VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná informatika

NÁSTROJE A TECHNIKY PRO PENETRAČNÍ TESTOVÁNÍ

Bakalářská práce

Autor práce: Lukáš Jirka, DiS

Vedoucí práce: Mgr. Antonín Přibyl

Jihlava 2026

Vysoká škola polytechnická Jihlava

Tolstého 16, 586 01 Jihlava

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	**Lukáš Jirka, DiS.**
Studijní program:	Aplikovaná informatika
Garant studijního programu:	Ing. Lenka Kuklišová Pavelková, Ph.D.
Název práce:	**Nástroje a techniky pro penetrační testování**
Vedoucí práce:	Mgr. Antonín Přibyl
Cíl práce:	Práce se zaměří na populární nástroje pro penetrační testování v distribuci Kali linux. Student popíše funkcionalitu základních nástrojů (Wireshark, Nmap, Aircrack, Johny the Ripper) a otestuje je na praktických příkladech.

Abstrakt

Bakalářská práce se zaměřuje na problematiku textově orientovaných operačních systémů, principy open-source softwaru a možnosti penetračního testování s využitím distribuce Kali Linux. Teoretická část popisuje vývoj systémů UNIX a GNU/Linux, jejich vzájemné rozdíly a související licenční podmínky. Dále popisuje vlastnosti Kali Linux a analyzuje vybrané nástroje. Nástroje jsou Wireshark, Bettercap, Nmap, Aircrack-ng, John the Ripper, Burp Suite, OWASP ZAP a SQLMap. Následující analýzy vybraných technik zahrnují diagnostiku sítě, MITM útoky, skenování portů a služeb, bezdrátové útoky, prolomení hesla, skenování webů a útoky na webové aplikace. V praktické části se testují jednotlivé nástroje na konkrétní techniky, hodnotí získaná data a analyzuje využití nástrojů při odhalování zranitelností. Cílem práce je vyhodnotit využití poskytovaných nástrojů v Kali Linux při konkrétních scénářích a zároveň dokázat jeho efektivní využití.

Klíčová slova

Kali Linux; textové operační systémy; penetrační testování; nástroje; útoky; zranitelnosti

Abstract

The bachelor's thesis focuses on text-based operating systems, the principles of open-source software, and the possibilities of penetration testing using the Kali Linux distribution. The theoretical part describes the development of UNIX and GNU/Linux systems, their differences, and the related licensing conditions. It also outlines the characteristics of Kali Linux and analyzes selected tools, including Wireshark, Bettercap, Nmap, Aircrack-ng, John the Ripper, Burp Suite, OWASP ZAP and SQLMap. The following analyses of selected techniques include network diagnostics, MITM attacks, port and service scanning, wireless attacks, password cracking, web scanning, and attacks on web applications. In the practical part, the individual tools are tested on specific techniques, the collected data are evaluated, and the use of the tools in identifying vulnerabilities is analyzed. The aim of the thesis is to evaluate the use of the tools provided in Kali Linux in specific scenarios and to demonstrate their effective use.

Keywords

Kali Linux; Text-based operating systems; Penetration testing; Tools; Attacks; Vulnerabilities

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „**AZ**“).

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle Směrnice pro vedení, vypracování a zveřejňování závěrečných prací na VŠPJ, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 13.04.2026

…….

Podpis studenta/ky

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce Mgr. Antonínovi Přibylovi, za odborné vedení bakalářské práce, rady, ochotu, vstřícnost a odborné připomínky.

Obsah

Seznam obrázků

Seznam tabulek

Seznam zkratek

AP	Access Point
EAPOL	Extensible Authentication Protocol over LAN
HW	Hardware
MSF	Metasploit Framework
OS	Operační systém
SW	Software
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VŠPJ	Vysoká škola polytechnická Jihlava

Úvod

Dnešní doba je dobou informačních technologií. Množství počítačových systémů, sítí a programů se exponenciálně zvyšuje, jak v domácí, tak i ve firemní sféře, a tak se exponenciálně zvyšují i možnosti kyberzločinců proniknout do systému. Šikovní hackeři mohou mimo jiné ukrást data, zašifrovat data či vsadit do systému škodlivý kód. Spolu se zvyšujícími se počty prvků informačních technologií narůstá i jejich zranitelnost, narůstá i počet kyberzločinců a narůstá i úroveň kybernetické bezpečnosti. Obecně platí, že čím více máme v systému instalovaných služeb či programů tak je systém více zranitelnější vůči útokům.

Za pozornost také stojí fakt, že v dnešní době probíhají školení o kybernetické bezpečnosti na úrovni obyčejných zaměstnanců pracující s počítači. Jak se například zachovat při přijetí podezřelého emailu, což ještě před pár lety bylo tabu. Školení zaměstnanců mimo jiné svědčí i o dopadu kybernetické bezpečnosti v současné době.

Penetrační testování simuluje útoky na počítačové systémy, cílem penetračního testování je odhalit zranitelnost systému a tak jej v předstihu ochránit před nežádoucími útoky. Tester se snaží proniknout do systému stejnými technikami, které používají hackeři. Výsledky zranitelnosti systému mohou být různé od neaktuálního software, slabá hesla či chybné konfiguraci v síti. V širším kontextu je penetrační testování metoda kybernetické bezpečnosti a je praktickou a nejdůležitější součástí bezpečnostního auditu.

Distribuce Linuxu zvaná Kali Linux, dříve Backtrack, obsahuje značný počet nástrojů a je speciálně vyvinutá pro penetrační testování. Kali Linux je distribuce odvozená od Debianu. Hlavními konkurenty jsou například Parrot OS (Debian) či BlackArch (Arch Linux). Mezi největší výhody zmiňovaných distribucí jsou již nainstalované nástroje pro penetrační testování, čímž tvoří balík nástrojů vše v jednom. V ostatních Linuxových distribucí instalované nejsou a je potřeba je doinstalovat. Závěrečná práce se bude věnovat nástroji a techniky pro penetrační testovaní pomocí linuxové distribuce Kali Linux.

Cílem bakalářské práce je, včetně představení distribuce Kali Linux, popsat nástroje určené pro penetrační testování a techniky možných útoků. V krátkosti se bude věnovat i teorii penetračního testování. V praktické části se nástroje i techniky pro penetrační testování převedou do praxe a výsledky se zdokumentují včetně popisů přesných postupů.

1 Teoretická část

Teoretická část se zaměří na vývoj a historii textově orientovaných operačních systémů, nebude zde rozebírána historie operačních systémů s grafickým uživatelským rozhraním. Vysvětlí pojmy a rozdíly mezi open-source a svobodným software. Vysvětlí licencování open-source projektů. Dále se bude se věnovat distribuci Kali Linux a jeho nástroji a techniky pro penetrační testování.

Operační systém je zodpovědný za správu všech prostředků počítačového systému, jako jsou disky, paměti, klávesnice či myš ale i software (Typy operačních systémů a jejich úplná historie, 2021). Operační systém je software komunikující s hardwarem i softwarem počítače a samotný provoz operačního systému je řízen uživatelem. Některé procesy jsou automaticky řízené operačním systémem samotným a běží na pozadí operačního systému bez zásahu uživatele. Dále umožňuje spouštět další aplikace, přidělovat a odebírat systémové prostředky procesům, spouštět programy a mnohem více. Základním funkcí operačního systému je jádro, což je základní program OS. Poskytuje základní úroveň kontroly nad každým hardwarovým zařízením a je jakýmsi prostředníkem mezi OS a hardwarem, k tomu však potřebuje ovladače. Mezi hlavní funkce jádra patří správa procesoru, čtení a zápis dat z paměti, zpracování příkazů, funkčnost monitoru, klávesnice či myši. Další nedílnou součástí OS jsou ovladače, které zajišťují komunikaci mezi hardwarem a operačním systémem. Dále OS obsahuje API, rozhraní pro programování aplikací, a grafické rozhraní, čili vykreslovaní grafických prvků (Typy operačních systémů a jejich úplná historie, 2021).

V minulosti fungovali operační systémy výhradně přes příkazové řádky CLI (Command Line Interface). Uživatel zadá do příkazové řádky příkaz, na jehož základě operační systém příkaz provede a případně dodá uživateli odpověď. Grafické uživatelské rozhraní, GUI (Graphical User Interface) je moderní grafická nadstavba operačních systémů. Je mnohem více user-friendly než CLI a funguje například na základě oken, které známe z OS Windows.

1.1 Historie textově orientovaných operačních systémů

První sálové počítače, mainframe, byli bez operačního systému. Každý program vložený do sálových počítačů musel obsahovat nezbytné kódy ke spuštění počítače, kódy pro komunikaci s připojeným hardwarem (ovladače) a samotný program, který měl počítač provést. I nejjednodušší program byl sám o sobě velmi složitý a musel být naprosto bezchybný. Často programátoři museli zadávat programy dokonce i v binární soustavě. Později se přešlo na knihovny pomocného kódu uložené na děrných štítcích a magnetických páskách, které byli spojeny s programem, aby pomohli při vstupních a výstupních operacích, do té doby bylo nutné ručně řídit vstup i výstup programu ručně (Historie operačních systémů, 2012). Za průkopníka se zajištěním správy vstupu i výstupu programu lze považovat GM-NAA I/O od General Motors, z roku 1956, určený pro IBM 704, což byl první operační systém vůbec. Začal se tak vyvíjet systémový software, který usnadňoval psaní a spouštění programů, programátoři se tak mohli soustředit čistě na svůj samotný program. V 60. letech minulého století byla firma IBM prvním vývojářem operačních systémů, které byli součástí jejich počítačů. Bohužel však přímo na konkrétní typ počítačů, protože systém byl napevno spjat

s hardwarem. Každý počítač měl jiný operační systém. Rozdílnosti v systémech ukončil až operační systém OS/360 od IBM na začátku 60. let minulého století, který mohl běžet na různých strojích té doby. Později začali počítače obsahovat stále více a více runtime knihoven (softwarových programů), které se spojili při startu počítače a byli základem pro dnešní operační systémy. Zároveň tak tvořili základní programové vybavení počítače. Na konci 60. let minulého století byla zdarma v programovacím jazyce C vyvinuta firmou IBM první verze operačního systému UNIX. Mnoho moderních OS jako Apple OS či všechny verze Linuxu jsou založena na OS UNIX.

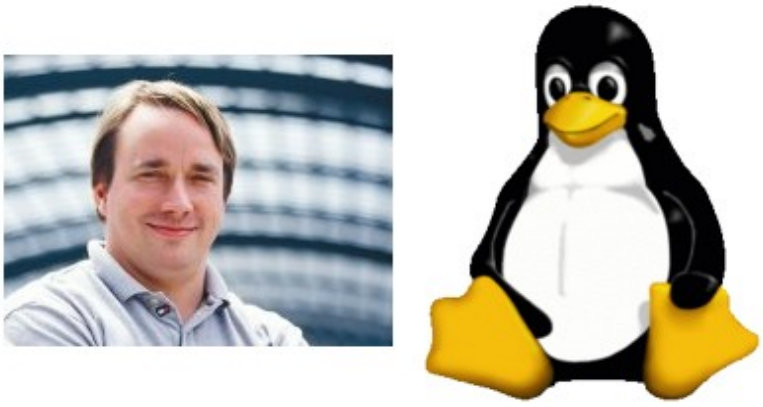
1.1.1 Historie a vývoj operačních systémů UNIX

První operační systémy byli dodávány velkým sálovým počítačům (mainframe). Technologický giganti té doby začali vyvíjeli své operační systémy. IBM vytvořil první operační systém, který v omezené míře podporoval multitasking (MFT). Multitasking znamená běh více programů najednou, což bylo do té doby naprosto nemyslitelné. Další operační systém, který lze označit za průlomový byl Multics (Multiplexed Information and Computing Service). Multics byl vyvíjen od roku 1964 a jeho základní myšlenka byla dodávka výpočetního výkonu počítačům podobně jako třeba dodávky elektřiny, plynu či vody (Historie operačních systémů, 2012). Na jeho vývoji se podíleli Bell Labs, General Motors i MIT a byl nasazen zejména také na sálové počítače. Multics znamenal pro své vývojáře zejména nekonečný a složitý vývoj avšak nové myšlenky použité v tomto OS inspirovali vývojáře k vytvoření OS UNIX (Uniplexed Information and Computing Service). UNIX vznikl v roce 1969 a je základem pro Linux i MacOS. UNIX vytvořila telekomunikační společnost AT&T ve své výzkumné divizi Bell Labs. Za zmínku rovněž stojí autoři UNIXU – Ken Thompson, který dříve pracoval na projektu Multics, a Dennis Ritchie, který je zároveň tvůrcem programovacího jazyka C (později i C++), do kterého byl UNIX od roku 1973 přepsán a mohl se tak dostat na nejrůznější platformy. Původně se jmenoval UNICS, což je ironický název na název Multics. Později se však jeho název ustálil na UNIX. Multics byl mimochodem mnohem složitější systém oproti UNIXU. Zmiňovaný OS položil základy moderních operačních systémů. Mezi jeho hlavní přednosti patří zejména multitasking a multiuser (systém dostupný více uživatelům zároveň), hierarchický souborový systém (adresáře uspořádané do stromové struktury), rovněž je snadno přenositelný a je snadno portován na různé počítače. Mezi jeho další výhody patří jednoduchost a nezávislost na jakémkoliv hardware. Zmiňovaný systém je dnes, i po tolika letech, stále jeden z nejpoužívanějších zejména především díky svému základu. Základ je všude stejný, ať se jedná o Linux, BSD, Minix, AIX, MacOS, Solaris a další. Je víceméně také důležité zmínit, že AT&T kvůli antimonopolním omezením nemohl UNIX do roku 1983 komerčně prodávat, a tak jej poskytl univerzitám. Takto se dostal UNIX na Berkeley, kde ho upravili a vytvořili tak BSD (Berkeley Software Distribution), vlastní verzi systému s novými funkcemi jako TCP/IP a virtuální pamětí. Od roku 1983 se UNIX prodával komerčně pod názvem System V. Za pozornost jistě ještě stojí operační systém VMS od firmy DEC (z roku 1978), který měl na svou dobu poměrně moderní 32 bitovou architekturu a podporoval multitasking (Historie operačních systémů, 2012).

1.1.2 Historie a vývoj jádra Linux

V roce 1969 byl uveden na trh operační systém UNIX, podporoval multitasking a bylo rovněž možné využívat jeden systém více uživateli zároveň (multi-user operating system). Jediná

nevýhoda systému byla, že byl vydáván pod placenou licencí. V roce 1991 dostal Linus Torvalds, tehdy 21letý student univerzity v Helsinkách ve Finsku, nápad pracovat na vývoji vlastního unixového jádra. Ve škole se dostal do styku s operačním systémem Minix, což je odlehčená verze UNIXU, kterou si ihned pořídil a velmi si Minix oblíbil. Později ale zjistil, že mu nestačí, chtěl využívat UNIX naplno. Minix byl příliš jednoduchý a bez možnosti získat zdrojové kódy. Protože byl studentem vysoké školy a neměl peníze na koupi licence UNIXU, rozhodl se tak vytvořit si vlastní jádro operačního systému inspirované UNIXEM a které by bylo zároveň provozovatelé i na běžném počítači. Do online diskuzní skupiny Usenet napsal Linus zprávu, ve které oznámil vývoj vlastního jádra operačního systému a okamžitě si jeho myšlenka našla mnoho příznivců, kteří začali na jeho vývoji spolupracovat. Vývoj jádra se rozrostl do obřích rozměrů a vznikl tak Linux. Jádro Linux ve svých novějších formách je komunitní dílo, Linus vytvořil první verze jádra. Název Linux dostalo jádro podle jména Linus. Název rovněž znamená zkratku Linux Is Not UniX, v překladu Linux není UNIX (Historie operačního systému GNU/Linux, 2025). Zkratka rovněž odráží pravdu o Linuxu, protože Linux opravdu není UNIX systém, ale pouze Unix-like systém, systém podobný UNIXU. Celý zdrojový kód UNIXU byl přepsán, ani jeden řádek zdrojového kódu nepochází z originálního UNIXU. Linux není původní certifikovaný UNIX, ale chová se stejně a vychází z jeho principů. V roce 1992 začal vydávat nové verze jádra Linus pod projektem GNU a svobodnou licencí GNU GPL, čímž vznikl operační systém GNU/Linux.



Obr. 1: Linus Torvalds a Tux, maskot jádra Linux
Zdroj: Historie operačního systému GNU/Linux (2025)

1.1.3 Historie a vývoj operačního systému GNU

V roce 1983 přišel Richard Matthew Stallman s nápadem nového svobodného operačního systému unixového typu, který by mohl každý využívat, studovat, upravovat a dále šířit. Založil projekt svobodného operačního systému GNU. Jeho hlavní myšlenka byla, že operační systém bude složen jen a pouze ze svobodného software. Proto sepsal Stallman licenci GNU GPL, pod kterou jsou šířeny všechny části systému GNU. GNU GPL je speciální druh licence, který zaručuje, že programy a jejich další verze zůstanou otevřené a dostupné pod opět pouze pod stejnou licencí. Během deseti let byl operační systém projektu GNU naprosto použitelný a zároveň plně kompatibilní s UNIX systémy. Obsahoval řadu nástrojů, systémové knihovny a aplikace, které známe z různých distribucí GNU/Linux, například překladač GCC, textové editory a další. Chybělo však jádro, které by zajistilo běh systému a komunikaci s hardwarem. GNU sám o sobě obsahoval pouze nástroje pro obsluhu operačního systému bez jádra

systému. Proto byl zahájen v roce 1990 vývoj jádra Hurd, jež v současné době stále není dokončen. Protože byl však vývoj jádra Linux mnohem rychlejší, úspěšnější a měl i větší počet příznivců, než vývoj jádra Hurd, došlo ke společnému používání operačního systému GNU s jádrem Linux. Čímž vznikl plný název operačního systému jako celku i s jádrem GNU/Linux. Uživatelé však často systém oslovují pouze Linux. Což je nevděčné vůči operačnímu systému projektu GNU, který obsahuje všechny nástroje, bez kterých se nedá jádro Linux ovládat. GNU je však plně kompatibilní i s jinými unixovými jádry například Minix, Hurd, BSD a další. GNU i Linux jsou dva na sobě nezávislé projekty, spolu tvoří však silnou a nejpoužívanější dvojici ve světě svobodného software. Kompletní operační systém však tvoří pouze a jedině jednotlivé distribuce GNU/Linux, jsou kombinací Jádra Linux, operačního systému GNU a dalších projektů, programů a nástrojů. Nikdy tak nepracujeme pouze s GNU/Linux ale i dalšími nástroji v kompletní linuxové distribuci (Historie operačního systému GNU/Linux, 2025). Kompletní konkrétní operační systémy jako celky vycházejí jako distribuce. Mají různá grafická prostředí, balíčkovací systémy, aplikace či nasazení (domácí desktop, enterprise prostředí, servery). Mezi nejznámější rodiny linuxových distribucí patří Debian, RHEL, ArchLinux a další, distribucí je celá řada a většina z nich se odvíjí od základní distribuce (například LinuxMint, KaliLinux či Ubuntu jsou linuxové distribuce z rodiny Debian. AlmaLinux, CentOS či Fedora jsou linuxové distribuce z rodiny Red Hat). Linus Torvalds se nadále věnuje vývoji a jako hlavní správce má stále rozhodovací práva v otázkách kolem projektu Linux Kernel. Slovo Kernel je anglický název pro jádro systému. Richard Matthew Stallman se věnuje obhajobě svobodného software a práci pro projekt GNU, vyvíjí také známý textový editor emacs (Historie operačního systému GNU/Linux, 2025).

Oba projekty mají své různé maskoty, GNU je anglickým názvem pro pakoně a má maskota pakoně Heckerta. GNU je rekurzivní zkratka slov GNU's Not UNIX. Maskotem Linuxu je tučňák s názvem Tux, což je opět zkratka slov Torvalds UniX a zároveň zkráceně tuxedo, neboli anglický název pro frak.

Obr. 2: Richard Matthew Stallman a Heckert, maskot operačního systému GNU
Zdroj: Historie operačního systému GNU/Linux (2025)

1.1.4 Rozdíly mezi operačními systémy UNIX a GNU/Linux

Největší a nejznatelnější rozdíl mezi zmíněnými operačními systémy je bezesporu v licencování. UNIX je proprietární operační systém s chráněnou značkou, některé jeho verze mají zdrojový kód uzavřený a jeho licence je placená. Proprietární znamená, že patří konkrétnímu vlastníkovi a je chráněn jeho autorskými nebo licenčními právy. Oproti tomu GNU/Linux je naprosto

svobodný open-source systém pod licencí GNU GPL a je zdarma. Zdrojové kódy jádra Linux jsou sepsány od nuly, nemají s UNIXEM žádné společné kódy. Linux je pouze přepsanou přesnou napodobeninou UNIXU, tzv. Unix-like systém.

Systémy UNIX jsou certifikované standardem POSIX. GNU/Linux sice certifikované standardem POSIX nejsou, jsou s ním ale plně kompatibilní. UNIX existuje v několika komerčních variantách, například AIX, HP-UX či Solaris. GNU/Linux existuje ve stovkách nekomerčních distribucích díky licenci GNU GPL. GNU GPL licence znamená, že distribuce lze volně stáhnout, upravit a nasdílet (má otevřený zdrojový kód). UNIX byl vyvíjen firmami, zatímco GNU/Linux se vyvíjí komunitně. Z hlediska využití je UNIX využíván nejvíce v korporátním prostředí například bankami, pojišťovnami či telekomunikacemi. GNU/Linux se využívá zejména na serverech či v mobilních přístrojích. GNU/Linux je mnohem otevřenější a flexibilnější než tradiční UNIX (Microsoft Copilot: váš AI pomocník, 2025).

Dále je možné rozebrat technické rozdíly, které jsou mezi zmíněnými operačními systémy již menší, avšak znatelné. Patří mezi ně například souborové systémy obou systémů, zatímco UNIX používal menší množství souborových systémů jako jsou UFS (UNIX File System) nebo jeho varianty FFS či BSD a často záviseli na konkrétní variantě UNIXU (například Solaris měl ZFS nebo AIX používal JFS). GNU/Linux podporuje celou řadu souborových systémů, například ext2, 3, 4, XFS, Btrfs, ZFS a mnoho dalších, které si může uživatel vybrat při instalaci. Souborové systémy nejsou kompatibilní mezi oběma zmíněnými operačními systémy. UNIX opět nemá jednotný standart adresářové struktury (je různý podle jednotlivé variantě UNIXU), zatímco GNU/Linux má adresářovou strukturu podle standartu FHS, což znamená, že jeho adresářové struktury jsou stejné v každé distribuci. Základ je stejný a začíná kořenem. Příkazy v shellu jsou si opět velmi podobné avšak opět rozdílné. Linux používá jednotné moderní GNU nástroje, zatímco systémy UNIX jsou opět rozdílné podle jednotlivé variantě systému a drží si vlastní specifické příkazy. Základní příkazy jsou však stejné. UNIX využívá základní POSIX shell sh, Bourne Shell. GNU/Linux používá rozšířený GNU shell bash, Bourne Again Shell, který má více funkcí a je kompatibilní s UNIX shellem sh (Microsoft Copilot: váš AI pomocník, 2025).

Rozdílů mezi operačním systémem UNIX a operačním systémem GNU/Linux je celá řada. Až na pár větších rozdílných vlastností jsou další, zejména technické rozdíly, spíše zanedbatelné. Oba systémy si jsou velice podobné. GNU/Linux je víceméně stejný pro všechny své odvozené distribuce. Unixové operační systémy si jsou mezi sebou také velice podobné, avšak rozdílné, záleží na jednotlivé variantě systému.

1.2 Open source, licencování a svobodný software

V další kapitole bakalářské práce se stručně objasní pojmy Open-source (otevřený versus uzavřený zdrojový kód) a svobodný software. Protože GNU/Linux je open-source je důležité pojem vysvětlit. Distribuce GNU/Linux mohou však obsahovat i proprietární software třetích stran s uzavřeným zdrojovým kódem. Distribuci je tak možné upravit a sdílet dále, ale již se musí vedle licence GNU GPL uvést i licence konkrétního proprietárního software. V bakalářské práci se již uvedlo, že samotná linuxová distribuce netvoří pouze operační systém GNU a jádro Linux ale i další projekty či software. Dokonce jsou k dispozici ke stažení distribuce Linuxu obsahující pouze svobodný software nebo distribuce Linuxu obsahující i software třetích

stran. Mimo jiné často u běžných uživatelů dochází k matení pojmů open-source a svobodný software, jedná se o dva různé termíny.

1.2.1 Open source

Slova open-source (v překladu otevřený zdroj) znamenají, že je k dispozici otevřený zdrojový kód software, operačního systému, programu či webové stránky a podobně. Zdrojový kód je kód, se kterým mohou počítačoví programátoři manipulovat, aby změnili způsob, jakým software funguje. V běžné praxi je například možné na libovolných webových stránkách zobrazit zdrojový kód stránky, každá webová stránka má k dispozici zdrojový kód. Zobrazením zdrojového kódu webové stránky lze nejsnáze běžnému uživateli přiblížit pojem zdrojový kód. Kód lze z webové stránky zkopírovat, upravit, vložit do něj pár řádků a znovu jej vystavit na web pod jinou doménou. Zdrojové kódy jsou v podstatě řádky textu sestavující software. Podstatou open-source je hlavně svoboda, svobodně do kódu nahlédnout, kód upravit, rozšířit, opravit chybu, zkontrolovat kód, přidat funkci, přepracovat část kódu a sdílet jej dále pro komerční i nekomerční účely. Je naprosto transparentní a veřejný, každý může přispět. Dokonce vznikli i online repozitáře pro úpravu a opětovné nahrání kódu zpět do repozitáře, nejznámější online repozitář je Github.

Na počátku tzv. hnutí svobodného software stál Richard Matthew Stallman z projektu GNU, hnutí bylo základem pro pojem open-source. Jeho vize byla programy svobodně vyvíjet. Už tehdy (v 80. letech) Stallman říkal, že svoboda není zadarmo, open-source neznamená nutně zdarma (Co je open source: kompletní průvodce s historií, licencemi a využitím, 2025). Jedná se o komunitní dílo, programy nevznikají v korporátních studiích ale na různých cloudech, veřejných úložištích či online repozitářích, kde každý uživatel (či pokud má-li uživatel k tomu dostatečné oprávnění) má možnost kód upravit. Github je online repozitář s pracovními postupy založenými na Git neboli na verzování daného kódu (What is Open Source Software (OSS)?, 2024). Komunitní uživatelé jsou nejen schopni kód upravit a vytvořit tak další verzi ale mohou jej i v závislosti na licenci dále šířit. Veškerá práce je prováděna veřejně v online repozitáři, kde jej může kdokoliv editovat. Na vývoji se podílí vývojáři z celého světa. Vývojáři do open-source projektů často investují svůj volný čas bez nároku na odměnu, jsou pouze dobrovolnými přispěvateli. Výsledný produkt ve většině případů nebude komerčně vydávaný a navíc je možné jej dále šířit, dle licence samozřejmě nebo pravidel pro účast na projektu. Nekomerčnost ale není pravidlem, autoři open-source projektů mohou za své dílo požadovat peníze. Problém však nastává v případě, kdy open-source licence vyžaduje, aby byl při prodeji zveřejněn i zdrojový kód. Někteří programátoři tak zjistili, že daleko výhodnější je účtovat uživatelům peníze za služby a podporu open-source softwaru, než za samotný software. Jejich software tak zůstává zdarma a peníze vydělávají podporou, certifikacemi nebo jinými službami. Společnost Red Hat a linuxová distribuce Red Hat Enterprise Linux (RHEL) je toho krásným příkladem (Co je open source: kompletní průvodce s historií, licencemi a využitím, 2025). Open-source projekty se také těší vysoké spolehlivosti a bezpečnosti. Největší nevýhodou menších open-source projektů je časová náročnost, vývojáři přece jen s výsledným produktem nespěchají, oficiální datum vydání často k dispozici není a také se může stát, že projekt upadne v zapomnění, zůstane ve verzi beta nebo se již neudržuje a nedokončí se. Přece jen záleží na komunitě a odhodlání jednotlivých členů komunity. Open-source software je také bez záruky nebo ochranou odpovědností.

Výhody jsou hlavně nízké náklady, rychlost aktualizací oproti proprietárnímu software a transparentnost. Příkladů je celá řada, například Linux, Mozilla Firefox, VLC přehrávač, Ruby on Rails, Bash, Kubernetes, Blockchain, jQuery, Node.js a mnoho dalších. Ačkoliv si mnohdy ani neuvědomujeme na počátku počítačové historie (hlavně značnou část internetu), byli projekty vytvořené právě na open-source technologiích, mnohé z nich používáme dodnes, příkladem může být webová aplikace Apache nebo některé cloudové služby. V podstatě každý kdo nyní využívá internet těží i z open-source software (Co je open source: kompletní průvodce s historií, licencemi a využitím, 2025).

Na druhé straně existuje software s uzavřeným zdrojovým kódem, kde korporace či menší firmy musí zaplatit programátory (mimo jiné) a tak je jejich produkt komerčně vydáván. Což je také důvod k uzavření zdrojového kódu, aby firma nepřišla o peníze. Pokud se odbočí od tématu a vezme se například v úvahu kolik peněz stojí vytvoření počítačové hry, je nemožné ji vydat s otevřeným zdrojovým kódem. Jedná se o tzv. proprietární software neboli software chráněný autorskými či licenčními právy a upravovat kód mohou pouze jeho autoři. Mezi největší výhody oproti open-source je rychlost tvorby projektu, v drtivé většině se projekty dokončí a oznámeno je i datum vydání. Mezi příklady uzavřeného zdrojového kódu patří například Microsoft Office, Adobe Photoshop nebo MacOS, příkladů je opět mnoho. Produkty open-source jsou však často spolehlivější ale jsou vykoupeny delším časem vývoje. S nástupem moderních platforem typu Github i velké korporace začali podporovat komunitní projekty a přispívají nejen finančně ale i kódem. Spolupráce open-source komunit s velkými korporacemi je čím dál častější, příkladem může být Kubernetes, open-source projekt, který se stal standardem pro orchestraci kontejnerů.

Open-source je tedy vyvíjen komunitně, zatímco proprietární software pouze jeho autory. Stejně rozdílné jsou i licence v používání softwaru. Proprietární software vymezuje podmínky, které omezují jeho úpravy nebo distribuci. Open-source umožňují používání, úpravy a sdílení software se specifickými podmínkami pro každý typ.

1.2.2 Softwarové licence

Při využívání každého software se musí souhlasit s licenčními a právními podmínkami. Podmínky jsou dané autory a musí se respektovat. Licenční podmínky všeobecně jsou rovněž chráněny zákonem a jejich porušení je trestné. Licencování software lze rozdělit na dva hlavní proudy, proprietární a open-source. Existují však i software s dvojí licencí, například ERP podnikové systémy mohou obsahovat databáze s licencí GNU GPL a doplňky s proprietární licencí (Co je open source: kompletní průvodce s historií, licencemi a využitím, 2025). Dnes lze nalézt komponenty Open-source ve většině komerčních software. Proprietární software zahrnuje ve většině případech software s uzavřeným zdrojovým kódem a vysloveně mnohé zakazují včetně případných úprav, distribuci kódu či šíření částí samotného programu. Licence proprietárního softwaru se víceméně vymezuje pouze na jeho používání. Open-source licence naopak umožňují úpravy a sdílení. Open-source Initiative (OSI) je nezisková organizace, která dohlíží na projekt Open-source a spravuje Open-source licence. Pokud se vývojář rozhodne například pro projekt ve veřejném repozitáři jako je Github, musí si vybrat a deklarovat Open-source licenci s definicí otevřeného zdrojového kódu, kde musí být jasně vymezená práva a povinnosti týkající se užívání, úprav a distribuce.

Kód je obvykle dodáván s licencí. Licence definuje, co uživatel může a nemůže se softwarem provádět. Některé licence umožňují používat a distribuovat kód pro jakýkoli účel, zatímco jiné vyžadují zaznamenávání všech změn. Další licence mohou stanovit, že kopie zdrojového kódu jsou zdarma a jsou k dispozici pro veřejné použití. Copyright je autorské právo v jehož rámci může autor omezit použití díla. Richard Stallman při vytváření licence GNU GPL přišel s termínem Copyleft. Copyleft není opakem copyrightu. Copyleft znamená, že upravený software se musí vydat pod stejnou licencí (What is Open Source Software (OSS)?, 2024). Licencí jsou stovky. Mezi dnešní, nejoblíbenější a zároveň nejběžnější Open-source licence patří (tabulka je však spíše ukázková s pár příklady pro pochopení, open-source licencí je skutečně mnoho):

Tab. 1: Přehled vybraných open-source licencí

Licence	***Typ***	***Charakteristika***
MIT	Tolerantní	Kód lze upravit a distribuovat, lze jej využít v komerčních produktech, musí však zachovat autorská práva (copyright) a kopii samotné licence. Minimální omezení. Nejtolerantnější a nejrozšířenější svobodná licence.
Apache 2.0	Tolerantní	Charakteristika je podobná jako MIT licence, vyžaduje však zaznamenání všech změn a řeší patenty.
BSD	Tolerantní	Podobná licence jako MIT. Je však volnější, musí se pouze zachovat autorská práva (copyright).
GNU LGPL	LGPL	Lesser GPL. Určeno pro knihovny, umožňuje propojení i s komerčními aplikacemi. Upravená knihovna musí být zachována pod LGPL licencí.
GNU GPL v2	Copyleft	Zdrojový kód musí být dostupný a veřejný. Odvozená díla musí být vydávána pod stejnou licencí a za stejných podmínek. Software lze prodat, kupující si ponechává svobodu. Předchozí autoři musí být uvedeni.
GNU GPL v3	Copyleft	Podobné jako GPLv2. Navíc však řeší patenty a je kompatibilní s Apache 2.0. Zdrojový kód nemusí být zpřístupněn veřejnosti.
Public Domain/CC0	Veřejná doména	Neomezené použití, úpravy a marketing. Autoři se vzdávají práv nebo se jejich dílo stalo volnou doménou.

Zdroj: Microsoft Copilot: váš AI pomocník (2025)

1.2.3 Svobodný software

Pokud se zmiňuje pojem open-source software, je nutné zmínit i pojem svobodný software. Lidé si často oba dva pojmy pletou a myslí si o nich, že se jedná o stejnou věc či je omylem zaměňují. V podstatě se není čemu divit, oba termíny označují téměř stejné a rozdílné jsou jen a pouze ve filozofii, ne v technických detailech. Pro úplnost je však nutné rozdíly popsat. Když Richard Stallman zakládal v roce 1983 projekt GNU, ve kterém bylo cílem vytvořit svobodný

operační systém podobný UNIXU, ale bez proprietárního kódu a poté v roce 1985 založil Free Software Foundation (FSF), k podpoře vývoji svobodného softwaru tak rozšířil i jeho filozofii a přesně v tomto momentu definoval čtyři filozofická pravidla svobodného softwaru (Microsoft Copilot: váš AI pomocník, 2025). Pravidla jsou:

- Svoboda spouštět program k jakémukoli účelu,
- Svoboda studovat, jak program funguje, a upravovat ho,
- Svoboda redistribuovat kopie,
- Svoboda vylepšovat program a sdílet úpravy.

Čtyři základní svobody definice svobodného softwaru jsou základní uživatelská práva. Uživatel musí mít kontrolu nad softwarem. Zatímco Open Source Initiative (OSI) stanovila v roce 1998 tzv. Open Source Definition, která má 10 bodů (Microsoft Copilot: váš AI pomocník, 2025). Body jsou:

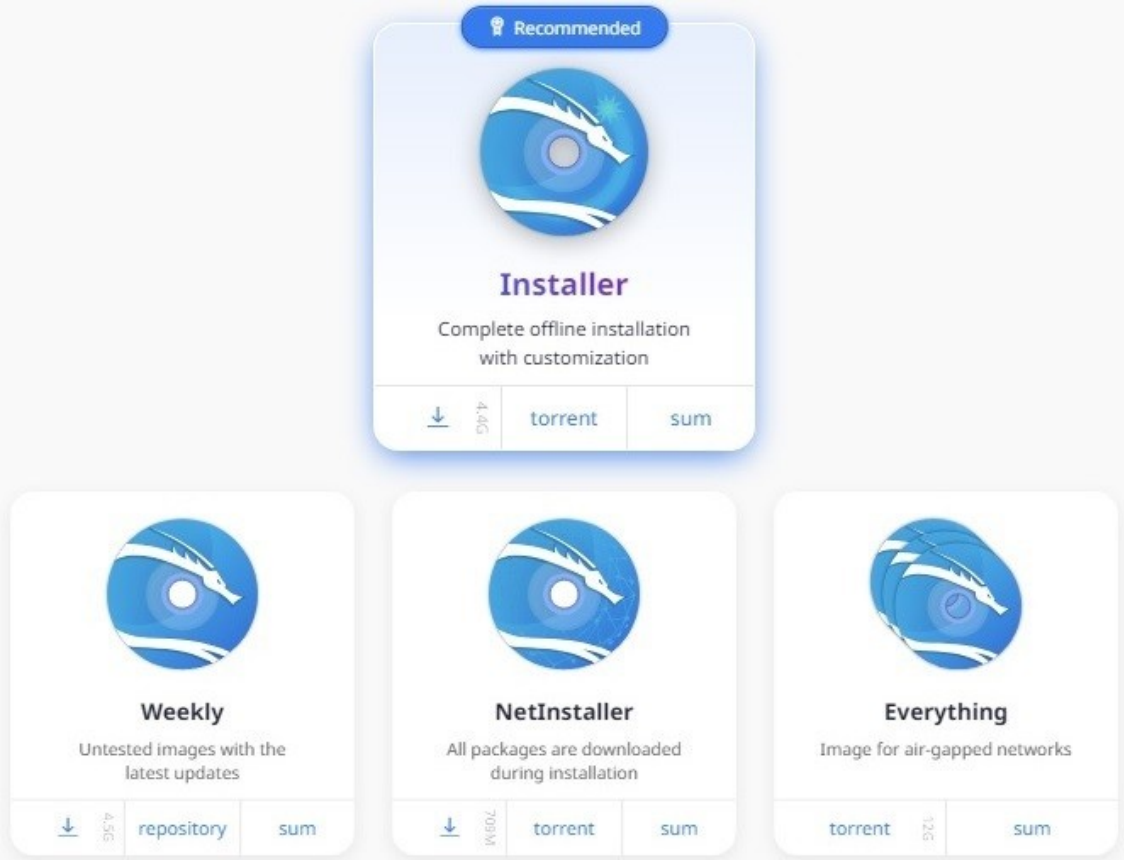
- Volná redistribuce – software lze volně prodávat nebo darovat,
- Zdrojový kód – musí být dostupný ke studiu a úpravám,
- Odvozená díla – licence musí umožňovat změny a nové projekty,
- Integrita zdrojového kódu – licence může vyžadovat patch formu, ale nesmí bránit úpravám,
- Žádná diskriminace osob či skupin – licence nesmí omezovat podle identity,
- Žádná diskriminace oblastí použití – nesmí zakazovat konkrétní obory (komerce, výzkum),
- Distribuce licence – práva platí pro všechny uživatele bez dalších smluv,
- Licence nesmí být vázána na produkt – platí i mimo konkrétní produkt,
- Licence nesmí omezovat jiný software – nesmí nutit ostatní software být open-source,
- Technologická neutralita – licence nesmí být vázána na konkrétní technologii či rozhraní.

U deseti bodů definice OSI se jedná hlavně o praktické využití vývoje software a komunitní spolupráci tak, aby software mohl být otevřeně vyvíjen, sdílen a používán. Obě definice vyžadují otevřený zdrojový kód, možnost úprav a redistribuce. Svobodný software formuluje práva uživatele jako politické hnutí, zatímco open-source technický vývoj jako vývojový model. Svobodný software definovaný Free Software Foundation (FSF) je směr filozofický, open-source definovaný Open Source Initiation (OSI) je směr praktický. Proto pojmy open-source software a svobodný software nejsou totožné. Dokonce se vedou i spory mezi fanoušky obou táborů. Slova zakladatele GNU projektu Richarda Stallmana k tématu jsou „svobodný software je politické hnutí, open-source je vývojářský model“.

1.3 Distribuce Kali Linux

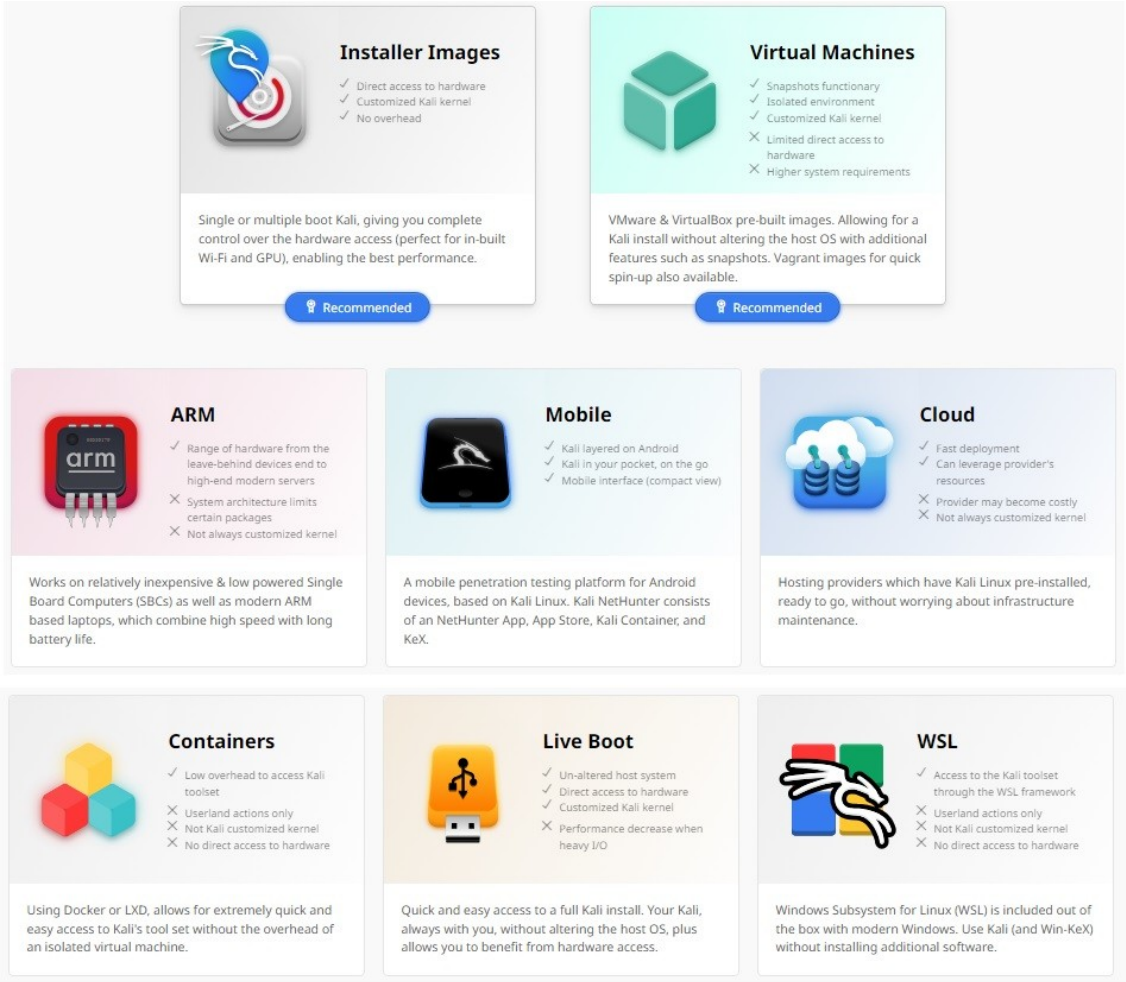
Linuxová distribuce Kali Linux je speciální pokročilá distribuce zabývající se nejen penetračním testováním neboli etickým hackováním ale i digitální forenzikou, výzkumem bezpečnosti či reverzním inženýrstvím. Kali Linux je open-source linuxová distribuce postavená na rodové linii linuxové distribuce Debian a vyvíjí ho a financuje společnost Offensive Security. Většina linuxových distribucí má svou rodovou linii, linie však není pravidlem. Distribuce obsahuje více jak 600 předinstalovaných nástrojů související s kybernetickou bezpečností (například Nmap,

Wireshark, Metasploit, Maltego a Burp Suite), které jsou ihned připraveny k použití a je zdarma, stejně jako většina open-source linuxových distribucí. Aktualizace Kali Linux jsou založeny na systému Rolling release, což znamená, že každých čtvrt roku vychází průběžné aktualizace bez nutnosti reinstalovat celý operační systém. Pro aktualizaci operačního systému Kali Linux stačí zadat pár příkazů do terminálu. Je spustitelný na mnoha různých platforem od běžných počítačů přes servery, virtuální stroje, cloudy či mobilní zařízení. Je distribuován jak v 32bitových, tak 64bitových ISO obrazech pro systémy založené na Intelu. Je také dostupný jako předpřipravené virtuální stroje pro VirtualBox a VMware. Obsahuje také různá desktopová prostředí například Xfce, GNOME nebo KDE. Spousta ISO obrazů pro různé platformy jsou volně ke stažení na stránkách Kali, stejně tak jsou dostupné ke stažení i různé varianty instalátorů. Ke stažení instalátoru vývojáři nabízí klasickou aktualizovanou stabilní čtvrtletní offline plnou instalaci, měsíční netestovanou nestabilní verzi s nejnovějšími aktualizacemi, net instalátor, který stáhne vše potřebné během instalace, či je možné všechny ISO obrazy stáhnout v podobě torrentů (k tomu je však zapotřebí torrent klient, například BitTorrent, Vuze nebo jiné). Další variantou Kali Linux je Kali Purple, což je speciální varianta Kali Linuxu zaměřená na defenzivní kyberbezpečnost pro blue-team čili na ochranu, monitoring a detekci útoků. Zatímco klasické Kali Linux je určené pro ofenzivní účely red-teamu. Ke všem staženým souborům je k dispozici kontrolní součet dat. Další alternativní linuxové distribuce věnované penetračnímu testování jsou Parrot OS, Black Arch (Arch Linux), BackBox či Wifislax (Slackware). Název Kali je odvozen od hinduistické bohyně času a změny, což symbolizuje plynulost nástrojů zahrnutých v distribuci. O instalaci systému, jeho aktualizaci a prvním spuštěním se bude bakalářská práce zabývat v praktické části (Microsoft Copilot: váš AI pomocník, 2025).



Obr. 3: Možnosti stáhnutí obrazu Kali Linux dle instalátoru

Zdroj: Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution (2025)



Obr. 4: Možnosti stáhnutí obrazu Kali Linux dle platformy

Zdroj: Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution (2025)

1.3.1 Historie Kali Linux

Kali Linux je založený na znalostech a zkušenostech z několika předchozích projektů, které vytvářeli různí vývojáři v malých týmech. Proto se distribuce Kali Linux formovala dlouhé roky. Za první projekt, ze kterého vyšel Kali Linux, se označuje Whoppix, což je také zkratka pro WhiteHat Knoppix. Byl založen na platformě linuxové distribuce Knoppix, což lze odvodit podle jména, a byl také první distribucí zaměřenou na etické hackování. Whoppix vyšel z verze 2.0 na verzi 2.7. Z Whoppixu se později vytvořil další projekt s názvem WHAX, což je zkratka pro WhiteHat Slax. Změna názvu na WHAX byla způsobena změnou základního operačního systému z Knoppixu na Slax. WHAX vyšel v začínající verzi v3 jako navazující verze na předchozí verze Whoppix 2.7. Přibližně ve stejnou dobu byl vyvíjen podobný operační systém Auditor Security Collection, zkracovaný pouze na Auditor, který také vzešel z Knoppixu. Později došlo je spojení zmíněných linuxových distribucí, Auditor, Knoppix a WHAX, do jednoho celku, čímž vznikla linuxová distribuce BackTrack. BackTrack byl založen na platformě Slackware od verze v1 do verze v3 ale později přešel na platformu Ubuntu s verzemi v4 až v5. Kali Linux využil všech zkušeností získaných veškerými předchozími projekty a byl oficiálně vydán v roce 2013 jako náhrada za distribuci BackTrack. Zároveň začal využívat stabilní vydání

Debianu jako základní platformy (Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution, 2025).

Tab. 2: Stručný přehled vývoje Kali Linux

Datum	***Projekt***	***Základní operační systém***
30. 8. 2004	Whoppix v2	Knoppix.
17. 7. 2005	WHAX v3	Slax.
26. 5. 2006	BackTrack v1	Slackware Live CD 10.2.0.
6. 3. 2007	BackTrack v2	Slackware Live CD 11.0.0.
19. 6. 2008	BackTrack v3	Slackware Live CD 12.0.0.
9. 1. 2010	BackTrack v4 (Pwnsauce)	Ubuntu 8.10 (Intrepid Ibex).
10. 5. 2011	BackTrack v5 (Revolution)	Ubuntu 10.04 (Lucid Lynx).
13. 3. 2013	Kali Linux v1 (Moto)	Debian 7 (Wheezy).
11. 8. 2015	Kali Linux v2 (Sana)	Debian 8 (Jessie).
16. 1. 2016	Kali Linux Rolling	Debian Testing.

Zdroj: Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution (2025)

1.3.2 Obecné vlastnosti

Zde je uveden souhrn několika všeobecných vlastností Kali Linux.

- Je zaměřen na bezpečnost a penetrační testování, primárně pro etické hackování, analýzu zranitelností a digitální forenziku,
- Obsahuje stovky předinstalovaných nástrojů pro testování sítí, webů, hesel, Wi-Fi, malware analýzu a další,
- Je postavený na platformě operačního systému distribuce Debian a proto používá balíčkovací systém APT,
- Je Rolling release model, aktualizace probíhají čtvrtletně, nástroje i systém jsou tak stále aktuální,
- Je optimalizovaný pro profesionály, výchozí nastavení systému je orientované pro bezpečnostní experty a pokročilé uživatele,
- Podporuje nejrůznější platformy například x86, ARM, virtuální stroje, mobilní zařízení (Kali NetHunter) i cloud,
- Nabízí Live USB režim a možnost běžet bez instalace (možnost spustit systém z USB bez zásahu do disku), což je ideální pro testování nebo forenziku,
- Obsahuje režimy jako Kali Undercover (příkaz kali-undercover), integraci s Tor, šifrování persistentního úložiště a jiné,
- Udržuje vlastní speciální repozitáře, obsahující stovky bezpečnostních nástrojů a jejich aktualizace. Díky tomu jsou nástroje dostupné rychleji než v běžných distribucích.
- Součástí systému jsou i vlastní nástroje a skripty vyvinuté přímo od vývojářů Kali Linux Offensive Security, například kali-tweaks, kali-undercover, kali-menu,
- Kali nabízí metabalíčky (např. kali-linux-top10, kali-linux-large), které umožňují snadno instalovat celé sady nástrojů dle zaměření uživatele,

- Kali Linux má detailní oficiální dokumentaci, návody, příklady konfigurací a aktivní komunitu, což usnadňuje práci začátečníkům i profesionálům,
- Kali podporuje automatizaci pomocí Ansible skriptování a předpřipravené konfigurace, což zjednodušuje nasazení ve větších prostředích.

1.3.3 Použití Kali Linux

Rozsah použití distribuce Kali Linux je skutečně široký, krom klasických výborných vlastností operačního systému Linux, se může chlubit zejména svým zaměřením, čímž si získal mnoho fanoušků. Obsahuje spoustu nástrojů, které umožní uživateli testovat zabezpečení sítí a aplikací, etické hackování, analýzu malwaru, digitální forenziku či výzkum zranitelností bez nutnosti jejich instalace. Spousta organizací jej používá pro komerční testování zranitelností systémů svých zákazníků. Kali Linux obsahuje mnoho nástrojů, které mohou být zneužity pro nelegální činnosti. Kali Linux je sám o sobě legální nástroj a jeho vývojáři při tom nenesou žádnou právní odpovědnost za jeho zneužití. Jasně uvádějí, že Kali Linux je určený pouze pro etické hackování, bezpečnostní testování, vzdělávání a forenzní analýzu. Kali Linux je zkrátka open-source software a vývojáři nenesou odpovědnost za zneužití jeho nástrojů. V licencích open-source je uvedeno, že software je poskytován tak jak je, bez záruky, bez odpovědnosti za škody a uživatel je tak plně odpovědný za své jednání. Což znamená, že pokud někdo zneužije Kali Linux k útoku, za útok zodpovídá pouze on a ne vývojáři. Vývojáři dokonce upozorňují na fakt, že Kali Linux není určený k páchání trestné činnosti, nechrání útočníka před odhalením, loguje a komunikuje jako běžný Linux. Kdo by jej chtěl zneužít, tak dělá na vlastní riziko (Microsoft Copilot: váš AI pomocník, 2025).

1.4 Analýza nástrojů Kali Linux

V kapitole analýza nástrojů Kali Linux se bude bakalářská práce zabývat analýzou vybraných nástrojů, které se poté využijí v praktické části bakalářské práce. Všechny analyzované nástroje jsou součástí distribuce Kali Linux a není tak nutná jejich dodatečná instalace. Bakalářská práce se bude zabývat testováním pouze vybraných nástrojů. Kompletní seznam nástrojů, jejich vlastností a oblasti použití je možný ke shlédnutí na webových stránkách distribuce Kali Linux (Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution, 2025). Přehled známých linuxových příkazů, které je možné uplatnit při penetračním testování, bude popsán na konci kapitoly.

1.4.1 Přehled a vlastnosti vybraných nástrojů

Pro lepší přehlednost nástrojů lze většinu nástrojů zařadit do kategorií dle využití.

Tab. 3: Přehled kategorií vybraných nástrojů Kali Linux a jejich popis

Kategorie	*Nástroje*	*Charakteristika*
Sběr informací	Nmap, Zenmap, Stealth scan, Dmitry, Maltego, Unicorscan.	Nástroje určené pro sběr informací a jejich následné formátování do dat, které lze dále použít.
Analýza zranitelností	Bed, Ohrwurm, Powerfuzzer, Sfuzz, Siparmyknife, Nikto, OpenVAS, Fierce, Netcat.	Nástroje určené ke kontrole zranitelností v systému. Informují o zranitelnostech.
Analýza webových aplikací	Burp suite, Httrack, SQLmap, Vega, Webscarab, Wpsscan, Zap, Skipfish, Owasp-zap, Wapiti, Xsspy.	Nástroje určené ke kontrole zranitelností a chyb webových stránek či aplikací.
Analýza databází	Bbqsl, SQLinjection, Oscanner, SQLninja, Tmscmd10g.	Nástroje sloužící k analýze databází z hlediska útoků a bezpečnostních problémů.
Útoky na hesla	Cewl, Crunch, Hashcat, Johnny the Ripper, Medusa, Ncrack.	Nástroje určené ke zpracování slov nebo hesel. Některé nástroje slouží ke sběru slov, jiné k útoku.
Bezdrátové útoky	Aircrack-ng, Fern-wifi-cracker, Kismet, Ghost Phisher, Wifite.	Nástroje sloužící k prolomení zabezpečení bezdrátových sítí. Například hesel, routerů, AP, MITM či sběru informací.
Reverzní inženýrství	Apktools, Ollydbg, Flasm, Nasm shell.	Nástroje určené pro manipulaci se zdrojovým kódem aplikace či software, čili k pochopení logiky zdrojového kódu.
Nástroje pro exploitaci	Metasploit, Searchsploit, Beef xss framework, Termineter, SET.	Nástroje sloužící k využití zranitelnost systémů.
Sniffing a Spoofing	Wireshark, Bettercap, Ettercap, Hamster, Driftnet, Responder, Macchanger.	Nástroje určené pro odposlech síťového provozu (sniffing) a podvržení identity (spoofing).
Post exploitace	MSF, Veil – pillage framework, Powersploit, Powershell empire	Post Exploitation. Nástroje zajišťující opětovný přístup v již hacknutých systémech nebo k udržení přístupu v systému.
Forenzní analýza	Autopsy, Binwalk, Galleta, Hashdeep, Volafox, Volatility.	Nástroje využívané k obnově informací ze systému nebo úložného zařízení. Používají se při vyšetřování

		kybernetického zločinu.
Nástroje pro report	Dradis, Faraday IDE, Pipal, Magictree, Metagoofil.	Nástroje určené k vytváření analýzy, reportu, po provedeném penetračním testu.
Sociální inženýrství	SET, Backdoor-f, U3-pwn, Ghost Phisher, Msf payload creator.	Nástroje sloužící pro sběr informací využívající lidského pochybení.

Zdroj: vlastní zpracování dle Kali Linux Tools (2025)

1.4.2 Wireshark

Wireshark je výkonný analyzátor síťového provozu a síťových protokolů. Zobrazuje části síťového provozu jako jsou informace o logických a fyzických adresách. Někdy je také nazývaný jako analyzátor paketů. Zachycuje pakety v reálném čase a zobrazuje dokonce i jejich obsah pokud není šifrovaný (18 Best Kali Linux Tools and How to Use Them, 2025). Dokáže zobrazit celou strukturu vybraného paketu od fyzické vrstvy až po aplikační vrstvu. Je možné jej rovněž využít pro řešení síťových problémů a odhalování podezřelých aktivit. Provoz sítě je možné filtrovat podle protokolu, IP adresy, portu a jiných parametrů. Filtrování umožňuje zúžit výsledky vyhledávání (25 Best Kali Linux Tools, 2023).

- **Kategorie -** Sniffing a Spoofing,
- **Alternativy** - Kismet, Tcpdump, Termshark, Bettercap, Intercepter-NG, Proxyman,
- **Možnosti** - Dokáže monitorovat sítový provoz v reálném čase, ukládá zachycené pakety (soubory PCAP) pro pozdější analýzu, převádí síťový provoz do obsahu čitelného pro člověka, kontroluje stovky různých protokolů, disponuje grafickým rozhraním.

1.4.3 Bettercap

Bettercap je síťový bezpečnostní framework navržen pro průzkum a útoky v bezdrátových i kabelových sítích. Zachycuje pakety a provádí útoky typu MITM (man-in-the-middle). MITM znamená útok na komunikaci mezi dvěma stranami, kde je útočník schopen komunikaci obou komunikujících sledovat, měnit nebo přesměrovat. Například když si dva lidé posílají zprávy a třetí člověk je schopen jejich komunikaci číst i upravovat. Bettercap obsahuje jak textové uživatelské rozhraní (CLI) tak i webové rozhraní (18 Best Kali Linux Tools and How to Use Them, 2025). Rovněž má vestavěný skriptovací základ. Je mladším a modernějším nástrojem nástroje Ettercap a byl navržen s cílem nahradit staré MITM nástroje. Vývoj Ettercap již delší dobou stagnuje a nevyvíjí se tak rychle, stále je ale klasikou v oblasti MITM útoků (25 Best Kali Linux Tools, 2023).

- **Kategorie -** Sniffing a Spoofing,
- **Alternativy** - Dsniff, Ettercap, Responder, Scapy, Evilginx2, MITMf, Yersinia,
- **Možnosti** - Funguje na wi-fi sítích, zařízení Bluetooth a bezdrátových zařízeních s rozhraním HID, podporuje útoky MITM s falešným využitím protokolů ARP, DNS a DHCP, jeho síťový sniffer dokáže shromažďovat přihlašovací údaje, může pasivně skenovat IP síťové hostitele, obsahuje rychlý skener portů.

1.4.4 Nmap

Nmap (Network Mapper) neboli mapovač sítě je základní hackerský nástroj s otevřeným zdrojovým kódem určený k prohledávání sítí. Odhaluje jaká zařízení a jaké služby v síti běží. Funguje tak, že odesílá nezpracované IP pakety a ty poté identifikují hostitele, služby, operační systémy i konfigurace firewallu. Dokáže tak zjistit ohromné množství informací včetně stavu portů. Dokáže i provádět test hesel hrubou silou a detekovat zranitelnosti systému (25 Best Kali Linux Tools, 2023).

- **Kategorie -** Sběr informací,
- **Alternativy** - Fing, Zenmap, Zmap, OpenVAS, Masscan, Unicornscan, Netdiscover,
- **Možnosti** - Vypíše otevřené, uzavřené nebo filtrované porty, identifikuje služby včetně jejich typů a verzí dle otevřeného portu, dokáže využít funkce skriptů ke kontrole konkrétních zranitelností, poskytuje vizuální grafické mapování sítě prostřednictvím rozhraní Zenmap, základní příkazy jsou Ping Sweep (-sn), Stealth sken (-sS), verze služeb (-sV) a jiné.

1.4.5 Aircrack-ng

Aircrack-ng je kompletní sada nástrojů pro posouzení zabezpečení bezdrátových sítí. Skládá se z několika programů. Obsahuje nástroj Airdecap-ng (dešifrování souborů s ochranou WEP a WPA), nástroj Airodump-ng (nástroj pro zachycení a shromažďování paketů, identifikaci AP a připojených klientů), nástroj Airtun-ng (tvorba virtuálních tunelových rozhraní), nástroj Besside-ng (prolamování WEP a WPA hesel), nástroj Airmon-ng (přepnutí síťové karty do monitorovacího režimu pro zachycení veškerého provozu), nástroj Aireplay-ng (vkládání rámců pro generování provozu a pro urychlení prolomení WEP) a nástroj Aircrack-ng (nástroj pro prolomení klíčů WEP a WPA/WPA2-PSK s analýzou zachycených paketů) (18 Best Kali Linux Tools and How to Use Them, 2025). Běžným způsobem testování sítě chráněné protokolem WPA2 je zachycení handshake (proces čtyřcestného ověřování při připojení klienta) a následné spuštění slovníkového útoku proti němu (25 Best Kali Linux Tools, 2023).

- **Kategorie -** Bezdrátové útoky,
- **Alternativy** - Hashcat, Fern wi-fi Cracker, Wifite, Reaver, Bully, Kismet, Ghost Phisher,
- **Možnosti** - Zaměření na bezdrátové lokální sítě 802.11, poskytuje nástroje příkazového řádku pro skriptování, provádí slovníkové útoky pomocí slovníku WEP a také fragmentační útoky.

1.4.6 John the Ripper

John the Ripper je rychlý a všestranný offline nástroj pro obnovení a prolomení hesel. Dokáže prolomit heslo pomocí několika metod na základě detekování typu hashe souboru. Umí použít různé metody k prolomení hesla. Při metodě slovníkových útoků projde seznam slov ze slovníku. Při metodě Brute-force útoku projde všechny možné kombinace znaků. Obsahuje však i další metody útoků na hesla (25 Best Kali Linux Tools, 2023). Používá různé hashe, šifry a šifrovací formáty. Za pomoci nástroje nazývaného pouze John dokáže John the Ripper nabídnout i grafické uživatelské rozhraní. Nejčastěji je nástroj John the Ripper použit při testování síly hesla.

- **Kategorie -** Útoky na hesla,
- **Alternativy** - Hydra, Hashcat, Patator, Medusa, Rainbow crack, Fail2ban, Ncrack,
- **Možnosti** - Automaticky identifikuje různé formáty hashů hesel, za pomoci Brute-force útoku vyzkouší všechny možné kombinace znaků (Incremental Mode), dokáže použít slovníkový útok proti hashům (Wordlist Mode), odhadne hesla použitím login/GECOS informace (Single Crack Mode), dokáže šifrovat soukromé klíče.

1.4.7 Burp Suite

Burp Suite je jeden z nejznámějších nástrojů určený ke skenování zranitelností webových aplikací. Zachycuje proxy komunikaci mezi prohlížečem a webovou aplikací. Což umožňuje kontrolovat, upravovat i přehrávat veškeré požadavky a odpovědi HTTP/HTTPS. Rovněž lze také odesílat upravené HTTP požadavky k odhalení zranitelností. Díky nástroji Burp Suite lze testovat známé zranitelnosti jako SQL injection, Cross-Site Scripting (XSS), falešné HTTP požadavky, přerušené ověřování a jiné (18 Best Kali Linux Tools and How to Use Them, 2025). Parametry nástroje lze ručně upravit. Burp Suite má grafické uživatelské rozhraní.

- **Kategorie -** Analýza webových aplikací,
- **Alternativy** - Nikto, OWASP ZAP, Nessus, Acunetix, Invicti, Fiddler,
- **Možnosti** - Komunitní edice je zdarma, zachycuje a upravuje webový provoz, podporuje statické i dynamické vyhledávání, lze ručně upravit a znovu odeslat požadavky pomocí opakovače, ve verzi Pro lze nastavit automatické skenování zranitelností a procházení aplikace.

1.4.8 OWASP ZAP

OWASP ZAP je open-source nástroj určený pro skenování webových aplikací a nejlepší nástroj na odhalení Cross-Site Scripting útoků. Funguje jako proxy mezi klientem a serverem a umožňuje zachytávat, upravovat i znovu odesílat požadavky HTTPS. Umožňuje pasivní a aktivní skenování. Pasivní automaticky vyhodnocuje zachycený provoz. Aktivní skenování generuje testovací payloady a cíleně vyhledává zranitelnosti typu XSS, SQLi či chybnou konfiguraci. V OWASP ZAP jsou i moduly pro fuzzing parametrů, analýzu struktury webu, automatizované útoky a nástroje pro ruční testování. Kombinace automatických i manuálních technik dělá z OWASP ZAP vhodný nástroj pro detekci XSS zranitelností, včetně DOM-based variant. OWASP ZAP obsahuje grafické rozhraní. (18 Best Kali Linux Tools and How to Use Them, 2025).

- **Kategorie -** Analýza webových aplikací,
- **Alternativy** - Burp Suite, Wapiti, Nikto, Skipfish, WFuzz, Arachni, WhatWeb, Gobuster,
- **Možnosti** - Kontrola a úprava provozu jako zachycený proxy, automatické vyhledávání zranitelností pomocí aktivních a pasivních skenerů, automatické procházení webu k nalezení všech jeho stránek, posílá nečekaná data za účelem nalezení možných chyb pomocí Fuzzeru.

1.4.9 SQLMap

SQLmap je open-source nástroj pro vyhledávání zranitelností typu SQL Injection. SQLmap zranitelnost plně automatizovaně vyhledá. Při podezření zranitelnosti webové aplikace způsobem útoku SQL Injection stačí nasměrovat SQLmap na adresu webového odkazu. Vyhledá injekce založené na booleovských časových, sjednocovacích, chybových a vrstvených dotazech. Funguje s MS SQL serverem, MySQL, PostgresSQL, Oracle i s dalšími databázovými servery (25 Best Kali Linux Tools, 2023).

- **Kategorie -** Analýza webových aplikací,
- **Alternativy** - Commix, TheDoc, Nycto-dork, jSQL injection, BeEf, DorkPlus, Havij,
- **Možnosti** - Stáhne kompletní databázi, případně konkrétní tabulku či sloupec, poskytuje shell operačního systému na určitých systémech, zahrnuje možnost prolomení hesla pomocí slovníkových útoků, podporuje eskalaci uživatelských oprávnění.

1.4.10 Další nástroje

V Linuxovém světe existuje bezpočet dalších nástrojů, příkazů a doplňků. Všechny další použitelné nástroje jsou v seznamu nástrojů na stránkách Kali Linux. Mezi vybrané užitečné linuxové příkazy obsažené v každé linuxové distribuci bez nutnosti instalace se řadí (Microsoft Copilot: váš AI pomocník, 2025):

- traceroute / tracepath - cesta paketů skrz síť,
- ping - ověření dostupnosti hostu, latence,
- dig - detailní DNS dotazy,
- nslookup - základní DNS dotazy,
- host – rychlé DNS dotazy,
- ip neigh – ARP cache,
- netstat – aktivní spojení, porty,
- lsof -i – procesy otevřené na portech,
- arp – ARP tabulka,
- hostnamectl – informace o systému,
- iptables / nft – zobrazení firewallových pravidel, analýza filtrování provozu,
- ss – moderní náhrada netstat, zobrazení otevřených portů a spojení,
- arping – ověření dostupnosti hostu na úrovni ARP v lokální síti,
- mtr – kombinace ping a traceroute, živá diagnostika trasy,
- tcpdump – zachytávání a analýza síťových paketů,
- tshark – CLI verze Wiresharku pro analýzu paketů,
- iftop – živý přehled síťového provozu,
- nethogs – zobrazení síťového provozu podle procesů,
- iptraf-ng – interaktivní monitorování síťové aktivity,
- curl – testování HTTP(S) požadavků a API,
- wget – ověření dostupnosti a stahování souborů,
- telnet – jednoduché testování dostupnosti portů,
- nc (netcat) – diagnostika spojení a testování portů.

1.5 Analýza technik penetračního testování

V kapitole analýza technik penetračního testování se bude bakalářská práce zabývat analýzou vybraných technik, které se poté využijí v praktické části bakalářské práce. Všechny analyzované techniky jsou zde uvedeny vzhledem k analyzovaným nástrojům distribuce Kali Linux (Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution, 2025). Bakalářská práce se bude zabývat testováním pouze vybraných technik. Kybernetických útočných technik je celá řada a v mnoha případech k tomu ani nejsou zapotřebí linuxové nástroje. Přehled známých kybernetických útočných technik bude popsán na konci kapitoly.

1.5.1 Přehled a vlastnosti vybraných technik

Pro lepší přehlednost penetračních technik lze většinu technik zařadit do kategorií dle využití.

Tab. 4: Přehled kategorií vybraných technik a jejich popis

Kategorie	***Techniky***	***Charakteristika***
Síťové útoky	DoS/DDoS	Zahlcení síťového provozu.
Diagnostika sítě	Analýza paketů, sledování síťového provozu	Zjišťování stavu sítě, problémů a komunikace mezi zařízeními.
MITM útoky	ARP spoofing, DNS spoofing	Útoky, kdy se útočník staví mezi komunikující strany.
Skenování portů a služeb	Port scanning, service discovery	Zjišťování otevřených portů, služeb a verzí softwaru.
Bezdrátové útoky	Útoky na Wi-Fi, zachytávání handshake	Testování zabezpečení bezdrátových sítí.
Útoky na autentizaci	Brute-force, dictionary attack	Prolamování hesel z hashů nebo Wi-Fi handshake.
Útoky na webové aplikace	XSS, SQL Injection, CSRF, DOM-based XSS, UNION-based, error-based, blind SQLi	Testování zabezpečení webů, hledání zranitelností.
Skenování webů	Enumerace, testování API, crawling	Zjišťování struktury webu a potenciálních slabin.
Malware útoky	Ransomware, trojské koně	Škodlivý software zaměřený na data nebo systém.
Sociální inženýrství	Phishing, vishing	Manipulace uživatele k získání údajů.

Zdroj: vlastní zpracování dle Microsoft Copilot: váš AI pomocník (2025)

1.5.2 Monitoring sítě

Monitoring neboli analýza síťového provozu je technika umožňující zachytávat datové pakety přímo na síťovém rozhraní a zkoumat jejich strukturu na úrovni jednotlivých protokolů. Zachycené pakety lze dekódovat podle vrstev modelu TCP/IP, je možné tak vidět fyzické rámce, IP hlavičky, TCP/UDP segmenty i samotná aplikační data. Analýza také odhaluje, jak probíhá navazování spojení (handshake), jaké porty a služby komunikují, jaké protokoly jsou používány a zda nedochází k opakovanému odesílání paketů, jejich ztrátám nebo k jiným chybám v komunikaci. Díky tomu lze vypozorovat špatné nastavení konfigurace, latenci, zahlcení sítě nebo podezřelé chování, jako je například skenování portů. Monitoring paketů tak poskytuje detailní přehled reálného provozu v síti a dokáže přesně diagnostikovat problémy i odhalovat bezpečnostní incidenty (Microsoft Copilot: váš AI pomocník, 2025).

- **Kategorie** - Diagnostika sítě,
- **Možnosti** - Získání nešifrovaných přihlašovací údajů, session cookies,
- **Doporučení** - Útočník nezjistí obsah paketů díky šifrované komunikaci HTTPS či TLS, při použití VPN se skryje skutečný provoz i cílové adresy, stačí používat firewall a IDS/IPS k blokaci skenování či omezit otevřené porty a služby.

1.5.3 MITM útoky

Man-in-the-midle (MITM) útok je široce znám jako útok na odposlech. Při útoku typu MITM se útočník zapojí do komunikace obou stran. Útočník je tak schopen dostat se do konverzace mezi uživatelem sítě a webovou aplikací (nachází se ve středu komunikace, dle názvu útoku) a dochází k narušení samotného šifrování (Nejčastější typy kybernetických útoků, 2023). Dostane se tak mezi odesílatele a adresáta zpráv a ani jedna strana nic nepozná. Útočník je tak schopen nejen číst odeslanou komunikaci ale rovněž s ní i manipulovat, upravit ji, narušit její přenos, filtrovat či ji jednoduše ukrást (například může vynést z komunikace důvěrné informace či je stáhnout). Při hijackingu útočník získá unikátní ID uživatele a při komunikaci s webovými stránkami se vydává za ukradenou identitu uživatele, může se tak přihlásit i do účtů oběti (10 nejčastějších typů kybernetických útoků, 2022). Útočníci se také mohou vložit mezi zařízení uživatele a síť na nezabezpečené veřejné síti Wi-Fi. Útočník je rovněž schopen do zařízení uživatele nainstalovat malware (10 typů kybernetických útoků, o kterých byste měli vědět v roce 2022 a rady, jak jim zabránit, 2022).

- **Kategorie** - MITM útoky,
- **Možnosti** - Odposlech, přesměrování a úprava komunikace, vynesení informací,
- **Doporučení** - Stačí dbát na bezpečnost webových stránek a používat na svých zařízeních šifrování, útokům MITM lze zabránit protokolem HTTP Strict Transport Security (HSTS), také je dobré se vyvarovat veřejným wi-fi sítím.

1.5.4 Skenování portů

Skenování portů (Port Scanning) je technika, která se používá pro zjištění otevřených portů. Funguje tak, že odesílá síťové pakety na různé porty cílového zařízení a podle odpovědi zjistí, zda je port otevřený (reaguje na dotaz a je aktivní), uzavřený (zařízení odpoví, že port není otevřen) a filtrován (port neodpovídá, může být blokován firewallem). Na otevřeném portu

je aktivní, čili poslouchá nějaký spuštěný program či službu. Což znamená , že daná služba na zařízení je dostupná zvenčí a může přijímat požadavky. Naopak uzavřený port je port, na kterém nejsou žádné programy, služby či daemony spuštěny. S větším počtem dostupných otevřených portů se dá zanalyzovat více konkrétnějších informací o cílovém zařízení. Šikovný hacker může pomocí nástrojů po skenování portů poměrně snadno nalézt v počítačové síti provozované služby, díky kterým je tak schopen identifikovat tu největší zranitelnost v síti, tu poté může zneužít pro cílený útok k nalezení způsobu průniku do systému (exploit). Pokud je však správně nakonfigurovaný firewall nemusí se tak podařit hackerovi zjistit stav portů. Porty slouží jako vstup pro jednotlivé služby běžící na zařízení. Každý port (číslovaný od 0 do 65535) je přiřazen určitému protokolu nebo programu. Například port 80 HTTP (webový provoz), port 443 HTTPS (šifrovaný webový provoz), port 22 SSH (vzdálená správa serverů) a podobně. Existuje několik různých typů skenování. TCP Scanning (klasické spojení přes TCP handshake, je snadno detekovatelné), SYN Scanning (poloviční handshake, rychlé a nenápadné, používané při penetračních testech), UDP Scanning (složitější, UDP nemá potvrzovací mechanismus, pomalejší a méně přesné), ACK scanning (lze zjistit pouze je-li port filtrovaný nebo nefiltrovaný, neurčíme otevřený nebo uzavřený port, dobré k zjišťování firewallu). V reálném světě je port scanning často prvním krokem útoku. Zároveň je ale i běžnou obrannou technikou (Port Scanning (Skenování portů), 2025).

- **Kategorie** - Skenování portů a služeb,
- **Možnosti** - Zjištění stavu portů a služeb, vyhledávání zranitelnosti k průniku,
- **Doporučení** - Použít firewall blokující nevyžádané příchozí pakety a odpovídá minimálně, aktivovat IDS/IPS (dokáže detekovat a omezit podezřelé skeny), povolit pouze nezbytné porty a ostatní uzavřít nebo filtrovat, skrývat služby za VPN (nebudou tak dostupné z veřejného internetu) či omezit přístup pomocí ACL a povolit tak jen konkrétní IP adresy.

1.5.5 Bezdrátové útoky

Bezdrátové sítě představují dnes jeden z nejrozšířenějších způsobů připojení k internetu. Používají k přenosu dat rádiové vlny s přesně danými frekvencemi (například 2,4 GHz či 5 GHz). Přístupový bod (Access Point neboli AP) vysílá signál v určité frekvenci a klientská zařízení se k němu připojují. Zařízení si mezi sebou vyměňují různé informace a data a jejich komunikace probíhá prostřednictvím rádiových vln (vlny jsou modulovány a přijímány routerem). Zařízení, která tvoří bezdrátovou síť, jsou kompatibilní se standardy IEEE 802.11. Rádiové protokoly rodiny 802.11 fungují v pásmu volného přístupu, což znamená, že se signál šíří všemi směry a na danou frekvenci se může připojit kdokoliv v dosahu i bez použití speciálních nástrojů. Právě otevřenost rádiového prostředí je hlavním důvodem, proč jsou bezdrátové sítě tak zranitelné. S nástupem průmyslu 4.0 a Internet of Things se počty zařízení schopných se připojit k bezdrátové síti rapidně zvyšují. Klasické drátové připojení ethernetovým kabelem je v dnešní době na ústupu, je možné se dokonce setkat s počítači či notebooky bez možnosti připojení RJ-45 konektorem ethernetového kabelu. Oblíbenost a počet bezdrátových sítí se odrazili i na množství útoků. Nejčastějšími typy útoků jsou útoky na hesla, na zastaralé protokoly WEP či WPA/WPA2, útoky na WPS, Rogue Access Point (falešný přístupový bod) nebo Evil Twin (záměrně vytvořená kopie legitimní sítě) a další.

Nejčastěji používaným nástrojem v oboru je Aircrack-ng a automatizační skripty jako Wifite2 nebo WEF (Typy útoků na Wi-Fi, 2023).

- **Kategorie** - Bezdrátové útoky,
- **Možnosti** - Odposlech, zachytávání handshake paketů, vytváření falešných AP,
- **Doporučení** - Silná hesla delší než 15 znaků, zakázat WPS v nastavení routeru, pravidelně měnit a zpřísňovat hesla, používat šifrování WPA3, případně WPA2-PSK se silným heslem, aktualizovat firmware přístupových bodů a routerů nebo omezit dosah signálu tak, aby nepřesahoval zbytečně mimo objekt.

1.5.6 Prolomení hesla

Útoky na hesla míří na nejrůznější uživatelská hesla uživatelských účtů s cílem dostat se k informacím uživatele. Jedná se o jeden z nejčastějších útoků kyberkriminality a zároveň jeden z nejčastějších útoků na zaměstnance pomocí sociálního inženýrství. Lidé by měli mít silná hesla, nikomu je nesdělovat a často je měnit (10 nejčastějších typů kybernetických útoků, 2022). Existují různé typy útoků na hesla, útoky s použitím hrubé síly (program zkouší různé sady znaků), slovníkové útoky a další. Útočník má k dispozici celou řadu nástrojů a způsobů včetně metod sociálního inženýrství. Keylogger je program určený k útoku, kdy se útočník pomocí speciálního softwaru pokusí sejmout údery prstů na klávesnici, respektive vyčíst z klávesnice použité znaky (10 typů kybernetických útoků, o kterých byste měli vědět v roce 2022 a rady, jak jim zabránit, 2022).

- **Kategorie** - Útoky na autentizaci,
- **Možnosti** - Přístup k účtům, informacím a osobním údajům, jednání za uživatele,
- **Doporučení** - Hesla často aktualizovat, používat silná alfanumerická hesla se speciálními znaky, neopakovat stejné heslo pro více uživatelských účtů, na každý účet jiné heslo.

1.5.7 Skenování webů

Nástroje, určené ke skenování webů, fungují jako interceptující proxy, což znamená, že se vloží mezi prohlížeč a server a umožní zachytávat, upravovat a znovu odesílat HTTP či HTTPS požadavky. Díky tomu lze sledovat, jak aplikace pracuje s uživatelskými vstupy, jaké parametry přijímá a jak reaguje na nestandardní nebo škodlivé vstupy. Součástí skenování je také automatizované procházení aplikace, neboli crawling, při kterém nástroj zkoumá všechny dostupné stránky. Provádí se aktivní testování, kdy se zkouší různé typy útoků jako jsou injection, manipulace s cookies, testování hesel nebo konfigurace serveru. Jedná se o kombinaci pasivní analýzy, která sleduje provoz a hledá podezřelé prvky a aktivní analýzy, která zasahuje do komunikace a testuje reakce aplikace. Proxy vrstva umožňuje testerovi ručně upravovat požadavky, měnit parametry, zkoušet různé payloady a ověřovat, zda aplikace správně filtruje vstupy nebo zda nevrací citlivé informace. (Microsoft Copilot: váš AI pomocník, 2025).

- **Kategorie -** Skenování webů,
- **Možnosti** - Odhalení zranitelností a konfigurace serveru, obcházení oprávnění,

- **Doporučení** - Zachování správného formátu na vstupu a odstranění nebezpečných znaků nebo kódu, nastavení správné konfigurace serveru a omezení informací, webové aplikace je také nutné často aktualizovat, dále používat silná hesla a chránit session cookies.

1.5.8 Cross-Site Scripting

Cílem Cross-Site Scripting (XSS) je útok vložením škodlivého skriptu do zdrojového kódu webové stránky či aplikace, díky neošetřenému vstupu vývojáři, buď přímo na server nebo prostřednictvím odkazu. Ve své podstatě je XSS velmi podobný SQL Injection. Když si uživatel stránku načte, prohlížeč skript spustí. Prohlížeč nepozná, že nepochází od legitimního webu. Po načtení webové stránky může útočník stránku pozměnit s cílem získat citlivé osobní informace uživatele nebo se dokonce za uživatele i vydávat. Běžný uživatel nemá šanci si všimnout škodlivého kódu ve zdrojovém kódu webové stránky či aplikace. Nejnáchylnější k útokům XSS jsou fóra, blogy a podobně čili weby umožňující uživateli přidat vlastní obsah. Možným nasazením interních aplikací přes internet nebo cloudových služeb se útoku nevyhnou ani velké organizace (Microsoft Copilot: váš AI pomocník, 2025).

- **Kategorie** - Útoky na webové aplikace,
- **Možnosti** - Krádež cookies a identity, přesměrování či úprava stránek,
- **Doporučení** - Escapování výstupu zajistí bezpečné zakódování uživatelského vstupu, Content Security Policy (CSP) omezuje spouštění neautorizovaných skriptů, HttpOnly cookies zabrání tomu, aby JavaScript mohl získat přístup k údajům uživatele.

1.5.9 SQL Injection

SQL (Structured Query Language) je dotazovací jazyk, který se používá při správě databází. SQL Injection znamená vložit škodlivý kód na databázový server webové služby pomocí jazyka SQL a databázový server kód (dotaz) vykoná, také je možné vložit škodlivý kód do vyhledávacího pole webové stránky. Škodlivý kód se obvykle napíše a vykoná do formuláře nebo do vyhledávacího pole webové stránky díky programátorsky neošetřenému vstupu. Velice podobně funguje i XSS útok (Nejčastější typy kybernetických útoků, 2023). SQL dotaz databázový server provede a vytvoří, přečte, upraví nebo odstraní data uložená v databázi (10 nejčastějších typů kybernetických útoků, 2022). V horším případě je útočník schopen převzít kontrolu (získat administrativní práva) nad celým serverem a získat přístup k datům či vynést důvěrné informace. Útočníci útočí zejména na databáze uchovávající osobní údaje, citlivá data uživatelů a čísla bankovních karet. SQL Injection je svojí jednoduchostí a rychlostí jednou z nejpopulárnějších metod jak využít zranitelnosti systému, o tom hovoří i statistiky. 42 % pokusů o útoky na veřejné systémy byli založené na SQL Injection (10 typů kybernetických útoků, o kterých byste měli vědět v roce 2022 a rady, jak jim zabránit, 2022).

- **Kategorie** - Útoky na webové aplikace,
- **Možnosti** - Získání dat či administrátorských práv, úprava a mazání záznamů,
- **Doporučení** - Lze použít systém detekce narušení (je navržen k neoprávněného přístupu k síti), na webové službě lze provést ověření dat zadaných uživatelem neboli ošetřit vstupy.

1.5.10 Další techniky

Ve světe kybernetických útoků existuje mnoho dalších útočných technik. Internet poskytuje mnoho zdrojů zabývající se informacemi o kybernetických útocích. Mezi další známé techniky lze zařadit (Microsoft Copilot: váš AI pomocník, 2025):

- Adware – nevyžádané reklamy a sledování chování,
- ARP Spoofing / ARP Poisoning – podvržení ARP odpovědí v lokální síti,
- Botnet – zneužití sítě infikovaných zařízení k útokům,
- Brute force API – opakované pokusy o přístup k rozhraním,
- Clickjacking – skryté přesměrování kliknutí na jiný prvek,
- DNS spoofing / poisoning – podvržení DNS odpovědí,
- DoS – útok z jednoho zdroje, který přetíží službu,
- DDoS – zahlcení služby masivním množstvím provozu,
- Drive-by download – automatické stažení malware při návštěvě stránky,
- Email spoofing – falšování odesílatele e-mailu,
- File Inclusion (LFI/RFI) – načtení škodlivého souboru do aplikace,
- HTTP Request Smuggling – manipulace s HTTP hlavičkami mezi proxy a serverem,
- HTTP Response Splitting – vložení škodlivé odpovědi rozdělením HTTP hlaviček,
- ICMP Flood – zahlcení cíle ICMP echo požadavky,
- Insider attack – útoky provedené zaměstnancem nebo osobou s přístupem,
- Keylogger – zaznamenávání stisků kláves,
- Malware – škodlivý software obecně,
- Phishing – podvodné získávání údajů pomocí falešných zpráv,
- Ransomware – zašifrování dat a požadavek výkupného,
- Rootkit – skrytí přítomnosti útočníka v systému,
- Session hijacking – převzetí aktivní relace uživatele,
- Smishing – phishing prostřednictvím SMS,
- Spyware – skryté sledování uživatele a sběr dat,
- Supply-chain attacks – napadení dodavatelského řetězce,
- SYN Flood – zahlcení serveru neúplnými TCP handshaky,
- Trojan – škodlivý kód maskovaný jako legitimní program,
- UDP Flood – zaplavení cíle velkým množstvím UDP paketů,
- Vishing – phishing pomocí telefonních hovorů,
- Watering hole – infikování webu, který oběť často navštěvuje,
- Whaling – phishing zaměřený na vedení firmy,
- Worm – samostatně se šířící malware bez interakce uživatele,
- Zero-day attack – zneužití dosud neopravené zranitelnosti.

1.6 Metasploit Framework

Projekt Metasploit je projektem počítačové bezpečnosti, který poskytuje informace o bezpečnostních zranitelnostech a pomáhá při penetračním testování. Projekt vytvořil H. D. Moore v roce 2003 jako přenosný síťový nástroj založený na Perlu. V roce 2007 byl přepsán v Ruby. V roce 2009 byl projekt Metasploit koupen společností Rapid7. Jeho

nejznámějším podprojektem je open-source Framework Metasploit, modulární platforma, která umožňuje psát, testovat a spouštět exploit kódy. Exploit kódy lze vytvořit nebo je lze převzít z databáze obsahující nejnovější dostupné a modulární exploity. Metasploit nyní zahrnuje více jak 1677 exploitů na 25 platformách, včetně Androidu, PHP, Pythonu, Javy, Cisco a dalších. Framework také obsahuje téměř 500 payloadů. Stačí získat informace o cíli pomocí skenování portů nebo skenování zranitelností a najít tak cestu do sítě. Pak už jde jen o výběr správného nástroje, exploitu nebo payloadu. Metasploit poskytuje exploity, payloady, pomocné funkce, enkodéry, posluchače, shellcode, post-exploitační kód a nops. Framework Metasploit má schopnost ukončit práci bez detekce (A step-by-step guide to the Metasploit Framework, 2024).

Framework Metasploit obsahuje sadu nástrojů, které lze využít k testování bezpečnostních zranitelností, sítí, provádění útoků a vyhýbání se detekci. Některé nástroje jsou již zabudované přímo ve Frameworku. V jádru je Metasploit Framework platforma běžně používaných nástrojů, které poskytují kompletní prostředí pro penetrační testování a vývoj exploitů. Framework je v informatice označení pro rámec, neboli kostru, která již poskytuje nástroje, knihovny, moduly, rozhraní a pluginy, aby se s nimi dalo rychleji pracovat a jednodušeji tak něco vyvíjet. Lze si jej představit jako balíček komponent, které připraví uživateli zázemí pro práci (Microsoft Copilot: váš AI pomocník, 2025).

Konzole Metasploit Framework (MSFConsole) je nejoblíbenějším příkazovým rozhraním pro přístup a práci s Metasploit Framework (MSF). Poskytuje vše v jednom a umožňuje přístup ke všem možnostem dostupným v MSF. MSFConsole umožňuje například skenovat cíle, využívat zranitelnosti a sbírat data. Pro spuštění MSFConsole slouží příkaz ./msfconsole, případně se musí instalovat balíček příkazem bundle install a poté použit příkaz ./msfconsole (Metasploit Framework, 2025).

Stejně jako mnoho dalších open-source nástrojů v informatice lze i Metasploit využít k nelegálním činnostem. Je kompatibilní s mnoha operačními systémy a v některých je i předinstalovaný. V Kali Linux je přímo zabudovaný. Od akvizice Metasploit Frameworku přidal Rapid7 otevřenou proprietární, komerční, edici nazvanou Metasploit Pro s plnou automatizací a pokročilými funkcemi. Metasploit Framework se stal hlavním nástrojem pro vývoj a zmírňování zneužití (A step-by-step guide to the Metasploit Framework, 2024).

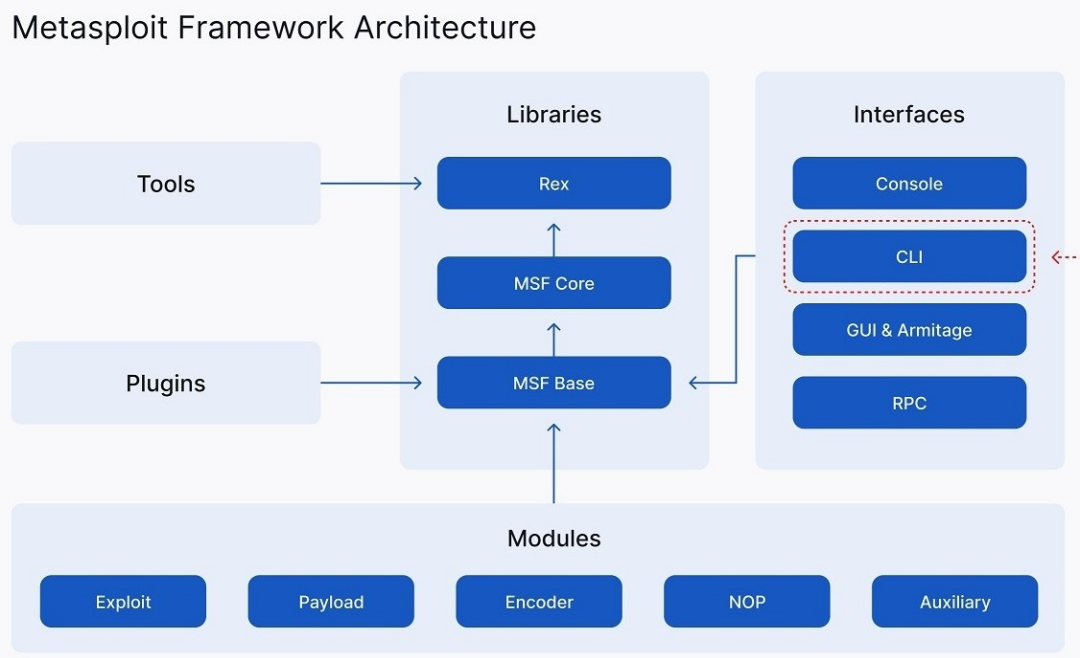
1.6.1 Architektura Metasploit

Framework Metasploit má svou architekturu, která se skládá z částí jako jsou rozhraní, knihovny, moduly, nástroje a pluginy, které spolu navzájem spolupracují a nabízejí tak komplexní platformu pro penetrační testování (What is Metasploit?, 2025).

- **Moduly** – moduly Metasploit jsou předpřipravené skripty s konkrétním účelem a funkcemi, které již byly vyvinuty a otestovány. Jsou organizovány do různých typů, včetně pomocných exploitů a payloadů. Slouží k provádění úkolů jako skenování a zneužívání cílů. Existuje několik hlavních typů modulů, jako jsou payloady (shellkódy odeslány s exploitem běžící po infiltraci na cílovém operačním systému tak, aby usnadnili navázat zpětné spojení), exploity (příkazy nebo kódy využívající zranitelnosti systému), posty (post-exploity umožňující provádět hlubší sběr informací a dále pronikat do systému po zneužití), enkodéry (zakrývají payloady během přenosu,

zajišťují úspěšné doručení do cílového systému bez detekce antivirovým softwarem), NOP (vytvářejí náhodné sekvence bajtů a udržují velikost zatížení při pokusu o exploit, aby obešly systémy detekce), auxiliaries (pomocné moduly zahrnují skenování zranitelností a portů, fuzzery, sniffery) a další moduly,

- **Rozhraní** - rozhraní jsou různé platformy, přes které mohou uživatelé přistupovat k Metasploit Frameworku. K dispozici jsou celkem čtyři rozhraní. MSFConsole (nejpoužívanější rozhraní, konzole umožňující přístup prostřednictvím příkazového řádku), MSFWeb (rozhraní založené na prohlížeči), Armitage (rozhraní založené na Javě s GUI) a RPC (vzdálený hovor, umožňující uživatelům programově řídit Metasploit Framework pomocí HTTP založených vzdálených procedur volání),
- **Knihovny** - obsahují různé funkce Metasploit Frameworku umožňující uživatelům spouštět exploity bez nutnosti psaní dalšího kódu. Existují celkem tři knihovny. REX (umožňuje většinu základních úkolů), MSFCore (poskytuje společné API), MSFBase (poskytuje uživatelsky přívětivé API),
- **Nástroje a pluginy** - doplňky k Frameworku Metasploit, které rozšiřují jeho funkčnost.



Obr. 5: Architektura Metasploit Framework
Zdroj: What is Metasploit? (2025)

1.7 Penetrační testování

Penetrační testování známé také jako Pen Test neboli etické hackování je testování bezpečnosti softwarových i webových aplikací, síťové infrastruktury a internetových služeb. Testování se provádí formou simulací reálného útoku jak zevnitř tak zvenčí pomocí stejných nástrojů a technik, které používají skuteční útočníci. Jeho cílem není vyřešit bezpečnostní problémy ale odhalit zranitelnosti a prověřit tak skutečnou úroveň zabezpečení. Výsledným cílem je zamezit či zmírnit riziko zranitelnosti systému. Pen Test je schopen prověřit nejen

technologickou část ale i organizační, za pomoci sociálního inženýrství je schopen prověřit zaměstnance. Jak se například zaměstnanec zachová při obdržení podezřelého emailu. Vyhodnocením testu je závěrečná zpráva s podrobnou analýzou systému, jeho zranitelnostech zařazených do stupnice dle rizikovosti a doporučení vedoucí k jejich eliminaci. Analýza je prováděna z pohledu útočníka. Jednat se o může o celou řadu pochybení, například slabá hesla, polozapomenuté služby, phishing, otevřené porty, staré verze systému a podobně. Penetrační testování je nutné pro velké podniky s velkým množstvím dat jako jsou finanční, vojenské či státní instituce kde se soustřeďují důležité informace o klientech, hesla, zakázky a důvěrná tajemství. Únik dat je pro instituce velmi nežádoucí. Důležité je penetrační testování také pro prevenci či pro již hacknutý systém, aby se zjistili přetrvávající hrozby a rizika (Penetrační testy infrastruktury sítě, webových aplikací a internetových služeb, 2025).

Na trhu je poměrně mnoho firem zabývající se penetračním testováním na objednávku, mezi nejčastější oblasti Pen Testu patří (Penetrační testování, 2025):

- **Penetrační testování webových aplikací** – testování webových stránek či internetových služeb,
- **Externí penetrační test perimetru sítě** - útok na systém zvenčí neboli průnik z internetu do interní sítě,
- **Penetrační testování vnitřní sítě** - útok na interní síť z vnitřní sítě, možné zneužití dat a údajů vlastními zaměstnanci, dodavateli či partnery,
- **Penetrační test wi-fi sítě** - prověření bezdrátové sítě,
- **Sociální inženýrství** - prověření chování a reakce zaměstnanců.

Dále firmy nabízejí penetrační testování mobilních aplikací, kontrolu zdrojového kódu, testy IoT zařízení, IT infrastruktury a jiné (Penetrační testování, 2025). K penetračním testům používají organizace oficiální metodiky a testování je rozděleno do několika fází.

1.7.1 Metody testování

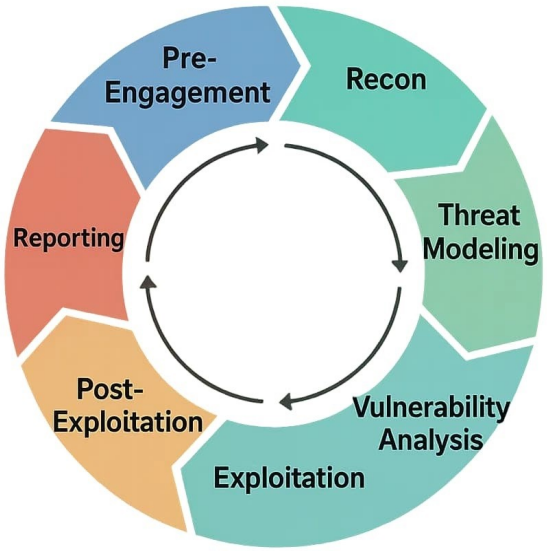
Mezi nejčastější oficiální metody používané organizacemi při penetračním testováním jsou metodiky OWASP, OSSTMM a PCI-DSS (Penetrační testování, 2024). Metodiky fungují jako strukturované postupy, podle kterých se penetrační test provádí. I když každá metodika má svůj určitý účel a zaměření všechny slouží hlavně k tomu, aby postup měl jasně dané fáze, pravidla a výsledky. Zkrátka aby test nebyl chaotický. Postup při určité metodice může být například sběr informací, analýza zranitelností, exploatace a závěrečná zpráva. Zajišťují tak strukturovanost kroků a zároveň slouží k tomu, aby se na nic nezapomnělo. Rozdíl jednotlivých metodik je hlavně v jejich zaměření. Opět jich existuje celá řada, zde je uvedeno pár příkladů metodik penetračního testování (Penetration Testing Methodology, 2025):

Tab. 5: Vybrané metodiky penetračního testování

Název	*Zkratka*	*Popis*
OWASP WSTG	Web Security Testing Guide	Metodika zaměřená na testování webových aplikací. Obsahuje strukturovaný seznam testů pro oblasti webu.
OWASP MSTG	Mobile Security Testing Guide	Metodika zaměřená na testování mobilních aplikací. Obsahuje postupy a doporučení pro zabezpečení mobilních aplikací.
PTES	Penetration Testing Execution Standart	Popisuje komplexní celý proces Pen Testu rozdělených do sedmi fázi, nástroje a technické postupy.
OSSTMM	Open Source Security Testing Methodology Manual	Univerzální metodika pro bezpečnost sítí, systémů, procesů i fyzické bezpečnosti. Využívá se pro infrastrukturu a síťové testy.
PCI-DSS	Penetration Testing Guidance	Standart pro organizace pracující s platebními kartami. Testy se zaměřují zejména na systémy zpracovávající platební data.

Zdroj: Microsoft Copilot: váš AI pomocník (2025)

Z metodik jsou nejznámější metodologie OWASP, které se dále mohou dělit podle zaměření. Například OWASP ASVS, OWASP Top 10, OWASP Mobile Top 10, OWASP SAMM a další (Microsoft Copilot: váš AI pomocník, 2025). Metodiky výše vyjmenované jsou standardizované a normované, mají svůj přesně daný postup. Jednotlivé kroky a postupy metodik penetračního testování jsou velmi podobné fázemi penetračního testování.



Obr. 6: Sedm fází penetračního testování metodologie PTES

Zdroj: Penetration Testing Methodology (2025)

1.7.2 Fáze testování

V praxi se u různých organizací setkáváme s fázemi testování, fáze slouží k tomu aby organizace měla svůj předem daný postup od začátku do konce. Zároveň je nabízí jako portfolio svých služeb zákazníkům. Fáze v nejběžnější a základní formě mohou být (Penetrační testy infrastruktury sítě, webových aplikací a internetových služeb, 2025):

- Plánování a příprava,
- Sběr informací (Reconnaissance),
- Vyhodnocení slabých míst (analýza zranitelností),
- Využití slabin (ověření zranitelností neboli exploitace (Exploitation)),
- Prověření získaných dat (post-exploitace (Post-Exploitation)),
- Vyhodnocení a závěrečná zpráva (report).

Takto vypadají základní fáze penetračního testování. Je možné se setkat s jiným slovním označením jednotlivé fáze ale princip je stejný. Rozdíl oproti metodikám penetračního testování je v jejich dodržování. Fáze jsou proměnné, organizace zabývající se penetračním testováním na objednávku může mezi fáze penetračního testování přidat pár kroků či postupů dle libosti a pak je předkládat zákazníkům. Fáze penetračního testování jsou spíše firemním procesem. Metody jsou však předem jasně dané, jejich kroky a postupy jsou neměnné. Pokud se organizace zaváže, že například během testování webové aplikace budou dodržovat metodiku OWASP WSTG, musí kroky a postupy metodiky dodržet. Je však možné ji zahrnout do vlastních fází penetračního testování jako jeden z kroků (Penetrační testování, 2024).

Mezi nejdůležitějšími fázemi každé objednávky penetračního testu od jakékoliv organizace jsou bezesporu implementace oprav zranitelností a retest. Zde je však také nutno podotknout, že implementaci oprav zranitelností nezajišťuje organice provádějící penetrační testování ale klient. Organizace tak pouze čeká na opravy analyzovaných zranitelností a poté následuje retest. Jedině takto se získá komplexně zabezpečený systém s napravenými zranitelnostmi. Od organizace provádějící penetrační testování je tak dobré požadovat (Penetrační testování, 2024):

- Penetrační test (fáze penetračního testu uvedené výše),
- Oprava zranitelností na straně klienta,
- Retest,
- Aktualizované vyhodnocení a závěrečná zpráva (report).

Obr. 7: Grafická ukázka fáze testování s přidanou metodikou OWASP

Zdroj: Penetrační testování (2024)

1.7.3 Režimy testování

Penetrační testování simuluje reálný útok útočníků na systém, pomocí stejných technik a nástrojů jako používají při útoku skuteční útočníci. Útok (testování) se provádí ve třech formách podle skutečných dispozic útočníků. Tři formy režimy testování jsou (Penetrační testování, 2025):

- **Black box** – Black box je režim formy testování z pohledu útočníka, kdy útočník (penetrační tester) nemá žádné předchozí znalosti o cílené IT infrastruktuře, hardwaru nebo softwaru. Režim testování Black box se snaží co nejpřesněji napodobit reálný útok a je velice časově náročný. Úkolem je najít všechny zranitelnosti, nasimulovat veškeré možné útoky a najít veškeré způsoby narušení systému. Je však nutné být opatrný aby útočník (penetrační tester) nepoškodil samotný systém. Black box je nejvíce používaná forma režimu testování a je nejpodobnější klasickému hackerskému útoku. Při tomto režimu testování se jedná o externí útok na systém, útok z internetu na interní síť,
- **White box** – White box je opakem Black box režimu testování. Útočník má k dispozici veškeré materiály, znalosti i plný přístup (včetně znalostí zdrojových kódů, konfigurace a architektuře), které plně využívá k útoku na systém či na jeho části. Je plně informován o tom jak systém funguje a zná jeho zabezpečení. White box režim testování se většinou v praxi provádí ještě před spuštěním systému či software do plného provozu ke zjištění zranitelnosti a slouží jako preventivní testování. White box režim testování lze přirovnat k útoku dlouholetého zaměstnance a jedná se o interní útok zevnitř, z interní sítě (respektive útočníka nezajímá jak by k systému přistupoval zvenčí),
- **Gray box** – Gray box je kombinací obou předchozích režimů testování. Útočník má k dispozici pouze určité znalosti o cílené IT infrastruktuře či softwaru, které lze uplatnit.

Má pouze částečné informace a nemá úplný přístup k systému. Gray box režim testování je zaměřen většinou na jeden jediný cíl, ke kterému má útočník pouze omezené informace. Testy se provádí bez nutnosti odstávky systému. Příkladem může být útočníkův přístup k interní síti organizace ale zároveň naprostá neznalost konfigurace serverů. Jeho úkolem je proniknout dál do systému přes otevřený port či jinak. Režim Gray box lze přirovnat k útoku bývalého zaměstnance.

1.7.4 Zranitelnosti

Zranitelnost je v odborných zdrojích definována jako slabé místo v systému, postupech zabezpečení systému, vnitřních kontrolách nebo implementaci, která by mohla být zneužita. Případné teoretické zneužití zranitelnosti se stává hrozbou pro systém. Může se nacházet nejen v software, v hardware ale i v lidech, kteří jsou součástí informačního systému. Lidský faktor nese velký podíl na celkové zranitelnosti systému, nedostatečné povědomí o kybernetické bezpečnosti se často stává terčem útoků. Oblastí útoku v oboru sociálního inženýrství je mnoho od phishingu, sociálních médií, hesel či e-mailů. Mezi další nejčastější zranitelnosti systému se řadí implementační chyba v kódu, díky které je schopen útočník narušit systém, získat přístup či data. Chyba v kódu je obvykle implementována v software během vývoje jako programátorská chyba. Samozřejmě bez úmyslné implementace vývojáři. Speciálním případem zranitelnosti je pojem zranitelnost nultého dne. Pojem zranitelnost nultého dne se používá pro neznámou zranitelnost a neexistuje pro ní záplata nebo ochrana. Útok, zneužívající zranitelnost nultého dne se označuje jako útok nultého dne. Pro útok nultého dne není obrany, většinou se na ní přijde až když je pozdě (Microsoft Copilot: váš AI pomocník, 2025).

Jednotlivé zranitelnosti se dělí podle stupně závažnosti, pro které se používá hodnocení zranitelností. Od žádné, nízké, střední, vysoké až po kritickou zranitelnost. Mezi praktické příklady zranitelnosti mohou být slabá hesla, nezabezpečené konfigurace serverů, otevřené porty, špatně nastavené firewally, zastaralé software i protokoly a jiné. Typická ukázka zranitelnosti může být například neošetření vstupu vývojářem v databázi, kde ve formuláři, vyhledávacím poli či URL odkazu se vloží škodlivý SQL kód. Kód se vykoná na databázovém serveru a útočník získá přístup k datům nebo je může změnit (Microsoft Copilot: váš AI pomocník, 2025). Veškeré zranitelnosti jsou schopni útočníci maximálně vytěžit a dostat se tak do systémů. Pokud je konkrétní zranitelnost zneužita cíleně, opakovaně a automatizovaně, pak se zneužití označuje jako exploit. Exploit je speciální automatizovaný kód čili program, skript nebo technika využívající cíleně konkrétní zranitelnost a cílem je získat přístup, obejít bezpečnostní opatření a podobně. Pokud zranitelnost existuje a je teoretická možnost, že bude zneužita, pak zmiňovanou možnost označujeme hrozbou.

1.7.5 Hrozby a rizika

Hrozba je okolnost či událost, která je schopna způsobit škodu organizaci například ovlivněním organizačních operací, systémů nebo jednotlivců prostřednictvím informačního systému. Hrozba představuje potencionálního útočníka nebo událost, která může zneužít slabinu v systému (zranitelnost). Z praktického hlediska lze uvést například hackera, který se pokouší dostat do sítě, malware nebo phishing potencionálně se pokoušející využít slabinu v systému

avšak slabina ještě není využita. Existuje pouze hrozba, že bude využita. Hrozba neznamená útok, útok nastává až tehdy kdy hrozba využije zranitelnosti. Lze si představit zloděje (hrozba), otevřené okno (zranitelnost) a pokud zloděj vleze do otevřeného okna je využita zranitelnost hrozbou a z události se stává incident. Exploit by byla konkrétní technika využití zranitelnosti (oknem) hrozbou (zlodějem), technika jako lezení po žebříku, skok do okna a podobně.

Hrozby lze dělit na hrozby vnitřní (hrozba přichází zevnitř organizace a jedná se o sabotáž) a vnější (hrozba přichází zvenku, mimo organizaci a zde se jedná se o hacking). Hrozby lze také dělit na základě dopadu na působení organizace na aktivní (dochází ke změně stavu systému a je narušená integrita dat, případně dostupnost služeb či jiné) a pasivní (nedochází ke změně stavu systému ani integrity dat či dostupnosti služeb a podobně ale dochází například k úniku informací) (Microsoft Copilot: váš AI pomocník, 2025).

Hrozba je příčinou bezpečnostní události neboli bezpečnostního incidentu. Pokud dochází k uskutečnění hrozby a zneužití zranitelnosti, jedná se o bezpečnostní událost. Pokud zmíněná událost naruší dostupnost služeb, důvěryhodnost organizace či její data jedná se o bezpečnostní incident. Bezpečnostní incident má zároveň i dopad na fungování napadené organizace, například dopad finanční, ztráta věrohodnosti či podobně a řadí se mezi aktivní hrozby (Microsoft Copilot: váš AI pomocník, 2025).

Pokud existuje určitá hrozba schopna způsobit škodu, představuje kybernetické riziko. Riziko je risk, že bude hrozba využita. Rizika by měla být důkladně zanalyzována a zároveň by mělo být rozhodnuto o způsobu jeho zvládání pomocí bezpečnostních opatření. Snaha zavést bezpečnostní opatření je nejlepším způsobem prevence jak se vyhnout či úplně zamezit rizika využití hrozby útočníkem.

1.7.6 Útočníci

K tématu je zapotřebí uvést, že pojem hacker je sice společností zažité pro počítačového člověka provádějící kybernetické útoky avšak zcela zcestné. Slovo hacker vzniklo v letech v 60. - 70. letech na univerzitě MIT v USA a pojem označoval programátora, který tvoří programy, sepisuje kódy a nemá nic společného s kybernetickými útoky. Oproti tomu pojem cracker bylo označení pro člověka, který se neustále zkoušel nabourat do systému, obcházet bezpečnostní opatření a provádět škodlivé útoky. Hacker tvoří, cracker ničí. V současném našem slovníku je rovněž zažito říkat cracknout hru či cracknout program. Slovo crack se vžilo pro upravený binární soubory schopný zpřístupnit software. Význam slov média naprosto překroutila a protože slovo hacker je nejspíše více atraktivní, v mediálním prostoru slovo hacker zvítězilo (Microsoft Copilot: váš AI pomocník, 2025).

V současné době je slovo hacker označení pro počítačového experta schopného vstoupit do systému i bez potřebných přístupových oprávnění čili provést kybernetický útok. Kybernetický útok je jakýkoli druh škodlivé činnosti, který se pokouší shromažďovat, narušovat, popírat, degradovat nebo zničit prostředky informačního systému nebo samotné informace. Svou podstatou je kybernetický útok naplněním konkrétní hrozby využívající konkrétní zranitelnost systému. Kybernetický útok prováděný hackery (kyberzločinci) se označuje jako hacking neboli česky hackování. Většinou se jedná o škodlivý kybernetický útok poškozující systém či podobně, což ale není podmínkou. Někteří hackeři při průniku do systému odhalení zranitelnosti nahlásí nebo tak jednají se souhlasem majitele systému. Ze zmíněného důvodu se

hackeři rozdělují do několika kategorií podle jejich cílů (Hackeři – proti komu se bráníme?, 2021):

- **White Hat hacker** – neboli také etický hacker. Etický hacker je protiklad Black Hat hackera. Jeho cílem je se také dostat do systému a použít přitom naprosto stejné nástroje či techniky. Rozdíl však spočívá v tom, že White Hat hacker testuje systém pouze se svolením organizace, která mu dala svolení svůj systém prověřit. Firmy si takto testují bezpečnost svých systémů a jedná se o penetrační testování. Pokud se etickému hackerovi při penetračním testování podaří přes nějakou zranitelnost dostat do systému, žádnou škodu nezpůsobí. Naopak výstupem jeho práce je report o zranitelnostech a jak je odstranit. Po odstranění zranitelností systém etický hacker opět prověří, zda je v pořádku. White Hat hackeři neboli etický hackeři jsou součástí firem, které si najímají organizace za účelem prověření své kybernetické bezpečnosti (Hackeři – proti komu se bráníme?, 2021),
- **Black Hat hacker** – je opakem White Hat hackera. Cílem Black Hat hackera je objevit slabiny systému bez vědomí organizace či majitele systému za účelem neoprávněného přístupu či jeho poškození. Jeho jediným cílem je systém poškodit či ho zneužít pro vlastní obohacení. O nalezené slabině okamžitě informuje ostatní hackery. Jedná se o nelegální činnost. Jako příklad poškození může být krádež peněz z účtu, nainstalovaný program pro sledování webkamery, šifrování dat a podobně (Hackeři – proti komu se bráníme?, 2021),
- **Gray Hat hacker** – Gray Hat hacker je kombinací obou předchozích typů hackerů. Často se ovšem do zmiňované kategorie řadí hackeři s nejasnými úmysly. Pokud někdo objeví slabinu v systému a zneužije ji bez vědomí jeho majitele ale zároveň nezpůsobí žádné škody a zranitelnost nahlásí, pak jeho úmysly nejsou jednoznačné. Může se jednat o útočníka, který se naboural do systému jen, aby zvýšil povědomí o jeho slabinách. Na druhou stranu se může chovat jako White Hat hacker hacker, ale jeho úmysly jsou neetické. I kdyby jeho úmysly byli čistě etické, bez souhlasu majitele systému se stále jedná o nelegální činnost (Hackeři – proti komu se bráníme?, 2021).

Pojem Hat hacker vzniklo z westernových Hollywoodských filmů, kde byla zažita konvence pro kovboje hrdiny nesoucí bílé klobouky a kovboje padouchy nesoucí černé klobouky. V některých případech ovšem nelze hackera podle výše uvedených kategorií jednoznačně určit, každý sleduje své cíle. V odborné literatuře existují pro doplnění i další druhy hackerů (Hacker, 2025):

- **Blue Hat hacker** - podobné jako White Hat hacker avšak spíše konzultant,
- **Red Hat hacker** - zaměření na boj proti Black Hat hackerům,
- **Green Hat hacker** – naprostý nováček.

1.7.7 Skupiny útočníků

Postupem času vznikli i hackerské skupiny. Tým hackerů je přece jen mnohem častějším jevem než jeden samotný hacker. Osamocený hacker je v současné době velkou vzácností v oblasti velkých kybernetických útoků. Skupiny spojují různé dovednosti (jeden je expert na sítě, druhý na sociální inženýrství, třetí na šifrování a podobně), zájem o peníze, navzájem se chrání, mají společné cíle a jiné. Ve světě internetu je rovněž zajímavé, že ačkoliv jsou členové skupiny spolu v úzkém kontaktu a její členové jsou rozesety všude po světě, oslovují se mezi sebou

výhradně pouze přezdívkami a navzájem se vůbec neznají. Jednají tak i pro své vlastní dobro, kdyby někoho z nich vyslýchali tak skutečně neznají druhého.

Pro účely penetračního testování se také vytvořili skupiny, většinou se však jedná o obránce a útočníky. Každý člen týmu se opět specializuje na svojí jedinečnou dovednost. I v reálném světě je více než pravděpodobné, že útočit bude skupina hackerů, než jeden hacker a proti nim bude také stát skupina profesionálů. K tomu se i vytvořili podmínky v oboru penetračního testování a vytvořili se týmy (Microsoft Copilot: váš AI pomocník, 2025):

- **Red Team** – simulují útok, jedná se o skupinu penetračních testerů, kteří se chovají jako Black Hat hackeři. Jejich cílem je proniknout do systému zadavatele a vynést data, zašifrovat data, obcházet zabezpečení, objevovat otevřené porty a spousty jiných záškodnických činností. Jejich jediným úkolem je za pomoci všech dostupných prostředků co nejvíce poškodit cíl a prověřit obranu. Cílem je dokázat jak by se skupina profesionálních útočníků dostala do systému a prověřit stav ochrany organizace. Jedná se o realistickou simulaci skutečného protivníka. Za příklady typické pro útok Red Team jsou phishing, sociální inženýrství, útoky na wi-fi a jiné (Penetrační testování, 2025),
- **Blue Team** – simulují obranu, opět se jedná o profesionální bezpečnostní tým penetračních testerů s cílem zabránit útoku a ochránit tak maximálně napadenou organizaci. Nutno podotknout, že jejich obrana je nepřetržitá, není jasné kdy Red Team zaútočí. V tomto testu může útok přijít kdykoliv a Blue Team chrání organizaci každý den. Schválně jsou takto testy připraveny, aby co se nejvíce podobali realitě a útok byl nečekaný. Úkolem Blue Teamu je monitoring sítě, portů, aktivit v síti, detekovat útoky a reagovat na incidenty. Za pomoci všech nástrojů se snaží zastavit Red Team, vylepšovat zabezpečení, vyšetřovat incidenty a zpevňovat obranu zároveň i proti ostatním skutečným kybernetickým útokům. K dispozici má správu systému SIEM (Security Information and Event Management), analýzu logů ale může i školit zaměstnance (Microsoft Copilot: váš AI pomocník, 2025),
- **Purple team** – je opět kombinací Red a Blue Team. Cílem Purple Team je snaha učit se jeden od druhého, rychleji vylepšovat bezpečnost, vylepšovat spolupráci obou týmů, sdílet informace o vylepšit jak obranu tak útok. Další cíle Purple Team je zajištění komunikace mezi oběma týmy a provádění pravidelného školení mezi oběma týmy. Red Team i Blue Team jsou svými možnostmi limitováni, Purple tým je spojuje do jednoho celku a zajišťuje zlepšení obou týmů. Purple tým je jakýsi prostředník mezi oběma týmy a jedná se spíše o metodologii než skutečný tým. Což je virtuální tým složený ze členů Red a Blue Team (Microsoft Copilot: váš AI pomocník, 2025).

Další možné známé týmy v oblasti penetračního testování jsou (Microsoft Copilot: váš AI pomocník, 2025):

- **Green Team** – zaměřuje se na výuku a trénink ostatních týmů,
- **Yellow Team** – zaměřuje se vývoj a DevOps, také integruje bezpečný kód do vývoje.

Další známé hackerské skupiny, které jsou spojovány s kybernetickými útoky, objevují se v mediích avšak netýkají se etického penetračního testování jsou pro doplnění:

- **Script Kiddies** – jedná se spíše o motivované nováčky, kteří využívají skripty a nástroje vytvořené skutečnými hackery k provedení útoků. Ačkoliv nejsou svými schopnostmi

nijak výjimečný mohou způsobit značné škody a jsou často kritizováni za nedostatek respektu a etického chování (Microsoft Copilot: váš AI pomocník, 2025),

- **Hacktivisté** – hackerská skupina, která používá své dovednosti k prosazování zejména politických a sociálních cílů. Bojují za zodpovědnost vlád, korporací, za lidská práva a životní prostředí. I když mají ušlechtilé cíle a myšlenky, stále tak jednají v rozporu se zákonem. Specializují se na zveřejňovaní citlivých informací (Hacker, 2025).
- **APT skupiny (Advanced Persistent Threat)** – skuteční profesionálové většinou najatí vládami k prosazení vládních zájmů s nejmodernějším vybavením. Působí ve službách armád, zpravodajských služeb nebo vládních agentur. Používají mimořádně nebezpečné malware i exploity a jejich útoky jsou zdrcující. Jsou špičkově organizovaní a cílí na vládní sítě, energetiku, obranný průmysl, zdravotnictví nebo výzkum (Microsoft Copilot: váš AI pomocník, 2025).

K účinné obraně před využitím slabin hackerskými útoky existuje pár všeobecných tipů jako používat silná hesla, aktualizovat software, používat antivirový software, chránit wi-fi sítě dostatečně silným heslem a šifrováním, dávat si pozor na emaily a odkazy, používat dvoufaktorové ověření a tak dále (Hacker, 2025). Dále je možné se rovněž setkat s pojmem vektor útoku. Vektorem útoku se označuje způsob, jakým je útok veden. Jedná se o způsob útoku, jakým dochází ke zneužití zranitelnosti. V České republice se sledováním hackerských aktivit a ochranou proti nim zabývá NÚKIB (Národní úřad pro kybernetickou a informační bezpečnost).

1.7.8 Cyber Kill Chain

Cyber Kill Chain (CKC) popisuje fáze kybernetického útoku, kterými musí útočník během útoku projít. V podstatě slouží k tomu, aby obránci pochopili jak útočníci postupují a mohli je zastavit v jednotlivých fázích útoku. Model prakticky dokázal, že útok není jednorázová akce, ale řetězec kroků, které lze narušit. Model Cyber Kill Chain byl původně vyvinut kolem roku 2011 soukromou firmou Lockheed Martin jako interní bezpečnostní rámec. Lockheed Martin je známý dodavatel amerického ministerstva obrany. Pojem Kill Chain je pojem z vojenské terminologie a popisuje kroky vedoucí k neutralizaci cíle. Jednotlivé kroky Cyber Kill Chain jsou:

- **Reconnaissance (průzkum)** – útočník zkoumá a sbírá informace o cíli,
- **Weaponization (ozbrojení)** – útočník vytváří škodlivý nástroj, payload (exploit),
- **Delivery (doručení)** – přenesení nástroje, payloadu (exploitu) do cílového prostředí pomocí emailů, webových stránek nebo USB flash disku,
- **Exploitation (zneužití)** – po doručení payloadu na zařízení oběti je spuštěn exploit a následuje zneužití zranitelnosti systému či uživatele,
- **Installation (instalace)** – instalace škodlivého kódu nebo zadních vrátek s cílem zajištění trvalého přístupu k zařízení oběti,
- **Command and control (C2) (Velení a řízení)** – útočník naváže komunikaci s napadeným hostitelem přes některý z C2 serverů,
- **Actions on objectives (Akce na cíle):** – útočník data krade, šifruje nebo provádí špionáž.

Teprve po úspěšném průchodu prvních šesti fází může útočník podniknout kroky k dosažení svých cílů. Jednotlivé kroky na sebe navazují, aby mohl být proveden následující krok musí být

předchozí krok splněn. Stačí jediné úspěšné protiopatření u konkrétní fáze a celý řetězec se zhroutí. Model Cyber Kill Chain slouží jako vodítko při analýze útoku (Microsoft Copilot: váš AI pomocník, 2025).

OODA loop je model strategie, který byl také původně vytvořen pro vojenské účely. Jedná se o uzavřenou smyčku kroků Observe (pozorování), Orient (orientace), Decide (rozhodnutí) a Act (akce). Principem je, že smyčka musí být narušena obránci dříve, než se smyčku podaří dokončit útočníkovi (Microsoft Copilot: váš AI pomocník, 2025).

1.7.9 Bezpečnostní audit

Bezpečnostní audit je typický pro organizace podléhající regulacím, normám či certifikacím. Penetrační testování je nedílnou součástí bezpečnostního auditu. Bezpečnostní audit je mnohem širší proces vedoucí je zjištění reálného a hlavně komplexního stavu bezpečnosti podniku. Proto se zde v krátkosti vysvětlí pojem bezpečnostní audit pro úplnost.

Bezpečnostního audit je systematické a komplexní hodnocení všech bezpečnostních opatření a postupů organizace. Cílem bezpečnostního auditu je posoudit stav bezpečnosti organizace, identifikovat zranitelnosti, odhalit slabá místa i rizika a doporučit kroky vedoucí ke zvýšení ochrany. Může se zaměřit nejen na kybernetickou (IT infrastruktura, zabezpečení software, sítí, datových úložišť a podobně) a fyzickou bezpečnost (budovy, prostory, kamery, fyzickou ostrahu či ochranu proti neoprávněnému vstupu) ale také i na organizační zabezpečení (bezpečnostní prověrky, politiky, školení, směrnice a jiné). Zabývá se kontrolou všech operací organizace tak, aby byla zajištěna ochrana dat, majetku a osob. Výsledkem bezpečnostního auditu je zhodnocení aktuální bezpečnosti organizace, doporučení vedoucí k minimalizaci rizik, doporučení pro zvýšení bezpečnosti, školení a doporučení pro zvýšení odolnosti v případě krizových situací či bezpečnostních incidentů. Bezpečnostní audit je skutečně velice komplexní proces vedoucí k získání přehledu o bezpečnosti organizace. Mezi jedním z mnoha oborů bezpečnostního auditu patří kybernetická bezpečnost a penetrační testování představuje jeho praktickou část (Microsoft Copilot: váš AI pomocník, 2025).

I ve světe bezpečnostních auditů existují profesionální firmy zabývající se jeho vykonáním. Pravidelné provádění bezpečnostního auditu i penetračních testů je důležité zejména k současnému technologickému vývoji. Je tak současně důležitou prevencí chránící nejen data, zaměstnance a majetek společnosti.

2 Praktická část

Praktická část bakalářské práce se bude zabývat testováním zmiňovaných nástrojů Kali Linux v praktické rovině a na konkrétních příkladech. Budou zde popsány možnosti využití jednotlivých technik penetračního testování, průběhy testů budou popsány a výsledky zdokumentovány.

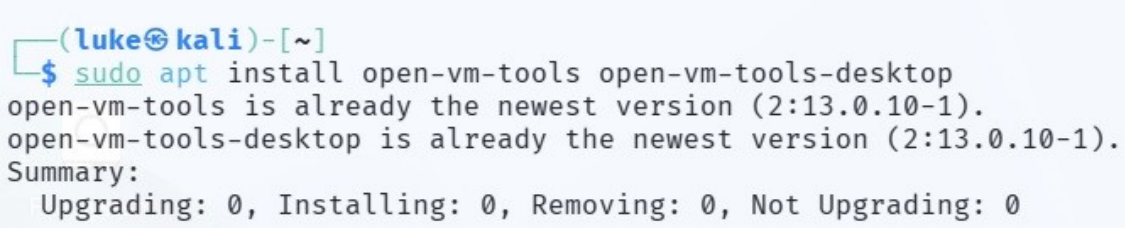
2.1 Příprava testovacího prostředí

Pro testování je možné použít tři různá konkurenční virtuální prostředí, VMware Workstation, Oracle VirtualBox a Hyper-V. Hyper-V je dostupné ve všech verzích operačních systémů Windows 8 a novějších, výjimku tvoří pouze Home edice Windows. Sice má nejlepší výkon a je stabilnější než konkurenční virtuální prostředí, je však vhodnější spíše pro produkty od firmy Microsoft a není vhodný pro linuxové desktopy. Aktivuje se v ovládacích panelech, programy a zde zapnout nebo vypnout funkce systému Windows. VirtualBox od firmy Oracle má sice slabší výkon, cílí však na uživatele svojí jednoduchostí a lze jej využít univerzálně jak pro OS Windows, tak i pro linuxovou virtualizaci. VirtualBox je zdarma a open-source (Microsoft Copilot: váš AI pomocník, 2025).

2.1.1 VMware Workstation 17 Pro 17.6.4

Pro testování bylo použito virtuální prostředí VMware Workstation 17 Pro 17.6.4. Důvodů pro výběr tohoto virtuálního prostředí je hned několik. Mimo jeho naprosté jednoduchosti a kompatibilitou s linuxovými systémy a také zejména možnost snímat obrazovku čili vytvářet screenshoty efektivněji než v konkurenčním virtuálním prostředí VirtualBox od firmy Oracle. Rovněž i instalace doplňků, nástrojů a ovladačů, které je nutné ve VirtualBox instalovat pomocí ISO souboru VBoxGuestAdditions.iso, pro lepší běh virtuálních systémů ve virtuálním prostředí a kvalitní pořizování screenshotů, lze ve VMware stáhnout pohodlně pomocí příkazu v terminálu (ve VirtualBoxu lze samozřejmě také stáhnout doplňky pomocí příkazu v terminálu, ale výsledek není plnohodnotný, neobsahuje nejnovější ovladače, chybí funkce drag and drop a jiné). Moderní linuxové distribuce jako jsou Kali Linux, Fedora, RHEL či Debian používají moderní jádro Linuxu (Kernel), které je se staršími VMware Tools nekompatibilní. VMware k moderním linuxovým systémům již nepřibaluje klasické VMware Tools, proto vyskakuje hlášení, že obraz VMware tools neexistuje a nabízí ke stažení ISO soubor linux.iso. Z důvodu moderních linuxových distribucí VMware přešel na princip open-vm-tools, balíčky open-vm-tools jsou uloženy přímo v repozitářích a nestahuje ani neinstaluje se tak ISO soubor linux.iso, jako tomu bylo dříve. Další obrovský důvod pro výběr VMware Workstation je jeho Pro verze, která byla po mnoho let placená a od roku 2024 je zdarma pro osobní i komerční použití. Zdarma byla dříve po celou dobu pouze verze VMware Workstation Player pro nekomerční použití. Jedná se poměrně o čerstvou novinku, která souvisí s převzetím VMware firmou Broadcom, okamžitě ukončila prodej licencí pro Pro verzi a placená zůstala pouze podpora. Pro firmu Broadcom již nejsou desktopové virtualizační nástroje typu Workstation a Fusion strategicky důležité a negenerují tak velký zisk jako velké firemní produkty VMware. Firma zrušila staré licenční modely a nyní se orientuje na cloudové a datacentrové platformy

(Microsoft Copilot: váš AI pomocník, 2025). Oproti tomu open-source projekt VirtualBox byl vždy zdarma. Poslední verzi VMware Workstation 17 Pro 17.6.4, která byla použita v praktické části bakalářské práce, je možné stáhnout z mnoha zdrojů (Index vmwareworkstationarchive/17.x/, 2026; VMware Workstation Pro, 2026).



Obr. 8: Příkaz open-vm-tools

Zdroj: Vlastní zpracování (2026)

2.1.2 Kali Linux 2025.4 installer amd64

Pro testovací účely byla stažena poslední čtvrtletní rolling-release verze linuxové distribuce Kali Linux z roku 2025. Jedná se o plný instalační ISO obraz o velikosti 4,40 GB, vhodný také k offline instalaci, se všemi dostupnými nástroji a aktualizacemi. Stažený ISO obraz se poté nainstaluje ve virtuálním prostředí. Ke stažení je k dispozici na stránkách projektu Kali Linux. Nutno podotknout, že mimo instalátor, se kterým bude bakalářská práce pracovat, je možné stáhnout i ISO obrazy jiné architektury, netinstalátory, live obrazy nebo přímo předpřipravené obrazy do virtuálních prostředí a další. Dostupné možnosti stažení distribuce Kali Linux jsou uvedeny na webových stránkách projektu Kali Linux (Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution, 2025).

2.1.3 Instalace VMware Workstation 17 Pro 17.6.4

Instalace VMware Workstation 17 Pro 17.6.4 je velice intuitivní a není potřeba jej rozepisovat. Samozřejmě je lepší jej instalovat se všemi nástroji a doplňky, které instalace nabízí. Po dokončení instalace se objeví v pravém dolním rohu nová tray ikona.

2.1.4 Instalace Kali Linux 2025.4 installer amd64

Postup instalace ISO obrazu ve virtuálním prostředí VMware je pro zkušeného člověka opět velice intuitivní. Po spuštění VMware Workstation 17 Pro se klikne na Create a New Virtual Machine a zvolí se vlastní pokročilá instalace (lepší možnost, umožňuje vlastní nastavení) čili Type of Configuration Custom (advanced). Dále pak po jednotlivých krocích:

- Hardware compatibility Workstation 17.5 or later,
- Installer disc image file (iso): kali-linux-2025.4-installer-amd64,
- Select a guest operating system: Linux Debian 12.x 64-bit,
- Virtual machine name: KaliLinux-2025.4, Location: C:\Virtual\KaliLinux-2025.4-x86_64-iso,
- Processors, Number of processors: 2, Number of cores per processors: 2, Total processor cores: 4,
- Memory of virtual machine: 8192MB,

- Network Type: Use network address translation (NAT) (host's IP address),
- Select I/O Controller Types: SCSI Controller: LSI Logic (recommended),
- Select a Disk Type: Virtual disk type: SCSI (recommended),
- Select a Disk: Create a new virtual disk,
- Specify Disk Capacity: Maximum disk size (GB): 35GB, Not Allocate all disk space now, Split virtual disk into multiple files,
- Specify Disk File: Disk File: KaliLinux-2025.4.vmdk,
- Ready to Create Virtual Machine: Check Hardware Configuration.

Takto byla zvolena konfigurace pro testovací účely bakalářské práce, vzhledem k dostupným zdrojům bylo zvoleno dostatečné množství celkového počtu jader procesoru, raději více paměti a optimální velikost disku kvůli aktualizacím. Ostatní kroky instalace jsou typické pro virtuální prostředí.

2.1.5 Boot menu Kali Linux

Po dokončení instalace ISO obrazu Kali Linux v prostředí VMware Workstation je nutné instalovat samotný Kali Linux, v již existujícím virtuálním obrazu. Po spuštění VMware Workstation se objeví na levé straně nová záložka KaliLinux-2025.4. V záložce je možnost Power on this virtual machine, která umožní instalovat zmíněný systém. Možnost Edit virtual machine settings umožňuje upravit nastavení virtuálního systému. Po kliknutí na možnost Power on this virtual machine se spustí boot menu instalátoru neboli základní nabídka možností instalace distribuce Kali Linux. Možnosti instalace jsou následující:

- **Graphical install** – klasický grafický průvodce instalace s myší,
- **Install** - průvodce instalace v textovém prostředí bez použití myši, pouze klávesnicí,
- **Advanced options** - pokročilé možnosti pro spuštění systému či instalaci (expert install, rescue mode, automated install, opět možné v textovém či v grafickém prostředí, či s hlasovým průvodcem (speech),
- **Accessible dark contrast installer menu** - instalace v černém pozadí,
- **Install with speech synthesis** - instalace s hlasovým průvodcem.

Nejvíce vhodná instalace je hned první možnost Graphical install. Grafická instalace Kali Linux je naprosto shodná s grafickou instalací linuxové distribuce Debian. Kali Linux je odvozený derivát Debianu.

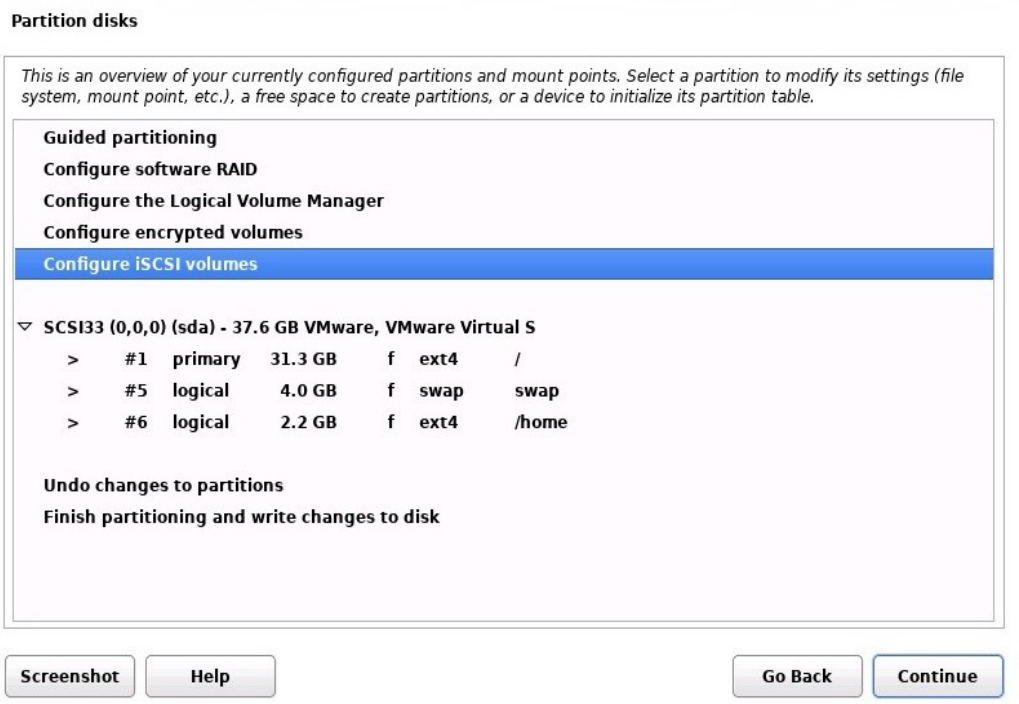
2.1.6 Grafická instalace Kali Linux

Zde je popsána grafická instalace Kali Linux, nahraná z boot menu, v pár krocích:

- Select a language: English,
- Select your location: other; Europe; Czechia,
- Configure locales: United States,
- Configure the keyboard: American English,
- Configure the network: Hostname: kali; Domain name: localdomain,
- Set up users and passwords: Full name for the new user: Luke; Username for your account: luke; Choose a password for the new user: -; Re-enter password to verify: -,
- Partition disks: Partitioning method: Guided - use entire disk,

- Partition disks: Select disk to partition: SCSI4,
- Partition disks: Partition scheme: Separate /home partition Partition,
- Partition disks: RAID: No; LVM: No; Configure encrypted volumes: No; Configure iSCSI Volumes: Nothing; Select Finish partitioning and write changes to disk,
- Partition disks: Write the changes to disks: Yes,
- Software selection: Choose software to install: Select Desktop environment; Xfce; Collection of tools; top10; default,
- Install the GRUB boot Loader: Install the GRUB boot loader to your primary drive: Yes,
- Install the GRUB boot Loader: Device for boot loader installation: /dev/sda,
- Finish the installation,
- Rebooting system and launching Kali.

Jeden z obtížnějších kroků pro začátečníka se může jevit část instalace týkající se rozdělení disků (partition disks). Pokud si začátečník není jistý, může zvolit možnost nerozdělovat disk. Vše tak bude na jednom disku s výjimkou swap oddílu, který se však rozdělí automaticky. Automaticky u novějších linuxových systémů, u starších stále platí, že jej musíme rozdělit manuálně. Swap oddíl je samostatná část disku nahrazující paměť, když už nestačí klasická RAM paměť. Využívá jej pouze systém, neukládají se do něj soubory, umožňuje hibernaci (uložení obsahu RAM na disk) a zajišťuje stabilnější chod systému při velké zátěži. Ve Windows existuje podobný, ale jiný mechanismus pagefile, soubor, který se dynamicky zvětšuje či zmenšuje a není na samostatném oddílu disku. Zde byl disk rozdělen na tři části, kromě kořenového (systémového) adresáře a automaticky vytvořeného swap oddílu, byl samostatně rozdělen i domovský adresář home. Adresář home proto, že zde slouží jako úložiště pro soubory uživatele, které se zde budou ukládat. Při poškození kořenového adresáře zůstane tak adresář home nepoškozený právě z důvodu umístění na jiném disku. Data na home tak zůstanou nepoškozená. I jiné adresáře obvykle jsou rozděleny na vlastním oddílu disku, adresáře jako /boot, /var, /srv, /usr a jiné. Děje se tak z důvodu stability, bezpečí a správy systému (Microsoft Copilot: váš AI pomocník, 2025).

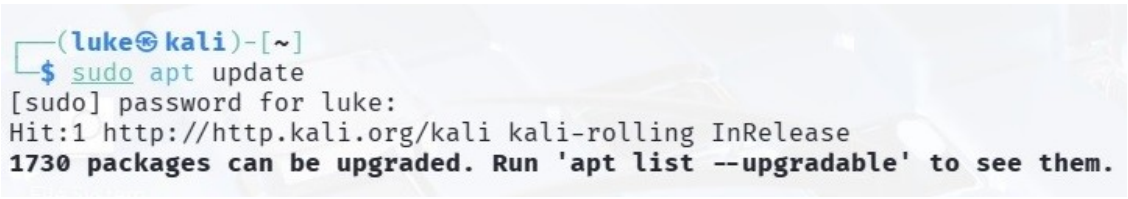


Obr. 9: Ukázka rozdělení disků při instalaci

Zdroj: Vlastní zpracování (2026)

2.1.7 Aktualizace instalace Kali Linux

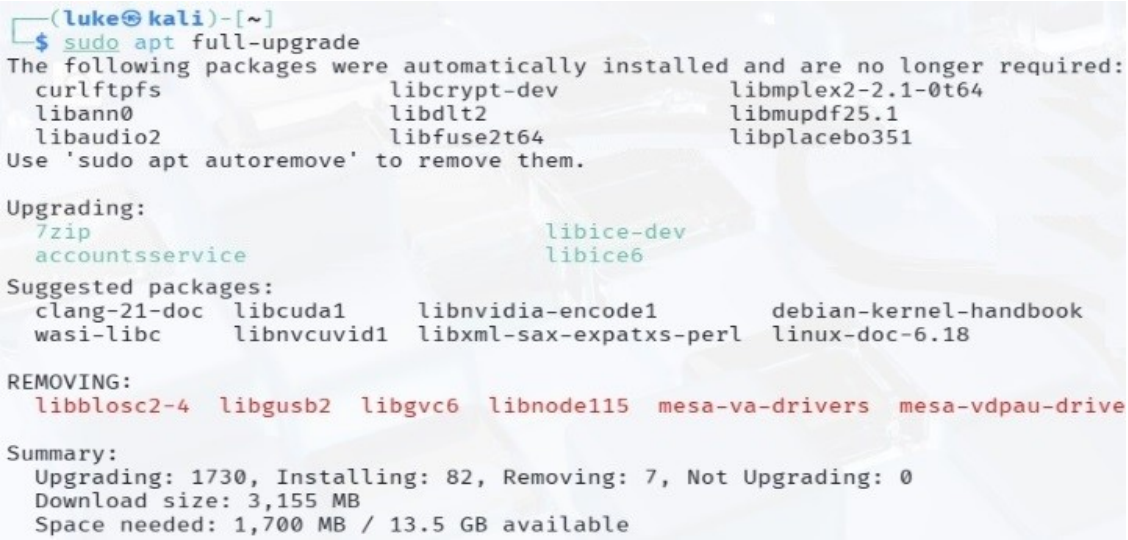
Z důvodu aktuálnosti systému a aktualizací seznamu dostupných balíčků je nutné provádět pravidelné aktualizace. Systém pozná, které programy mají nové verze a bezpečnostní opravy. Po aktualizacích bude systém bezpečnější a verze programů aktuální. Pro demonstrativní účely byla stažena a instalována verze Kali Linux 2025.4 a nyní bude povýšena na verzi Kali Linux 2026.1. Příkazem apt update se získá přehled o obnově seznamu dostupných balíčků, systém tak pozná, které balíčky mají nové verze:



Obr. 10: Příkaz apt update

Zdroj: Vlastní zpracování (2026)

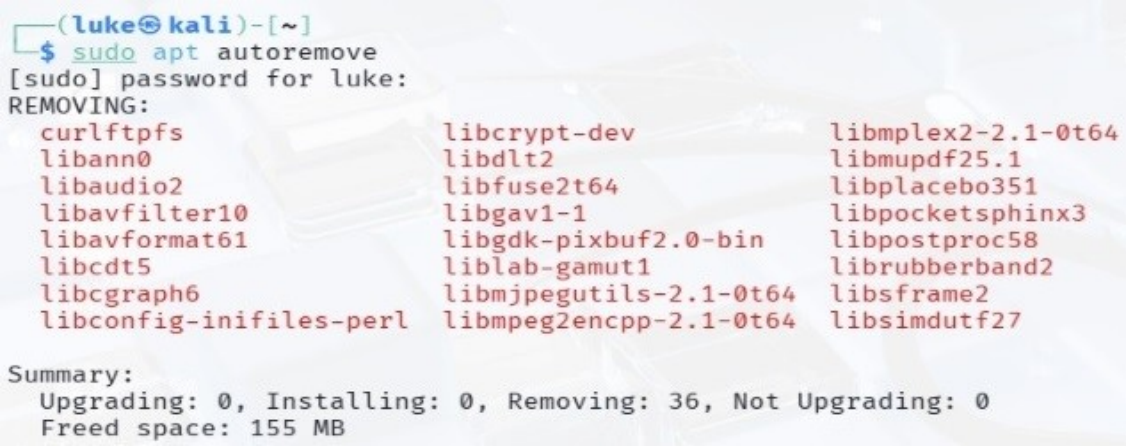
Z hlášení vyplývá, že příkaz apt update nalezl 1730 nových balíčků, které je vhodné upgradovat příkazem apt full-upgrade. Příkaz nainstaluje všechny dostupné aktualizace a zároveň může měnit i závislosti, může tak odstraňovat nebo nahrazovat balíčky. Alternativou je příkaz apt upgrade, který však aktualizuje pouze to, co lze bez odstranění balíčků. Apt full-upgrade aktualizuje všechno, i když se musí něco odstranit:



Obr. 11: Příkaz apt full-upgrade

Zdroj: Vlastní zpracování (2026)

Po nainstalování všech balíčků je systém aktuální. Příkazem apt autoremove je možné smazat staré, již nepotřebné a neaktuální balíčky ze systému:



Obr. 12: Příkaz apt autoremove

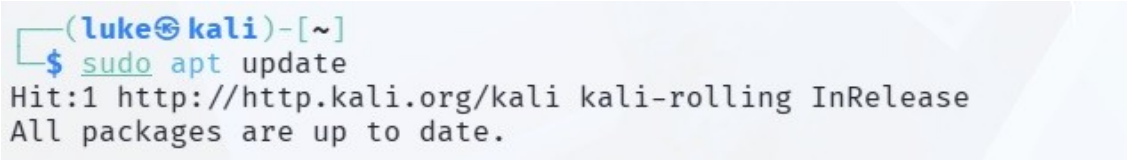
Zdroj: Vlastní zpracování (2026)

Potvrzením se vymažou staré nepotřebné balíčky a uvolní se tak místo na disku. Dalším příkazem apt clean je možné vymazat stažené instalační balíčky z cache dočasné paměti systému pro uvolnění místa na disku:

Obr. 13: Příkaz apt clean

Zdroj: Vlastní zpracování (2026)

Příkazem apt clean se vymažou stažené instalační balíčky z dočasné paměti systému. Přepínač -y za příkazy apt full-upgrade -y a apt autoremove -y automaticky odpoví ano na všechny dotazy během instalace nebo aktualizace. Systém je nutné ještě restartovat příkazem reboot pro jeho plné obnovení po aktualizacích. Po restartu je možné ověřit příkazy apt full-upgrade, apt update či cat /etc/os-release, zda je systém skutečně aktuální:



Obr. 14: Příkaz apt update (up to date)

Zdroj: Vlastní zpracování (2026)

Obr. 15: Příkaz cat /etc/os-release

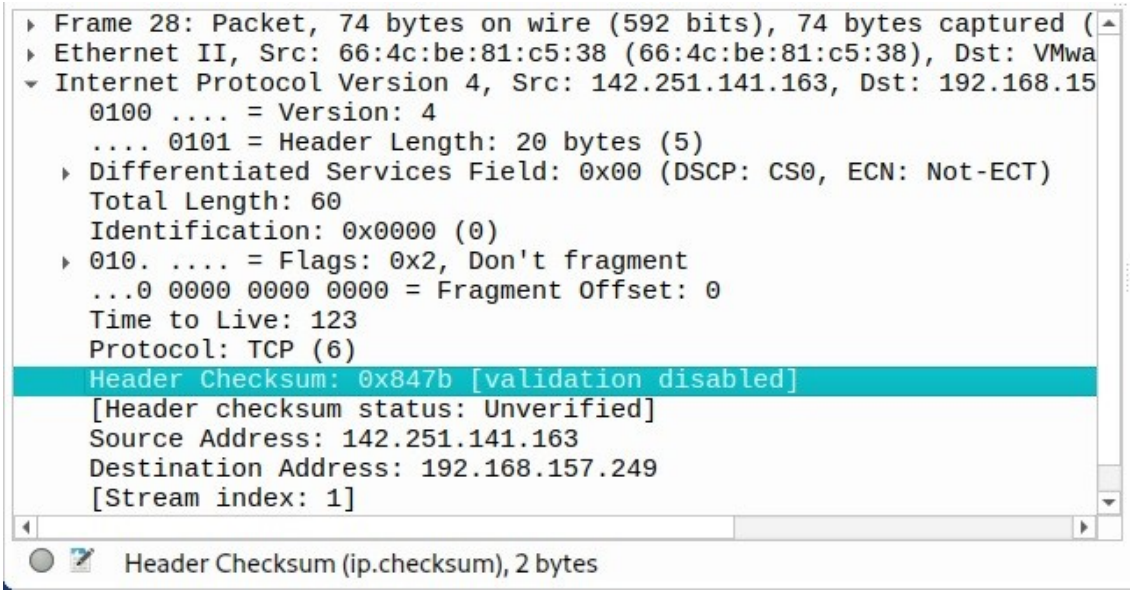
Zdroj: Vlastní zpracování (2026)

Příkazem cat /etc/os-release je možné spatřit řádek VERSION_ID="2026.1" či řádek VERSION="2026.1" indikující novou verzi systému 2026.1 (Microsoft Copilot: váš AI pomocník, 2025).

2.2 Wireshark – monitoring sítě

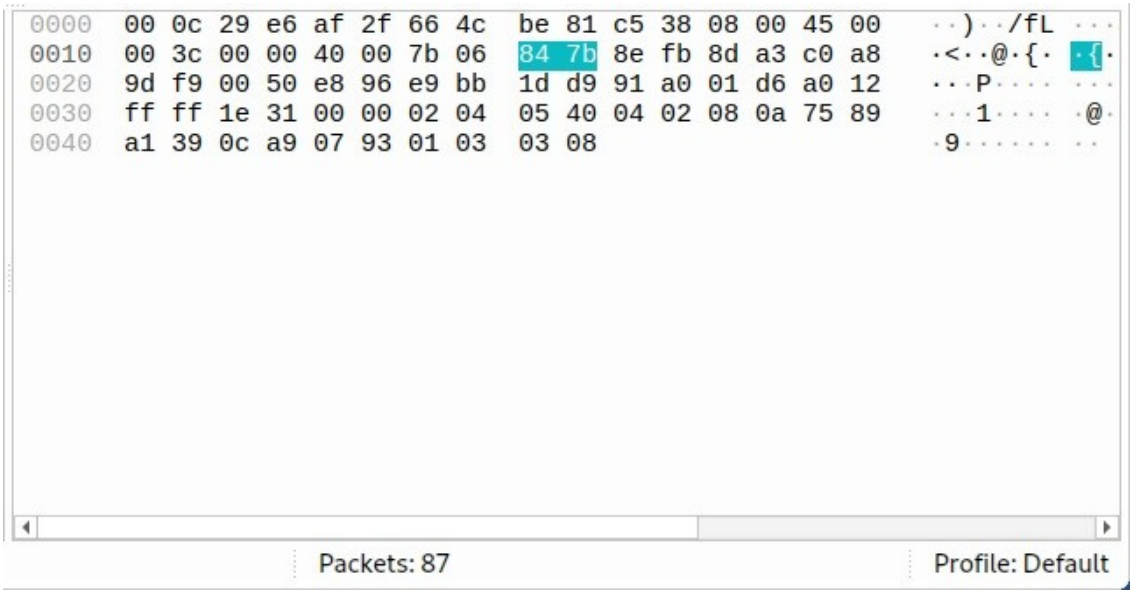
Wireshark je analyzátor síťového provozu a síťových protokolů. Využívá se zejména pro diagnostiku problémů v síti, například ztráta paketů, analýza handshake paketů, zda DNS odpovídá správně či jestli firewall něco neblokuje. Je možné jej rovněž využít pro odhalování podezřelých aktivit, například špatně odesílající data, komunikace s podezřelými IP, jestli probíhá nečekaný broadcast, multicast či ARP spoofing. Analyzuje a kontroluje stovky různých protokolů, například HTTP(S), TLS, TCP, ARP, DHCP, DNS, QUIC, HTTP/3, VoIP (SIP, RTP), SMB, SSH nebo MQTT. Dokáže ukládat zachycené pakety (soubory s příponou PCAP/CAP) pro pozdější analýzu. Někdy je také nazývaný jako analyzátor paketů či paketový sniffer. Zachycuje pakety v reálném čase a zobrazuje dokonce i jejich obsah pokud není šifrovaný. Dokáže zobrazit celou strukturu vybraného paketu od fyzické vrstvy až po aplikační vrstvu podle modelu OSI/ISO. Model OSI/ISO má vrstvy aplikační (HTTP, HTTPS, FTP, DNS), prezentační (TLS, SSL), relační (TLS handshake, session management), transportní (TCP, UDP), síťovou (IP, ICMP), linkovou (Ethernet, Wi-Fi) a fyzickou (kabely, rádiové vlny). Vrstvy dokáže srovnat přehledně od nejnižší po nejvyšší (packet details) a hned vedle i zobrazit přehled bajtů jednotlivých vrstev reprezentované jako raw bajty v hexadecimální a ASCII podobě (packet bytes). Seznam paketů v hlavním okně ukazuje síťový provoz na daném síťovém rozhraní (packet list). V domácí Wi-Fi síti se bohužel router chová jako switch a Wireshark nezachytí provoz všech zařízení v síti. Router posílá data jen tomu zařízení, komu jsou určena. Jediná možnost jak vidět všechna zařízení v síti je propojit je hubem, hub posílá data všem zařízením v síti. Wireshark dokáže

poskytnout mnoho informací o paketech. Podle protokolu lze určit na jaké vrstvě modelu OSI/ISO služba probíhá, jednotlivé vrstvy paketu lze otevřít pro více informací. Rovněž Wireshark nabízí množství filtrů pro vybrání správných paketů. Jednotlivé pakety jsou barevně odděleny, barvy lze však vypnout. Program má grafické rozhraní (Wireshark Basics, 2023).



Obr. 16: Wireshark, detail paketu

Zdroj: Vlastní zpracování (2026)



Obr. 17: Wireshark, přehled bajtů

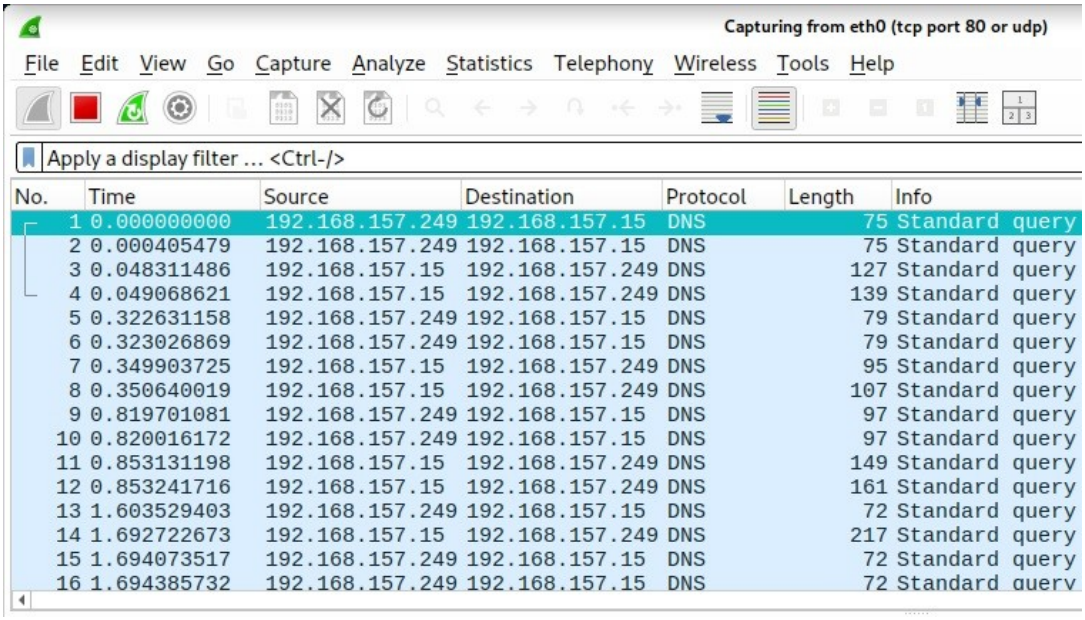
Zdroj: Vlastní zpracování (2026)

2.2.1 Filtrování paketů

Provoz sítě je možné filtrovat podle protokolu, IP adresy, portu a jiných parametrů. Filtrování umožňuje zúžit výsledky vyhledávání pouze na vybrané pakety. Wireshark má vysoce propracovaný systém filtrování, který umožňuje i napovídání filtru, podobně jako v programátorských aplikací. Filtrování lze rozdělit do dvou hlavních filtrů, zachytávací filtr (capture filter) při spuštění, kdy Wireshark dává možnost vybrat si síťové rozhraní,

a zobrazovací filtr paketů (display filter). Rozdíl mezi nimi je naprosto jednoduchý a logický. Zachytávací filtr v úvodním okně, při výběru síťového rozhraní, umožní aktivovat filtr již při spuštění vyhledávání paketů. Program bude vyhledávat pouze ty pakety, které se týkají zachytávacího filtru, ostatní pakety nezobrazí. Samozřejmě je později možné doplnit zachytávací filtr i zobrazovacím filtrem. Oproti tomu zobrazovací filtr najde všechny dostupné pakety. Z nich vybere pouze ty, pro které bude nějaký zobrazovací filtr později aktivován. Při psaní filtrů program našeptává pokračování nabídky filtru a je možné, podobně jako v programování, je spojit porovnávacími operátory, rovná se (==), nerovná se (!=), větší než (>), menší než (<), větší nebo rovno (>=) a podobně. Stejně jako s porovnávacími operátory je možné psát filtry i pomocí logických operátorů, AND (&&), OR (||), XOR (^^), NOT (!) a dalších. Výběr pro psaní filtrů je skutečně rozsáhlý, filtry nám dokážou vybrat pár paketů z tisíce. Zachytávací filtry jsou obecnějšího rázu a lze mezi ně řadit například filtry (Wireshark User's Guide, 2026):

- tcp port 80 or udp,
- host 127.0.0.1,
- port 443,
- tcp port 8088,
- udp range 1-100.

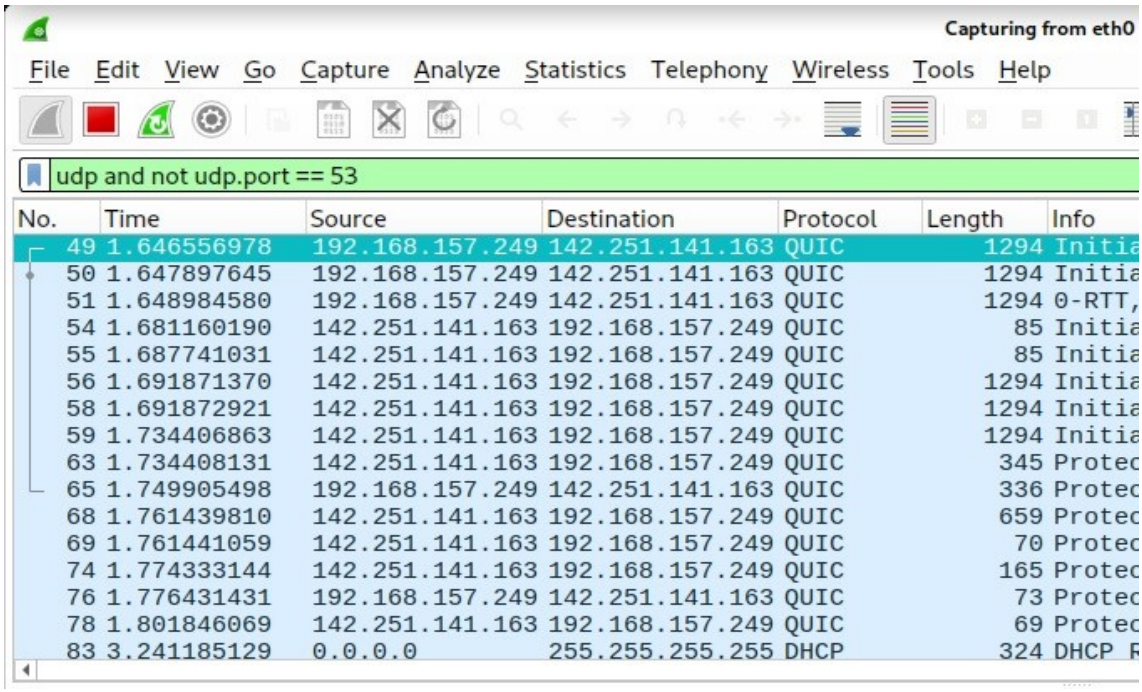


Obr. 18: Wireshark, zachytávací filtr (HTTP a UDP)

Zdroj: Vlastní zpracování (2026)

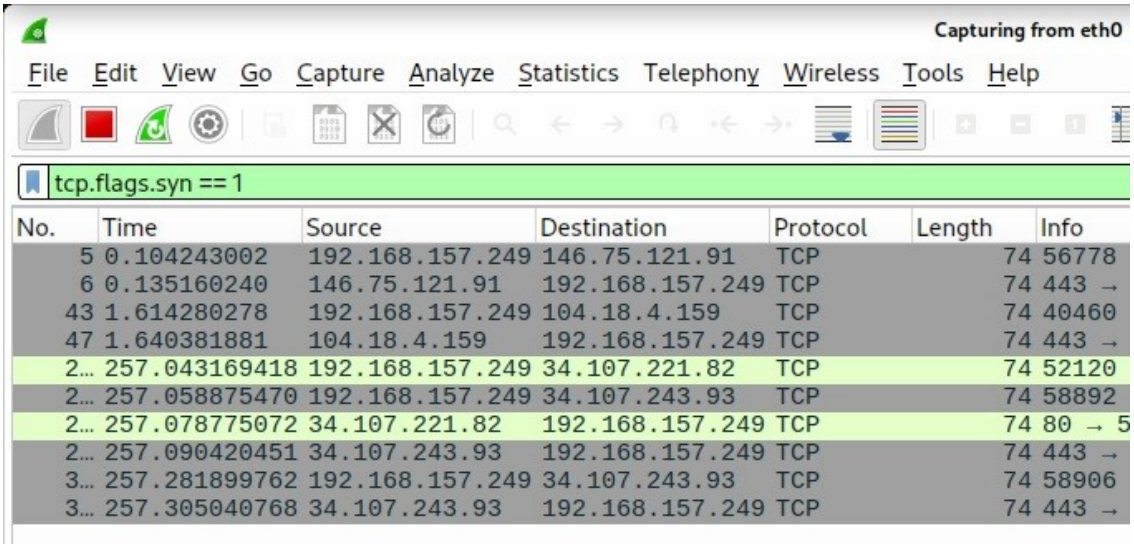
Zobrazovací filtry jsou oproti zachytávacím filtrům konkrétnější, cílí na menší počet filtrovaných paketů a lze mezi ně řadit například filtry:

- udp and not udp.port == 53,
- tls.handshake.type == 1 or tls.handshake.type == 2,
- frame contains "login" or frame contains "password",
- http and frame.len > 500,
- tcp.flags.syn == 1.



Obr. 19: Wireshark, zobrazovací filtr (UDP kromě DNS)

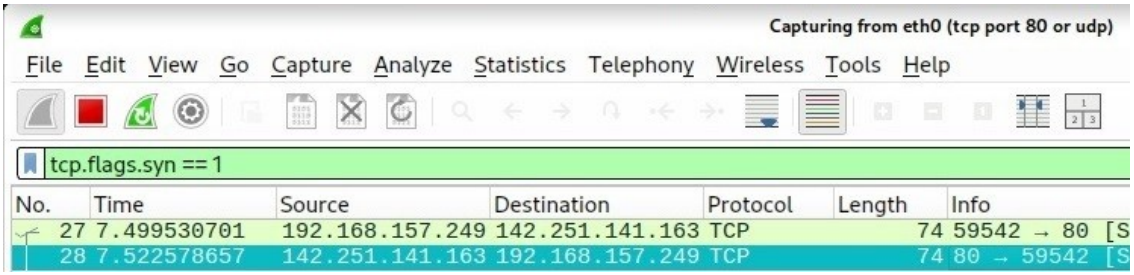
Zdroj: Vlastní zpracování (2026)



Obr. 20: Wireshark, ukázka zobrazovacího filtru (TCP pakety se SYN)

Zdroj: Vlastní zpracování (2026)

Zobrazovací filtry lze samozřejmě i kombinovat se zachytávacími filtry, například:



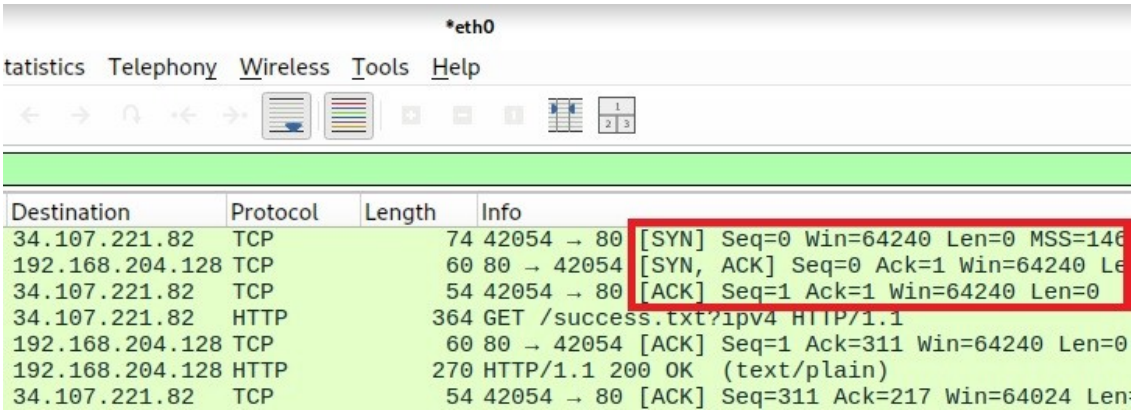
Obr. 21: Wireshark, ukázka kombinace obou filtrů (HTTP a UDP + TCP (SYN a SYN/ACK)

Zdroj: Vlastní zpracování (2026)

2.2.2 TCP, TLS protokoly

TCP je protokol vytvářející spolehlivé spojení mezi dvěma zařízeními. Pokud se obě strany dohodnou, že spolu budou komunikovat, proběhne tzv. třícestný handshake. TCP handshake probíhá ve třech krocích (SYN, SYN/ACK, ACK). TCP protokol zajišťuje, aby data dorazila správně, bez chyb a v přesném pořadí. Je pomalejší, ale spolehlivý, využívá se k posílání e-mailů či souborů a podobně. Podobný protokol ze stejné transportní vrstvy modelu OSI/ISO, protokol UDP, již není tak spolehlivý, neposílá pakety v přesném pořadí, ani nepošle znovu ztracené pakety, ale je velmi rychlý a využívá se při přehrávání filmů, online her nebo videohovorů. Proto UDP handshake nevyžaduje, kdežto TCP ano (Začněte skutečně ovládat PC. Odposlouchávejte domácí síť, 2014):

- **SYN (Synchronize)** - SYN je zpráva, kterou klient navazuje spojení se serverem a podává serveru své počáteční sekvenční číslo. Sekvenční číslo je důležité pro řízení toku dat. TCP potřebuje vědět, v jakém pořadí data přicházejí, aby obě strany, klient i server věděly, kde přesně komunikace začíná, v jakém pořadí mají být data složena a jestli se nějaký paket neztratil nebo neopakoval. TCP je totiž spolehlivý protokol. Musí zajistit, že data dorazí správně a ve správném pořadí. Při ztrátě paketu TCP protokol pošle nový paket,
- **SYN/ACK (Synchronize / Acknowledge)** - server (SYN) odpoví kombinovanou zprávou, že chce komunikovat, a pošle klientovi také své sekvenční číslo a klient (ACK) odpoví, že zprávu dostal. TCP je obousměrný protokol. Což znamená, že klient posílá data serveru, server posílá data klientovi. Každá strana posílá svůj vlastní proud dat a každá strana potřebuje své vlastní sekvenční číslo. Jedná se tedy o potvrzení a vlastní požadavek na synchronizaci,
- **ACK (Acknowledge)** - klient pošle poslední ACK zprávu, potvrzení, že dostal od serveru SYN/ACK zprávu. Handshake dokončen a spojení je navázáno. Obě strany jsou připraveny komunikovat, sekvenční čísla jsou synchronizovaná a žádná strana neposílá data do prázdna.



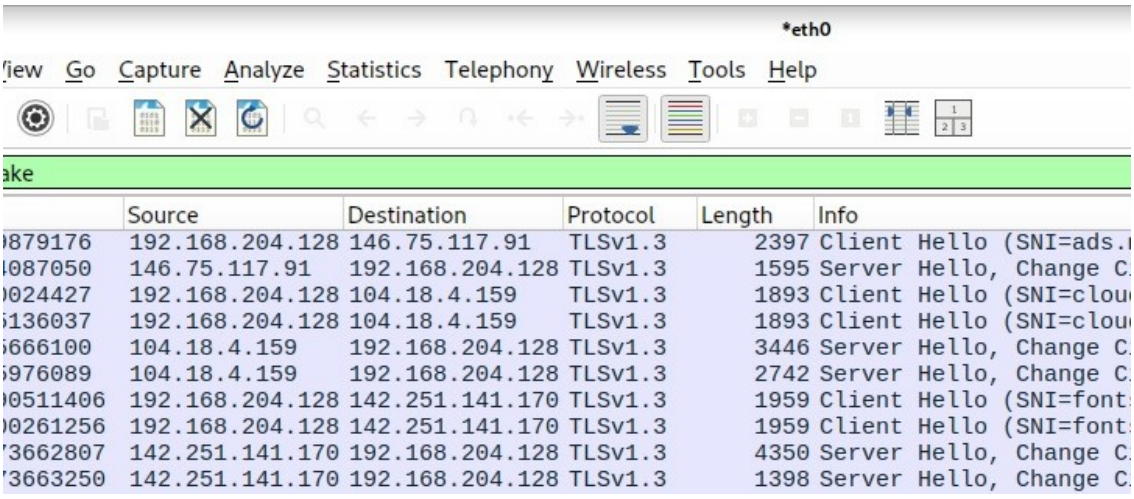
Obr. 22: Wireshark, navázání třícestného handshake

Zdroj: Vlastní zpracování (2026)

TLS je kryptografický protokol zajišťující bezpečnou komunikaci mezi dvěma stranami, mezi webovým prohlížečem a serverem. Poskytuje veškeré bezpečnostní záruky, důvěrnost, integritu a autentizaci v podobě šifrované komunikace a digitálních certifikátů. TLS je nástupce staršího SSL a v dnešní době se používá prakticky všude, kde je zapotřebí bezpečná

komunikace, například u protokolu HTTPS. Bez použití TLS protokolu lze komunikaci odposlouchávat, může být upravena či přesměrována na jiný server. Proto je tak nebezpečné využívat HTTP protokol místo HTTPS. Mimo HTTPS je TLS využíván při e-mailové komunikaci, VPN, VoIP nebo při komunikaci mezi servery. Varianta DTLS je bezpečnostní protokol pro datagramy UDP. Moderní TLS 1.3 je oproti TLS 1.2 rychlejší, bezpečnější a jednodušší. Zkracuje dobu trvání handshake, používá moderní šifry a vyžaduje i forward secrecy, což je náhrada za staré RSA výměny klíčů a slabé šifrovací módy. Zrychluje dobu navázání spojení a snižuje možnost chybné konfigurace. Handshakem si klient i server ověří identitu, domluví šifrovací algoritmy a vytvoří společný šifrovací klíč pro vzájemnou bezpečnou komunikaci. Během handshake server pošle svůj certifikát k ověření a poté se obě strany dohodnou na klíčích pro šifrování dat (Wireshark Basics, 2023):

- **ClientHello** – klient zahájí komunikaci a pošle serveru podporované verze TLS, seznam šifrovacích dat a náhodná data pro generování klíčů. Z těchto parametrů si server může vybrat,
- **ServerHello** – server odpoví na požadavky a pošle vlastní náhodná data pro generování klíčů, čímž začne ověřování,
- **Certifikát serveru** – server pošle svůj certifikát k zajištění autentizace serveru, klient ověří jeho platnost, důvěryhodnost a zda odpovídá doméně,
- **Výměna klíčů** – klient i server si vymění veřejné klíče a vypočítají si vlastní, společný tajný klíč (ECDHE), ze kterého se odvodí symetrické klíče pro šifrování, takzvaný session key, se kterým se bude komunikace šifrovat. ECDHE je dnes standardem, protože poskytuje forward secrecy. RSA klíče jsou starší variantou,
- **Finished** – obě strany si pošlou šifrovanou zprávu „Finished“. Od teď běží mezi oběma stranami zabezpečené spojení.



Obr. 23: Wireshark, navázání zabezpečeného handshake

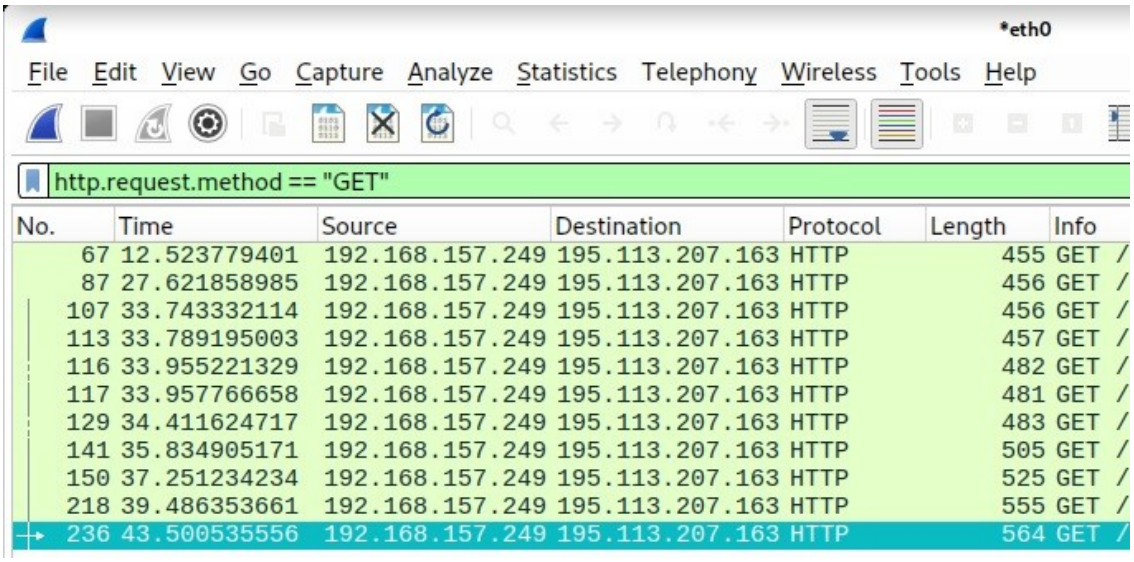
Zdroj: Vlastní zpracování (2026)

2.2.3 HTTPS, HTTP protokoly

Wireshark umožňuje zachytávat síťový šifrovaný (HTTPS) i nešifrovaný (HTTP) provoz sítě. V dnešní době je však nešifrovaný provoz spíše výjimkou, lze však díky němu zachytit v programu Wireshark užitečné informace. Následující informace byly získány díky

nešifrovanému provozu. Nešifrované informace jsou čitelné, šifrované čitelné nejsou (Wireshark Basics, 2023).

Postup k získání nešifrovaných informací je skutečně snadný. Přes prohlížeč se lze připojit k nějakému serveru přes HTTP protokol, a v nabídce se vyplní login a heslo. Po jejich vyplnění program Wireshark získá pakety HTTP komunikace se serverem, ve kterých budou zaznamenány přihlašovací údaje. Lze použít zobrazovací filtr http.request.method == "GET", který nám vypíše protokoly HTTP získané metodou GET:



Obr. 24: Wireshark, získaný HTTP paket

Zdroj: Vlastní zpracování (2026)

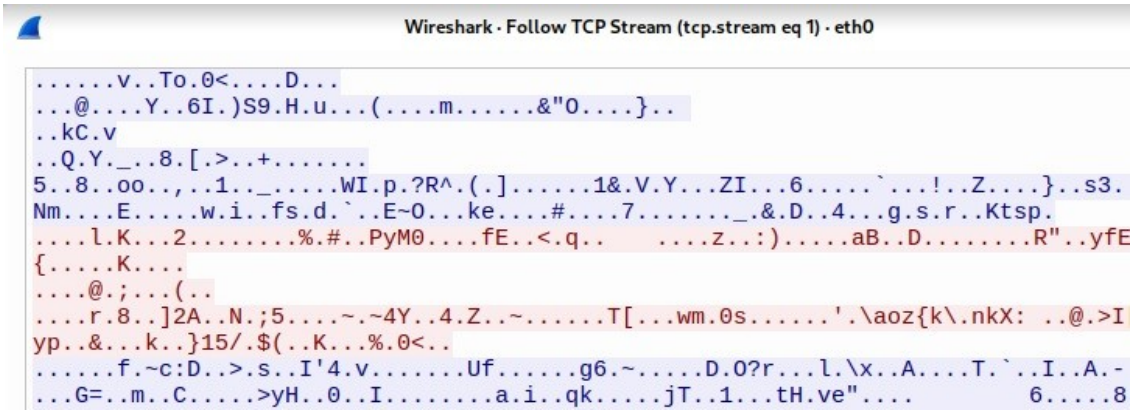
Na paket lze kliknout pravým tlačítkem myši a zde vybrat z nabídky Follow a poté HTTP Stream nebo TCP Stream. Zobrazí se textový výpis streamu, ze kterého lze vyčíst požadované informace, uživatelské jméno a heslo:

Obr. 25: Wireshark, získané informace HTTP komunikace

Zdroj: Vlastní zpracování (2026)

HTTP stream nebo TCP stream je rekonstruovaný datový proud, data jdoucí po sobě v pořadí, jak byla přenesena v rámci jednoho TCP spojení, čili se jedná o sloučený obsah paketů. Wireshark vezme všechny pakety patřící k jednomu spojení, seřadí je, odstraní fragmentaci a zobrazí souvislý textový tok. Tyto textové toky jsou barevně odděleny, červeně jsou data od klienta, modře data od serveru. TCP stream je celý proud dat spojení, HTTP stream je jen HTTP

komunikace uvnitř. V případě šifrované komunikace, HTTPS, jsou přihlašovací údaje v paketu chráněny protokolem TLS a data jsou nečitelná, šifrovaná (Wireshark Basics, 2023):

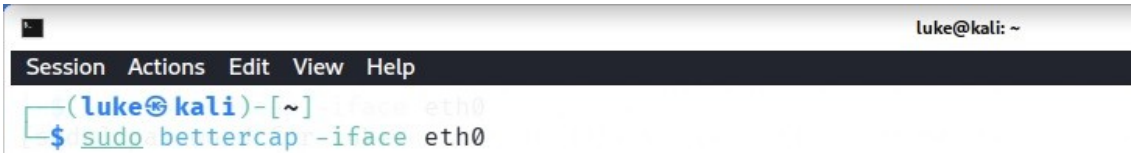


Obr. 26: Wireshark, nečitelná data HTTPS komunikace

Zdroj: Vlastní zpracování (2026)

2.3 Bettercap – MITM útoky

Bettercap je již útočný framework, nástroj pro vedení nejrůznějších MITM (man-in-the-middle) útoků na sítě čili odposlouchávání komunikace mezi komunikujícími zařízeními. Bettercap je mladším sourozeneckým programem nástroje Ettercap, jehož vývoj byl postaven na vedlejší kolej. Bettercap nabízí mnohem více možností a modulů pro řízené útoky než nástroj Ettercap, zejména útoky typu MITM. Může být použit pro ARP spoofing (neboli ARP poisoning), analýzu paketů (odkazy, přihlašovací údaje a hesla), obcházení HTTPS i HSTS, DNS spoofing (neboli DNS poisoning čili přesměrování DNS požadavku), vložením kódu do webových stránek a další. V práci bude nástroj Bettercap představen zejména k útokům typu spoofing neboli poisoning. V praxi mají oba anglické termíny stejný význam. Bettercap se spouští se příkazem přes síťové rozhraní bettercap -iface eth0 (Bettercap, 2026):



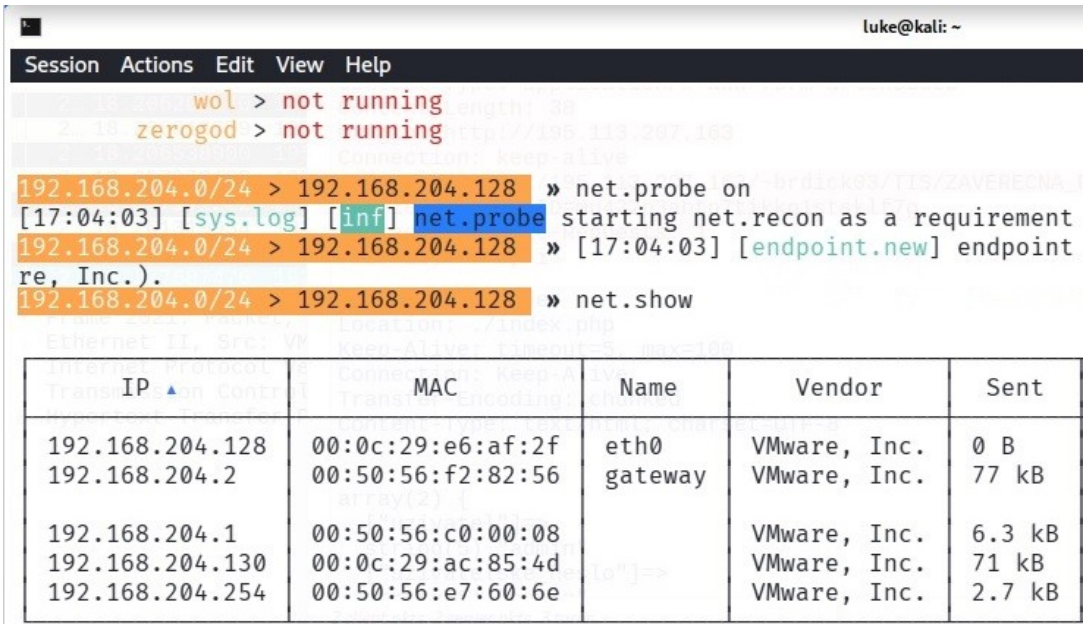
Obr. 27: Bettercap, příkaz spuštění nástroje

Zdroj: Vlastní zpracování (2026)

Nápovědu pro nástroj Bettercap lze vyvolat příkazem help, a zároveň slouží jako seznam aktivních i neaktivních modulů, které lze využít. Seznam modulů lze aktualizovat opětovným vypsáním příkazu help. V základním nastavení funguje pouze modul events.stream (modul běží v pozadí a slouží k zapisování událostí). Další moduly jsou neaktivní. Dále lze napsat za příkaz help název modulu, čímž lze vyvolat nápovědu pouze pro daný modul. Moduly se zapínají a vypínají jednoduchými příkazy, například net.probe on a net.probe off. Dále existují další příkazy pro modifikaci modulu, nejčastěji příkaz set. Základními příkazy nástroje Bettercap jsou (Bettercap, 2026):

- **net.probe on/off** - vypíše všechna zařízení připojená k síti i s MAC adresami,
- **net.recon on/off** - automaticky spuštěn příkazem net.probe. Přidá IP adresy do seznamu tak, že je lze sledovat a vyhledat,

- **net.show** - vypíše všechna možná zařízení připojená k síti.



Obr. 28: Bettercap, základní příkazy

Zdroj: Vlastní zpracování (2026)

2.3.1 ARP Spoofing

Nevýhodou protokolu ARP (Address Resolution Protocol) je jeho slabé zabezpečení a jednoduchost. ARP protokol umožňuje přiřadit IP adresy zařízení k MAC adresám. Síťová komunikace mezi zařízeními v síti potřebuje znát MAC adresy. Proto je zapotřebí ARP protokolu, aby mohl server a klient spolu komunikovat. Server využije ARP protokolu a vyšle ARP požadavek (request) všem zařízením v síti broadcast vysíláním, což znamená, že se zeptá všech zařízení v síti, kdo má konkrétní IP adresu. Na volání požadavku odpoví pouze jedno zařízení tak, že z konkrétní IP adresy odpoví svou MAC adresu. Jiná zařízení v síti volání ARP požadavky ignorují. Po sdělení MAC adresy mohou spolu zařízení komunikovat. ARP protokol je jednoduchý, skládá se pouze z požadavku a odpovědi. Každé zařízení má ARP tabulku, která překládá IP adresy v síti na MAC adresy (arp -a). Každý router s výchozí IP adresou komunikuje výhradně přes MAC adresy. MAC adresu lze lehce modifikovat ARP protokolem (Zsecurity, 2026).

ARP spoofing spočívá ve vyslání dvou ARP požadavků, jeden na router (AP) a druhý na zařízení oběti. Router, když dostane vzkaz, že IP adresa útočníka je IP adresa oběti, tak modifikuje ARP tabulku a přidělí IP adresu oběti k MAC adrese útočníka. Modifikaci lze udělat i se zařízením oběti, po vyslání ARP požadavků se předělá ARP tabulka oběti, zařízení útočníka je IP adresou routeru (AP) a přidělí IP adresu routeru k MAC adrese útočníka. Výsledkem bude, že oběť si bude myslet, že komunikuje s routerem a router si bude myslet, že komunikuje s obětí. Všechna síťová komunikace tak bude nyní chodit přes zařízení útočníka. Komunikace od oběti bude chodit do routeru přes útočníka a opačně komunikace od routeru bude chodit k oběti také přes útočníka. ARP protokol není dostatečně zabezpečen, proto lze vše zmíněné provádět. Klientské zařízení může akceptovat odpověď ARP protokolu i přesto, že nevyslali žádný ARP požadavek. Což znamená, že ARP protokol umožňuje poslat odpověď bez nutnosti vznést

požadavek, jak routeru, tak i oběti. Rovnou se akceptuje odpověď. Neexistuje ani žádné ověření. Zmiňované slabosti ARP protokolu dovolují spustit ARP spoofing útoky. V Kali Linux existuje velice jednoduchý program, který lze spustit příkazem arpspoof, bakalářská práce však využije Bettercap pro jeho více možností (Zsecurity, 2026):

```
┌──(luke㉿kali)-[~]
└─$ sudo arpspoof
[sudo] password for luke:
Version: 2.5a2
Usage: arpspoof [-i interface] [-c own|host|both] [-t target] [-r] host
arpspoof -i eth0 -t 10.0.2.7 10.0.2.1
arpspoof -i eth0 -t 10.0.2.1 10.0.2.7
```

Obr. 29: Bettercap, příkaz arpspoof

Zdroj: Vlastní zpracování (2026)

Nyní již konkrétní příklad se dvěma virtuálními OS Linux a OS Windows. Příkazem help arp.spoof lze zobrazit všechny dostupné možnosti nástroje arp.spoof, mezi které patří i příkaz arp.spoof.fullduplex. U příkazu arp.spoof.fullduplex lze nastavit hodnotu true, false a default. True znamená, že jak na router, tak i na zařízení oběti bude útočeno. Nastavení default cílí pouze na zařízení oběti, nikoli routeru, což je užitečné například pokud router má nějakou funkci ochrany proti útokům a podobně. Nastaví se postupně, případně lze sestavit i skript (Zsecurity, 2026):

- **set arp.spoof.fullduplex true** - ostatní potřebné moduly se automaticky nastaví na on,
- **set arp.spoof.targets 192.168.204.130** – cíl, oběť, na které se uplatní ARP Spoofing,
- **arp.spoof on** - zapnutí nástroje ARP Spoofing,
- **net.sniff on** - zapnutí sledování síťové komunikace na cílovém zařízení.

```
wifi > not running
wol > not running
zerogod > running
192.168.204.0/24 > 192.168.204.128 » net.show
```

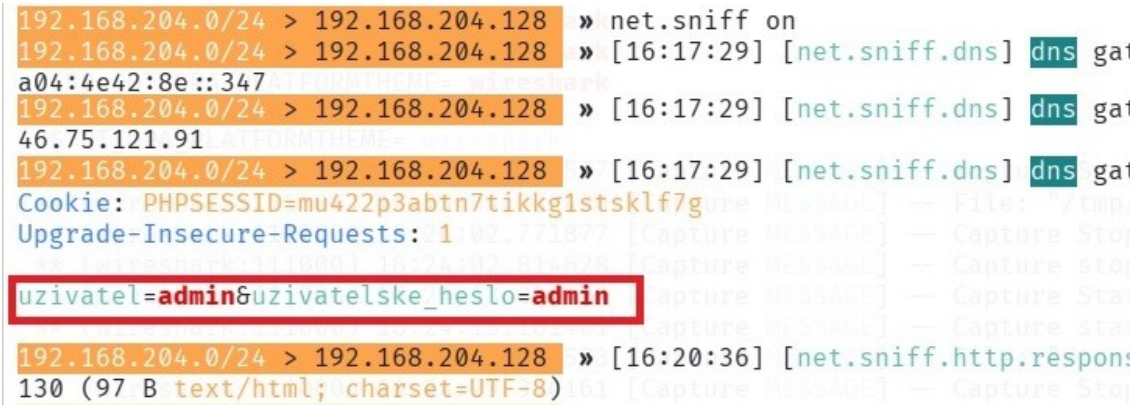
IP ▲	MAC	Name	Vendor
192.168.204.128	00:0c:29:e6:af:2f	eth0	VMware, Inc.
192.168.204.2	00:50:56:f2:82:56	gateway	VMware, Inc.
192.168.204.1	00:50:56:c0:00:08	DESKTOP-KKGBCI7.local	VMware, Inc.
192.168.204.130	00:0c:29:ac:85:4d		VMware, Inc.
192.168.204.254	00:50:56:e7:60:6e		VMware, Inc.

```
↑ 2.4 MB / ↓ 7.7 MB / 151962 pkts
192.168.204.0/24 > 192.168.204.128 » set arp.spoof.fullduplex true
192.168.204.0/24 > 192.168.204.128 » set arp.spoof.targets 192.168.204.130
192.168.204.0/24 > 192.168.204.128 » arp.spoof on
192.168.204.0/24 > 192.168.204.128 » [16:11:14] [sys.log] [war] arp.spoof ful
RP spoofing mechanisms, the attack will fail.
192.168.204.0/24 > 192.168.204.128 » [16:11:14] [sys.log] [inf] arp.spoof arp
```

Obr. 30: Bettercap, parametry příkazu arp.spoof

Zdroj: Vlastní zpracování (2026)

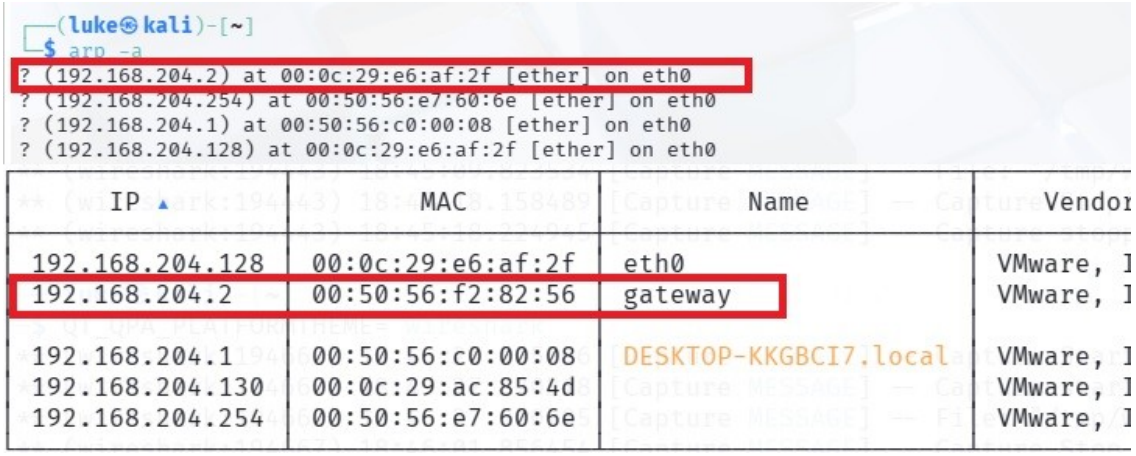
Informace získané příkazem net.sniff se získaly nešifrovaným HTTP přihlášením na zařízení s IP adresou 192.168.204.130. Podobné informace lze získat i programem Wireshark přes HTTP stream. Informace získané přes protokol HTTPS jsou opět nečitelné. Získání těchto údajů přes Wireshark je popsáno v předešlé kapitole:



Obr. 31: Bettercap, získané přihlašovací údaje

Zdroj: Vlastní zpracování (2026)

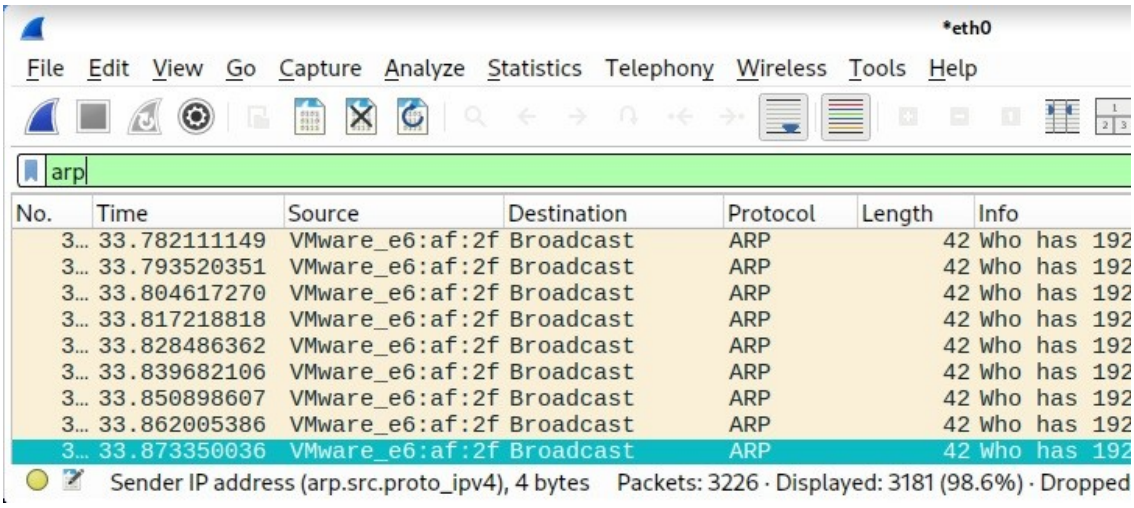
Na dalším obrázku je vidět ARP tabulka zařízení 192.168.204.130. Došlo zde k záměně MAC adresy routeru. MAC adresa routeru je shodná se zařízením 192.168.204.128:



Obr. 32: Bettercap, záměna MAC adres

Zdroj: Vlastní zpracování (2026)

ARP Spoofing útok je nejlépe vidět na paketovém snifferu Wireshark, kde útočník (192.168.204.128) neustále vysílá ARP požadavky na broadcast. Z 3226 paketů se 3181 paketů týká protokolu ARP:



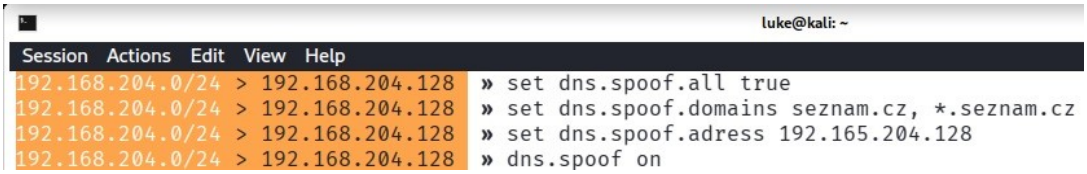
Obr. 33: Bettercap, ARP spoofing ve Wireshark

Zdroj: Vlastní zpracování (2026)

2.3.2 DNS Spoofing

DNS je server překládající názvy domén na IP adresy. Pokud se zadá název domény do prohlížeče, požadavek jde na DNS server a poté odpoví zpět IP adresou, kde je doména umístěna a prohlížeč nahraje stránku z IP adresy serveru. V případě MITM jdou požadavky na překlad názvu domén na zařízení, které stojí mezi komunikací serveru a klienta, ještě dříve než na DNS server, z čehož plyne možnost jednoduchého podvržení falešné stránky. Zařízení stojící mezi komunikací serveru a klienta je schopno podvrhnout IP adresu vyslanou na DNS server a odeslat ji zpět. Možností, co se s podvržením dá dělat, je obrovské, pokud se útočník dostane do sítě, lze podvrhnout vlastní stránku, cizí stránku z internetu, phishingovou stránku, přihlašovací stránku, škodlivý kód, malware a mnoho dalších zákeřných možností. Ve zkratce je DNS spoofing přesměrováním požadavku názvu domény kamkoliv. Je však nutné, aby zařízení bylo v komunikaci mezi serverem a klientem. Což umožňuje ARP Spoofing. ARP Spoofing může být také spuštěn, pokud chce útočník útočit technikou DNS Spoofing, což není nutností, výrazně však ulehčuje práci. Na příkladu se přesměruje doména na lokální stránky spuštěné lokálně na webovém serveru Apache. Ve webovém serveru Apache hraje důležitou roli soubor index.html v adresáři /var/www/html, který je možné upravit. Moduly arp.spoof, net.probe, net.recon, net.sniff, zerogod a events.stream v programu Bettercap již běží. Nápovědu lze opět vyvolat příkazem help dns.spoof a DNS modul je nutné nastavit na (Zsecurity, 2026):

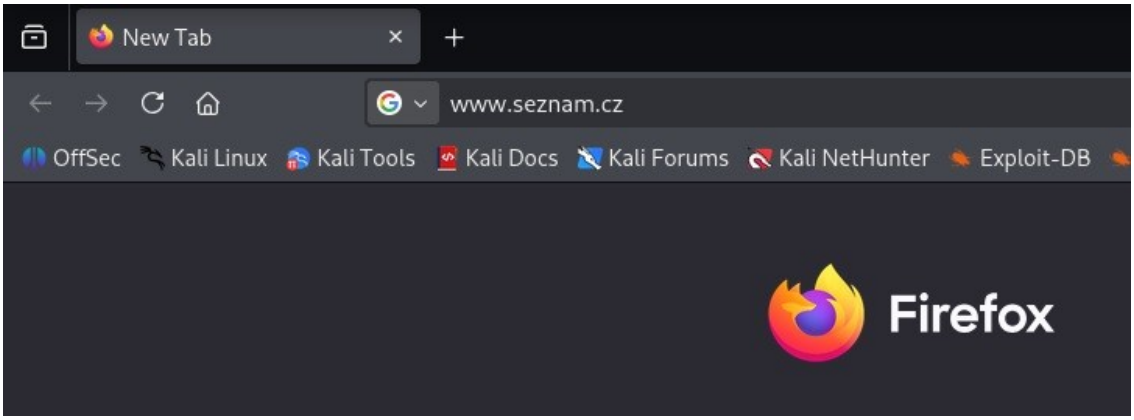
- **set dns.spoof.all true** - bettercap odpoví na všechny DNS dotazy,
- **set dns.spoof.domains seznam.cz, *.seznam.cz** – zadávaná doména, která bude přesměrována na zařízení 192.168.204.130. Hvězdička pro případ kdyby bylo do prohlížeče zadáno www.seznam.cz. Více domén se musí oddělit čárkou,
- **set dns.spoof.adress 192.168.204.128** – IP adresa zařízení, na kterém běží aktivovaný webový server Apache a na kterou bude oběť přesměrována,
- **dns.spoof on** – start modulu dns.spoof,
- **service apache2 start** – na druhém zařízení, 192.168.204.128, se musí web server Apache aktivovat. Je možné upravit soubor index.html v adresáři /var/www/html.



Obr. 34: Bettercap, parametry příkazu dns.spoof

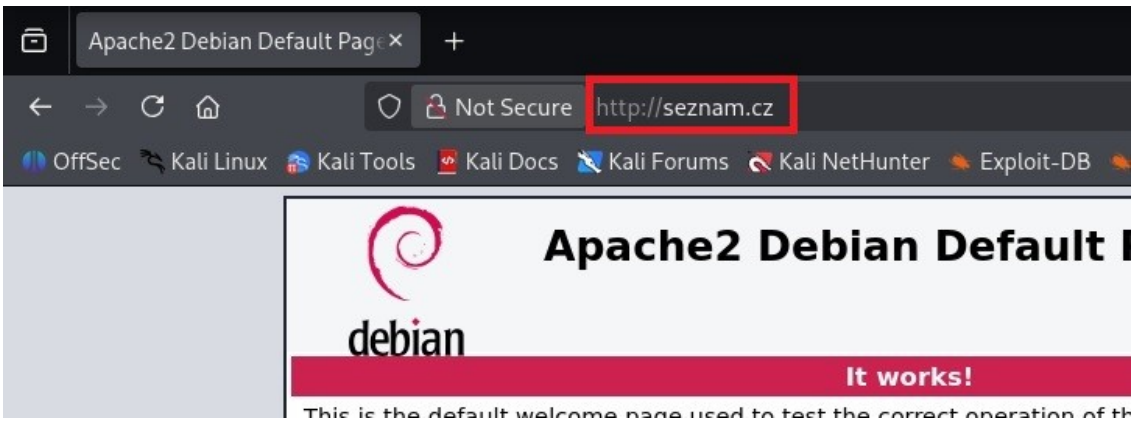
Zdroj: Vlastní zpracování (2026)

Nesmí se zapomenout aktivovat webový server na zařízení 192.168.204.128, také je dobré vymazat historii prohlížení v prohlížeči, aby stránky nezůstaly v cache paměti a zároveň chvíli počkat, než se změny zapíší do systému. Po zadání seznam.cz se požadavek přesměruje na 192.168.204.128 a v liště prohlížeče název domény seznam.cz zůstává:



Obr. 35: Bettercap, zadání seznam.cz do prohlížeče

Zdroj: Vlastní zpracování (2026)



Obr. 36: Bettercap, přesměrování DNS Spoofing

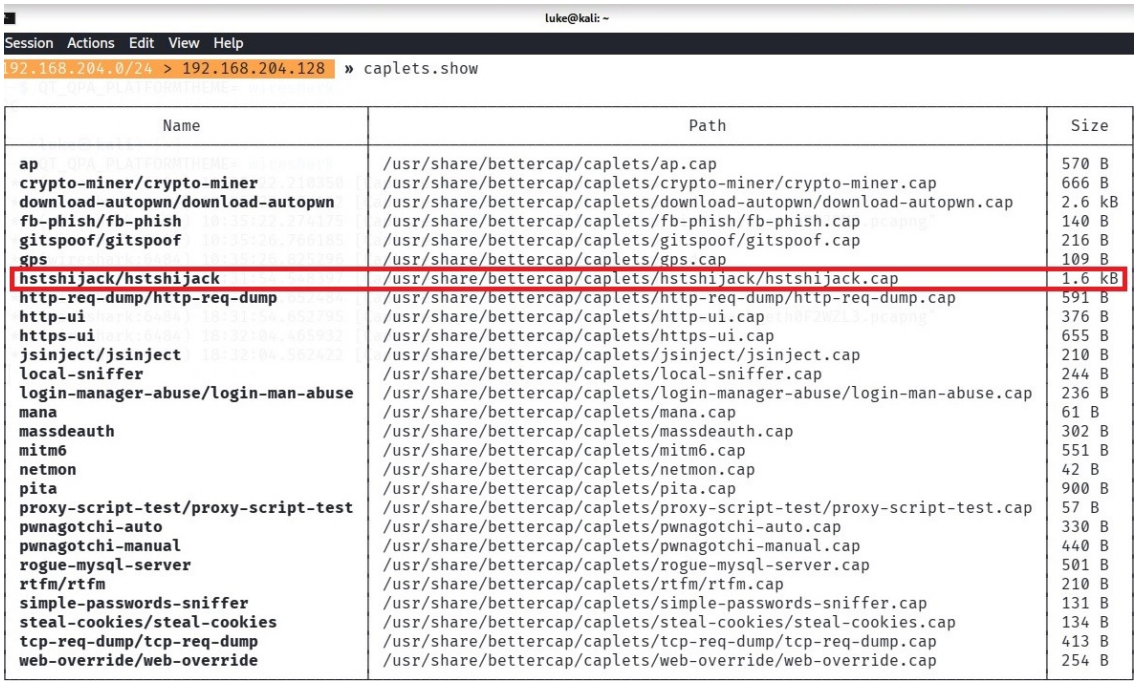
Zdroj: Vlastní zpracování (2026)

2.3.3 HTTPS, HSTS protokoly

HTTP (Hypertext Transfer Protocol) protokol komunikuje s okolím nešifrovaně. Neoprávněná osoba jej může číst, editovat a posílat dál, protože není nijak zabezpečen. Problém byl opraven v HTTPS (Hypertext Transfer Protocol Secure). HTTPS je šifrovaná, zabezpečená komunikace, pro člověka nečitelná. Vrstva Secure zaručuje šifrování dat. Používá protokol TLS (Transport Layer Security) nebo SSL (Secure Sockets Layer) k zašifrování dat a je tak velmi obtížné informace z dat získat. Nejjednodušší metodou je obejít HTTPS. Základem je podvržení stránek oběti tak, že místo klasického HTTPS šifrování stránka použije pouze HTTP. Pak bude moct

útočník získaná data přečíst jako normální text. Podvržení si však lze všimnout ukazatelem vlevo vedle lišty, kde se zadává URL stránky. Bude tam chybět ukazatel pro zabezpečené šifrování. Ukazateli zabezpečení stránek však zpravidla mnoho lidí pozornost nevěnuje, je však vidět jako znamení podvrhu. Pro zmiňované účely lze ručně nakonfigurovat a stáhnout modul sslstrip. Kali Linux obsahuje caplet hstshijack v základním nastavení, který je již nastaven pro účely podvržení protokolu HTTPS. Caplet lze editovat dle libosti. Opět je dobré preventivně před testováním vymazat historii prohlížení v prohlížeči z důvodu cache paměti. Následuje postup, jak HTTPS protokol změnit na HTTP protokol. Postup je podobný jako u kapitoly ARP Spoofing, je nutné dostat se mezi komunikaci dvou zařízení a v Bettercap postupně zadávat (HTTP Strict Transport Security (HSTS), 2026):

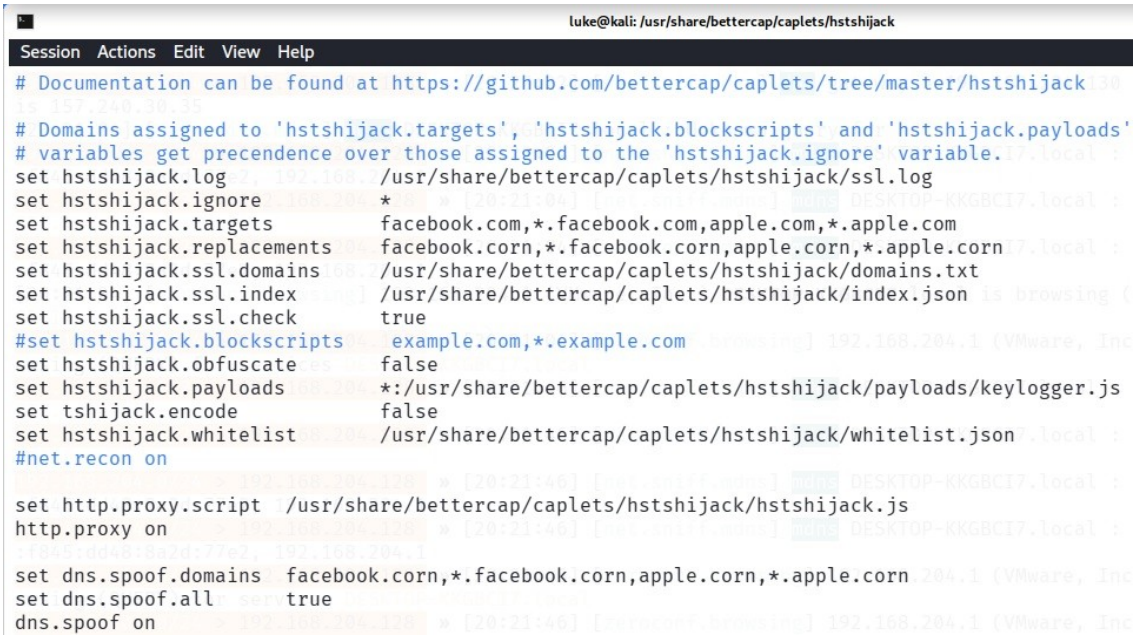
- net.probe on,
- net.recon on,
- set arp.spoof.fullduplex true,
- **set arp.targets 192.168.204.130** – zařízení oběti,
- arp.spoof on,
- **set net.sniff.local true** - Bettercap bude sniffovat i lokální data. Protože data budou přicházet ze zařízení oběti, které je také v lokální síti,
- net.sniff on,
- **caplets.show** – příkaz není nutný, jedná se o výpis seznamu capletů a jejich umístění na disku,
- **sudo apt install sslstrip** – v terminálu Linuxu se stáhne balíček, který se automaticky přesune do adresáře capletu hstsjihack v adresáři /usr/share/bettercap/caplets/hstshijack,
- **hstsjihack/hstsjihack** – příkaz opět v Bettercap, pouze se potvrdí Enterem.



Name	Path	Size
ap	/usr/share/bettercap/caplets/ap.cap	570 B
crypto-miner/crypto-miner	/usr/share/bettercap/caplets/crypto-miner/crypto-miner.cap	666 B
download-autopwn/download-autopwn	/usr/share/bettercap/caplets/download-autopwn/download-autopwn.cap	2.6 kB
fb-phish/fb-phish	/usr/share/bettercap/caplets/fb-phish/fb-phish.cap	140 B
gitspoof/gitspoof	/usr/share/bettercap/caplets/gitspoof/gitspoof.cap	216 B
gps	/usr/share/bettercap/caplets/gps.cap	109 B
hstshijack/hstshijack	/usr/share/bettercap/caplets/hstshijack/hstshijack.cap	1.6 kB
http-req-dump/http-req-dump	/usr/share/bettercap/caplets/http-req-dump/http-req-dump.cap	591 B
http-ui	/usr/share/bettercap/caplets/http-ui.cap	376 B
https-ui	/usr/share/bettercap/caplets/https-ui.cap	655 B
jsinject/jsinject	/usr/share/bettercap/caplets/jsinject/jsinject.cap	210 B
local-sniffer	/usr/share/bettercap/caplets/local-sniffer.cap	244 B
login-manager-abuse/login-man-abuse	/usr/share/bettercap/caplets/login-manager-abuse/login-man-abuse.cap	236 B
mana	/usr/share/bettercap/caplets/mana.cap	61 B
massdeauth	/usr/share/bettercap/caplets/massdeauth.cap	302 B
mitm6	/usr/share/bettercap/caplets/mitm6.cap	551 B
netmon	/usr/share/bettercap/caplets/netmon.cap	42 B
pita	/usr/share/bettercap/caplets/pita.cap	900 B
proxy-script-test/proxy-script-test	/usr/share/bettercap/caplets/proxy-script-test/proxy-script-test.cap	57 B
pwnagotchi-auto	/usr/share/bettercap/caplets/pwnagotchi-auto.cap	330 B
pwnagotchi-manual	/usr/share/bettercap/caplets/pwnagotchi-manual.cap	440 B
rogue-mysql-server	/usr/share/bettercap/caplets/rogue-mysql-server.cap	501 B
rtfm/rtfm	/usr/share/bettercap/caplets/rtfm/rtfm.cap	210 B
simple-passwords-sniffer	/usr/share/bettercap/caplets/simple-passwords-sniffer.cap	131 B
steal-cookies/steal-cookies	/usr/share/bettercap/caplets/steal-cookies/steal-cookies.cap	134 B
tcp-req-dump/tcp-req-dump	/usr/share/bettercap/caplets/tcp-req-dump/tcp-req-dump.cap	413 B
web-override/web-override	/usr/share/bettercap/caplets/web-override/web-override.cap	254 B

Obr. 37: Bettercap, seznam Capletů a jejich umístění

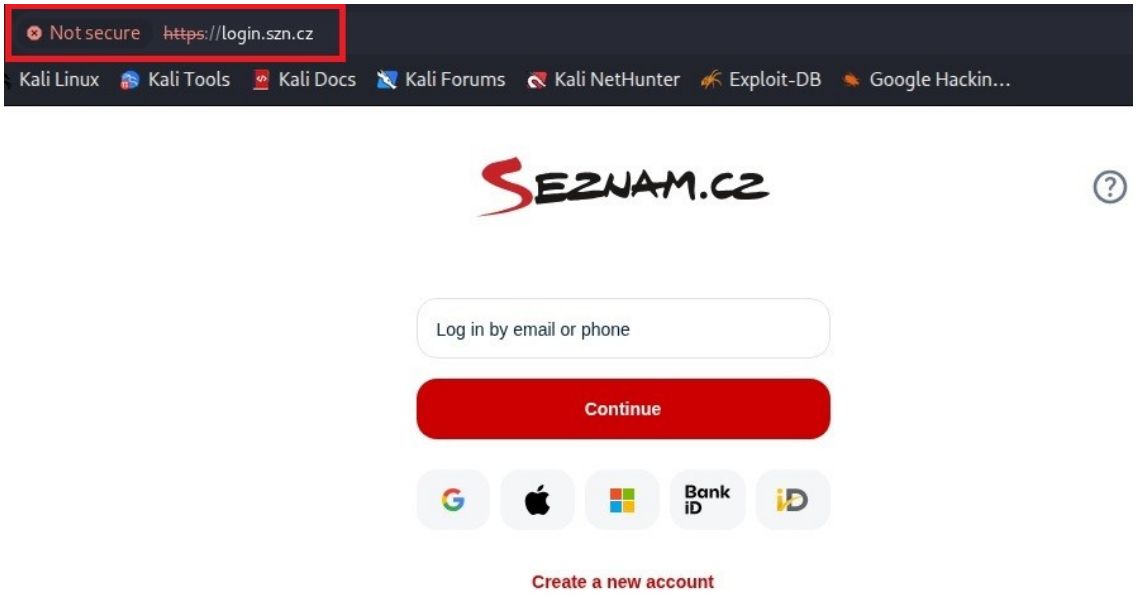
Zdroj: Vlastní zpracování (2026)



Obr. 38: Bettercap, ukázka souboru hstshijack.cap

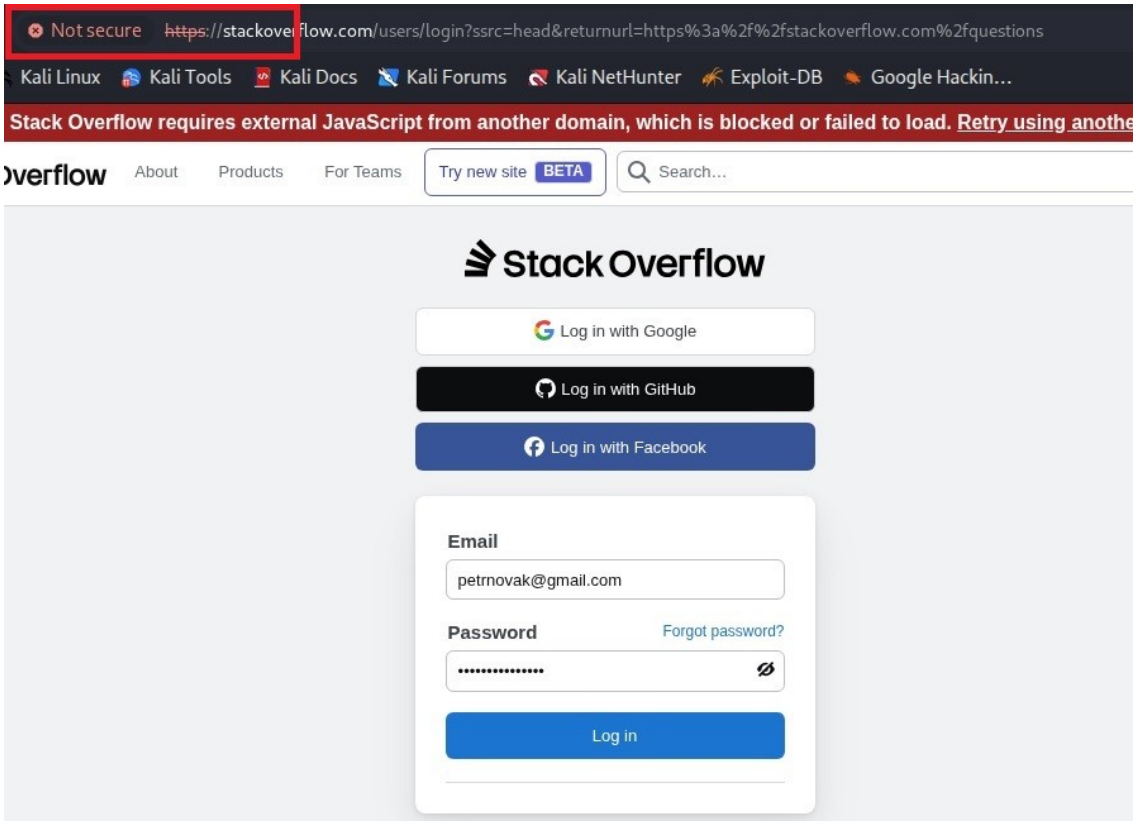
Zdroj: Vlastní zpracování (2026)

Poté, co se připojí oběť na jakoukoliv stránku s HTTPS zabezpečením, nahraje se rovnou bez HTTPS zabezpečení a tak by se měly zjistit i přihlašovací údaje přes Wireshark nebo Bettercap. Bohužel se nepodařilo získat vypsání přihlašovacích údajů, mnohokrát vyzkoušeno. Podle dostupných informací je zmiňovaná problematika často spojována s prohlížeči. Nejlepší na ochranu je Mozilla Firefox, kde je téměř nemožné načítat HTTPS stránky jako HTTP, vypíše se hlášení internet není dostupný. Další zkoušený prohlížeč Google Chromium byl dostupnější, stránky HTTPS načetl jako HTTP, nicméně ani tady se nepodařilo při přihlášení získat požadované informace. Nejvíce dostupné prohlížeče by měly být, vedle starého Internet Explorer, ty méně známé jako Vivaldi či Brave. Moderní prohlížeče jsou s každou novou aktualizací bezpečnější vůči známým útokům. Zmiňovaným způsobem lze získat přihlašovací údaje ze stránek, které používají HTTPS, což se však nepodařilo. Na stránky se lze přihlásit a i poté fungují jako HTTP stránky bez HTTPS zabezpečení, nicméně přihlašovací údaje Wireshark ani Bettercap nezachytí. Nejpopulárnější stránky jako Facebook, síť X a jiné mají ochranu zabezpečení o úroveň lepší, protokolem HSTS. HSTS protokol je těžké obejít a i k tomu poslouží soubor hststhijack.cap.



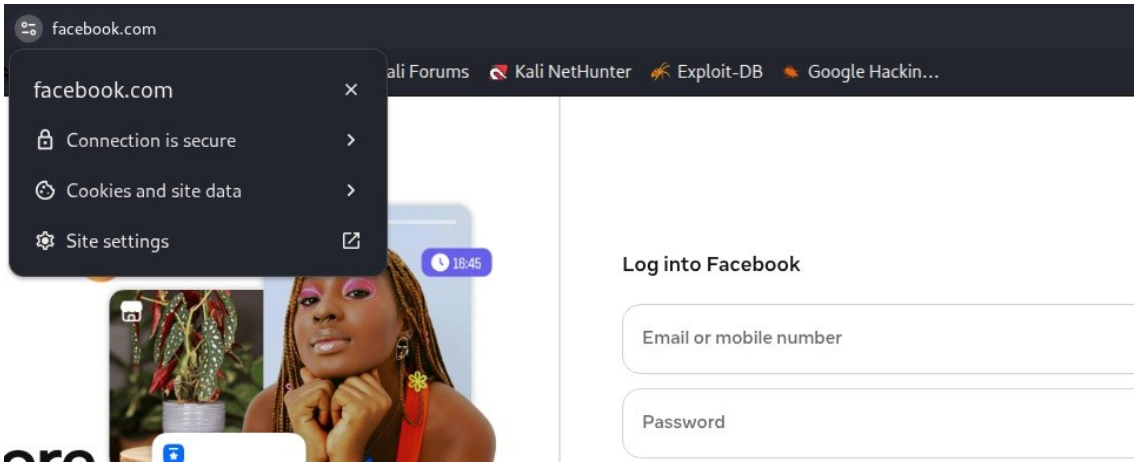
Obr. 39: Bettercap, ukázka prolomení HTTPS login.szn.cz

Zdroj: Vlastní zpracování (2026)



Obr. 40: Bettercap, ukázka prolomení HTTPS stackoverflow.com

Zdroj: Vlastní zpracování (2026)

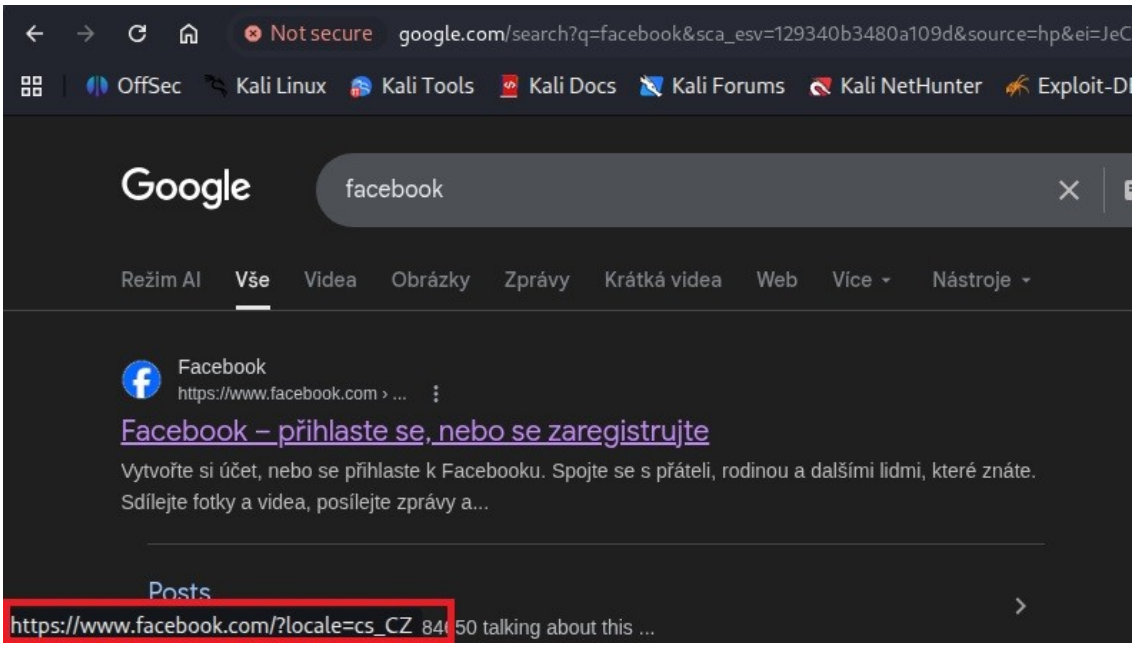


Obr. 41: Bettercap, ukázka HSTS ochrany facebook.com

Zdroj: Vlastní zpracování (2026)

HSTS (HTTP Strict Transport Security) používají nejvíce populární stránky. Jak lze spatřit na obrázku, i stránky Facebooku jsou chráněny protokolem HSTS. I kdyby se na stránky Facebooku použila metoda popsaná výše, stejně je moderní prohlížeče nahrají s HTTPS zabezpečením. V moderních prohlížečích jsou nahrány seznamy webových stránek určené pro protokol HSTS, které se nahrají jedině s HTTPS zabezpečením. Což v praxi znamená, že je nemožné prolomit HSTS protokol, protože je implementován přímo v prohlížečích, lokálně na zařízení potenciální oběti. Jediným řešením je přesvědčit prohlížeč, aby nahrál jinou stránku s dojmem, že nahrává pravou stránku. Cílem je tedy přepsat všechny odkazy na HSTS stránky podobnými URL odkazy. Například facebook.com na facebook.corn či faceboook.com, novinky.cz na movinky.cz, kali.org na kali.ong, www.kb.cz na wvvw.kb.cz a podobně. Opět se pracuje se souborem hstshijack.cap v adresáři /usr/share/bettercap/caplets/hstshijack, který musí být správně sepsán. Příkazy i postup jsou totožné jako u HTTPS prolomení na HTTP, nic se nemění. Pouze se musí editovat správně soubor hstshijack.cap. Opět se doporučuje vymazat historii prohlížeče (HTTP Strict Transport Security (HSTS), 2026).

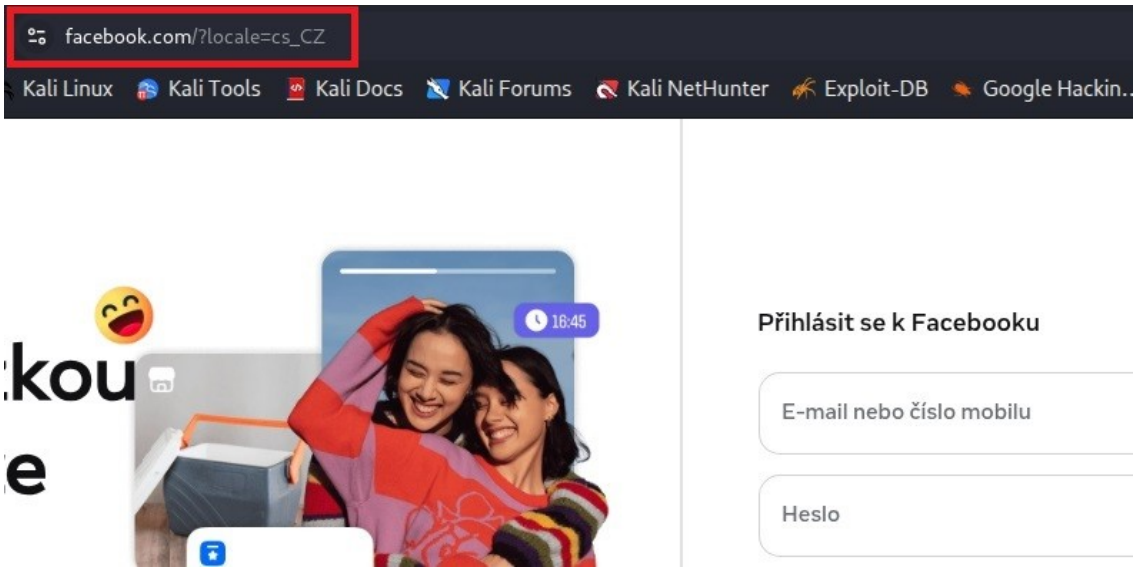
URL odkazy se při pokusu o prolomení HSTS nezadávají ručně do lišty prohlížeče, ale přes odkaz na jiné stránce či přes vyhledávače, lze i odkazem přes obrázek a podobně. Nutno však dodat, že vyhledávač musí mít prolomené HTTPS zabezpečení. Z výsledků vyhledávání lze spatřit odkazy místo facebook.com na facebook.corn. Když se klikne na odkaz Facebooku, prohlížeč jej přesměruje na facebook.corn a nahraje se stránka facebook.corn, která je naprosto totožná jako originální a původní stránka facebook.com, zároveň již není HTTPS šifrována. Po vyplnění údajů na podvržené stránce se z nešifrované komunikace tak opět získají přihlašovací údaje. Bohužel se však opět nepodařilo úspěšně dokončit zmíněnou techniku. Moderní prohlížeče mají časté bezpečnostní aktualizace a pokud se nějaká hackerská technika dostane do širšího podvědomí, autoři prohlížeče zajisté rychle vydají nové aktualizace znemožňující prolomení bezpečnosti jejich produktu. I autoři videí na YouTube prohlašují na zmíněné téma, že se jedná o techniku závislou na mnoha scénářích (HTTP Strict Transport Security (HSTS), 2026):



Obr. 42: Bettercap, HSTS ochrana, odkaz facebook.corn

Zdroj: Vlastní zpracování (2026)

Zde vlevo dole v červeném rámečku by se měl objevit odkaz facebook.corn:



Obr. 43: Bettercap, HSTS ochrana, lišta facebook.corn

Zdroj: Vlastní zpracování (2026)

Zde by se měl v liště prohlížeče objevit nápis facebook.corn, HTTPS šifrování by mělo být prolomené. Zde je patrné, že stránky Facebooku jsou chráněny HSTS protokolem. Zároveň po vyplnění přihlašovacích údajů by mělo dojít k jejich zachycení.

2.4 Nmap – skenování portů

Nmap, Network Mapper, je mutliplatformní open-source nástroj pro práci se sítí. Jedná se o program, který prozkoumává síťová zařízení a zjišťuje, zda jsou zařízení v síti aktivní, jaké porty mají otevřené a jaké služby na portech běží, dokáže zjistit dokonce i, v jaké verzi jsou na

portech služby dostupné. Dokáže také odhalit operační systém vzdáleného zařízení a pomocí skriptovacího systému NSE provádět pokročilé testy, které odhalují chyby v konfiguraci nebo odhalují zranitelnosti sítě. V praxi Nmap představuje nástroj, který umožní odhadnout strukturu sítě. Správcům sítí pomáhá udržovat pořádek a bezpečnost, bezpečnostním specialistům umožňuje odhalit rizika dříve, než je někdo zneužije, a běžnému uživateli může odhalit, jaká všechna zařízení jsou připojena do jeho domácí sítě. Nmap dokáže zmapovat strukturu sítě tak, že zmapuje celou síť. Zasílá vybrané typy paketů na specifické IP adresy a analyzuje odpovědi. Nmap rovněž disponuje grafickým rozhraním, které se nazývá Zenmap. Komunikace probíhá na základě transportních protokolů, z nichž nejvíce používané jsou TCP (Transmission Control Protocol) a UDP (User Datagram Protocol), které zajišťují přenos dat (Nmap, 2026).

Každý z protokolů má vlastní sadu portů, které jsou číslované od jedné do 65535. Za konkrétním portem se skrývá konkrétní služba. Porty jsou standardizované a rozdělené do tří základních skupin. Plně standardizované, dobře známé porty, mají čísla od jedné do 1023. Na zmiňovaných portech běží nejobvyklejší služby, například HTTP (port 80, protokolu TCP), HTTPS (port 443, protokolu TCP) či DNS (port 53, protokolu UDP i TCP). Další střední skupina jsou registrované porty od 1024 do 49151 a jejich použití je omezované. Při použití portu aplikací by se měl port registrovat a oznámit, že se bude používat. Z hlediska mapování portů střední skupiny je možnost mapování pouze na konkrétní službu. Poslední skupina portů jsou porty dynamické, od 49152 do 65535. Nejsou pevně zařazené ani registrované, používají se dynamicky. Pokud jedno zařízení komunikuje s jiným, tak se otevře jeden z dynamických portů na straně jedné a rovněž dynamicky se otevře i port pro komunikaci na straně druhé. Čili nelze zjistit s jistotou, co se za dynamickým portem skrývá za službu, jen, že je otevřený. U všech očíslovaných portů nelze však také s jistotou říci, že například na portu 80 běží služba HTTP. Na jakých portech jaké služby běží určuje administrátor. Díky základnímu nastavení sítě však služby díky číslování portů odhadnout lze. Pokud se například oskenuje adresní rozsah nějaké firmy, lze zjistit, že na těchto IP adresách je otevřený port 80 nebo port 443, což znamená, že tam budou nějaké webové servery a už hacker ví, kde útočit nebo kde navázat komunikaci (Nmap, 2026).

Tab. 6: Přehled nejzákladnějších portů

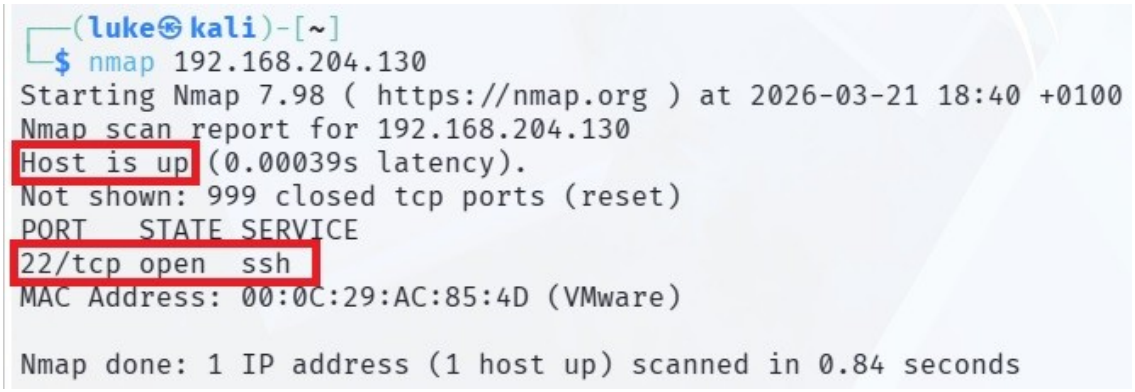
Služba	***Port***	***Daemon***	***Verze***	***Popis***
FTP	21	vsftpd	3.0.5	FTP server
SSH	22	ssh	9.7	Vzdálený přístup
DNS	53	bind9	9.20.1	DNS server
HTTP	80/443	apache2, httpd	2.4.62	Webový server
NTP	123	chronyd	4.5	Časový server
Samba	445	smbd	4.20.0	Sdílení souborů
CUPS	631	cups	2.4.10	Tiskový server
Proxy	8080	squid	6.10	Proxy server

Zdroj: vlastní zpracování dle Microsoft Copilot: váš AI pomocník (2025)

2.4.1 Základní příkazy

Příkazy Nmap mají ohromné množství parametrů a možností, které lze najít na manuálových stránkách nástroje příkazem man nmap či nmap -h. Mezi základní příkazy Nmap se řadí (Nmap, 2026):

nmap 192.168.204.130 – základní příkaz bez jakéhokoliv parametru, cílený na jednu IP adresu, jedno zařízení v síti. Zařízení v síti je aktivní (host is up) a port číslo 22 protokolu TCP je otevřen se službou SSH:



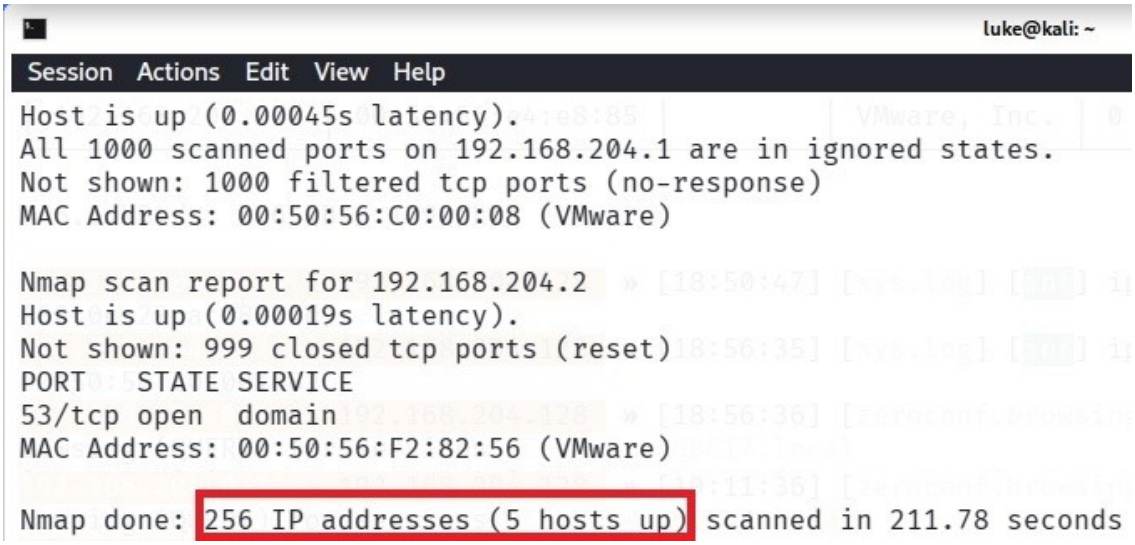
Obr. 44: Nmap, příkaz Nmap na jedno zařízení

Zdroj: Vlastní zpracování (2026)

Podobně je možné skenovat i jednotlivá zařízení v síti příkazy:

- **nmap 192.168.204.130 192.168.204.128 192.168.204.2** – více zařízení najednou,
- **nmap 192.168.204.0,1,2,3,4** – opět více zařízení najednou oddělené pouze čárkou,
- **nmap scanme.nmap.org** - lze skenovat i textovou doménu.

nmap 192.168.204.0/24 – skenování všech zařízení v síti s maskou 24. Vypíše informace o všech aktivních zařízení v síti:



Obr. 45: Nmap, příkaz Nmap na celou síť

Zdroj: Vlastní zpracování (2026)

Další příkazy, kterými je také možné skenovat všechna zařízení v síti:

- **nmap 192.168.204.***– skenuje všechna zařízení v síti,

- **nmap 192.168.204.130-254** - skenování části sítě dle rozsahu IP adres.

nmap -p 22 192.168.204.130 - skenování pouze vybraného portu 22 na zařízení 192.168.204.130. Pokud se napíše příkaz bez uvedení parametru p, automaticky Nmap naskenuje tisíc nejpoužívanějších portů:

```
┌──(luke㉿kali)-[~]
└─$ nmap -p 22 192.168.204.130
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-21 19:26 +0100
Nmap scan report for 192.168.204.130
Host is up (0.00059s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:AC:85:4D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.76 seconds
```

Obr. 46: Nmap, příkaz na skenování portů

Zdroj: Vlastní zpracování (2026)

Opět existuje celá řada možností, jaké zvolit příkazy při skenování portů:

- **nmap -p 22,80,443 192.168.204.130** – skenuje pouze vybrané porty,
- **nmap -p 22-80 192.168.204.130 –** skenuje rozsah portů,
- **nmap -p- 192.168.204.130** – skenuje všechny porty od 0 do 65535,
- **nmap -p T:* 192.168.204.130** – skenuje pouze všechny TCP porty,
- **nmap --top-ports 10 192.168.204.130** – skenuje pouze nejpoužívanějších 10 portů.

nmap -sV 192.168.204.130 – malé s v tomto parametru značí typ skenu a za typ skenu (za malé s) se přidávají další možnosti. V tomto případě se jedná o užitečný parametr velké V, které nám odhalí verze služeb na daných portech, což může být užitečné při průniku do sítě. Takto lze skenovat i celou síť, nejen jedno zařízení:

```
┌──(luke㉿kali)-[~]
└─$ nmap -sV 192.168.204.130
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-21 20:16 +0100
Nmap scan report for 192.168.204.130
Host is up (0.00056s latency).
Not shown: 996 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 10.2p1 Debian 2 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.65 ((Debian))
139/tcp open  netbios-ssn Samba smbd 4
445/tcp open  netbios-ssn Samba smbd 4
MAC Address: 00:0C:29:AC:85:4D (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results
Nmap done: 1 IP address (1 host up) scanned in 12.29 seconds
```

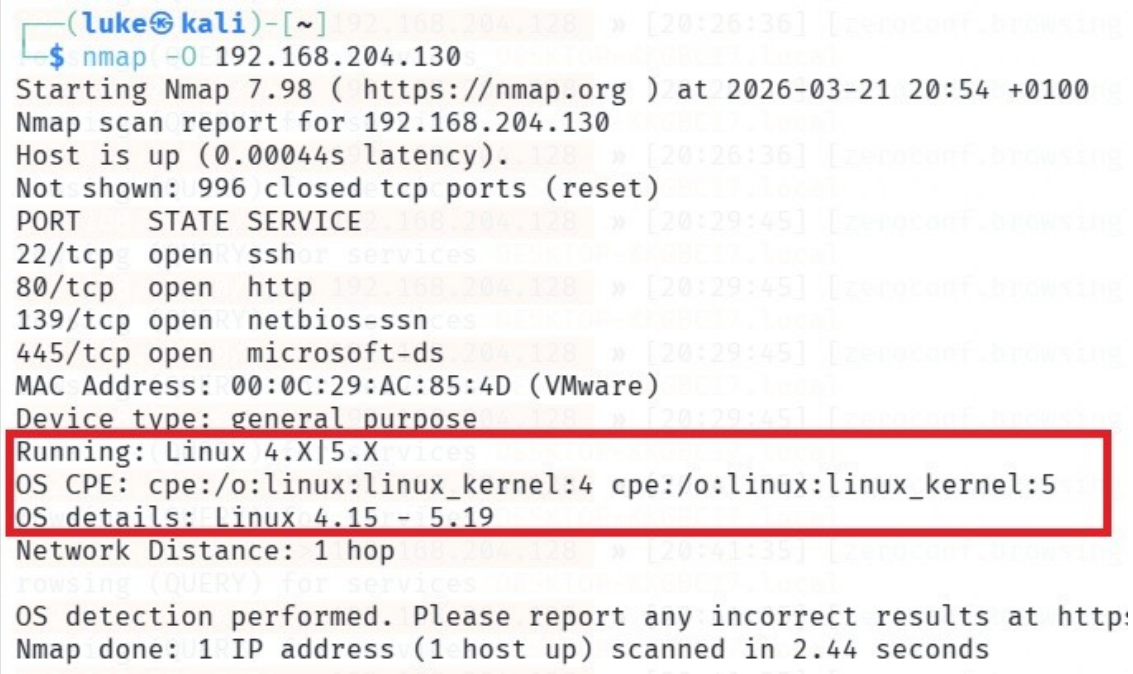
Obr. 47: Nmap, příkaz na skenování služeb

Zdroj: Vlastní zpracování (2026)

Další možnosti přepínače malé s:

- **nmap -sS 192.168.204.130** - skenuje nejpoužívanějších tisíc TCP portů,
- **nmap -sU 192.168.204.130 –** skenuje pouze UDP porty,
- **nmap -sP 192.168.204.0/24** - ping aktivních zařízení v síti, bez skenování portů,
- **nmap -sn 192.168.204.0/24** – také ping aktivních zařízení v síti, bez skenování portů,
- **nmap -sX 192.168.204.130** - Xmas scan, skenuje TCP porty pomocí speciálních flagů.
- **nmap -sA 192.168.204.130** – ACK scan, zjišťuje zda porty filtruje firewall,
- **nmap -sT 192.168.204.130** - použije plný three-way handshake,
- **nmap -sN 192.168.204.130** - speciální stealth technika bez TCP flagů,
- **nmap -sC 192.168.204.130** – skenuje verze služeb a spustí základní NSE skripty pro detekci služeb.

nmap -O 192.168.204.130 – Nmap detekujte operační systém zařízení:



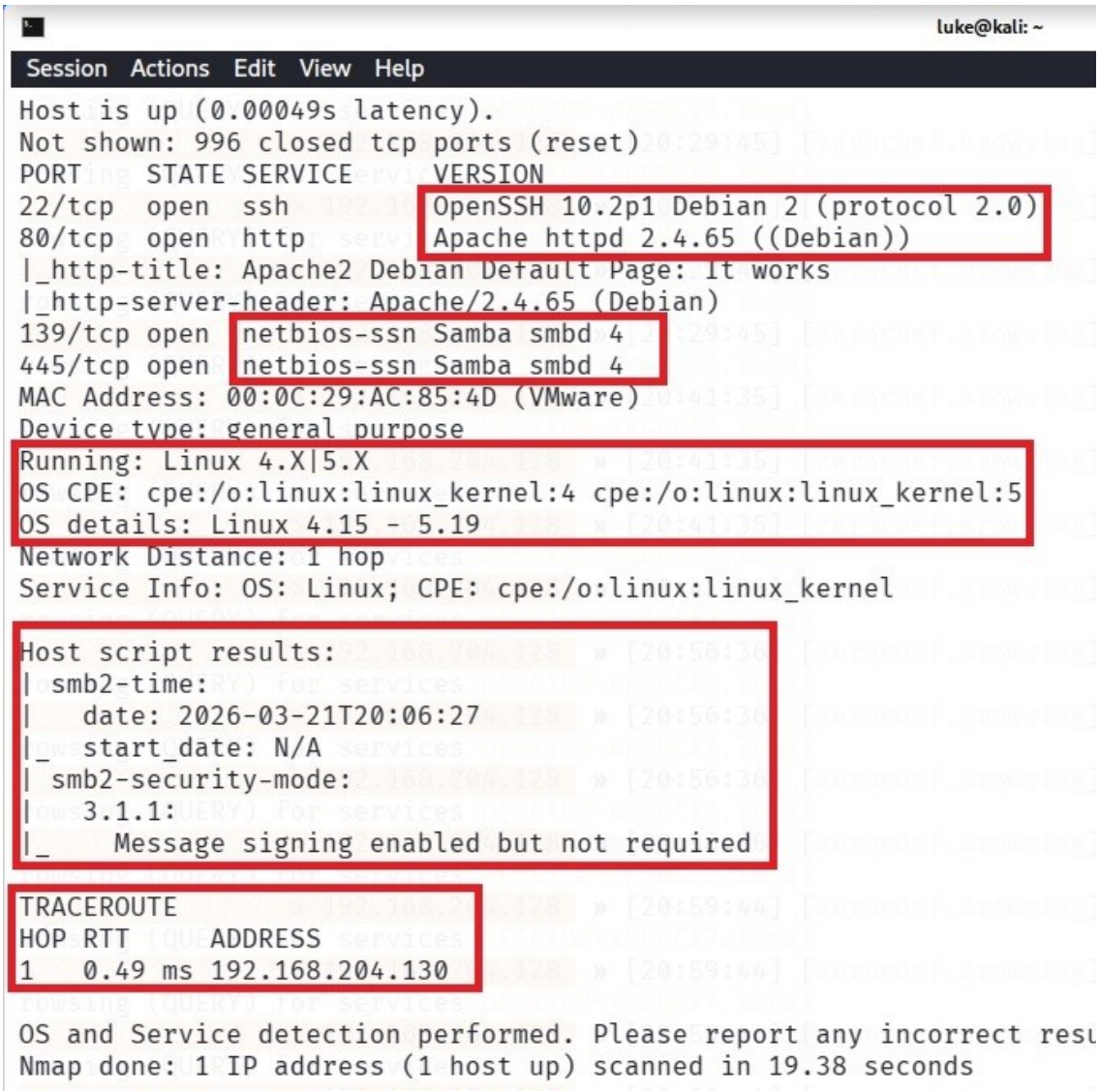
Obr. 48: Nnmap, příkaz na detekci OS

Zdroj: Vlastní zpracování (2026)

Další možnosti parametru velké O mohou být:

- **nmap -O --osscan-limit 192.168.204.130** - provádí detekci OS pouze u hostů,
- **nmap -O 192.168.204.130 -vvvv** – detekce OS s velmi podrobným verbose výstupem.

nmap -A 192.168.204.130 - aggressive scanning, dokáže zobrazit nejvíce detailů o portech v jednom příkazu. Vyhledá informace o službách a jejich verzích i operačních systémech, provede traceroute a spustí základní NSE skripty pro detekci služeb:



Obr. 49: Nmap, příkaz na aggressive scanning

Zdroj: Vlastní zpracování (2026)

nmap -oX output.xml 192.168.204.0/24 - uloží výsledky do XML souboru output.xml, který lze snadno převést do HTML nebo do jiných formátů, což je užitečné pro generování síťových map:

Obr. 50: Nmap, příkaz pro scan output

Zdroj: Vlastní zpracování (2026)

Další možnosti práce se soubory či výstupů do souborů mohou být:

- **nmap -oN output.txt 192.168.204.0/24** - uloží výsledky skenu do textového souboru,
- **nmap -oG output.txt 192.168.204.0/24** - uloží výsledky skenu do grepovatelného formátu (Grepable Output),
- **nmap -oA output 192.168.204.0/24** - uloží výsledky skenu do tří formátů najednou (Nmap, XML a Grepable),
- **nmap -iL /input.txt** – nástroj bude skenovat seznam IP adres z textového souboru.

Některé parametry nástroje Nmap lze mezi sebou různě kombinovat, mezi další užitečné parametry, které lze při různých příkazech použít se řadí (Nmap, 2026):

-T0 až -T5 – rychlost skenu, čím vyšší číslo, tak rychlejší a agresivnější sken, ale také vyšší riziko falešných výsledků, blokace nebo odhalení.

-v až -vvvv - zvyšuje množství zobrazovaných informací (verbose), čím více v, tak podrobnější výstup.

Ostatní užitečné linuxové příkazy:

- **service --status-all | grep +** - zobrazí pouze běžící služby,
- **ss -tulnp** – zobrazí naslouchající porty a procesy,
- **netstat -tulnp** - starší, ale stále používaný nástroj podobný příkazu ss.

2.4.2 Nmap Scripting Engine

Skripty Nmap Scripting Engine (NSE) jsou nedílnou součástí instalačního balíčku Nmap a zároveň jeho nejsilnější stránkou. Skripty jsou uložené v adresáři /usr/share/nmap/scripts a mají příponu *.nse. Celkem jich je okolo 600 (další je možné stáhnout ze stránek nmap.org) a je možné mezi nimi vyhledat konkrétní skript použitím příkazu grep. Dále umožňují vytvářet, spouštět a sdílet vlastní automatizované skripty psané v programovacím jazyce Lua a jejich cílem je automatizovat pokročilé skenovací úlohy, jako jsou objevování sítě, detekce zranitelností a jejich zneužívání, testování hrubou silou, zjišťování služeb a jejich verzí, detekce zadních vrátek, sběru informací a podobně. NSE skripty tedy jednoduše pomáhají skenování částečnou automatizací a je možné využít stávající skripty nebo je možné si je vytvořit. V případě vlastního skriptu je nutné jej pojmenovat s příponou *.nse a umístit jej do správného adresáře. Pro použití základních NSE skriptů se používá přepínač -sC, podobně jako přepínač -A (-sC je ekvivalentem pro --script=default a provádí sken skriptů spuštěním výchozí sady základních skriptů). Každý skript obsahuje pole, které skript přiřazuje k jedné nebo více kategoriím. Složitější výběr skriptů lze provádět pomocí operátorů and, or a not pro tvorbu booleovských výrazů. Operátory mají stejnou přednost jako v každém typickém programování, přednost se může změnit pomocí závorek. Pro spuštění skriptů se používá příkaz nmap --script s doplněním názvu konkrétního skriptu a cíle. Na jeden cíl lze pochopitelně využít více skriptů v jednom příkazu. Zobrazení nápovědy pro konkrétní skript lze například pro skript ftp-vsftpd-backdoor.nse, způsobem --script-help ftp-vsftpd-backdoor.nse. NSE skripty lze aktualizovat pomocí příkazu nmap --script-updatedb (Nmap Script Engine (NSE), 2025):

```
┌──(luke㉿kali)-[~]
└─$ sudo nmap --script-updatedb
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-22 19:35 +0100
NSE: Updating rule database.
NSE: Script Database updated successfully.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.51 seconds
```

Obr. 51: Nmap, skripty Nmap, aktualizace

Zdroj: Vlastní zpracování (2026)

```
┌──(luke㉿kali)-[/usr/share/nmap/scripts]
└─$ ll /usr/share/nmap/scripts/ | wc -l
614
```

Obr. 52: Nmap, celkový počet skriptů

Zdroj: Vlastní zpracování (2026)

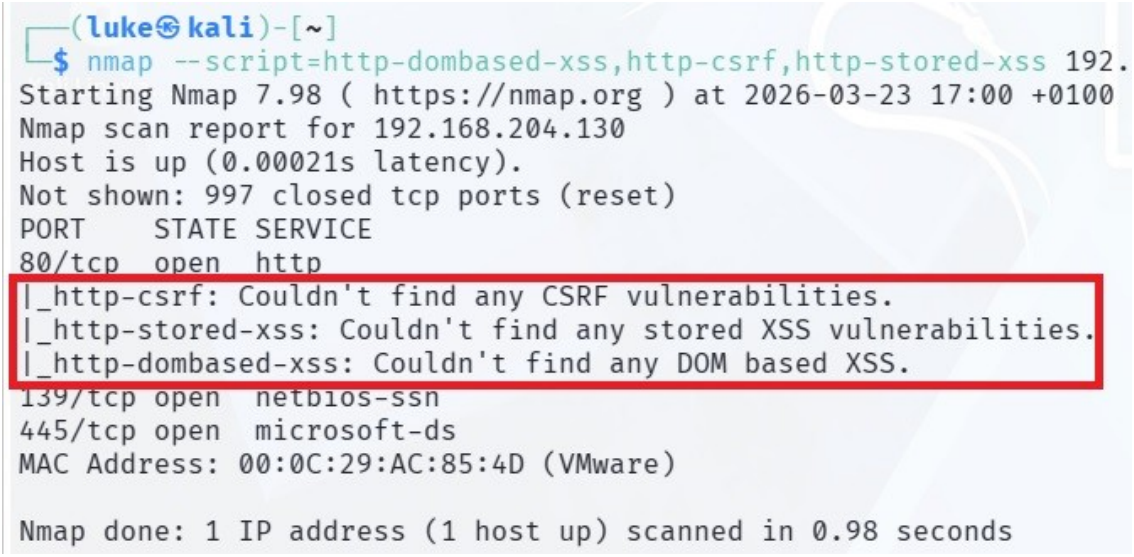
Jednotlivé konkrétní NSE skripty jsou součástí konkrétní kategorie, k nimž patří. Což v praxi znamená, že například skripty http-dombased-xss, http-csrf, http-stored-xss, smb-vuln-ms10-061, smb-vuln-ms10-054 a samba-vuln-cve-2012-1182 jsou součástí kategorie skriptů vuln (kategorie je nutné psát malými písmeny), a pokud se spustí celý skript, takto nmap --script=vuln 192.168.204.130, spustí se všechny skripty kategorie vuln. Ke spuštění pouze jednoho jediného skriptu kategorie vuln je nutné konkrétní skript zapsat například takto, nmap --script=http-dombased-xss 192.168.204.130 (Nmap Script Engine (NSE), 2025):

```
luke@kali: ~
Session  Actions  Edit  View  Help

┌──(luke㉿kali)-[~]
└─$ nmap --script=vuln 192.168.204.130
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-23 16:42 +0100
Nmap scan report for 192.168.204.130
Host is up (0.00033s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE
80/tcp  open  http
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:AC:85:4D (VMware)

Host script results:
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: ERROR:
more fields are missing); aborting [9]
|_smb-vuln-ms10-054: false
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB:
one or more fields are missing); aborting [9]

Nmap done: 1 IP address (1 host up) scanned in 42.12 seconds
```

Obr. 53: Nmap, výpis kategorie vuln

Zdroj: Vlastní zpracování (2026)

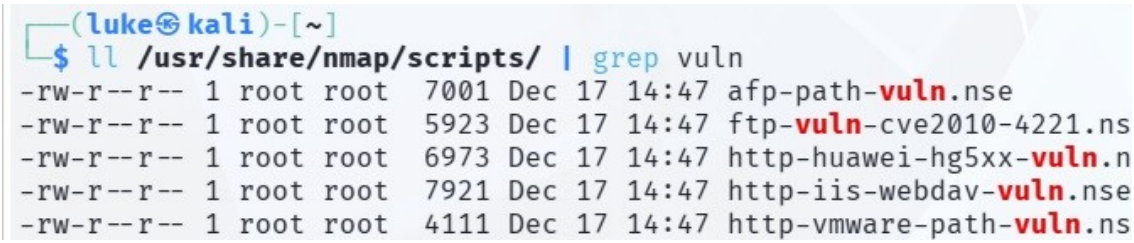
Z pokusu vyplývá, že HTTP skripty http-dombased-xss: Couldn't find any DOM based XSS (skript nenašel žádnou DOM-based XSS zranitelnost), http-csrf: Couldn't find any CSRF vulnerabilities (nenašel žádnou CSRF zranitelnost), http-stored-xss: Couldn't find any stored XSS vulnerabilities (nenašel žádnou uloženou (stored) XSS zranitelnost) a Samba skripty smb-vuln-ms10-061: Could not negotiate a connection (skript se nedokázal připojit k SMB službě, takže test neproběhl), smb-vuln-ms10-054: false (test proběhl a zranitelnost nebyla nalezena) a samba-vuln-cve-2012-1182: Could not negotiate a connection (opět se nepodařilo navázat SMB spojení, takže skript nemohl nic otestovat). Výsledek je, že port 80 je otevřený a HTTP skripty se spustily, ale nic nenašly. Porty 139 a 445 jsou otevřené a SMB skripty se pokusily spustit, ale některé se nepřipojily a jeden test vyšel jako nezranitelný. Spuštění pouze zkoušených HTTP skriptů kategorie vuln, aniž by se uplatnily všechny skripty kategorie vuln:



Obr. 54: Nmap, výpis HTTP skriptů

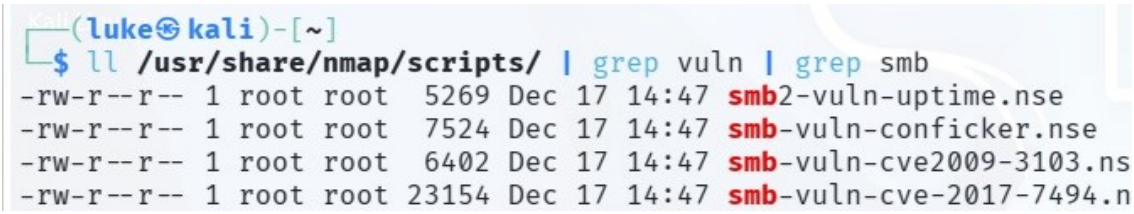
Zdroj: Vlastní zpracování (2026)

Výběrem určitých HTTP skriptů lze výsledek konkretizovat. Z adresáře umístění skriptů lze vygrepovat pouze skripty kategorie vuln či pouze skripty kategorie vuln, týkající se například služby Samba:



Obr. 55: Nmap, výpis skriptů kategorie vuln

Zdroj: Vlastní zpracování (2026)



Obr. 56: Nmap, výpis skriptů kategorie vuln, služba samba

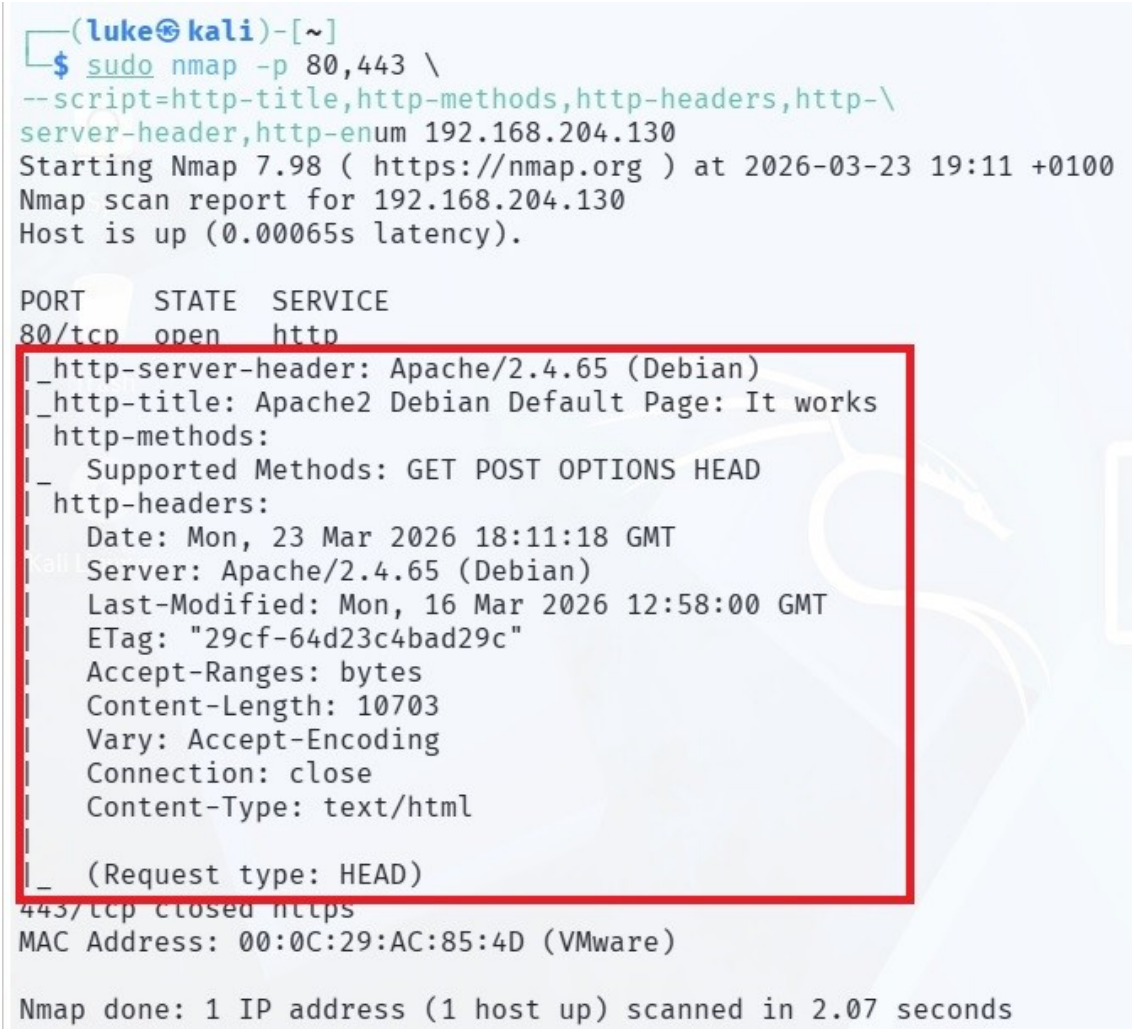
Zdroj: Vlastní zpracování (2026)

Následující seznam popisuje každou kategorii skriptů (Nmap Script Engine (NSE), 2025):

- **auth** - práce s autentizací (přihlášení, obcházení, anonymní přístup),
- **broadcast** - hledání hostů v lokální síti pomocí broadcastu,
- **brute** - hrubá síla, zkoušení hesel a přihlašovacích údajů,
- **default** - skripty, které se spouští automaticky při -sC nebo -A,
- **discovery** - zjišťování informací o síti a službách (titulek webu, SMB sdílení, SNMP info),
- **dos** - testy, které mohou způsobit DoS nebo zatížení cíle,
- **exploit** - aktivní zneužití konkrétní zranitelnosti,
- **external** - skripty komunikující s externími službami (například whois),
- **fuzzer** - posílání náhodných/neočekávaných dat pro hledání chyb,
- **intrusive** - rizikové skripty, mohou být škodlivé nebo zatěžující,
- **malware** - detekce známých malware chování nebo backdoorů,
- **safe** - bezpečné skripty, minimální riziko, nezatěžují systém,
- **version** - pomáhají s detekcí verzí služeb (pouze při -sV),
- **vuln** - kontrola konkrétních známých zranitelností.

Skripty mají i možnost zadání argumentů v příkazové řádce, kde vedle kategorie či konkrétního skriptu je možné doplnit příkaz i s argumentací (například přidáním wordlist.txt při slovníkovém útoku). Možné je při psaní skriptů do příkazové řádky použít i booleovské výrazy. Dále i módy bezpečnosti default, safe, not intrusive, default or safe, default and safe, default or safe or intrusive. Nmap je skutečně rozsáhlý nástroj s mnoha možnostmi, které jsou uvedeny na oficiálních stránkách nástroje Nmap a na jeho manuálových stránkách. Na stránkách Nmap jsou mimo jiné uvedeny i seznamy konkrétních skriptů a jejich kategorií s popisy. Důležité při psaní vícero skriptů, je dodržet následující nmap --script=skript1,skript2,skript3 (čárka bez mezer) nebo nmap --script=skript1 --script=skript2 --script=skript3 (Nmap Script Engine (NSE), 2025).

Příklad č. 1 - sudo nmap -p 80,443 --script=http-title,http-methods,http-headers,http-server-header,http-enum 192.168.204.130:



Obr. 57: Nmap, příklad první

Zdroj: Vlastní zpracování (2026)

Výsledek příkladu č. 1 - na portu 80 běží Apache 2.4.65 na Debianu, server podporuje metody GET, POST, OPTIONS a HEAD. Posílá standardní HTTP hlavičky včetně typu obsahu, délky a data poslední úpravy. Titulek stránky je výchozí. Port 443 je uzavřený a skript http-enum nenašel žádné další cesty, jedná se o čistou instalaci Apache bez dalších služeb nebo aplikací.

Příklad č. 2 - sudo nmap -p 22,80,443,445 --script=discovery,safe,version 192.168.204.130:

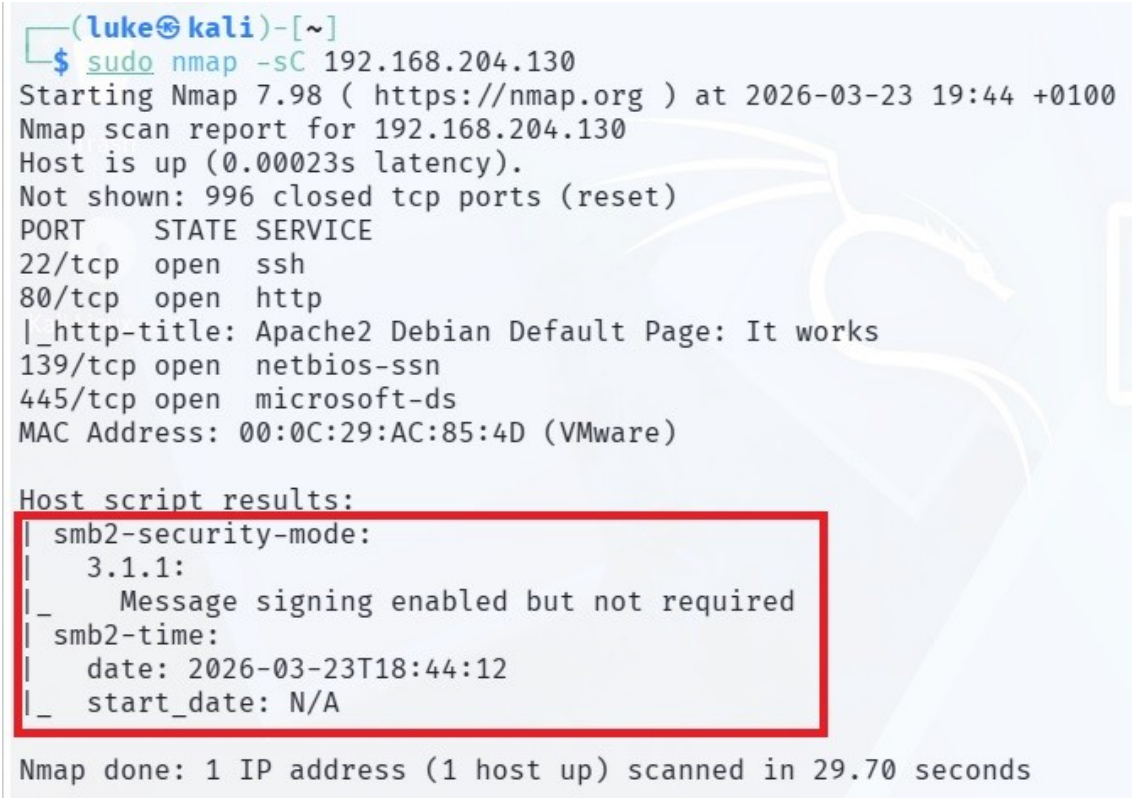
```
┌──(luke㉿kali)-[~]
└─$ sudo nmap -p 22,80,443,445 \
--script=discovery,safe,version \
192.168.204.130
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-23 19:34 +0100
No profinet devices in the subnet
Pre-scan script results:
| targets-ipv6-multicast-echo:
|   IP: fe80::20c:29ff:feac:854d  MAC: 00:0c:29:ac:85:4d  IFACE: e
|_  Use --script-args=newtargets to add the results as targets
| broadcast-ping:
|   IP: 192.168.204.2  MAC: 00:50:56:f2:82:56
|_  Use --script-args=newtargets to add the results as targets
| eap-info:
| Available authentication methods with identity="anonymous" on in
|   unknown  EAP-TTLS
|   unknown  EAP-TLS
|   unknown  EAP-MSCHAP-V2
|_  unknown  PEAP
| targets-ipv6-multicast-invalid-dst:
|   IP: fe80::20c:29ff:feac:854d  MAC: 00:0c:29:ac:85:4d  IFACE: e
|_  Use --script-args=newtargets to add the results as targets
Bug in broadcast-upnp-info: no string output.
| broadcast-igmp-discovery:
|   192.168.204.1
|     Interface: eth0
|     Version: 2
|     Group: 224.0.0.251
|     Description: mDNS (rfc6762)
|   192.168.204.1
|_    Interface: eth0
```

Obr. 58: Nmap, příklad druhý

Zdroj: Vlastní zpracování (2026)

Výsledek příkladu č. 2 - výsledky kategorií skriptů discovery, safe a version neříkají nic o cílovém serveru, ale místo toho spustily hromadu broadcast a síťových discovery skriptů, které jen ukázaly zařízení a multicast provoz v celé síti (DHCP server, router, multicast skupiny, IPv6 adresy, IGMP, LLMNR a mDNS). Skripty nejsou zaměřené na konkrétní porty, ale na síť jako celek.

Příklad č. 3 - sudo nmap -sC 192.168.204.130 (sudo nmap --script=default 192.168.204.130):

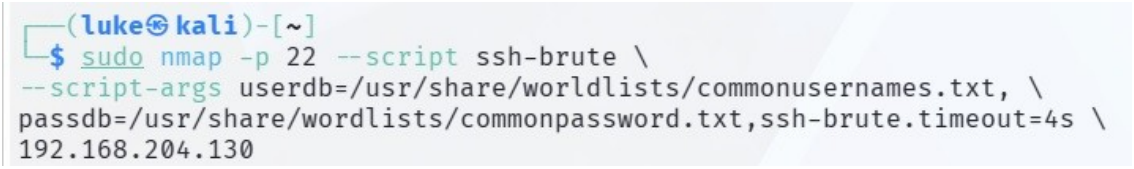


Obr. 59: Nmap, příklad třetí

Zdroj: Vlastní zpracování (2026)

Výsledek příkladu č. 3 - server má otevřené porty SSH (22), HTTP (80) a SMB (139/445), web běží na výchozí Apache stránce, SMB používá protokol 3.1.1 s povoleným (ale nevyžadovaným) podpisováním zpráv a SMB server ukázal svůj aktuální čas. Žádné další informace ani zranitelnosti.

Ukázka kódu - brute force útoku na port 22 se službu OpenSSH a použitím slovníkového útoku s wordlistem jako argumentem v příkazu (Nmap Script Engine (NSE), 2025):



Obr. 60: Nmap, ukázka kódu

Zdroj: Vlastní zpracování (2026)

2.4.3 Generování síťových map

Generování síťových map je velice jednoduchý proces, při kterém se získá mnoho informací o zařízeních připojených do sítě a mnoho dalších užitečných informací o síti. Síťové mapy jsou velice snadno zneužitelné na veřejných Wi-Fi sítích bez možnosti autentizace heslem. Každé zařízení připojené k veřejné síti se tak snadno může stát terčem útoku. Celou síť lze naskenovat pomocí nástroje Nmap během několika kroků (Nmap, 2026):

nmap -T5 -A -v 192.168.204.0/24 -oX network-map.xml - naskenuje celou síť do souboru network-map.xml. XML soubor je nečitelný pro člověka, avšak dá se převést na HTML soubor (-A aggressive - vše, -oX - output soubor xml, -v - verbose, -T5 - nejvyšší rychlost):

```
┌──(luke㉿kali)-[~]
└─$ nmap -T5 -A -v 192.168.204.0/24 -oX network-map.xml
Starting Nmap 7.98 ( https://nmap.org ) at 2026-03-23 20:07 +0100
NSE: Loaded 158 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 20:07

┌──(luke㉿kali)-[~]
└─$ ll
total 104
-rw-rw-r-- 1 luke luke   116 Mar 16 12:48 bettercap.history
drwxr-xr-x 2 luke luke  4096 Mar 10 09:55 Desktop
drwxr-xr-x 2 luke luke  4096 Mar 10 09:55 Documents
drwxr-xr-x 2 luke luke  4096 Mar 10 09:55 Downloads
drwxr-xr-x 2 luke luke  4096 Mar 10 09:55 Music
-rw-rw-r-- 1 luke luke 65588 Mar 23 20:09 network-map.xml
drwxr-xr-x 2 luke luke  4096 Mar 10 09:55 Pictures
drwxr-xr-x 2 luke luke  4096 Mar 10 09:55 Public
drwxr-xr-x 2 luke luke  4096 Mar 10 09:55 Templates
drwxr-xr-x 2 luke luke  4096 Mar 10 09:55 Videos
```

Obr. 61: Nmap, XML soubor síťové mapy

Zdroj: Vlastní zpracování (2026)

xsltproc ./network-map.xml -o ./network-map.html – převede soubor *.xml na *.html.

firefox network-map.html – čitelná, naskenovaná mapa sítě, vytvořená nástrojem Nmap:

Obr. 62: Nmap, HTML soubor síťové mapy, adresy

Zdroj: Vlastní zpracování (2026)

Ports

The 996 ports scanned but not shown below are in state: **closed**

- 996 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product
22	tcp	open	ssh	syn-ack	OpenSSH
80	tcp	open	http	syn-ack	Apache httpd
	http-title	Apache2 Debian Default Page: It works			
	http-methods	Supported Methods: GET POST OPTIONS HEAD			
	http-server-header	Apache/2.4.65 (Debian)			
139	tcp	open	netbios-ssn	syn-ack	Samba smbd
445	tcp	open	netbios-ssn	syn-ack	Samba smbd

Remote Operating System Detection

- Used port: **22/tcp** (**open**)
- Used port: **1/tcp** (**closed**)
- Used port: **38684/udp** (**closed**)
- OS match: **Linux 4.15 - 5.19** (**100%**)

Host Script Output

Script Name	Output
smb2-time	date: 2026-03-23T19:08:21 start_date: N/A
smb2-security-mode	3.1.1: Message signing enabled but not required

Obr. 63: Nmap, HTML soubor síťové mapy, porty

Zdroj: Vlastní zpracování (2026)

Ostatní užitečné linuxové příkazy:

- **netdiscover** – zobrazí všechna připojená zařízení v síti a seřadí je do tabulky,
- **nc -nvlp 1234** - poslouchání portu pomocí Nmap, kde -LP je poslech na portu 1234,
- **searchsploit** – slouží k rychlému vyhledávání známých zranitelností v Exploit-DB.

2.5 Aircrack-ng – bezdrátové útoky

Aircrack-ng je kompletní sada nástrojů určená k penetračnímu testování bezdrátových sítí. Slouží k analýze a testování zabezpečení bezdrátových sítí, pracuje s různými typy šifrování a protokolů. Skládá se z několika dílčích nástrojů, z nichž každý má svou originální funkci. Zde jsou vyjmenovány ty nejdůležitější (How to Use Aircrack-ng: A Guide to Network Compromise, 2025):

- **airbase-ng** - vytváření softwarového přístupového bodu pro testování sítí,
- **aircrack-ng** - analýza zachycených paketů a testování síťového zabezpečení,
- **airdecap-ng** - dešifrování zachycených paketů, pokud je k dispozici klíč,
- **aireplay-ng** – generování rámců pro testování chování Wi-Fi,
- **airserv-ng** - sdílení Wi-Fi karty přes síť,

- **airmon-ng** - přepnutí Wi-Fi adaptéru do monitorovacího režimu,
- **airodump-ng** - pasivní zachytávání paketů a informací o sítích,
- **airodump-ng-oui-update** - aktualizace databáze výrobců MAC adres,
- **airtun-ng** - tvorba virtuálních tunelů pro experimentální účely,
- **besside-ng** - automatizované testování zabezpečení Wi-Fi, pasivní sběr dat.

K zachytávání paketů a vytváření handshake ze síťového provozu při využití nástroje aircrack-ng je zapotřebí mít výkonný Wi-Fi adaptér, který podporuje monitor mode, packet injection, má správné ovladače, případně podporu i 5 GHz a stabilitu při přepínání režimu z monitor mode na managed mode. Většina Wi-Fi adaptérů má dostupný pouze managed mode, což je aktivní, běžný režim, kterým je připojení na Access Point (router). Přijímá a odesílá rámce, což je základní mód pro normální používání internetu. Některé Wi-Fi adaptéry se mohou přepnout na monitor mode, což je i potřebné pro práci s aircrack-ng. Monitor mode pasivně zachycuje rámce. Wi-Fi adaptér odposlouchává, co se vysílá vzduchem a vidí všechny typy rámců, beacon, probe i data. Je tak schopen odhalit informace o blízkých routerech, včetně jejich MAC adres. Proto jsou Wi-Fi sítě tak nebezpečné, protože jsou viditelné. Monitor mode nic neovlivňuje, pouze přijímá a z toho důvodu také během monitor mode není dostupný internet. Packet injection je aktivní vysílání generovaných speciálních rámců, které mohou ovlivnit ovládání Wi-Fi sítě tak, že vyvolají reakci routeru, například opětovným přihlášením, změnou kanálu a zejména deautentizace. Wi-Fi adaptéry mohou být interní i externí. Interní, nasazené na základní desce, jsou spolehlivější a stabilnější oproti tomu, externí Wi-Fi USB adaptéry dosahují většího rozsahu díky anténě a celkově většího výkonu. Důležitý je čipset Wi-Fi adaptéru, kde absolutně kralují čipsety od firmy Atheros, díky open-source ovladačům. Králem mezi Wi-Fi USB adaptéry s výbornou podporou Linuxu je AWUS036NHA s čipsetem Atheros AR9271 a ovladačem ath9k_htc od firmy Alfa Network, který je však dnes k sehnání pouze bazarově. Pro účely bakalářské práce byl použit USB Wi-Fi adaptér AWUS036ACHM s čipsetem MediaTek MT7610U, ovladačem mt76, podporou 5Ghz a výbornou podporou Linuxu od firmy Alfa Network. Nastavení síťového rozhraní na monitor mode lze několika způsoby (How to Use Aircrack-ng: A Guide to Network Compromise, 2025):

- **Příkazy pro vytvoření mode monitor na vlastní kartě** - ifconfig (zjistí se rozhraní), iwconfig (nastavení rozhraní), sudo systemctl stop NetworkManager (deaktivace sítě), sudo ip link set wlan0 down (deaktivace rozhraní), sudo iw wlan0 set monitor control (mode monitor), sudo ip link set wlan0 up (aktivace rozhraní). Během testování musí síť zůstat deaktivovaná. Pokud se síť aktivuje, wlan0 se přepne do managed mode. Při navrácení do původního stavu se opakuje postup odzadu, jen příkaz pro managed mode je sudo iw wlan0 set type managed. Aktivuje monitor mode přímo na vlastní síťové kartě s názvem původního rozhraní,
- **Příkazy pro vytvoření mode monitor na přidané virtuální kartě** - sudo iw dev wlan0 interface add mon0 type monitor (přidání virtuálního rozhraní mon0), sudo ip link set mon0 up (aktivace rozhraní) a do původního stavu zpět pomocí sudo ip link set mon0 down (deaktivace rozhraní), sudo iw dev mon0 del (výmaz rozhraní). Přidává nové virtuální síťové rozhraní s názvem mon0 pro monitor mode a rozhraní wlan0 zůstává,
- **Příkazy pomocí nástroje aircrack-ng pro vytvoření monitor mode na původní kartě** - sudo airmon-ng start wlan0 (přidání a start virtuálního rozhraní mon0), sudo airmon-ng stop wlan0mon (odebrání a deaktivace virtuálního rozhraní mon0). Zmiňovaný

způsob je ideální pro práci s nástroji aircrack-ng, avšak původní síťové rozhraní wlan0 je nahrazeno rozhraním s názvem wlan0mon.

Tab. 7: Přehled zabezpečení bezdrátových sítí

Typ	***Šifrování***	***Popis***	***Handshake***
WEP	RC4	Nepoužívané, zastaralé a prolomitelné zabezpečení.	Shared Key
WPA	TKIP	Aircrack-ng dokáže zachytit handshake a ověřit sílu hesla.	PSK (Pre-Shared Key)
WPA2	AES-CCMP	Stejné jako u WPA, nejpoužívanější současné zabezpečení.	PSK (Pre-Shared Key)
WPA3	AES-CCMP/GCMP	Nejodolnější a nejnovější, kde heslo nelze zjistit, dokážou se však zachytit rámce.	SAE

Zdroj: vlastní zpracování dle *Microsoft Copilot: váš AI pomocník* (2026)

Ze seznamu zabezpečení bezdrátových sítí se zabezpečení WEP prakticky již nikde nepoužívá, jeho zabezpečení je slabé a snadno prolomitelné. WPA se používá zřídka a také již nepatří v současné době mezi používané zabezpečení. WPA3-SAE je zase příliš moderní a vysoce zabezpečené, nicméně jeho rozšíření je stále vzácné. WPA2-PSK je standardem dnešní doby a jeho zabezpečení je vysoké. Stále ale platí, že silné heslo dokáže zabezpečit bezdrátovou síť lépe bez ohledu na její typ než heslo 123 se zabezpečením WPA3-SAE (Bezdrátová síť, 2025).

2.5.1 Prolomení zabezpečení WEP

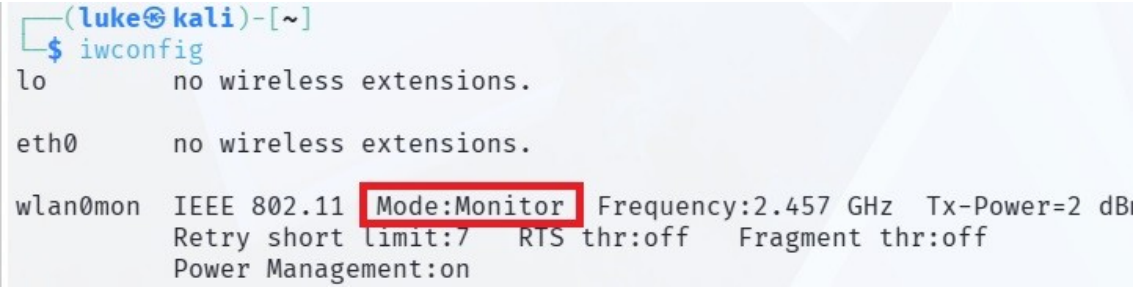
Zabezpečení typu WEP se dnes prakticky již nevidí, přesto se někde ještě používá. Postup pro prolomení zabezpečení je velice podobný jako u prolomení zabezpečení typu WPA/WPA2 při použití sady nástrojů aircrack-ng. Vzhledem k zastaralému typu šifrování RC4 a inicializačnímu vektoru (IV) o délce 24 bitů s rychlým opakováním je velice snadné zabezpečení WEP prolomit. Právě kvůli rychlému opakování může útočník pakety porovnávat a odvodit tak heslo. Postup prolomení lze s několika výjimkami provést totožně. Jelikož se zde jedná o zabezpečení typu WEP a není tedy zapotřebí handshake, je lépe provést útok na AP příkazem s parametrem --fakeauth čili sudo aireplay-ng --fakeauth 0 místo parametru --deauth 0. Parametr --deauth způsobí AP odpojení od sítě, kdežto --fakeauth se snaží přesvědčit AP, že útočník je legitimní klient a posílá tak rámce potřebné k naplnění souboru *.cap pakety s IV, které poté slouží k prolomení hesla. Ze souboru *.cap se poté nástrojem aircrack-ng získá heslo ve tvaru odděleném dvojtečkami, dvojtečky se poté musí z výsledku odstranit (Tutorial: How to Crack WPA/WPA2, 2010).

2.5.2 Prolomení zabezpečení WPA/WPA2

Postup při prolomení bezpečnosti WPA sadou nástrojů aircrack-ng je naprosto totožný jako postup při prolomení bezpečnosti WPA2, oba typy zabezpečení se liší jen ve způsobu šifrování. Na pokusy při testování byl použit USB Wi-Fi adaptér AWUS036ACHM, který dokáže monitor mode i injection mode, s čipsetem MediaTek MT7610U a ovladačem mt76 od firmy Alfa Network. Po připojení do USB portu se nemusí nic instalovat ani ve Windows. Ve VMware

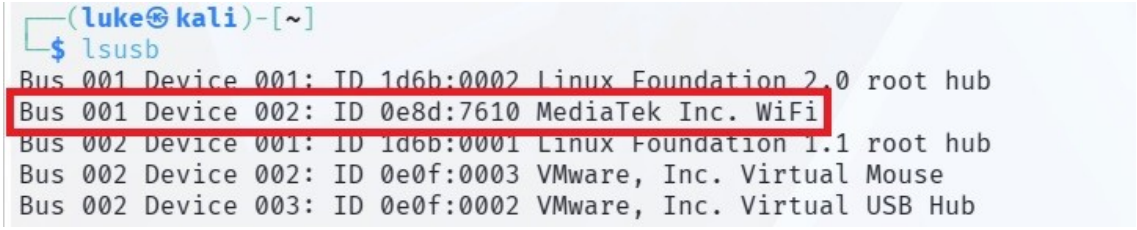
Workstation se připojí nejrychleji do virtuálního systému ikonou vpravo dole, kde jsou umístěné hardwarové nástroje. Wi-Fi adaptér je zde pojmenován čipsetem MediaTek WiFi. Poté se jen potvrdí možnost Connect to host. Při zapojení do běžícího virtuálního systému se systém zeptá automaticky. Poté ihned funguje bez jakékoliv instalace ovladačů. Správné nastavení adaptéru lze ověřit příkazy (Tutorial: How to Crack WPA/WPA2, 2010):

- **ifconfig** – dostupné rozhraní wlan0,
- **iwconfig** – vypsání módu rozhraní wlan0, buď managed nebo monitor,
- **lsusb** – vypsání adaptéru na USB portu, MediaTek Inc. WiFi,
- **cat /etc/os-release** – informace o aktuálnosti distribuce Kali Linux,
- **uname -a** – informace o aktuálnosti jádra a systému.



Obr. 64: Aircrack-ng, příkaz iwconfig

Zdroj: Vlastní zpracování (2026)



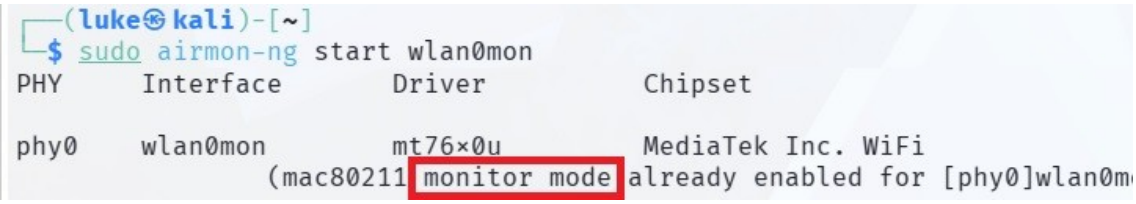
Obr. 65: Aircrack-ng, příkaz lsusb

Zdroj: Vlastní zpracování (2026)

Kroky k prolomení zabezpečení WPA/WPA2 za pomocí nástroje aircrack-ng jsou následující:

sudo airmon-ng check kill – doporučený příkaz, vypne všechny procesy překážející adaptéru plně vykonávat svou funkci. Rovněž je lze vypnout příkazem pro ukončení procesu kill -9 PID. PID všech překážejících procesů vypíše následující příkaz sudo airmon-ng start wlan0.

sudo airmon-ng start wlan0 – doporučený postup pro nástroje aircrack-ng při nastavení módu adaptéru na monitor mode. Rozhraní wlan0 se tak přejmenuje na wlan0mon. Lze ověřit příkazy iwconfig nebo sudo airmon-ng start wlan0mon:

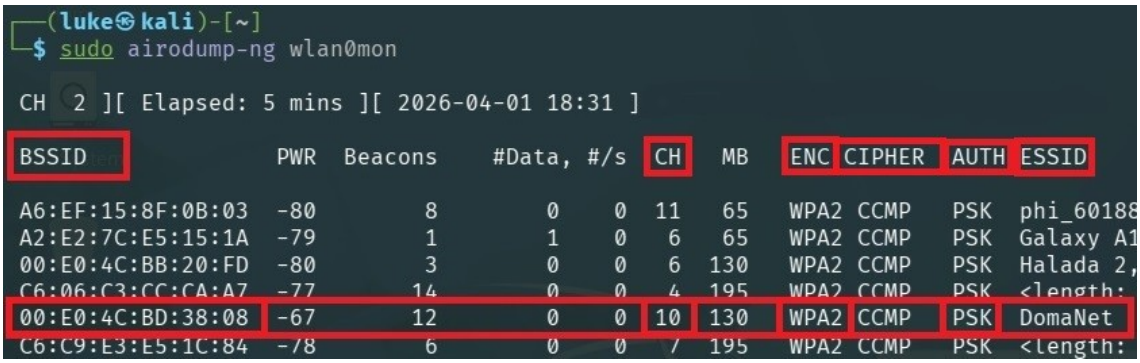


Obr. 66: Aircrack-ng, příkaz airmon

Zdroj: Vlastní zpracování (2026)

sudo airodump-ng wlan0mon – zde lze podle BSSID nalézt MAC adresu AP (routeru). Důležité je také zkontrolovat dle řádku BSSID, CH (Channel), ENC, CIPHER, AUTH a ESSID. Beacons značí

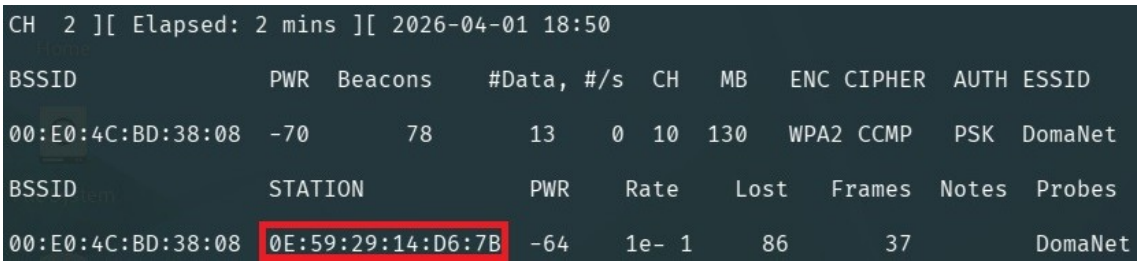
počet majákových rámců vysílaných Wi-Fi routerem, aby o sobě dal vědět. Výpis se ukončí CTRL+C:



Obr. 67: Aircrack-ng, příkaz airdump

Zdroj: Vlastní zpracování (2026)

sudo airodump-ng wlan0mon -d 00:E0:4C:BD:38:08 – příkaz airodump s konkrétní MAC adresou vybere pouze jeden router či více routerů, pokud se zadá více MAC adres. Zde konkrétně je vidět i zařízení (mobilní telefon jako station) připojené k síti:

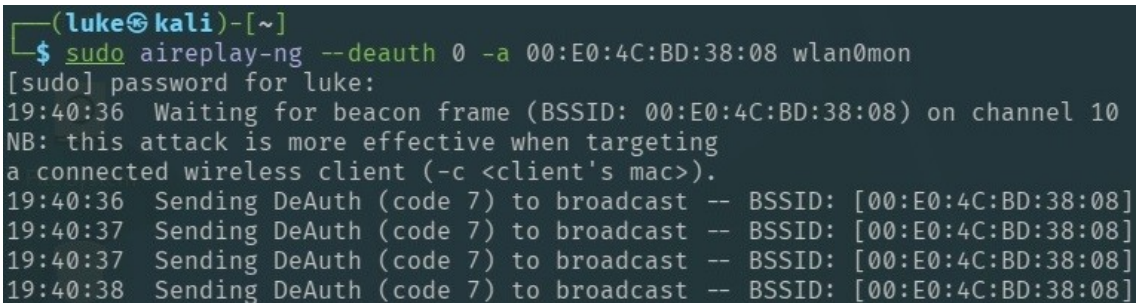


Obr. 68: Aircrack-ng, příkaz airdump, station

Zdroj: Vlastní zpracování (2026)

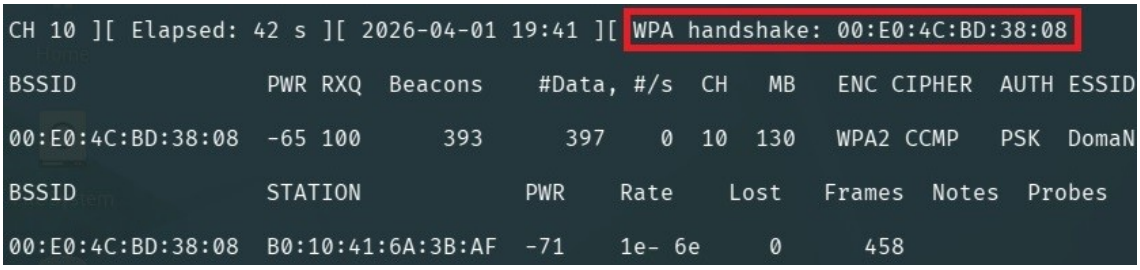
sudo airodump-ng --bssid 00:E0:4C:BD:38:08 -c 10 -w DomaNet wlan0mon – příkaz airodump s vybranou MAC adresou AP, kanálem (-c), zápisem do souboru (-w) a vybraným rozhraním. Zápisem do souboru se rozumí pojmenování souborů, které se získají z paketu, který se zachytí a uloží na disk. Důležitý je zde zejména soubor s příponou *.cap, kde jsou uloženy informace o heslu. Dále například soubor *.csv s informacemi o síti. Po spuštění příkazu se zápisem do souboru je nutné otevřít další okno terminálu, kde se vypíše příkaz na deauthentizaci AP. V prvním okně s terminálem se bude sledovat zachycení WPA handshake a ve druhém okně se spustí příkaz na deautentizaci AP.

sudo aireplay-ng --deauth 0 -a 00:E0:4C:BD:38:08 wlan0mon – deauthentication 0 znamená počet deautentizací použitých proti AP, deautentizace se nezastaví, dokud se nezastaví ručně klávesami CTRL+C. Jedná se o velmi nebezpečný krok, deauthentizace znamená, že každou půlsekundu dojde k odpojení AP od sítě. Jinými slovy nepůjde internet na cílovém AP a oběť si tak může všimnout, že něco není v pořádku. Svým způsobem se jedná o DoS útok na AP, nic se k danému AP nepřipojí, pokud příkaz běží. Proces deautentizace je spuštěn tak dlouho, dokud se ve druhém okně terminálu nezachytí WPA handshake. Deauth 0 znamená posílání nekonečně mnoho deauth paketů, deauth 5 pět paketů, 10 deset a podobně. Parametr -c v příkazu aireplay znamená MAC adresu konkrétního zařízení, na které se bude příkaz provádět, čili se nemusí útočit čistě jen na AP, ale i na zařízení k němu připojené. Například sudo aireplay-ng --deauth 0 -a 00:E0:4C:BD:38:08 -c 88:88:76:43:56:R4 wlan0mon:



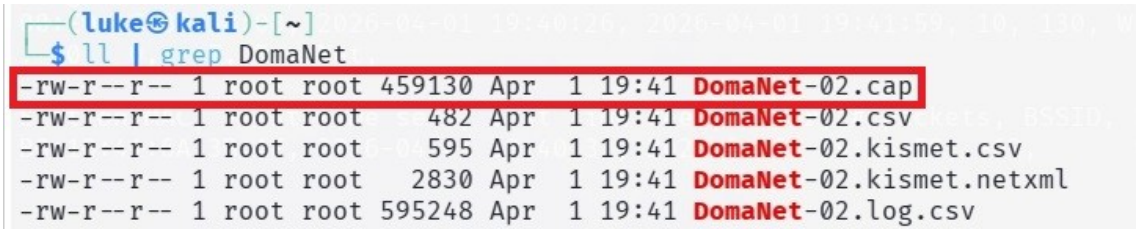
Obr. 69: Aircrack-ng, příkaz aireplay

Zdroj: Vlastní zpracování (2026)



Obr. 70: Aircrack-ng, zachycení handshake

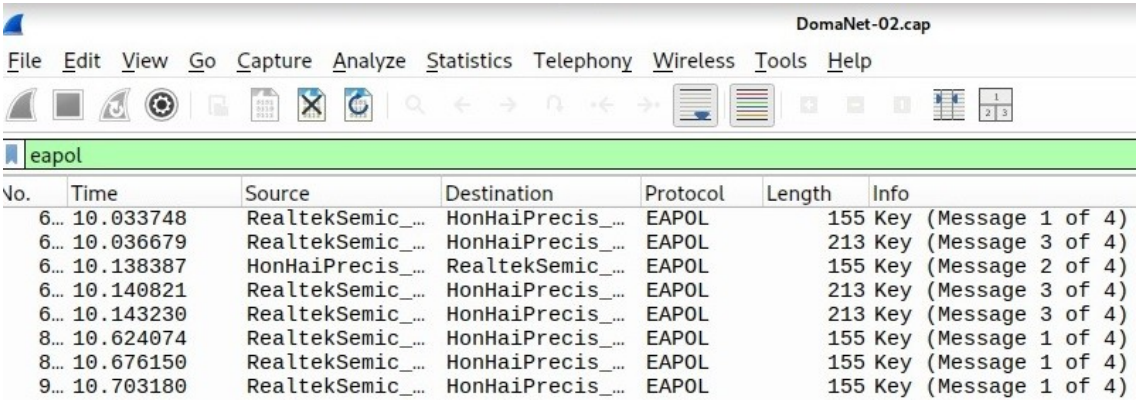
Zdroj: Vlastní zpracování (2026)



Obr. 71: Aircrack-ng, získané soubory

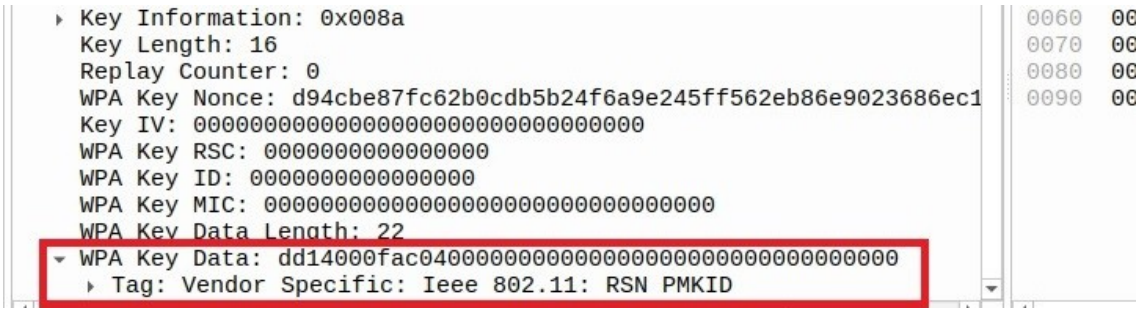
Zdroj: Vlastní zpracování (2026)

Soubor *.cap lze otevřít v programu Wireshark, kde lze vyfiltrovat pakety dle protokolu EAPOL (Extensible Authentication Protocol over LAN). Pakety EAPOL v informační části na konci řádku ukazují Message 1 of 4, Message 2 of 4 a podobně, což označuje four-way handshake (čtyřcestný handshake). Dále při prozkoumání paketu protokolu EAPOL lze spatřit WPA Key Data, data potřebná pro crackování hesla:



Obr. 72: Aircrack-ng, pakety EAPOL ve Wireshark

Zdroj: Vlastní zpracování (2026)



Obr. 73: Aircrack-ng, WPA Key Data ve Wireshark

Zdroj: Vlastní zpracování (2026)

sudo airmon-ng stop wlan0mon – po skončení zachycení handshake je dobré přepnout adaptér zpět do managed mode a příkazem iwconfig ověřit, zda je skutečně v módu managed.

2.5.3 Crackování hesel

Nyní je k dispozici soubor *.cap, který je klíčový pro získání hesla. Nejdříve se bude pokračovat v sadě nástrojů aircrack-ng a heslo se zkusí získat pomocí slovníkového útoku.

gunzip -k rockyou.txt.gz – nejprve se rozbalí wordlist rockyou.txt.gz. Parametr -k zaručuje, že původní zabalený soubor zůstane nedotčený a k němu se vedle vytvoří nový, rockyou.txt.

sudo aircrack-ng DomaNet-02.cap -w /usr/share/wordlists/rockyou.txt – příkaz aircrack-ng spustí slovníkový útok na soubor DomaNet-02.cap pomocí vybraného wordlistu rockyou.txt. Bohužel, slovníkové útoky mají jen malou šanci na úspěch, pokud v onom wordlistu nebude heslo zachycené v souboru. Slovníkový útok probíhá tak, že se z wordlistu vezme například slovo Pavel a aircrack-ng vyzkouší Pavel123, Pavel567 nebo zkouší i znaky P4v3l, P4vel či !Pav3l!. Šance na úspěch je tak mizivá, pokud nebude slovo přímo ve wordlistu. Při pokusu o prolomení hesla se také rapidně zvyšuje výkon CPU (Tutorial: How to Crack WPA/WPA2, 2010):

Obr. 74: Aircrack-ng, heslo nenalezeno

Zdroj: Vlastní zpracování (2026)

Heslo se za pomoci slovníkového útoku nepodařilo nalézt. Na internetu je k dispozici velké množství wordlistů, pokud však slovo není ve wordlistech obsaženo, je téměř nulová šance heslo zjistit.

Nástroj Crunch je již obsažen v základní verzi distribuce Kali Linux a nemusí se tak dodatečně instalovat. Jedná se o jednoduchý, ale silný nástroj, na jehož výstupu lze použít nástroj aircrack-ng i bez slovníkového útoku. Jeho síla spočívá v tom, že na vstupu mu lze dodat vstupní informace, které dokážou zúžit možnosti hesla podobně jako Hashcat. Svým způsobem z Crunche dělá nástroj brute force útoku ve službách aircrack-ng (Tutorial: How to Crack WPA/WPA2, 2010).

crunch 5 9 abcdefghijklmnopqrstuvwxyz1234567890 | aircrack-ng DomaNet-02.cap -b 00:E0:4C:BD:38:08 -w- - základní sestavení příkazu je po vypsání názvu nástroje následující. Minimální délka hesla, maximální délka hesla, jaké alfanumerické znaky se použijí pro brute force útok a poté následuje výstup nástroje aircrack-ng, který je podobný jako při slovníkovém útoku aircrack-ng jen s tou výjimkou, že na konci je ukončen -w-, čili žádný wordlist na vstupu.

crunch 9 9 0123456789 -t 606%%%%%% | aircrack-ng DomaNet-02.cap -b 00:E0:4C:BD:38:08 -w- - protože brute force útoky jsou neskutečně dlouhé, je k tomu zapotřebí výkonné GPU a samotnému CPU trvá ještě déle, možnosti pro prolomení hesla se ještě zúží. Parametr -t přednastavuje první tři čísla hesla:

Obr. 75: Aircrack-ng, heslo nalezeno

Zdroj: Vlastní zpracování (2026)

Při zúžení vyhledávacích parametrů se programu aircrack-ng za pomoci nástroje Crunch podařilo hrubou silou prolomit heslo během sedmi minut. Což znamená, že šestimístné numerické heslo lze prolomit velmi snadno i na starších zařízeních. Posledních 6 čísel bylo z obrázku záměrně vymazáno z důvodu bezpečnosti. Z toho plyne, jak moc je složitost hesel důležitá.

2.6 John the Ripper – prolomení hesla

John the Ripper se používá pro testování síly hesel. Analyzuje a rozpoznává různé formáty hashů hesel a odhaluje slabá hesla pomocí slovníkových, single-crack a brute force útoků. Jedná se o offline open-source nástroj pro audit hesel při testování zabezpečení. Slouží zejména ke kontrole síly hesel, k obnově zapomenutých hesel i k testování hesel různých typů souborů jako ZIP, RAR a podobně. Pracuje tedy s hash soubory, ne přímo s hesly, ale s jejich otisky. Metody, které využívá ke kontrole síly hesel, jsou celkem tři (John the Ripper, 2025):

- **Single-Crack Mode** – využívá informace z uživatelských účtů použitím přihlašovacích údajů či jména. Příklad john --single hash.txt,
- **Wordlist Mode** – spustí slovníkový útok proti hashům tak, že zkouší hesla ze seznamu slov a jejich varianty. Příklad john --wordlist=rockyou.txt hash.txt,
- **Incremental Mode** – alias brute force útok, vyzkouší všechny možné kombinace znaků. Příklad john --incremental hash.txt.

Nástroj John obsahuje spoustu pomocných nástrojů, které dokážou převést různé formáty na cracknutelné hashe, například zip2john (získá hash ze ZIP souboru), rar2john (získá hash z RAR souboru), ssh2john (získá hash z SSH klíče) a podobně. Postup prolomení hesla je následující. Nejprve se získá hash (unshadow passwd shadow > hash.txt), poté se spustí crackování hashe (john hash.txt) a nakonec se zobrazí výsledky (john --show hash.txt). Nástroj John the Ripper je rychlý a má silný systém pravidel pro generování variant hesel (John the Ripper, 2025).

2.6.1 Formáty a detekce hashů

Hash je prostě výraz plný znaků, písmen či čísel, který sám o sobě nedává žádný smysl, ale ve skutečnosti se pod zaheshovaným výrazem skrývá heslo. Hash je prostě kryptograficky zašifrované heslo. Začátek hashe začíná obvykle znakem dolaru, například y, podle toho lze určit o jaký kryptografický typ hashe se jedná. $2y$ je bcrypt, y yescrypt, 6 SHA512 a podobně, ale není podmínkou, některé hashe prefixy nemají. Hashe zaručují a posilují bezpečnost hesel. Konkrétní typy hashů vyžadují konkrétní techniky, proto je tak důležité vědět, o jaký typ hashe se jedná. John podporuje stovky hashových formátů a zároveň je dokáže také automaticky rozpoznat podle prefixů nebo podle struktury. Formát hashe lze zadat i ručně, proto je dobré o nich také něco vědět. Detekce formátu hashů se řídí několika pravidly (John the Ripper, 2025):

- **Automatická detekce** – v případě, že má hash jednoznačný prefix (například $2y$, 6 nebo $K4$), pak lze jednoduše poznat typ hashe,
- **Heuristika podle délky a struktury** – hashe lze poznat podle formátu, délky a struktury. Pokud však některé hashe vypadají podobně (například Raw MD5 a LM DES), John se může splést a určí špatný typ,
- **Ručně zadaný formát** - je možné zadat formát ručně, pouze se zadá příkaz john --format=(typ) (hash.txt) a hash se poté vygeneruje dle daného formátu.

Tab. 8: Příklady některých formátů hashů

Hash	*Prefix*	*Ukázka hashe*
MD5	žádný	5d41402abc4b2a76b9719d911017c592
MD4	žádný	a448017aaf21d8525fc10ae87aa6729d
SHA-1	žádný	a9993e364706816aba3e25717850c26c9cd0d89d
CRC32	žádný	414fa339
bcrypt	$2y$, $2b$, $2a$	$2y$12$eImiTXuWVxfM37uY4JANjQ...
sha512crypt	6	6rounds=5000$abcdefghijklmnop$...
scrypt	$scrypt$ nebo 7	$scrypt$ln=16,r=8,p=1$...
Argon2id	$argon2id$	$argon2id$v=19$m=65536,t=3,p=4$...

Zdroj: vlastní zpracování dle *Microsoft Copilot: váš AI pomocník* (2026)

john --list=formats - vypíše všechny formáty hashů, se kterými John pracuje. Při zadání příkazu bez určení konkrétního formátu hashe John rovněž napoví o typu formátu:

```
┌──(luke㉿kali)-[~]
└─$ john --list=formats
descrypt, bsdicrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,
tripcode, AndroidBackup, adxcrypt, agilekeychain, aix-ssha1, aix-ssha25
aix-ssha512, andOTP, ansible, argon2, as400-des, as400-ssha1, asa-md5,
AxCrypt, AzureAD, BestCrypt, BestCryptVE4, bfegg, Bitcoin, BitLocker,
bitshares, Bitwarden, BKS, Blackberry-ES10, WoWSRP, Blockchain, chap,
416 formats (149 dynamic formats shown as just "dynamic_n" here)
```

Obr. 76: John the Ripper, zkrácený seznam formátů

Zdroj: Vlastní zpracování (2026)

2.6.2 Crackování hashů, módy

Single-Crack mode – dokáže prolomit hashe, které jsou založeny na uživatelských specifických informacích, jako je jméno, uživatelské jméno nebo jiné údaje o uživateli. Například pokud existuje uživatelské jméno alice, John vyzkouší různé kombinace znaků, písmen a čísel na jméno jako alice123, alice2025, alice2, aLicE, Al1c3 a podobně. Single-Crack mode je velice rychlý, avšak ne příliš úspěšný, zvláště pak, pokud je heslo složité nebo nemá nic společného s informacemi o uživateli (Password cracking with John the Ripper on Linux, 2025).

john --single --format=sha512crypt hash.txt – zde je použito jednoduché heslo, tak aby ho Single-Crack mode našel, parametr formát nemusel být vypsán, nástroj by ho identifikoval:

Obr. 77: John the Ripper, Single-Crack mode

Zdroj: Vlastní zpracování (2026)

Wordlist mode – všechny dostupné wordlisty v systému lze najít vypsáním příkazu wordlist do terminálu, nejpoužívanější a nejobsáhlejší je však wordlist rockyou.txt v adresáři /usr/share/wordlists/. John the Ripper má vlastní wordlist passwords.lst v adresáři /usr/share/john/, odkud také bere hesla, pokud se spustí bez určení parametru s výběrem slovníku. Také je možné vytvořit si vlastní wordlist, v podstatě se jedná jen o seznam výrazů. Jeden řádek, jeden výraz (Password cracking with John the Ripper on Linux, 2025).

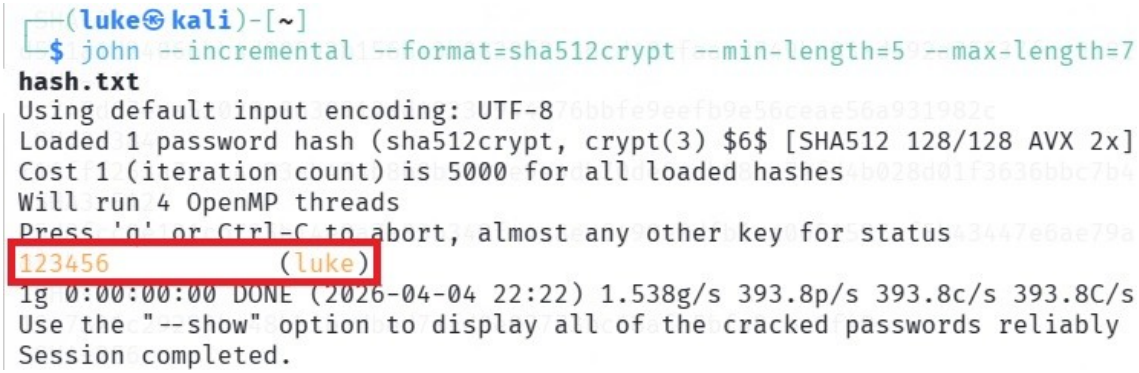
john --format=sha512crypt --wordlist=/usr/share/wordlists/rockyou.txt --max-length=10 hash.txt – heslo je opět velice snadné, aby jej slovník našel:

Obr. 78: John the Ripper, Wordlist mode

Zdroj: Vlastní zpracování (2026)

Incremental mode - vyzkouší každý možný znak a každou možnou variaci znaků. Nejpomalejší, ale nejúčinnější na silná hesla, nejlépe na náhodná nebo neodhadnutelná hesla. Začíná kombinací abc a pokračuje abc123, čím více se limitují možnosti vyhledávání hesla, například minimální a maximální délkou hesla, formátem a tak podobně, proto rychleji se heslo najde. Brute force útok je způsob nalezení hesla naprosto spolehlivý a jistý, ale za cenu dlouhé doby a velké spotřeby zdrojů (Password cracking with John the Ripper on Linux, 2025).

john --incremental --format=sha512crypt --min-length=5 --max-length=7 hash.txt:



Obr. 79: John the Ripper, Incremental mode

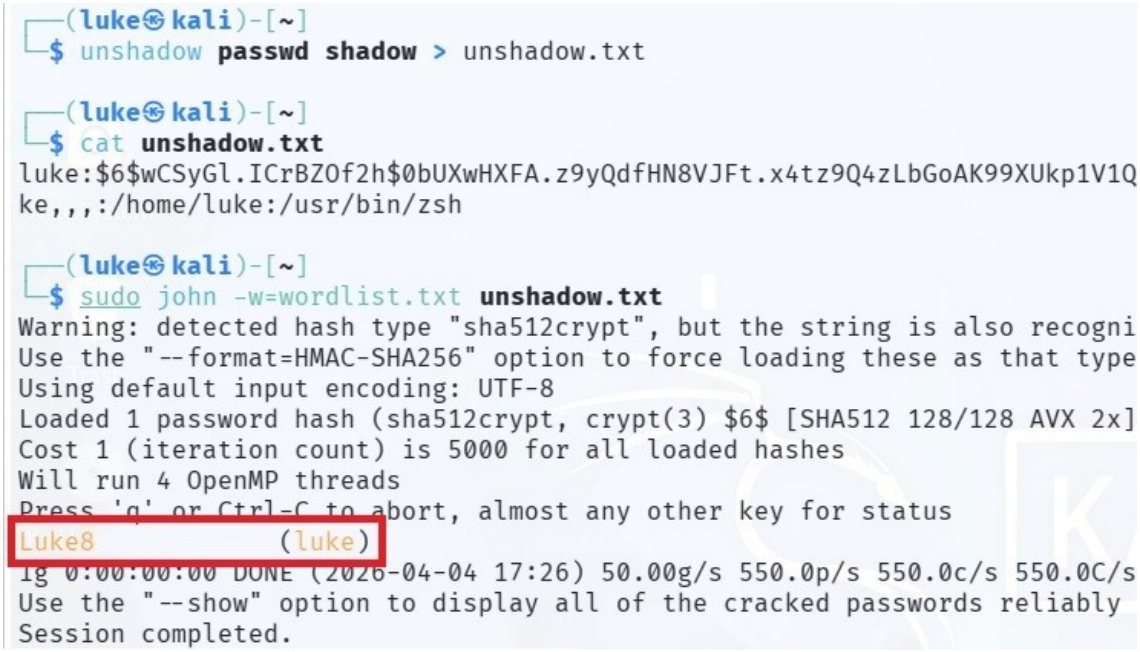
Zdroj: Vlastní zpracování (2026)

2.6.3 Crackování hashů, ostatní

Systémové hashe – v případě obnovení zapomenutého hesla nebo prostě jen ke zjištění zaheshovaného hesla ze souboru /etc/shadow, lze opět využít služeb nástroje John the Ripper. Stačí zkopírovat řádek uživatele, jehož heslo chceme získat, ze souboru /etc/passwd nejlépe do nového souboru se stejným názvem passwd v jiném adresáři, kde bude pouze zmiňovaný řádek. Stejně tak i řádek se zaheshovaným heslem uživatele ze souboru /etc/shadow, opět do nového souboru se stejným názvem shadow. Je lepší dělat kopie souborů. Poté se oba soubory spojí do jednoho příkazem unshadow, což vytvoří soubor čitelný pro Johna (Password cracking with John the Ripper on Linux, 2025).

unshadow passwd shadow > unshadow.txt – spojí oba soubory do jednoho.

john -w=wordlist.txt unshadow.txt – takto získáme systémové heslo ze zaheshované podoby:



Obr. 80: John the Ripper, unshadow heslo

Zdroj: Vlastní zpracování (2026)

ZIP a RAR Hash – nejprve se vytvoří soubor ZIP s heslem, aby se později mohl z hesla vytvořit hash. Pomocí Johna se hash prolomí a zjistí se heslo.

zip --password 123456 shadow.zip shadow.txt – zaheshuje se ZIP soubor.

zip2john shadow.zip > shadow.txt – příkazem zip2john se vytvoří soubor s hashem.

cat shadow.txt – potvrzení, zda se jedná skutečně o hash.

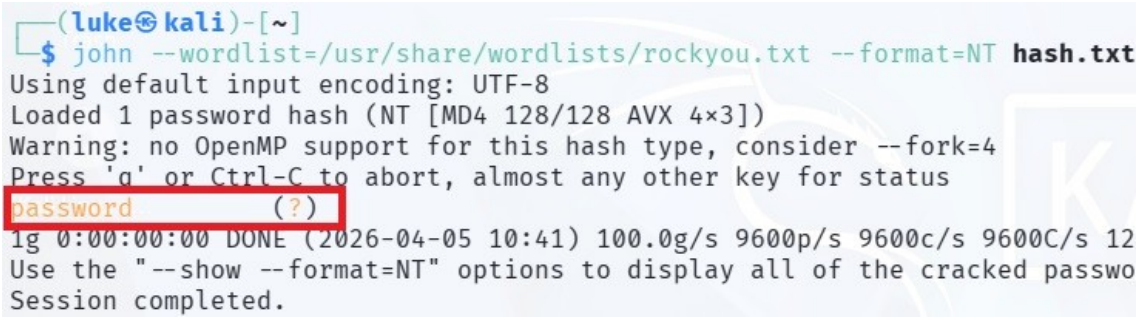
john --wordlist=/usr/share/wordlists/rockyou.txt shadow.txt – zjištění hesla ZIP souboru ze zaheshovaného řádku v souboru shadow.txt. Podobně funguje i zjištění hesla RAR souboru, jen s výjimkou v podobě příkazu rar2john místo zip2john:

Obr. 81: John the Ripper, ZIP hash

Zdroj: Vlastní zpracování (2026)

Windows Hash – John the Ripper dokáže crackovat hashe i z prostředí Windows. Windows ukládá hesla jako NT hash (MD4 otisk hesla v Unicode). Používá se v NTLM autentizaci, při přihlašování ke starším službám. Hashe se ukládají v SAM databázi nebo v NTDS.dit.

john --wordlist=/usr/share/wordlists/rockyou.txt --format=NT hash.txt:



Obr. 82: John the Ripper, NT hash

Zdroj: Vlastní zpracování (2026)

Mask attack – podobně jako nástroje Crunch a Hashcat, i John the Ripper pracuje s mask attack, pokud se částečně zná heslo a nejlépe i počet a typ znaků. K mask attack nelze použít Single-Crack, Incremental ani Wordlist mode, protože se jedná svým způsobem o samostatný mód.

john --format=NT --mask='pas?l?l?l?l' hash.txt – zná se počet znaků, první tři znaky a všechny znaky jsou písmena:

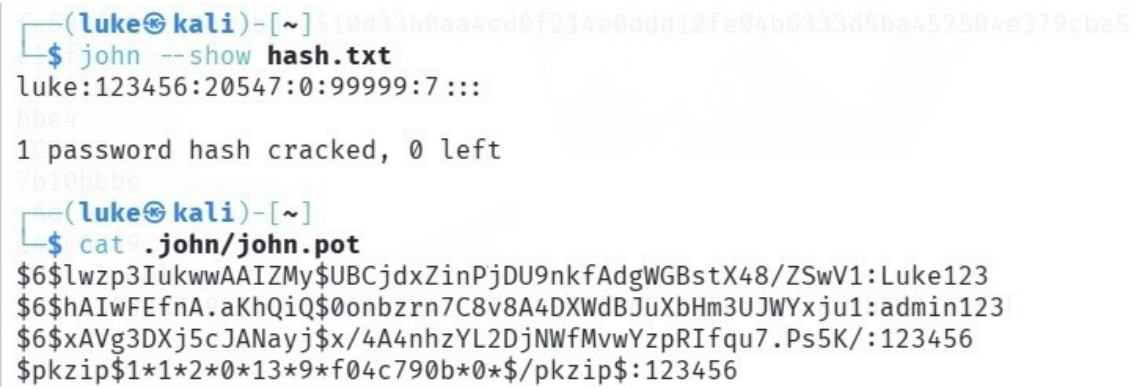
Obr. 83: John the Ripper, Mask attack

Zdroj: Vlastní zpracování (2026)

john.pot – výsledky získaných hesel se ukládají do souboru v adresáři /user/.john, proto John nikdy necrackuje heslo dvakrát a napíše, že heslo již bylo získáno. V případě, že se použije před příkazem sudo, John ukládá výsledky hesel do stejného souboru, ale v adresáři /root/.john.

john --show hash.txt – vypíše již získaná hesla z daného souboru.

cat /home/user/.john/john.pot – vypíše získaná hesla i s jejich hashy:

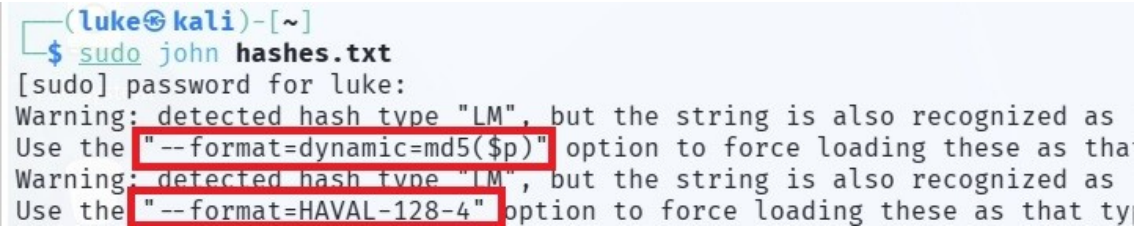


Obr. 84: John the Ripper, soubor john.pot

Zdroj: Vlastní zpracování (2026)

John automaticky identifikuje formát hashe a v případě neurčení typu formátu v příkazu je vypíše na obrazovku, což může být rychlejší metoda, než hledání v seznamu formátů grepem.

john hashes.txt:



Obr. 85: John the Ripper, identifikace hashů

Zdroj: Vlastní zpracování (2026)

Další užitečné příkazy mohou být:

- **hashid (hash)** - příkaz, který identifikuje a vypíše typ hashe,
- **hash-identifier (hash.txt)** - příkaz, který identifikuje a vypíše typ hashe ze souboru.

2.6.4 Nástroj Hashcat

Hashcat je nejspíše nejlepší nástroj na prolomení hesla i hashů. Od doby verze 6.0, kdy přešel na nový hash mode 22000 se již nemusí při pokusech o prolomení zabezpečení Wi-Fi WPA a WPA2 získat handshake tak, že klienta odpojí od sítě tak, jak tomu bylo u nástroje aircrack-ng. Další výhoda je v tom, že se získá pouze jeden soubor se všemi typy handshake a s informacemi o heslu (PMKID a EAPOL protokoly v jednom souboru). Již se nejedná o binární formát, jedná se jen o text, což usnadňuje kopírování, vkládání a podobně. Hashcat také obsahuje nejlepší nástroje pro zachycení WPA handshake ve formátu hash módu 22000. Opět jsou k dispozici tři typy útoků na hesla Dictionary útok, Brute-Force útok a Rule-based útok. Problematika a fungování nástroje Hashcat je však složitější, zde se představí jen základní operace (Cracking WPA/WPA2 with hashcat, 2025).

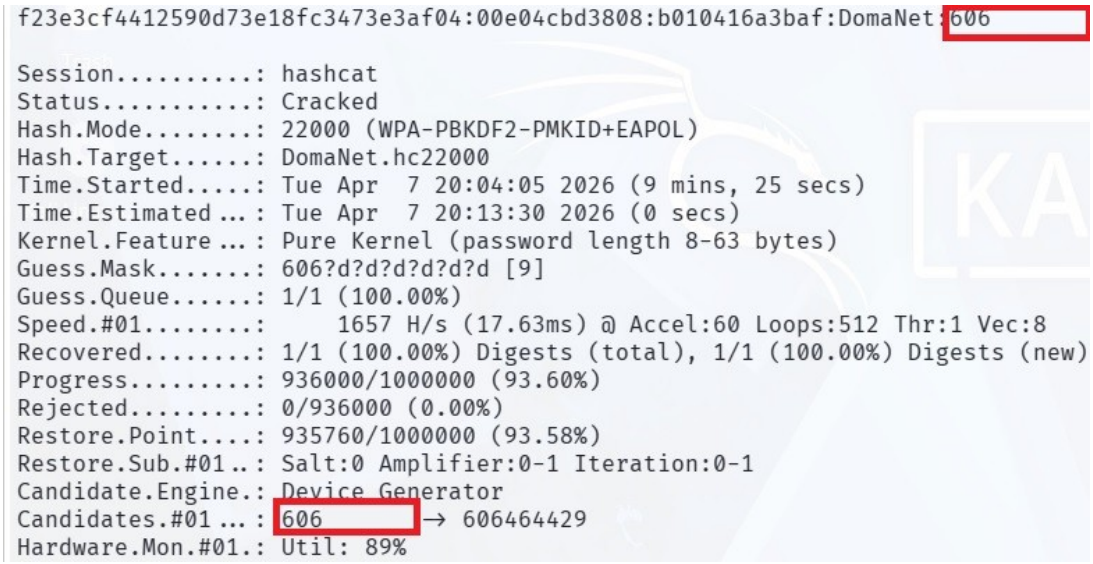
Příprava prostředí je podobná jako u nástroje aircrack-ng. Zapotřebí jsou navíc nástroje hcxdumptool a hcxpcapngtool (hcxstools). Existují dva způsoby, jak je instalovat, klasicky příkazem sudo apt install hcxdumptool a sudo apt install hcxtools (tato verze instalace však způsobuje problémy, protože není aktuální), druhý způsob je instalovat hcxdumptool a hcxtools přímo ze stránek GitHubu (https://github.com/ZerBea/hcxdumptool, https://github.com/ZerBea/hcxtools). Rychlost crackování hesla závisí na výkonnosti GPU a pokud není GPU dostatečně výkonné, tak pomůže CPU, CPU však nebývá tak výkonné jako GPU. Rychlost crackování hesla lze ocenit zejména při Brute-Force útokům. Je dobré také k tomu mít nějaké předpoklady, aby se zúžilo vyhledávání, například starší TP-Link routery používali jen osmi číslicový numericky hesla, což ohromně zúží výběr na 8 náhodných čísel. Následující kroky nainstalují nástroj hcxdumptool (Cracking WPA/WPA2 with hashcat, 2025):

- **git clone https://github.com/ZerBea/hcxdumptool.git** – zkopíruje adresář z GitHub.com na disk do aktuálního adresáře,
- **cd /home/luke/hcxdumptool** – přesune se do potřebného adresáře,
- **sudo apt-get install libcurl4-openssl-dev libssl-dev pkg-config** – nainstaluje závislosti,
- **sudo apt install libpcap-dev** – doinstaluje vývojové knihovny libpcap, potřebné ke kompilaci,
- **make** – instaluje potřebný software,
- **sudo make install** - instaluje potřebný software.

Nyní lze nástroj hcxdumptool používat. Nástroj Hashcat je již v Kali Linuxu předinstalovaný, příkazem sudo apt get install hashcat lze ověřit jeho aktuální verzi. Následující kroky vedou k získaní informací o síťovém provozu:

- **sudo systemctl stop NetworkManager.service** – vypne službu NetworkManager,
- **sudo systemctl stop wpa_supplicant.service** – vypne wpa supplicant,
- **hcxdumptool -i wlan0 -w dumpfile.pcapng --rds=1 -F** – nástroj hcxdumptool definuje rozhraní a uloží všechny získané informace o síťovém provozu do souboru dumpfile.pcapng. Parametr --rds znamená směrování antény a způsob, jakým se sbírají rámce. Parametr -F znamená vynucení zachytávání všech rámců,
- **sudo systemctl start wpa_supplicant.service** – zapne službu wpa supplicant,
- **sudo systemctl start NetworkManager.service** - zapne službu NetworkManager,

- **hcxpcapngtool -o hash.hc22000 -E wordlist dumpfile.pcapng** – převede zachycený provoz do hash formátu 22000. Parametr -E wordlist uloží všechny zachycené názvy SSID do souboru s názvem wordlist,
- **hashcat -m 22000 hash.hc22000 wordlist.txt** - spustí Hashcat slovníkový útok na soubor získaný z WPA provozu za pomoci slovníku wordlist.txt,
- **hcxpcapngtool -o output.hc22000 input.cap** – převod souboru *.cap na soubor *.hc22000,
- **hashcat -m 22000 -a 3 output.hc22000 606?d?d?d?d?d?d** – útok s maskou na číselné heslo o devíti znacích začínající čísly 606, parametr -a 3 znamená masku, parametr -m znamená typ hashe:



Obr. 86: Hashcat, crackování hesla

Zdroj: Vlastní zpracování (2026)

Nástroj Hashcat má opět mnoho možností. Rychlost prolomení šestimístného číselného hesla byla podobná rychlosti prolomení hesla pomocí nástroje Crunch ve spolupráci s aircrack-ng. Největší rozdíl mezi oběma nástroji je, že aircrack-ng se s pomocí deautentizačních útoků dokáže zaměřit na jeden AP, na který útočí, a přitom vytváří několik souborů. Hlavní soubor *.cap je binární, čili není čitelný pro člověka. Nástroj Hashcat vytvoří pouze jeden soubor, textový a čitelný pro člověka, ale na druhou stranu sbírá všechny informace okolo, tedy všechny typy handshake. Nezaměřuje se pouze na jeden AP.

2.7 Burp Suite – skenování webů

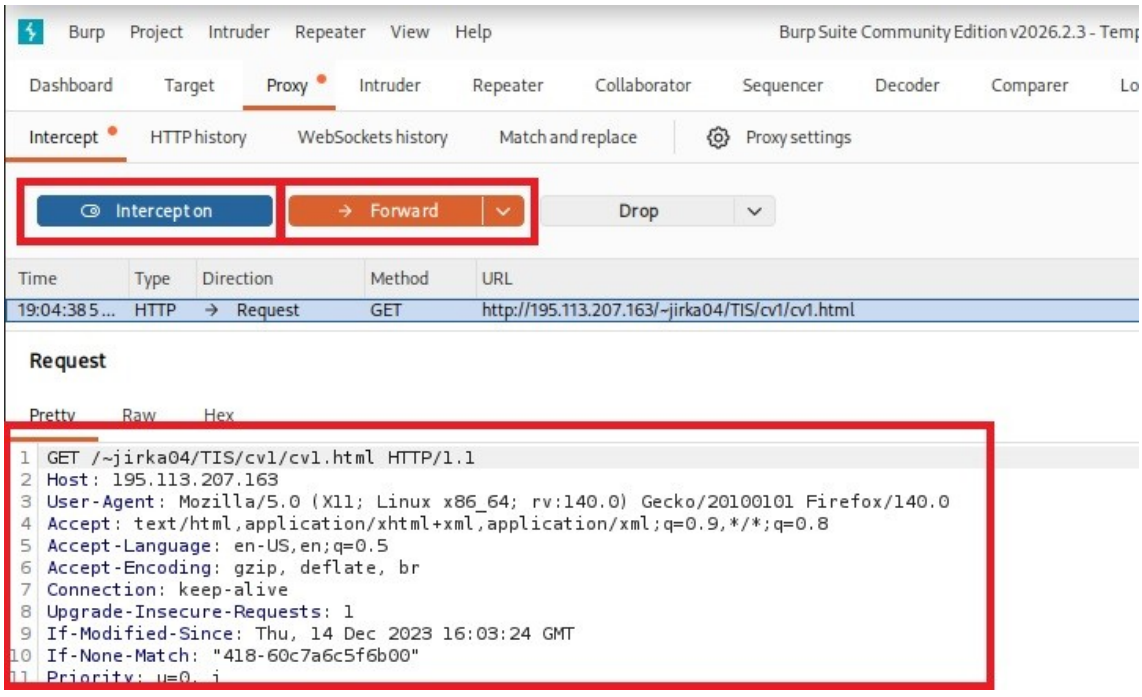
Nástroj Burp Suite slouží zejména jako proxy server mezi klientem a serverem. Když klient odešle požadavek na server, Burp Suite požadavek zachytí, analyzuje a až poté ho přepustí dále na server. Zachycené požadavky lze rovněž editovat, než se pošlou dále. Což se týká všech informací od klienta, jako jsou přihlašovací údaje, formuláře, informace o hostitelském zařízení a podobně. Požadavků může být mnoho. Burp Suite má grafické rozhraní, které se načte vypsáním příkazu burpsuite do terminálu. Burp Suite má placenou Professional verzi, ve které je více možností. Community Edition je zdarma a nabízí o něco méně než ve verzi Professional. Při základním nastavení Burp Suite ve verzi Community Edition je zapotřebí při jeho startu zvolit Temporary project in memory (ukládat na disk jde pouze v placené Professional verzi)

a Use Burp defaults (základní nastavení Burp Suite). Poté se načte hlavní nabídka (What is Burp Suite?, 2024).

V záložce Proxy je nejdříve nutné nastavit rozhraní na kartě Proxy settings. Kdyby nebylo na kartě Proxy settings již nastaveno Proxy listeners interface 127.0.0.1:8080, nastaví se ručně přes tlačítko Add. Zvolí se možnost Specific address nebo Loopback only (libovolná možnost), nastaví se adresa 127.0.0.1, port 8080 a křížkem se okno uzavře. Což znamená, že všechny požadavky z prohlížeče se budou zachytávat na adrese localhost 127.0.0.1 v Burp Suite. Nastavení proxy serveru musí být na adrese 127.0.0.1:8080. V prohlížeči Firefox se najde v Nastavení, Nastavení, v části General a úplně dole Network Settings, Settings je nutné zvolit Manual Proxy Configuration adresa 127.0.0.1, port 8080 a zaškrtnout pole Also use this proxy for HTTPS a potvrdit tlačítkem OK. Pokud se ukončí práce s Burp Suite, nesmí se zapomenout vrátit nastavení ve Firefoxu a zaškrtnout možnost No proxy v Settings, v Network settings. Jinak nebude prohlížeč fungovat. Dále vlevo nahoře je tlačítko Intercept off/on, což znamená v překladu zachytávání vypnuto/zapnuto (What is Burp Suite?, 2024).

2.7.1 Zachycení a analýza HTTP(S) komunikace

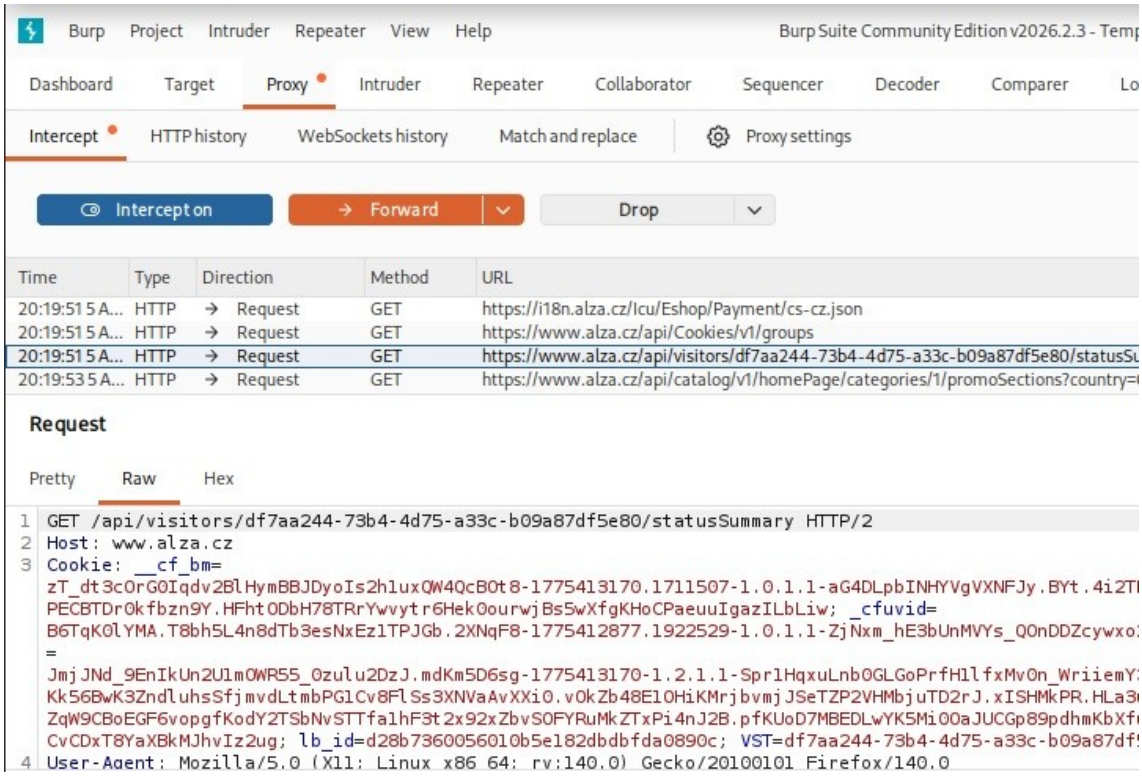
HTTP komunikace – při načítání jakékoli nešifrované HTTP stránky lze zachytit požadavky mezi klientem a serverem. V Burp Suite, v záložce Proxy, když se zapne Intercept on a stránku HTTP opět načteme, lze vidět všechny požadavky ze strany klienta. V prohlížeči klienta se po aktualizaci stránka nenačítá, čeká na povolení ze strany Burp Suite. Obsah stránky lze pochopitelně měnit a až poté kliknout Forward, čímž se povolí přeposlání požadavku na server a klientovi se stránka začne načítat. Stránka se načítá postupně podle toho, jak se potvrzují jednotlivé požadavky tlačítkem Forward, protože těch požadavků je více. Na Forward lze například kliknout i osmkrát, než se celá stránka načte. Jeden klik na Forward, jeden přeposlaný požadavek na server (Burp Suite: The Basics — TryHackMe Walkthrough, 2025):



Obr. 87: Burp Suite, zachycení HTTP komunikace

Zdroj: Vlastní zpracování (2026)

HTTPS komunikace – při používání proxy se nenačítají HTTPS stránky a hlásí MITM útok, proto je nutné nejdříve stáhnout certifikát ze stránek Burp Suite. Tak bude možné zachytávat požadavky i přes HTTPS šifrování. Certifikát je ke stažení vpravo nahoře na stránkách http://burp, CA Certificate. Poté, co se stáhne, se musí vložit do prohlížeče. Ve Firefoxu se najde v Nastavení, Nastavení, Privacy & Security a zde Certificates, poté View Certificates, Import (vybere se certifikát) a zaškrtnou se obě možnosti Trust a potvrdí tlačítkem OK. Nyní je certifikát naimportovaný. Pokud se půjde na nějakou HTTPS stránku a zapne se v Burp Suite Intercept on, vše půjde stejně jako v případě HTTP komunikace (Burp Suite: The Basics — TryHackMe Walkthrough, 2025):



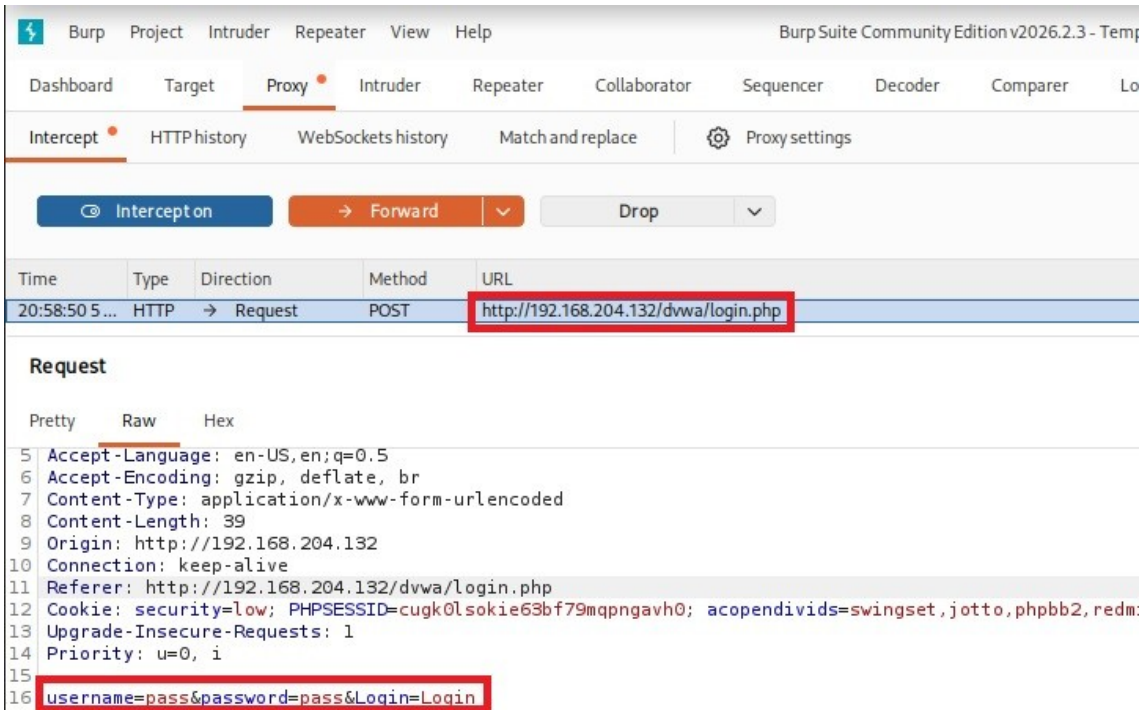
Obr. 88: Burp Suite, zachycení HTTPS komunikace

Zdroj: Vlastní zpracování (2026)

2.7.2 Modifikace požadavků

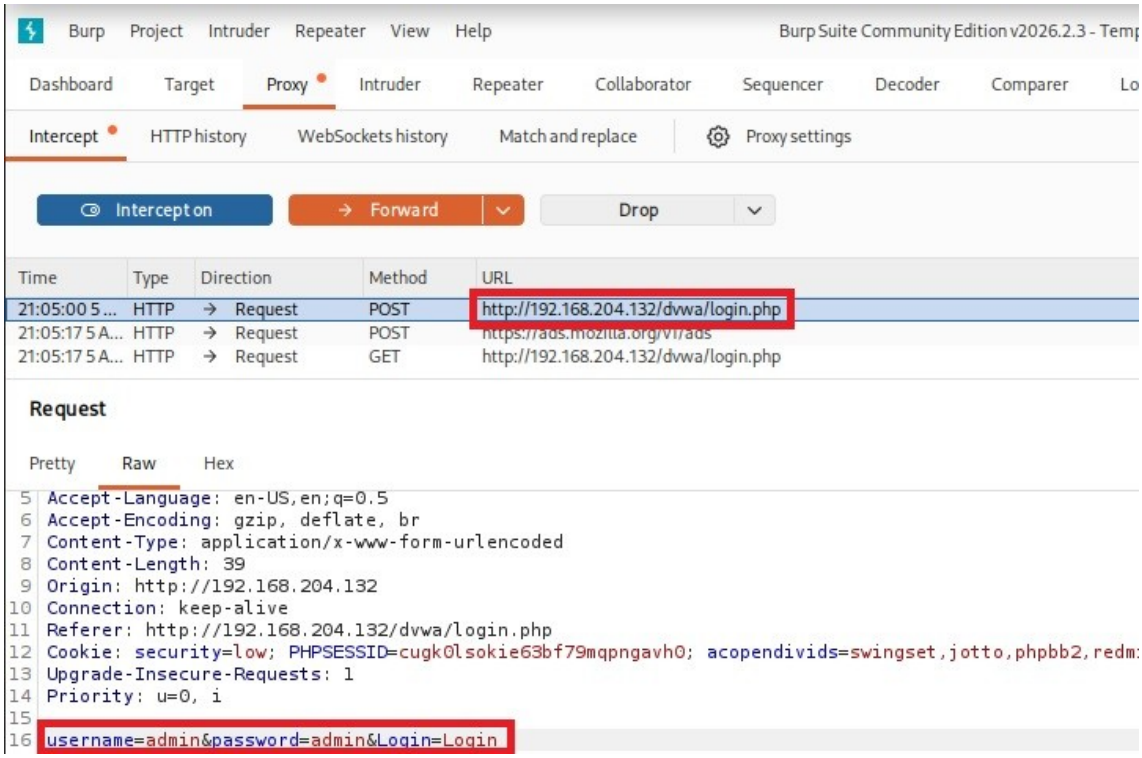
Ke změně obsahu požadavků, pro demonstrativní účely, bude zapotřebí OWASP Broken Web, který legálně umožňuje vyzkoušet nástroje pro penetrační testování. V OWASP Broken Web se klikne na Damn Vulnerable Web Application (DVWA), kde jsou přihlašovací údaje admin a admin. Ještě dříve, než dojde k přihlášení, se zapne v Burp Suite Intercept on a poté se přihlásí do DVWA špatnými údaji, třeba Pass a Pass. Protože se jedná o HTTP komunikaci, přihlašovací údaje lze okamžitě spatřit jako plain text v požadavku metody POST v Burp Suite. Zároveň se ještě na DVWA nenačetla stránka po přihlášení, neustále se čeká na potvrzení požadavku v Burp Suite. Což umožňuje požadavek editovat před potvrzením tlačítkem Forward. V tuto chvíli se nabízí ještě možnost použít Drop, neboli zahodit požadavek. Pokud se řádky s přihlašovacími údaji v Burp Suite nepřepíší, přihlášení se nepodaří. Naopak pokud se opět přihlásí se špatnými přihlašovacími údaji a řádky požadavku se přepíší na správné,

požadavek se odešle a přihlášení se podaří. Takto lze libovolně editovat jakékoli požadavky (Burp Suite: The Basics — TryHackMe Walkthrough, 2025):



Obr. 89: Burp Suite, potvrzení špatných údajů

Zdroj: Vlastní zpracování (2026)

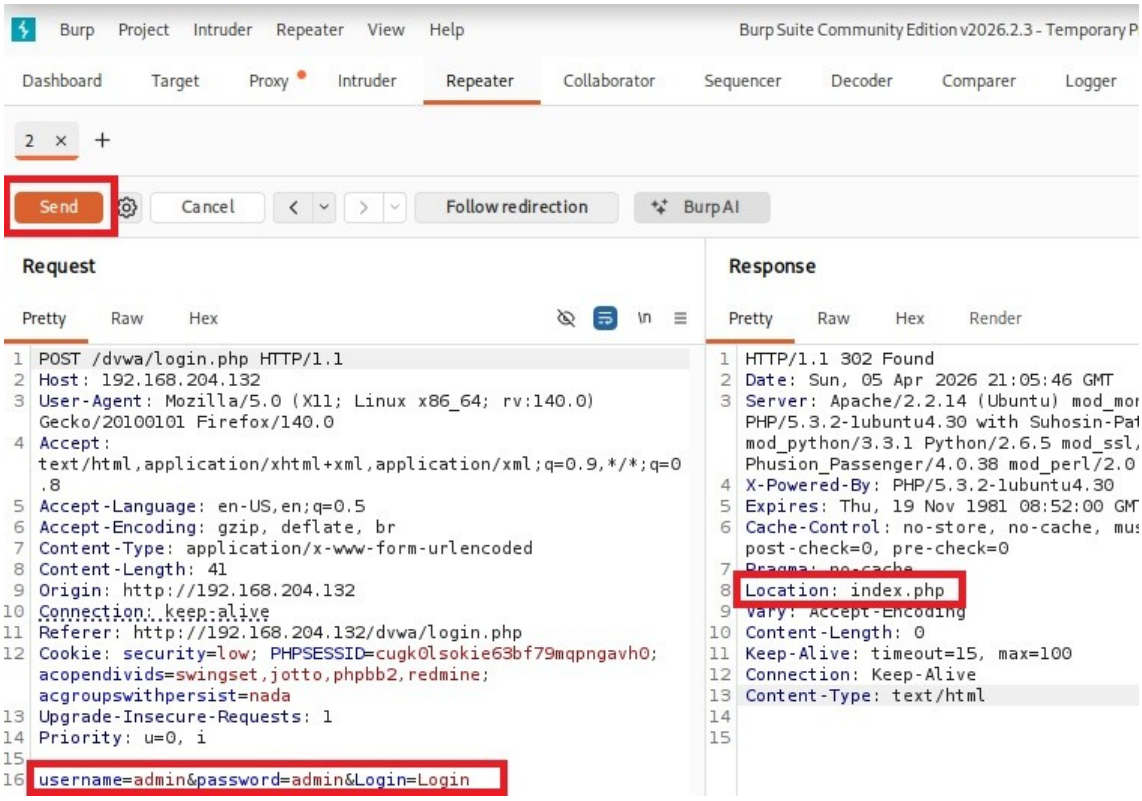


Obr. 90: Burp Suite, přepsání správných údajů

Zdroj: Vlastní zpracování (2026)

2.7.3 Request a response

Další užitečnou záložkou v neplacené verzi je záložka Repeater. V Repeateru jsou dvě okna Request (požadavek) a Response (odpověď). Pokud se pošle požadavek do Repeateru, lze v něm rovnou odhalit, jaká se vrátí odpověď ze serveru. V OWASP Broken Webu se přihlásí na Damn Vulnerable Web Application (DVWA) se špatnými přihlašovacími údaji a díky záložce Proxy a možnosti Intercept on se zachytí požadavek, na který se klikne pravým tlačítkem myši a vybere se možnost Send to Repeater. V záložce Repeater lze poté dát možnost vlevo nahoře na Send a Burp Suite ukáže odpověď serveru. Což je velmi užitečné pro editaci požadavků, hned lze spatřit, jaká bude odpověď. Na příkladu špatných přihlašovacích údajů při přihlášení server odešle odpověď v podobě odkazu Location: login.php a hned lze pochopit, že přihlašovací údaje jsou špatné. Se správnými přihlašovacími údaji přijde v odpovědi Location: index.php (Burp Suite: The Basics — TryHackMe Walkthrough, 2025):



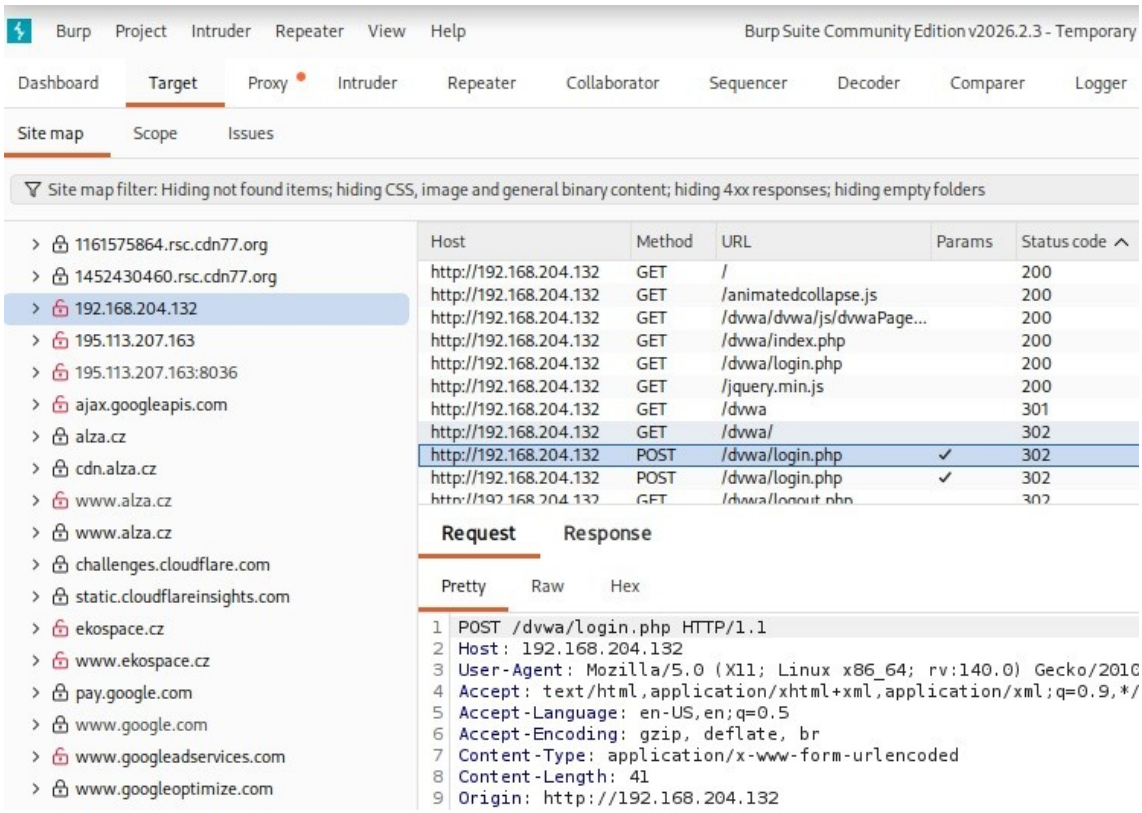
Obr. 91: Burp Suite, request a response

Zdroj: Vlastní zpracování (2026)

2.7.4 Analýza webů

Všechny webové závislosti jdou přes Burp Suite. V záložce Target jsou k vidění všechny webové adresy, které se navštívily a všechny jejich podpůrné soubory, jako jsou obrázky, CSS soubory, URL odkazy, které jsou nezbytné pro načtení webu. Rovněž jsou vidět i stavové kódy stránek, kde 200 znamená v pořádku načtená stránka, 302 je přesměrování, 404 nenalezená stránka, 500 chyba serveru a podobně. Dále jsou k vidění záložky Request (požadavek) a Response (odpověď). V Response lze spatřit kompletní HTML kód stránky, hexadecimální zápis i náhled stránky. Ukazatel typu požadavku, například metoda POST (odeslání formuláře na server) a metoda GET (přijetí odpovědi na požadavek ze serveru). V záložce Target jsou vidět všechny

závislosti, které stránky potřebují pro svůj provoz. Díky záložce Target lze spatřit všechno, co se děje na pozadí, během připojení k určitému webu (Burp Suite: The Basics — TryHackMe Walkthrough, 2025):



Obr. 92: Burp Suite, analýza webů

Zdroj: Vlastní zpracování (2026)

V Burp Suite lze rovněž v záložce Intruder vytvořit slovníkový útok na hesla, bohužel postup působí dost neohrabaně a možností je málo. K útoku na hesla je přece jen lepší použít jiné nástroje. Burp Suite je výborný nástroj na webové analýzy, v mnohém je lepší než OWASP ZAP, avšak pouze díky své placené Professional verzi, která nabízí i skenování webu za účelem odhalení zranitelností. Mezi jeho největší přednosti v neplacené Community Edition patří zachycení a editace požadavků. Neplacená Community Edition nabízí méně možností.

2.8 OWASP ZAP – Cross-Site Scripting

OWASP ZAP (Zed Attack Proxy) je bezplatný open-source nástroj určený pro testování bezpečnosti webových aplikací. Funguje jako proxy server mezi klientem a serverem. Zachycuje provoz automaticky i manuálně a vyhledává zranitelnosti typu SQL injection a XSS. Účel nástroje je stejný jako u Burp Suite, jedná se o dva podobné programy se stejnými nebo podobnými funkcemi. Burp Suite je sice oblíbenější a výkonnější, nicméně verze Professional, která umožňuje skenování webových aplikací, není zdarma. OWASP ZAP je oproti Burp Suite zcela zdarma, včetně možnosti skenování webových aplikací. ZAP automaticky skenuje HTTP požadavky (request) a odpovědi (response) pasivně i aktivně. Aktivní skenování znamená simulaci útoků k nalezení zranitelností. OWASP je mezinárodní otevřená komunita, která se zaměřuje na zvyšování bezpečnosti software. ZAP se musí nejdříve nainstalovat (sudo apt install zaproxy), v některých verzích distribuce Kali Linux je v základním vybavení. Program se

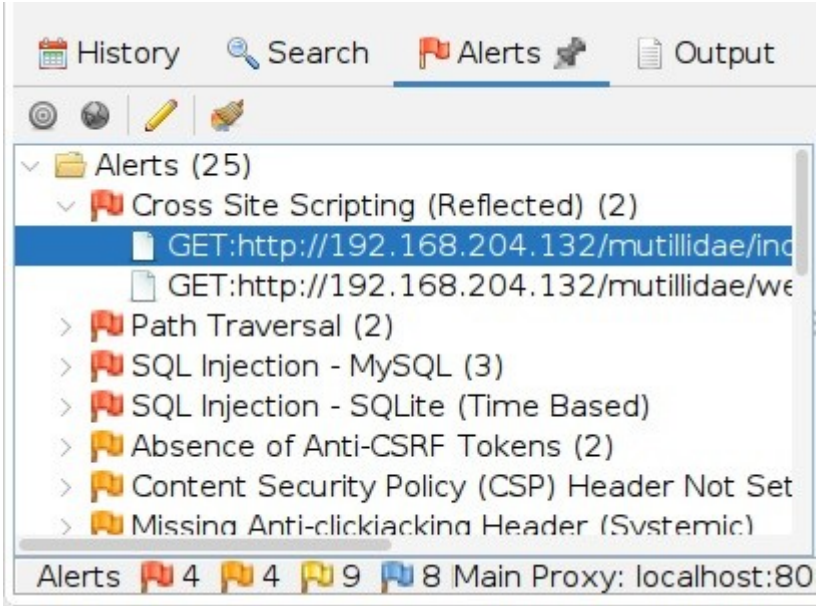
spustí příkazem zaproxy. V úvodní obrazovce se ZAP zeptá, zda se po ukončení programu má práce zachovat. Pokud se bude dělat na nějakém projektu delší dobu, zvolí se ano. Pokud se však ukončí práce bez uložení, zvolí se ne. Čili no, I do not want to persist this session at this moment in time. Na výběr je automatické a manuální skenování. Zranitelnosti typu XSS jsou prováděny v jazyce JavaScript. Důležitá je zde opět znalost jazyka JavaScript (Zed Attack Proxy (ZAP), 2025).

2.8.1 Vyhledávání XSS zranitelnosti

Automatické skenování - nejdříve ZAP provede funkci Crawl, poté se na nalezené prvky spustí aktivní testy a nakonec se zobrazí zjištěné zranitelnosti. Crawl (crawling neboli spidering) projde odkazy a formuláře, objeví dostupné stránky a vstupní body aplikace, zatímco na pozadí běží pasivní sken, který analyzuje provoz webu (HTTP hlavičky, cookies a nastavení). Po skončení crawlingu se zapne aktivní sken, který posílá požadavky na web, aby odhalil zranitelnosti, aktivní proto, že mění data. Aktivní sken poté nahlásí výsledky skenování (Zed Attack Proxy (ZAP), 2025).

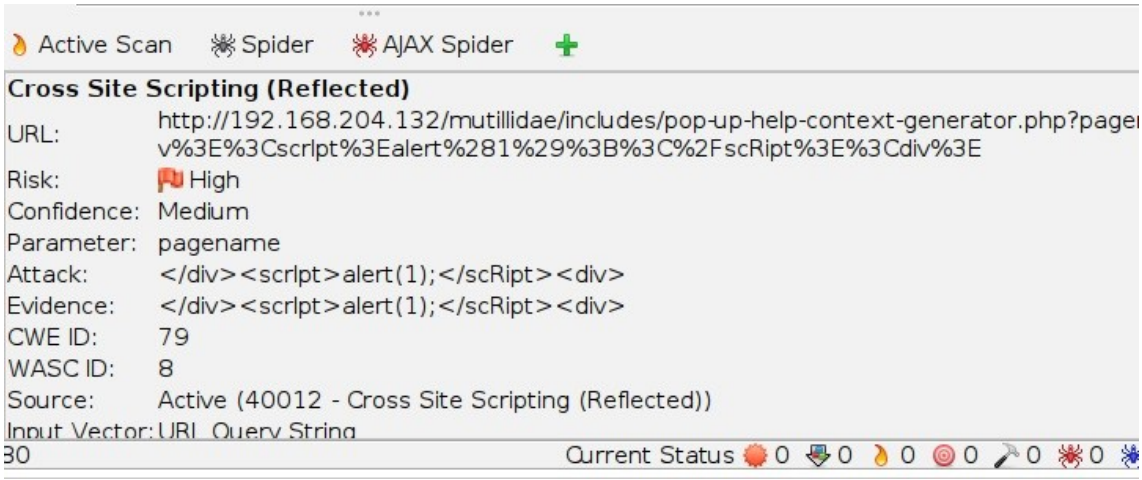
Manuální skenování (Explore) - manuální skenování se provádí ručně, pokud se nechce spoléhat na automatické skenování. Otevře se web v prohlížeči napojeném na ZAP, přes nastavenou proxy (nebo ZAP prohlížeč), podobně jako je tomu u Burp Suite, a zachytávají se všechny požadavky, které lze prohlížet, upravovat a testovat. Na požadavcích lze testovat útoky typu XSS, SQL(i), testovat přihlašovací údaje a spouštět fuzzing na vybrané části nebo odhalovat skryté cesty. Princip je velmi podobný jako u Burp Suite, upraví se požadavek a vyhodnotí odpověď. Na odpovědi lze testovat útoky (Zed Attack Proxy (ZAP), 2025).

Na příkladu byla v OWASP Broken Web nalezena zranitelnost typu XSS pomocí automatického skenování OWASP ZAP:



Obr. 93: OWASP ZAP, vyhledaná zranitelnost XSS

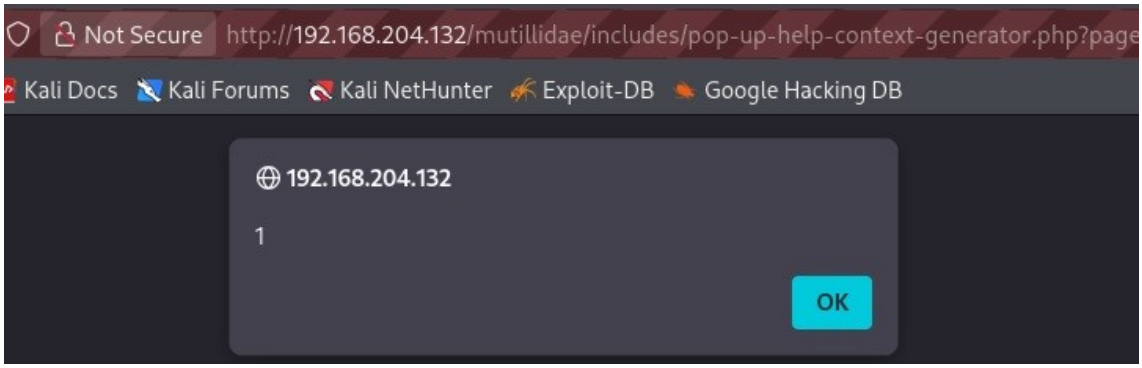
Zdroj: Vlastní zpracování (2026)



Obr. 94: OWASP ZAP, informace o zranitelnosti XSS

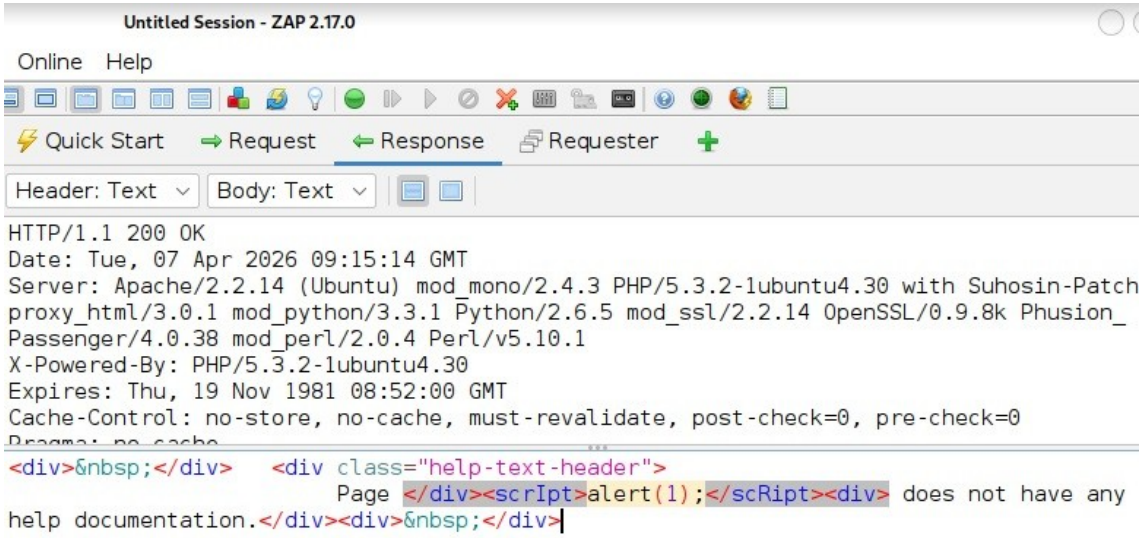
Zdroj: Vlastní zpracování (2026)

Pravým tlačítkem myši lze konkrétní zranitelnost otevřít v prohlížeči:



Obr. 95: OWASP ZAP, zranitelnost XSS na webu

Zdroj: Vlastní zpracování (2026)



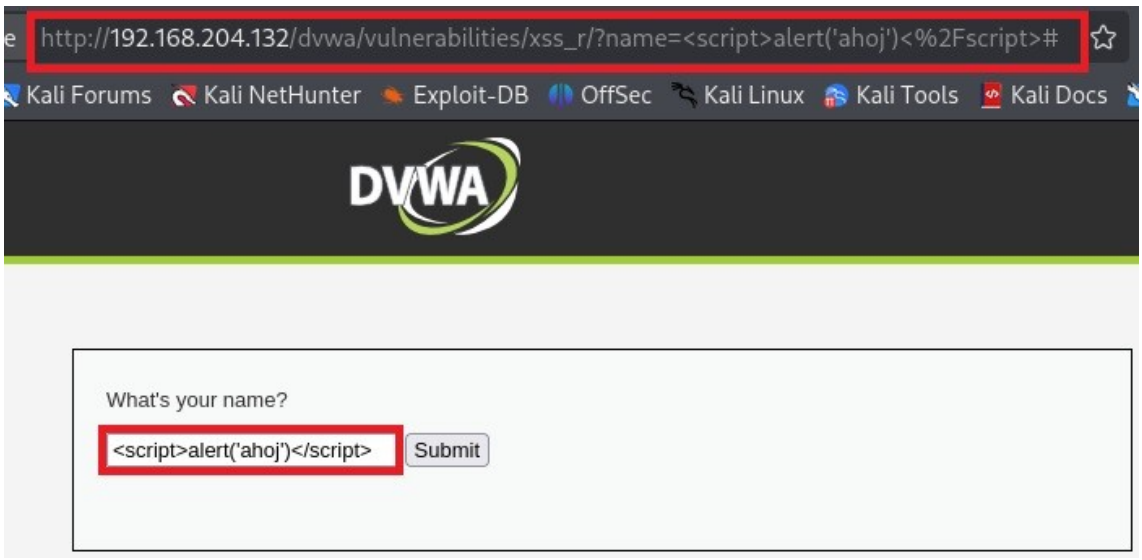
Obr. 96: OWASP ZAP, kód zranitelnosti XSS

Zdroj: Vlastní zpracování (2026)

2.8.2 XSS Reflected/Stored

Pomocí klávesy F12 ve webové aplikaci, ve vývojovém prostředí, lze v záložce Console vložit vlastní JavaScriptový kód, kde si lze vyzkoušet reakci webu na kód. XSS lze rozdělit na dva typy. Jednak na Stored, kde je JavaScript vložený přímo v kódu stránky (při načtení stránky vyskočí JavaScript) a jednak na Reflected, který není uložený v kódu stránky, ale v URL odkazu stránky (XSS Playground, 2025).

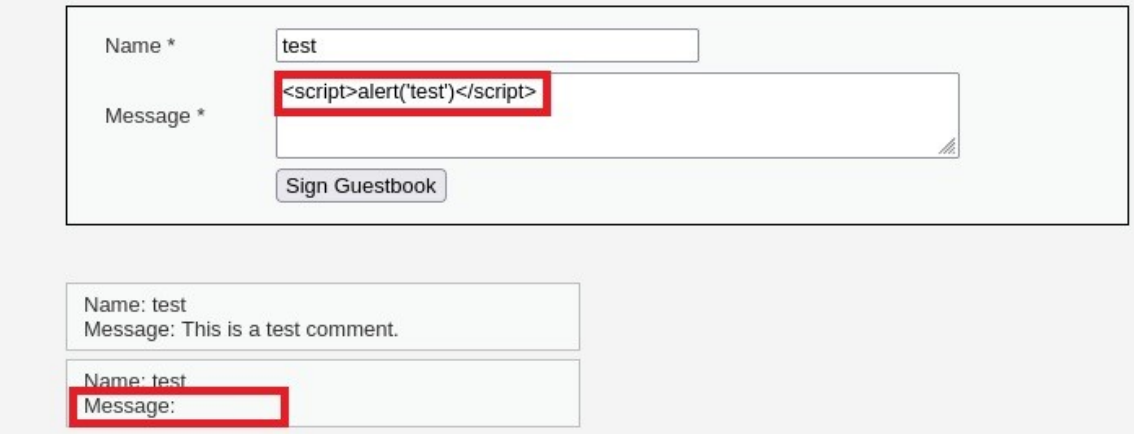
Reflected – opět lze vyzkoušet v OWASP Broken Webu, kde se v DVWA, ve sloupci výběru zvolí možnost XSS Reflected. Zde se napíše JavaScriptový kód (musí být označen HTML značkami <script> a </script>) <script>alert('ahoj')</script>, potvrdí se a kód se přidá za URL stránky. Upravenou URL adresu lze někomu poslat a JavaScriptový kód se u něj spustí. Kód se neukládá do kódu stránky, po aktualizaci stránky JavaScriptový kód zmizí. JavaScriptový kód má obrovské možnosti a záleží na autorech stránek, jak nastaví zabezpečení svého webu:



Obr. 97: OWASP ZAP, JavaScript v URL odkazu

Zdroj: Vlastní zpracování (2026)

Stored – Kód JavaScriptu se ukládá přímo do kódu stránek, typicky přes nějaký formulář. Je uložený přímo v kódu a nedá se odstranit, častými obětmi Stored XSS kódu jsou stránky, kde se něco vkládá, obvykle internetová fóra a podobně. Příklad vložení kódu do formuláře je stejný jako u Reflected. Při každé návštěvě stránek se JavaScript spustí. Formuláře mají obvykle omezený počet znaků, což lze opět vyřešit ve vývojovém prostředí stisknutím klávesy F12 v záložce Inspector, kde se najde správný prvek a zvýší se zde počet maxlength="10" třeba na 100. Pak bude možné vkládat i dlouhé kódy i do jiných kolonek formuláře:

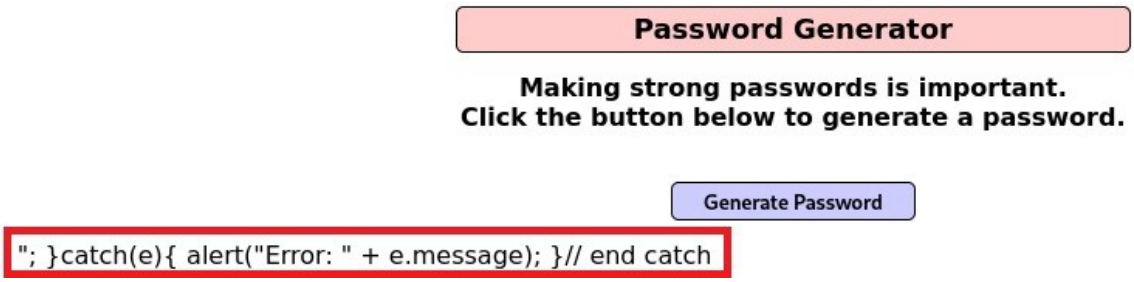


Obr. 98: OWASP ZAP, JavaScript v kódu stránky

Zdroj: Vlastní zpracování (2026)

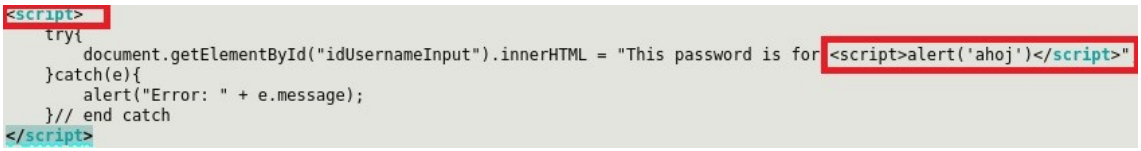
2.8.3 DOM Based XSS

DOM Based XSS – DOM Based Cross-Site Scripting (Document Object Model) znamená vložení JavaScriptového kódu přímo do HTML kódu stránky. Lze opět vyzkoušet v OWASP Broken Web, kde se z úvodní stránky přejde na OWASP Mutillidae II, OWASP 2013, A3 Cross Site Scripting (XSS), DOM Injection, Password Generator. Zde je k dispozici generátor hesel napsaný v JavaScriptu, že je napsaný v JavaScriptu, se pozná podle toho, že když hesla generuje, načítá se pouze prvek generace hesel, nikoliv celá stránka. Stránka nevysílá požadavek na server, aby něco vrátil, aktualizuje se pouze prvek Generate Password. Na stránce není žádný formulář. Zadá se tedy do URL stránky za username=<script>alert('ahoj')</script> a poté, co se požadavek potvrdí, server vrátí "; }catch(e){ alert("Error: " + e.message); }// end catch na konci stránky. Podle těchto prvků lze nalézt ve zdrojovém kódu stránky řádek, kde se nachází vložený JavaScriptový kód. Zde lze vidět, že před ním není řádně ukončen jiný skript, musí se tedy začít ukončením předcházejícího skriptu username=</script><script>alert('ahoj')</script>. Nyní vložený kód funguje (XSS Playground, 2025):



Obr. 99: OWASP ZAP, DOM Based JavaScript rozbitý skript

Zdroj: Vlastní zpracování (2026)



Obr. 100: OWASP ZAP, DOM Based JavaScript kód

Zdroj: Vlastní zpracování (2026)

Příležitostí na procvičení nejen XSS DOM Based ale i ostatních typů XSS je na internetu spousta, nemusí se jednat se o OWASP Broken Web:

Obr. 101: OWASP ZAP, DOM Based JavaScript špatný skript

Zdroj: Vlastní zpracování (2026)

2.9 SQLMap – SQL Injection

SQLMap je open-source nástroj umožňující vyhledat, zda jsou webové stránky zranitelné vůči technice SQL Injection (SQLi) a automatizovat procesy, čímž šetří čas. SQL Injection je metoda útoku na databázový systém, která využívá špatně napsaný kód aplikace, většinou se jedná o aplikaci webovou, například redakční systém nebo e-shop. V praxi SQL Injection znamená vložení škodlivého kódu jazyka SQL nejčastěji do formuláře webové aplikace nebo do vyhledávací lišty prohlížeče. Dalšími možnostmi mohou být cookies (hodnoty z cookies se převezmou do SQL dotazu), hlavičky HTTP (hodnoty se převezmou z HTTP hlavičky do SQL dotazu) nebo API požadavky (zranitelný backend přes API, posílání JSON nebo XML na server). Server převezme dotazy od uživatele a přepošle je na databázový server, který informace potvrdí a přepošle zpět na server. Poté je jako potvrzené informace přepošle uživateli. Nejčastějším cílem SQL Injection jsou právě informace databázového serveru. Do tabulek je tak útočník v případě úspěšného rozbití databáze schopen zasahovat a je možné získat celou databázi uživatelů i s jejich hesly. Jestli je server zranitelný, se často zkouší vložením rovné uvozovky v liště prohlížeče za odkazem po přihlášení na webové stránky nebo do formuláře a pokud se v chybě, která nastane, objeví prvky SQL jazyka či hlášení, které není typické po vložení špatných přihlašovacích údajů (například došlo k neočekávané chybě místo hlášení přihlašovací údaje nejsou platné), je databázový server zranitelný. Znamená to, že webová stránka není ošetřena proti SQL útokům. Samozřejmě, že kromě uvozovek se mohou vložit i další prvky jazyka SQL jako jsou neuzavřené textové řetězce, neuzavřené závorky, neočekávané operátory a různé jiné znaky. Základem je znalost jazyka SQL. Termín Blind SQL Injection se používá pro databázi, která nevyhazuje žádná chybová hlášení. Pokud se do vstupu napíší například uvozovky, nevypíše se žádné chybové hlášení ani prvky SQL jazyka, pouze se aktualizuje stránka. Proto je těžší databázi rozbít. SQLMap odhalí zranitelnosti mnohem rychleji skenem webových stránek, než se snažit ji prozkoumávat ručně. Jedinou obranou proti útokům SQL Injection je programátorsky ošetřit zdrojové kódy webové aplikace a upravit tak vstupní hodnoty. Vstupní hodnoty mohou obsahovat například pouze text, pouze číslo, naopak nebudou obsahovat speciální znaky a podobně. Při ostrém použití SQLMap na webové aplikace se doporučuje při použití nástroje SQLMap použít nástroj ProxyChains, aby správci napadeného serveru neviděli v logování správnou IP adresu útočníka. Pokud se provádí útok

z veřejných bezdrátových sítí, pak není ProxyChains zapotřebí (What is SQL injection (SQLi)?, 2026).

2.9.1 Základní principy SQLi

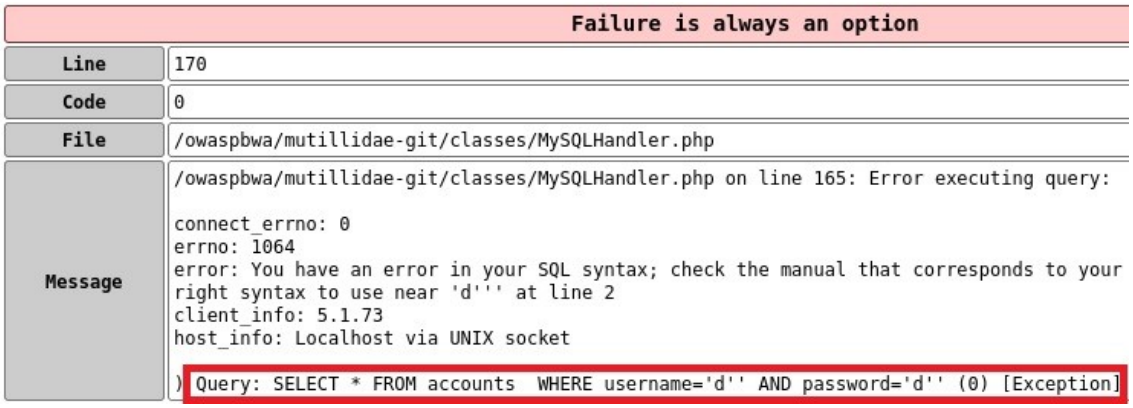
SQL Injection (SQLi) lze například vyzkoušet při přihlášení uživatele do nějakého formuláře. Na pozadí běží komunikace s databázovým serverem, který vyhodnotí, zda jsou přihlašovací údaje správné. Pokud nejsou programátorsky ošetřeny vstupní údaje přihlašování do formuláře, tak tam uživatel může zadávat co chce. Pokud je tedy na vstupu například username i userID a místo zadání pravých informací útočník zadá škodlivé SQL dotazy, databázový server pak na jeho dotazy reaguje (SQL Injection, 2026):

- **Znak '** - pokud se znak ' napíše v přihlašovacím formuláři (na začátku nebo na konci slova) či v odkazu v liště prohlížeče a vyjedou nějaké texty s příkazy jazyka SQL či se vypíše hlášení neočekávaná chyba znamená to, že webová aplikace nebo stránka není chráněna proti SQL Injection a vstupní údaje jsou tak neošetřené, neboli nechráněné. Pokud se naopak stránka nebo aplikace nahraje znova je chráněná. Ukázka dotazu, který vznikne s dvojitými uvozovkami, znamená, že dotaz je nefunkční, server nemůže vrátit hodnoty a databáze je rozbitá:
  http://www.nejakastranka.cz/clanek.php?id=42',
  SELECT * FROM users WHERE email = 'user@email.com' AND password = ' pass'',
- **''OR 1=1--**' - booleovský výraz určí, že podmínka 1=1-- jsou pravda: SELECT * FROM users WHERE email = 'user@email.com' AND password = '' or 1=1--' (znamená, že se získají informace s heslem uživatele user@email.com, logika je převrácená pouze u hesla). Pokud se stane, že obě podmínky jsou pravdivé, vyjede celá tabulka se všemi sloupci, protože jsou splněny obě podmínky.
  SELECT * FROM users WHERE email = ''or 1=1--' AND password = '' or 1=1'.
  SQL dotaz výše vrátí všechny řádky z tabulky, protože obojí je pravda,
- **Další typy znaků či složených znaků mohou být:** -, &, ^, *, ' or ' '-', ' or ' '&', ' or ' '^', '' or '' ''^'', '' or '' ''*'', or true, or 1=1, ' or true-- a další. Možností je samozřejmě více. Představeny zde byli pouze základní možnosti při splnění obou podmínek. Nutná je znalost SQL jazyka.

Pro demonstraci příkladu byl použit open-source program OWASP Broken Web Apps VM 1.2, který je volně ke stažení. Krátký popis instalace je následující. Po extrahování ZIP souboru se extrahovaný soubor s příponou *.vmx připojí přes možnost Open a Virtual Machine ve VMware Workstation, čímž získáme virtuální OS. Jedná se o předpřipravený obraz VMware. Poté si lze ještě dodatečně upravit potřebné nastavení. V textovém režimu se objeví v textu přihlašovací údaje na roota a IP adresa virtuálního OS (you can access the web apps at IP address). Virtuální OS OWASP musí být ve VMware spuštěný a z Kali Linuxu se na něj lze přihlásit vypsáním jeho IP adresy do internetového prohlížeče. OWASP Broken Web Apps VM 1.2 je prostředí určené přímo pro testování webových zranitelností (OWASP Broken Web Applications Project, 2016).

Na úvodní stránce OWASP Broken Web Apps VM 1.2 se klikne na OWASP Mutillidae II, poté v nabídce vlevo OWASP 2013, A1 Injection (SQL), SQLi – Extract Data a User Info (SQL). Zde se simuluje útok na web pomocí SQL Injection bez přihlašovacích údajů a bez jakékoli nápovědy. Zde se může útočník pokusit rozbít databázi. Po pokusu o přihlášení jména s uvozovkou a hesla

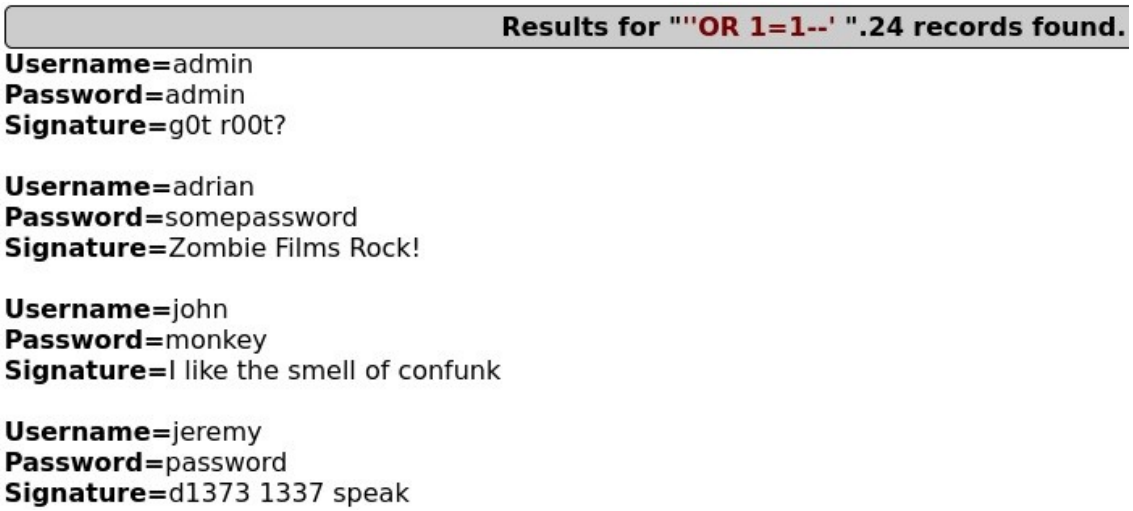
s uvozovkou se objevují chybné příkazy SQL jazyka, zejména řádek Query: SELECT * FROM accounts WHERE username='d'' AND password='d'' (0) [Exception], který naznačuje rozbití databáze (OWASP Broken Web Applications Project, 2016):



Obr. 102: SQLMap, Owasp Broken Web, rozbití databáze

Zdroj: Vlastní zpracování (2026)

Nejen, že se podařilo rozbít databázi, což je vidět na username 'd'' a password 'd'', zjistil se i sloupec databáze accounts. Nyní musí username skončit jako true a password také, aby došlo k úspěšnému přihlášení. Poněvadž username ani password jsou neznámé, lze si pomoct booleovskými příkazy ''OR 1=1--' (může být i d' OR '1'='1, jedná se především o správnost uvozovek) tak, aby obě podmínky skončily jako true a získá se tak kompletní seznam uživatelů i s jejich hesly. Hesla by však správně měla být zaheshovaná, ne jako zde, plain text:



Obr. 103: SQLMap, Owasp Broken Web, získání databáze

Zdroj: Vlastní zpracování (2026)

Obr. 104: SQLMap, Owasp Broken Web, databáze s jedním vstupem

Zdroj: Vlastní zpracování (2026)

2.9.2 Nástroj ProxyChains

Nástroj ProxyChains umožňuje přesměrovat síťovou komunikaci přes různé proxy servery, což je důležité zejména při práci se síťovými nástroji. Poté je obtížné odhalit pravou IP adresu. Správci sítí si totiž mohou lehce zjistit IP adresu počítače útočníka, proto je při použití nástroje SQLMap doporučeno používat ProxyChains. Před příkazem, například SQLMap, se zadá příkaz proxychains a příkaz tak bude mít formu proxychains sqlmap a URL odkaz. Je však nutné vyplnit konfigurační soubor proxychains.conf, protože v základním sestavení konfigurační soubor v adresáři /etc neobsahuje žádné adresy proxy serverů. Seznamy proxy serverů je možné najít na internetu (například http://www.torvpn.com/en/proxy-list nebo https://www.socks-proxy.net/). Postup aktivace nástroje ProxyChains v konfiguračním souboru proxychains.conf je následující (Kali Linux: proxychains a SQL Injection s utilitou sqlmap, 2016):

- **Řádek socks4 127.0.0.1 9050** - zakomentovat, vypne se localhost,
- **Vypsání adres proxy serverů** – pod ProxyList. Jedna adresa, jedna řádka (minimálně 10 adres a časem měnit),
- **Syntaxe adresy** - typ, IP adresa, port, user, password (user a password nevyplňovat),
- V konfiguračním souboru si lze zakomentováním a odkomentováním zvolit jeden ze tří různých modelů chování ProxyChains (odkomentovaný je funkční),
- **Dynamické řetězení (dynamic_chain)** - prochází se seznam proxy serverů, záznam po záznamu, odshora dolů. Při odmítnutí spojení jednoho serveru (reject, timeout nebo jiný) se přejde na další server bez chybového hlášení,
- **Statické řetězení (strict_chain)** - prochází se seznam proxy serverů, záznam po záznamu, odshora dolů. Pokud narazí nástroj na problematický proxy server, řetězení se ukončí,
- **Náhodné řetězení (random_chain)** - seznam náhodně vybraných adres. Možné také zadat hodnotu chain_len (délka řetězu, standardně nastavena 2), počet proxy serverů, přes které je komunikace přesměrovávána,
- **Řádek quiet_mode** - zakomentovat, tichý režim, nevypisují se adresy proxy serverů při přesměrovávání,
- **Řádek proxy_dns** - zakomentovat, přes proxy servery DNS neprojde. Je třeba mít funkční DNS,

- **Proxychains curl ipconfig.me/ip** – test příkazem, zda je ProxyChains funkční.

Při provedení příkazu curl by se měl zobrazit řetěz proxy serverů (řádek quiet_mode musí být zakomentovaný):

```
┌──(luke㉿kali)-[~]
└─$ proxychains curl ipconfig.me/ip
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Random chain  ...  103.36.35.249:5678  ...  timeout
[proxychains] Random chain  ...  103.189.218.158:1080  ...  timeout
[proxychains] Random chain  ...  190.228.33.44:1085  ...  3.33.139.32:80  ...  OK
Moved Permanently. Redirecting to https://ifconfig.me/
```

Obr. 105: ProxyChains, řetězení proxy

Zdroj: Vlastní zpracování (2026)

2.9.3 Identifikace databáze a získávání dat

Pro prolomení databáze, do které proběhlo přihlášení jsou zapotřebí cookies. Při přihlášení do Damn Vulnerable Web Application (DVWA) v OWASP Broken Web Applications Project VM 1.2 se použily přihlašovací údaje admin, admin a cookies lze nalézt v prohlížeči Firefox zde: Nastavení, More Tools, Web Developer Tools, záložka Storage (nebo F12), řádek PHPSESSID a cookies se musí zkopírovat a přepsat do následující podoby, aby se poté daly využít v příkazu SQLMap: PHPSESSID=8mt6k4peklcgvggrsum4dmij10; security=low (cookies lze samozřejmě zkopírovat i přes nástroje Burp Suite či OWASP ZAP) (What is SQL injection (SQLi)?, 2026).
Databáze DVWA se musí nejdříve prolomit zadáním speciálního znaku, ideálně třeba 1', nebo podobného, díky kterému se objeví chybové hlášení. Chybové hlášení dává možnost zkopírovat URL odkaz s chybným vstupem (id=1'). Díky zkopírovanému URL odkazu SQLMap provede analýzu databáze, kterou prolomí díky chybnému vstupu neboli parametru v URL odkazu. Postupně je tak možné zadávat SQLMap příkazy, které poskytnou informace o databázi a jejích datech, kde přepínač -u odkazuje na URL testované stránky, -p na testovaný klíčový parametr (id=1') a protože došlo k přihlášení do systému, musí se příkaz doplnit i o cookies, přepínačem --cookie. Bez přihlášení do systému se samozřejmě cookies vypisovat nemusí. Existují dvě nejčastější HTTP metody, přes které může zranitelná databáze přijímat příkazy, metody GET a POST. Metod je více, SQLMap dokáže pracovat s libovolnou HTTP metodou, například PUT, DELETE, HEAD, OPTIONS, TRACE, pokud se správně nastaví. Metody GET a POST jsou však nejčastější, ale nejsou jediné (What is SQL injection (SQLi)?, 2026):

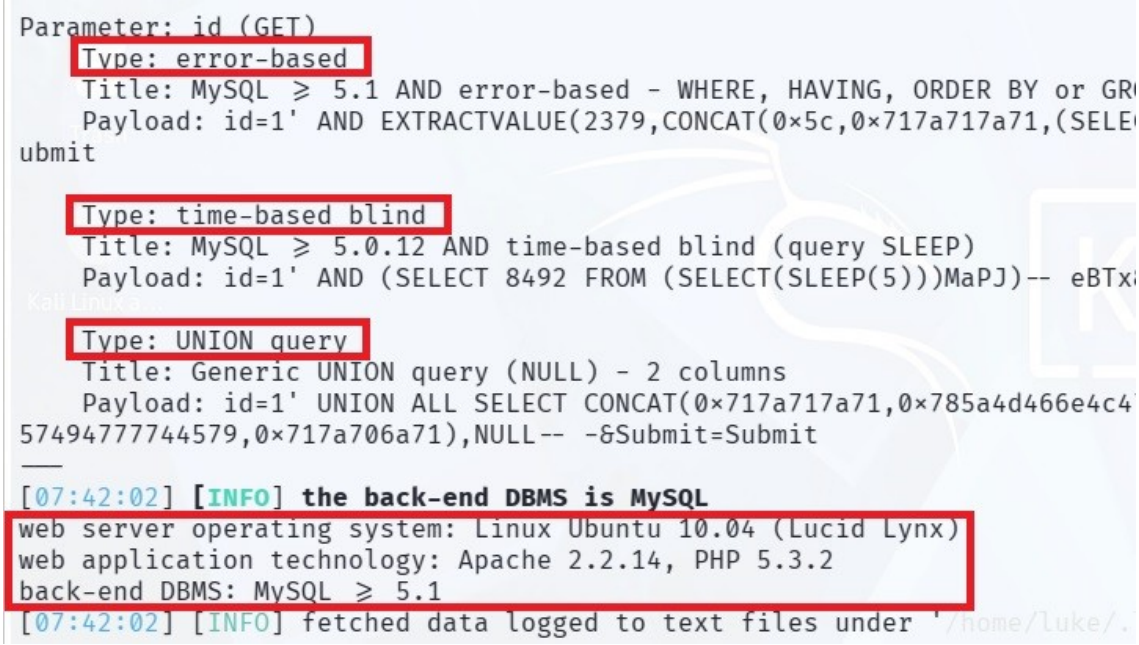
Obr. 106: SQLMap, URL odkaz s parametrem ID

Zdroj: Vlastní zpracování (2026)

2.9.4 Metoda GET

Pro prolomení databáze, ve které proběhlo přihlášení, jsou zapotřebí cookies. Metoda GET spočívá v tom, že parametry odkazu (?id=1%27&Submit=Submit#) jsou přímo v kompletním

URL odkazu (-u "URL"). SQLMap je automaticky vidí v adrese (What is SQL injection (SQLi)?, 2026):
sqlmap -u "http://192.168.204.132/dvwa/vulnerabilities/sqli/?id=1%27&Submit=Submit#" -p id --cookie="PHPSESSID=8mt6k4peklcgvggrsum4dmij10; security=low"

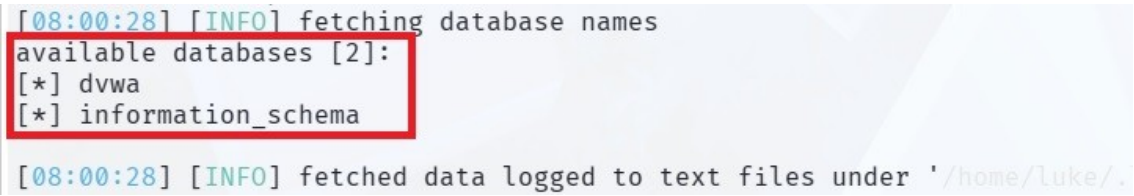


Obr. 107: SQLMap, příkaz metody GET

Zdroj: Vlastní zpracování (2026)

Základním příkazem metodou GET se získají informace o databázi (MySQL 5.1), webové aplikaci (Apache a PHP) a operačním systému (Linux Ubuntu) na kterých databáze běží. Dále se získala informace o tom, že parametr ID (id=1') je zranitelný vůči SQL Injection. Získají se také informace o všech typech SQL Injection (error-based, blind a union) a jejich payloadech.

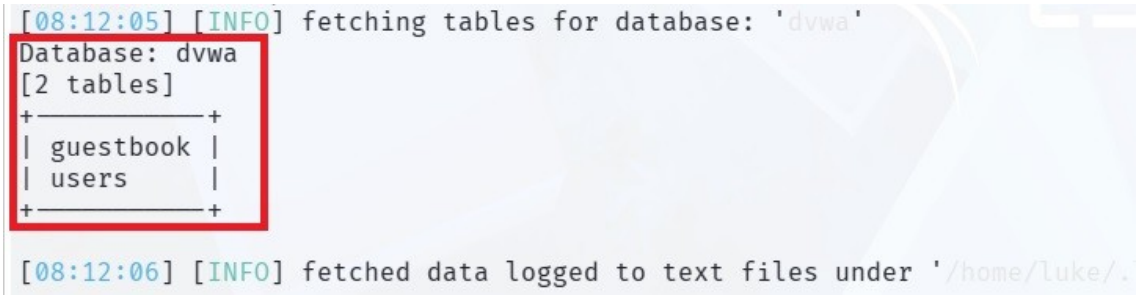
Parametr --dbs - parametr dbs vypíše všechny existující databáze v systému:
sqlmap -u "http://192.168.204.132/dvwa/vulnerabilities/sqli/?id=1%27&Submit=Submit#" -p id --cookie="PHPSESSID=8mt6k4peklcgvggrsum4dmij10; security=low" --dbs



Obr. 108: SQLMap, příkaz metody GET s parametrem dbs

Zdroj: Vlastní zpracování (2026)

Parametry --database, --tables a --columns - parametrem -D lze vybrat konkrétní databázi a parametr --tables vypíše všechny tabulky obsažené v databázi:
sqlmap -u "http://192.168.204.132/dvwa/vulnerabilities/sqli/?id=1%27&Submit=Submit#" -p id --cookie="PHPSESSID=8mt6k4peklcgvggrsum4dmij10; security=low" -D dvwa --tables

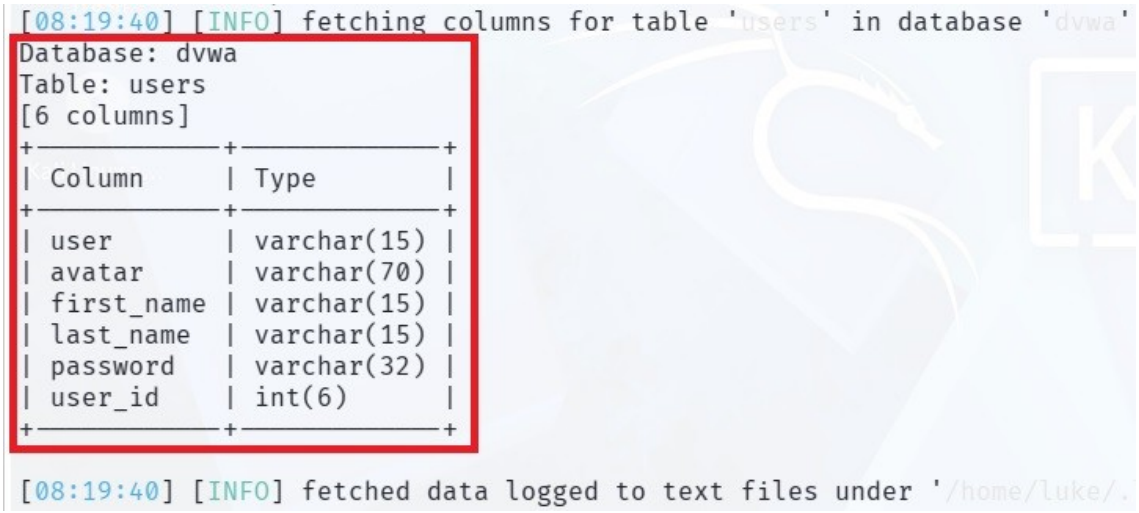


Obr. 109: SQLMap, příkaz metody GET s parametrem tables

Zdroj: Vlastní zpracování (2026)

Databáze má dvě tabulky a každá databáze má sloupce. Sloupce tabulky users se vypíší příkazem --columns:

sqlmap -u "http://192.168.204.132/dvwa/vulnerabilities/sqli/?id=1%27&Submit=Submit#" -p id --cookie="PHPSESSID=8mt6k4peklcgvggrsum4dmij10; security=low" -D dvwa -T users --columns



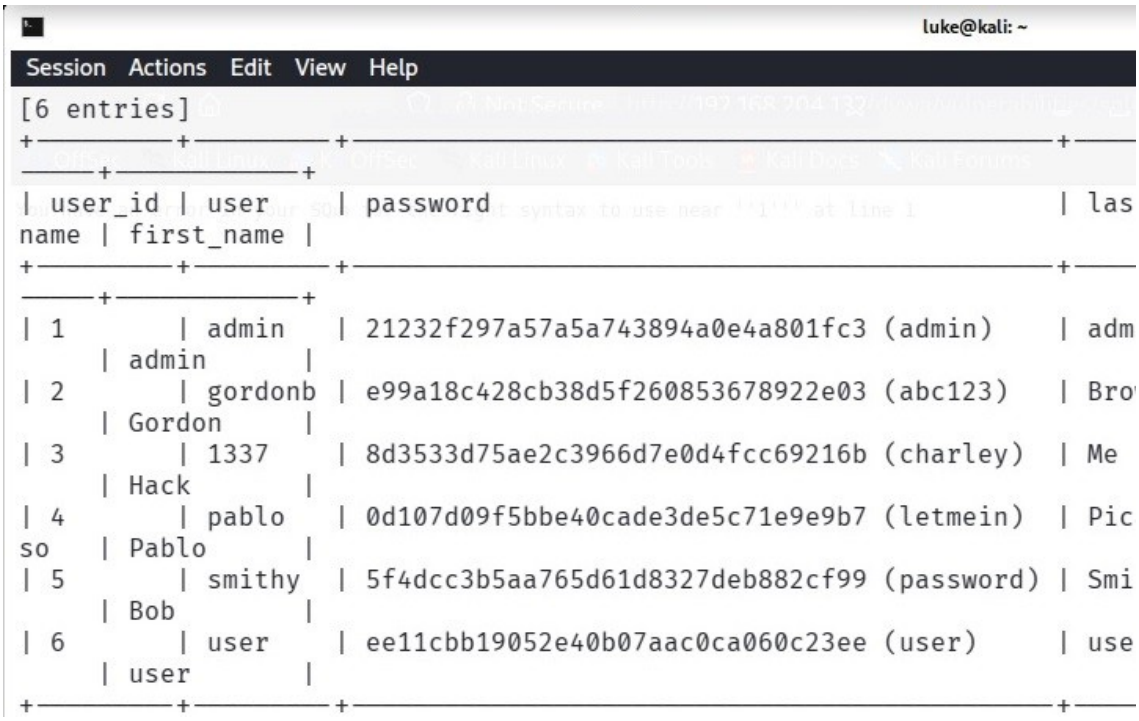
Obr. 110: SQLMap, příkaz metody GET s parametry D,T a Columns

Zdroj: Vlastní zpracování (2026)

Zmiňované parametry vypsaly sloupce tabulky users databáze dvwa. Parametr -D je zkratkou pro --database při výběru konkrétní databáze. Parametr -T je zkratkou pro --tables při výběru konkrétní tabulky. Parametr -C je zkratkou pro --columns při výběru konkrétního sloupce.

Parametr --dump – parametrem dump se kompletně identifikuje celá vybraná tabulka users databáze dvwa se všemi dostupnými informacemi, včetně možnosti slovníkového útoku na zaheshovaná hesla:

sqlmap -u "http://192.168.204.132/dvwa/vulnerabilities/sqli/?id=1%27&Submit=Submit#" -p id --cookie="PHPSESSID=8mt6k4peklcgvggrsum4dmij10; security=low" -D dvwa -T users --column --dump



Obr. 111: SQLMap, příkaz metody GET s parametrem dump

Zdroj: Vlastní zpracování (2026)

Rychlá možnost, jak vše vypsat najednou, lze s parametrem pouze --dump, bez parametrů -D (--database), -T (--tables) a -C (--columns). Lépe je však informace o databázi sbírat přes parametry, pokud se naleznou chyby nebo do akce zasáhnou firewally, které zakazují parametr --dump, pak lze jen těžko zjistit, který krok se přeskočil v celém pořadí kroků.

2.9.5 Metoda POST

Metoda POST (Post Request) spočívá v tom, že parametry odkazu (id=1%27&Submit=Submit#) jsou v těle požadavku (formulář je vidí až po odeslání), SQLMap je tak nevidí automaticky a musí se dodat (--data="param1=hodnota&param2=hodnota" nebo celý požadavek předat přes -r *.txt) (What is SQL injection (SQLi)?, 2026):

Vzory příkazu:
sqlmap -u "URL" --data="" -p id --dump, sqlmap -r *.txt -p id --dump
Základní příkaz:
sqlmap -u "http://192.168.204.132/dvwa/vulnerabilities/sqli/" --data="id=1&Submit=Submit#" -p id --cookie="PHPSESSID=ftpn1art6u15h2b1sasc9kaps2; security=low" --dump
Výsledky i s parametry jsou podobné jako u metody GET, jen je použita metoda POST.

2.9.6 Další parametry a payloady

Nástroj SQLMap nabízí další obrovské možnosti payloadů, parametrů a možností, které se pro přehled ve stručnosti shrnou v konečné podkapitole o nástroji SQLMap. SQL Injection Blind je typ testu SQL Injection, který při vstupu nevrací žádné chyby, žádná chybová hlášení ani neukazuje prvky SQL jazyka při rozbití databáze. Vrací jen správné výstupy, proto je těžší databázi rozbít. Při tomto testu se testuje rozbití databáze naslepo (What is SQL injection (SQLi)?, 2026).

Dostupné payloady:

- Boolean-based blind - testuje rozdíl v chování stránky podle podmínky true nebo false,
- Error-based - zkouší vyvolat chybovou hlášku databáze,
- Inline query - dotaz vložený přímo do jiné části kódu nebo příkazu,
- Stacked queries - testuje, jestli backend dovolí více příkazů v jednom requestu,
- Time-based blind - způsobí, že server odpoví pomaleji, když je podmínka true,
- Union query - zkouší spojit výsledky pomocí Union Select, viditelný výstup,
- Out-of-band (OOB) - používá externí komunikaci (např. DNS) k ověření reakce serveru.

Další parametry:

- --levels - určuje, kolik různých testů SQLMap vyzkouší (level 1 až 5),
- --risks - určuje, jak agresivní testy SQLMap použije (level 1 až 3),
- --banner - zkusí získat verzi databázového serveru,
- --batch - automaticky odpovídá na všechny dotazy bez ptaní,
- --sql-shell - otevře interaktivní SQL konzoli přes SQLMap,
- --os-shell – otevře interaktivní systémovou konzoli,
- --wizard - spustí jednoduchého průvodce pro začátečníky,
- --user - zobrazí uživatele databáze,
- --is-dba - uživatel databáze s administrátorským oprávněním,
- --schema - zobrazí strukturu databáze,
- sqlmap -hh - kompletní nápověda ke všem parametrům.

Dostupné možnosti slovníkového útoku:

- základní slovníkový útok,
- upravený, vlastní slovníkový útok,
- ze souborů se záznamy více slovníků, neboli wordlistů.

Závěr

V prvních kapitolách bakalářské práce byla krátce vysvětlena historie textově orientovaných operačních systémů a pojmů jako open-source a licencování open-source projektů, tak aby si mohl čtenář uvědomit z čeho vzešel operační systém GNU/Linux, se kterým bakalářská práce pracovala. Konkrétně s open-source linuxovou distribucí Kali Linux, z jakých předešlých linuxových distribucí vznikla, za jakým účelem a proč byla vybrána k testování. Ze sady nástrojů distribuce Kali Linux byli vybrány a analyzovány nástroje a k nim byli vybrány vhodné testovací techniky. V praktické části bakalářské práce došlo k vyzkoušení konkrétního nástroje prostřednictvím konkrétní techniky v prostředí distribuce Kali Linux na virtuálním stroji a výsledky zdokumentovány.

Při zkoušení nástrojů v praktické části se ukázalo, že některé vybrané nástroje byli velmi časově náročné. Zejména pak prolamování hesla pomocí nástroje John the Ripper trvalo neskutečně dlouho vzhledem k technickému stavu počítače, na kterém byla technika zkoušena a nástroje podobně zaměřeny jako aircrack-ng a jiné. Nakonec se však podařilo dosáhnout cíle. Dalším problémem bylo pořizování snímků obrazovky během práce s virtuálním strojem. Nakonec byl však problém vyřešen přechodem na konkurenční virtuální platformu VMWare. Vzhledem k neurčitým nedostatečným znalostem autora bakalářské práce trvala práce s nástroji déle, než si je přisvojil. Nakonec však nástroje dostatečně prozkoumal a využil.

Je až s podivem jak jednoduše lze podniknout hackerské útoky na informační systémy pokud jsou k tomu využity správné nástroje. Informační systémy nejsou dostatečně chráněny před útoky ani v dnešní době. Proto je tak moc důležité provádět pravidelné penetrační testování. Většina lidí nezná pokročilé open-source nástroje a neustále dělají stejné chyby jako například slabá hesla, stejná hesla pro více účtů nebo si nedávají pozor na kontrolu odkazů získaných e-mailem. Při psaní práce autor narazil na jednu hojně diskutovanou techniku, která zároveň tvoří více než polovinu potenciálních útoků s velkou úspěšností. Zmiňovanou technikou je sociální inženýrství. Sociální inženýrství patří mezi nejběžnější zranitelnosti. V Kali Linux existují nástroje, které se zmiňovanou technikou zabývají, vzhledem ale k nedostatečnému prostoru bylo rozhodnuto ji nezařadit do bakalářské práce. Sociální inženýrství je tak objemné téma, že je možné se mu věnovat v jiné práci. Za další možné téma k prozkoumání rovněž stojí Metasploit Framework i Hashcat. Bakalářská práce se Metasploit Framework věnovala spíše teoreticky, vzhledem však k jeho rozsahu použití a komplexnosti celého Frameworku by rozhodně bylo lepší se mu věnovat více.

Penetrační testování v distribuci Kali Linux nabízí obecně tolik přitažlivé kouzlo open-source technologií. Kouzlo, že informační technologie nabízí skutečný svobodný a ideologický rozvoj osobnosti, který byl tolik přítomný v 80. letech GNU projektem i jinými a který tak uvadl s příchodem proprietárních, uzavřených operačních systémů v 90. letech.

Seznam použité literatury

10 nejčastějších typů kybernetických útoků. [online]. 2022. Dostupné z: https://www.datasys.cz/10-nejcastejsich-typu-kybernetickych-utoku/

10 typů kybernetických útoků, o kterých byste měli vědět v roce 2022 a rady, jak jim zabránit. [online]. 2022. Dostupné z: https://cz.linkedin.com/pulse/10-typ%C5%AF-kybernetick%C3%BDch-%C3%BAtok%C5%AF-o-kter%C3%BDch-byste-m%C4%9Bli-v%C4%9Bd%C4%9Bt-

18 Best Kali Linux Tools and How to Use Them. [online]. 2025. Dostupné z: https://www.simplilearn.com/top-kali-linux-tools-article

25 Best Kali Linux Tools. [online]. 2023. Dostupné z: https://phoenixnap.com/kb/kali-linux-tools

25 Top Penetration Testing Tools for Kali Linux in 2025. [online]. 2025. Dostupné z: https://www.stationx.net/penetration-testing-tools-for-kali-linux

A step-by-step guide to the Metasploit Framework. [online]. 2024. Dostupné z: https://www.hackthebox.com/blog/metasploit-tutorial#mcetoc_1iaar6o1j7f3

Bettercap. [online]. 2026. Dostupné z: https://www.bettercap.org

Bezdrátová síť. [online]. [2025]. Dostupné z: https://cs.wikipedia.org/wiki/Bezdr%C3%A1tovv%C3%A1_s%C3%AD%C5%A5

BORŮVKOVÁ, Jana, Stanislava DVOŘÁKOVÁ a Hana VOJÁČKOVÁ. Jak psát práce na VŠPJ:Typografická pravidla pro studenty VŠPJ. Jihlava: Vysoká škola polytechnická Jihlava, 2021. ISBN 978-80-88064-54-1

Burp Suite: The Basics — TryHackMe Walkthrough. [online]. [cit. 2024-09-19] https://iritt.medium.com/burp-suite-the-basics-d9d838544547

Citace.com. [online]. [cit. 2020-12-21]. Dostupné z: citace.com

Co je open source: kompletní průvodce s historií, licencemi a využitím. [online]. [cit. 2025-09-05]. Dostupné z: https://informatecdigital.com/cs/co-je-open-source/

Cracking WPA/WPA2 with hashcat. [online]. 2025. Dostupné z: https://hashcat.net/wiki/doku.php?id=cracking_wpawpa2

Hacker. [online]. 2025. Dostupné z: https://www.eset.com/cz/hacker

Hackeři – proti komu se bráníme?. [online]. 2021. Dostupné z: https://www.digitalnisebeobrana.cz/hackeri-proti-komu-se-branime

Historie operačních systémů. [online]. 2012. Dostupné z: https://kvint.webnode.cz/historie/historie-operacnich-systemu/

Historie operačního systému GNU/Linux. [online]. 2025. Dostupné z: https://www.root.cz/texty/historie-operacniho-systemu-gnulinux/

How to Use Aircrack-ng: A Guide to Network Compromise. [online]. [cit. 2025-12-17]. Dostupné z: https://www.stationx.net/how-to-use-aircrack-ng-tutorial/

HTTP Strict Transport Security (HSTS). [online]. 2026. Dostupné z: https://hstspreload.org/

Index vmwareworkstationarchive/17.x/. [online]. 2026. Dostupné z: https://archive.org/download/vmwareworkstationarchive/17.x

John the Ripper. [online]. [cit. 2025-08-25]. Dostupné z: https://www.geeksforgeeks.org/linux-unix/how-to-use-john-the-ripper-in-kali-linux/

Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution. [online]. 2025. Dostupné z: https://www.kali.org/

Kali Linux: proxychains a SQL Injection s utilitou sqlmap. [online]. [cit. 2016-11-28]. Dostupné z: https://www.root.cz/clanky/kali-linux-proxychains-a-sql-injection-s-utilitou-sqlmap

Kali Linux Tools. [online]. 2025. Dostupné z: https://www.geeksforgeeks.org/linux-unix/kali-linux-tools

Metasploit Framework. [online]. 2025. Dostupné z: https://docs.rapid7.com/metasploit/msf-overview/

Microsoft Copilot: váš AI pomocník. [online]. 2025. Dostupné z: https://copilot.microsoft.com/

Nejčastější typy kybernetických útoků. [online]. 2023. Dostupné z: https://kybez.cz/najcastejsie-typy-kybernetickych-utokov/

Nmap.org. [online]. [2026]. Dostupné z: https://nmap.org/

Nmap Script Engine (NSE). [online]. [cit. 2025-08-29]. Dostupné z: https://www.geeksforgeeks.org/linux-unix/how-to-use-nmap-script-engine-nse-scripts-in-linux/

Nová citační norma ČSN ISO 690:2011 – Bibliografické citace. [online]. Dostupné z: https://www.iso690.zcu.cz

OWASP Broken Web Applications Project. [online]. [cit. 2016-09-29]. Dostupné z: https://sourceforge.net/projects/owaspbwa/

Password cracking with John the Ripper on Linux. [cit. 2025-09-22]. Dostupné z: https://linuxconfig.org/password-cracking-with-john-the-ripper-on-linux

Penetration Testing Methodology. [online]. 2025. Dostupné z: https://deepstrike.io/blog/penetration-testing-methodology

Penetration Testing Tutorial: What is PenTest?. [online]. [cit. 2024-06-17]. Dostupné z: https://www.guru99.com/learn-penetration-testing.html

Penetrační testování. [online]. 2024. Dostupné z: https://www.etnetera.cz/penetracni-testovani

Penetrační testování. [online]. 2025. Dostupné z: https://bezpecneict.cz/sluzby/penetracni-testy

Penetrační testy infrastruktury sítě, webových aplikací a internetových služeb. [online]. 2025. Dostupné z: https://tns.cz/penetracni-testovani

Port Scanning (Skenování portů). [online]. 2025. Dostupné z: https://blog.jirivanek.eu/cs/port-scanning-skenovani-portu/

SQL Injection. [online]. 2026. Dostupné z: https://www.w3schools.com/sql/sql_injection.asp

Top 21 Kali Linux tools and how to use them. [online]. 2025. Dostupné z: https://www.techtarget.com/searchsecurity/tip/Top-Kali-Linux-tools-and-how-to-use-them

Tutorial: How to Crack WPA/WPA2. [online]. [cit. 2010-03-07]. Dostupné z: https://www.aircrack-ng.org/doku.php?id=cracking_wpa

Typy operačních systémů a jejich úplná historie. [online]. [cit. 2021-03-17]. Dostupné z: https://www.redeweb.com/cs/sou%C4%8Dasnost%2C-d%C3%A1rek/OS/

Typy útoků na Wi-Fi. [online]. 2023. Dostupné z: https://sapsan-sklep.pl/cs/blogs/clanky/typy-utoku-na-wi-fi

VMware Workstation Pro. [online]. 2026. Dostupné z: https://www.techspot.com/downloads/189-vmware-workstation-for-windows.html#

What is Burp Suite?. [online]. [cit. 2025-02-21]. Dostupné z: https://www.geeksforgeeks.org/ethical-hacking/what-s-burp-suite/

What is Metasploit?. [online]. 2025. Dostupné z: https://www.upguard.com/blog/metasploit

What is Open Source Software (OSS)?. [online]. [cit. 2024-07-29]. Dostupné z: https://github.com/resources/articles/what-is-open-source-software

What is SQL injection (SQLi)?. [online]. 2026. Dostupné z: https://portswigger.net/web-security/sql-injection#what-is-sql-injection-sqli

Wireshark Basics. [online]. 2023. Dostupné z: https://support.inductiveautomation.com/hc/en-us/articles/6346064943373-Wireshark-Basics

Wireshark User's Guide. [online]. 2026. Dostupné z: https://www.wireshark.org/docs/wsug_html_chunked/index.html

XSS Playground. [online]. 2026. Dostupné z: https://labs.hackxpert.com/XSS_Playground/HTMLInjection.php

Začněte skutečně ovládat PC. Odposlouchávejte domácí síť. [online]. [cit. 2014-04-17]. Dostupné z: https://www.zive.cz/clanky/zacnete-skutecne-ovladat-pc-odposlouchavejte-domaci-sit/sc-3-a-173324/default.aspx

Zed Attack Proxy (ZAP). [online]. 2025. Dostupné z: https://www.zaproxy.org/

Zsecurity.org. [online]. 2026. Dostupné z: https://zsecurity.org/