uľahčenie správy a s cieľom zvýšenia flexibility úprav či budúceho dopĺňania, resp. rozširovania kontrolných testov o nové otázky a/alebo možnosti odpovedí bez nutnosti používania rôznych externých nástrojov bola vytvorená vlastná pomocná webová stránka slúžiaca ako generátor SHA-256 hashov. Tento generátor je určený predovšetkým pre vyučujúcich a umožňuje zjednodušiť proces doplnenia nových otázok do vytvorených testov.

Generátor má jednoduché a intuitívne používateľské rozhranie. Po načítaní stránky sa zobrazí jedno textové pole, do ktorého používateľ zadá reťazec správnych odpovedí v požadovanom tvare, napr. **q3bd**. Aplikácia následne zabezpečuje:

- **validáciu vstupu** – overuje, či bol zadáný korektný formát vstupného reťazca (napr. q1ac, q2b, atď.),
- **automatické zoradenie písmen označujúcich správne odpovede** – pre zabezpečenie konzistentnosti reťazcov, z ktorých je vytvorený SHA-256 hash pre kontrolu správnosti odpovedí (teda napr. reťazec **q2dac** sa prekonvertuje na **q2acd** ešte pred samotným hashovaním),
- **generovanie SHA-256 hashu** – prostredníctvom javascriptovej knižnice **js-sha256**, ktorá je načítaná z externého zdroja (CDN).

Po spracovaní vstupu sa zobrazia dva výstupy:

- **upravený reťazec** – vstupný reťazec v jednotnom formáte (napr. q2acd),
- **výsledný hash** – 64-miestny hexadecimálny reťazec predstavujúci SHA-256 hash reťazca správnych odpovedí pre zadanú otázku s konkrétnym poradovým číslom.

Používateľ tak môže takto vytvorený hash jednoducho skopírovať a vložiť ho ako očakávanú hodnotu pre príslušnú otázku do JavaScriptového objektu **spravneHash** v súbore **test-eval.js** vo formáte napr.:

```
const spravneHash = {
  q2: 'e9c1e8fefcc96d6e3fa96df65c6e19b7266f3c66e2b9c865f5a6c6cf85bdbb2a'
};
```

Úvod
Teória
Nástroje
Praktická časť
Zadanie
Otázky
Záver

### Kontrolné otázky

*Poznámka: Niektoré otázky môžu mať viac než jednu správnu odpoveď. Pre správne zodpovedanie otázky je nutné označiť **všetky správne** odpovede.*

**1. Akú funkciu plní ARP protokol v rámci sieťovej komunikácie?**

- ☐ Zabezpečuje preklad fyzickej adresy na logickú v lokálnej sieti
- ☐ Priraduje porty k IP adresám
- ☐ Zisťuje fyzickú adresu zariadenia na základe jeho známej IP adresy
- ☐ Poskytuje kryptografickú ochranu komunikácie medzi dvoma zariadeniami

**2. Ktoré z nasledujúcich tvrdení správne popisujú útok typu ARP spoofing?**

- ☐ Útočník odosiela do siete falošné ARP odpovede, aby dosiahol zmenu IP adresy v ARP tabuľke zariadenia
- ☐ Jedná sa o typ útoku, pri ktorom útočník podvrhne svoju MAC adresu namiesto skutočnej MAC adresy zariadenia s hľadanou IP adresou v odpovedi na ARP žiadosť iného zariadenia
- ☐ Cieľom útoku je presmerovať sieťovú komunikáciu cez zariadenie útočníka
- ☐ ARP spoofing sa využíva primárne za účelom narušenia dostupnosti cieľovej služby

Obrázok 5.1 Ukážka (1) vytvorenej webovej stránky: kontrolný test.

Laboratórna úloha č. 3

FEKT

ÚSTAV  
TELEKOMUNIKACÍ

Bezpečnosť spojenej vrstvy

Návrh, správa a bezpečnosť počítačových sítí (MPC-NSB)

Úvod

Teória

Nástroje

Praktická časť

Zadanie

Otázky

Záver

Úvod

Cieľom laboratórnej úlohy je analyzovať zraniteľnosti na úrovni spojenej vrstvy referenčného modelu ISO/OSI a demonštrovať možné riziká a útoky.

V prvej časti laboratórnej úlohy s využitím vhodných nástrojov vo virtuálnom stroji Kali Linux realizujete **simuláciu sieťového útoku ARP spoofing**, počas ktorej sa pokúsíte, akým spôsobom je možné podvrhnúť falošné záznamy do ARP tabuľky klienta, presmerovať sieťovú komunikáciu cez zariadenie útočníka a analyzovať jej obsah. Okrem vykonania samotného útoku, kedy si vyskúšate prácu s nástrojmi **arp spoof** a **Etercap**, sa tiež naučíte analyzovať a vhodne interpretovať zachytené sieťové dáta v prostredí **sieťového analyzátora Wireshark**. Následne budete implementovať ochranné opatrenia (statické ARP záznamy, monitorovanie ARP záznamov) a testovať ich účinnosť.

Teoretický úvod

V tejto laboratórnej úlohe budete oboznámení s **protokolom ARP**, ktorý slúži k prekladu logických adries zariadení na adresy fyzické. Ďalšia časť bude zameraná na problematiku týkajúcu sa bezpečnostných hrozieb na úrovni spojenej vrstvy referenčného modelu ISO/OSI. Vysvetlený bude **princíp útoku ARP Spoofing**, u ktorého bude uskutočnená tiež jeho praktická realizácia.

ARP protokol

ARP protokol (z angl. *Address Resolution Protocol*) je protokol pracujúci na úrovni **spojenej vrstvy** referenčného modelu ISO /OSI, ktorý zabezpečuje mapovanie („preklad“) logickej IP adresy zariadenia na jeho fyzickú adresu (spravidla MAC). Protokol ARP teda slúži na **prevod logických IP adries na fyzické MAC adresy** v lokálnej sieti a napomáha zariadeniam k nájdeniu odpovedajúcej adresy druhej úrovne (t. j. fyzickej adresy) iného zariadenia na základe jeho sieťovej IP adresy (t. j. adresy tretej úrovne). Tento proces je nevyhnutný pre správne smerovanie paketov na spojenej vrstve ISO/OSI modelu. (viď [1] alebo [2])

Obrázok 5.2 Ukážka (2) vytvorenej webovej stránky: teoretický úvod.



| Úvod | Teória | Nástroje | Praktická časť | Zadanie | Otázky | Záver |
|------|--------|----------|----------------|---------|--------|-------|
|------|--------|----------|----------------|---------|--------|-------|

### Nástroj arpspoof

**Arpspoof** je jednoduchý nástroj integrovaný v systéme Kali Linux, ktorý môže byť vhodným prostriedkom k vykonaniu útoku typu ARP spoofing. S pomocou tohto nástroja je možné docieľiť presmerovanie sieťovej prevádzky (resp. komunikácie) medzi zariadeniami v sieti tým, že útočník predstiera identitu iného zariadenia (zvyčajne brány) a rozosiela falošné ARP odpovede s cieľom modifikovať záznamy obsiahnuté v ARP tabuľkách príslušných zariadení. Nástroj arpspoof môže byť využitý pre testovanie zraniteľností ARP protokolu a na analýzu prebiehajúcej komunikácie.

Nástroj arpspoof je v Kali Linux integrovaný ako súčasť balíčka **dsniff**. Jeho inštalácia do prostredia Kali Linux je možná pomocou príkazu:

```
sudo apt update && sudo apt install dsniff -y
```

Štruktúra príkazu nástroja **arpspoof** je nasledujúca:

```
arpspoof [-i interface] [-t target] host
```

kde pomocou prepínača **-i** špecifikujeme konkrétne rozhranie, ktoré je využité pre ARP Spoofing (v prípade napr. ethernetu sa použije typický rozhranie **eth0**), ďalej prepínač **-t** určuje IP adresu „obete“, t. j. cieľového zariadenia, ktorému má byť odoslaná falošná ARP odpoveď. A nakoniec parameter **host** určuje adresu zariadenia, pre ktoré chce útočník monitorovať a zachytávať prichádzajúcu sieťovú komunikáciu na danej linke (resp. pre ktoré podvrhne falošnú ARP odpoveď).

Príklad použitého príkazu nástroja **arpspoof** pre simuláciu útoku ARP spoofing môže byť napr.:

```
arpspoof -i eth0 -t <cieľová_IP> <IP_brány>
```

čo značí, že k realizácii útoku je použité ethernetové rozhranie **eth0**, falošné ARP odpovede sú odoslané zariadeniu s IP adresou **<cieľová\_IP>** a útočník odchyťava komunikáciu prichádzajúcu na zariadenie predstavujúce východziu bránu s IP adresou **<IP\_brány>** danej (lokálnej) siete.

Obrázok 5.3 Ukážka (3) vytvorenej webovej stránky: praktická časť (postup).

## 6. ZÁVER

Cieľom tejto diplomovej práce bolo vytvoriť komplexný a prakticky využiteľný súbor laboratórnych úloh pre výuku predmetu *Návrh, správa a bezpečnosť počítačových sítí* vyučovaného na Fakulte elektrotechniky a komunikačných technológií VUT v Brne v niekoľkých magisterských študijných programoch. Práca svojím zameraním reaguje na aktuálnu potrebu praktickej výuky v oblasti sieťovej bezpečnosti, ktorej význam v kontexte dnešnej digitalizovanej spoločnosti neustále narastá. Vzhľadom na špecifiká predmetu a profil absolventa bolo potrebné vytvoriť metodicky správne spracované, odborne podložené a prakticky aplikovateľné návody, ktoré študentom sprostredkujú nielen teoretické vedomosti, ale predovšetkým poskytnú oporu pri získavaní praktických zručností v oblasti návrhu bezpečnostných mechanizmov a zabezpečenia počítačových sítí a zároveň budú slúžiť ako praktická pomôcka pri riešení zadanej problematiky v prostredí laboratórnej výuky.

V prvej časti práce bol navrhnutý súbor jedenástich tematicky rôznorodých laboratórnych úloh, ktoré pokrývajú kľúčové oblasti návrhu, správy a bezpečnosti počítačových sítí a svojim obsahom nadväzujú na teoretické prednášky predmetu. Každá úloha bola stručne charakterizovaná a začlenená do logickej osnovy výuky podľa svojej náročnosti a tematického zamerania.

Z pripraveného zoznamu laboratórnych úloh pre počítačové cvičenia boli následne vybrané štyri úlohy, pre ktoré boli vytvorené podrobné návody. Tieto návody boli koncipované s dôrazom na didaktickú vhodnosť, prehľadnosť a praktickú využiteľnosť. Každý návod zahŕňa popis cieľov úlohy, teoretický úvod popisujúci základné teoretické poznatky a skutočnosti potrebné pre následnú praktickú realizáciu úlohy, prehľad použitých nástrojov, detailný postup riešenia úlohy, zadanie samostatnej úlohy, súbor kontrolných otázok pre overenie znalostí a záverečné zhrnutie. Ich štruktúra bola navrhnutá tak, aby umožňovala postupné oboznamovanie sa s technickou problematikou a zároveň podporovala samostatnosť a analytické myslenie študentov. Vytvorené návody môžu byť preto efektívne využité v pedagogickom procese s cieľom podporiť rozvoj praktických zručností študentov. Pri ich tvorbe bol kladený dôraz na zrozumiteľnosť, didaktickú vhodnosť a postupné budovanie schopnosti samostatného riešenia úloh. Praktická časť každého návodu bola zároveň doplnená o schému topológie siete a ilustrované ukážky, ktoré napomáhajú orientácii pri práci vo virtuálnom prostredí.

Ako doplnujúci materiál pre jednotlivé návody k príslušným úlohám boli vytvorené sprievodné dokumentácie pre vyučujúcich. Každá dokumentácia dopĺňa samotný návod o očakávané výstupy praktickej činnosti študentov, riešenia samostatných úloh a odpovede na kontrolné otázky, čím uľahčuje hodnotenie a riadenie výučby.

Na realizáciu praktickej časti laboratórnych úloh boli vytvorené tri samostatné virtuálne stroje s predinštalovaným systémom Kali Linux. Tieto stroje boli nakonfigurované tak, aby bolo možné pomocou ich kombinácie simulovať rôzne

scenáre v rámci vytvorených laboratórnych úloh, napr. klient–server, MitM útočník, prístupový bod, RADIUS server, anonymný klient Tor siete a iné. Inštalácia a potrebná konfigurácia týchto strojov prebehla v rámci testovania vytvorených návodov pre vybrané úlohy, vďaka čomu sú vytvorené virtuálne stroje plne pripravené pre nasadenie do výuky. Vytvorené virtuálne stroje<sup>24</sup> s potrebnou konfiguráciou budú neskôr poskytnuté vyučujúcemu ako hotové výučbové prostredie, ktoré môže byť okamžite využité v rámci počítačových cvičení bez potreby dodatočných zásahov.

Významným prínosom tejto práce je aj vytvorenie HTML verzií všetkých štyroch praktických návodov, ktoré boli vytvorené za účelom následného zverejnenia návodov a bezproblémového prístupu k nim pre potreby ich nasadenia do praktickej výuky. Webové stránky boli vytvorené pomocou technológií HTML, CSS a JavaScript, pričom zachovávajú pôvodnú štruktúru dokumentov. Vďaka tomu sú návody prístupné online prostredníctvom bežného webového prehliadača, sú responzívne, prehľadné a dostupné aj na mobilných zariadeniach. Táto forma zverejnenia zároveň umožňuje prípadné úpravy vo vytvorených návodoch, jednoduchú aktualizáciu obsahu a jeho možné rozšírenie o ďalšie prvky. Plánované je zverejnenie návodov v prostredí systému Moodle priamo v e-learningovom kurze predmetu MPC-NSB, kde môžu zároveň slúžiť spoločne s ďalšími materiálmi ako ľahko dostupný študijný materiál. Vytvorené HTML stránky tak predstavujú nielen prostriedok vhodný pre samotnú praktickú realizáciu jednotlivých úloh, ale môžu pozitívne podporiť celkový priebeh výuky a dosahovanie výučbových cieľov celého predmetu.

Veľkým benefitom vytvorených HTML stránok je aj doplňujúca interaktívna funkcionalita vo forme kontrolných testov, ktorá umožňuje študentom samostatne preveriť úroveň porozumenia riešenej problematike. S cieľom zabezpečiť objektivnosť a eliminovať možnosť triviálneho získania správnych odpovedí prostým nahliadnutím do zdrojového kódu stránky bol vyhodnocovací mechanizmus implementovaný tak, že jednotlivé kombinácie správnych odpovedí pre každú zo zadaných testových otázok sú uložené výhradne vo forme SHA-256 hashov. Samotné vyhodnotenie prebieha porovnaním vypočítaného hashu zo zadaných odpovedí s preddefinovanou hodnotou, čo výrazne zvyšuje integritu testovacieho procesu a zároveň neodhaľuje správne odpovede ani pri prehliadaní zdrojového kódu. Tento spôsob zabezpečenia bol podporený aj vytvorením samostatného generátora hashov, ktorý umožňuje vyučujúcim komfortne dopĺňať nové otázky do testu a zabezpečiť ich automatizované vyhodnocovanie bez potreby zásahu do logiky samotnej webovej stránky.

Návody k vybraným laboratórnym úlohám, či už v textovej podobe alebo vo forme webových stránok, sú ako prílohy k tejto záverečnej práci poskytnuté v dvoch jazykoch: slovenská verzia návodov bola vytvorená z dôvodu zachovania jednotnosti s jazykom

---

<sup>24</sup> Pripravené virtuálne stroje nebolo možné z kapacitných dôvodov vložiť do informačného systému spoločne s ostatnými prílohami k tejto záverečnej práci, a z toho dôvodu boli odovzdané vedúcemu práce osobne.

textovej časti tejto záverečnej práce, pričom bol následne uskutočnený ich preklad do českého jazyka, nakoľko sú tieto verzie návodov vhodnejšie pre nasadenie do výuky.

Výsledkom práce je teda nielen prvotný návrh a spracovanie súboru laboratórnych úloh, ale predovšetkým plnohodnotný praktický výstup vo forme štandardizovaných návodov a predpripraveného virtuálneho prostredia, ktoré môžu byť bezprostredne implementované do výuky. Práca tak prispieva k zvýšeniu kvality výuky predmetu a zároveň podporuje nielen zvyšovanie úrovne teoretických poznatkov a súhrnných vedomostí, ale predovšetkým praktické kompetencie študentov v oblasti návrhu, správy a zabezpečenia počítačových sietí.

## LITERATÚRA

- [1] Kali Linux. *The most advanced Penetration Testing Distribution*. In: kali.org [online]. Dostupné z: <https://www.kali.org/> [cit. 2024-11-23].
- [2] VMware. *Using VMware Workstation Player for Windows*. *VMware Workstation Player for Windows 17.0* [online]. 2024. Dostupné z: <https://docs.vmware.com/en/VMware-Workstation-Player-for-Windows/17.0/workstation-player-16-windows-user-guide.pdf> [cit. 2024-10-16].
- [3] Kali Linux – *Introduction*. *What is Kali Linux & Kali's features* [online]. 2024. Dostupné z: <https://www.kali.org/docs/introduction/> [cit. 2024-11-13].
- [4] SHARPE, R. Warnicke, E. Lamping, U. *Wireshark User's Guide. Version 4.5.0* [online]. 2024. Dostupné z: [https://www.wireshark.org/docs/wsug\\_html\\_chunked/](https://www.wireshark.org/docs/wsug_html_chunked/) [cit.2024-11-23].
- [5] SATIH, Sachin. *Top 18 Kali Linux Tools for 2025*. *Simplilearn*. [online]. 2024. Dostupné z: <https://www.simplilearn.com/top-kali-linux-tools-article> [cit. 2024-11-13].
- [6] *Google Nexus 7 Pwn Pad – Kali Linux*. [online]. 2014. [cit. 2024-11-13]. Dostupné z: <https://www.soom.cz/clanky/1142--Google-Nexus-7-Pwn-Pad-Kali-Linux>
- [7] *OWASP Top 10:2021*. *Owasp.org* [online]. 2021. Dostupné z: <https://owasp.org/Top10/> [cit.2024-11-13].

## ZOZNAM SYMBOLOV SKRATIEK

|                |                                                                              |
|----------------|------------------------------------------------------------------------------|
| <b>AAA</b>     | Authentication, Authorization and Accounting                                 |
| <b>AH</b>      | Authentication Header                                                        |
| <b>AP</b>      | Access Point (prístupový bod bezdrôtovej siete)                              |
| <b>ARP</b>     | Address Resolution Protocol                                                  |
| <b>BPDU</b>    | Bridge Protocol Data Unit                                                    |
| <b>CHAP</b>    | Challenge Handshake Authentication Protocol                                  |
| <b>CPU</b>     | Central Processing Unit                                                      |
| <b>CSRF</b>    | Cross-Site Request Forgery                                                   |
| <b>CSS</b>     | Cascading Style Sheets                                                       |
| <b>CTF</b>     | Capture the Flag                                                             |
| <b>DoS</b>     | Denial of Service                                                            |
| <b>DDoS</b>    | Distributed Denial of Service                                                |
| <b>DHCP</b>    | Dynamic Host Configuration Protocol                                          |
| <b>DNAT</b>    | Destination Network Address Translation                                      |
| <b>DNS</b>     | Domain Name System                                                           |
| <b>DNSSEC</b>  | Domain Name System Security Extensions                                       |
| <b>EAP</b>     | Extensible Authentication Protocol                                           |
| <b>EAP-TLS</b> | EAP Transport Layer Security                                                 |
| <b>ESP</b>     | Encapsulating Security Payload                                               |
| <b>FW</b>      | firewall                                                                     |
| <b>GPG</b>     | GNU Privacy Guard                                                            |
| <b>GUI</b>     | Graphical User Interface                                                     |
| <b>HTML</b>    | Hypertext Markup Language                                                    |
| <b>HTTP</b>    | Hypertext Transfer Protocol                                                  |
| <b>HTTPS</b>   | Hypertext Transfer Protocol Secure                                           |
| <b>I2P</b>     | Invisible Internet Project                                                   |
| <b>ICMP</b>    | Internet Control Management Protocol                                         |
| <b>IEEE</b>    | Institute of Electrical and Electronics Engineers                            |
| <b>IKE</b>     | Internet Key Exchange                                                        |
| <b>IP</b>      | Internet Protocol                                                            |
| <b>IPv6</b>    | Internet Protocol version 6                                                  |
| <b>IPsec</b>   | Internet Protocol Security                                                   |
| <b>LAN</b>     | lokálna sieť (z angl. <i>Local Area Network</i> )                            |
| <b>LDAP</b>    | Lightweight Directory Access Protocol                                        |
| <b>LMS</b>     | Learning Management System                                                   |
| <b>MAN</b>     | mestská (tiež metropolitná) sieť (z angl. <i>Metropolitan Area Network</i> ) |
| <b>MAC</b>     | Media Access Control                                                         |
| <b>MitM</b>    | Man in the Middle                                                            |

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| <b>MPC-NSB</b> | vyučovaný predmet <i>Návrh, správa a bezpečnosť počítačových sítí</i> |
| <b>NAT</b>     | Network Address Translation                                           |
| <b>NGFW</b>    | Next-Generation Firewall                                              |
| <b>OS</b>      | operačný systém                                                       |
| <b>OSPF</b>    | Open Shortest Path First                                              |
| <b>OWASP</b>   | Open Web Application Security Project                                 |
| <b>PAP</b>     | Password Authentication Protocol                                      |
| <b>PGP</b>     | Pretty Good Privacy                                                   |
| <b>PHP</b>     | Hypertext Preprocessor                                                |
| <b>RADIUS</b>  | Remote Authentication Dial-In User Service                            |
| <b>RFC</b>     | Request for Comments                                                  |
| <b>RIP</b>     | Routing Information Protocol                                          |
| <b>RM</b>      | referečný model ISO/OSI                                               |
| <b>ISO/OSI</b> |                                                                       |
| <b>SNAT</b>    | Source Network Address Translation                                    |
| <b>SQL</b>     | Structured Query Language                                             |
| <b>SSH</b>     | Secure Shell                                                          |
| <b>SSID</b>    | Service Set Identifier                                                |
| <b>STP</b>     | Spanning Tree Protocol                                                |
| <b>TCP</b>     | Transmission Control Protocol                                         |
| <b>ToR</b>     | The Onion Routing                                                     |
| <b>TLS</b>     | Transport Layer Security                                              |
| <b>UDP</b>     | User Datagram Protocol                                                |
| <b>VM</b>      | virtuálny stroj (z angl. <i>virtual machine</i> )                     |
| <b>WEP</b>     | Wired Equivalent Privacy                                              |
| <b>WPA2</b>    | Wi-Fi Protected Access                                                |
| <b>WPA3</b>    | Wi-Fi Protected Access 2                                              |
| <b>WPS</b>     | Wi-Fi Protected Access 3                                              |
| <b>XXS</b>     | Cross-Site Scripting                                                  |

## ZOZNAM PRÍLOH

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| PRÍLOHA A - TEXT LABORATÓRNEJ ÚLOHY Č. 3 .....                            | 60  |
| PRÍLOHA B - DOKUMENTÁCIA PRE VYUČUJÚCEHO K LABORATÓRNEJ ÚLOHE Č. 3 .....  | 86  |
| PRÍLOHA C - TEXT LABORATÓRNEJ ÚLOHY Č. 4 .....                            | 91  |
| PRÍLOHA D - DOKUMENTÁCIA PRE VYUČUJÚCEHO K LABORATÓRNEJ ÚLOHE Č. 4 .....  | 112 |
| PRÍLOHA E - TEXT LABORATÓRNEJ ÚLOHY Č. 8 .....                            | 117 |
| PRÍLOHA F - DOKUMENTÁCIA PRE VYUČUJÚCEHO K LABORATÓRNEJ ÚLOHE Č. 8 .....  | 145 |
| PRÍLOHA G - TEXT LABORATÓRNEJ ÚLOHY Č. 11 .....                           | 150 |
| PRÍLOHA H - DOKUMENTÁCIA PRE VYUČUJÚCEHO K LABORATÓRNEJ ÚLOHE Č. 11 ..... | 171 |
| PRÍLOHA I - OBSAH PRILOŽENÉHO ARCHÍVU .....                               | 178 |



## **Príloha A - Text laboratórnej úlohy č. 3**

Laboratórna úloha č. 3

# **BEZPEČNOSŤ SPOJOVEJ VRSTVY**

# 0. Úvod k laboratórnej úlohe

Cieľom laboratórnej úlohy je analyzovať zraniteľnosti na úrovni spojovej vrstvy referenčného modelu ISO/OSI a demonštrovať možné riziká a útoky.

V prvej časti laboratórnej úlohy s využitím vhodných nástrojov vo virtuálnom stroji Kali Linux realizujete **simuláciu sieťového útoku ARP spoofing**, počas ktorej sa pokúsíte podvrhnúť falošné záznamy do ARP tabuľky klienta, presmerovať sieťovú komunikáciu cez zariadenie útočníka a analyzovať jej obsah. Okrem vykonania samotného útoku, kedy si vyskúšate prácu s nástrojmi **arp spoof** a **Etttercap**, sa tiež naučíte analyzovať a vhodne interpretovať zachytené sieťové dáta v prostredí **sieťového analyzátora Wireshark**. Následne budete implementovať ochranné opatrenia (statické ARP záznamy, monitorovanie ARP záznamov) a testovať ich účinnosť.

## Požiadavky pre vypracovanie úlohy:

- software: VMware Workstation Player pre virtualizáciu staníc,
- virtuálne stroje: tri virtuálne stroje s Kali Linux.

## 1. Teoretický úvod

V tejto laboratórnej úlohe budete oboznámení s **protokolom ARP**, ktorý slúži k prekladu logických adries zariadení na adresy fyzické. Ďalšia časť bude zameraná na problematiku týkajúcu sa bezpečnostných hrozieb na úrovni spojovej vrstvy referenčného modelu ISO/OSI. Vysvetlený bude **princíp útoku ARP spoofing**, u ktorého bude uskutočnená tiež jeho praktická realizácia.

### 1.1. ARP protokol

ARP protokol (z angl. *Address Resolution Protocol*) je protokol pracujúci na úrovni spojovej vrstvy referenčného modelu ISO/OSI, ktorý zabezpečuje mapovanie („preklad“) logickej IP adresy zariadenia na jeho fyzickú adresu (spravidla MAC). Protokol ARP teda slúži zariadeniam v lokálnej sieti k nájdeniu odpovedajúcej adresy druhej úrovne (t. j. fyzickej adresy) iného zariadenia na základe jeho sieťovej IP adresy (t. j. adresy tretej úrovne). [1], [2]

Keď zariadenie potrebuje odoslať IP paket inému uzlu nachádzajúcemu sa v tej istej lokálnej sieti, najprv prostredníctvom ARP požiadavky zisťuje, aká MAC adresa prislúcha k požadovanej IP adrese. Po získaní odpovede odosielateľ vytvorí ethernetový rámec so správnou cieľovou MAC adresou, ktorú mu dané zariadenie poskytlo v zaslanej ARP odpovedi, a odosiela ho na úrovni spojovej vrstvy.

Pre ARP protokol sú známe nedostatky determinujúce zraniteľnosti, v dôsledku ktorých je ARP protokol náchylný na sieťové útoky. Medzi podstatné zraniteľnosti ARP protokolu možno zaradiť nasledujúce:

- **Chýbajúca autentifikácia:** ARP protokol nemá zabudovaný žiadny mechanizmus autentifikácie, v čoho dôsledku môže ktorékoľvek zariadenie v sieti odosielať ARP odpovede bez overenia identity. Tento nedostatok umožňuje útočníkovi podvrhnúť falošnú MAC adresu pre konkrétnu IP adresu a docieľiť tak „otravu“ ARP tabuľky na zariadení odosielajúcom žiadosť *ARP Request* (tzv. *ARP Cache Poisoning* útok).
- **Dynamická ARP tabuľka:** ARP tabuľka je dynamická, čo znamená, že zariadenia automaticky aktualizujú svoje záznamy na základe prijatých ARP odpovedí. Útočník môže popísaný mechanizmus zneužiť tak, že zariadeniu poskytne falošné údaje a prepíše pôvodné hodnoty fyzických adries iných zariadení zaznamenané v ARP tabuľke.
- **ARP odpovede bez vyžiadania:** Zariadenia v sieti môžu prijímať a ukladať ARP odpovede, aj keď si ich samé nevyžiadali (t. j. keď tieto zariadenia neodoslali konkrétnu žiadosť *ARP Request*). Útočník môže využiť uvedenú vlastnosť ARP protokolu k rozosieleniu nesprávnych, falošných ARP odpovedí (tzv. „*gratuitous ARP replies*“), čím môže ovplyvniť obsah ARP tabuliek u zariadení a celkovo manipulovať s priebehom sieťovej komunikácie.
- **Zraniteľnosť voči Man-in-the-Middle útokom:** Vzhľadom na absenciu zabezpečenia môžu útočníci presmerovať dátovú komunikáciu medzi zariadeniami cez svoje vlastné zariadenie úplne transparentne z pohľadu obete. To umožňuje odpočúvanie komunikácie, modifikáciu paketov alebo realizáciu útokov typu *Denial-of-Service* (DoS) o odopretie poskytovania služby.
- **Neexistuje vstavaná ochrana:** Nakoľko ARP protokol pracuje na spojenej vrstve OSI modelu, samotný nemá zabudované mechanizmy, ktoré by umožnili kontrolovať autentickosť prijatých ARP odpovedí, čo môže podnietiť útoky súvisiace s podvrhnutím falošných informácií a „otravou“ ARP tabuľky (viď vyššie). Preto je nutné implementovať dodatočné ochranné opatrenia, medzi ktoré patria: konfigurácia statických ARP záznamov, využitie nástrojov pre ARP monitoring (napr. Arpwatch) alebo definícia pravidiel firewallu obmedzujúcich prijímanie neoprávnených ARP odpovedí.

### ARP tabuľka

Každé zariadenie v sieti si vo svojej pamäti aktívne udržiava tabuľku obsahujúcu záznamy o dvojiciach k sebe prislúchajúcich logických IP adries a fyzických (MAC) adries, tzv. ARP tabuľku (tiež *ARP cache*). Záznamy v ARP tabuľke majú zvyčajne krátku životnosť (napr. 5 minút) a sú v stanovených intervaloch pravidelne

aktualizované<sup>25</sup>. Záznamy k sebe prislúchajúcich dvojíc IP adresy a fyzickej adresy pre jednotlivé zariadenia v danej sieti sa do ARP tabuľky ukladajú buď na základe činnosti samotného ARP protokolu, alebo je možné potrebné informácie zadať do ARP tabuľky aj manuálne.

### Štruktúra ARP správ

ARP správy sú prenášané na spojovej vrstve ISO/OSI modelu, t. j. nevyužívajú žiadny protokol vyššej (IP alebo TCP) a sú zapuzdrené priamo do ethernetového rámca. Protokol ARP funguje nezávisle od vyšších vrstiev a práve vďaka tomu môžu byť ARP správy odosielané aj v prípade, kedy ešte nepoznáme MAC adresu cieľového zariadenia, nakoľko úvodná správa ARP Request, ako bude popísané ďalej, je odoslaná na všesmerovú fyzickú adresu v príslušnej lokálnej sieti.

Protokol ARP definuje dva základné typy správ:

- **ARP Request (žiadosť)** – vysielaná do siete v prípade, že zariadenie potrebuje zistiť MAC adresu príjemcu pre známu IP adresu;
- **ARP Reply (odpoveď)** – odpoveď na žiadosť obsahujúca príslušnú MAC adresu.

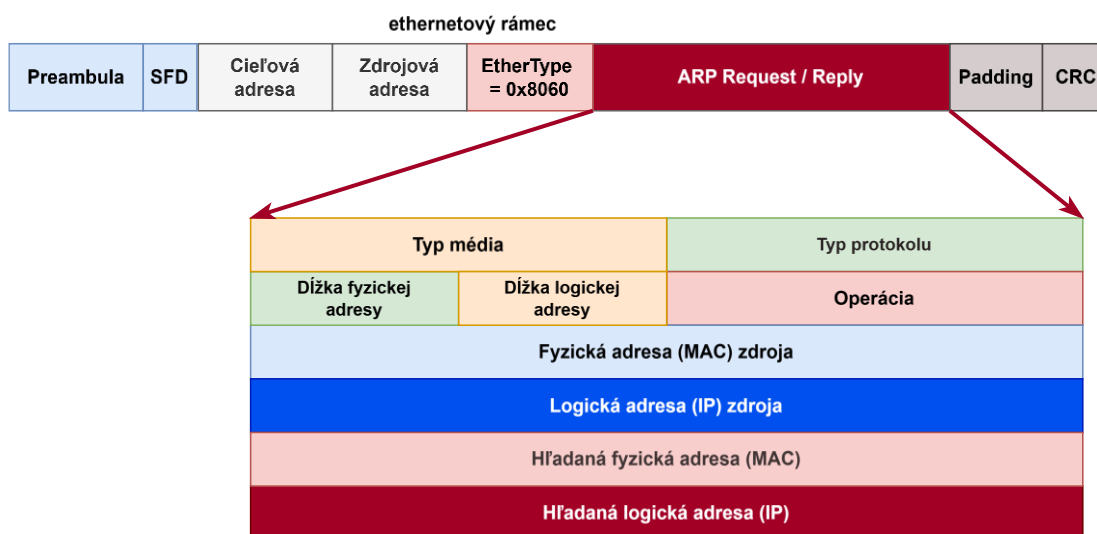
Každá z uvedených ARP správ (*request* alebo *reply*) obsahuje nižšie uvedené polia, jej štruktúra je schematicky znázornená na obr. 1.1:

- **Typ média** (16 b): určuje typ protokolu na spojovej vrstve (napr.: 1 = Ethernet),
- **Typ protokolu** (16 b) – určuje protokol vyššej vrstvy, ktorý využíva ARP (napr. 0x0800 pre IPv4),
- **Dĺžka fyzickej adresy** (8 b) – určuje dĺžku MAC adresy v bajtoch (typicky 6 B);
- **Dĺžka logickej adresy** (8 b) – určuje dĺžku IP adresy v bajtoch (typicky 4 B);
- **Operácia** (16 b) – označuje typ správy (1 = ARP Request, 2 = ARP Reply);
- **Fyzická adresa zdroja** (48 b) – MAC adresa zariadenia, ktoré odosiela ARP správu;
- **Logická adresa zdroja** (32 b) – IP adresa zariadenia, ktoré odosiela ARP správu;
- **Hľadaná fyzická adresa** (48 b) – MAC adresa cieľového zariadenia (v prípade správy ARP Request je toto pole prázdne);
- **Hľadaná logická adresa** (32 b) – IP adresa cieľového zariadenia, pre ktoré sa zisťuje MAC adresa.

ARP správa je následne na spojovej vrstve vložená do ethernetového rámca, ktorého **EtherType pole** je nastavené na hodnotu 0x0806, čím sa indikuje, že dáta prenášané v dátovej časti rámca sú typu ARP.

---

<sup>25</sup> V prípade, kedy nie je záznam v stanovenej dobe aktualizovaný, dochádza k jeho trvalému odstráneniu z prekladovej ARP tabuľky v pamäti zariadenia.



Obrázok 1.1 Všeobecná štruktúra správy ARP protokolu a jej zapuzdrenie do ethernetového rámca.

## Popis fungovania ARP protokolu, výmena správ

### 1. ARP Request (Žiadosť):

Zariadenie, ktoré pre svojho komunikačného partnera (iné zariadenie) potrebuje zistiť fyzickú MAC adresu pre konkrétnu IP adresu tohto zariadenia, najprv skontroluje svoju ARP tabuľku. V prípade, že požadovaný záznam nie je v ARP tabuľke obsiahnutý, vyšle žiadosť *ARP Request* vo forme broadcastu na adresu **FF:FF:FF:FF:FF:FF**, ktorú prijmu všetky zariadenia v lokálnej sieti. V žiadosti uvedie svoju IP a MAC adresu spolu s IP adresou cieľa, ktorú chce nájsť.

### 2. ARP Reply (Odpoveď):

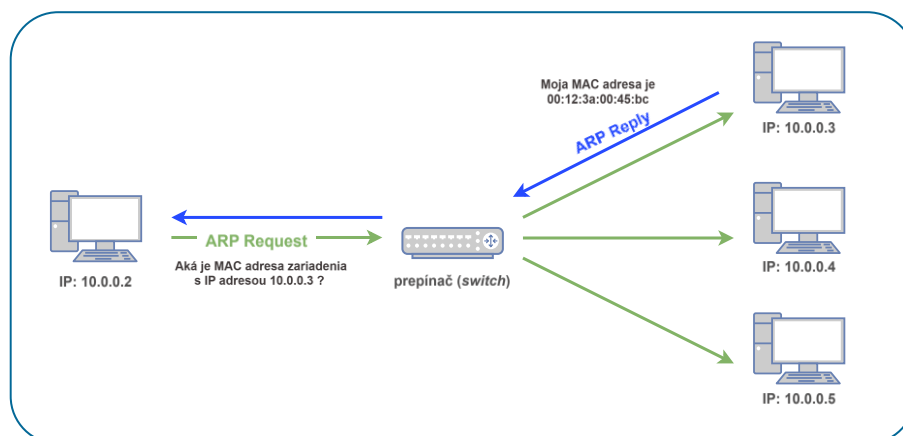
Zariadenie s požadovanou (hľadanou) IP adresou odpovedá unicastovou správou *ARP Reply* adresovanou priamo zariadeniu iniciujúcemu ARP komunikáciu, ktoré odoslalo do siete správu *ARP Request*. V tejto odpovedi odosiela informáciu o svojej MAC adrese vyplnením príslušného poľa v ARP záhlaví. Ostatné stanice v sieti prijatú správu *ARP Request* ignorujú a zahodia.

### 3. Vytvorenie záznamu v ARP tabuľke:

Po prijatí odpovede si žiadajúce zariadenie uloží IP a k nej odpovedajúcu hľadanú MAC adresu cieľového zariadenia do ARP tabuľky vo svojej pamäti. Ak už zariadenie pozná MAC adresu z predchádzajúcej komunikácie, použije pri ďalšej komunikácii uložený záznam bez nutnosti opakovaného odosielania obdobnej ARP žiadosti.

#### 4. Timeout a obnova:

Záznamy v ARP tabuľke majú obmedzenú platnosť (typicky napr. 5 minút). Po uplynutí času sa z ARP tabuľky vymažú, aby sa predišlo neaktuálnym údajom pri zmene konfigurácie siete.



Obrázok 1.2 Schematické znázornenie výmeny správ ARP protokolu<sup>26</sup>.

V prípade záujmu o rozšírenie poznatkov ohľadom ARP protokolu a jeho fungovania je odporúčené nahliadnuť do ďalšej literatúry, ktorú poskytujú napr. zdroje [3], [4].

### 1.2. Hrozby na spojovej vrstve: ARP spoofing

**ARP spoofing** je typ sieťového útoku používaný útočníkmi k presmerovaniu komunikácie v lokálnej sieti (LAN) prostredníctvom manipulácie s ARP protokolom (*Address Resolution Protocol*). Tento protokol sa využíva na mapovanie sieťových IP adries na fyzické adresy zariadení v rámci danej lokálnej siete. Útok *ARP spoofing* patrí do skupiny tzv. MitM útokov, kedy sa útočník dostáva do pozície prostredníka v rámci komunikácie dvoch komunikujúcich strán.

Základný princíp útoku *ARP spoofing* teda spočíva v tom, že útočník po zachytení žiadostí *ARP Request* od iných zariadení vyšle do siete podvrhnuté ARP odpovede, v ktorých uvedie falošné mapovanie požadovanej IP adresy na fyzickú adresu svojho zariadenia, vďaka čomu sa môže vydávať za očakávané zariadenie (napr. za východziu bránu zo siete alebo server). Cieľové zariadenie, ktoré pôvodnú *ARP* žiadosť o mapovanie odoslalo, si zapíše tento podvrhnutý zápis do svojej ARP tabuľky, a tak bude všetka ďalšia komunikácia odoslaná na zadanú IP adresu príjemcu následne presmerovaná na zariadenie útočníka. Týmto spôsobom môže útočník zachytávať a manipulovať sieťovú prevádzku.

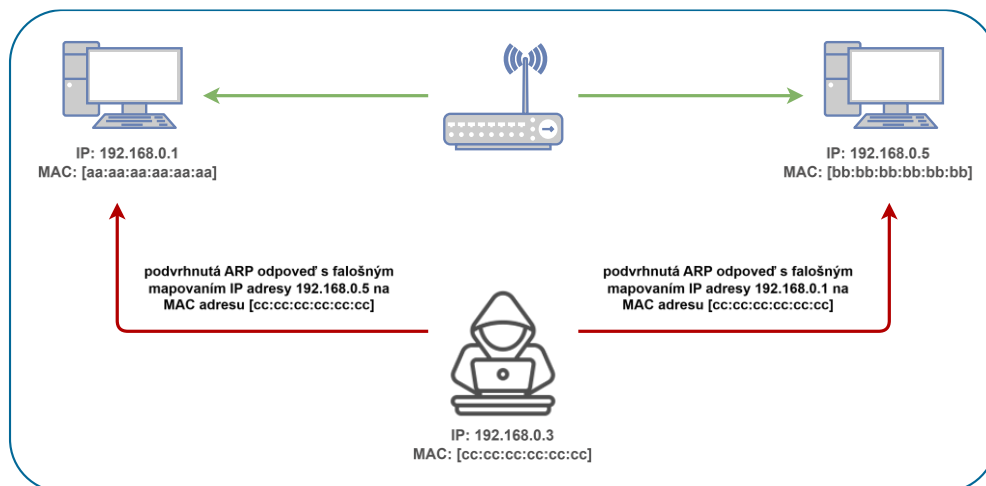
<sup>26</sup> Prevzaté z [3].

## Priebeh útoku

Pri útoku *ARP spoofing* útočník najskôr identifikuje cieľové zariadenia v sieti, napríklad klienta a bránu (*gateway*). Následne začne odosielať falošné ARP odpovede, v ktorých informuje obeť (klienta), že jeho MAC adresa patrí bráne, a zároveň druhé zariadenie, t. j. bránu v podvrhnutých ARP odpovediach informuje, že jeho MAC adresa patrí klientovi. Týmto spôsobom dôjde k presmerovaniu celej sieťovej komunikácie cez útočnickovo zariadenie.

Pokiaľ sa útočníkovi podarí popísaný útok úspešne realizovať, jeho zariadenie sa dostáva do pozície MitM, kedy je cez dané zariadenie preposielaná všetka legitímna komunikácia medzi klientom a bránou, vďaka čomu môže útočník zachytávať, modifikovať alebo blokovať sieťovú komunikáciu medzi týmito dvoma zariadeniami. V závislosti na ďalších použitých nástrojoch môže útočník ďalej napríklad sledovať prihlasovacie údaje, odpočúvať nešifrovaná dáta, presmerovať prebiehajúce dátové prenosy na iný cieľový server alebo realizovať útoky typu DoS (*Denial of Service*) s cieľom útočiť na celkovú dostupnosť siete.

Podrobný popis *ARP spoofing* útoku možno nájsť v [5], [6].



Obrázok 1.3 Schematické znázornenie priebehu *ARP spoofing* útoku.<sup>27</sup>

## Ochranné opatrenia proti *ARP spoofing* útoku

Útoky typu *ARP spoofing* možno do značnej miery obmedziť, resp. zmierniť ich dopad prostredníctvom implementácie vhodných ochranných mechanizmov.

Jedným zo základných preventívnych opatrení je **statické nastavenie ARP záznamov** na dôležitých zariadeniach v sieti, čím sa zabráni ich prepísaniu v dôsledku prijatia neočakávaných ARP odpovedí s falošne priradenou dvojicou IP ↔ MAC adresa.

<sup>27</sup> Prevzaté z [7].

Medzi doplnkové ochranné opatrenia, ktoré je možné proti *ARP spoofing* útokom nasadiť, patrí mechanizmus filtrovania MAC adries, tzv. **MAC filtering**, a to najmä v menších alebo staticky konfigurovaných sieťach. Ide o bezpečnostný mechanizmus, pri ktorom sieťové zariadenie (napr. prepínač alebo smerovač) povoľuje pripojenie len tým zariadeniam, ktorých MAC adresy sú vopred povolené v zozname. Týmto spôsobom je možné zamedziť neoprávnenému zariadeniu (napr. útočníkovi) prístup do siete, čím sa znižuje riziko manipulácie s obsahom ARP tabuliek.

V sieťovej infraštruktúre podporujúcej pokročilejšie funkcie je možné nasadiť **Dynamic ARP Inspection** (DAI) na sieťových prvkoch (typicky prepínačoch), ktorá porovnáva ARP odpovede so známymi, dôveryhodnými údajmi a v prípade nenájdenia zhody blokuje podozrivú prevádzku, resp. povolí prijatie len legitímnych odpovedí.

Ďalšou možnosťou je **segmentácia siete a použitie VLAN**, čím sa obmedzí dosah potenciálneho útočníka vďaka izolovaniu vybraných zariadení do virtuálnej siete. A taktiež je možné využiť nástroje umožňujúce detekciu podozrivej aktivity, spomedzi ktorých možno spomenúť napríklad **arpwatch**, ktoré dokážu upozorniť správcu siete na náhle zmeny MAC adries v ARP tabuľkách. Pravidelné monitorovanie a analýza sieťovej prevádzky pomocou Wiresharku tiež prispieva k rýchlej identifikácii potenciálnych nežiadúcich hrozieb.

### 1.3. Ďalšie hrozby na úrovni spojovej vrstvy

Útok *ARP spoofing* nie je jedinou známou hrozbou vyskytujúcou sa v počítačových sieťach na úrovni spojovej vrstvy. Existuje množstvo ďalších útokov, ktoré môžu byť vykonané na spojovej vrstve. Nižšie bude uvedených niekoľko z nich.

#### MAC Flooding

MAC Flooding je útok cielený najmä na sieťové prepínače (switch), ktorý spočíva v „zaplavení“ tabuľky MAC adries na prepínači veľkým množstvom falošných MAC adries. Pamäť prepínača disponuje obmedzenou kapacitou pre ukladanie informácií mapovania MAC adries na príslušné porty, a v prípade, kedy dôjde k prekročeniu dostupnej kapacity, degraduje prepínač svojou funkčnosťou na hub, čo bude mať za následok, že prepínač začne preposielať prijaté rámce ďalej do všetkých portov, čo umožní útočníkovi odpočúvať celú sieťovú komunikáciu.

Vhodnou obranou proti MAC Flooding útoku môže byť použitie a správna konfigurácia mechanizmu **Port Security** na prepínači, ktorý obmedzuje počet pripojených MAC adries na jednom porte a ďalej umožňuje stanoviť zoznam povolených MAC adries na príslušnom porte.

#### VLAN Hopping

VLAN Hopping je útok, ktorý umožňuje útočníkovi získať neoprávnený prístup k iným VLAN segmentom v sieti. VLAN (*Virtual Local Area Network*) je technológia



umožňujúca logické oddelenie sieťovej prevádzky celej siete v rámci jedného fyzického prepínača alebo skupiny viacerých prepínačov. Každá VLAN sieť tak predstavuje samostatný segment siete, čo prakticky znamená, že zariadenia v rôznych VLAN sa nemôžu medzi sebou priamo dorozumievať (i keď sa fyzicky nachádzajú v spoločnej lokálnej sieti) bez použitia smerovača alebo špeciálne definovaných pravidiel. Siete VLAN sa používajú najmä za účelom zvýšenia bezpečnosti, zjednodušenie managementu a správy siete a tiež pre rozdelenie celkovej prevádzky a záťaže v počítačovej sieti, vďaka čomu je možné minimalizovať riziko preťaženia, resp. nežiadúceho zahľtenia siete.

Útok VLAN Hopping obchádza túto segmentáciu a umožňuje útočníkovi komunikovať s VLAN, do ktorej by nemal mať prístup. Pri útoku dochádza k zneužitiu nesprávnej konfigurácie sieťového prepínača, ktorého úlohou je prenos dát medzi VLAN. VLAN Hopping môže byť vykonaný dvoma hlavnými spôsobmi: **switch spoofing**, kedy sa útočník vydáva za dôveryhodný, legítimný prepínač a získava prístup k viacerým VLAN, a tzv. **double tagging**, pri ktorom útočník manipuluje so značkami VLAN (tzv. VLAN tag) v záhlaví ethernetových rámcov, čo bude mať za následok presmerovanie paketov do inej VLAN siete.

Za účelom zaistenia vhodnej ochrany pred VLAN Hoppingom je potrebné dbať na správne nastavenie konfigurácie sieťových prepínačov, zakázať možnosť automatického vytvárania trunkov a obmedziť VLAN *tagging* len na dôveryhodné zariadenia.

## Útoky na Spanning Tree Protocol

**Spanning Tree Protocol (STP)** je sieťový protokol, ktorého použitie umožní zamedziť vzniku sieťových slučiek v ethernetových sieťach s redundantnými spojeniami. Nesprávna konfigurácia preposielania rámcov, v ktorej dôsledku dochádza k využívaniu záložných ciest pre komunikáciu, môže mať za následok zahľtenie siete, opakované preposielanie paketov a celkovo nežiadúcim spôsobom ovplyvniť komunikáciu v danej sieti. Protokol STP umožňuje zariadeniam, resp. prepínačom aktívne identifikovať redundantné cesty v sieti a dočasne niektoré porty deaktivovať, aby bolo možné zabrániť vzniku smerovacích slučiek.

Protokol STP používa na výmenu informácií medzi sieťovými prepínačmi špeciálne správy nazývané **BPDU (Bridge Protocol Data Unit)**. BPDU správy pomáhajú určiť hierarchiu prepínačov a vybrať tzv. **root bridge**, t. j. hlavný prepínač, ktorý predstavuje referenčný bod pre výpočet najefektívnejších ciest v sieti s redundantnými spojeniami, teda pre vytvorenie kostry siete slúžiacej pre vytvorenie prenosových ciest medzi uzlami s cieľom eliminovať vznik smerovacích slučiek.

Útočník môže popísaný mechanizmus zneužiť odosielaním falošných BPDU správ s nižšou prioritou, v dôsledku čoho sa bude jeho zariadenie pre ostatné prepínače v sieti ako root bridge. Následne môže ovplyvniť konštrukciu kostry siete a docieľiť vedenie komunikácie cez svoje zariadenie, čím neoprávnene získa možnosť odpočúvať a prípadne i manipulovať s dátovým obsahom. Okrem toho môže útočník opakovane meniť

topológiu siete neustálym posielaním podvrhnutých BPDU správ, čo môže mať za následok narušenie prevádzky, celkovú nestabilitu siete, časté zmeny v prepojeniach medzi portami a nakoniec môže viesť až k fatálnym výpadkom alebo úplnému narušeniu sieťovej komunikácie.

## 1.4. Kali Linux a použité nástroje

V rámci tejto laboratórnej úlohy budú pre útok ARP spoofing a analýzu komunikácie postupne využité nižšie uvedené nástroje:

- **arpspoof**: jednoduchý nástroj umožňujúci posielanie falošných ARP odpovedí.
- **Ettercap**: pokročilý MitM nástroj s možnosťou ARP *spoofingu* a schopnosťou cieľenej modifikácie sieťovej komunikácie.
- **Wireshark**: sieťový analyzátor určený pre monitorovanie prebiehajúcej sieťovej komunikácie (resp. jednotlivých dátových paketov) a možnú následnú analýzu manipulovaných ARP odpovedí.
- **Arpwatch**: nástroj na monitorovanie ARP zmien a detekciu podvrhnutých MAC adries.

### Nástroj arpspoof

Arpspoof je jednoduchý nástroj integrovaný v systéme Kali Linux, ktorý môže byť vhodným prostriedkom k vykonaniu útoku typu ARP spoofing. S pomocou tohto nástroja je možné docieľiť presmerovanie sieťovej prevádzky (resp. komunikácie) medzi zariadeniami v sieti tým, že útočník predstiera identitu iného zariadenia (zvyčajne brány) a rozosiela falošné ARP odpovede s cieľom modifikovať záznamy obsiahnuté v ARP tabuľkách príslušných zariadení. Nástroj arpspoof môže byť využitý pre testovanie zraniteľností ARP protokolu a na analýzu prebiehajúcej komunikácie.

Nástroj arpspoof je v Kali Linux integrovaný ako súčasť balíčka `dsniff`. Jeho inštalácia do prostredia Kali Linux je možná pomocou príkazu:

```
sudo apt update && sudo apt install dsniff -y
```

Štruktúra príkazu nástroja arpspoof je nasledujúca:

```
arpspoof [-i interface] [-t target] host
```

kde pomocou prepínača **-i** špecifikujeme konkrétne rozhranie, ktoré je využité pre ARP *spoofing* (v prípade napr. ethernetu sa použije typicky rozhranie `eth0`), ďalej prepínač **-t** určuje IP adresu „obete“, t. j. cieľového zariadenia, ktorému má byť odoslaná falošná ARP odpoveď. A nakoniec parameter **host** určuje adresu zariadenia, pre ktoré chce útočník monitorovať a zachytávať prichádzajúcu sieťovú komunikáciu na danej linke (resp. pre ktoré podvrhne falošnú ARP odpoveď). [8]

Príklad použitého príkazu nástroja `arp spoof` pre simuláciu útoku ARP spoofing môže byť napr.:

```
arp spoof -i eth0 -t <cieľová_IP> <IP_brány>
```

čo značí, že k realizácii útoku je použité ethernetové rozhranie `eth0`, falošné ARP odpovede sú odoslané zariadeniu s IP adresou `<cieľová_IP>` a útočník odchyťáva komunikáciu prichádzajúcu na zariadenie predstavujúce východziu bránu s IP adresou `<IP_brány>` danej (lokálnej) siete.

### Nástroj `ettercap`

Ettercap predstavuje komplexný nástroj, ktorý je taktiež súčasťou Kali Linux, vhodný pre analýzu sieťovej prevádzky a realizáciu MitM útokov, a to vrátane ARP spoofingu. Ponúka väčšie spektrum funkcií v porovnaní s nástrojom `arp spoof` a podporuje interaktívne sledovanie a manipuláciu s paketmi v reálnom čase. Môže byť použitý prostredníctvom príkazového riadku (terminál) alebo je možné pre prácu s nástrojom využiť užívateľsky prívetivé grafické rozhranie<sup>28</sup>. Medzi hlavné poskytované funkcie nástroja `ettercap` patrí:

- automatická detekcia zariadení v sieti,
- realizácia MitM útokov vrátane ARP spoofingu a DNS spoofingu a tiež možnosť manipulácie s obsahom správ HTTPS a iných protokolov,
- možnosť používania vlastných filtrov na zmenu obsahu paketov. [9]

V rámci praktickej časti laboratórnej úlohy budú postupne použité oba zmienené nástroje k simulácii útoku ARP spoofing. Zásadné rozdiely medzi predstavenými nástrojmi uvádza priložená tabuľka 1.1.

### Analyzátor sieťovej komunikácie

Wireshark je jeden z najpoužívanějších nástrojov na zachytávanie a analýzu sieťových paketov. Používa sa na sledovanie prebiehajúcej dátovej komunikácie s cieľom diagnostiky problémov v sieti, odhaľovanie podozrivej aktivity a skúmanie prípadných bezpečnostných incidentov v monitorovanej sieti. Tento nástroj umožňuje používateľovi detailne sledovať sieťovú komunikáciu a analyzovať dátové jednotky jednotlivých protokolov na všetkých vrstvách ISO/OSI modelu.

Pre účely vypracovania tejto laboratórnej úlohy bude nástroj Wireshark použitý na monitorovanie komunikácie ARP protokolu, identifikáciu podvrhnutých ARP odpovedí počas útoku ARP spoofing a na overenie správnosti implementácie ochranných opatrení a ich efektivity v ochrane voči uvedenému typu útoku.

---

<sup>28</sup> Pre účely vypracovania úloh tohto laboratórneho cvičenia bude nástroj `ettercap` v Kali Linux používaný v jeho grafickom režime.


Tabuľka 1.1 Porovnanie nástrojov arpspoof a ettercap.

| <i>Funkcia:</i>            | <b>arpspoof</b>                                                     | <b>ettercap</b>                                                                                                            |
|----------------------------|---------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <i>Úroveň komplexnosti</i> | Jednoduchý nástroj vhodný pre simuláciu útoku ARP <i>spoofing</i> . | Komplexný nástroj poskytujúci viaceré možnosti, ako napr.: analýza sieťových parametrov, dostupnosť zariadení, ...         |
| <i>Podpora filtrov</i>     | Nie                                                                 | Áno                                                                                                                        |
| <i>Grafické rozhranie</i>  | Nie                                                                 | Áno                                                                                                                        |
| <i>Typy útokov</i>         | ARP <i>spoofing</i>                                                 | ARP <i>spoofing</i> , DNS <i>spoofing</i> , generovanie vlastných paketov, možnosť neoprávnenej manipulácie s obsahom, ... |

Nástroj Wireshark umožňuje nasledovné:

- **Zachytávanie sieťovej prevádzky** – Wireshark umožňuje sledovať všetku komunikáciu prebiehajúcu cez vybrané sieťové rozhranie (Ethernet, Wi-Fi, tunelované pripojenia, virtuálne adaptéry atď.).
- **Analýza paketov** – Umožňuje detailné skúmanie obsahu paketov na všetkých vrstvách OSI modelu, vrátane podrobného zobrazenia a analýzy záhlavia protokolov ako napr. ARP, TCP, UDP, ICMP, DNS či HTTP.
- **Filtrovanie paketov** – Pomocou použitia filtrov umožňuje zobrazíť len potrebné, relevantné dáta, napríklad iba zachytenú ARP komunikáciu alebo HTTP požiadavky. Filtrovanie je možné na základe protokolov, IP adries, MAC adries, portov a ďalších parametrov.
- **Rekonštrukcia sieťovej komunikácie** – Wireshark umožňuje analyzovať kompletný priebeh komunikácie medzi zariadeniami v sieti, a to vrátane jej obsahu (napr. sledovanie obsahu nezašifrovanej komunikácie protokolu HTTP, analýza požiadaviek a odpovedí).
- **Sledovanie podozrivej aktivity, detekcia útokov a bezpečnostných hrozieb** – Nástroj Wireshark je možné využiť i k odhaľovaniu podvrhnutých ARP odpovedí, MitM útokov, DoS útokov a iných bezpečnostných incidentov.
- **Exportovanie a spracovanie dát** – Zachytené dátové pakety (komunikáciu) je možné uložiť do .pcap súborov a analyzovať neskôr.

### **Základné príkazy Wiresharku (CLI verzia – TShark)**

Nástroj Wireshark je možné využiť aj v jeho „príkazovej“ podobe **TShark**, ktorá umožňuje monitorovať a analyzovať sieťovú komunikáciu v systéme Linux z prostredia terminálu . Nižšie je uvedených niekoľko užitočných príkazov:

- Zachytávanie paketov na konkrétnom rozhraní:

```
tshark -i eth0
```

*Uvedený príkaz spustí zachytávanie sieťovej prevádzky na rozhraní `eth0`.*

- Použitie filtra k zobrazeniu len paketov ARP protokolu:

```
tshark -i eth0 -Y "arp"
```

- Zachytenie paketov a uloženie do súboru:

```
tshark -i eth0 -w nazov_suboru.pcap
```

- Zobrazenie iba požadovaných polí v paketoch:

```
tshark -r zachyt.pcap -T fields -e ip.src -e ip.dst
```

*Uvedený príkaz zobrazí len zdrojovú `ip.src` a cieľovú `ip.dst` IP adresu zachytených paketov.*

V rámci tejto laboratórnej úlohy bude využitá verzia nástroja Wireshark umožňujúca využitie všetkých jeho funkcionalít prostredníctvom grafického užívateľského rozhrania. Podrobnejší popis možností Wiresharku a ich využitia pre potreby tejto úlohy bude uvedený neskôr v praktickej časti tohto návodu.

## ARPwatch

Nástroj ARPwatch sa využíva pre **sledovanie zmien a detekciu anomálií v ARP tabuľkách** na sieťových rozhraniach zariadení a sieťových prvkov a umožňuje vytvárať hlásenia (*alerts*) v prípade výskytu nežiadúcich a/alebo neočakávaných zmien. Použitie tohto nástroja je obzvlášť výhodné najmä v prostredí s veľkým počtom zariadení, kde môže byť užitočným prostriedkom pre rýchlu detekciu potenciálnych sieťových ARP *spoofing* útokov.

## 2. Praktická časť

V rámci praktickej časti bude **realizovaný útok typu MitM, konkrétne ARP spoofing**. Simulovaný útok bude prebiehať vo virtuálnej sieti pozostávajúcej z troch virtuálnych strojov s Kali Linux (klient, server a útočník), ktorej topológia je schematicky znázornená nižšie na obr. 2.1. Komunikácia medzi uvedenými virtuálnymi strojmi prebieha skrz virtuálny switch VMware virtual switch, ako je znázornené na uvedenom obrázku. Cieľom úspešnej realizácie ARP spoofing útoku je modifikovať sieť tak, aby všetka komunikácia medzi virtuálnymi strojmi klienta a serveru prebiehala výhradne cez zariadenie útočníka.

### 2.1. Topológia virtuálnej siete a nastavenie virtuálnych strojov

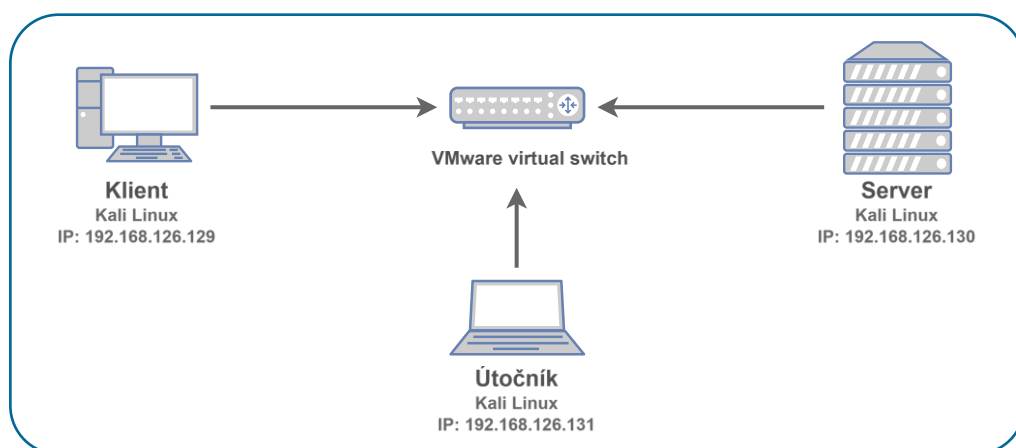
#### Použité virtuálne stroje:

- obeť (klient): bežný počítač v sieti, cieľ útoku
- server: poskytovateľ sieťovej služby (napr. webserver, *gateway* = brána)
- útočník: vykonáva ARP spoofing, zachytáva sieťovú komunikáciu medzi klientom a serverom

#### Sieťová konfigurácia:

- obeť: 192.168.126.129
- server: 192.168.126.130
- útočník: 192.168.126.131

Všetky virtuálne stroje budú pripojené do rovnakej virtuálnej siete (nastavenie sieťového adaptéra v režime napr. **Bridged** alebo **Host-Only**), aby mohlo byť na VM predstavujúcom „útočníka“ realizované zachytávanie dátových prenosov (komunikácie) medzi klientom a serverom.



Obrázok 2.1 Topológia siete laboratórnej úlohy.

## 2.2. Zoznámenie sa s použitými nástrojmi

### Prehľad základných príkazov pre jednotlivé používané nástroje

- Zobrazenie ARP tabuľky:

```
arp -a
```

*Dobrovoľné: Pred zahájením praktickej časti si vyskúšajte použitie uvedeného príkazu na zariadeniach klienta a severu. Pozorujte záznamy uvedené v ARP tabuľke na oboch zariadeniach.*

- ARP spoofing pomocou nástroja arpspoof:

```
sudo arpspoof -i eth0 -t 192.168.1.10 192.168.1.1
```

- Spustenie Ettercapu cez terminál:

```
sudo ettercap -Tq -M arp:remote /192.168.1.10/ /192.168.1.1/
```

### Použitie Wiresharku na analýzu ARP spoofing útoku

#### 1. Spustenie zachytávania sieťovej prevádzky

Po spustení programu Wireshark je potrebné vybrať monitorované sieťové rozhranie, cez ktoré prebieha komunikácia (napr. eth0 pre Ethernet). Po jeho výbere kliknutím na tlačidlo **Start** spustíte zachytávania paketov.

#### 2. Filtrovanie paketov ARP protokolu

Pre zobrazenie dátových jednotiek prislúchajúcich ku komunikácii ARP protokolu, je vhodné použiť filter pre komunikáciu: **arp**. Tento filter zobrazí len ARP žiadosti a ARP odpovede odosielané v monitorovanej sieti.

#### 3. Identifikácia podvrhnutých ARP odpovedí

V rámci analýzy zachytenej dátovej komunikácie je nutné zamerať sa na rôzne prvky podozrivej aktivity, ktorá môže indikovať prebiehajúci ARP spoofing útok. Môže sa jednať napr. o:

- neočakávané ARP odpovede bez predchádzajúcich ARP žiadostí,
- rovnaké IP adresy priradené k rôznym fyzickým MAC adresám,
- časté opakovanie ARP odpovedí smerujúcich na jedno cieľové zariadenie (obet').

#### 4. Ukladanie a analýza dát

Zachytené pakety zaznamenanaj ARP komunikácie je možné vo Wiresharku uložiť do samostatného .pcap súboru a analyzovať neskôr pomocou príkazu:

```
tshark -r subor.pcap | grep ARP
```

## 2.3. Postup pre vypracovanie laboratórnej úlohy

### A) Príprava prostredia

#### Spustenie virtuálnych strojov:


- Otvorte VMware Workstation Pro (umiestnený na ploche).
- Spustíte postupne všetky tri virtuálne stroje (klient, server, útočník). Uistite sa v správnosti konfigurácie sieťových parametrov, overte priradenie IP adries.
- Prihláste sa do prostredia Kali Linux na VM útočníka.

VM „útočník“ – prihlasovacie údaje: **Username: kali**, **Password: kali**

VM „klient“ – prihlasovacie údaje: **Username: klient**, **Password: kali**

VM „server“ – prihlasovacie údaje: **Username: server**, **Password: kali**

#### Overenie sieťovej konektivity:

- Otvorte **terminál** (kliknite na ikonu terminálu  v záhlaví horného pracovného panelu alebo stlačte **Ctrl + Alt + T**).
- Na každom VM si zobrazte priradené IP adresy (na rozhraní **eth0**) pomocou príkazu:

```
ip a
```

- Z klienta vyskúšajte pripojenie na server pomocou príkazu **ping**.

```
ping <ip_adresa_servera>
```

Ak prichádza odpoveď **ping echo reply** zo strany servera, sieťová komunikácia medzi zariadeniami funguje.

- Obdobným spôsobom overte možnosť spojenia v opačnom smere komunikácie.
- Po overení funkčnosti spojenia si **zobrazte prekladové ARP tabuľky** na oboch zariadeniach pomocou príkazu:

```
arp -a
```

#### Spustenie Wiresharku a sledovanie ARP paketov

- Na klientovi spustíte Wireshark pomocou príkazu:

```
sudo wireshark &
```

Prepínač **'sudo'** spustí nástroj Wireshark s oprávneniami administrátora, čo je nevyhnutné pre zachytávanie sieťovej prevádzky v Kali Linux.



- V hlavnom okne vyberte sieťové rozhranie (napr. **eth0**).
- Do okna pre filtrovanie napíšte **arp** a použitie zvoleného filtra komunikácie potvrdíte stlačením **Enter**.  
Použitie filtra **'arp'** zaistí, že spomedzi všetkých zachytených dátových jednotiek prenesených v rámci komunikácie cez zvolené rozhranie, budú zobrazené len správy protokolu ARP.
- Kliknite na **Start Capturing Packets**.

## B) Vykonanie ARP spoofing útoku

Princípom simulácie útoku bude dosiahnuť „otravu“ ARP tabuliek na zariadení klienta a na serveri, a to v dôsledku podvrhnutia falošných ARP odpovedí zo strany útočníka, ktoré budú oznamovať skutočnosť, že server so sieťovou IP adresou **192.168.126.130** má priradenú fyzickú MAC adresu odpovedajúcu MAC adrese zariadenia útočníka [**cc: cc: cc: cc: cc: cc**] a obdobne v druhom smere komunikácie bude serveru poskytnutá informácia, že klient s IP adresou **192.168.126.129** tak isto disponuje fyzickou MAC adresou zariadenia útočníka [**cc: cc: cc: cc: cc: cc**] (viď obr. 1.3).

Podvrhnutie ARP odpovedí obom komunikujúcich zariadeniam bude mať za následok odosielanie všetkej komunikácie v smere **server → klient** a tak isto v opačnom smere **klient → server** na zariadenie útočníka. Útočník môže zachytené správy modifikovať a následne v zmenenej podobe preposlať koncovému adresátovi, čo bude mať za následok narušenie integrity komunikácie.

### Použitie nástroja arpspoof

- Pred spustením samotného útoku je potrebné povoliť presmerovanie IP paketov na zariadení útočníka, aby mohol správne sprostredkovať komunikáciu medzi klientom a serverom:

```
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

Použitie uvedeného príkazu zabezpečí, že útok nebude spôsobovať výpadok prebiehajúceho legitímneho spojenia medzi klientom a serverom, no zároveň bude útočník schopný komunikáciu zachytávať a preposielať – čo je *typický priebeh Man-in-the-Middle útoku*.

- Na útočnickom stroji ďalej otvorte terminál (nové terminálové okno) a spustíte nasledujúce príkazy pre vykonanie ARP *spoofing* útoku postupne pre oba smery komunikácie. **Pre podvrhnutie ARP odpovede klientovi použijete príkaz:**

```
sudo arpspoof -i eth0 -t 192.168.126.129 192.168.126.130
```

*\*\* do príkazu arpspoof vstupujú ako parametre jednotlivých prepínačov hodnoty na základe uvedenej topológie vytvorenej pre túto laboratórnu úlohu*

- Následne opakujte postup, ale tentokrát **pre podvrhnutie ARP odpovedí serveru:**

```
sudo arpspoof -i eth0 -t 192.168.126.130 192.168.126.129
```

Po použití uvedených příkazů budou generované falošné ARP odpovědi a následně odeslané ze stroje útočníka zařízením s uvedenými IP adresami, vďaka čomu sa útočník sa dostane do pozície prostredníka komunikácie MitM medzi klientom a serverom.

- Po zadání příkazů začne nástroj **arpspoof** opakovane odosielať na uvedené cieľové IP adresy žiadosti ARP Request. **Je nutné nechať tento príkaz na oboch zariadeniach (klient, server) spustený po celú dobu trvania útoku!** Akékoľvek ďalšie príkazy je vhodné spustiť v samostatnom okne terminálu.
- Zadanie postupne oboch **arpspoof** príkazů zaručí, že je útok ARP *spoofing* kompletný. Všetka komunikácia prebiehajúca v smere *klient* → *server* a tiež *server* → *klient* bude od tohto momentu prechádzať cez zariadenie útočníka.

#### Presmerovanie komunikácie – overenie útoku

- Po spustení **arpspoof** z klienta na server aj zo servera na zariadenie klienta je možné overiť úspešnosť útoku jednoduchým príkazom **ping**. Na klientovi spustíte:

```
ping 192.168.126.130
```

- Na zariadení útočníka sledujte, či sa generované pakety protokolu ICMP objavujú v zachytenej komunikácii vo Wiresharku. Alternatívne je možné overiť úspešnosť útoku aj prostredníctvom nástroja **tcpdump**, a to nasledovným príkazom:

```
sudo tcpdump -i eth0 icmp
```

Uvedený príkaz zobrazuje **všetky ICMP pakety** (napr. *ping*), ktoré prechádzajú cez rozhranie **eth0**. Ak útok prebieha správne a je zapnuté IP forwarding, na zariadení útočníka budú zachytené odeslané správy ICMP echo a príslušné a odpovede (*reply*) medzi klientom a serverom.

#### Overenie zmien v ARP tabuľke klienta

- Na zariadení klienta si zobrazte ARP tabuľku zadáním príkazu do nového terminálového okna a pozorujte zmeny porovnaním obsahu tabuľky pred vykonaním útoku:

```
arp -a
```

Ak je útok ARP *spoofing* úspešný, MAC adresa prislúchajúca k IP adrese serveru bude v prekladovej ARP tabuľke zmenená na MAC adresu zariadenia útočníka.

- Obdobne postupujte i pri kontrole ARP tabuľky na serveri.

```
(klient@kali-klient)-[~]
$ arp -a
? (192.168.142.2) at 00:50:56:f0:7a:e5 [ether] on eth0
? (192.168.142.254) at 00:50:56:ec:e3:69 [ether] on eth0
? (192.168.142.130) at 00:0c:29:80:63:d6 [ether] on eth0
? (192.168.142.128) at 00:0c:29:b6:d3:cc [ether] on eth0

(klient@kali-klient)-[~]
$ arp -a
? (192.168.142.2) at 00:50:56:f0:7a:e5 [ether] on eth0
? (192.168.142.254) at 00:50:56:ec:e3:69 [ether] on eth0
? (192.168.142.130) at 00:0c:29:b6:d3:cc [ether] on eth0
? (192.168.142.128) at 00:0c:29:b6:d3:cc [ether] on eth0
```

Obrázok 2.2 Zmeny v ARP tabuľke klienta.

### Overenie detekcie útoku pomocou nástroja arpwatch

- Po úspešnom spustení útoku ARP *spoofing* môžete overiť detekciu zmien v ARP tabuľke aj pomocou nástroja arpwatch.
- Na jednom z napadnutých strojov (klient/server) otvorte terminálové okno a nainštalujte arpwatch pomocou príkazov:

```
sudo apt update
sudo apt install arpwatch -y
```

- Spustíte arpwatch na sieťovom rozhraní, ktorým je napadnuté zariadenie pripojené do siete, v ktorej prebieha útok:

```
sudo arpwatch -i eth0
```

- Počas toho, kým prebieha simulovaný útok pomocou arpspoof, sledujte výstup arpwatch v termináli alebo kontrolujte logovací súbor:

```
sudo tail -f /var/log/syslog
```

alebo:

```
sudo cat /var/lib/arpwatch/arp.dat
```

- Všímajte si správy, ako napríklad:

```
changed ethernet address for 192.168.126.130
ethernet address 00:0c:29:b6:d3:cc found at 192.168.126.130
```

alebo:

**arpwatch:** ethernet address for **192.168.126.130** changed from **00:0c:29:d6:30:5b** to **00:0c:29:b6:d3:cc**

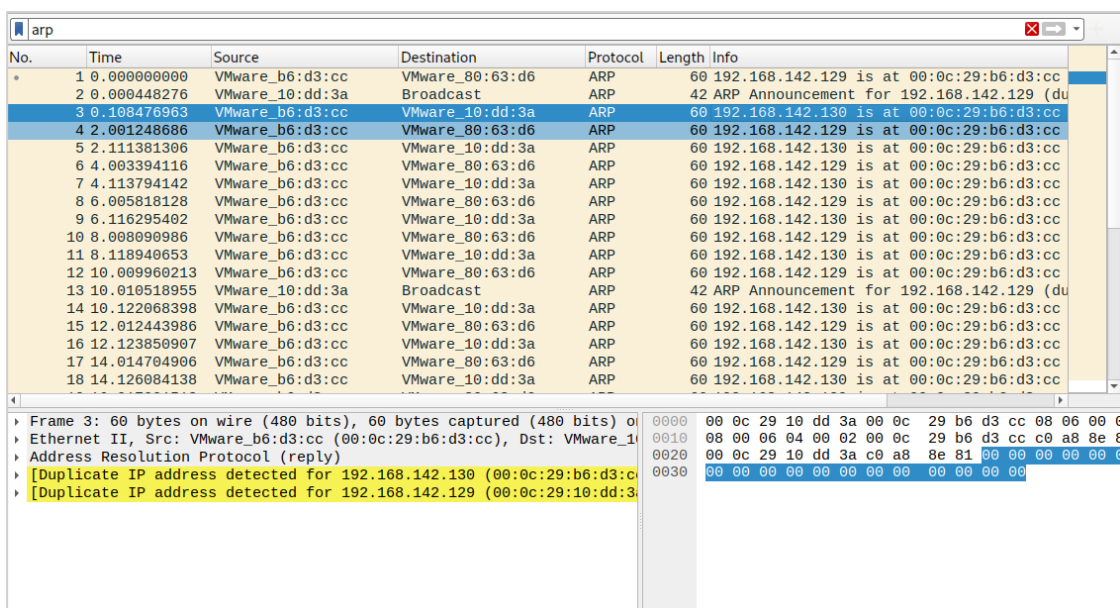
Zobrazené správy (a zmeny v ARP tabuľke klienta) indikujú, že uvedená IP adresa (server) bola spárovaná s novou MAC adresou (útočníka) – čo je typický dôsledok ARP spoofingu útoku.

### Zachytávanie a analýza dát vo Wiresharku

V rámci tejto laboratórnej úlohy bude nástroj Wireshark využitý primárne za účelom **monitorovania prebiehajúcej ARP komunikácie** vo vytvorenej virtuálnej sieti a k následnej analýze zachytených správ (žiadostí a odpovedí) ARP protokolu.

- Vráťte sa do Wiresharku.
- Pozorujte podvrhnuté ARP odpovede a analyzujte ich obsah.

Kliknutím na paket zobrazíte jeho detaily. Skontrolujte, **ako sa zmenili riadiace informácie v záhlaví ARP protokolu** po vykonaní ARP *spoofing* útoku (zdrojová a cieľová MAC adresa by mali byť zmenené na útočnickovu).



| No. | Time         | Source          | Destination     | Protocol | Length | Info                                     |
|-----|--------------|-----------------|-----------------|----------|--------|------------------------------------------|
| 1   | 0.000000000  | VMware_b6:d3:cc | VMware_80:63:d6 | ARP      | 60     | 192.168.142.129 is at 00:0c:29:b6:d3:cc  |
| 2   | 0.000448276  | VMware_10:dd:3a | Broadcast       | ARP      | 42     | ARP Announcement for 192.168.142.129 (du |
| 3   | 0.108476963  | VMware_b6:d3:cc | VMware_10:dd:3a | ARP      | 60     | 192.168.142.130 is at 00:0c:29:b6:d3:cc  |
| 4   | 2.001248686  | VMware_b6:d3:cc | VMware_80:63:d6 | ARP      | 60     | 192.168.142.129 is at 00:0c:29:b6:d3:cc  |
| 5   | 2.111381306  | VMware_b6:d3:cc | VMware_10:dd:3a | ARP      | 60     | 192.168.142.130 is at 00:0c:29:b6:d3:cc  |
| 6   | 4.003394116  | VMware_b6:d3:cc | VMware_80:63:d6 | ARP      | 60     | 192.168.142.129 is at 00:0c:29:b6:d3:cc  |
| 7   | 4.113794142  | VMware_b6:d3:cc | VMware_10:dd:3a | ARP      | 60     | 192.168.142.130 is at 00:0c:29:b6:d3:cc  |
| 8   | 6.005818128  | VMware_b6:d3:cc | VMware_80:63:d6 | ARP      | 60     | 192.168.142.129 is at 00:0c:29:b6:d3:cc  |
| 9   | 6.116295402  | VMware_b6:d3:cc | VMware_10:dd:3a | ARP      | 60     | 192.168.142.130 is at 00:0c:29:b6:d3:cc  |
| 10  | 8.008090986  | VMware_b6:d3:cc | VMware_80:63:d6 | ARP      | 60     | 192.168.142.129 is at 00:0c:29:b6:d3:cc  |
| 11  | 8.118940653  | VMware_b6:d3:cc | VMware_10:dd:3a | ARP      | 60     | 192.168.142.130 is at 00:0c:29:b6:d3:cc  |
| 12  | 10.009960213 | VMware_b6:d3:cc | VMware_80:63:d6 | ARP      | 60     | 192.168.142.129 is at 00:0c:29:b6:d3:cc  |
| 13  | 10.010518955 | VMware_10:dd:3a | Broadcast       | ARP      | 42     | ARP Announcement for 192.168.142.129 (du |
| 14  | 10.122068398 | VMware_b6:d3:cc | VMware_10:dd:3a | ARP      | 60     | 192.168.142.130 is at 00:0c:29:b6:d3:cc  |
| 15  | 12.012443986 | VMware_b6:d3:cc | VMware_80:63:d6 | ARP      | 60     | 192.168.142.129 is at 00:0c:29:b6:d3:cc  |
| 16  | 12.123850907 | VMware_b6:d3:cc | VMware_10:dd:3a | ARP      | 60     | 192.168.142.130 is at 00:0c:29:b6:d3:cc  |
| 17  | 14.014704906 | VMware_b6:d3:cc | VMware_80:63:d6 | ARP      | 60     | 192.168.142.129 is at 00:0c:29:b6:d3:cc  |
| 18  | 14.126084138 | VMware_b6:d3:cc | VMware_10:dd:3a | ARP      | 60     | 192.168.142.130 is at 00:0c:29:b6:d3:cc  |

Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on  
Ethernet II, Src: VMWare\_b6:d3:cc (00:0c:29:b6:d3:cc), Dst: VMWare\_10:dd:3a (00:0c:29:10:dd:3a)  
Address Resolution Protocol (reply)  
[Duplicate IP address detected for 192.168.142.130 (00:0c:29:b6:d3:cc)]  
[Duplicate IP address detected for 192.168.142.129 (00:0c:29:10:dd:3a)]

Obrázok 2.3 Ukážka zachytenej komunikácie (Wireshark)

### ARP spoofing pomocou nástroja Ettercap

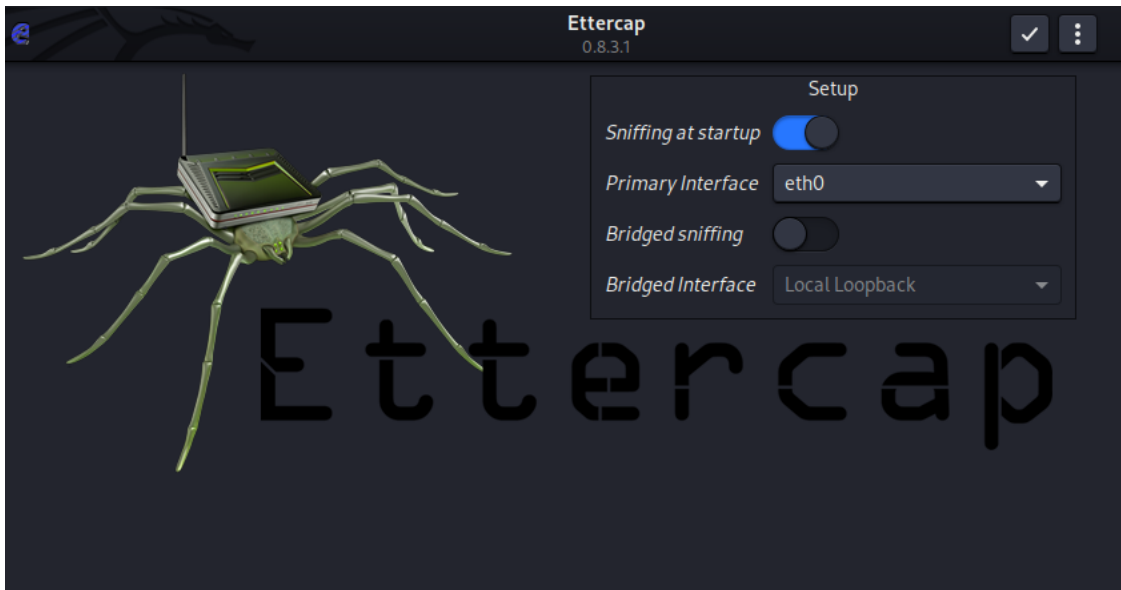
- Na stroji útočníka spustíte Ettercap cez terminál:

```
sudo ettercap -G
```

- Ako primárne sieťové rozhranie vyberte eth0.

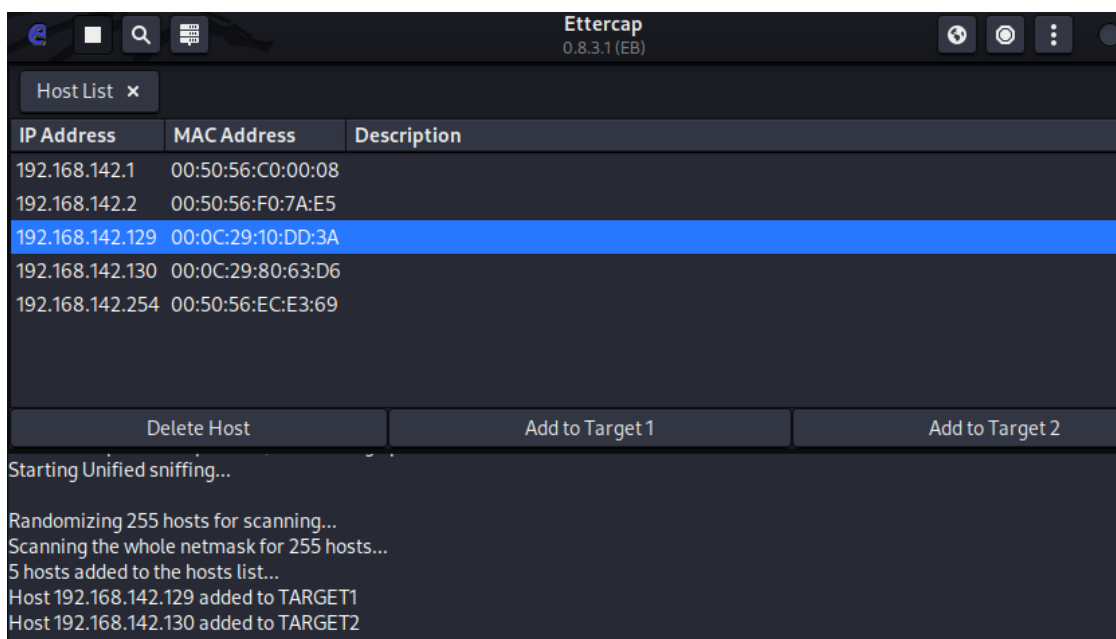
- Po spustení ponechajte všetky nastavenia bez zmeny (viď obr. 2.4) a pokračujte kliknutím na ☒ v záhlaví zobrazeného okna.

Kliknutím na **Options** (tri bodky v záhlaví) sa uistite, že je zaškrtnutá voľba **Promisc mode**, čo v priebehu simulácie zabezpečí, že zariadenie útočníka bude pracovať v promiskuitnom režime, vďaka čomu bude možné odchyťovať všetku komunikáciu na lokálnej linke prechádzajúcu cez zvolené rozhranie.



Obrázok 2.4 Nástroj Ettercap: úvodné grafické rozhranie.

- Pred výberom cieľových zariadení je potrebné vykonať *sken* siete. Kliknite na „tri bodky“ v záhlaví, ďalej zvolíte **Hosts > Scan for hosts**, čím naplníte zoznam dostupných zariadení v sieti.
- Po dokončení skenovania si zoznam zariadení zobrazíte cez **Hosts > Hosts List** (viď obr. 2.5).
- Z tohto zoznamu vyberte cieľové zariadenia *ARP spoofing* útoku a pridajte ich postupne do **Target 1** (klient) a **Target 2** (server).
- Ďalej pokračujte kliknutím na MitM Menu (ikona zemegule) záhlaví grafického rozhrania, z ponuky vyberte na **MitM > ARP Poisoning**, zaškrtnite **Sniff remote connections** a túto voľbu potvrdíte pomocou **OK**, čím zahájite útok.



Obrázok 2.5 Nástroj Ettercap: výpis nájdených zariadení v sieti.

### Overenie úspešnosti MitM útoku pomocou Ettercap

- Po spustení útoku v prostredí nástroja Ettercap v režime MitM, môžete jeho úspešnosť overiť obdobne ako v prípade prvého útoku: pomocou aplikácie **ping** (resp. odoslaním ICMP *echo request* správy z klienta na server):

```
ping 192.168.126.130
```

Ak odosielané správy ICMP protokolu prechádzajú cez zariadenie útočníka (resp. ak sú tieto pakety viditeľné v zachytenej komunikácii vo Wiresharku alebo ak program Ettercap zobrazí záznam o komunikácii medzi týmito IP adresami), znamená to, že ARP *spoofing* útok prebehol z pohľadu úspešne a komunikácia bola presmerovaná.

## 2.4. Samostatná úloha

### C) Implementácia ochranných opatrení

V poslednej časti laboratórnej úlohy si prakticky vyskúšate možnosti ochrany proti útoku ARP *spoofing*.

**Cieľom vašej samostatnej práce bude implementovať statické ARP záznamy, ktoré predstavujú jeden z možných druhov ochranných opatrení proti zmienenému typu útokov, a následné otestovanie účinnosť tejto ochrany.**

#### Konfigurácia statických ARP záznamov

- Na klientovi a serveri zadefinujte statické ARP záznamy:

```
sudo arp -s 192.168.1.1 00:11:22:33:44:55  
sudo arp -s 192.168.1.20 AA:BB:CC:DD:EE:FF
```

*\*\* **Poznámka:** ako fyzické MAC adresy a sieťové IP adresy voľte **konkrétne adresy** vami používaných VMs.*

- Overte uloženie vami zadaných statických záznamov zobrazením ARP tabuľky na oboch zariadeniach výpisom aktuálneho obsahu ARP tabuľky.

#### Testovanie účinnosti ochrany

- Opakujte útok pomocou nástroja `arpspoof` podľa predošlého postupu a overte, či nedochádza ku zmene MAC adres v ARP tabuľke.
- V prípade nemennosti informácií v záznamoch v ARP tabuľke na oboch VMs (klient, sever) je implementovaná ochrana voči podvrhnutiu falošných MAC adres vďaka nastaveniu statických záznamov v ARP tabuľke účinná.

### 3. Záver

V tejto laboratórnej úlohe ste sa zoznámili s problematikou bezpečnosti spojovej vrstvy počítačových sietí. Prakticky ste overili **priebeh útoku ARP spoofing**, v rámci ktorého útočník podvrhnutím falošných ARP odpovedí docielil uvedenie nesprávnych informácií o fyzických MAC adresách v prekladových ARP tabuľkách komunikujúcich zariadení, čo môže mať za následok presmerovanie komunikácie práve skrz zariadenie útočníka.

Účinnou ochranou proti útokom založených na „otrave“ prekladovej ARP tabuľky je **konfigurácia statických záznamov** pre mapovanie medzi sieťovými a fyzickými adresami, ktorá zamedzí získavaniu informácií o MAC adresách s využívaním ARP protokolu medzi zariadeniami v sieti, a tým aj nežiaducim zneužitím ARP odpovedí k podvrhnutiu falošnej fyzickej (MAC) adresy.

#### 3.1. Kontrolné otázky

1. Akú funkciu plní ARP protokol v rámci sieťovej komunikácie?
  - A) Zabezpečuje preklad fyzickej adresy na logickú v lokálnej sieti
  - B) Priradzuje porty k IP adresám
  - C) Zisťuje fyzickú adresu zariadenia na základe jeho známej IP adresy
  - D) Poskytuje kryptografickú ochranu komunikácie medzi dvoma zariadeniami
2. Ktoré z nasledujúcich tvrdení správne popisujú útok typu ARP spoofing?
  - A) Útočník odosiela do siete falošné ARP odpovede, aby dosiahol zmenu IP adresy v ARP tabuľke zariadenia
  - B) Jedná sa o typ útoku, pri ktorom útočník podvrhne svoju MAC adresu namiesto skutočnej MAC adresy zariadenia s hľadanou IP adresou v odpovedi na ARP žiadosť iného zariadenia
  - C) Cieľom útoku je presmerovať sieťovú komunikáciu cez zariadenie útočníka
  - D) ARP spoofing sa využíva primárne za účelom narušenia dostupnosti cieľovej služby
3. Aký je rozdiel medzi dynamickým a statickým ARP záznamom?
  - A) Dynamický záznam je uložený trvalo, statický len dočasne
  - B) Statický je uložený manuálne, dynamický sa generuje automaticky
  - C) Dynamický záznam sa nikdy neaktualizuje podľa aktuálnej situácie v sieti
  - D) Dynamický je bezpečnejší ako statický
4. Prečo je pri MitM útoku dôležité zapnúť IP forwarding?
  - A) Aby bolo možné odosielať pakety cez zabezpečené HTTPS spojenie
  - B) Pretože umožní odosielanie a prijímanie ICMP správ
  - C) Aby útočník mohol presmerovať sieťovú komunikáciu cez svoje zariadenie
  - D) Umožňuje zakázať použitie MAC filtering mechanizmu



5. Ktorý z nasledujúcich nástrojov slúži primárne na analýzu sieťovej komunikácie?
- A) arpspoof
  - B) Ettercap
  - C) Wireshark
  - D) arping
6. Ktoré z nasledujúcich javov môžu naznačovať prebiehajúci ARP *spoofing* v sieti?
- A) Znížená latencia a zvýšená prenosová rýchlosť v sieti
  - B) Výskyt ARP odpovedí, ktoré priradujú rovnakú MAC adresu k viacerým IP adresám
  - C) Výskyt "duplicate IP" varovaní v systéme
  - D) Výskyt viacerých ARP odpovedí bez predchádzajúcich požiadaviek
7. Ktoré tvrdenia vystihujú rozdiely medzi nástrojmi arpspoof a Ettercap?
- A) Ettercap dokáže analyzovať a upravovať dáta vyšších vrstiev (napr. HTTP)
  - B) arpspoof je jednoduchý CLI nástroj bez možnosti manipulácie so samotnými dátami
  - C) Ettercap neumožňuje vizualizáciu MitM útokov cez GUI rozhranie
  - D) arpspoof automaticky obnovuje ARP tabuľky po útoku
8. K čomu slúži nástroj arpspoof počas útoku typu MitM?
- A) Odosiela falošné ARP odpovede, aby sa útočník dostal do pozície medzi dvoma zariadeniami (MitM)
  - B) Skenuje sieť pre zistenie aktívnych služieb
  - C) Skenuje sieť pre zistenie pripojených koncových zariadení
  - D) Blokuje komunikáciu medzi routerom a klientom
9. Ktoré z nasledujúcich opatrení môžu pomôcť chrániť sieť pred ARP *spoofingom*?
- A) Použitie šifrovania TLS
  - B) Konfigurácia statických ARP záznamov
  - C) Nasadenie *Dynamic ARP Inspection* (DAI)
  - D) Použitie VLAN segmentácie
10. Aký filter vo Wiresharku použijete na zobrazenie len ARP paketov (požiadaviek aj odpovedí)?
- A) arp
  - B) ip.arp == 1
  - C) eth.type == 0x0806
  - D) arp.request

## 4. Literatúra

- [1] *ARP (Address Resolution Protocol)*. *SecuriaPro.sk* [online]. Dostupné z: <https://www.secu-riapro.sk/slovník-pojmov/arp/> [cit. 2024-11-23].
- [2] Noite.pl. *ARP Protocol – Address Resolution Protocol*. In: *Network Basic. AL0- 012* [online]. 2016. s. 118–130. Dostupné z: <https://books.google.sk/books?id=wcFxCwAAQBAJ> [cit. 2024-11-23].
- [3] *What Is Address Resolution Protocol (ARP)?* *fortinet.com* [online]. Dostupné z: <https://www.fortinet.com/resources/cyberglossary/what-is-arp> [cit. 2024-11-23].
- [4] ATKINSON, RJ. *Address Resolution Protocol (ARP) for the Identifier-Locator Network Protocol for IPv4 (ILNPv4)*. In: *Internet Requests for Comments*. [online]. RFC Editor, 2012. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc6747> [cit. 2024-11-23].
- [5] WAGNER, R. *Address Resolution Protocol Spoofing and Man in the Middle Attacks*. SANS Institute. 2001. [online]. Dostupné z: <https://www.sans.org/reading-room/whitepapers/threats/address-resolution-protocol-spoofing-man-in-the-middle-attacks-474> [cit. 2024-11-23].
- [6] Al Sukkar, G. Saifan, R. Khwaldeh, S. Maqableh, M. et Jafar, I. *Address Resolution Protocol (ARP): Spoofing Attack and Proposed Defense*. In: *Communications and Network*. 2016. s. 118–130. [online]. Dostupné z: <http://hdl.handle.net/123456789/856> [cit. 2024-11-23].
- [7] MORSY, Sabah M. and NASHAT, Dalia. *D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing*. In: *IEEE Access*. 2022. s. 49142–49153. [online]. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9766351> [cit. 2024-11-23].
- [8] *arp spoof(8) – Linux man page*. In: *Linux Documentation*. [online]. Dostupné z: <https://linux.die.net/man/8/arp spoof> [cit. 2024-11-26].
- [9] Ettercap project. [online]. Dostupné z: <https://www.ettercap-project.org/> [cit. 2024-11-26].

## **Príloha B - Dokumentácia pre vyučujúceho k laboratórnej úlohe č. 3**

Laboratórna úloha č. 3

### **BEZPEČNOSŤ SPOJOVEJ VRSTVY**

# 1. Základné informácie k laboratórnej úlohe

Laboratórna úloha č. 3 je zameraná na problematiku bezpečnosti spojovej vrstvy. V teoretickej časti sú diskutované zraniteľnosti na úrovni spojovej vrstvy RM OSI a v nadväzujúcej praktickej časti je **realizovaný útok ARP spoofing**.

Študenti realizujú simuláciu útoku typu MitM s využitím troch virtuálnych strojov so systémom Kali Linux, pričom postupne využijú nástroje **arp spoof**, Ettercap, Wireshark a prípadne **tcpdump**. Cieľom laboratórnej úlohy je porozumieť princípu fungovaniu ARP protokolu, simulovať útok typu ARP *spoofing*, analyzovať jeho dopad na prebiehajúcu komunikáciu a navrhnúť ochranné opatrenia proti tomuto typu útoku.

## 2. Očakávané výstupy práce študentov

Úlohou študentov je postupne podľa krokov podrobne popísaných v priloženom návode **simulovať útok ARP spoofing s využitím nástroja arp spoof** v prostredí terminálu, a následne obdobný typ útoku realizovať aj s **pomocou nástroja Ettercap**.

V oboch prípadoch taktiež sledujú vo Wiresharku zachytenú komunikáciu VMs, v ktorej sa zamerajú najmä na analýzu ARP správ. Pre overenie úspešnosti útoku je potrebné skontrolovať úspešnosť ARP *Cache Poisoning*-u („otravy“ ARP tabuliek) na zariadení klienta a na serveri, t. j. či naozaj došlo k podvrhnutiu ARP odpovedí s MAC adresou útočnickovho zariadenia.

Úspešnosť samotného ARP *spoofing* útoku sa overí **sledovaním komunikácie prechádzajúcej cez zariadenie útočníka vo Wiresharku**, ktorá by mala obsahovať pakety odosielané medzi klientom a serverom (napr. ping medzi týmito doma zariadeniami).

### 2.1. Riešenie samostatnej úlohy

V rámci samostatnej úlohy majú študenti za cieľ navrhnúť a implementovať ochranu voči ARP *spoofingu*, a to napr. vhodnou **konfiguráciou statických ARP záznamov**.

Účinnosť implementovanej ochrany je demonštrovaná neúspešnosťou ďalšieho *spoofingu*, nakoľko po nastavení statických ARP záznamov v prekladových tabuľkách na zariadení klienta a na serveri nie je možné docieľiť zápis podvrhutej MAC adresy útočníka do prekladovej ARP tabuľky, a teda ani presmerovanie komunikácie cez jeho zariadenie.

Pre kontrolu správnosti implementácie statických ARP záznamov je vhodné overiť výpis príkazu **arp -a** pre zobrazenie ARP tabuliek (klient a server), prípadne výstup nástroja **tcpdump** u útočníka, ktorý by v tomto prípade už nemal zachytiť žiadnu komunikáciu klienta so serverom.

## 2.2. Odpovede na kontrolné otázky

1. Akú funkciu plní ARP protokol v rámci sieťovej komunikácie?
  - A) Zabezpečuje preklad fyzickej adresy na logickú v lokálnej sieti
  - B) Priradzuje porty k IP adresám
  - C) Zisťuje fyzickú adresu zariadenia na základe jeho známej IP adresy ☒
  - D) Poskytuje kryptografickú ochranu komunikácie medzi dvoma zariadeniami
2. Ktoré z nasledujúcich tvrdení správne popisujú útok typu *ARP spoofing*?
  - A) Útočník odosiela do siete falošné ARP odpovede, aby dosiahol zmenu IP adresy v ARP tabuľke zariadenia
  - B) Jedná sa o typ útoku, pri ktorom útočník podvrhne svoju MAC adresu namiesto skutočnej MAC adresy zariadenia s hľadanou IP adresou v odpovedi na ARP žiadosť iného zariadenia ☒
  - C) Cieľom útoku je presmerovať sieťovú komunikáciu cez zariadenie útočníka ☒
  - D) *ARP spoofing* sa využíva primárne za účelom narušenia dostupnosti cieľovej služby
3. Aký je rozdiel medzi dynamickým a statickým ARP záznamom?
  - A) Dynamický záznam je uložený trvalo, statický len dočasne
  - B) Statický je uložený manuálne, dynamický sa generuje automaticky ☒
  - C) Dynamický záznam sa nikdy neaktualizuje podľa aktuálnej situácie v sieti
  - D) Dynamický je bezpečnejší ako statický
4. Prečo je pri MitM útoku dôležité zapnúť IP forwarding?
  - A) Aby bolo možné odosielať pakety cez zabezpečené HTTPS spojenie
  - B) Pretože umožní odosielanie a prijímanie ICMP správ
  - C) Aby útočník mohol presmerovať sieťovú komunikáciu cez svoje zariadenie ☒
  - D) Umožňuje zakázať použitie MAC filtering mechanizmu
5. Ktorý z nasledujúcich nástrojov slúži primárne na analýzu sieťovej komunikácie?
  - A) arpspoof
  - B) Ettercap
  - C) Wireshark ☒
  - D) arpinger
6. Ktoré z nasledujúcich javov môžu naznačovať prebiehajúci *ARP spoofing* v sieti?
  - A) Znížená latencia a zvýšená prenosová rýchlosť v sieti
  - B) Výskyt ARP odpovedí, ktoré priradzujú rovnakú MAC adresu k viacerým IP adresám ☒
  - C) Výskyt "duplicate IP" varovaní v systéme ☒
  - D) Výskyt viacerých ARP odpovedí bez predchádzajúcich požiadaviek ☒

7. Ktoré tvrdenia vystihujú rozdiely medzi nástrojmi arpspoof a Ettercap?
- A) Ettercap dokáže analyzovať a upravovať dáta vyšších vrstiev (napr. HTTP) ☒
  - B) arpspoof je jednoduchý CLI nástroj bez možnosti manipulácie so samotnými dátami ☒
  - C) Ettercap neumožňuje vizualizáciu MitM útokov cez GUI rozhranie
  - D) arpspoof automaticky obnovuje ARP tabuľky po útoku
8. K čomu slúži nástroj arpspoof počas útoku typu MitM?
- A) Odosiela falošné ARP odpovede, aby sa útočník dostal do pozície medzi dvoma zariadeniami (MitM) ☒
  - B) Skenuje sieť pre zistenie aktívnych služieb
  - C) Skenuje sieť pre zistenie pripojených koncových zariadení
  - D) Blokuje komunikáciu medzi routerom a klientom
9. Ktoré z nasledujúcich opatrení môžu pomôcť chrániť sieť pred ARP *spoofingom*?
- A) Použitie šifrovania TLS
  - B) Konfigurácia statických ARP záznamov ☒
  - C) Nasadenie *Dynamic ARP Inspection* (DAI) ☒
  - D) Použitie VLAN segmentácie ☒
10. Aký filter vo Wiresharku použijete na zobrazenie len ARP paketov (požiadaviek aj odpovedí)?
- A) arp ☒
  - B) ip.arp == 1
  - C) eth.type == 0x0806 ☒
  - D) arp.request

### 2.3. Dopĺňajúce otázky

Nižšie uvedené otázky môžu byť využité pri kontrole výstupov samostatnej práce študentom s cieľom overiť, či skutočne porozumeli riešenej problematike v praktickej časti laboratórnej úlohy.

#### 1. Popíšte, akú úlohu zohráva ARP protokol v komunikácii medzi zariadeniami v lokálnej sieti.

- ARP (*Address Resolution Protocol*) slúži na dynamické mapovanie logických IP adries na fyzické MAC adresy v rámci lokálnej siete, resp. na zistenie fyzickej (MAC) adresy zariadenia so známou IP adresou.

## 2. Čo je ARP spoofing a v čom tento útok spočíva?

- ARP *spoofing* je príkladom MitM (Man-in-the-Middle) útoku, pri ktorom útočník najskôr posiela do siete falošné ARP odpovede, s cieľom podvrhnúť MAC adresu svojho zariadenia pre prekladové záznamy prislúchajúce IP adrese dôveryhodného uzlu v sieti (napr. bráne alebo serveru), čo má za následok presmerovanie komunikácie práve cez zariadenie útočníka.

## 3. Aký je rozdiel medzi dynamickým a statickým ARP záznamom?

- Dynamické záznamy v ARP tabuľkách sú vytvárané a tiež priebežne aktualizované automaticky na základe komunikácie ARP protokolu, statické záznamy sú nastavené manuálne administrátorom, pomocou príkazu `arp -s <IP> <MAC>`. Pri použití statických záznamov nedochádza k ich automatickej zmene. Sú jedným zo spôsobov ochrany proti *spoofingu* (resp. proti podvrhnutiu falošnej MAC adresy), no vyžadujú ručnú správu.

## 4. Prečo je dôležité aktivovať IP forwarding pri MitM útoku?

- IP *forwarding* zabezpečuje, že sieťové pakety prijaté na jednom rozhraní útočnickovho zariadenia budú automaticky preposielané na druhé rozhranie smerom k cieľovému uzlu. V kontexte MitM útoku to znamená, že útočník dokáže nielen zachytávať, ale aj transparentne preposielať všetku komunikáciu medzi obeťami (napr. klientom a serverom), čím zostáva útok utajený a pritom funkčný bez akéhokoľvek prerušenia sieťového spojenia medzi legitímnymi zariadeniami.

## 5. Vysvetlite, akým spôsobom ste nastavili ciele pri použití nástroja Ettercap. Prečo sú jednotlivé kroky dôležité?

- Pri použití nástroja Ettercap je potrebné najprv vykonať sken siete (*Scan for hosts*) za účelom zistenia dostupných zariadení v danej sieti. Výsledkom bude zoznam, tzv. *Host List*, z ktorého sa následne vyberú cieľové IP adresy konkrétnych zariadení (napr. klient a server), ktoré majú byť cieľom útoku. Vybrané zariadenia sa označia ako *Target 1* a *Target 2*. Táto voľba je kľúčová pre správne nasmerovanie *spoofingu*, na konkrétne zariadenia

## 6. Aký filter Wiresharku použijete na zobrazenie len ARP paketov?

- Pre filtrovanie paketov ARP protokolu v prostredí nástroja Wireshark je nutné použiť filter `arp`, ktorý zobrazí výlučne len ARP požiadavky (*Request*) a odpovede (*Reply*).

## **Príloha C - Text laboratórnej úlohy č. 4**

Laboratórna úloha č. 4

# **BEZPEČNOSŤ SIEŤOVEJ VRSTVY**



# 0. Úvod k laboratórnej úlohe

Cieľom laboratórnej úlohy je zoznámiť sa s možnými hrozbami a analyzovať zraniteľnosti, ktoré ohrozujú sieťovú vrstvu počítačových sietí, predovšetkým s útokom *IP spoofing*, a tiež demonštrovať spôsoby kryptografického zabezpečenia dátových prenosov s využitím technológie IPsec.

V prvej časti laboratórnej úlohy pomocou vhodných nástrojov vo virtuálnom stroji Kali Linux najskôr realizujete **simuláciu sieťového útoku *IP spoofing*** s cieľom demonštrovať spôsob manipulácie s riadiacimi informáciami obsiahnutými v IP záhlaví prenášaného dátového paketu, a to konkrétne generovaním paketov s podvrhnutou zdrojovou IP adresou zariadenia, ktoré má byť cieľom tohto útoku. Okrem vykonania samotného útoku budete jeho priebeh monitorovať a následne analyzovať v prostredí sieťového analyzátoru Wireshark. Nakoniec sa budete venovať **implementácii bezpečnostného rozšírenia IPsec** za účelom zabezpečenia dátových prenosov na úrovni sieťovej vrstvy, a to najskôr v transportnom a neskôr i v tunelovom režime.

## Požiadavky pre vypracovanie úlohy:

- software: VMware Workstation Player pre virtualizáciu staníc,
- virtuálne stroje: tri virtuálne stroje s Kali Linux.

## 1. Teoretický úvod

V tejto laboratórnej úlohe budete oboznámení s problematikou týkajúcou sa možných bezpečnostných hrozieb na úrovni sieťovej vrstvy referenčného modelu ISO/OSI. Budete oboznámení so základným princípom **útoku *IP spoofing***, u ktorého si vyskúšate aj jeho praktickú simuláciu. V druhej časti počítačového cvičenia sa pokúsite za účelom ochrany prebiehajúcich dátových prenosov na úrovni sieťovej vrstvy implementovať **bezpečnostné rozšírenie IPsec**, ktoré prostredníctvom kryptografických mechanizmov zaisťuje dôvernosť a tiež autentickosť IP prenášaných paketov.

### 1.1. Hrozby na sieťovej vrstve: IP spoofing

**IP spoofing** je typ sieťového útoku, ktorý spočíva vo falšovaní (resp. podvrhnutí) zdrojovej IP adresy v IP záhlaví odosielaných paketov v snahe vzbudiť dojem, že tieto pakety sú odosielané z iného zariadenia než je zariadenie útočníka. Táto technika umožňuje útočníkovi obísť určité bezpečnostné opatrenia, akými môžu byť pravidlá firewallu pre filtrovanie komunikácie alebo mechanizmy autentifikácie, ktoré používajú ako identifikátor v procese overovania identity práve IP adresu. *IP spoofing* môže byť

útočníkom využitý napr. pri realizácii DDoS útokov<sup>29</sup>, kedy je spravidla cieľom útočníka dosiahnuť zahltenie cieľovej stanice (obete) v dôsledku generovania a odosielenia falošných požiadaviek s podvrhnutými IP adresami. [1]

Základný mechanizmus IP *spoofing* útoku spočíva odosielení takých IP paketov, v ktorých IP záhlaví útočník manuálne, cielene upraví informácie prenášané v poli zdrojovej IP adresy napr. na skutočnú IP adresu obete (t. j. konkrétneho zariadenia v sieti) alebo legitímneho servera. [1].

Vhodnou ochranou proti popísanému typu útoku môže byť použitie techník ako napr. *ingress filtering*<sup>30</sup>, ktoré umožnia blokovat' príjem prichádzajúcich paketov so zdrojovou IP adresou, ktorá nezodpovedá očakávanej adrese pre dané rozhranie, alebo **použitie bezpečnostného rozšírenia IPsec (*Internet Protocol Security*)** pre zabezpečenie komunikácie pomocou mechanizmov autentizácie a šifrovania prenášaných dát. Zabezpečeniu dátových prenosov pomocou IPsec rozšírenia bude venovaná ďalšia časť tejto laboratórnej úlohy.

## 1.2. Technológia IPsec

**IPsec (*Internet Protocol Security*)** predstavuje bezpečnostné rozšírenie sieťového IP protokolu. Jedná sa o sadu protokolov, ktoré spoločne poskytujú kombináciu bezpečnostných mechanizmov pre komplexné zabezpečenie dátových prenosov prebiehajúcich na úrovni sieťovej vrstvy počítačových sietí s využitím IP protokolu. IPsec je používaný v rôznych prostrediach, spravidla sa s jeho implementáciou môžeme stretnúť napr. pri zabezpečovaní virtuálnych privátnych sietí VPN.

IPsec umožňuje zaistiť **integritu a autentizáciu** prenášaných IP paketov a tiež ich **šifrovanie**, a to na úrovni IP protokolu, čím chráni dátové prenosy pred neoprávneným prístupom a manipuláciou.

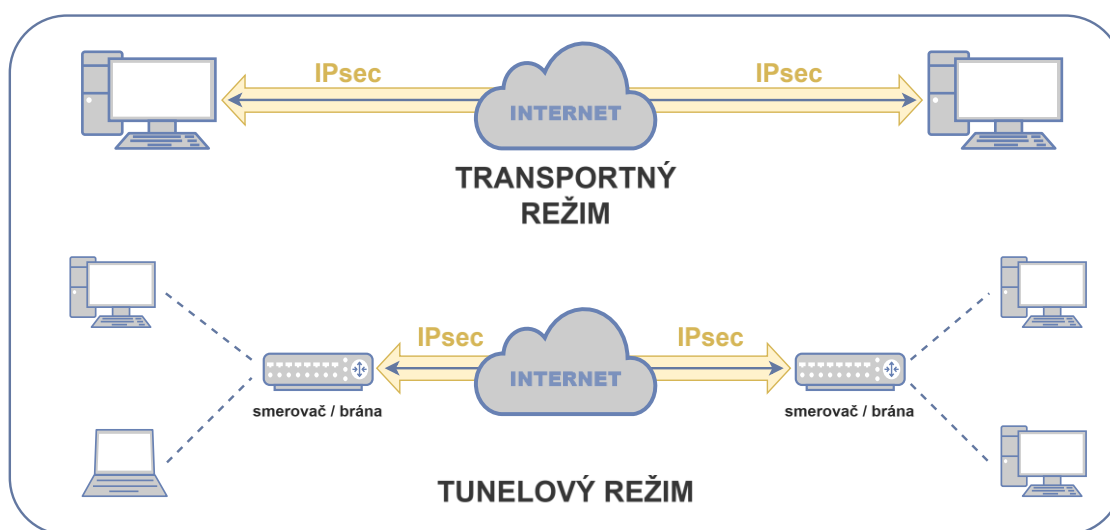
Protokol IPsec môže byť implementovaný pre fungovanie v dvoch základných módoch. Rozlišujeme:

- **transportný mód:** šifrovaný je iba obsah prenášaného IP paketu, pričom IP záhlavie paketu zostáva nezmenené, nie je šifrované. Transportný mód býva spravidla používaný pre zabezpečenie komunikácie medzi koncovými bodmi v sieti.
- **tunelový mód:** šifrovaný je celý pôvodný IP paket vrátane IP záhlavia, ktorý je následne vložený do nového IP paketu s novým IP záhlavím. Tento režim sa používa pri vytváraní tunelov medzi sieťami, napríklad medzi dvomi bránami.

---

<sup>29</sup> Typicky sa s IP *spoofingom* môžeme stretnúť pri realizácii SYN flood útoku, viac viď [2], [3], alebo pri útoku ICMP flood či Smurf útoku.

<sup>30</sup> Pre viac informácií o technike *Network Ingress Filtering* viď [5]111.



Obrázok 1.1 IPsec: transportný a tunelový režim<sup>31</sup>.

### Súčasťi protokolu IPsec

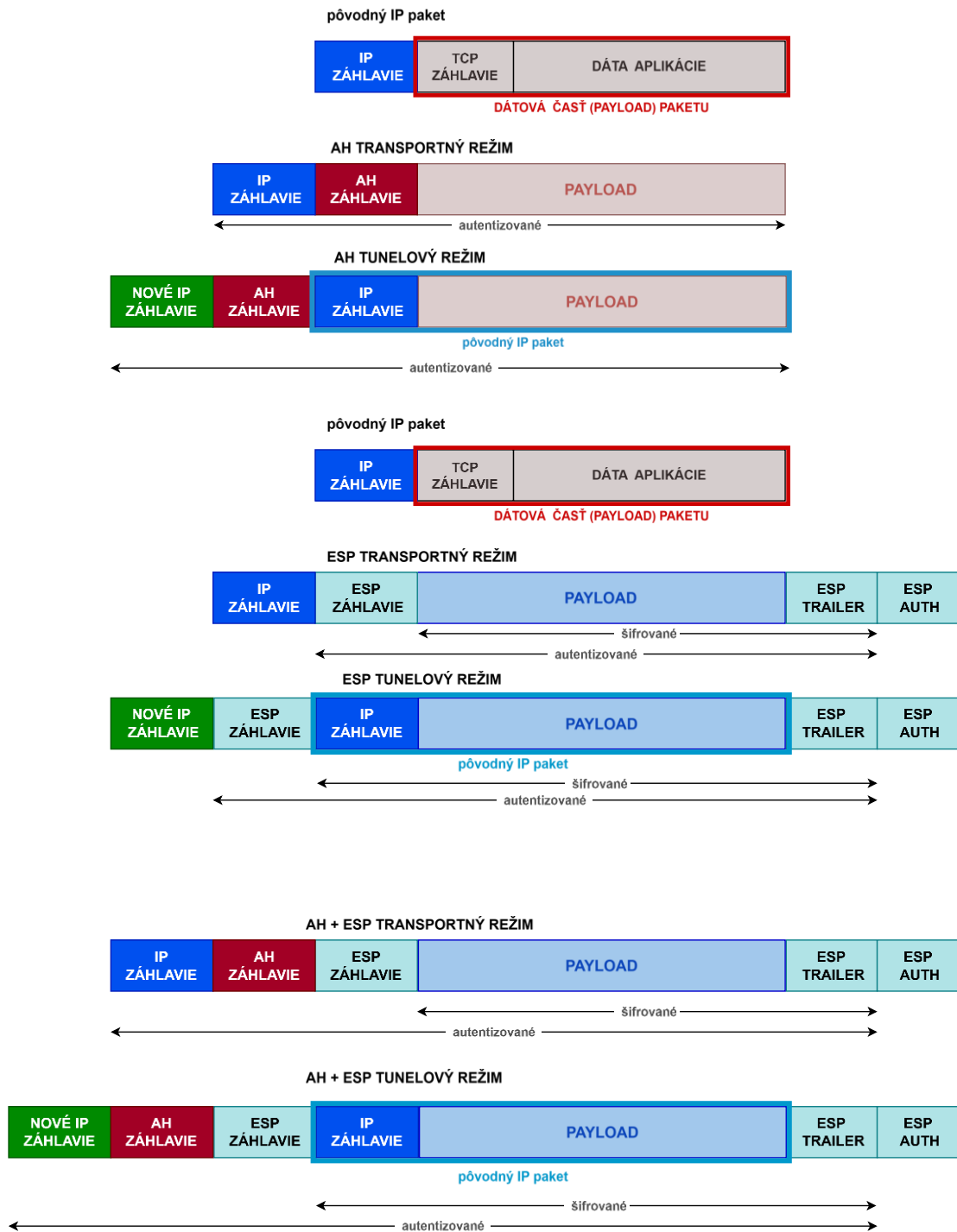
Bezpečnostné rozšírenie IPsec poskytuje možnosti komplexného zabezpečenia prenášaných dátových jednotiek prostredníctvom dvoch hlavných protokolov:

- **Authentication Header (AH):** používa sa za účelom overenia autenticity prenášaných IP paketov. Poskytuje ochranu prenášaných dát pred neoprávnenou zmenou, ale nezaist'uje ich šifrovanie, a teda ochranu informačného obsahu dát pred neoprávneným odposluchom.
- **Encapsulating Security Payload (ESP):** zaist'uje šifrovanie prenášaného dátového obsahu (resp. celého IP paketu v tunelovom režime) a súčasne zaist'uje i overenie autenticity obsahu IP paketu. Autenticita IP záhlavia je zaistená len v prípade použitia ESP súčasne s AH protokolom.

Zapuzdrenie prenášaného IP paketu, ktoré spočíva v pridaní odpovedajúcich záhlaví a zápätí nesúcich riadiace informácie, pri použití AH a/alebo ESP protokolu pre zabezpečenie ním prenášaného dátového obsahu, je znázornené pre oba prenosové IPsec režimy (transportný o tunelový) na obr. 1.2.

Podrobnejší popis technológie IPsec možno nájsť napr. v [4].

<sup>31</sup> Prevzaté z [3].



Obrázok 1.2 Schematické znázornenie zapuzdrovania IPsec paketu.

### 1.3. Kali Linux a použité nástroje

V tejto laboratórnej úlohe budú pre útok IP *spoofing*, záznam a analýzu prebiehajúcej dátovej komunikácie a neskôr pre implementáciu bezpečnostného rozšírenia IPsec postupne využité nižšie uvedené nástroje:

- **hping3**: pokročilý nástroj určený ku generovaniu a odosielaniu vlastných IP paketov, ktorého použitie je vhodné napr. práve pre účely simulácie IP spoofing útoku.
- **Wireshark**: sieťový analyzátor určený pre sledovanie a záznam prebiehajúcej sieťovej komunikácie (resp. jednotlivých paketov) s možnosťou následnej analýzy zachytených paketov.
- **strongSwan**: *open-source* knižnica vhodná pre implementáciu bezpečnostného rozšírenia IPsec, používaná tiež pre konfiguráciu bezpečných VPN spojení.

#### Nástroj hping3

**hping3** je flexibilný nástroj vhodný pre použitie ku generovaniu TCP/IP paketov. Umožňuje simuláciu rôznych typov útokov, ako sú napríklad IP spoofing, *port scanning* či rôzne typy DoS útokov, môže byť použitý tiež za účelom testovania firewallov. Pomocou nástroja **hping3** je možné vytvárať vlastné pakety s upraveným IP záhlavím a sledovať, ako cieľové systémy reagujú na neštandardné sieťové pakety (napríklad či odpovedajú na požiadavky, blokujú komunikáciu alebo generujú chyby apod.).

Pri práci s nástrojom **hping3** je možné využiť „pomocníka“ k zobrazeniu dostupných príkazov podporujúcich rozličné funkcie tohto nástroja, a to pomocou príkazu:

```
hping3 --help
```

Nástroj **hping3** poskytuje tiež možnosť pre overenie dostupnosti cieľovej služby (napr. webového servera). Pomocou nižšie uvedeného príkazu je možné overiť, či je testovaná služba aktuálne dostupná a či firewall umožňuje, resp. neblokuje prístup k uvedenému portu:

```
sudo hping3 -S 192.168.126.130 -p 80 -c 3
```

kde prepínač **-S** zaistí odoslanie postupne **troch** TCP/IP paketov s nastaveným príznakom **SYN = 1** na **port 80** cieľového zariadenia s uvedenou **IP adresou**.

Pre účely simulácie IP *spoofingu* je možné **hping3** použiť nasledovne:

```
sudo hping3 -a 192.168.126.129 -S 192.168.126.130 -p 80 -c 5
```

Uvedený príkaz vygeneruje celkom **päť** TCP SYN paketov smerovaných na **port 80** cieľovej IP adresy **192.168.126.130**, pričom do záhlavia týchto paketov bude vložená (pomocou prepínača **-a**) falošná zdrojová IP adresa s hodnotou **192.168.126.129**.

Vysvetlenie použitých prepínačov príkazu a ich parametrov:

- **sudo** spustenie nástroja s oprávneniami správcu,
- **hping3** spustenie príslušného nástroja,
- **-a 192.168.126.129** nastavenie falošnej (*spoofovanej*) zdrojovej IP adresy,
- **-S** nastavenie TCP príznaku SYN<sup>32</sup> na hodnotu 1,
- **-p 80** cieľový port, typicky používaný pre HTTP,
- **-c 5** počet odoslaných paketov (v tomto prípade 5).

## Wireshark

Wireshark je pokročilý sieťový analyzátor, ktorý umožňuje zachytávať, vizualizovať a analyzovať sieťovú prevádzku v reálnom čase. Je vhodný pre analýzu komunikačných protokolov a obsahu prenášaných dátových jednotiek, detekciu podozrivých paketov a identifikáciu sieťových problémov. Tento nástroj ponúka možnosť filtrovať zachytenú komunikáciu na základe rôznych kritérií (napr. zdrojová/destinovaná IP, protokol, port).

S nástrojom Wireshark ste sa oboznámili už v predošlom cvičení, v rámci ktorého ste si tiež vyskúšali aj jeho praktické použitie. Pre potreby praktickej časti tejto úlohy je nižšie v tabuľke 1.1 uvedený prehľad niekoľkých možných filtrov pre selektívne zobrazenie vhodných paketov zo zaznamenatej komunikácie.

Tabuľka 1.1 Wireshark: príklady použitých filtrov komunikácie.

| účel                                                           | filter                                                                |
|----------------------------------------------------------------|-----------------------------------------------------------------------|
| <b>zobrazenie všetkých paketov ICMP protokolu:</b>             | <b>icmp</b>                                                           |
| <b>sledovanie dátovej komunikácie medzi dvoma IP adresami:</b> | <b>ip.src == 192.168.126.129 &amp;&amp; ip.dst == 192.168.126.130</b> |
| <b>zobrazenie všetkých ESP<sup>33</sup> paketov:</b>           | <b>esp</b>                                                            |

<sup>32</sup> Príznakový bit SYN = 1 v záhlaví TCP protokolu indikuje zahájenie, resp. vytvorenie TCP spojenia medzi koncovými bodmi komunikácie.

<sup>33</sup> ESP = *Encapsulating Security Payload* – súčasť rozšírenia IPsec, protokol pre šifrovanie komunikácie.

## Strongswan

strongSwan je *open-source* implementácia protokolov IPsec a IKE (*Internet Key Exchange*)<sup>34</sup>, ktorá umožňuje vytvoriť bezpečné prepojenie dvoch alebo viacerých uzlov, resp. zariadení, na sieťovej vrstve. Podporuje IPv4 aj IPv6, transportný aj tunelový režim IPsec, a v rámci protokolu IKE podporuje statickú i dynamickú výmenu kľúčov.

Pre konfiguráciu bezpečnostného rozšírenia IPsec k zabezpečeniu sieťovej komunikácie je kľúčová práca s nižšie uvedenými konfiguračnými súborami:

- **etc/ipsec.conf** – hlavný konfiguračný súbor pre definíciu spojení, (nastavenie napr. IP adries, režimu prenosu, spôsobu autentifikácie)
- **etc/ipsec.secrets** – konfiguračný súbor obsahujúci zdieľané tajomstvá alebo kľúče potrebné pre autentifikáciu komunikujúcich strán.

Pre prácu s knižnicou strongSwan je nutné spustiť samotnú službu pomocou príkazu:

```
sudo systemctl start strongswan
```

Knižnica strongSwan umožňuje vytváranie IPsec spojení, resp. tunelov medzi dvoma koncovými bodmi komunikácie. Spustenie, resp. manuálne vytvorenie zabezpečeného spojenia je možné príkazom:

```
sudo ipsec up <názov_spojenia>
```

Taktiež je možné zobrazit' stav implementovaných IPsec tunelov vytvorených medzi zariadeniami:

```
sudo ipsec statusall
```

---

<sup>34</sup> IKE (*Internet Key Exchange*) je protokol používaný v rámci IPsec spojenia pre bezpečnú výmenu kryptografických kľúčov a vyjednanie bezpečnostných parametrov medzi dvoma komunikujúcimi stranami. Jeho cieľom je vytvoriť a spravovať tzv. Security Associations (SA), ktoré definujú, aké kryptografické mechanizmy budú použité k zaisteniu dôvernosti a integrity prenášaných dát. V rámci knižnice strongSwan predstavuje IKE nevyhnutnú súčasť, ktorá zaisťuje bezpečné nadviazanie IPsec spojenia a následné obnovenie a/alebo ukončenie vytvoreného tunela.

## 2. Praktická časť

V rámci praktickej časti počítačového cvičenia bude vytvorená **simulácia útoku IP Spoofing** pomocou nástroja **hping3** s následnou analýzou komunikácie cez **Wireshark**. Simulovaný útok bude prebiehať vo virtuálnej sieti pozostávajúcej z troch virtuálnych strojov s Kali Linux (klient, server a útočník), ktorej topológia je schematicky znázornená nižšie na obr. 2.1. Komunikácia medzi uvedenými virtuálnymi strojmi prebieha skrz virtuálny prepínač VMware virtual switch, ako je znázornené na uvedenom obrázku.

Cieľom IP *spoofingu* môže byť v dôsledku generovania IP paketov s podvrhnutou zdrojovou IP adresou vyvolanie nadväzujúceho DDoS útoku s cieľom zapríčiniť nemožnosť plnohodnotného fungovania zariadenia, ktoré je cieľom (tzv. obeťou) zamýšľaného útoku. Následne si v nadväzujúcej ďalšej časti za účelom zabezpečenia prebiehajúcich dátových prenosov na úrovni sieťovej vrstvy vyskúšate **konfiguráciu bezpečnostného rozšírenia IPsec**, ktoré zaistuje v rámci komunikácie medzi dvoma zariadeniami dôvernosť prenášaných IP paketov prostredníctvom šifrovania a tiež ich autentickosť a odolnosť voči narušeniu integrity.

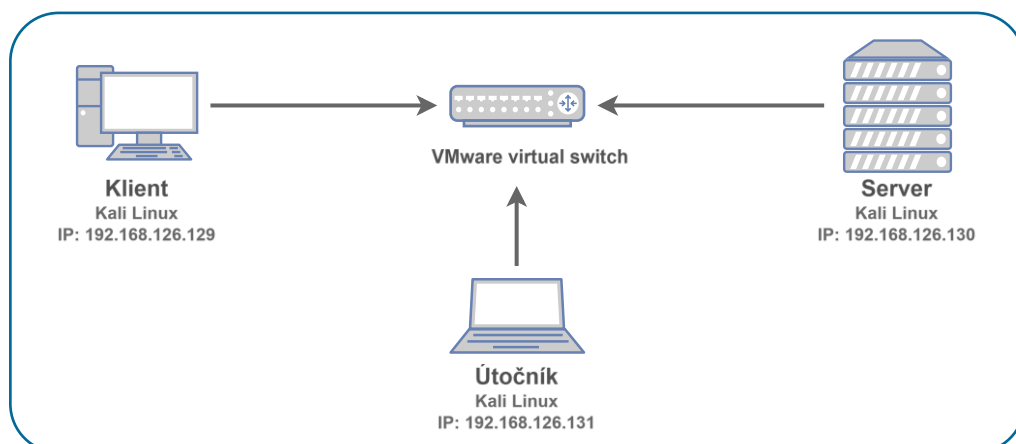
### 2.1. Topológia virtuálnej siete a nastavenie virtuálnych strojov

Použité virtuálne stroje:

- **obet' (klient):** bežný počítač v sieti, cieľ útoku.
- **server:** poskytovateľ sieťovej služby (napr. webserver, gateway).
- **útočník:** vykonáva IP *spoofing*, generuje IP pakety s falošnou zdrojovou IP adresou.

Sieťová konfigurácia:

- obet': 192.168.126.129
- server: 192.168.126.130
- útočník: 192.168.126.131



Obrázok 2.1 Topológia siete laboratórnej úlohy.



Všetky virtuálne stroje musia byť prepojené v rovnakej virtuálnej podsieti pomocou sieťového režimu **Host-only** alebo **NAT**, aby mohlo byť na VM predstavujúcim „útočníka“ realizované zachytávanie dátových prenosov (komunikácie) medzi klientom a serverom.

## 2.2. Zoznámenie sa s použitými nástrojmi

Prehľad základných príkazov pre jednotlivé používané nástroje

- **hping3**: spustenie simulácie IP spoofingu:

```
sudo hping3 -a <zdroj_IP> -S <ciel_IP> -p <port> -c <pakety>
```

- **Wireshark**: za účelom prehľadnejšej analýzy zachytenej dátovej komunikácie je doporučené využívanie vhodných filtrov pre zobrazenie vybraných paketov, napr. na základe konkrétnej zdrojovej a/alebo cieľovej IP adresy:

```
ip.src == 192.168.126.129 && ip.dst == 192.168.126.130
```

- **strongSwan**: pre konfiguráciu zabezpečeného IPsec pripojenia medzi klientom a serverom pomocou knižnice Strongswan bude nutná vhodná modifikácia základných konfiguračných súborov, a to konkrétne:
  - **/etc/ipsec.conf** (hlavný konfiguračný súbor IPsec)
  - **/etc/ipsec.secrets** (súbor obsahujúci autentifikačné kľúče)

## 2.3. Postup pre vypracovanie laboratórnej úlohy


### A) Príprava prostredia

#### Spustenie virtuálnych strojov:

- Otvorte VMware Workstation Pro (umiestnený na ploche).
- Spustite postupne všetky tri virtuálne stroje (klient, server, útočník).
- Uistite sa v správnosti konfigurácie sieťových parametrov, overte priradenie IP adries zariadeniam.
- Prihláste sa do prostredia Kali Linux na VM útočníka.

|              |                                                                |
|--------------|----------------------------------------------------------------|
| VM „útočník“ | – prihlasovacie údaje: <b>Username: kali, Password: kali</b>   |
| VM „klient“  | – prihlasovacie údaje: <b>Username: klient, Password: kali</b> |
| VM „server“  | – prihlasovacie údaje: <b>Username: server, Password: kali</b> |

### Overenie sieťovej konektivity:

- Otvorte **terminál** (kliknite na ikonu terminálu  v záhlaví horného pracovného panelu alebo stlačte Ctrl + Alt + T).
- Na jednotlivých VMs si zobrazte priradené IP adresy (na rozhraní **eth0**):

```
ip a
```

- Z klienta odošlite testovaciu požiadavku (ping) a vyskúšajte pripojenie na server pomocou príkazu:


```
ping <ip_adresa_servera>
```

- Ak klientovi prichádza odpoveď **ping echo reply** zo strany servera, sieťová komunikácia medzi zariadeniami funguje.
- Obdobným spôsobom overte možnosť spojenia v opačnom smere komunikácie.

### B) IP spoofing útok pomocou hping3

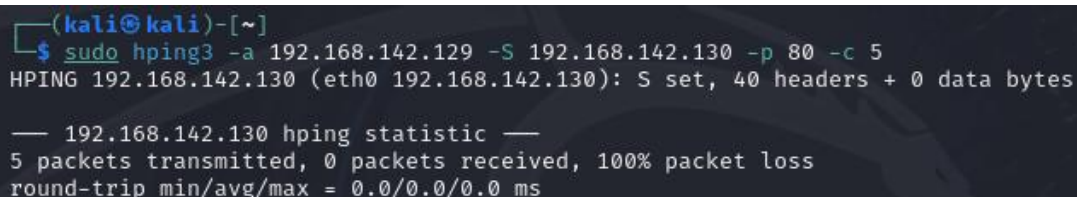
Pomocou nástroja hping3 na VM útočníka je možné simulovať IP *spoofing* útok generovaním IP paketov s podvrhnutou zdrojovou IP adresou. Cieľom je odosielať pakety vzbudzujúce dojem, akoby boli odoslané samotným klientom. Týmto útokom je možné overiť, že cieľové zariadenie (v tomto prípade server) nedokáže rozpoznať skutočného odosielateľa (= útočníka). Prebiehajúca komunikácia vo vytvorenej sieti bude monitorovaná pomocou nástroja Wireshark.

### Použitie nástroja hping3

- Na stroji útočníka otvorte terminál kliknutím na ikonu  umiestnenú v záhlaví hlavného pracovného okna alebo v menu zvolíte **Applications > System Tools > Terminal**. Otvorí sa okno s príkazovým riadkom.
- Pre spustenie IP *spoofingu* zadajte príkaz:

```
sudo hping3 -a 192.168.126.129 -S 192.168.126.130 -p 80 -c 5
```

Po zadaní príkazu bude zahájená simulácia útoku, resp. generovanie dátových paketov, ktoré sa budú javiť, ako keby boli odosielané zo zariadenia klienta.



```
(kali@kali)-[~]  
$ sudo hping3 -a 192.168.142.129 -S 192.168.142.130 -p 80 -c 5  
HPING 192.168.142.130 (eth0 192.168.142.130): S set, 40 headers + 0 data bytes  
  
— 192.168.142.130 hping statistic —  
5 packets transmitted, 0 packets received, 100% packet loss  
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Obrázok 2.2 Spustenie IP spoofing útoku v termináli Kali Linux.

## Sledovanie prichádzajúcej komunikácie (server) vo Wiresharku

- Na serveri spustíte **Wireshark** pomocou príkazu:

```
sudo wireshark &
```

Prepínač '**sudo**' spustí nástroj Wireshark s oprávneniami administrátora, čo je nevyhnutné pre zachytávanie sieťovej prevádzky v Kali Linux.

- V hlavnom okne vyberte sieťové rozhranie **eth0** – dvojitém kliknutím spustíte zachytávanie (alebo kliknite na **Start Capturing Packets** v záhlaví panelu s nástrojmi).
- Do okna pre filtrovanie komunikácie zadajte:

```
ip.src == 192.168.126.129 && ip.dst == 192.168.126.130
```

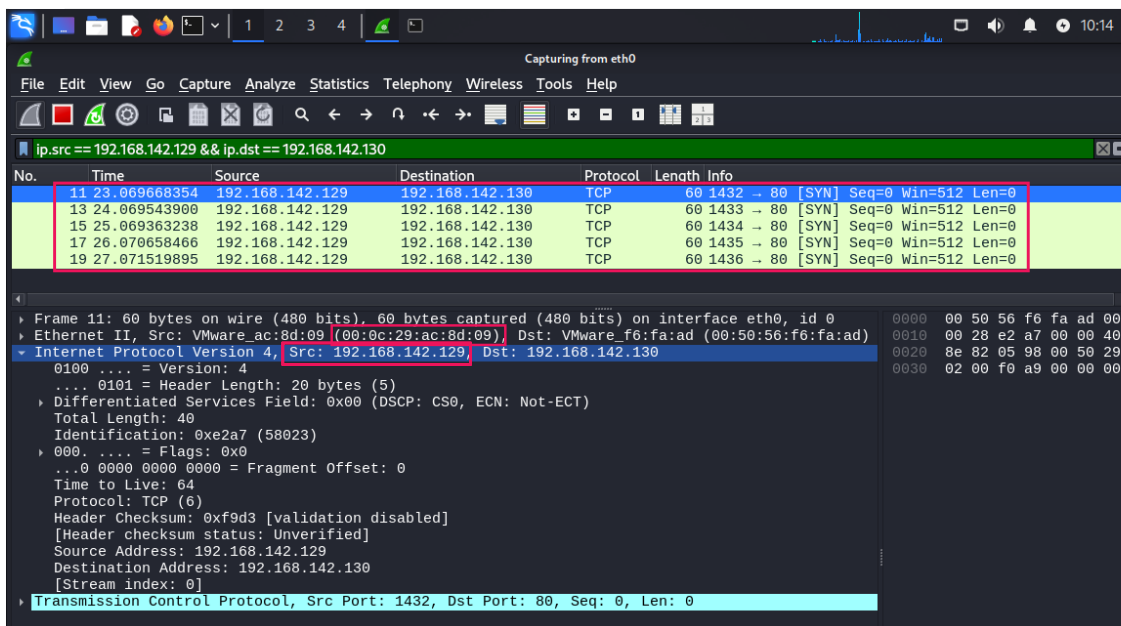
túto voľbu potvrdíte stlačením **Enter**.

Použitie uvedeného filtra zaistí, že spomedzi všetkých zachytených dátových jednotiek prenesených v rámci komunikácie cez zvolené rozhranie budú zobrazené len pakety údajne „pochádzajúce od klienta“, t. j. pakety vyhovujúce filteru **ip.src == 192.168.126.129** a ďalej tiež pakety, ktoré majú byť doručené na server t. j. pakety vyhovujúce druhému zo zadaných filtrov **ip.dst == 192.168.126.130**. V skutočnosti sa však jedná o pakety generované na strane útočníka nástrojom **hping3**.

- Kliknutím pravým tlačidlom myši na niektorý zo zobrazených paketov a následným výberom možnosti „**Follow > TCP Stream**“ je možné zobrazit' celkový prehľadný priebeh spojenia medzi komunikujúcimi zariadeniami.
- Pre podrobnejšiu analýzu komunikácie kliknite na vybraný paket a v dolnej časti sledovaného okna Wiresharku si zobrazte sekciu **Internet Protocol Version 4**, kde si môžete bližšie zobrazit' konkrétne hodnoty *Source IP* (zdrojová) a *Destination IP* (cieľová adresa).

Zdrojová IP adresa by mala mať hodnotu 192.168.126.129 odpovedajúcu VM klienta, hoci bol príslušný paket v skutočnosti odoslaný zo stanice útočníka (viď zachytená komunikácia na obr. 2.3).

- Pre overenie úspešnosti IP spoofingu ďalej analyzujte fyzické MAC adresy zariadení, z ktorých boli jednotlivé pakety odoslané (zobrazenie v sekcii **Ethernet II**).



Obrázok 2.3 Ukážka zachytenej komunikácie po spustení útoku IP spoofing (server).

### C) Zabezpečenie komunikácie s využitím IPsec

V nasledujúcej časti laboratórnej úlohy si vyskúšate prácu s knižnicou **strongSwan** podporujúcu implementáciu IPsec-u.

#### Konfigurácia IPsec v transportnom režime

- Na zariadení klienta a na serveri spustíte inštaláciu knižnice **strongSwan**:

```
sudo apt update && sudo apt install strongswan -y
```

Použitie uvedeného príkazu zabezpečí inštaláciu balíka **strongSwan**, ktorý obsahuje potrebné komponenty pre vytvorenie zabezpečeného IPsec spojenia. Aktualizácia balíčkov v Kali Linux (príkaz: **apt update**) zabezpečí, že sa použijú aktuálne dostupné verzie.

- Na oboch zariadeniach (klient a server) upravte konfiguračný súbor **/etc/ipsec.conf** – pre editáciu súborov v textovom režime môžete využiť napr. editor **nano**, a to nasledovne:

- pre presun do adresára, kde sa príslušný konfiguračný súbor nachádza, môžete v otvorenom terminálovom okne použiť príkaz:

```
cd /etc
```

- následne otvorte súbor na úpravu:

```
sudo nano ipsec.conf
```

- do súboru vložte nasledujúcu konfiguráciu:

```
conn test
    left=192.168.126.129          # klient (aktuálny VM)
    right=192.168.126.130        # server (protistrana)
    authby=secret
    auto=start
    ike=aes256-sha1-modp1024
    esp=aes256-sha1
    keyexchange=ikev2
```

**!! Uvedená konfigurácia sa týka nastavení IPsec na strane klienta.**

Týmto nastavením sa definujú parametre zabezpečeného spojenia medzi vašim klientom (**left**) a serverom (**right**). Parameter **authby=secret** určuje, že sa bude pre autentizáciu komunikujúcich strán používať zdieľané tajomstvo PSK (*pre-shared key*). Ďalej nastavujeme výmenu kľúčov pomocou IKEv2, a špecifikujeme šifrovacie a autentifikačné algoritmy pre fázu IKE (**ike**) aj samotné dáta (**esp**). Parameter **auto=start** zabezpečí automatické nadviazanie definovaného IPsec spojenia pri štarte služby.

- Po úprave konfigurácie stlačte kombináciu kláves **Ctrl + O** (uloženie), potvrdte názov súboru klávesou **Enter** a následne ukončíte textový editor **nano** pomocou kombinácie **Ctrl + X**.
- Alternatívne môžete úpravu konfigurácie ukončiť stlačením **Ctrl + X**, následne pre potvrdenie uloženia vykonaných zmien stlačte **Y** (ako Yes/áno), a nakoniec potvrdte stlačením **Enter**. Týmto spôsobom sa zmeny uložia a editor sa zároveň zatvorí.
- Ďalej upravte konfiguračný súbor **/etc/ipsec.secrets**:

```
sudo nano ipsec.secrets
```

- do ktorého vložíte nasledovné:

```
192.168.126.129 192.168.126.130 : PSK "tajneheslo"
```

Tento riadok definuje spoločný zdieľaný kľúč (PSK), ktorý bude použitý na autentizáciu medzi dvoma zariadeniami s uvedenými IP adresami (medzi klientom a serverom). **Je dôležité zadať na oboch stranách komunikácie identický kľúč**, inak proces autentizácie neprebehne úspešne a zabezpečené

spojenie nebude možné nadviazať. PSK predstavuje jednoduchý spôsob autentifikácie, ktorý je vhodný pre menšie a testovacie prostredia, no menej bezpečný pre rozsiahle produkčné nasadenie.

- Po úprave súboru opäť použite kombináciu **Ctrl + O** → **Enter** → **Ctrl + X** alebo alternatívne **Ctrl + X** → **Y** → **Enter** pre potvrdenie a uloženie vykonaných zmien.
- Po zmene v konfigurácii je potrebné službu reštartovať, príp. môžete overiť, či je služba po spustení aktívna:

```
sudo systemctl restart strongswan-starter  
sudo systemctl status strongswan-starter
```

- Obdobne postupujte pri inštalácii strongSwan a konfigurácii zabezpečeného IPsec spojenia aj na strane serveru, pričom dbajte na správne definovanie jednotlivých parametrov (IP adres) pri úprave konfigurácie.

### Pripojenie a overenie konfigurácie IPsec

- Na oboch zariadeniach (klient i server) spustite IPsec a následne vytvorte medzi nimi zabezpečené spojenie na základe predchádzajúcej konfigurácie:

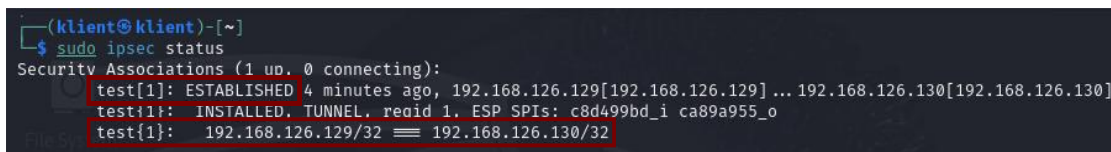
```
sudo ipsec start
```

```
sudo ipsec up test
```

- Zobrazte si stav vytvoreného spojenia pomocou príkazu:

```
sudo ipsec status
```

V prípade správnej konfigurácie, by malo byť spojenie v stave **ESTABLISHED** (viď obr. 2.4). Pre zobrazenie podrobnejších informácií o spojení je možné alternatívne použiť príkaz `ipsec statusall`.



```
(klient@klient)-[~]  
$ sudo ipsec status  
Security Associations (1 up, 0 connecting):  
test[1]: ESTABLISHED 4 minutes ago, 192.168.126.129[192.168.126.129] ... 192.168.126.130[192.168.126.130]  
test[1]: INSTALLED, TUNNEL, reqid 1, ESP SPIs: c8d499bd_i ca89a955_o  
test[1]: 192.168.126.129/32 == 192.168.126.130/32
```

Obrázok 2.4 Overenie vytvorenia IPsec spojenia (klient).

- Dôležité je overiť tiež to, či komunikácia medzi zariadeniami klient ↔ server skutočne prebieha cez vytvorený IPsec tunel.
- Na serveri opäť otvorte nástroj Wireshark a spustite zachytávanie sieťovej prevádzky na rozhraní `eth0`.

- V termináli (server) s pomocou nástroja `tcpdump` sledujte komunikáciu na strane serveru, ktorá bude vykazovať známky používania IPsec tunela:

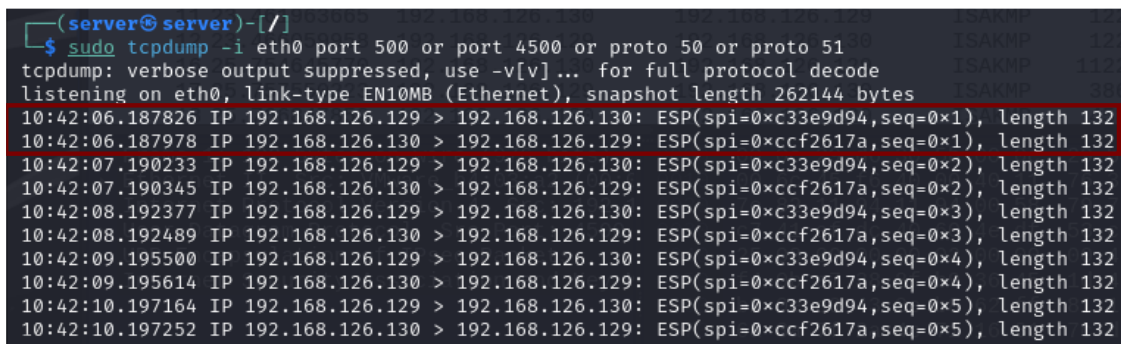
```
sudo tcpdump -i eth0 port 500 or port 4500 or proto 50 or proto 51
```

Použitie `tcpdump` s uvedenými parametrami zaistí sledovanie prevádzky súvisiacej s IPsec: **UDP port 500 (IKE), protokol 50 (ESP), protokol 51 (AH)**. Pre prípad použitia prekladu adres je vhodné overiť aj komunikáciu cez UDP port 4500 (NAT-T).

- Z klientskeho VM odošlite `ping` na server:

```
ping 192.168.126.130
```

- Na serveri sledujte jednak výstup nástroja `tcpdump` v otvorenom okne terminálu, a súčasne i záznam komunikácie vo Wiresharku.
- Nižšie uvedený výpis dokazuje, že vytvorený **šifrovaný IPsec tunel funguje** a teda že medzi klientom a serverom prebieha **výmena šifrovaných ESP paketov** (protokol 50) práve cez tento tunel – podrobnejšie viď obr. 2.5.



Obrázok 2.5 Výpis nástroja `tcpdump`: sledovanie komunikácie prostredníctvom vytvoreného IPsec tunela (server).

- Pre zobrazenie šifrovanej komunikácie vo Wiresharku je vhodné použiť filter:

```
esp || udp.port == 500 || udp.port == 4500
```

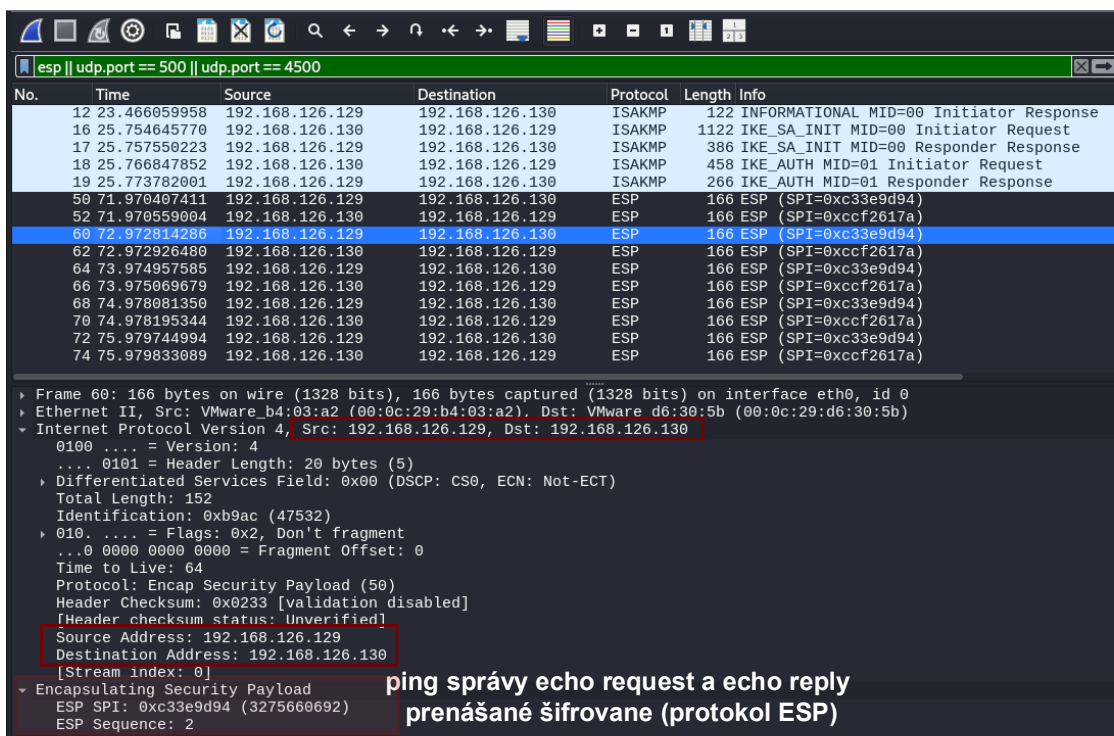
Použitie uvedeného filtra zobrazí:

- **ESP pakety** (protokol 50),
- **IKE komunikáciu** (na porte UDP 500 – vytvorenie IPsec tunela),
- **IKE cez NAT-T** (UDP port 4500 – ak sa IPsec prispôsobuje NAT-u).



- **Ďalšia možnosť overenia:**

Pomocou nástroja hping3 opäť spustíte na VM útočníka simuláciu útoku IP Spoofing a sledujte, či sú odosielané pakety od útočníka na strane serveru odmietnuté. Správne nakonfigurované spojenie by malo zabrániť prijatiu neautentifikovanej komunikácie. Pribeh komunikácie sledujte tiež pomocou nástroja Wireshark.



Obrázok 2.6 Ukážka zachytenej komunikácie – prenos cez šifrovaný IPsec tunel (server).



## 2.4. Samostatná úloha

### D) Implementácia IPsec v tunelovom režime

V poslednej časti úlohy si samostatne vyskúšate praktické nasadenie bezpečnostného rozšírenia **IPsec v tunelovom režime** pre zabezpečenie komunikácie odosielanej cez simulovanú „verejnú sieť“.

**Cieľom vašej samostatnej práce bude najskôr simulovať vlastnú verejnú sieť a následne správnym spôsobom nakonfigurovať IPsec rozšírenie v tunelovom režime pre zabezpečenie komunikácie medzi klientom a serverom.**

**Po implementácii otestujete a porovnáte výkon siete pri použití transportného a tunelového IPsec režimu, a to najmä z pohľadu latencie komunikácie, spoľahlivosti prenosu a stability vytvoreného spojenia.**

*Zachytenú komunikáciu analyzujte vo Wiresharku a porovnajte rozdiely medzi oboma režimami. Pri analýze venujte pozornosť aj štruktúre paketov (napr. viditeľnosť vnútorných záhlaví, šifrovanie obsahu a použité protokoly).*

### 3. Záver

V tejto laboratórnej úlohe ste sa zoznámili s problematikou bezpečnosti sieťovej vrstvy počítačových sietí a v tejto súvislosti tiež s rizikami spojenými s podvrhnutím IP adresy (tzv. IP spoofing), čo predstavuje častý spôsob narušenia integrity alebo dôvernosti komunikácie na sieťovej vrstve.

V praktickej časti ste **pomocou nástroja hping3** realizovali simuláciu útoku, ktorý ilustruje, akým spôsobom je možné oklamať cieľové zariadenie použitím falošnej zdrojovej IP adresy uvedenej v záhlaví odosielaných dátových jednotiek. Následne ste si vyskúšali **praktickú implementáciu rozšírenia IPsec**, ktorý umožňuje komplexné zabezpečenie IP komunikácie, a to vrátane šifrovania dát, autentizácie a zaistenia integrity prenosu. Porovnaním transportného a tunelového IPsec režimu ste získali prehľad o rôznych spôsoboch ochrany dátovej prevádzky a taktiež o ich vplyve na výslednú podobu paketov či celkový výkon počítačovej siete.

#### 3.1. Kontrolné otázky

##### 1. Čo je cieľom IP *spoofing* útoku?

- A) Zmeniť MAC adresu útočníka
- B) Získať neautorizovaný prístup predstieraním cudzej IP adresy
- C) Presmerovať legítimnú komunikáciu cez vlastné zariadenie
- D) Zamedziť šifrovaniu dát medzi serverom a klientom

##### 2. Ktoré z nasledujúcich tvrdení platia o nástroji hping3?

- A) Umožňuje simulovať IP *spoofing* a rôzne typy sieťových útokov
- B) Je určený na šifrovanie komunikácie pomocou IPsec
- C) Dokáže vygenerovať vlastné TCP/IP pakety podľa špecifikácie
- D) Je to nástroj na konfiguráciu VPN tunelov medzi vzdialenými sieťami

##### 3. Vyberte nesprávne tvrdenia týkajúce sa IP *spoofing* útoku:

- A) IP spoofing automaticky zahŕňa zmenu MAC adresy
- B) Má za následok zvýšenie prenosovej rýchlosti v sieti
- C) Spoofovaný paket má zvyčajne neplatný kontrolný súčet
- D) Využíva manipuláciu s IP záhlavím paketov

##### 4. Aký je hlavný rozdiel medzi transportným a tunelovým režimom IPsec?

- A) V tunelovom režime sa šifruje len záhlavie IP paketu
- B) Transportný režim sa používa v bezdrôtových sieťach
- C) V transportnom režime sú šifrované len užívateľské dáta, IP záhlavie paketu ostáva nezmenené
- D) Tunelový režim nemôže byť využitý v IPv6 sieti

**5. Čo spôsobí nastavenie parametra `authby=secret` v súbore `ipsec.conf`?**

- A) Povolenie anonymného prístupu
- B) Vypnutie autentizácie
- C) Autentizáciu pomocou predzdieľaného tajomstva (PSK)
- D) Použitie certifikátov

**6. Protokol AH (*Authentication Header*) v IPsec:**

- A) Umožňuje zašifrovať celý IP paket
- B) Zaisťuje autentizáciu a integritu paketu bez šifrovania
- C) Poskytuje možnosť tunelovania prenosu cez HTTPS
- D) Zaisťuje dôvernosť riadiacich informácií v IP záhlaví

**7. Vyberte nesprávne tvrdenia o tunelovom režime IPsec:**

- A) Zabezpečuje celý IP paket vrátane pôvodného záhlavia
- B) Nie je vhodný pre spojenie medzi dvoma bránami
- C) Používa sa najmä pri zabezpečení VPN
- D) Prenáša pakety cez šifrovaný SSH tunel

**8. V akých situáciách je vhodné použiť IPsec v transportnom režime?**

- A) Komunikácia medzi klientom a serverom v rovnakej sieti
- B) Prepojenie dvoch vzdialených sietí cez internet
- C) Na zabezpečenie SSH spojenia
- D) Ochrana komunikácie medzi aplikáciami v rámci jedného servera

**9. Ako môže použitie IPsec ovplyvniť výkonnosť počítačovej siete?**

- A) Zvýšená latencia v dôsledku šifrovania a dešifrovania paketov
- B) Znížená kvalita prenosu spôsobená v dôsledku použitia NAT
- C) Väčší objem prenášaných dát v dôsledku pridaných záhlaví
- D) Zablokovanie komunikácie medzi zariadeniami, ktoré nepodporujú IPsec

**10. V konfigurácii IPsec spojenia je parameter `left` používaný na určenie:**

- A) Dátumu vypršania platnosti certifikátu
- B) IP adresy vzdialeného servera
- C) Lokálnej IP adresy koncového zariadenia, kde je konfigurácia definovaná
- D) Zdieľaného hesla pre tunelové šifrovanie

## 4. Literatúra

- [1] CLOUDFARE. *IP Spoofing Explained*. Cloudflare.com [online]. [cit. 2024-11-26]  
Dostupné z: <https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/>
- [2] CLOUDFARE. *SYN flood attack*. [online]. 2023. [cit. 2024-11-26]. Dostupné z:  
<https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/>
- [3] KAVISANKAR, L. and CHELLAPPAN, C. *A mitigation model for TCP SYN flooding with IP spoofing*. In: 2011 International Conference on Recent Trends in Information Technology (ICRTIT). 2011. s. 251–256. [online]. Dostupné z:  
<https://ieeexplore.ieee.org/document/5972435> [cit. 2024-11-26].
- [4] CISCO. IPsec Overview. [online]. 2023 [cit. 2025-04-13]. Dostupné z:  
[https://www.cisco.com/c/en/us/td/docs/net\\_mgmt/vpn\\_solutions\\_center/2-0/ip\\_security/provisioning/guide/IPsecPG1.html](https://www.cisco.com/c/en/us/td/docs/net_mgmt/vpn_solutions_center/2-0/ip_security/provisioning/guide/IPsecPG1.html)
- [5] FERGUSON, P. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. [Internet Requests for Comments]. RFC Editor, 2000. [cit. 2024-11-26]. Dostupné z:  
<https://datatracker.ietf.org/doc/html/rfc2827>
- [6] STRONGSWAN. *strongSwan – the OpenSource IPsec-based VPN Solution*. [online]. 2024 [cit. 2025-04-20]. Dostupné z: <https://www.strongswan.org/>

## **Príloha D - Dokumentácia pre vyučujúceho k laboratórnej úlohe č. 4**

Laboratórna úloha č. 4

### **BEZPEČNOSŤ SIEŤOVEJ VRSTVY**

# 1. Základné informácie k laboratórnej úlohe

Laboratórna úloha č. 4 sa venuje bezpečnostným aspektom sieťovej vrstvy ISO/OSI modelu. Cieľom je prakticky demonštrovať, ako je možné narušiť sieťovú komunikáciu pomocou techniky IP *spoofing* a následne túto komunikáciu zabezpečiť implementáciou protokolu IPsec.

Študenti pracujú v prostredí troch virtuálnych strojov s operačným systémom Kali Linux, kde každý stroj zohráva inú úlohu – klient, server a útočník. V priebehu úlohy budú študenti simulovať IP *spoofing* útok pomocou nástroja **hping3**, analyzovať prenosy vo Wiresharku a následne implementovať zabezpečenú komunikáciu pomocou knižnice **strongSwan**, ktorá umožňuje implementovať IPsec protokol v transportnom režime. Nakoniec budú študenti porovnávať účinnosť a základné charakteristiky transportného a tunelového režimu IPsec z hľadiska bezpečnosti a výkonu siete.

## 2. Očakávané výstupy práce študentov

Po úspešnom absolvovaní tejto úlohy by mali študenti byť schopní popísať a prakticky demonštrovať priebeh útoku typu IP spoofing. Pomocou nástroja hping3 odošlú pakety s falošnou zdrojovou IP adresou a následne tieto pakety zachytia a analyzujú v nástroji Wireshark, kde porovnajú IP a MAC adresy v záhlaví zachytených dátových jednotiek.

Ďalej by mali študenti správne nakonfigurovať IPsec komunikáciu v transportnom režime medzi klientom a serverom, pre dosiahnutie toho využiť *open-source* knižnicu strongSwan a následne overiť stav spojenia pomocou príkazu **ipsec statusall**. Dôležitým bodom je aj porovnanie obsahu zachytenej komunikácie pred a po nasadení zabezpečenia IPsec, kde by v prípade úspešnej konfigurácie mali byť v zázname komunikácie viditeľné šifrované ESP pakety. Overenie výstupov a tiež správnosti implementácie zabezpečenia IPsec prebieha na základe analýzy paketov vo Wiresharku alebo prostredníctvom nástroja **tcpdump** a zároveň kontrolou aktívneho IPsec spojenia. Úspešnosť úlohy je možné hodnotiť podľa schopnosti študenta jasne vysvetliť a demonštrovať rozdiely medzi nezašifrovanou a zašifrovanou komunikáciou.

### 2.1. Riešenie samostatnej úlohy

V rámci samostatnej úlohy majú študenti za úlohu simulovať prostredie verejnej siete, čo môžu dosiahnuť napríklad zmenou typu sieťového rozhrania v prostredí VMware (NAT alebo bridged). V tomto prostredí následne nakonfigurujú IPsec spojenie medzi klientom a serverom, tentokrát však v tunelovom režime. Ich cieľom bude zaistiť, aby celá IP komunikácia prechádzala cez šifrovaný tunel, čím sa zaistí, že šifrovaná bude nielen dátová časť paketu, ale tiež pôvodné IP záhlavie. V nástroji Wireshark by mali byť viditeľné iba šifrované ESP pakety, ktorých dátovú časť nebude

možné priamo zobrazit', resp. čítať v otvorenej podobe, čím sa preukáže úspešná implementácia tunelového režimu.

Študenti následne vykonajú meranie výkonnostných parametrov siete (latencia a priepustnosť) v rôznych režimoch – nezašifrovaná komunikácia, IPsec v transportnom a v tunelovom režime. Na základe zobrazených výsledkov vypracujú jednoduchú krátku správu, v ktorej porovnávajú výhody a nevýhody oboch bezpečnostných režimov z pohľadu výkonu a úrovne zabezpečenia. Táto časť je dôležitá nielen pre overenie praktických zručností študentov, ale aj schopnosti správne analyzovať a vhodným spôsobom interpretovať dosiahnuté výsledky.

## 2.2. Odpovede na kontrolné otázky

### 1. Čo je cieľom IP *spoofing* útoku?

- A) Zmeniť MAC adresu útočníka
- B) Získať neautorizovaný prístup predstieraním cudzej IP adresy ☒
- C) Presmerovať legitímnu komunikáciu cez vlastné zariadenie
- D) Zamedziť šifrovaniu dát medzi serverom a klientom

### 2. Ktoré z nasledujúcich tvrdení platia o nástroji hping3?

- A) Umožňuje simulovať IP *spoofing* a rôzne typy sieťových útokov ☒
- B) Je určený na šifrovanie komunikácie pomocou IPsec
- C) Dokáže vygenerovať vlastné TCP/IP pakety podľa špecifikácie ☒
- D) Je to nástroj na konfiguráciu VPN tunelov medzi vzdialenými sieťami

### 3. Vyberte nesprávne tvrdenia týkajúce sa IP *spoofing* útoku:

- A) IP spoofing automaticky zahŕňa zmenu MAC adresy ☒
- B) Má za následok zvýšenie prenosovej rýchlosti v sieti ☒
- C) Spoofovaný paket má zvyčajne neplatný kontrolný súčet ☒
- D) Využíva manipuláciu s IP záhlavím paketov

### 4. Aký je hlavný rozdiel medzi transportným a tunelovým režimom IPsec?

- A) V tunelovom režime sa šifruje len záhlavie IP paketu
- B) Transportný režim sa používa v bezdrôtových sieťach
- C) V transportnom režime sú šifrované len užívateľské dáta, IP záhlavie paketu ostáva nezmenené ☒
- D) Tunelový režim nemôže byť využitý v IPv6 sieti

### 5. Čo spôsobí nastavenie parametra `authby=secret` v súbore `ipsec.conf`?

- A) Povolenie anonymného prístupu
- B) Vypnutie autentizácie
- C) Autentizáciu pomocou predzdieľaného tajomstva (PSK) ☒
- D) Použitie certifikátov

**6. Protokol AH (Authentication Header) v IPsec:**

- A) Umožňuje zašifrovať celý IP paket
- B) Zaisťuje autentizáciu a integritu paketu bez šifrovania ☒
- C) Poskytuje možnosť tunelovania prenosu cez HTTPS
- D) Zaisťuje dôvernosť riadiacich informácií v IP záhlaví

**7. Vyberte nesprávne tvrdenia o tunelovom režime IPsec:**

- A) Zabezpečuje celý IP paket vrátane pôvodného záhlavia
- B) Nie je vhodný pre spojenie medzi dvoma bránami ☒
- C) Používa sa najmä pri zabezpečení VPN
- D) Prenáša pakety cez šifrovaný SSH tunel ☒

**8. V akých situáciách je vhodné použiť IPsec v transportnom režime?**

- A) Komunikácia medzi klientom a serverom v rovnakej sieti ☒
- B) Prepojenie dvoch vzdialených sietí cez internet
- C) Na zabezpečenie SSH spojenia
- D) Ochrana komunikácie medzi aplikáciami v rámci jedného servera ☒

**9. Ako môže použitie IPsec ovplyvniť výkonnosť počítačovej siete?**

- A) Zvýšená latencia v dôsledku šifrovania a dešifrovania paketov ☒
- B) Znížená kvalita prenosu spôsobená v dôsledku použitia NAT
- C) Väčší objem prenášaných dát v dôsledku pridaných záhlaví ☒
- D) Zablokovanie komunikácie medzi zariadeniami, ktoré nepodporujú IPsec ☒

**10. V konfigurácii IPsec spojenia je parameter left používaný na určenie:**

- A) Dátumu vypršania platnosti certifikátu
- B) IP adresy vzdialeného servera
- C) Lokálnej IP adresy koncového zariadenia, kde je konfigurácia definovaná ☒
- D) Zdieľaného hesla pre tunelové šifrovanie

### **2.3. Dopĺňajúce otázky**

Nižšie uvedené otázky môžu byť využité pri kontrole výstupov samostatnej práce študentom s cieľom overiť, či skutočne porozumeli riešenej problematike v praktickej časti laboratórnej úlohy.

**1. Aký vplyv má použitie zabezpečenia pomocou IPsec-u na výkon siete (latenciu, rýchlosť pre-nosu dát, atď.)?**

- Použitie IPsec-u môže ovplyvniť výkon siete, a to predovšetkým zvýšením latencie a miernym znížením priepustnosti. Tento efekt je spôsobený dodatočným spracovaním paketov v procesoch šifrovania a dešifrovania na



oboch koncoch komunikácie. V tunelovom režime môže byť vplyv ešte výraznejší, keďže dochádza k zapuzdreniu celého IP paketu, čím sa zväčšuje aj jeho veľkosť.

## 2. Vysvetlite princíp útoku IP spoofing.

- Útok IP spoofing spočíva v podvrhnutí zdrojovej IP adresy v paketoch generovaných na strane útočníka tak, aby sa tieto pakety javili, že pochádzajú z dôveryhodného zariadenia (obete). Tento typ útoku umožňuje obchádzať bezpečnostné mechanizmy a ACL pravidlá a môže byť tiež využitý ako súčasť širších útokov, napríklad DDoS.

## 3. Aký je rozdiel medzi transportným a tunelovým režimom IPsec?

- Pri použití IPsec v transportnom režime sa šifruje iba dátová časť paketu, pričom IP záhlavie zostáva pôvodné, nezmenené. Tento režim sa najčastejšie využíva za účelom zabezpečenia komunikácie medzi dvoma koncovými bodmi. Naopak, tunelový mód šifruje celý pôvodný IP paket vrátane záhlavia, ktorý je následne zapuzdrený do nového IP paketu s priradeným novým IP záhlavím. Využitie tunelového režimu je zväčša pre zabezpečenie komunikácie medzi dvoma sieťami alebo bránami.

## 4. Vysvetlite funkciu a účel použitia jednotlivých súčastí IPsec-u, a to konkrétne protokolov AH a ESP.

- Protokol AH (*Authentication Header*) zabezpečuje autentizáciu a integritu prenášaných dát vrátane IP záhlavia, pričom ale nešifruje ich obsah. Protokol ESP (*Encapsulating Security Payload*) umožňuje šifrovanie, čím zabezpečuje dôvernosť komunikácie, a zaisťuje tiež integritu, avšak len dátovej časti paketu. Za účelom dosiahnutia komplexnej ochrany prebiehajúcej komunikácie, t. j. zaistenia autentickosti, dôvernosti a integrity dátového obsahu, je vhodné používať protokoly AH a ESP súčasne.

## 5. Akým spôsobom je možné s využitím programu Wireshark overiť správnosť fungovania IPsec? Demonštrujte na príklade, môžete využiť časť zachytenej dátovej komunikácie.

- Vo Wiresharku je možné filtrovať komunikáciu protokolu ESP s použitím filtra **esp**. Po úspešnej konfigurácii zabezpečenia IPsec by mali byť pakety v tomto formáte nečitateľné, čo znamená, že ich obsah (dátová časť) bude zašifrovaný. V porovnaní s nezašifrovanou komunikáciou, kde sú údaje viditeľné (napr. komunikácia protokolov ICMP alebo HTTP), je to jasný dôkaz funkčného IPsec spojenia. Okrem toho je možné analyzovať aj záhlavia paketov a overiť, že ESP zabezpečuje komunikáciu medzi správnymi zariadeniami.

## **Príloha E - Text laboratórnej úlohy č. 8**

Laboratórna úloha č. 8

### **Autentizácia pomocou EAP a RADIUS**

## 0. Úvod k laboratórnej úlohe

Cieľom laboratórnej úlohy je priblížiť študentom existujúce možnosti centralizovanej autentizácie v porovnaní s mechanizmami pre lokálne overenie identity užívateľov, resp. zariadení, oboznámiť ich s princípmi zabezpečenia prístupu do siete pomocou **autentizačných protokolov EAP (*Extensible Authentication Protocol*) a RADIUS (*Remote Authentication Dial-In User Service*)** a poskytnúť im priestor k získaniu skúseností s implementáciou RADIUS servera ako centrálného prvku pre riadenie prístupu a zabezpečené pripojenie k sieti.

V tejto úlohe získate základné poznatky o priebehu autentizácia klienta vo Wi-Fi sieti s podporou IEEE 802.1X. V teoretickej časti bude vysvetlené akým spôsobom sa konfiguruje FreeRADIUS server a ako prebieha proces overovania identity užívateľa prostredníctvom externého autentizačného mechanizmu. V praktickej časti si najskôr vyskúšate vo vytvorenej virtuálnej sieti pozostávajúcej z troch virtuálnych strojov predstavujúcich klienta, prístupový bod a autentizačný server v prostredí VMware **nasadiť a konfigurovať FreeRADIUS server**, predstavujúci centrálny bod autentizácie jeho prepojenie s prístupovým bodom prostredníctvom protokolu EAP a tiež samotnú autentizáciu klienta. Významnou súčasťou úlohy bude následná analýza priebehu autentizačného procesu pomocou sieťovej analýzy, ktorá študentom umožní lepšie porozumieť fungovaniu úzkej spolupráce medzi jednotlivými vrstvami sieťovej architektúry, predovšetkým medzi linkovou, sieťovou a aplikačnou vrstvou.

### Požiadavky pre vypracovanie úlohy:

- software: VMware Workstation Player pre virtualizáciu staníc,
- virtuálne stroje: tri virtuálne stroje s Kali Linux.

## 1. Teoretický úvod

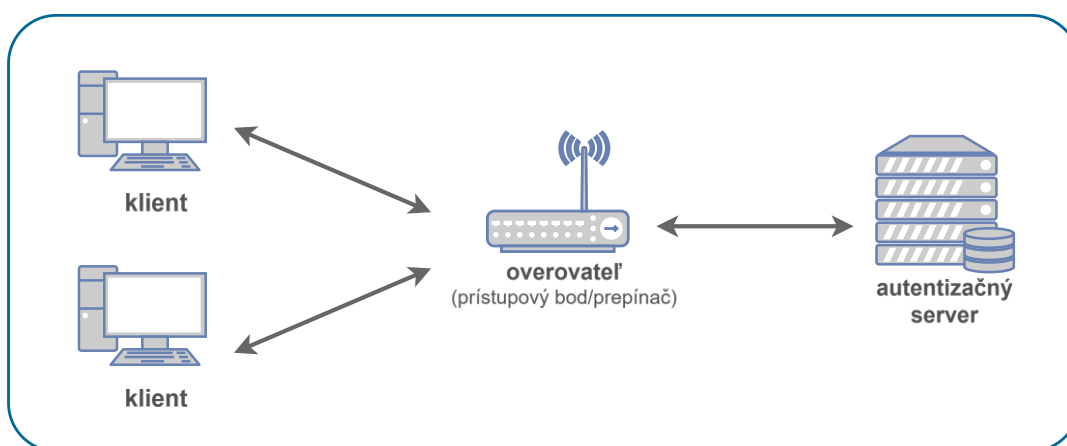
V tejto laboratórnej úlohe venovanej autentizačným protokolom EAP a RADIUS sa zoznámite s možnosťami centralizovaného riadenia prístupu a autentizácie užívateľov v počítačových sieťach. Autentizácia predstavuje jeden z najdôležitejších aspektov nevyhnutných ku dosiahnutiu maximálnej úrovne zabezpečenia.

Kombinácia **autentizačnej metódy EAP (*Extensible Authentication Protocol*) a systému RADIUS (*Remote Authentication Dial-In User Service*)** sa využíva najmä v podnikových Wi-Fi sieťach a VPN pre centralizovanú správu overovania identity overovania používateľov a následného riadenia ich prístupu k zdieľaným sieťovým prostriedkom.

## 1.1. IEEE 802.1X

IEEE 802.1X je sieťový štandard definovaný organizáciou IEEE<sup>35</sup> slúžiaci pre kontrolu prístupu do lokálnych sietí. Je základom pre kontrolu prístupu realizovanej na úrovni fyzických portov (tzv. *port-based network access control* – PNAC), čo v praxi znamená, že zariadenie (resp. klient) nemôže získať plnohodnotný prístup k sieťovým službám, pokiaľ úspešne neprebehne proces jeho autentizácie. Tento proces prebieha na základe spolupráce troch hlavných súčastí:

- **klient** (*supplicant*) – resp. koncové zariadenie (čo môže byť napr. notebook alebo virtuálny stroj), ktoré sa pokúša pripojiť do siete a predkladá svoju identitu;
- **overovateľ** (*authenticator*) – spravidla prístupový bod<sup>36</sup> do bezdrôtovej siete alebo prepínač (switch), ktorý reguluje prístup koncových klientov do siete a sprostredkúva výmenu autentizačných informácií medzi daným klientom a autentizačným serverom;
- **autentizačný server** – overovací server (typicky RADIUS), ktorý na základe stanovených konfiguračných pravidiel a uložených prístupových údajov v procese autentizácie rozhodne, či bude klientovi povolený alebo zamietnutý prístup do siete.



Obrázok 1.1 Základné komponenty 802.1X a ich vzájomné prepojenie <sup>37</sup>.

<sup>35</sup> IEEE (*Institute of Electrical and Electronics Engineers*) je medzinárodná organizácia zameraná na vývoj technických štandardov v oblasti elektrotechniky, elektroniky, výpočtovej techniky a telekomunikácií. Organizácia IEEE je zodpovedná za tvorbu mnohých známych sieťových štandardov, medzi ktoré patria, okrem vyššie uvedeného IEEE 802.1X pre autentizáciu, tiež napr. IEEE 802.3 pre špecifikáciu technológie Ethernet alebo IEEE 802.11 zameraný na bezdrôtové technológie Wi-Fi.

<sup>36</sup> Pre označenie prístupového bodu bezdrôtovej siete je zaužívané použitie skratky „AP“ (z angl. termínu slova *Access Point*).

<sup>37</sup> Prevzaté z [1].

Štandard IEEE 802.1X sa často využíva v kombinácii s protokolom EAP (*Extensible Authentication Protocol*) a v podnikových sieťach predstavuje bežný spôsob riadenia bezpečného prístupu. Podrobné znenie štandardu IEEE 802.1X možno nájsť na oficiálnych stránkach [2], bližšie vysvetlenie problematiky a priebeh procesu autentizácie je vysvetlený napr. v literatúre [3], [4], [5].

## 1.2. Extensible Authentication Protocol (EAP)

**EAP predstavuje flexibilný autentizačný *framework*** navrhnutý za účelom podpory rôznych metód autentizácie pre prístupujúcich užívateľov. Nejedná sa o konkrétne autentizačný mechanizmus, resp. protokol, ale o rámec, ktorý implementuje **niekoľko rôznych autentizačných metód**. Jednotlivé autentizačné metódy sú založené na princípe výmeny správ medzi klientom (napr. používateľským zariadením) a autentizačným serverom (napr. RADIUS serverom). Je široko využívaný v bezdrôtových sieťach a tiež zabezpečených VPN pripojeniach.

EAP definuje rôzne metódy autentizácie, pričom medzi najznámejšie patria:

- **EAP-MD5:** jednoduchá autentizačná metóda založená na princípe výzva-odpoveď, kedy klient žiadajúci o overenie identity obdrží od autentizačného servera náhodne generovanú výzvu (*challenge*), na ktorú odosiela odpoveď (*response*) obsahujúcu hash určený z overovacieho faktoru klienta, typicky hesla, a náhodnej výzvy, ktorú získal od servera. Túto metódu však nemožno považovať za dostatočne bezpečnú, nakoľko sú obe správy (výzva aj odpoveď) prenášané nešifrovane v otvorenej podobe.
- **EAP-TLS:** metóda využíva certifikáty pre autentizáciu klienta a servera. Jedná sa o najbezpečnejšiu alternatívu spomedzi známych a využívaných metód, avšak prináša nutnosť zaistenia mechanizmov pre vydávanie a následnú správu kryptografických certifikátov.
- **EAP-TTLS:** rozširuje metódu EAP-TLS o podporu kombinácie certifikátov a použitia tradičných prihlasovacích (autentizačných) údajov, napr. v podobe prístupového mena a hesla. Certifikát sa používa len pre overenie identity servera, čo zjednodušuje implementáciu autentizačnej metódy.
- **PEAP** (*Protected EAP*): metóda obsahuje mechanizmy pre tunelovanie autentizačných údajov cez šifrované spojenie pomocou TLS. Používateľské meno a heslo sú prenášané vnútri vytvoreného bezpečného tunela, čo zvyšuje mieru zaistenia dôvernosti autentizačných údajov žiadateľa o overenie identity.

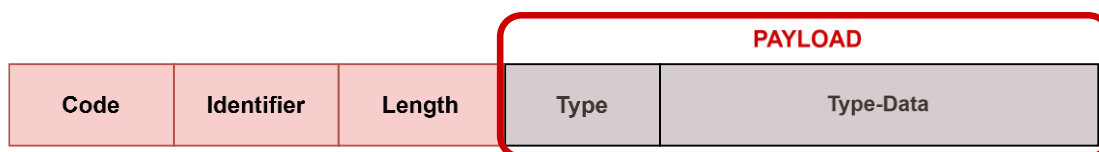
Protokol EAP definuje základný rámec pre autentizačný proces. Každá autentizačná metóda využíva rovnaké typy základných EAP správ, ktoré si medzi sebou vymieňajú klient a autentizačný server (sprostredkované cez AP, resp. prepínač). Rozlišujeme štyri základné typy EAP správ, ich prehľad je uvedený v tab. 1.1.

Tabuľka 1.1 Prehľad typov EAP správ.

| Typ správy                         | Kód | Odosielateľ | Popis                                                                               |
|------------------------------------|-----|-------------|-------------------------------------------------------------------------------------|
| <b>EAP-Request</b><br>(požiadavka) | 1   | Server      | Výzva klientovi pre zadanie požadovaných údajov (napr. identita, heslo, certifikát) |
| <b>EAP-Response</b><br>(odpoveď)   | 2   | Klient      | Odpoveď klienta na výzvu servera – obsahuje požadované údaje.                       |
| <b>EAP-Success</b>                 | 3   | Server      | Potvrdenie úspešnej autentizácie – klient má povolený prístup do siete.             |
| <b>EAP-Failure</b>                 | 4   | Server      | Oznámenie o neúspešnej autentizácii – prístup klienta je zamietnutý.                |

Každá EAP správa má nasledovný základný formát (viď obr. 1.2 nižšie):

- **Code** – označuje typ správy (1=Request, 2=Response, 3=Success, 4=Failure);
- **Identifier** – slúži pre spárovanie výzvy a odpovede;
- **Length** – dĺžka EAP správy v B;
- **Payload** – pole využívané v procese autentizácie, je obsiahnuté len v správach EAP-Request a EAP-Response a ďalej sa delí na:
  - **Type**<sup>38</sup> – určuje, čo bude obsahom nasledujúcich dát (Type-Data) – napríklad, či ide o výmenu identity klienta, výzvu na zadanie hesla, odoslanie certifikátu apod. Pole definuje konkrétnu EAP metódu alebo príslušný krok, resp. fázu autentizačného procesu.
  - **Type-Data** – obsahuje vlastné dáta podľa konkrétneho typu správy (napr. reťazec s identitou klienta, náhodná výzva, hash hesla, certifikát atď.)



Obrázok 1.2 Schematické znázornenie štruktúry správy protokolu EAP<sup>39</sup>.

<sup>38</sup> Príklady hodnôt poľa Type sú napr.: **1 = Identity**: požiadavka/odpoveď s identitou (napr. používateľské meno); **2 = Notification**: správa slúži pre informovanie klienta, resp. žiadateľa; **3 = NAK**: odmietnutie typu autentizácie zo strany klienta; **4 = MD5-Challenge**: správa slúži pre odoslanie výzvy a odpovede počas autentizácie MD5.

Uvedený zoznam možných hodnôt nie je konečný, uvedený je len prehľad základných typov, s ktorými ste mali možnosť oboznámiť sa aj na prednáškach predmetu MPC-NSB.

<sup>39</sup> Prevzaté z oficiálnych výučbových materiálov k prednáškam predmetu MPC-NSB – vypracoval garant predmetu a prednášajúcim doc. Karel Burda, CSs. (viď tiež E-learning predmetu).

V procese autentizácie si medzi sebou klient (resp. žiadateľ) a autentizačný server vymieňajú EAP správy sprostredkované cez AP. Typicky **server odosiela požiadavky** (*Request*), na ktoré **klient reaguje svojou odpoveďou** (*Response*). V závislosti na konkrétnom type použitej autentizačnej metódy sa potom líši obsah dátovej časti prenášanej v príslušnej EAP správe. Ako príklad možno uviesť rôzne typy správ, ktoré sú odosielané v procese autentizácie MD5<sup>40</sup>:

- **EAP-Request/Identity:**  
Server žiada klienta o jeho meno (identitu). V poli Type sa uvádza **1 (Identity)**, nasleduje reťazec „Who are you?“.
- **EAP-Response/Identity:**  
Klient ako odpoveď odosiela svoju identitu, zvyčajne vo forme používateľského mena. V poli Type sa opäť uvádza **1 (Identity)**, nasleduje reťazec s identitou.
- **EAP-Request/MD5-Challenge:**  
Server posíla výzvu (*challenge*) pre zadanie autentizačných údajov.
- **EAP-Response/MD5-Challenge:**  
Klient reaguje na výzvu servera a odpovedá správou, ktorá obsahuje údaje pre overenie identity. V prípade autentizácie MD5 sa jedná o *hash* reťazca zloženého z ID žiadateľa, hesla a výzvy, ktorú prijal.

### Autentizácia EAP-MD5

EAP-MD5 je jednoduchá autentizačná metóda používaná pre overovanie identity užívateľov v počítačových sieťach. Funguje tak, že server (autentizátor) pošle klientovi výzvu (*challenge*), a klient na základe svojho mena, hesla a prijatej výzvy vypočíta odpoveď (*response*), ktorú odosiela späť. Server následne porovná klientovu odpoveď s očakávaným výsledkom a na základe toho povolí alebo naopak zamietne prístup klienta do siete.

Výhodou metódy je jej rýchlosť a jednoduchosť, ale medzi hlavné nevýhody patrí absencia mechanizmov pre šifrovanie a akúkoľvek formu ochrany samotných údajov – heslo klienta je totiž prakticky možné získať pomocou útoku hrubou silou<sup>41</sup>. EAP-MD5 nepodporuje overenie servera, a preto sa považuje za nevhodnú pre verejné siete, ale výborne sa hodí na výučbové účely pre vysvetlenie základných princípov a jednotlivých krokov autentizačného procesu.

Priebeh autentizácie EAP-MD5 zahŕňa celkom päť základných krokov, ktorých rozbor prináša nasledujúca tabuľka č. 1.2 (schematické znázornenie priebehu komunikácie vid' na obr. 1.3).

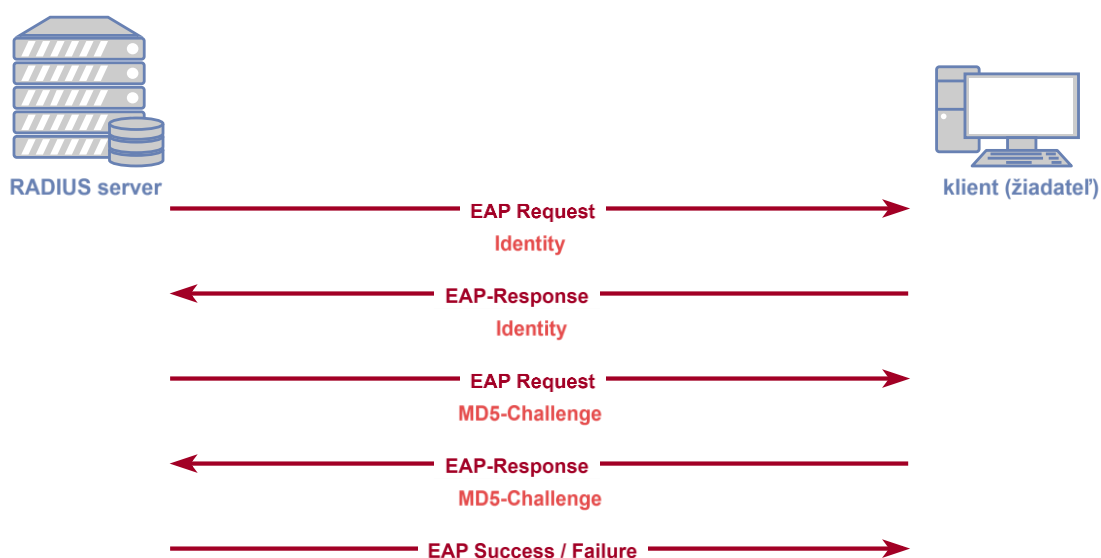
---

<sup>40</sup> Popis autentizácie MD5 bude nasledovať.

<sup>41</sup> Pokiaľ útočník zachytí správu s identitou klienta (ID), výzvu servera a následne odpoveď od klienta obsahujúcu *hash* reťazca (ID || heslo || výzva), mohol by sa teoreticky pokúsiť postupným skúšaním rôznych možných vstupov (resp. hesiel) nájsť výstupný *hash* zhodný s tým od klienta. V prípade úspechu by bolo zrejmé, že našiel odpovedajúce heslo.

Tabuľka 1.2 Správy odosielané v procese autentizácie EAP-MD5.

| <b>Krok</b> | <b>Odosielateľ →<br/>Prijímateľ</b> | <b>Typ správy</b>                           | <b>Obsah</b>                                                                             |
|-------------|-------------------------------------|---------------------------------------------|------------------------------------------------------------------------------------------|
| 1           | Autentizátor → Klient               | <b>EAP-Request</b><br><b>Identity</b>       | Výzva na zaslanie identity (napr. „Kto si?“)                                             |
| 2           | Klient → Autentizátor               | <b>EAP-Response</b><br><b>Identity</b>      | Klient odošle svoju identitu (napr. „peter“)                                             |
| 3           | Autentizátor → Klient               | <b>EAP-Request</b><br><b>MD5-Challenge</b>  | Výzva obsahujúca náhodný reťazec ( <i>challenge</i> ) a ID                               |
| 4           | Klient → Autentizátor               | <b>EAP-Response</b><br><b>MD5-Challenge</b> | Klient vygeneruje MD5 hash: MD5(ID + heslo + <i>challenge</i> )                          |
| 5           | Autentizátor → Klient               | EAP-Success /<br>EAP-Failure                | Server porovná vypočítaný hash a pošle odpoveď o výsledku autentizácie (úspech/neúspech) |



Obrázok 1.3 Schematické znázornenie výmeny správ medzi klientom a autentizačným serverom pri EAP-MD5 autentizácii<sup>42</sup>.

<sup>42</sup> Prevzaté z oficiálnych výučbových materiálov k prednáškam predmetu MPC-NSB – vypracoval garant predmetu a prednášajúcim doc. Karel Burda, CSs. (viď tiež E-learning predmetu).



### EAPoL (EAP over LAN)

Pri komunikácii prebiehajúcej v sieťach IEEE 802.1X sa EAP správy medzi klientom (*supplicant*) a prístupovým bodom (*authenticator*) prenášajú vo formáte EAP over LAN na spojovej vrstve – tzv. **EAPoL správy**. EAPoL predstavuje rozšírenie protokolu EAP, navrhnuté špecificky pre použitie v sieťach založených na prenosových technológiách Ethernet a Wi-Fi, kde sa autentizačné údaje prenášajú priamo na spojovej vrstve, t. j. na druhej vrstve RM ISO/OSI. EAPoL slúži len na komunikáciu klienta s AP a jeho úlohou je prenášať EAP správy zapuzdrené do ethernetového rámca na spojovej vrstve. Štandard EAPoL definuje päť základných typov správ (viď tab. 1.3).

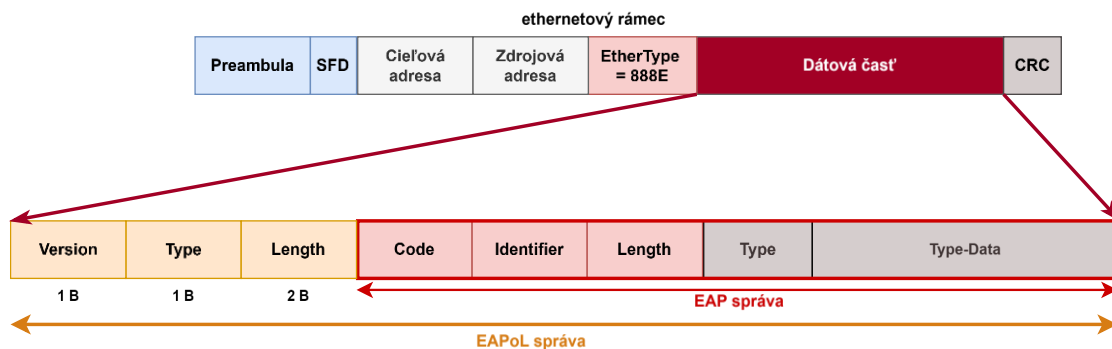
Tabuľka 1.3 Prehľad typov EAPoL správ.

| Typ  | Názov správy                 | Účel                                                                               |
|------|------------------------------|------------------------------------------------------------------------------------|
| 0x00 | EAP-Packet                   | Obsahuje samotnú EAP správu                                                        |
| 0x01 | EAPoL-Start                  | Iniciuje autentizačný proces                                                       |
| 0x02 | EAPoL-Logoff                 | Označuje, že klient sa odhlasuje zo siete, slúži k ukončeniu spojenia              |
| 0x03 | EAPoL-Key                    | Používa sa na výmenu šifrovacích kľúčov (napr. v protokole WPA/WPA2)               |
| 0x04 | EAPoL-Encapsulated-ASF-Alert | Menej bežná, málo používaný typ správy určený pre špecifické bezpečnostné výstrahy |

Proces zapuzdrenia EAPoL správy do ethernetového rámca na úrovni spojovej vrstvy je graficky znázornený na obr. 1.4, pohľad na štruktúru samotnej EAPoL správy taktiež. Popis a význam jednotlivých polí nasleduje:

- **Version** – verzia protokolu EAPoL (napr. 0x01 pre IEEE 802.1X);
- **Type** – typ EAPoL správy (napr. 0x00 pre EAP-Packet);
- **Length** – dĺžka dátovej časti správy v B;
- **Body** – samotný obsah, resp. správa EAP protokolu (napr. EAP výzva (*Request*), odpoveď (*Response*), apod.) – pole je prítomné len u správy typu **EAP-Packet**.

Prístupový bod (AP) následne zapuzdruje tieto EAP správy do správ protokolu RADIUS na aplikačnej vrstve a preposiela ich ďalej na autentizačný server (podrobnejší popis bude nasledovať).



Obrázok 1.4 Schematické znázornenie zapuzdrenia EAPoL správy do rámca technológie Ethernet<sup>43</sup>.

Výhodou EAP je jeho modularita a tak isto aj schopnosť prispôbiť sa rôznym scenárom z reálneho sveta. Metódy *frameworku* EAP predstavujú základné piliere zabezpečenia najmä v dnešných Wi-Fi sieťach s WPA-Enterprise, nakoľko umožňujú výber spomedzi širokého spektra rozličných autentizačných metód podľa konkrétnych požiadaviek.

Podrobný popis architektúry, typov a formátov odosielaných správ a používaných autentizačných metód je možné nájsť v oficiálnom štandarde definujúcom EAP, dokumente RFC: *Extensible Authentication Protocol* (EAP), viď [6]. Viac o jednotlivých autentizačných metódach dostupných v rámci *frameworku* EAP, vrátane detailnej analýzy výhod a naopak obmedzení a nedostatkov jednotlivých prístupov sa možno dočítať v literatúre [7].

### 1.3. RADIUS (*Remote Authentication Dial-In User Service*)

RADIUS predstavuje aplikačný protokol založený na architektúre *klient-server*, ktorý poskytuje **služby AAA** (*Authentication, Authorization, Accounting*), teda autentizáciu, autorizáciu a účtovanie. Bol vyvinutý ako mechanizmus centralizovanej autentizácie užívateľov. V kontexte IEEE 802.1X je kľúčovým prvkom autentizačný server RADIUS, kedy v lokálnych sieťach zastupuje jeho úlohu prístupový bod, ktorý vystupuje ako klient protokolu RADIUS a všetky správy prijaté od klient preposiela na autentizačný server.

Medzi hlavné súčasti architektúry RADIUS patrí:

- **klient RADIUS:** zariadenie, ktoré pre prístupujúceho užívateľa sprostredkúva proces autentizácie (napr. Wi-Fi prístupový bod alebo VPN server). Toto zariadenie prijíma požiadavky na pripojenie od používateľov a odosiela ich na autentizačnému RADIUS serveru;

<sup>43</sup> Prevzaté z oficiálnych výučbových materiálov k prednáškam predmetu MPC-NSB – vypracoval garant predmetu a prednášajúcim doc. Karel Burda, CSs. (viď tiež E-learning predmetu). Pozn.: podrobný popis položiek záhlavia Ethernetu nie je uvádzaný, nakoľko sa predpokladá na strane študentov znalosť problematiky (viď E-learning predmetu MPC-NSB, Téma 3).

- **RADIUS server:** spracováva autentizačné požiadavky, overuje zadané autentizačné údaje a na základe výsledku procesu overovania identity tiež rozhoduje o pridelení, resp. zamietnutí prístupu;
- **databáza užívateľov:** jedná sa o úložisko obsahujúce prihlasovacie (autentizačné) údaje užívateľov, prípadne i ďalšie informácie a oprávnenia potrebné pre účely následného riadenia prístupu k zdieľaným prostriedkom. Databázou môže byť napr. súbor, LDAP server alebo iný adresárový systém.

Protokol RADIUS využíva na transportnej vrstve jednoduchý bezstavový protokol UDP a typicky komunikuje na portoch 1812 (pre autentizáciu) a 1813 (účtovanie). Po úspešnej autentizácii môže byť prístupujúcemu užívateľovi povolený prístup do siete alebo k požadovaným prostriedkom.

### Autentizačný server RADIUS v procese autentizácie

V procese autentizácie EAP-MD5 vystupujú tri kľúčové komponenty – klient (žiadateľ), prístupový bod (AP) a autentizačný server RADIUS (viď obr. 1.5). Prístupový bod vystupuje v úlohe sprostredkovateľa, ktorý iba prenáša autentizačné správy medzi klientom a serverom, pričom, komunikácia medzi klientom a AP prebieha vo forme EAP správ zapuzdrených v protokole EAPoL (EAP over LAN), ktorý sa prenáša cez Ethernet.



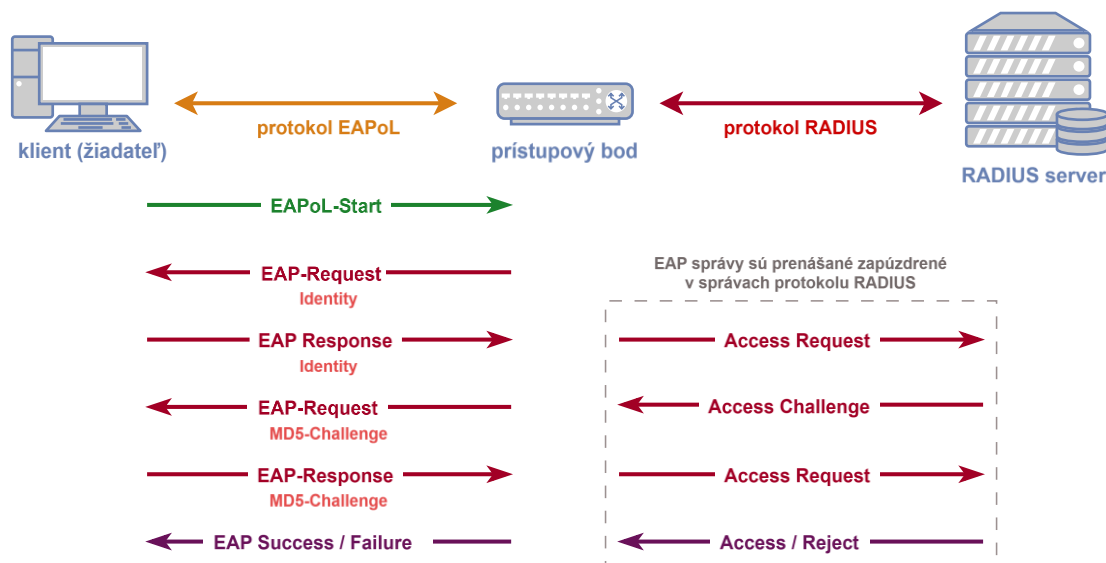
Obrázok 1.5 Komponenty siete s autentizáciou EAP-MD5 a ich vzájomná komunikácia.

Autentizačný server je zodpovedný za samotné overenie identity klienta. Keď AP získa identitu klienta vo forme správy **EAP-Response/Identity**, zapuzdruje túto správu do správy protokolu RADIUS typu **Access-Request** a odosiela ju na server. Server odpovedá výzvou typu **Access-Challenge**, ktorá obsahuje náhodný reťazec, (výzvu, *challenge*), a ten je následne odoslaný klientovi cez AP vo forme **EAP-Request/MD5-Challenge**.

Klient z výzvy, ID výmeny a svojho hesla vypočíta *hash* a ten odosiela ako reakciu (odpoveď, *response*) späť ako **EAP-Response/MD5-Challenge**. Túto odpoveď AP opäť zabalí do správy **Access-Request** a preposiela na RADIUS server. Server overí správnosť výpočtu (porovná ho s vlastnou vypočítanou hodnotou *hash*) a na základe toho rozhodne o úspešnosti autentizácie. V prípade úspechu odošle správu **Access-Accept**,

inak **Access-Reject**. AP následne doručí klientovi výslednú správu **EAP-Success**, resp. **EAP-Failure**.

Takto navrhnutý mechanizmus zabezpečuje, že heslo klienta nie je nikdy prenášané v otvorenej podobe. AP pritom nepozná žiadne autentizačné údaje – jeho úlohou je len zapuzdrenie EAP správ do RADIUS formátu a späť. Protokol RADIUS sa tak používa výlučne medzi AP a autentizačným serverom, zatiaľ čo EAPoL slúži na prenos správ medzi klientom a AP. Podrobne je možné celý priebeh komunikácie vidieť na obr. 1.6.



Obrázok 1.6 Schematické znázornenie autentizácie EAP-MD5 podľa štandardu IEEE 802.1X <sup>44</sup>.

Pre viac informácií a podrobnejšie vysvetlenie a popis vyššie uvedených protokolov a metód autentizácie je možné nahliadnuť do literatúry [8], [9], [10].

## 1.4. Použité nástroje

### Wireshark

Wireshark je sieťový analyzátor, ktorý umožňuje sledovať dátové prenosy. Použitie tohto nástroja ste si prakticky vyskúšali už v rámci niekoľkých predošlých laboratórnych úloh, takže jeho bližší popis nebude už ďalej podrobne uvádzaný.

<sup>44</sup> Prevzaté z oficiálnych výučbových materiálov k prednáškam predmetu MPC-NSB – vypracoval garant predmetu a prednášajúcim doc. Karel Burda, CSs. (viď tiež E-learning predmetu).

## FreeRADIUS

**FreeRADIUS** predstavuje *open-source* možnosť implementácie RADIUS protokolu pre autentizáciu, autorizáciu a účtovanie prístupov v sieťach. Umožňuje implementáciu RADIUS serverov a používa sa v podnikových aj verejných sieťach pre potreby centralizovaného overovania užívateľov. FreeRADIUS podporuje širokú škálu autentizačných protokolov, vrátane EAP-MD5, PEAP, EAP-TTLS, a mnohých ďalších, vďaka čomu je vhodným riešením pre použitie v rôznorodých sieťových prostrediach.

Okrem samotnej autentizácie disponuje FreeRADIUS aj možnosťou monitorovania aktivít užívateľov v sieti a vytvárania záznamov o ich prístupoch k prostriedkom. FreeRADIUS je modulárny a flexibilný, čo umožňuje jeho integráciu s databázami, LDAP servermi a inými externými autentizačnými systémami.

### Inštalácia a konfigurácia RADIUS servera

V rámci praktickej časti budete realizovať vlastnú implementáciu RADIUS servera. Pre tento účel bude využitý nástroj FreeRADIUS<sup>45</sup> dostupný v Kali Linux.

Inštalácia FreeRADIUS na Kali Linux:

```
sudo apt-get install freeradius
```

Konfigurácia autentizačnej politiky v súbore `/etc/freeradius/3.0/users` pre pridanie nového užívateľa:

```
testuser Cleartext-Password := "heslo123"
```

Spustenie vytvoreného RADIUS servera:

```
sudo systemctl start freeradius
```

## Hostapd

**Hostapd** predstavuje softvérové riešenie umožňujúce bežným klientskym zariadeniam (napr. virtuálnemu stroju s Kali Linux) emulovať funkcionality plnohodnotného prístupového bodu (*authenticator*) v zmysle štandardu IEEE 802.1X. Je navrhnutý najmä pre bezdrôtové siete, ale v laboratórnych podmienkach môže byť využitý taktiež aj v simulovanom ethernetovom prostredí. Umožňuje implementáciu autentizačných mechanizmov, správu SSID<sup>46</sup>, zabezpečenie siete a komunikáciu s RADIUS serverom. V rámci laboratórnej úlohy bude nástroj `hostapd` použitý za účelom vytvorenia jednoduchého autentizačného bodu siete, ktorý sprostredkúva výmenu údajov medzi klientom a RADIUS serverom prostredníctvom EAP.

---

<sup>45</sup> Viac informácií o FreeRADIUS je dostupných na oficiálnych stránkach projektu, viď [11].

<sup>46</sup> SSID (*Service Set Identifier*) predstavuje názov bezdrôtovej siete, ktorý sa zobrazuje používateľom pri pripojení ku konkrétnej Wi-Fi.

Základné parametre ako názov siete (SSID), rozhranie sieťovej karty, povolenie mechanizmu autentizácie IEEE 802.1X a údaje o RADIUS serveri sú súčasťou konfiguračného súboru **hostapd.conf**. Príklad konfigurácie:

```
interface=eth0
driver=wired
ssid=EduLab
ieee8021x=1
auth_server_addr=192.168.126.130
auth_server_port=1812
auth_server_shared_secret=heslo1245
```

Vysvetlenie jednotlivých nastavení, ktoré sú obsahom uvedenej konfigurácie:

- **interface=eth0** – určuje sieťové rozhranie, na ktorom bude hostapd počúvať a sprostredkovať autentizačné procesy;
- **driver=wired** – špecifikuje priebeh 802.1X autentizácie cez ethernetové pripojenie;
- **ssid=EduLab** – nastavuje názov siete (SSID), ktorý bude vysielaný AP;
- **ieee8021x=1** – aktivuje podporu autentizačného protokolu IEEE 802.1X;
- **auth\_server\_addr=192.168.126.130** – IP adresa RADIUS servera.
- **auth\_server\_port=1812** – štandardný port pre autentizačné požiadavky použitý na strane RADIUS servera;
- **auth\_server\_shared\_secret=heslo1245** – zdieľané heslo (tajný kľúč) na zabezpečenie komunikácie medzi AP a RADIUS serverom.

Po definícii konfiguračných parametrov a uložení príslušného konfiguračného súboru nasleduje spustenie hostapd pomocou príkazu:

```
sudo hostapd hostapd.conf
```

Uvedený príkaz zabezpečí načítanie konfiguračného súboru a spustenie služby **hostapd** s požadovanými nastaveniami. Je nevyhnutné spustiť ho s oprávneniami administrátora (sudo), pretože práca so službou **hostapd** zahŕňa manipuláciu so sieťovými rozhraniami, ich nastavením a konfiguráciu autentizačných procesov, ktoré vyžadujú vyššie systémové oprávnenia. **Pozn.: správna konfigurácia a spustenie tohto nástroja je nevyhnutná pre úspešné sprostredkovanie autentizácie medzi klientom a RADIUS serverom.**

## Wpa\_supplicant

Jedná sa o softvérový nástroj, ktorý **umožňuje zariadeniu (klientovi) bezpečne sa pripojiť k bezdrôtovej sieti vyžadujúcej autentizáciu pomocou protokolu IEEE 802.1X**. Vytvorená inštancia nástroja **wpa\_supplicant** funguje ako klientská

aplikácia, ktorá obsluhuje a riadi procese autentizácie klienta a sprostredkúva výmenu autentizačných EAP správ medzi klientskym zariadením a prístupovým bodom<sup>47</sup>. Tento nástroj slúži teda k zaisteniu bezpečnej komunikácie klienta s prístupovým bodom (AP) pomocou protokolov 802.1X a EAP, pričom podporuje rôzne typy EAP autentizačných metód (napr. EAP-MD5, EAP-TLS, EAP-TTLS, PEAP a ďalšie).

---

<sup>47</sup> AP zastávajúci funkciu *authenticator* v procese autentizácie IEEE 802.1X.

## 2. Praktická časť

V rámci praktickej časti úlohy získate komplexný prehľad o možnostiach implementácie autentizačných protokolov EAP a RADIUS a tiež vlastné praktické skúsenosti s nastavením a správnou konfiguráciou vhodných metód autentizácie a pokročilého riadenia prístupu k zdrojom v počítačovej sieti.

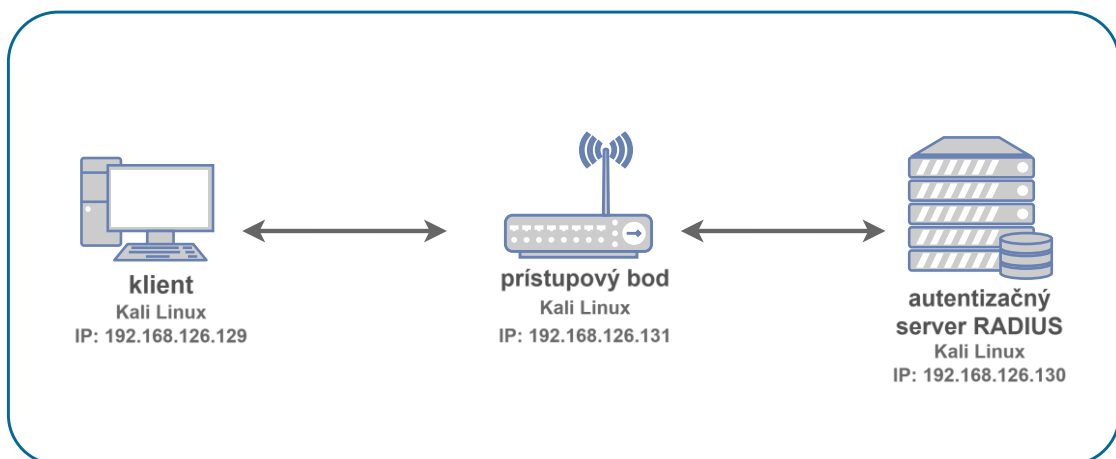
### 2.1. Topológia virtuálnej siete a nastavenie virtuálnych strojov

Vytvorená sieť bude pozostávať z troch virtuálnych strojov:

- **klient:** simulované koncové zariadenie, ktoré sa pripája do siete cez 802.1X
- **Access Point** (*authenticator*) = **prístupový bod:** zariadenie využívajúce `hostapd`, sprostredkovateľ autentizácie, resp. komunikácie klienta s autentizačným RADIUS serverom
- **RADIUS server:** zariadenie s implementovanou inštanciou autentizačného servera pomocou nástroja `freeradius`.

Sieťová konfigurácia:

- klient: 192.168.126.129
- prístupový bod: 192.168.126.131
- RADIUS server: 192.168.126.130



Obrázok 2.1 Topológia siete laboratórnej úlohy.

Všetky virtuálne stroje musia byť prepojené v rovnakej virtuálnej podsieti, doporučené použiť sieťový režim: "**Host-Only**".





### Poznámka k vytvorenej topológii:

Pre emuláciu prístupového bodu (AP) použite virtuálny stroj, ktorý bol v predchádzajúcich laboratórnych úlohách použitý ako zariadenie „útočníka“. Príslušný VM je pripravený pre použitie nástroja `hostapd`, aby mohol funkčne zastúpiť prístupový bod simulovanej siete.

## 2.2. Zoznámenie sa s použitými nástrojmi

Prehľad základných príkazov pre jednotlivé používané nástroje

- **FreeRADIUS**

- inštalácia:

```
sudo apt install freeradius
```

- spustenie:

```
sudo systemctl start freeradius
```

- overenie stavu:

```
sudo systemctl status freeradius
```

- záznamy o udalostiach (logy) sú dostupné v súbore:

```
/var/log/freeradius/radius.log
```

- **Hostapd**

- inštalácia:

```
sudo apt install hostapd
```

- konfiguračný súbor obsahujúci nastavenie SSID, parametrov siete a parametrov súvisiacich s procesom autentizácie:

```
/etc/hostapd/hostapd.conf
```

- **Wpa\_supplicant**

- nastavenie služby `wpa_supplicant` sa uskutočňuje úpravou konfiguračného súboru:

```
/etc/wpa_supplicant/wpa_supplicant.conf
```

- príklad spustenia procesu autentizácie:

```
sudo wpa_supplicant -i eth0 -c  
/etc/wpa_supplicant/wpa_supplicant.conf -D wired
```

## 2.3. Postup pre vypracovanie laboratórnej úlohy

### A) Príprava prostredia

#### Spustenie virtuálnych strojov:


- Otvorte VMware Workstation Pro (umiestnený na ploche).
- Spustite postupne všetky tri virtuálne stroje (klient, AP, RADIUS server).
- Skontrolujte, že všetky VMs sú pripojené do rovnakej virtuálnej siete (napr. NAT alebo Host-Only). Uistite sa tiež v správnosti konfigurácie sieťových parametrov, overte priradenie IP adries.
- Prihláste sa do prostredia Kali Linux postupne na všetky VMs.

VM „klient“ – prihlasovacie údaje: **Username:** klient, **Password:** kali

VM „AP“ – prihlasovacie údaje: **Username:** kali, **Password:** kali

VM „RADIUS“ – prihlasovacie údaje: **Username:** server, **Password:** kali

#### Overenie sieťovej konektivity:

- Otvorte **terminál** (kliknite na ikonu terminálu  v záhlaví horného pracovného panelu alebo stlačte **Ctrl + Alt + T**).
- Na jednotlivých VMs si zobrazte priradené IP adresy (na rozhraní **eth0**) pomocou príkazu:

```
ip a
```

- Skontrolujte sieťovú konektivitu medzi všetkými strojmi pomocou **ping** z klienta na AP a server, a tiež v oboch smeroch medzi AP a RADIUS serverom.


```
ping <ip_adresa_zariadenia>
```

Ak prichádza odpoveď **ping echo reply** zo strany servera, sieťová komunikácia medzi zariadeniami funguje.

### B) Konfigurácia prístupového bodu

Aby mohol prístupový bod vytvorenej siete vystupovať v procese autentizácie klienta ako *authenticator* a sprostredkovať tak komunikáciu medzi klientom a autentizačným RADIUS serverom, je potrebné príslušný VM vhodne nakonfigurovať, pre tento účel bude použitý nástroj **hostapd**, pomocou ktorého je možné na „bežnom zariadení“ emulovať funkcionality plnohodnotného prístupového bodu.

## Použitie nástroja hostapd

- Na VM, ktorý bude plniť úlohu prístupového bodu, otvorte terminál kliknutím na ikonu  umiestnenú v záhlaví hlavného pracovného okna alebo v menu zvolíte **Applications > System Tools > Terminal**. Otvorí sa okno s príkazovým riadkom.
- Nainštalujte hostapd pomocou príkazu:

```
sudo apt install hostapd bridge-utils -y
```

- Aby mohol VM obstarávať funkcie AP v autentizačnom procese, je nutné uskutočniť patričné zmeny v konfigurácii, konkrétne v konfiguračnom súbore **/etc/hostapd/hostapd.conf**. Vytvorte alebo upravte súbor:

```
sudo nano /etc/hostapd/hostapd.conf
```

- Do súboru vložte nasledovnú konfiguráciu:

```
interface=eth0
driver=wired
ssid=EduLab
ieee8021x=1
auth_server_addr=192.168.126.130
auth_server_port=1812
auth_server_shared_secret=radiusheslo
```

- Po úprave konfigurácie stlačte kombináciu kláves **Ctrl + O** (uloženie), potvrdíte názov súboru klávesou **Enter** a následne ukončíte textový editor **nano** pomocou kombinácie **Ctrl + X**.
- Alternatívne môžete úpravu konfigurácie ukončiť stlačením **Ctrl + X**, následne pre potvrdenie uloženia vykonaných zmien stlačte **Y** (ako Yes/áno), a nakoniec potvrdíte stlačením **Enter**. Týmto spôsobom sa zmeny uložia a editor sa zároveň zatvorí.
- Následne spustíte službu hostapd pomocou príkazu:

```
sudo hostapd /etc/hostapd/hostapd.conf
```

V prípade úspešného spustenia služby sa v používanom terminálovom okne zobrazí stav **AP-ENABLED**.

### C) Konfigurácia RADIUS servera

Implementácia a následná konfigurácia vlastného autentizačného RADIUS servera na jednom z používaných VMs bude realizovaná pomocou nástroja FreeRADIUS.

#### Inštalácia nástroja FreeRADIUS

- Na serveri spustíte terminál.
- Po zobrazení okna s príkazovým riadkom najskôr skontrolujte, či je FreeRADIUS na VM nainštalovaný, príp. ho doinštalujte pomocou príkazu:

```
sudo apt update  
sudo apt install freeradius
```

#### Konfigurácia užívateľa

- Po inštalácii je potrebné na autentizačnom RADIUS serveri vytvoriť záznam o autentizačných údajoch klienta (žiadateľa):
  - Pre potreby editovania konfiguračných súborov nástroja FreeRADIUS bude potrebné **použiť privilegovaný režim**. Zadať príkaz:

```
sudo -i
```

ako heslo (*password*) zadajte: **kali**

- Otvorte súbor:

```
sudo nano /etc/freeradius/3.0/users
```

- a do súboru vložte nasledujúci záznam:

```
student Cleartext-Password := "tajneheslo"
```

#### Konfigurácia prístupového bodu (AP)

- V ďalšom kroku je potrebné definovať tiež parametre spojenia medzi RADIUS serverom a prístupovým bodom.
  - Otvorte súbor:

```
sudo nano /etc/freeradius/3.0/clients.conf
```

- a do súboru vložte nasledujúci záznam:

```
client ap {  
    ipaddr = 192.168.126.131  
    secret = radiusheslo  
}
```

## Spustenie služby

- Následne server reštartujte a spustite službu:

```
sudo systemctl start freeradius
```

- Po spustení služby FreeRADIUS sledujte logy (záznamy udalostí) na overenie správnosti použitých nastavení – potrebné logy sa automaticky zobrazia v termináli po zadaní príkazu:

```
sudo journalctl -u freeradius -f
```

Logy obsahujú dôležité informácie o priebehu autentizačného procesu, ako sú úspešné alebo neúspešné autentizačné pokusy, chyby v konfigurácii, odmietnuté žiadosti o prístup alebo problémy so sieťovou komunikáciou. Ak sa v logoch objavia chybové hlásenia (napr. neplatné prihlasovacie údaje, nesprávny kľúč, resp. *shared secret* medzi AP a RADIUS serverom apod.), je potrebné všetky zaznamenané chyby analyzovať a opraviť.

- Príklad ukážkového výstupu logu FreeRADIUS:

```
(0) Received Access-Request Id 45 from 192.168.126.128:54321 to 192.168.126.130:1812 length 150
(0) User-Name = "student"
(0) NAS-IP-Address = 192.168.126.128
(0) Called-Station-Id = "00-11-22-33-44-55:EduLab"
(0) Calling-Station-Id = "66-77-88-99-AA-BB"
(0) EAP-Message = 0x0200000d0173747564656e74
(0) Message-Authenticator = 0x1234567890abcdef1234567890abcdef
(0) # Executing section authorize from file /etc/freeradius/3.0/sites-enabled/default
(0)   authorize {
(0)     ok
(0)   } # authorize = ok
(0) Found Auth-Type EAP
(0) # Executing group from file /etc/freeradius/3.0/sites-enabled/default
(0)   authenticate {
(0)     eap: Peer sent EAP Response (code 2) ID 0 length 13
(0)     eap: No EAP Start, assuming it's an on-going EAP conversation
(0)   } # authenticate = ok
(0) Sent Access-Accept Id 45 from 192.168.126.130:1812 to 192.168.126.128:54321 length 80
```

## D) Konfigurácia klienta

Pre zaistenie bezpečného pripojenia k bezdrôtovej sieti a zaisteniu autentizácie pomocou protokolu IEEE 802.1X a komunikácie s autentizačným serverom RADIUS bude na strane klienta použitý nástroj `wpa_supplicant`.

### Použitie nástroja `wpa_supplicant`

- Na VM klienta spustite terminál.
- Nainštalujte `wpa_supplicant` a spustite:

```
sudo apt install wpasupplicant -y
```

- Vytvorte alebo upravte konfiguračný súbor služby `wpa_supplicant`:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

- Do súboru vložte nasledovnú konfiguráciu:

```
network={
    ssid="EduLab"
    key_mgmt=IEEE8021X
    eap=MD5
    identity="student"
    password="tajneheslo"
}
```

- Upravený súbor uložte pomocou **Ctrl + O**, potvrdíte stlačením klávesy **Enter** a nakoniec ukončíte pomocou **Ctrl + X**.
- Následne spustíte `wpa_supplicant`:

```
sudo wpa_supplicant -i eth0 -c
/etc/wpa_supplicant/wpa_supplicant.conf -D wired
```

Sledujte okno terminálu, v ktorom by sa mala zobrazíť hláška potvrdzujúca úspešnosť pripojenia.

### Sledovanie a analýza komunikácie vo Wiresharku

- Na klientovi (prípadne na AP) spustíte **Wireshark** pomocou príkazu:

```
sudo wireshark &
```

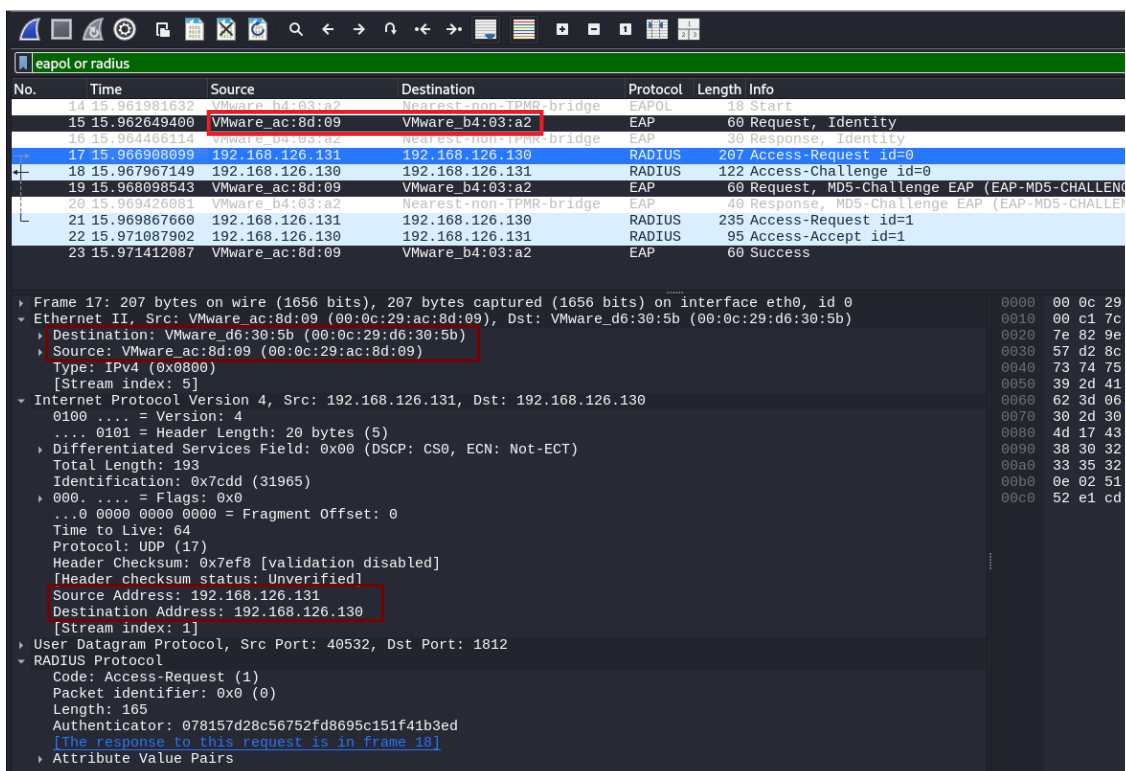
Prepínač `'sudo'` spustí nástroj Wireshark s oprávneniami administrátora, čo je nevyhnutné pre zachytávanie sieťovej prevádzky v Kali Linux.

- Zvoľte rozhranie **eth0** a následne spustíte zachytávanie dátovej komunikácie kliknutím na tlačidlo "**Start Capturing**" (ikona modrého žraloka) alebo voľba **Capture > Start**.
- Nastavte filter<sup>48</sup> pre zachytávanie paketov protokolov EAP a RADIUS. Do poľa pre filter v hornej časti Wiresharku zadajte

|               |
|---------------|
| eap or radius |
|---------------|

- Po spustení zachytávania komunikácie sa pokúste klientom pripojiť k sieti a sledujte prebiehajúci proces autentizácie. Zamerajte sa najmä na:
  - zahájenie komunikácie a výmenu EAP správ pri autentizácii EAP-MD5 (Identity Request/Response, Success/Failure);
  - v zachytenej komunikácii na AP analyzujte požiadavky Access-Request a odpovede Access-Accept/Reject v RADIUS protokole.
- Rozkliknite zachytené pakety a analyzujte ich podrobnosti:
  - v EAP paketoch si všimajte hodnoty ako "Identity", "Request", "Response",
  - hodnoty v poli **Identifier** v príslušných správach Request/Response,
  - v paketoch protokolu RADIUS sledujte poradie odoslaných správ a ich odpovedajúcu hodnotu v poli **Identifier**,
  - ďalej atribúty ako User-Name, NAS-IP-Address, Message-Authenticator,
  - overte zhodu medzi identitou v EAP a RADIUS správach;
  - a nakoniec sledujte, či sa v prípade úspešnej autentifikácie objaví správa "Access-Accept".
- Po dokončení procesu môžete zachytávanie prebiehajúcej komunikácie ukončiť kliknutím na "**Stop Capturing**". Ukážku zachytenej komunikácie môžete vidieť nižšie na obr. 2.2.

<sup>48</sup> Na strane klienta bude postačujúci filter **eap**, nakoľko k výmene správ protokolu RADIUS dochádza len v rámci komunikácie medzi AP a serverom.



Obrázok 2.2 Ukážka zachytenej komunikácie – výmena správ v priebehu autentizácie EAP-MD5.

## 2.4. Samostatná úloha

### E) Konfigurácia autentizačnej metódy EAP-TLS

V poslednej časti laboratórnej úlohy si vyskúšate implementáciu autentizačnej metódy EAP-TLS, pričom využijete v procese overenia identity klienta možnosť multifaktorovej autentizácie, a to v kombinácii hesla a certifikátu. Využijete Wireshark pre zachytenie a analýzu výmeny správ. Nakoniec porovnáte rozdiely medzi použitými metódami autentizácie na základe analýzy dátovej komunikácie – venujte pozornosť rozdielom v porovnaní s EAP-MD5.

**Cieľom vašej samostatnej práce bude implementovať autentizačnú metódu založenú na certifikátoch EAP-TLS, namiesto autentizačného mechanizmu EAP-MD5, a následne nakonfigurovať autentizačnú politiku využívajúcu multifaktorovú autentizáciu (heslo + certifikát). Vygenerujte vlastné certifikáty pre server a klienta a analyzujte rozdiely medzi jednotlivými metódami autentizácie na základe zachytených dátových paketov.**



### Užitočné príkazy:

- Pre automatické vytvorenie základnej certifikačnej autority (CA), generovanie certifikátu pre server aj kľúčov je možné použiť:

```
cd /etc/freeradius/3.0/certs/  
sudo ./bootstrap
```

- Vytvorené certifikáty sa nachádzajú v adresári `/etc/freeradius/3.0/certs/` – konkrétne:
  - `ca.pem` – certifikát certifikačnej autority
  - `server.pem` – certifikát servera
  - `server.key` – súkromný kľúč servera
  - `client.pem` – certifikát klienta
  - `client.key` – súkromný kľúč klienta
- úprava konfiguračného súboru `/etc/freeradius/3.0/mods-enabled/eap` na RADIUS serveri:

```
tls {  
    default_eap_type = mschapv2  
    copy_request_to_tunnel = yes  
    use_tunneled_reply = yes  
}
```

```
tls-config tls-common {  
    private_key_file = /etc/freeradius/3.0/certs/server.key  
    certificate_file = /etc/freeradius/3.0/certs/server.pem  
    ca_file = /etc/freeradius/3.0/certs/ca.pem  
}
```

- pre kopírovanie súborov (certifikátov) medzi zariadeniami je možné využiť `scp`
- uloženie certifikátov na strane klienta do adresára:  
`/etc/wpa_supplicant/certs/`
- taktiež bude potrebná vhodná úprava konfiguračného súboru na strane klienta, kedy v súbore `/etc/wpa_supplicant/wpa_supplicant.conf` správne nakonfigurujete použitie certifikátov:

```
network={
    ssid="TESTNET"
    key_mgmt=WPA-EAP
    eap=TTLS
    identity="client"
    password="heslo_klienta"
    ca_cert= <certifikat_cert_autority>
    client_cert= <certifikat_klienta>
    private_key= <sukromny_kluc_klienta>
    phase2="auth=MSCHAPV2"
}
```

### 3. Záver

V tejto laboratórnej úlohe ste sa zoznámili s možnosťami centralizovanej autentizácie klientov s využitím autentizačného servera RADIUS.

V praktickej časti ste vo vytvorenej virtuálnej sieti simulovali pomocou troch VMs **autentizačný proces využívajúci metódu EAP-MD5** a mali možnosť analyzovať priebeh celej komunikácie prístupujúceho klienta so vzdialeným autentizačným RADIUS serverom, ktorej sprostredkovateľom bol prístupový bod. V prostredí nástroja Wireshark ste analyzovali výmenu EAP správ, ktoré sú prenášané v priebehu autentizácie, a to jednak medzi klientom a prístupovým bodom prostredníctvom protokolu EAPoL na spojovej vrstve, a následne aplikačným protokolom RADIUS medzi prístupovým bodom a autentizačným serverom. Vašou samostatnou úlohou bolo následne **implementovať metódu EAP-TLS** využívajúcu pre overenie identity klienta certifikát s verejným kľúčom a uskutočniť jej porovnanie s predošlou použitou autentizačnou metódou.

#### 3.1. Kontrolné otázky

1. Ktoré tvrdenia správne popisujú fungovanie autentizačného mechanizmu podľa IEEE 802.1X?
  - A) Overenie identity prebieha ešte pred pridelením IP adresy klientovi
  - B) IEEE 802.1X je vhodný len pre bezdrôtové siete
  - C) Komunikácia medzi klientom a prístupovým bodom prebieha cez EAPoL
  - D) IEEE 802.1X zabezpečuje šifrovanie prenosu autentizačných údajov
2. Ktoré z nasledujúcich výrokov platia o úlohe prístupového bodu (*authenticator*) v architektúre IEEE 802.1X?
  - A) Posudzuje platnosť prihlasovacích údajov a vydáva rozhodnutie o prístupe
  - B) Vystupuje ako sprostredkovateľ komunikácie medzi klientom a autentizačným RADIUS serverom
  - C) S klientom komunikuje prostredníctvom protokolu EAPoL
  - D) Generuje prístupové heslá pre klientov v lokálnej sieti
3. Vyberte nesprávne tvrdenia o protokole RADIUS:
  - A) Komunikácia medzi klientom a RADIUS serverom prebieha prostredníctvom transportného protokolu UDP na porte 1812
  - B) RADIUS šifruje celé pakety pomocou TLS
  - C) RADIUS umožňuje centralizované overenie identity
  - D) RADIUS prenáša EAPoL správy ako súčasť autentizačných požiadaviek
4. Ktoré typy EAP metód využívajú digitálne certifikáty?
  - A) EAP-TLS
  - B) EAP-MD5
  - C) EAP-PEAP
  - D) EAP-TTLS

5. Ktoré tvrdenia o nástroji FreeRADIUS sú pravdivé?
  - A) Podporuje rôzne autentizačné metódy, vrátane EAP
  - B) podporuje použitie iba jednej autentizačnej metódy v jednom okamihu
  - C) Môže byť konfigurovaný na prácu s TLS
  - D) Nepodporuje použitie autentizačnej metódy EAP-MD5
6. Aké informácie sú prenášané v správe *Access-Request* protokolu RADIUS?
  - A) Užívateľské meno (User-Name)
  - B) Hash hesla alebo autentizačný token
  - C) ID a heslo užívateľa (klienta)
  - D) IP a MAC adresa klienta
7. Aký príkaz v Kali Linux slúži na spustenie služby FreeRADIUS?
  - A) sudo start radiusd
  - B) sudo systemctl start freeradius
  - C) radiusctl enable
  - D) freeradius --run
8. Ktoré EAP správy sú typicky súčasťou autentizačného procesu pri overovaní identity s využitím metódy EAP-MD5?
  - A) Identity Request
  - B) Identity Response
  - C) EAPOL Success/Failure
  - D) Access Request
9. Čo je typické pre komunikáciu medzi klientom a AP počas výmeny EAP správ?
  - A) Komunikácia prebieha pomocou protokolu EAPoL
  - B) Pakety sú prenášané v ethernetovom rámci na spojenej vrstve
  - C) Všetka komunikácia je šifrovaná pomocou TLS
  - D) Klient komunikuje priamo s autentizačným RADIUS serverom
10. Ktoré z nasledujúcich tvrdení o EAP over LAN (EAPoL) sú nepravdivé?
  - A) EAPoL sa používa na prenos EAP správ cez káblové alebo bezdrôtové LAN siete
  - B) EAPoL správy sú zapuzdrené priamo do IP paketov
  - C) EAPoL zaisťuje komunikáciu medzi klientom a AP
  - D) EAPoL šifruje všetky EAP správy pomocou TLS

## 4. Literatúra

- [1] Cloudradius: *Breaking Down the 802.1X Protocol*. [online]. 2024. [cit. 2025-04-20]. Dostupné z: <https://www.cloudradius.com/breaking-down-the-802-1x-protocol/>
- [2] IEEE Standard for Local and metropolitan area networks: *Port-Based Network Access Control* (802.1X). 2010. ISBN 978-0-7381-6204-2. [online]. [cit. 2025-04-20]. Dostupné z: <https://standards.ieee.org/ieee/802.1X/7345/>
- [3] Cisco: *Understanding 802.1X Port-Based Authentication*. [online]. Dostupné z: <https://www.cisco.com/c/en/us/support/docs/lan-switching/802-1x/8207-802-1x.html> [cit. 2025-04-20].
- [4] LinuxHint: *What is IEEE 802.1X?* [online]. 2024. [cit. 2025-04-20]. Dostupné z: [https://linuxhint.com/ieee\\_802\\_1x\\_protocol\\_intro/](https://linuxhint.com/ieee_802_1x_protocol_intro/)
- [5] Red Hat: *802.1X Authentication* [online]. 2024. [cit. 2025-04-20]. Dostupné z: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/6/html/deployment\\_guide/s1-wificonfig-8021x](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/s1-wificonfig-8021x)
- [6] ABOBA, B., BLUNK, L., VOLLBRECHT, J., CARLSON, J., LEVKOWETZ, H. *Extensible Authentication Protocol* (EAP). RFC 3748. [Internet Requests for Comments]. RFC Editor, 2004. [cit. 2025-04-20]. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc3748>
- [7] A Survey of Authentication Protocols in IEEE 802.1X Standard. In: *International Journal of Computer Applications*. [online]. [cit. 2025-04-20]. Dostupné z: <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.3918>
- [8] FADEL, Michael. *Authentication Protocols: Your Guide to the Basics* [online]. San Francisco: WorkOS, 2022. [cit. 2025-04-20]. Dostupné z: <https://workos.com/blog/authentication-protocols-your-guide-to-the-basics>
- [9] MORTÁGUAA, D. André ZÚQUETEB and Paulo SALVADOR. *Enhancing 802.1X authentication with identity providers using EAP-OAUTH and OAuth 2.0*. In: *Computer Networks*. 2024. s.1389–1286. [online]. [cit.2024-12-07]. Dostupné z: <https://doi.org/10.1016/j.comnet.2024.110337>
- [10] NAMAN, D. Mohammad ABDULWAHAB and Abbas IBRAHIM. *RADIUS Authentication on Unifi Enterprise System Controller using Zero-Handoff Roaming in Wireless Communication*. In: *JASTT*, vol.1. 2020. s.118–124. [online]. Dostupné z: [10.38094/jastt1427](https://doi.org/10.38094/jastt1427). [cit.2024-12-07]
- [11] FREERADIUS PROJECT. *FreeRADIUS Documentation*. [online]. 2023 [cit. 2025-04-18]. Dostupné z: <https://wiki.freeradius.org/>

## **Príloha F - Dokumentácia pre vyučujúceho k laboratórnej úlohe č. 8**

Laboratórna úloha č. 8

### **Autentizácia pomocou EAP a RADIUS**

# 1. Základné informácie k laboratórnej úlohe

Laboratórna úloha č. 8 je venovaná možnostiam realizácie centralizovanej autentizácie v počítačových sieťach pomocou protokolov IEEE 802.1X, EAP a RADIUS. Cieľom úlohy je, aby si študenti prakticky osvojili základné pojmy súvisiace s autentizačným rámcom EAP, konfiguráciou prístupového bodu, autentizačného servera a klienta, a aby boli schopní analyzovať priebeh celého procesu autentizácie na základe použitých autentizačných protokolov.

Úlohou študentov je nakonfigurovať sieť s využitím troch zariadení – klienta, prístupového bodu a RADIUS servera. Využili pritom nástroje ako FreeRADIUS, hostapd a wpa\_supplicant. Cieľom je teda vytvoriť úplnú funkčnú autentizačnú infraštruktúru a overiť správnosť autentizácie na základe rôznych nastavení.

## 2. Očakávané výstupy práce študentov

Praktická činnosť študentov v rámci tejto úlohy spočívala **v simulácii procesu autentizácie klienta prístupujúceho do vytvorenej siete**, a to s použitím nástrojov ako FreeRADIUS, hostapd a wpa\_supplicant. Úlohou študentov bolo uskutočniť vhodné úpravy v konfigurácii použitých VM tak, aby dosiahli ich požadovanú funkčnosť pre plnohodnotné zapojenie do autentizačného procesu, t.j. bolo potrebné nakonfigurovať **jeden virtuálny stroj ako prístupový bod** sprostredkujúci komunikáciu overovaného klienta a autentizačného servera, a tiež **d'alší VM ako samotný server RADIUS**.

Po uskutočnení laboratórnej úlohy by mali byť študenti schopní popísať priebeh autentizačného procesu a analyzovať odosielané EAP správy vďaka záznamu komunikácie vo Wiresharku. Správnosť realizácie úlohy je možné overiť na základe splnenia nasledujúcich bodov:

- Na prístupovom bode správne beží služba hostapd s nakonfigurovaným smerovaním autentizačných požiadaviek.
- FreeRADIUS server je správne nakonfigurovaný a prijíma EAP požiadavky od prístupového bodu.
- wpa\_supplicant na klientovi úspešne odosiela autentizačné správy (pozorovanie stavov "Authentication Success" v zachytenej komunikácii vo Wiresharku alebo v zobrazených logoch v termináli).
- Pomocou Wiresharku je zachytená komunikácia identifikujúca priebeh autentizácie klienta (pakety EAP, prípadne RADIUS správy Access-Request a Access-Accept).
- Výsledkom procesu je úspešná autentizácia klienta (logy potvrdzujú úspešný EAP *handshake* medzi klientom a RADIUS serverom).

## 2.1. Riešenie samostatnej úlohy

V rámci samostatnej úlohy majú študenti za úlohu implementovať použitie autentizačnej metódy EAP-TTLS a multifaktorovej autentizácie klienta s využitím kombinácie hesla a certifikátu. Pre konfigurácie autentizácie založenej na využití certifikátu bude potrebné, aby študenti vytvorili na RADIUS serveri vlastnú certifikačnú autoritu a následne vygenerovali dvojicu certifikátov:

- certifikát pre autentizačný server (resp. pre FreeRADIUS),
- certifikát klienta.

Následne vhodnou zmenou konfigurácie na serveri i klientovi zadefinujú použitie autentizačnej metódy EAP-TTLS, pričom je dôležité dbať na správnosť použitia vygenerovaných certifikátov. Taktiež je potrebné, aby študenti realizovali prenos certifikátov a súkromného kľúča klienta na jeho zariadenie (napr. pomocou scp).

Ak je `wpa_supplicant` správne nakonfigurovaný, tak bude úspešné pripojenie indikované správou: **"CTRL-EVENT-CONNECTED"**. Vo Wiresharku bude správne zachytená komunikácia, v ktorej bude možné pozorovať správy protokolu TLS prenášané vytvoreným šifrovaným tunelom.

## 2.2. Odpovede na kontrolné otázky

1. Ktoré tvrdenia správne popisujú fungovanie autentizačného mechanizmu podľa IEEE 802.1X?
  - A) Overenie identity prebieha ešte pred pridelením IP adresy klientovi ☒
  - B) IEEE 802.1X je vhodný len pre bezdrôtové siete
  - C) Komunikácia medzi klientom a prístupovým bodom prebieha cez EAPoL ☒
  - D) IEEE 802.1X zabezpečuje šifrovanie prenosu autentizačných údajov
2. Ktoré z nasledujúcich výrokov platia o úlohe prístupového bodu (*authenticator*) v architektúre IEEE 802.1X?
  - A) Posudzuje platnosť prihlasovacích údajov a vydáva rozhodnutie o prístupe
  - B) Vystupuje ako sprostredkovateľ komunikácie medzi klientom a autentizačným RADIUS serverom ☒
  - C) S klientom komunikuje prostredníctvom protokolu EAPoL ☒
  - D) Generuje prístupové heslá pre klientov v lokálnej sieti
3. Vyberte nesprávne tvrdenia o protokole RADIUS:
  - A) Komunikácia medzi klientom a RADIUS serverom prebieha prostredníctvom transportného protokolu UDP na porte 1812
  - B) RADIUS šifruje celé pakety pomocou TLS ☒
  - C) RADIUS umožňuje centralizované overenie identity
  - D) RADIUS prenáša EAPoL správy ako súčasť autentizačných požiadaviek ☒



4. Ktoré typy EAP metód využívajú digitálne certifikáty?
- A) EAP-TLS ☒
  - B) EAP-MD5
  - C) EAP-PEAP
  - D) EAP-TTLS ☒
5. Ktoré tvrdenia o nástroji FreeRADIUS sú pravdivé?
- A) Podporuje rôzne autentizačné metódy, vrátane EAP ☒
  - B) podporuje použitie iba jednej autentizačnú metódy v jednom okamihu
  - C) Môže byť konfigurovaný na prácu s TLS ☒
  - D) Nepodporuje použitie autentizačnej metódy EAP-MD5
6. Aké informácie sú prenášané v správe *Access-Request* protokolu RADIUS?
- A) Užívateľské meno (User-Name) ☒
  - B) Hash hesla alebo autentizačný token ☒
  - C) ID a heslo užívateľa (klienta)
  - D) IP a MAC adresa klienta
7. Aký príkaz v Kali Linux slúži na spustenie služby FreeRADIUS?
- A) sudo start radiusd
  - B) sudo systemctl start freeradius ☒
  - C) radiusctl enable
  - D) freeradius --run
8. Ktoré EAP správy sú typicky súčasťou autentizačného procesu pri overovaní identity s využitím metódy EAP-MD5?
- A) Identity Request ☒
  - B) Identity Response ☒
  - C) EAPOL Success/Failure ☒
  - D) Access Request
9. Čo je typické pre komunikáciu medzi klientom a AP počas výmeny EAP správ?
- A) Komunikácia prebieha pomocou protokolu EAPoL ☒
  - B) Pakety sú prenášané v ethernetovom rámci na spojovej vrstve ☒
  - C) Všetka komunikácia je šifrovaná pomocou TLS
  - D) Klient komunikuje priamo s autentizačným RADIUS serverom
10. Ktoré z nasledujúcich tvrdení o EAP over LAN (EAPoL) sú nepravdivé?
- A) EAPoL sa používa na prenos EAP správ cez káblové alebo bezdrôtové LAN siete
  - B) EAPoL správy sú zapuzdrené priamo do IP paketov ☒
  - C) EAPoL zaisťuje komunikáciu medzi klientom a AP
  - D) EAPoL šifruje všetky EAP správy pomocou TLS ☒

## 2.3. Doplnujúce otázky

Nižšie uvedené otázky môžu byť využité pri kontrole výstupov samostatnej práce študentom s cieľom overiť, či skutočne porozumeli riešenej problematike v praktickej časti laboratórnej úlohy.

**1. Aké sú výhody používania RADIUS servera v porovnaní s metódami umožňujúce lokálne overenie identity prístupujúceho užívateľa?**

- Centralizované riadenie prístupu, možnosť správy veľkého množstva užívateľov, vyššia bezpečnosť a jednoduchšia správa prístupových politík.

**2. Vysvetlite úlohu protokolu IEEE 802.1X v procese autentizácie.**

- Riadi prístup do siete na základe overenia identity užívateľa pomocou externého autentizačného servera (napr. RADIUS).

**3. Aké sú hlavné komponenty v architektúre 802.1X?**

- Klient, resp. žiadateľ o overenie (*supplicant*), prístupový bod (*authenticator*), autentizačný server RADIUS.

**4. Aké typy autentizačných metód podporuje RADIUS?**

- Napr. EAP-TLS, EAP-PEAP, EAP-TTLS, EAP-MD5, PAP, CHAP.

**5. Aké sú možné bezpečnostné riziká súvisiace s používaním protokolov EAP a RADIUS v otvorených sieťach? Uved'te príklad opatrení, pomocou ktorých by bolo možné tieto riziká zmierniť.**

- Možnosť odpočúvania prebiehajúcej komunikácie alebo *spoofingu*, riziko MitM útokov, zneužitie slabých autentizačných mechanizmov. Riešením je používať pokročilé metódy pre šifrovanie komunikácie (napr. implementácia protokolu TLS).

**6. Pomocou akého príkazu je možné spustiť RADIUS server na Kali Linux?**

- `sudo systemctl start freeradius`

**7. Aké filtre by ste použili vo Wiresharku pre zobrazenie EAP správ?**

- `eap` alebo `radius`.

## **Príloha G - Text laboratórnej úlohy č. 11**

Laboratórna úloha č. 11

### **ANONYMIZAČNÉ SIETE**

## 0. Úvod k laboratórnej úlohe

Cieľom laboratórnej úlohy je oboznámiť študentov s použitím anonymizačných sietí, ich významom v oblasti ochrany súkromia a bezpečnosti online komunikácie a ozrejmiť im základné princípy ich fungovania, pričom hlavná pozornosť bude venovaná **anonymizačnej sieti Tor** (z angl. *The Onion Routing*), ktorá využíva pre zaistenie dôvernosti, anonymity užívateľov a utajenia komunikácie medzi účastníkmi techniku tzv. cibuľového smerovania (*onion routing*). V rámci laboratórnej úlohy budú preto vysvetlené základné princípy fungovania a účel použitia anonymizačných sietí, ďalej bude predstavený hlavný koncept celosvetovo známej anonymizačnej siete Tor a taktiež bude objasnené použitie vrstvomého smerovania v tomto type sietí.

V praktickej časti sa študenti budú venovať inštalácii a konfigurácii nástroja Tor v prostredí Kali Linux, prehliadaniu internetu prostredníctvom Tor siete a analýze sieťovej komunikácie pomocou nástroja Wireshark. Študenti tak nadobudnú praktické zručnosti v oblasti anonymizácie internetovej komunikácie a naučia sa analyzovať tok dát v prostredí anonymizačných sietí.

### Požiadavky pre vypracovanie úlohy:

- software: VMware Workstation Player pre virtualizáciu staníc,
- virtuálne stroje: dva, resp. tri virtuálne stroje s Kali Linux.

## 1. Teoretický úvod

V tejto laboratórnej úlohe venovanej problematike anonymizačných sietí sa zoznámite so základnými princípmi fungovania anonymizačných sietí, ich základnou štruktúrou a tiež s procesom tzv. viacvrstvomého („cibuľového“) šifrovania, ktoré je typické práve pre dosiahnutie anonymity klientov, resp. komunikujúcich koncových zariadení prostredníctvom anonymizačných sietí.

### 1.1. Anonymizačné siete

**Anonymizačné siete** predstavujú pokročilejšie bezpečnostné technológie navrhnuté a používané za účelom **ochrany identity používateľov a ich súkromia** v digitálnom prostredí dnešných počítačových sietí. Jedná sa o špeciálne typy sietí navrhnuté za účelom ochrany identity a polohy používateľov, ktoré umožňujú prostredníctvom šifrovania informácií obsiahnutých v prenášaných dátových jednotkách **zaistiť anonymné prehliadanie internetu**, resp. komunikáciu, a tým i **ukrytie identity** prístupujúcich užívateľov (resp. zariadení) naprieč rozsiahlym konglomerátom vzájomne prepojených sietí. Ich cieľom je minimalizovať možnosť sledovania zdrojovej IP adresy, lokalizácie či iných identifikačných údajov používateľa. Medzi najznámejšie

anonymizačné siete patria napríklad **Tor**, **I2P**, **Freenet** a i. V tejto úlohe bude najväčšia pozornosť venovaná primárne anonymizačnej sieti Tor (*The Onion Router*).

## 1.2. Tor (The Onion Router)

Tor predstavuje projekt vyvíjaný s cieľom poskytnúť užívateľom možnosti anonymného vystupovania a komunikácie v digitálnom prostredí internetu. Je založený na **princípe tzv. „cibuľového“ smerovania (*onion routing*)**, kde komunikácia medzi klientom a cieľovým serverom prebieha cez sériu náhodne vybraných sprostredkovateľov nazývaných „Tor relé“ (typicky sa jedná o medziľahlé smerovače na prenosovej trase). Každý takýto medziľahlý uzol pozná len bezprostredne predchádzajúci a ďalší nasledujúci bod komunikácie, vďaka čomu je možné zamedziť úplné sledovanie priebehu celej komunikácie, a to vrátane informácií o koncových bodoch. Anonymizačná sieť Tor je navrhnutá tak, aby bolo možné:

- zabezpečiť anonymitu klienta voči cieľovej službe,
- napr. v prípade tzv. skrytých služieb zabezpečiť taktiež anonymitu cieľa voči klientovi,
- zamedziť tretím stranám (napr. poskytovateľom internetového pripojenia, prevádzkovateľom Wi-Fi sietí apod.) získať prehľad o tom, aké stránky používateľ navštevuje alebo s kým pri prístupe do siete komunikuje.

### Architektúra Tor siete

Pre ďalší popis a vysvetlenie princípov tzv. „cibuľového“ smerovania sú dôležité dva základné pojmy: **vrstvy a uzly**. Pri smerovaní dátových jednotiek (tzv. buniek) smerom k adresátovi s využitím „cibuľového“ smerovania každý medziľahlý uzol podieľajúci sa na komunikácii (resp. smerovaní) vždy dešifruje len jednu „vrstvu“ aplikovaného šifrovania. Po dešifrovaní, t. j. odstránení vonkajšej vrstvy sa odhalí ďalšia nasledujúca adresa na trase k príjemcovi dátovej jednotky, iné však zostávajú stále chránené, utajené šifrovaním, čo predstavuje základný mechanizmus pre zaistenie anonymity užívateľov (resp. jednotlivých uzlov, zariadení).

Typická sieť Tor pozostáva z nasledujúcich **základných komponentov (uzlov)**:

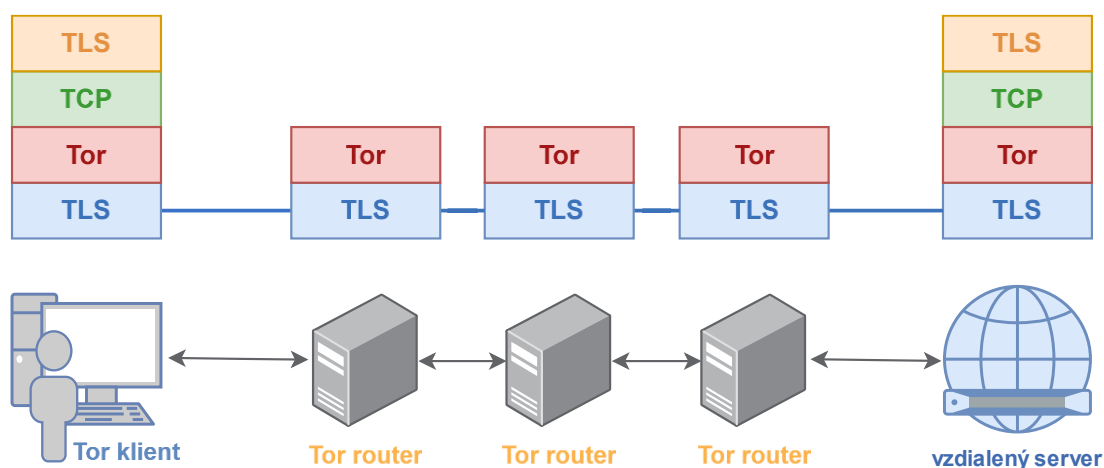
- **Tor klient** – aplikačné rozhranie na strane používateľa využívajúceho možnosti anonymného prehliadania, typicky v podobe Tor prehliadača alebo systémového démona *tor*.
- **Tor smerovače (*Tor relays, onion routers*)** – medziľahlé uzly (spravidla smerovače), ktorých úlohou je smerovanie dátových jednotiek (buniek) v Tor sieti. Rozlišujeme:
  - **vstupný uzol (*Entry Node*)** – prvý bod (uzol) v komunikačnej sieti, kde po prvýkrát dochádza k šifrovaniu prenášaných dát;

- **relé uzly (*Relay Nodes*)** – medziľahlé<sup>49</sup> uzly, ktoré prenášajú šifrované dáta, resp. postupne ich (de)šifrujú pomocou svojich kľúčov pre šifrovanie, resp. dešifrovanie;
- **výstupný uzol (*Exit Node*)** – posledný bod v sieti, kde sú dáta dešifrované a odoslané na cieľovú adresu (prijemcovi);
- **Adresárové servery** – centrálné uzly, ktoré spravujú zoznam dôveryhodných Tor smerovačov a poskytujú informácie o dostupných uzloch ostatným prvkom v Tor sieti.

### Princíp šifrovania v anonymizačnej sieti

Mechanizmy siete Tor používané pre zaistenie anonymity používateľov sú aplikované na úrovni sieťovej vrstvy, ako ju poznáme zo sieťového modelu TCP/IP. Celková architektúra siete Tor pozostáva z nasledujúcich vrstiev:

- **Linková vrstva** – je realizovaná TLS spojmami medzi jednotlivými prvkami Tor siete, kde protokol TLS zohráva zásadnú úlohu pri autentizácii prvkov a pri ochrane dôvernosti a integrity prenosov. Každý individuálny úsek komunikácie medzi dvoma susednými uzlami (napr. klient ↔ OR1) je zabezpečený samostatným TLS tunelom.
- **Sieťová vrstva** – je realizovaná protokolom Tor, ktorý zaisťuje vytváranie a následný prenos a smerovanie buniek, vrátane ich šifrovania.
- **Transportná vrstva** – za účelom spoľahlivosti prenosu a možnosti šifrovania pomocou TLS využíva spojoovo orientovaný a spoľahlivý protokol TCP.
- **Aplikačná vrstva** – pozostáva z klientskych aplikácií na strane koncového používateľa využívajúcich transportný protokol TCP (napr. HTTP, FTP).



Obrázok 1.1 Vrstvová architektúra siete Tor<sup>50</sup>.

<sup>49</sup> Tieto medziľahlé *relay* uzly sú sprostredkovateľmi prenosu šifrovanej dátovej komunikácie medzi odosielateľom a prijemcom využívajúcich anonymizačnú sieť Tor.

<sup>50</sup> Prevzaté z oficiálnych výučbových materiálov k prednáškam predmetu MPC-NSB – vypracoval garant predmetu a prednášajúcim doc. Karel Burda, CSs. (viď tiež E-learning predmetu).

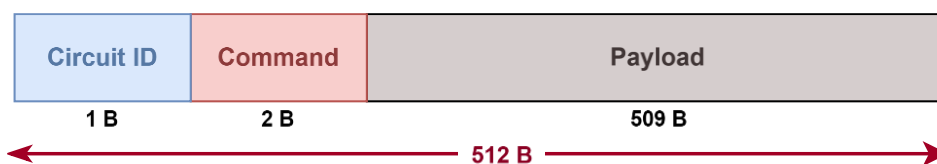
V sieti Tor prebieha komunikácia vo forme **buniek (cells)**, ktoré predstavujú základnú jednotku prenosu dát medzi klientom a uzlami v sieti. Každá bunka má **fixnú veľkosť 512 bajtov**. Nemenná veľkosť bunky umožňuje minimalizovať možnosť analýzy komunikácie na základe veľkosti prenášaných dátových jednotiek. Všetky správy, bez ohľadu na ich skutočný obsah alebo dĺžku, sú preto zapuzdrené do rovnako veľkých buniek, čím sa znižuje riziko, že by pozorovateľ (typicky útočník) mohol identifikovať vzorce komunikácie alebo konkrétny typ dát.

Tor bunky možno rozdeliť podľa účelu ich použitia na viacero typov. Napríklad:

- **Bunky typu CREATE a CREATED** sa používajú pri vytváraní šifrovaného okruhu medzi klientom a jednotlivými uzlami (resp. medzi dvojicou susedných uzlov v Tor sieti).
- **Bunky označené ako RELAY, RELAY\_EXTEND, RELAY\_DATA** apod. slúžia k prenosu šifrovaných dát cez jednotlivé uzly v rámci okruhu.
- Na ukončenie okruhu sa používa **bunka typu DESTROY**, ktorá signalizuje, že daný komunikačný okruh má byť okamžite zrušený a všetky naviazané šifrovacie kľúče zneplatnené.

Každá Tor bunka má pevne daný formát a obsahuje viacero polí, ktoré slúžia na identifikáciu, správu prenosu a prenos samotných dát. Presný obsah bunky sa môže mierne líšiť v závislosti od jej typu. Štruktúra štandardnej Tor bunky je znázornená na obr. 1.2, význam príslušných polí je nasledovný:

- **Circuit ID** – identifikátor okruhu, ku ktorému bunka patrí. Umožňuje multiplexovanie viacerých okruhov cez jedno TCP spojenie.
- **Command** – určuje typ bunky (napr. CREATE, RELAY, DESTROY, atď.).
- **Payload** – telo bunky, ktoré je počas prenosu v Tor sieti šifrované. Jeho obsah sa líši podľa konkrétneho typu bunky.



Obrázok 1.2 Schematické znázornenie štruktúry Tor bunky<sup>51</sup>.

<sup>51</sup> Prevzaté z oficiálnych výučbových materiálov k prednáškam predmetu MPC-NSB – vypracoval garant predmetu a prednášajúcim doc. Karel Burda, CSs. (viď tiež E-learning predmetu).

## Vytváranie okruhov a princíp cibulového smerovania v sieti Tor

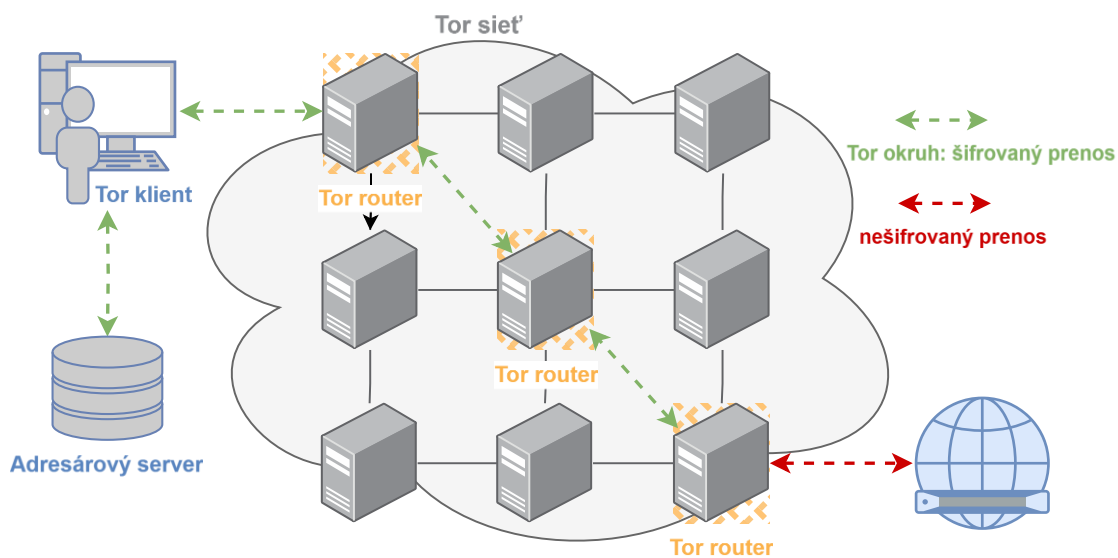
Kľúčovým mechanizmom zabezpečenia anonymity je vytváranie tzv. okruhov (*circuits*) a používanie techniky viacvrstvého šifrovania, známeho aj ako *onion routing*.

Tor vytvára medzi klientom a cieľovým serverom viacvrstvový (tzv. „cibuľový“<sup>52</sup>) šifrovaný kanál, prostredníctvom ktorého sú údaje, resp. dátové pakety presmerované cez niekoľko náhodne vybraných uzlov v Tor sieti. Jednotlivé uzly poznajú len odosielateľa a prijímateľa ich časti trasy, nemajú znalosť o iných uzloch, ktoré boli alebo budú zapojené do celej komunikácie potrebnej pre zaistenie doručenia danej dátovej jednotky (bunky) od jej odosielateľa až k vybranému príjemcovi, čo zabezpečuje anonymitu.

Proces vytvárania okruhu prebieha v niekoľkých etapách:

1. **Výber uzlov:** klient si zo zverejneného zoznamu vyberie náhodne vhodné Tor uzly. Výber prebieha podľa špecifických pravidiel – napríklad vstupné uzly musia byť stabilné a dôveryhodné.
2. **Dohoda na kľúčoch:** pomocou protokolu podobného Diffie-Hellmanovej výmene kľúčov si klient postupne vytvorí s každým uzlom samostatný šifrovací kľúč. Všetky tieto výmeny prebiehajú cez vstupný uzol, pričom klient nadväzuje spojenie s ďalšími uzlami „cez“ predchádzajúce (šifrovane).
3. **Postupné rozšírenie okruhu:** najskôr sa vytvorí šifrované spojenie klienta so vstupným uzlom (pomocou CREATE a CREATED buniek), následne sa cez tento uzol vytvorí šifrovaný tunel postupne ku všetkým nasledujúcim uzlom až nakoniec k výstupnému uzlu.

Takto vytvorený okruh slúži ako trasa, po ktorej sú ďalej prenášané šifrované dáta.



Obrázok 1.3 Komponenty Tor siete a znázornenie vytvoreného okruhu<sup>53</sup>.

<sup>52</sup> Označenie je prevzaté z prekladu angl. slova *onion*, ktoré značí cibuľu. Princíp šifrovania, resp. dešifrovania paketov odosielaných cez Tor sieť sa podobá vrstveniu cibule – a práve na základe tejto podobnosti bol vytvorený aj jej názov.

<sup>53</sup> Prevzaté z [5].



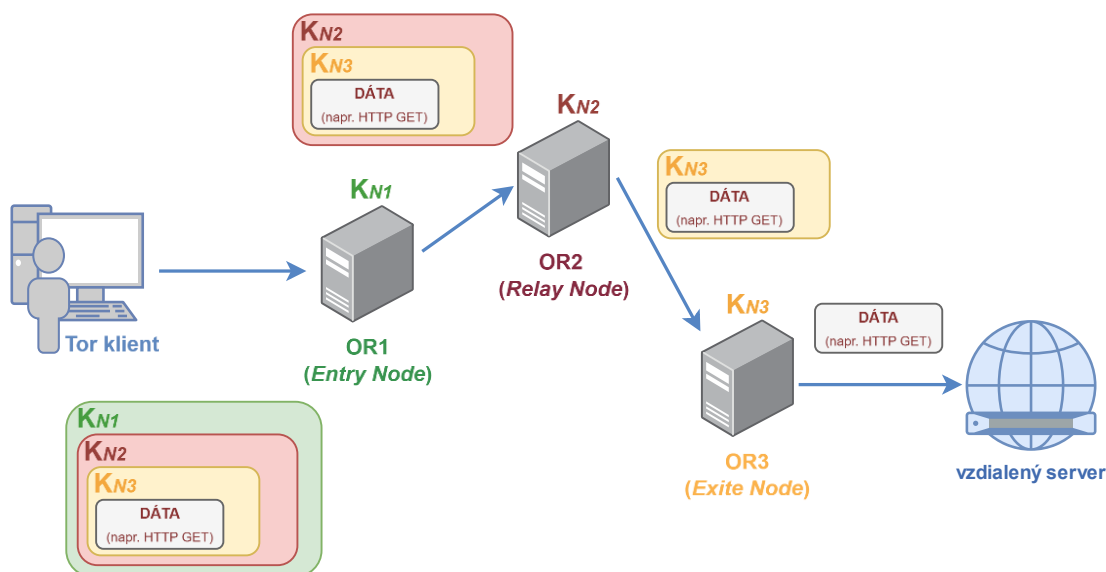
Po vytvorení kompletného okruhu od odosielateľa až cieľovému príjemcovi je možné zahájiť dátový prenos, ktorý je podrobený „cibuľovému smerovaniu“. Mechanizmus tohto viacvrstvého šifrovania funguje nasledovne:

1. Klient si najprv pripraví dátovú jednotku (napr. požiadavku HTTP), ktorú chce doručiť na cieľový server.
2. Túto jednotku následne **viacnásobne šifruje** – každá vrstva je určená jednému uzlu v poradí od výstupného po vstupný:
  - najprv pre výstupný uzol (vnútorná vrstva),
  - potom pre stredný uzol,
  - napokon pre vstupný uzol (vonkajšia vrstva).
3. Keď šifrovaná správa dorazí do vstupného uzla, ten v procese dešifrovania odstráni iba svoju vrstvu (vonkajšiu) a odošle jej obsah, ktorý je zašifrovaný pomocou kľúča pre ďalší medzilahlý uzol v poradí, ďalej smerom k ďalšiemu uzlu vo vytvorenom okruhu.
4. Nasledujúci uzol opäť odstráni svoju vrstvu a odošle zašifrované dáta ďalej.
5. Proces sa opakuje, až kým bunka nedorazí k výstupnému uzlu. Ten dešifruje poslednú vrstvu a odosiela pôvodnú požiadavku vytvorenú klientom, resp. odosielateľom na cieľový server na internete (napr. webovú stránku).
6. Pri spätnej odpovedi sa postupuje obdobne, akurát sa jednotlivé šifrovacie kľúče aplikujú pri vytváraní vrstiev v opačnom poradí – výstupný uzol odpoveď zašifruje a pošle späť cez ten istý okruh, pričom každý uzol dešifruje iba svoju časť, až sa odpoveď dostane k pôvodnému klientovi.

Aplikácia viacvrstvého šifrovania na prenášané dáta je schematicky znázornená na obr. 1.4. Popísaný mechanizmus zaručí, že žiaden z uzlov nemá úplnú znalosť o celej komunikácii. Vstupný uzol pozná IP klienta, ale nie cieľový server. Výstupný uzol naopak pozná cieľ, ale nie klienta. Všetky medzilahlé uzly poznajú len svojich bezprostredných susedov.

Použitie anonymizačných sietí prináša množstvo výhod, medzi ktoré nepochybne patrí ukrytie IP adresy používateľov, čo je tiež jeden zo základných predpokladov pre ochranu pred nežiadúcim sledovaním a profilovaním. Na druhej strane, prenos dát prostredníctvom anonymizačnej siete môže byť znateľne pomalší, nakoľko nutnosť prenosu dát cez viaceré medzilahlé uzly, kedy dochádza navyše k šifrovaniu (resp. dešifrovaniu) týchto dát, môže mať za následok výsledné spomalenie internetového pripojenia. Medzi ďalšie nevýhody a riziká anonymizačných sietí možno zaradiť i možnosť kompromitácie výstupného uzla smerom k príjemcovi dát (*exit node*) a tiež skutočnosť, že použitie anonymizačných sietí nezaručí kompletnú ochranu pred všetkými formami sledovania komunikácie v počítačových sieťach, akou môže byť napr. sledovanie časových korelácií medzi dátovými prenosmi s ich následnou analýzou, a tak isto neposkytuje ochranu pred inými typmi sieťových útokov či útokov na koncové zariadenia, napr. prostredníctvom škodlivého kódu (malware) na strane užívateľa apod.

Viac informácií o koncepte anonymizačných sietí a o samotnej sieti Tor je možné nájsť v publikáciách [1], [2], [3], [4].



Obrázok 1.4 Schematické znázornenie vrstveného šifrovania v Tor sieti<sup>54</sup>.

### 1.3. Použité nástroje

#### Tor v Kali Linux

Tor predstavuje jednoduchý softvérový nástroj umožňujúci **anonymné prehliadanie internetu cez Tor sieť**. Vytvorenie, resp. simuláciu vlastnej anonymizačnej siete založenej na využití služby Tor je možné uskutočniť i v prostredí systému Kali Linux, a to jej inštaláciou priamo prostredníctvom príkazového riadku (terminálu) pomocou príkazov:

```
sudo apt update
sudo apt install tor
```

Po úspešnej inštalácii nasleduje spustenie služby Tor:

```
sudo systemctl start tor
sudo systemctl enable tor
```

Overenie, či je služba aktívna, je možné pomocou príkazu:

```
sudo systemctl status tor
```

<sup>54</sup> Prevzaté z [6].

Pre overenie pripojenia na Tor sieť možno použiť napr. nižšie uvedený príkaz:

```
curl --socks5-hostname 127.0.0.1:9050  
https://check.torproject.org/
```

Uvedený príkaz spustí odoslanie jednoduchkej HTTP GET požiadavky na stránku <https://check.torproject.org><sup>55</sup> cez vytvorenú Tor sieť. Pozn.: TCP port 9050 je predvolený pre Tor klienta, ktorý beží na Kali Linux. Sieť Tor v tomto prípade funguje ako proxy server, resp. sprostredkovateľ komunikácie medzi klientom na vašom zariadení a dotazovaným cieľovým serverom – všetka komunikácia je teda presmerovaná práve cez Tor sieť.

Po odoslaní požiadavky server **check.torproject.org** analyzuje vašu IP adresu a vráti odpoveď, či komunikácia prebieha cez Tor, alebo nie. Pokiaľ je použitie siete Tor správne nastavené, na výstupe v termináli sa zobrazí hláška:

**"Congratulations. This browser is configured to use Tor."**

## Tor Browser

Jedná sa o upravenú verziu webového prehliadača nakonfigurovanú pre anonymné používanie využívajúc práve sieť Tor pre **zabezpečenie súkromia a anonymity jeho užívateľov** v online prostredí. Tor Browser je navrhnutý tak, aby bolo s jeho použitím možné ukrytie nielen samotnej identity užívateľa, ale tiež jeho polohy a aktivity na internete.

**Tor Browser** realizuje šifrovanie prenášaných užívateľských dát v niekoľkých vrstvách. Dátové prenosy sú kompletne šifrované a odosielané cez Tor sieť, ktorá pozostáva z veľkého množstva (typicky tisícov) *relay* uzlov sprostredkujúcich prenos zabezpečenej šifrovanej komunikácie. Každý *relay* uzol na ceste prenosu dát smerom k príjemcovi vždy dešifruje len jednu (vonkajšiu) vrstvu, čím nikdy nezíska úplnú, kompletnú informáciu o danom prenose, vďaka čomu anonymita komunikujúcich strán zostáva zachovaná.

## Wireshark

Wireshark je sieťový analyzátor, ktorý umožňuje sledovať dátové prenosy. V prípade Tor je možné zachytiť šifrované pakety a analyzovať ich štruktúru, no obsah zostáva chránený šifrovaním.

Použitie nástroja Wireshark ste si prakticky vyskúšali už v rámci niekoľkých predošlých laboratórnych úloh, takže jeho bližší popis nebude už ďalej podrobne uvádzaný.

---

<sup>55</sup> Viac o filozofii a štruktúre projektu Tor, ktorého cieľom je poskytovanie špecializovaného Tor Browser prehliadača pre anonymné prehliadanie, je možné nájsť na oficiálnych stránkach: [7].

## 2. Praktická časť

V rámci praktickej časti úlohy si vyskúšate **anonymné prehliadanie prostredníctvom anonymizačného webového prehliadača Tor Browser**. Následne budete analyzovať zachytený tok dát v anonymizačnej sieti pomocou nástroja Wireshark, na základe čoho získate prehľad o výhodách, nevýhodách, a rizikách spojených s použitím prostriedkov pre anonymizáciu v dnešných počítačových sieťach.

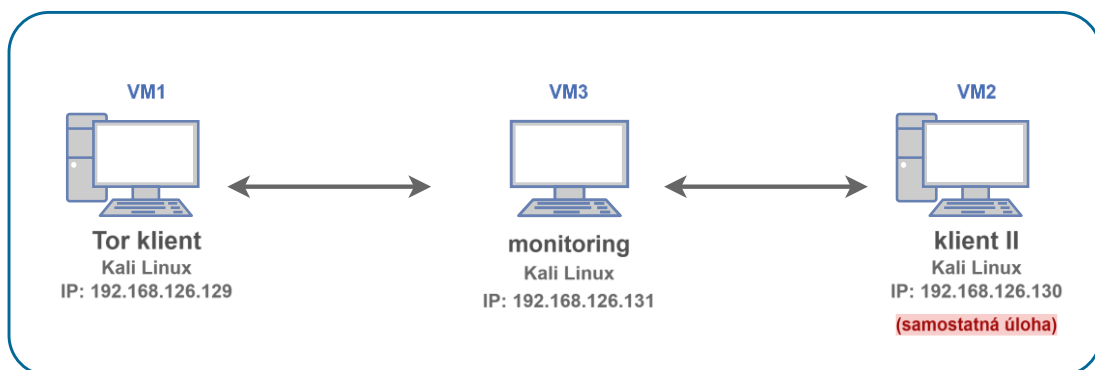
### 2.1. Topológia virtuálnej siete a nastavenie virtuálnych strojov

Vytvorená sieť bude pozostávať z troch virtuálnych strojov:

- **klient pre anonymné prehliadanie** a prístup na internet cez Tor sieť,
- **klient II** – simulácia klasického pripojenia, t. j. bežné pripojenie bez použitia Tor (využitie v časti **Samostatná úloha**),
- **monitorovacie zariadenie**, ktoré bude slúžiť pre účely sledovania a následnej analýzy prebiehajúcej komunikácie v sieti.

Sieťová konfigurácia:

- Tor klient (VM1): 192.168.126.129
- klient II (VM2): 192.168.126.130
- monitorovacie zariadenie (VM3): 192.168.126.131



Obrázok 2.1 Topológia siete laboratórnej úlohy.

### 2.2. Zoznámenie sa s použitými nástrojmi

Prehľad základných príkazov pre jednotlivé používané nástroje

- Uvedenie základných príkazov pre prácu so službou Tor na klientskom zariadení a pre inštaláciu a následné **použitie prehliadača Tor Browser** pre anonymné prehliadanie na internete bolo súčasťou teoretického úvodu, a z toho dôvodu ich opakovaný prehľad nie je ďalej uvádzaný. Všetky potrebné príkazy budú uvedené následne v praktickej časti v jednotlivých krokoch pre vypracovanie laboratórnej úlohy.

## Použitie Wiresharku pre analýzu komunikáciu cez sieť Tor

- V rámci analýzy zaznamenatej komunikácie je vhodné použiť filter:

```
tcp.port == 9050
```

Použitie uvedeného filtra zaručí zobrazenie TCP komunikácie na príslušnom porte, t. j. 9050 – čo je port využitý na strane klienta pre komunikáciu so SOCKS proxy pre jej ďalšie presmerovanie cez anonymizovanú Tor sieť.

## 2.3. Postup pre vypracovanie laboratórnej úlohy

### A) Príprava prostredia

#### Spustenie virtuálnych strojov:

- Otvorte VMware Workstation Pro (umiestnený na ploche).
- Spustite postupne všetky tri virtuálne stroje s Kali Linux.
- Uistite sa, že všetky VMs sú pripojené do rovnakej virtuálnej siete (napr. režim Host-only alebo NAT).
- Prihláste sa do prostredia Kali Linux na jednotlivých VMs.

VM „Tor klient“ – prihlasovacie údaje: **Username:** klient, **Password:** kali

VM „Klient II“ – prihlasovacie údaje: **Username:** server, **Password:** kali

VM „monitoring“ – prihlasovacie údaje: **Username:** kali, **Password:** kali

- Skontrolujte sieťovú konektivitu medzi strojmi (pomocou príkazu **ping**).

### B) Príprava klienta pre využitie anonymizovanej siete

#### Inštalácia Tor na klientskom VM:

- Na jednom z VMs, ktorý bude v simulovanej sieti zastávať úlohu klienta, otvorte terminál kliknutím na ikonu umiestnenú v záhlaví hlavného pracovného okna alebo v menu zvolíte **Applications > System Tools > Terminal**. Otvorí sa okno s príkazovým riadkom.
- Aktualizujte balíčky Kali Linux príkazom:

```
sudo apt update && sudo apt upgrade -y
```

- Pomocou nasledujúcich príkazov nainštalujte službu Tor:

```
sudo apt update  
sudo apt install tor
```

*Po úspešnej inštalácii služba Tor beží automaticky na pozadí systému.*

## Inštalácia Tor Browseru

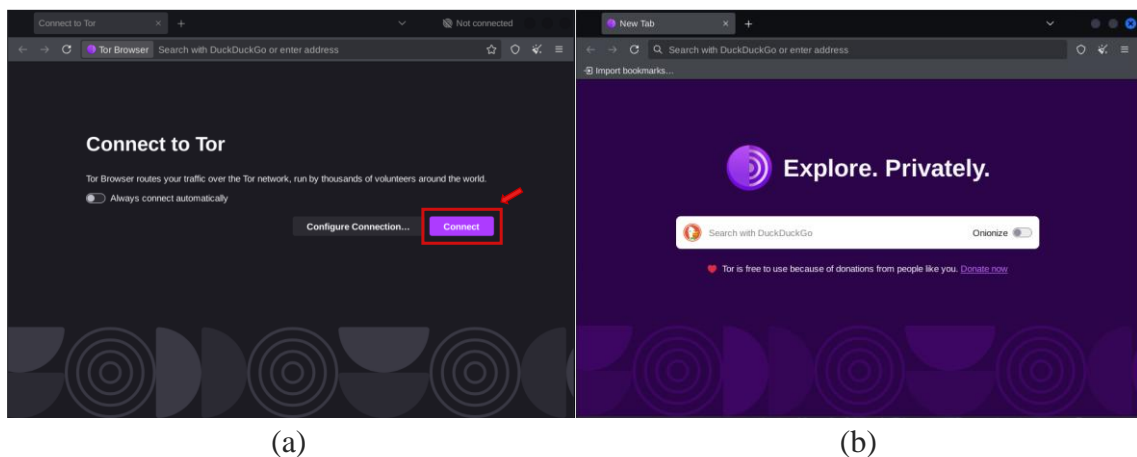
- Stiahnite inštalačný balík Tor Browser zo stránok projektu:

```
sudo apt install torbrowser-launcher -y
```

- Po stiahnutí spustíte Tor Browser zadáním nižšie uvedeného príkazu do terminálu (alebo v menu: **Applications** → **Internet** → **Tor Browser Launcher**):

```
torbrowser-launcher
```

- Po spustení akceptujte podmienky, potvrdíte a nechajte prebehnúť aktualizáciu, ak je dostupná, a následne kliknite na **Connect**.
- Po úspešnom pripojení sa otvorí anonymné prehliadacie okno, čo značí, že **Tor Browser** je pripravený na anonymné prehliadanie.



Obrázok 2.2 Pripojenie k prehliadaču Tor Browser (a), načítanie úvodnej domovskej stránky (b).

## Pripojenie k Tor sieti a overenie funkčnosti

- Po spustení Tor Browseru by sa automaticky mala otvoriť stránka (viď obr. 2.3): <https://check.torproject.org/>
- V prípade úspešného pripojenia sa objaví hláška: "Congratulations. This browser is configured to use Tor."
- Ak nedôjde k automatickému načítaniu stránky, skúste kliknúť na možnosť **Connect** alebo reštartujte Tor Browser.

## C) Sledovanie sieťovej prevádzky

### Sledovanie prichádzajúcej komunikácie vo Wiresharku na VM3

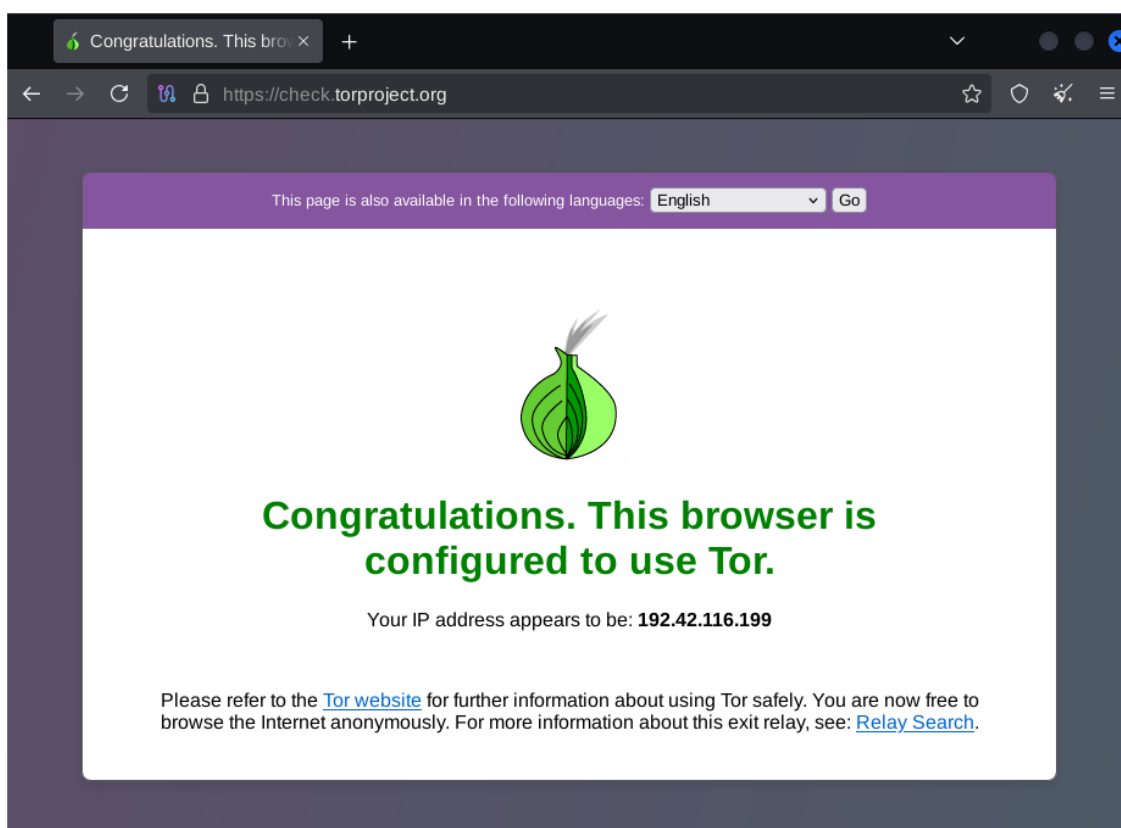
- Prejdite na VM3.
- Otvorte nové terminálové okno a spustíte nástroj Wireshark:

```
sudo wireshark &
```

- Vyberte správne sieťové rozhranie (napr. `eth0`).
- Aplikujte vhodný filter pre zachytávanie komunikácie prechádzajúcej cez sieť Tor, napr.:

```
tcp.port == 9001 || tcp.port == 443
```

- Spustíte zachytávanie sieťovej komunikácie na zvolenom rozhraní kliknutím na **Start Capturing**.



Obrázok 2.3 Úvodná stránka – potvrdenie úspešného pripojenia.

- V spustenom Tor Browseri na klientovi (VM1) sa pokúste prístupit' na webovú stránku <https://whatismyipaddress.com> a sledujte priebeh zaznamenananej komunikácie vo Wiresharku na monitorovacom zariadení (VM3). **Zaznamenajte si IP adresu zobrazenú stránkou.**
- Výsledkom by mal byť záznam komunikácie, v ktorom je možné pozorovať veľké množstvo TCP spojení, kedy ale **nebude možné ďalej analyzovať prenášaný obsah**, a to konkrétne HTTP/HTTPS požiadavky v čitateľnej

podobe, nakoľko sa jedná o šifrovaný prenos skrz vytvorenú Tor sieť. Venujte pozornosť veľkosti dátových jednotiek.

*Pozn.: štandardná veľkosť Tor bunky je 512 B. Táto bunka sa však následne zapuzdruje do TLS záznamu (TLS record), ktorý obsahuje okrem dátovej časti (= Tor bunky) aj ďalšie riadiace informácie. Z toho dôvodu je možné vidieť v zázname komunikácie inú veľkosť dátovej jednotky (napr. 590 B), dôležitá je avšak jej nemennosť v priebehu komunikácie.*

| No. | Time         | Source          | Destination     | Protocol | Length | Info                                                 |
|-----|--------------|-----------------|-----------------|----------|--------|------------------------------------------------------|
| 62  | 10.077992000 | 94.23.148.66    | 192.168.126.129 | TCP      | 60     | 9800 → 57774 [ACK] Seq=5990 Ack=5323 Win=64240 Len=0 |
| 70  | 10.152799950 | 94.23.148.66    | 192.168.126.129 | TLSv1.3  | 590    | Application Data                                     |
| 71  | 10.156462772 | 192.168.126.129 | 94.23.148.66    | TLSv1.3  | 590    | Application Data                                     |
| 72  | 10.156627722 | 94.23.148.66    | 192.168.126.129 | TCP      | 60     | 9800 → 57774 [ACK] Seq=6526 Ack=5859 Win=64240 Len=0 |
| 74  | 10.194594018 | 94.23.148.66    | 192.168.126.129 | TLSv1.3  | 590    | Application Data                                     |
| 75  | 10.196556267 | 192.168.126.129 | 94.23.148.66    | TLSv1.3  | 590    | Application Data                                     |
| 76  | 10.196556477 | 94.23.148.66    | 192.168.126.129 | TCP      | 60     | 9800 → 57774 [ACK] Seq=7062 Ack=6395 Win=64240 Len=0 |
| 81  | 10.226632002 | 94.23.148.66    | 192.168.126.129 | TLSv1.3  | 590    | Application Data                                     |
| 82  | 10.228445962 | 192.168.126.129 | 94.23.148.66    | TLSv1.3  | 590    | Application Data                                     |
| 83  | 10.228446252 | 94.23.148.66    | 192.168.126.129 | TCP      | 60     | 9800 → 57774 [ACK] Seq=7598 Ack=6931 Win=64240 Len=0 |
| 84  | 10.294087187 | 94.23.148.66    | 192.168.126.129 | TLSv1.3  | 590    | Application Data                                     |
| 85  | 10.294598838 | 192.168.126.129 | 94.23.148.66    | TLSv1.3  | 590    | Application Data                                     |
| 86  | 10.294598948 | 94.23.148.66    | 192.168.126.129 | TCP      | 60     | 9800 → 57774 [ACK] Seq=8134 Ack=7467 Win=64240 Len=0 |

Frame 70: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on interface eth0, id 0  
 Ethernet II, Src: VMware\_f6:fa:ad (00:50:56:f6:fa:ad), Dst: VMware\_b4:03:a2 (00:0c:29:b4:03:a2)  
 Destination: VMware\_b4:03:a2 (00:0c:29:b4:03:a2)  
 Source: VMware\_f6:fa:ad (00:50:56:f6:fa:ad)  
 Type: IPv4 (0x0800)  
 [Stream index: 1]  
 Internet Protocol Version 4, Src: 94.23.148.66, Dst: 192.168.126.129  
 Transmission Control Protocol, Src Port: 9800, Dst Port: 57774, Seq: 5990, Ack: 5323, Len: 536  
 Transport Layer Security  
 TLSv1.3 Record Layer: Application Data Protocol: Application Data  
 Opaque Type: Application Data (23)  
 Version: TLS 1.2 (0x0303)  
 Length: 531  
 Encrypted Application Data [...]: 532e4b8464a7bd6197ae318d77b04b67698c17ebcaa856b033a3a8077b7d4ce288b3816f498d6f5cdd8f97fc03554ae3b3

Obrázok 2.4 Ukážka zachytenej komunikácie: využitie Tor Browser pre anonymné prehliadanie<sup>56</sup>.

- K overeniu informácií o pridelenej IP adrese<sup>57</sup> použite službu whois a analyzujte zistené informácie:

```
whois 193.189.100.201
```

IP adresa, ktorú uvidíte na stránke ako napr. [whatismyipaddress.com](http://whatismyipaddress.com), je **IP adresa výstupného Tor uzla**, ktorý kontaktuje cieľový server na konci vytvoreného Tor okruhu. Cieľový server tak nemá vedomosť o tom, aký konkrétny klient ho so svojou požiadavkou kontaktov, čo názorne demonštruje spôsob, akým Tor umožňuje zaistiť anonymitu klientov.

<sup>56</sup> Vo výstupe môžete vidieť komunikáciu klienta s zlom s IP adresou 94.23.148.66 – jedná sa o IP adresu, ktorá pravdepodobne patrí jednému z uzlov siete Tor. Komunikácia na porte 9000 je pre túto sieť typická. Tor štandardne pre medzifahlé uzly (*relays*) využíva porty ako 9001, 9003 apod., avšak vzhľadom k tomu, že Tor sieť je dynamická a uzly sa môžu meniť, je bežné, že klient nadviaže spojenie s rôznymi IP adresami na rôznych portoch, ako napríklad práve 9000, 9001 alebo 9003.

<sup>57</sup> Pri zadávaní príkazu do terminálového okna použite IP adresu pridelenú Vášmu klientovi na VM1.



| Time        | 192.168.126.129 | 94.23.148.66                                         | Comment                                                     |
|-------------|-----------------|------------------------------------------------------|-------------------------------------------------------------|
| 9.748423723 | 57774           | 57774 → 9000 [SYN] Seq=0 Win=64240 Len=0 MSS         | TCP: 57774 → 9000 [SYN] Seq=0 Win=64240 Len=0 MSS           |
| 9.768111029 | 57774           | 9000 → 57774 [SYN, ACK] Seq=0 Ack=1 Win=64240        | TCP: 9000 → 57774 [SYN, ACK] Seq=0 Ack=1 Win=64240          |
| 9.768456270 | 57774           | 57774 → 9000 [ACK] Seq=1 Ack=1 Win=64240 Len=0       | TCP: 57774 → 9000 [ACK] Seq=1 Ack=1 Win=64240 Len=0         |
| 9.781625301 | 57774           | Client Hello (SNI=www.f2neitsb6y5wljbagwz.co)        | TLSv1.3: Client Hello (SNI=www.f2neitsb6y5wljbagwz.co)      |
| 9.781625351 | 57774           | 9000 → 57774 [ACK] Seq=1 Ack=518 Win=64240 Len=0     | TCP: 9000 → 57774 [ACK] Seq=1 Ack=518 Win=64240 Len=0       |
| 9.803170438 | 57774           | Server Hello, Change Cipher Spec, Application Data   | TLSv1.3: Server Hello, Change Cipher Spec, Application Data |
| 9.803452035 | 57774           | 57774 → 9000 [ACK] Seq=518 Ack=1169 Win=65535 Len=0  | TCP: 57774 → 9000 [ACK] Seq=518 Ack=1169 Win=65535 Len=0    |
| 9.810071096 | 57774           | Change Cipher Spec, Application Data                 | TLSv1.3: Change Cipher Spec, Application Data               |
| 9.810140880 | 57774           | 9000 → 57774 [ACK] Seq=1169 Ack=598 Win=64240 Len=0  | TCP: 9000 → 57774 [ACK] Seq=1169 Ack=598 Win=64240 Len=0    |
| 9.810585387 | 57774           | Application Data                                     | TLSv1.3: Application Data                                   |
| 9.810585577 | 57774           | 9000 → 57774 [ACK] Seq=1169 Ack=631 Win=64240 Len=0  | TCP: 9000 → 57774 [ACK] Seq=1169 Ack=631 Win=64240 Len=0    |
| 9.829499085 | 57774           | Application Data                                     | TLSv1.3: Application Data                                   |
| 9.848758025 | 57774           | Application Data                                     | TLSv1.3: Application Data                                   |
| 9.849586326 | 57774           | 57774 → 9000 [ACK] Seq=631 Ack=1327 Win=65535 Len=0  | TCP: 57774 → 9000 [ACK] Seq=631 Ack=1327 Win=65535 Len=0    |
| 9.853747568 | 57774           | 9000 → 57774 [PSH, ACK] Seq=1327 Ack=631 Win=6 Len=0 | TCP: 9000 → 57774 [PSH, ACK] Seq=1327 Ack=631 Win=6 Len=0   |
| 9.873075962 | 57774           | Application Data                                     | TLSv1.3: Application Data                                   |

Obrázok 2.5 *Flow Graph*<sup>58</sup> komunikácie medzi Tor klientom a vzdialeným serverom.

The screenshot shows the homepage of WhatIsMyIPAddress.com. The main content area displays the user's IP address as 193.189.100.201 (IPv4) and 2a0f:df00:0:255::201 (IPv6). Below this, it provides information about the ISP (KeFF Networks Ltd), services (Suspected Network, Sharing Device), and location (Stockholm, Sweden). A prominent red button labeled 'HIDE MY IP ADDRESS NOW' is visible. To the right, there is a map showing the location near Stockholm, Sweden, with a note that the location might not be accurate and a link to 'Update My IP Location'.

Obrázok 2.6 Tor klient: ukážka výstupu po prístupe na webové stránky [whatismyipaddress.com](https://whatismyipaddress.com) (pridelenie IP adresy).

<sup>58</sup> Flow Graph záznamu komunikácie si môžete zobrazit' cez voľbu **Statistics > Flow Graph** v záhlaví panela nástrojov hlavného okna nástroja Wireshark.

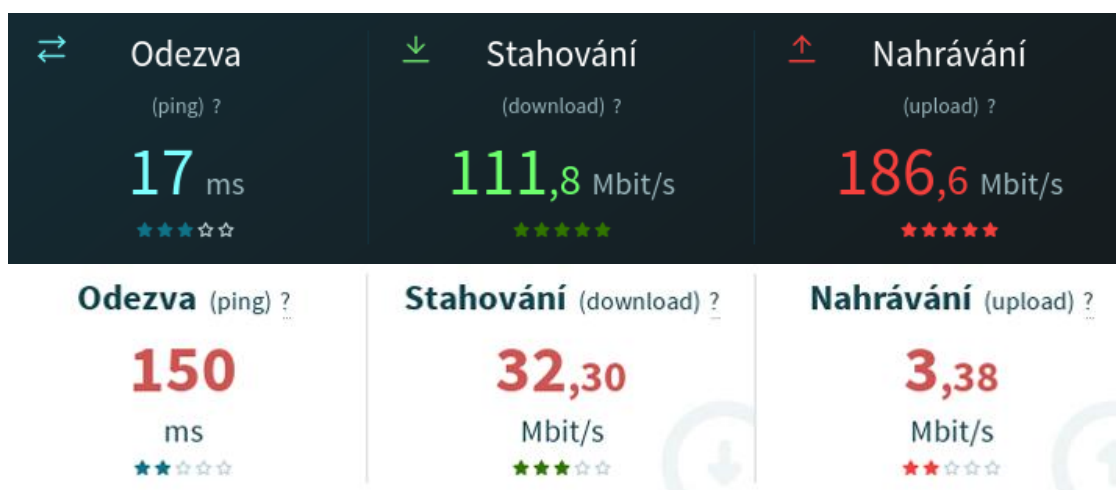
- Po zistení IP adresy klienta s Tor (napr. pomocou [whatismyipaddress.com](https://whatismyipaddress.com) alebo z výpisu vo Wiresharku), overte, či táto pridelená IP adresa skutočne prislúcha niektorému z výstupných uzlov Tor siete pomocou niektorej zo stránok:
  - <https://www.dan.me.uk/tornodes> – webová stránka poskytuje aktuálny zoznam Tor výstupných (*exit*) uzlov vrátane ich IP adries, portov a krajín. Poskytujete možnosti filtrovania a vyhľadávania konkrétnych adries.
  - <https://www.netify.ai/resources/tor> – webová stránka obsahuje podrobné informácie o Tor sieti, vrátane jej štruktúry, identifikácie prevádzky a aktuálneho zoznamu výstupných uzlov.
- Ak sa pridelená IP adresa nachádza v niektorom zo zoznamov, jedná sa skutočne o výstupný uzol (*exit node*). Všimnite si aj ďalšie informácie, ako napr. krajinu, port a názov siete. Tieto informácie si zaznamenajte a porovnajte s výsledkom zisteným pomocou `whois`.
- Otestujte funkčnosť pripojenia na `.onion` adresu cez Tor Browser – tieto adresy predstavujú špeciálne domény určené pre webové služby dostupné iba cez anonymizovanú Tor sieť. Zabezpečujú anonymitu nielen používateľa, ale tiež cieľového servera. Ich cieľom je skryť fyzické umiestnenie služby a znemožniť bežné sledovanie prevádzky. V prehliadači na klientovi s Torom (VM1) otvorte nasledovný odkaz:
 

<https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/>
- Jedná sa o špeciálnu verziu vyhľadávača DuckDuckGo pre Tor sieť. Vyskúšajte vyhľadať kľúčové slovo **"Tor exit node"** a sledujte, či sa načítajú výsledky. Môžete tiež porovnať rýchlosť, štruktúru a vzhľad stránky so štandardnou verziou DuckDuckGo.

## 2.4. Samostatná úloha

V poslednej časti laboratórnej úlohy realizujete porovnanie šifrovanej komunikácie, resp. prenosu dátových jednotiek štandardným spôsobom (t. j. s využitím aplikačného protokolu HTTPS) a prostredníctvom Tor siete. Na základe analýzy dátovej komunikácie vo Wiresharku porovnáte rozdiely medzi uvedenými možnosťami dátového prenosu.

- A) Cieľom vašej samostatnej práce bude s využitím VM2 (nového klienta) pristupovať na internet „klasicky“ (t. j. bez použitia siete Tor), kedy bude pre dátový prenos využitý protokol HTTP, resp. zabezpečený HTTPS, a následne porovnať zásadné rozdiely v anonymite, štruktúre prenosu, veľkosti dátových jednotiek a viditeľnosti dát pri použití Tor Browseru (Tor siete) a bežného prehliadača využívajúceho HTTPS pripojenie.
- B) Na základe záznamu komunikácie vo Wiresharku analyzujte hlavné rozdiely v zabezpečení identity pri použití Tor vs. klasického HTTPS pripojenia a porovnajte výhody, resp. nevýhody oboch prístupov. Taktiež analyzujte rozdiely v IP adresách, ktoré cieľový server prideli klientom na VM1 a VM2 (môžete využiť nástroj `whois`).
- C) Nakoniec uskutočnite meranie prenosovej rýchlosti a latencie oboch klientov. Doporučené je využiť stránku: <https://www.rychlost.cz>. Sledujte parametre rýchlosti sťahovania/odosielania a latenciu (dobu odozvy – ping), následne namerané výsledky oboch klientov vzájomne porovnajte.



Obrázok 2.7 Porovnanie výsledkov meraní prenosových parametrov: klient bez Tor (hore) vs. klient s Tor (dole).

### 3. Záver

V tejto laboratórnej úlohe ste sa oboznámili so základnými princípmi fungovania anonymizačných sietí a ich významom pre ochranu súkromia a zachovanie anonymity používateľov pri komunikácii na internete.

Prakticky ste si vyskúšali **inštaláciu Tor klienta** v systéme Kali Linux a tiež použitie **prehliadača Tor Browser** špeciálne prispôbeného pre anonymné prehliadanie, nadviazanie spojenia cez Tor sieť, ako aj prehliadanie internetu anonymným spôsobom. Súčasťou úlohy bola aj analýza zachytenej komunikácie pomocou nástroja Wireshark, kde ste mohli sledovať *handshake* fázu pre vytvorenie spojenia so sieťou Tor a následné šifrované dátové prenosy. Súčasťou samostatnej úlohy bolo taktiež **porovnanie priebehu šifrovanej komunikácie prostredníctvom protokolu HTTPS a komunikácie odosielanej práve cez anonymizačnú sieť Tor** a sledovanie významných rozdielov porovnaním oboch uvedených prístupov. V rámci vzájomného porovnania ste taktiež uskutočnili meranie prenosových parametrov, v priebehu ktorého ste mohli pozorovať dlhšiu dobu odozvy a nižšie prenosové rýchlosti v prípade použitia Tor.

#### 3.1. Kontrolné otázky

1. Čo je hlavným cieľom využívania anonymizačných sietí?
  - A) Dosiahnuť vysokú prenosovú rýchlosť komunikácie
  - B) Zamedziť identifikácii používateľa v celosvetovej sieti
  - C) Šifrovať komunikáciu a zabezpečiť dôvernosť prenosu medzi klientom a cieľovým serverom
  - D) Zaistiť anonymitu používateľa pri prístupe k internetu ☒
2. Na akom princípe funguje tzv. „cibuľové smerovanie“ (*onion routing*)?
  - A) Každý medziľahlý uzol na ceste od klienta k serveru pozná vždy celú trasu prenosu až k cieľu
  - B) Dáta sú šifrované v niekoľkých vrstvách a dešifrované postupne na každom uzle
  - C) Každý uzol na prenosovej trase musí poznať IP adresu cieľového servera
  - D) Dáta sú prenášané pomocou transportného protokolu UDP
3. Označte nesprávne tvrdenia o medziľahlých uzloch prenosu (*Tor Nodes*):
  - A) Vstupný Tor uzol (*Entry Node*) je prvým bodom kontaktu medzi klientom a Tor sieťou
  - B) Komunikujú medzi sebou s využitím transportného protokolu UDP
  - C) Každý uzol pozná IP adresu klienta
  - D) Tor Exit Node smeruje dáta na cieľový server a pozná IP adresu klienta

4. Ktoré z nasledujúcich charakteristík platia pre Tor bunky (*cells*)?
- A) Majú pevne stanovenú veľkosť 512 bajtov
  - B) Vždy obsahujú riadiace aj dátové informácie
  - C) Ich prenos na transportnej vrstve zabezpečuje protokol UDP
  - D) V záhlaví IP protokolu obsahujú zdrojovú IP adresu klienta
5. Aké rozdiely možno pozorovať medzi bežným HTTPS prístupom a prístupom cez Tor v nástroji Wireshark?
- A) V prípade použitia Tor sú IP adresy výstupných uzlov odlišné od zdrojovej IP adresy klienta
  - B) HTTPS nezahŕňa žiadne mechanizmy pre šifrovanie, Tor používa pre zaistenie dôveryhodnosti prenosu TLS
  - C) Tor používa fixnú veľkosť buniek
  - D) Tor umožňuje sledovanie zdrojovej IP adresy klienta rovnako ako HTTPS
6. V čoho dôsledku je pripojenie cez Tor zvyčajne pomalšie než priame pripojenie k internetu?
- A) Dáta sa prenášajú cez niekoľko medziľahlých uzlov
  - B) Používateľ (klient) musí pred odoslaním dát tieto dáta najskôr elektronicky podpísať pomocou asymetrického kryptosystému
  - C) Tor používa zastaraný kryptografický algoritmus
  - D) Každý medziľahlý uzol musí uskutočniť viacnásobné operácie šifrovania (resp. dešifrovania)
7. Čo je .onion adresa?
- A) Doména bežne dostupná pri klasickom internetovom prehliadaní
  - B) IP adresa výstupného uzla Tor siete
  - C) Špeciálna adresa určená pre skryté služby v sieti Tor
  - D) Označuje cieľovú službu, ktorá je dostupná len pri znalosti IP adresy cieľového servera tejto služby
8. Na základe akých znakov je možné identifikovať vo Wiresharku, že komunikácie prebieha prostredníctvom Tor siete?
- A) Použitím filtra `tcp.port == 9001`
  - B) Vyhľadaním EAPoL správ
  - C) Identifikáciou TLS paketov s fixnou veľkosťou dát
  - D) Zhodou zdrojových a cieľových IP adries pre všetky pakety prislúchajúce k rovnakému dátovému toku

9. Vyberte správne tvrdenia o výstupnom uzle Tor siete (*Exit Node*):
- A) Je posledným uzlom vo vytvorenom Tor okruhu
  - B) Odosiela dešifrovanú komunikáciu na cieľovú službu
  - C) Ako jediný pozná IP adresu používateľa (klienta)
  - D) Zabezpečuje TLS šifrovanie medzi klientom a serverom
10. Čo je úlohou adresárového servera v sieti Tor?
- A) Zabezpečuje šifrovanie prenosu dát medzi uzlami v sieti
  - B) Poskytuje IP adresy užívateľov v sieti
  - C) Obsahuje informácie o dostupných Tor uzloch
  - D) Pridáva nové vrstvy šifrovania pre každú odoslanú Tor bunku
11. Aké zásadné rozdiely ste pozorovali pri meraní rýchlosti pripojenia cez anonymnú sieť Tor a bez použitia Tor?
- A) Vyššiu latenciu cez Tor
  - B) Prenosovú rýchlosť bola približne rovnaká
  - C) Nižšiu rýchlosť downloadu pri použití Tor
  - D) Nižšiu latenciu prenosu v prípade bežného pripojenia

## 4. Literatúra

- [1] Dingledine, Roger, Mathewson, Nick and Syverson, Paul *Tor: The Second Generation Onion Router*. In: Paul Syverson, vol. 13, 2004. Dostupné z: <https://ieeexplore.ieee.org/document/10330539> [cit. 2024-12-02].
- [2] Rahman, Mohammad Saidur and Diadamo, Stephen and Mehic, Miralem and Fleming, Charles. *Quantum Secure Anonymous Communication Networks*. 2024. [online]. Dostupné z: [https://www.researchgate.net/figure/Three-layer-encrypted-message-in-a-Tor-network\\_fig1\\_380600931](https://www.researchgate.net/figure/Three-layer-encrypted-message-in-a-Tor-network_fig1_380600931) [cit. 2024-12-02].
- [3] MURDOCH, Steven J. a George DANEZIS. *Low-cost traffic analysis of Tor*. In: Proceedings of the 2005 IEEE Symposium on Security and Privacy. IEEE, 2005, s. 183–195. ISBN 0-7695-2339-0. [cit. 2025-04-25].
- [4] DINGLEDINE, Roger, Nick MATHEWSON a Paul SYVERSON. *Tor: The second-generation onion router*. In: Proceedings of the 13th USENIX Security Symposium. San Diego: USENIX Association, 2004, s. 303–320. Dostupné tiež z: <https://www.usenix.org/legacy/events/sec04/tech/dingledine.html>
- [5] MYRA SECURITY. *What is the Tor Network?* [online]. Myra Security, [cit. 2025-05-04]. Dostupné z: <https://www.myrasecurity.com/en/knowledge-hub/tor-network/>
- [6] PALLOTTI, Massimo a KULSHRESTHA, Mayank. *Tor Traffic in Enterprise Networks: Risks and Realities* [online]. Unit 42 – Palo Alto Networks, 2023. [cit. 2025-05-04]. Dostupné z: <https://unit42.paloaltonetworks.com/tor-traffic-enterprise-networks/>
- [7] TOR PROJECT. *Anonymity Online*. [online]. 2025 [cit. 2024-12-07]. Dostupné z: <https://www.torproject.org/>

## **Príloha H - Dokumentácia pre vyučujúceho k laboratórnej úlohe č. 11**

Laboratórna úloha č. 11

### **ANONYMIZAČNÉ SIETE**



# 1. Základné informácie k laboratórnej úlohe

Laboratórna úloha č. 11 je venovaná použitiu anonymizačných sietí pre zachovanie anonymity a ochranu súkromia používateľa internetu, a to najmä na **praktické zoznámenie sa s využitím anonymizačnej siete Tor prostredníctvom špecializovaného prehliadača Tor Browser** v Kali Linux. Cieľom úlohy je teda prakticky demonštrovať princíp fungovania siete Tor, vrátane základnej konfigurácie, prístupu na web prostredníctvom siete Tor a porovnania s bežnou komunikáciou využívajúcou aplikačný protokol HTTPS. Študenti si overia, akým spôsobom v sieti Tor prebieha anonymizácia prenosu dát, ako funguje viacvrstvové šifrovanie (*onion routing*) a aké obmedzenia alebo riziká sa viažu na jeho použitie.

## 2. Očakávané výstupy práce študentov

Praktická činnosť študentov v rámci tejto úlohy spočívala vo vlastnom testovaní možnosti anonymného prehliadania v prostredí Kali Linux vďaka použitiu prehliadača Tor Browser. Úlohou študentov je uskutočniť potrebnú konfiguráciu na zariadení klienta tak, aby bolo možné pre prístup k internetu využiť práve Tor Browser. Na základe záznamu komunikácie na Wiresharku študenti analyzujú priebeh dátovej komunikácie odoslanej cez anonymizačnú sieť Tor. Študenti by mali byť schopní vysvetliť základné princípy a priebeh „cibuľového smerovania“ v anonymizačnej sieti.

Po spustení a nakonfigurovaní virtuálnych strojov študenti vykonajú inštaláciu a spustenie Tor prehliadača (Tor Browser) na jednom zo strojov (klient VM1), pripoja sa k anonymizačnej sieti a prístupia na webové stránky pomocou Tor. Na príslušnom stroji spustia nástroj Wireshark pre monitorovanie komunikácie a jej následnú analýzu, pričom sa zamerajú na dátové prenosy využívajúce Tor SOCKS proxy a konkrétne porty ako 9001 a 9050 a preskúmajú aplikované vrstvy šifrovania (*onion routingu*).

### 2.1. Riešenie samostatnej úlohy

Samostatná úloha študentov priamo nadväzuje na praktickú časť a jej cieľom je **porovnať priebeh dátovej komunikácie s webovým serverom cez aplikačný protokol HTTPS a šifrovanej komunikácie cez Tor sieť**. Pre tento účel môžu študenti využiť ďalší VM, ktorý majú vo VMware k dispozícii. Študenti budú pracovať s dvoma klientskymi koncovými zariadeniami, pričom jeden klient bude pre prístup k internetu a webové prehliadanie, konkrétne na webovú stránku: <https://www.whatismyipaddress.com>, využívať protokol HTTPS, druhý bude na internet pristupovať prostredníctvom siete Tor. S využitím nástroja Wireshark porovnajú oba prístupy, pričom sa zamerajú na hlavné rozdiely súvisiace s ochranou súkromia a anonymity používateľov a tiež na rozdiely v šifrovaní prenášaného dátového obsahu súvisiaceho so zaistením dôvernosti prebiehajúcej komunikácie. Študenti porovnajú rozdiely v prenášaných informáciách,

veľkosti prenášaných dátových jednotiek, obsahu ich záhlaví, IP adresách komunikujúcich uzlov, šifrovaní a časovej odozve. Na vlastnom príklade zachytenej komunikácie vo Wiresharku študenti demonštrujú zásadné rozdiely v podobe prenášaných paketov ako sú napr. skrytá, resp. šifrovaná IP adresa pri prenose cez sieť Tor, viditeľná, resp. nešifrovaná IP adresa pri prenose cez HTTPS apod.

Študenti by mali po spracovaní samostatnej úlohy dospieť k nasledujúcim záverom:

- **Zobrazená IP adresa na stránke:** klient bez Toru (VM2) uvidí svoju reálnu (univerzitnú) verejnú IP adresu z rozsahu akademickej počítačovej siete CESNET, Tor klient (VM1) uvidí IP adresu výstupného uzla Tor siete.
- **WHOIS informácie o pridelení IP adrese:** IP bez Toru bude registrovaná na akademickú alebo verejnú inštitúciu (napr. CESNET, SANET), IP adresa výstupného Tor uzla bude patriť súkromnému poskytovateľovi VPS alebo anonymnému hostingu (napr. QuxLabs, OVH), často v inej krajine.
- **Analýza komunikácie vo Wiresharku:** v komunikácii klienta bez Toru (VM2) sú viditeľné štandardné HTTPS požiadavky smerované priamo na cieľový server, pri Tor klientovi komunikácia smeruje najskôr na IP adresu výstupného uzla a využíva porty typické pre Tor (napr. 9000).
- **Veľkosti prenášaných dát:** v Tor sieti sú prenášané Tor bunky s konštantnou veľkosťou 512 bajtov na úrovni TLS *payload*, čo znižuje možnosť analýzy pomocou veľkosti dát, bez použitia Toru veľkosti prenášaných paketov závisia od konkrétnych dát a môžu byť počas komunikácie premenlivé.
- **Viditeľné IP adresy vo Wiresharku:** v komunikácii klienta bez Toru (VM2) je jasne viditeľná cieľová IP adresa navštívenej webovej stránky, naopak pri komunikácii cez Tor sieť nie je skutočný cieľ spojenia zrejmy – Tor klient (VM1) komunikuje iba s IP adresou výstupného uzla Toru.
- **Šifrovanie:** v oboch prípadoch je použité HTTPS, ale Tor navyše šifruje celú trasu medzi klientom a výstupným uzlom (aplikuje viacnásobné šifrovanie).

Pri meraní dosiahnutých prenosových rýchlostí a latencie dátových prenosov oboch klientov by mali študenti pozorovať, že klient s Torom (VM1) bude mať výrazne nižšiu prenosovú rýchlosť a vyššiu latenciu než klient bez Toru (VM2). Dôvodom je štruktúra Tor siete, kedy každá odoslaná dátová jednotka je niekoľkonásobne šifrovaná a prechádza cez niekoľko medziláhlých uzlov (Tor smerovačov). Použitý spôsob smerovania a viacvrstvového šifrovania (*onion routing*) výrazne spomaľuje prenos a zvyšuje odozvu.

Ďalší faktor, ktorý môže merania ovplyvniť, je skutočnosť, že niektoré testovacie stránky nemusia byť optimalizované pre použitie Tor a môžu vykazovať nestabilné výsledky.

## 2.2. Odpovede na kontrolné otázky

1. Čo je hlavným cieľom využívania anonymizačných sietí?
  - A) Dosiahnuť vysokú prenosovú rýchlosť komunikácie
  - B) Zamedziť identifikácii používateľa v celosvetovej sieti ☒
  - C) Šifrovať komunikáciu a zabezpečiť dôvernosť prenosu medzi klientom a cieľovým serverom
  - D) Zaistiť anonymitu používateľa pri prístupe k internetu ☒
2. Na akom princípe funguje tzv. „cibuľové smerovanie“ (*onion routing*)?
  - A) Každý medziľahlý uzol na ceste od klienta k serveru pozná vždy celú trasu prenosu až k cieľu
  - B) Dáta sú šifrované v niekoľkých vrstvách a dešifrované postupne na každom uzle ☒
  - C) Každý uzol na prenosovej trase musí poznať IP adresu cieľového servera
  - D) Dáta sú prenášané pomocou transportného protokolu UDP
3. Označte nesprávne tvrdenia o medziľahlých uzloch prenosu (*Tor Nodes*):
  - A) Vstupný Tor uzol (*Entry Node*) je prvým bodom kontaktu medzi klientom a Tor sieťou
  - B) Komunikujú medzi sebou s využitím transportného protokolu UDP ☒
  - C) Každý uzol pozná IP adresu klienta ☒
  - D) Tor Exit Node smeruje dáta na cieľový server a pozná IP adresu klienta ☒
4. Ktoré z nasledujúcich charakteristík platia pre Tor bunky (*cells*)?
  - A) Majú pevne stanovenú veľkosť 512 bajtov ☒
  - B) Vždy obsahujú riadiace aj dátové informácie
  - C) Ich prenos na transportnej vrstve zabezpečuje protokol UDP
  - D) V záhlaví IP protokolu obsahujú zdrojovú IP adresu klienta
5. Aké rozdiely možno pozorovať medzi bežným HTTPS prístupom a prístupom cez Tor v nástroji Wireshark?
  - A) V prípade použitia Tor sú IP adresy výstupných uzlov odlišné od zdrojovej IP adresy klienta ☒
  - B) HTTPS nezahŕňa žiadne mechanizmy pre šifrovanie, Tor používa pre zaistenie dôvernosti prenosu TLS
  - C) Tor používa fixnú veľkosť buniek ☒
  - D) Tor umožňuje sledovanie zdrojovej IP adresy klienta rovnako ako HTTPS

6. Z akého dôvodu je pripojenie cez Tor zvyčajne pomalšie než priame pripojenie k internetu?
- A) Dáta sa prenášajú cez niekoľko medziľahlých uzlov ☒
  - B) Používateľ (klient) musí pred odoslaním dát tieto dáta najskôr elektronicky podpísať pomocou asymetrického kryptosystému
  - C) Tor používa zastaraný kryptografický algoritmus
  - D) Každý medziľahlý uzol musí uskutočniť viacnásobné operácie šifrovania (resp. dešifrovania) ☒
7. Čo je .onion adresa?
- A) Doména bežne dostupná pri klasickom internetovom prehliadaní
  - B) IP adresa výstupného uzla Tor siete
  - C) Špeciálna adresa určená pre skryté služby v sieti Tor ☒
  - D) Označuje cieľovú službu, ktorá je dostupná len pri znalosti IP adresy cieľového servera tejto služby
8. Na základe akých znakov je možné identifikovať vo Wiresharku, že komunikácie prebieha prostredníctvom Tor siete?
- A) Použitím filtra tcp.port == 9001 ☒
  - B) Vyhľadáním EAPoL správ
  - C) Identifikáciou TLS paketov s fixnou veľkosťou dát ☒
  - D) Zhodou zdrojových a cieľových IP adries pre všetky pakety prislúchajúce k rovnakému dátovému toku
9. Vyberte správne tvrdenia o výstupnom uzle Tor siete (Exit Node):
- A) Je posledným uzlom vo vytvorenom Tor okruhu ☒
  - B) Odošle dešifrovanú komunikáciu na cieľovú službu ☒
  - C) Ako jediný pozná IP adresu používateľa (klienta)
  - D) Zabezpečuje TLS šifrovanie medzi klientom a serverom
10. Čo je úlohou adresárového servera v sieti Tor?
- A) Zabezpečuje šifrovanie prenosu dát medzi uzlami v sieti
  - B) Poskytuje IP adresy užívateľov v sieti
  - C) Obsahuje informácie o dostupných Tor uzloch ☒
  - D) Pridáva nové vrstvy šifrovania pre každú odoslanú Tor bunku
11. Aké zásadné rozdiely ste pozorovali pri meraní rýchlosti pripojenia cez anonymnú sieť Tor a bez použitia Tor?
- A) Vyššiu latenciu cez Tor ☒
  - B) Prenosová rýchlosť bola približne rovnaká
  - C) Nižšiu rýchlosť downloadu pri použití Tor ☒
  - D) Nižšiu latenciu prenosu v prípade bežného pripojenia ☒

## 2.3. Dopĺňujúce otázky

Nižšie uvedené otázky môžu byť využité pri kontrole výstupov samostatnej práce študentom s cieľom overiť, či skutočne porozumeli riešenej problematike v praktickej časti laboratórnej úlohy.

### 1. Vysvetlite, čo je anonymizačná sieť a aký je hlavný účel jej použitia.

- Anonymizačná sieť predstavuje technológie určenú k zaistieniu ochrany súkromia a anonymity používateľov počas ich online aktivity v prostredí internetu. Jej hlavným cieľom je utajiť IP adresu koncového používateľa a smerovať všetku jeho komunikáciu cez viacero sprostredkovateľov (medziľahlých uzlov) tak, aby nebolo možné v žiadnom bode prenosu jasne identifikovať pôvodcu (a príp. i adresáta) zaslanej dátovej jednotky.

### 2. Popíšte princíp „cibuľového smerovania“ v sieti Tor.

- Sieť Tor pracuje na princípe tzv. *onion routing*, t. j. mechanizme šifrovania spočívajúcom vo viacnásobnom šifrovaní prenášaných dát, ktoré sú počas prenosu postupne dešifrované na jednotlivých medziľahlých uzloch. Dáta sú odoslané cez sériu uzlov, pričom každý uzol dokáže dešifrovať len jednu („vonkajšiu“) vrstvu, čím pozná len predchádzajúci a nasledujúci uzol, čo je základný predpoklad k tomu, aby mohla byť zaistená anonymita koncového používateľa.

### 3. Čo je úlohou *Entry Node* a *Exit Node* v Tor sieti?

- *Entry Node* (vstupný uzol) je prvý bod v Tor sieti, ktorý prijíma zašifrované dáta od klienta. *Exit Node* (výstupný uzol) je posledný uzol, ktorý dáta dešifruje a posiela ich na cieľový server. *Entry Node* pozná IP klienta, ale nemá vedomosť o tom, kto je cieľovým adresátom komunikácie. Naopak *Exit Node* pozná cieľovú IP adresu (t. j. adresáta), ale nie pôvodcu dát.

### 4. Akým spôsobom dochádza k zaistieniu anonymity používateľa pri využití anonymizačnej siete?

- Anonymita je dosiahnutá vrstvením šifrovania a smerovaním komunikácie cez niekoľko uzlov, kde každý pozná len časť prenosovej trasy. Tým sa znemožní sledovanie úplnej komunikácie, vrátane jednoznačného určenia koncových bodov komunikácie.

### 5. Aké sú hlavné výhody a nevýhody používania siete Tor?

- Výhody: vysoká úroveň anonymity používateľa, ochrana pred sledovaním ochrana súkromia koncového užívateľa.
- Nevýhody: nižšia rýchlosť prenosu, potenciálne nežiadúce blokovanie niektorých služieb, možnosť kompromitácie výstupných uzlov, obmedzená podpora niektorých aplikácií.

**6. Pomocou akého filtra vo Wiresharku je možné zobrazit' len komunikáciu prenášanú anonymizačnou sieťou Tor? Existuje konkrétny filter pre zobrazenie správ (resp. buniek) Tor protokolu?**

- Komunikácia v anonymizačnej sieti Tor síce nevyužíva žiadny konkrétny aplikačný protokol označený napr. ako "tor", ktorý by slúžil práve pre potreby zaistenia anonymity užívateľov, ale je možné použiť vhodné filtre pre filtrovanie komunikácie na úrovni transportnej vrstvy. Príklady možného filtrovania zachytenej komunikácie na základe portov:

```
tcp.port == 9001 || tcp.port == 9050 || tcp.port == 443
```

*(ak sa používa aj HTTPS)*

**7. Demonštrujte na praktickom príklade (môžete využiť napr. časť zaznamenananej komunikácie vo Wiresharku) rozdiely medzi bežnou šifrovanou HTTPS komunikáciou a komunikáciou cez sieť Tor.**

- V prípade šifrovanej komunikácie cez HTTPS je vo Wiresharku zrejmé IP adresa klienta aj cieľového servera, zatiaľ čo pri prenose cez Tor nie sú IP adresy koncových zariadení na jednotlivých uzloch siete viditeľné, resp. nie sú prenášané IP záhlaví v otvorenej, čitateľnej podobe.

**8. Aké rozdiely možno pozorovať pri meraní rýchlosti pripojenia medzi klientom používajúcim Tor a klientom bez Tor? Uveďte hlavný dôvod tohto rozdielu.**

- Klient používajúci Tor (VM1) bude mať nižšiu prenosovú rýchlosť a vyššiu latenciu. Dôvodom je, že Tor smeruje šifrovanú komunikáciu cez viacero medziľahlých uzlov, čo má za následok vznik oneskorenia a zníženie dosiahnutej prenosovej rýchlosti a celkovej priepustnosti spojenia.

**9. Z akého rozsahu bola pridelená IP adresa klientovi bez použitia Toru (VM2)?**

- IP adresa klienta bez Toru (VM2) bola pridelená z verejného rozsahu siete organizácie CESNET, konkrétne ide o rozsah 147.251.0.0/16, ktorý je registrovaný pre Vysoké učení technické v Brně (VUT v Brně).

## Príloha I - Obsah priloženého archívu

Priložený archív s názvom **Diplomova\_praca\_Priloha.zip** odovzdaný ako elektronická príloha k tejto diplomovej práci, obsahuje všetky podklady k vybraným laboratórnym úlohám (úlohy č. 3, 4, 8 a 11), a to v slovenskej aj českej jazykovej verzii. Archív je rozdelený do nasledovných častí:

- **PDF** – obsahuje .pdf verziu súboru s textovou časťou Diplomovej práce. Ďalej je rozdelený na dva adresáre:
  - **SK** – obsahuje slovenské verzie študentských návodov a dokumentácie pre vyučujúceho pre každú úlohu vo formátoch .pdf
  - **CZ** – obsahuje české verzie študentských návodov a dokumentácie pre vyučujúceho pre každú úlohu vo formátoch .pdf
- **HTML** – rozdelený na tri adresáre
  - **Hash\_generator** – adresár obsahuje súbor hash.html implementujúci webovú stránku s vytvoreným generátorom pre výpočet SHA-256 *hashu*
  - **SK\_HTML\_verzia** – obsahuje pre každú úlohu html kódy a ďalšie dôležité súbory pre zobrazenie webových stránok so slovenskou verziou návodu
  - **CZ\_HTML\_verze** – obsahuje pre každú úlohu html kódy a ďalšie dôležité súbory pre zobrazenie webových stránok s českou verziou návodu
- **README.txt** – sprievodný súbor so stručnými inštrukciami k použitiu priloženého obsahu.

Schematické znázornenie adresárovej štruktúry priloženého archívu:

