Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Diplomová práce

Etický hacking a ochrana webových aplikací

Bc. Matěj Juričič

© 2024 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Matěj Juričič

Informatika

Název práce

Etický hacking a ochrana webových aplikací

Název anglicky

Ethical hacking and web application protection

Cíle práce

Cílem diplomové práce bude navrhnout testovací scénáře pro útoky na webové aplikace, provést následné porovnání různých způsobů ochrany a doporučit nejefektivnější postupy pro ochranu webových aplikací na základě naměřených výsledků. Práce se také bude zaměřovat na rizika, jimž jsou uživatelé vystaveni, bezpečnost webů a webových aplikací a na volně dostupné nástroje pro etický hacking. Dílčím cílem práce bude aplikace testovacích scénářů a vyhodnocení jejich účinnosti při ochraně proti těmto útokům.

Metodika

Práce bude rozdělena na teoretickou a praktickou část. V teoretické části bude provedena komparace literárních zdrojů a použita deskripční metoda pro podrobné zachycení problematiky, kde budou zahrnuty útoky a nástroje pro jejich provedení, ochranné nástroje a způsoby ochrany. Tyto informace budou následně aplikovány v praktické části. Zdrojem informací budou vědecké, odborné články a publikace, které se zaměřují na dané téma. V praktické části budou použity metody pozorování a měření provedených útoků a způsobů ochrany. Následně bude provedena komparace výsledků a na základě nich budou vybrány nejvhodnější testovací scénáře a postupy pro ochranu webových aplikací.

Doporučený rozsah práce

60 – 80 stran

Klíčová slova

Etický hacking, kybernetická bezpečnost, penetrační testy, rizika, testovací scénář

Doporučené zdroje informací

ALLSOPP, Wil. Advanced Penetration Testing. 2017. Indianapolis, Indiana: John Wiley, 2017. ISBN 978-1-119-36768-0.

HARPER, Dr. Allen, Daniel REGALADO, Ryan LINN a kol. Gray Hat Hacking: The Ethical Hacker's Handbook, Fifth Edition. 2018. United States: McGraw-Hill Education, 2018. ISBN 978-1-26-010842-2.

KOLOUCH, Jan, Pavel BAŠTA a spol. CyberSecurity. 2019. Praha: CZ.NIC, z. s. p. o, 2019. ISBN 978-80-88168-34-8.

NORTON, Alan T. Computer Hacking Beginners Guide: How to Hack Wireless Network, Basic Security and Penetration Testing, Kali Linux, Your First Hack. 2016. San Francisco: Independently published, 2018. ISBN 9781980390978.

SINGH, Glen D. The Ultimate Kali Linux Book: Second Edition. 2019. Birmingham: Packt Publishing, 2022. ISBN 978-1-80181-893-3.

Předběžný termín obhajoby

2023/24 LS – PEF

Vedoucí práce

Ing. Václav Lohr, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 4. 7. 2023

doc. Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 3. 11. 2023

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 18. 03. 2024

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Etický hacking a ochrana webových aplikací" 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 30.3.2023 _____

Poděkování

Rád bych touto cestou poděkoval Ing. Václavu Lohrovi, Ph.D., za odborné vedení práce, věnovaný čas a cenné rady, které mi poskytl k vypracování této práce. Dále velmi děkuji celé své rodině za plnou podporu nejen při zpracovávání práce, ale i během celého studia.

Etický hacking a ochrana webových aplikací

Abstrakt

Tato diplomová práce se zaměřuje na problematiku etického hackingu, bezpečnosti webových aplikací a jejich ochranu před různými formami útoků. Má za cíl navrhnout a aplikovat testovací scénáře pro simulaci útoků na webovou aplikaci, následně porovnat různé metody ochrany a doporučit nejefektivnější postupy založené na naměřených výsledcích. Práce se rovněž zaměřuje na analýzu rizik, jimž jsou uživatelé vystaveni, a bezpečnostní opatření webových aplikací. V teoretické části jsou probírány pojmy, jako je etický hacking, typy hackerů, metody etického hackingu, a bezpečnostní hrozby webových aplikací. Dále jsou popsány nástroje pro etický hacking a technologie pro ochranu webových aplikací.

Praktická část práce zahrnuje tvorbu prostředí pro testování a vytváření testovacích scénářů. Prostředí je postaveno na platformě Virtual Box s operačním systémem Kali Linux. Pro testování jsou využity volně dostupné nástroje pro etický hacking, jako je Burp Suite, Sqlmap a Commix. Provedené testy jsou detailně analyzovány a porovnány s cílem vybrat nejlepší postupy pro ochranu webových aplikací.

Výsledky práce poskytují doporučení pro implementaci zvolených postupů ochrany a srovnání účinnosti útoků před a po aplikaci těchto opatření. Závěrečná část diskutuje výsledky a navrhuje možnosti dalšího výzkumu v oblasti bezpečnosti webových aplikací.

Klíčová slova: Etický hacking, testovací scénáře, nástroje pro etický hacking, webová aplikace, virtuální prostředí, Kali Linux, bezpečnost, hrozby, metodologie penetračního testování, DVWA

Ethical hacking and web application protection

Abstract
This thesis focuses on the issues of ethical hacking, web application security and protection against various forms of attacks. It aims to design and apply test scenarios to simulate attacks on a web application, then compare different protection methods and recommend the most effective approaches based on the measured results. The work also focuses on the analysis of the risks to which users are exposed and the security measures of web applications. The theoretical part discusses concepts such as ethical hacking, types of hackers, methods of ethical hacking, and security threats to web applications. Tools for ethical hacking and technologies for protecting web applications are also described.
The practical part of the work includes creating a testing environment and creating test scenarios. The environment is built on the Virtual Box platform with the Kali Linux operating system. For testing, freely available ethical hacking tools such as Burp Suite, Sqlmap and Commix are used. The tests performed are analysed and compared in detail to select the best practices for protecting web applications.
The results of the work provide recommendations for the implementation of the selected protection practices and a comparison of the effectiveness of attacks before and after the application of these measures. The final section discusses the results and suggests avenues for further research in web application security.

Keywords: Ethical hacking, test scenarios, ethical hacking tools, web applications, virtual environments, Kali Linux, security, threats, penetration testing methodology, DVWA

Obsah

1 Úvod **10**

2 Cíl práce a metodika **11**

2.1 Cíl práce 11

2.2 Metodika 11

3 Teoretická východiska **12**

3.1 Etický hacking 12

3.1.1 Typy hackerů 12

3.1.2 Metodologie 16

3.1.3 Typy etického hackingu 19

3.1.4 Etika a právní aspekty 21

3.2 Webové aplikace 24

3.2.1 Bezpečnostní hrozby a potencionální rizika 24

3.2.2 Způsoby ochrany webových aplikací 27

3.2.3 Metody a technologie pro ochranu webových aplikací 28

3.2.4 Bezpečnostní opatření 29

3.3 Prostředí 31

3.3.1 Virtual Box 31

3.3.2 Kali Linux 33

3.3.3 DVWA 35

3.4 Nástroje pro etický hacking 37

3.4.1 Volně dostupné nástroje pro etický hacking 37

3.4.2 Burp Suite 38

3.4.3 Sqlmap 40

3.4.4 Commix 40

3.5 Testovací scénáře 42

3.5.1 Scénářové penetrační testování 42

3.5.2 Výhody testování podle scénářů 43

3.5.3 Tvorba 44

3.6 Komparace 45

3.6.1 Postup komparace 45

4 Vlastní práce **47**

4.1 Tvorba prostředí 47

4.1.1 Volba virtuálního prostředí 48

4.1.2 Instalace a konfigurace operačního systému Kali Linux 49

4.1.3 Testovací aplikace 52

4.1.4 Zabezpečení testovacího prostředí 54

4.2 Tvorba testovacích scénářů 56
4.2.1 Nalezení zranitelností aplikace 56
4.2.2 Návrhy testovacích scénářů 61
4.3 Testování 63
4.3.1 Provedení testů 63
4.3.2 Výsledky 73
4.3.3 Návrh způsobů ochrany proti daným útokům na webové aplikace 73
4.4 Aplikace ochran 77
4.4.1 Aplikace ochran webové aplikace 77
4.5 Komparace účinnosti útoků před a po aplikování ochran 85
4.5.1 Testované aspekty 85
4.5.2 Komparace jednotlivých útoků 87
4.5.3 Výběr nejlepších postupů pro ochranu 90
4.5.4 Vyhodnocení 92

5 Výsledky a diskuse 93
5.1.1 Možnosti dalších testů 95

6 Závěr 96

8 Seznam použitých zdrojů 97

9 Seznam obrázků, tabulek, grafů a zkratek 102
9.1 Seznam obrázků 102
9.2 Seznam tabulek 103
9.3 Seznam grafů 103

Přílohy 104

1 Úvod

Rozvoj informačních technologií a neustálý nárůst používání webových aplikací vytváří závažnou potřebu pro zajištění bezpečnosti těchto online platforem. S růstem komplexity a sofistikovanosti útoků, jež jsou na tyto aplikace namířeny, se stává nezbytným zkoumat a vyvíjet metody ochrany, které zajistí integritu, důvěrnost a dostupnost informací pro uživatele. Tato diplomová práce se zaměřuje na klíčové aspekty bezpečnosti webových aplikací, přičemž klade důraz na identifikaci rizik a následnou implementaci testovacích scénářů a ochranných opatření.

V teoretické části práce jsou analyzovány základní koncepty etického hackingu, typy hackerů a metody etického hackingu. Dále se práce věnuje bezpečnostním hrozbám a potencionálním rizikům spojeným s webovými aplikacemi, stejně jako různým způsobům jejich ochrany. V rámci této části jsou rozebrány nástroje pro etický hacking, které budou v praktické části využity pro provedení penetračních testů.

Praktická část práce se soustředí na vytváření prostředí pro testování a následně na aplikaci testovacích scénářů, které simuluji různé formy útoků na webové aplikace. Výsledky těchto testů budou následně analyzovány s cílem vybrat nejefektivnější postupy pro ochranu webových aplikací. V neposlední řadě bude provedena komparace a zhodnocení účinnosti jednotlivých testovacích scénářů a navržených ochranných opatření.

Tato práce přináší pohled na problematiku bezpečnosti webových aplikací a jejich ochrany, poskytující tak poznatky pro odvětví informační bezpečnosti a výzkum v oblasti etického hackingu.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem diplomové práce je navrhnout testovací scénáře pro útoky na webové aplikace, provést následné porovnání různých způsobů ochrany a doporučit nejefektivnější postupy pro ochranu webových aplikací na základě naměřených výsledků. Práce se také zaměřuje na rizika, jimž jsou uživatelé vystaveni, bezpečnost webů a webových aplikací a na volně dostupné nástroje pro etický hacking. Dílčím cílem práce je aplikace testovacích scénářů a vyhodnocení jejich účinnosti při ochraně proti těmto útokům.

2.2 Metodika

Práce je rozdělena na teoretickou a praktickou část. V teoretické části je provedena komparace literárních zdrojů a je použita deskripční metoda pro podrobné zachycení problematiky, kde jsou zahrnuty útoky a nástroje pro jejich provedení, ochranné nástroje a způsoby ochrany. Tyto informace jsou následně aplikovány v praktické části. Zdrojem informací jsou vědecké, odborné články a publikace, které se zaměřují na dané téma. V praktické části jsou použity metody pozorování a měření provedených útoků a způsobů ochrany. Následně je provedena komparace výsledků a na základě nich jsou vybrány nejvhodnější testovací scénáře a postupy pro ochranu webových aplikací.

3 Teoretická východiska

3.1 Etický hacking

Etický hacking je proces, při kterém se provádí posouzení zabezpečení organizace pomocí technik podobných těm, které používají hackeři. Jedná se ale o legální činnost, která vyžaduje schválení a autorizaci od dané organizace nebo osoby. (Coursera, 2023) Cílem etického hackingu je využít taktik, technik a strategií kyberzločinců k identifikaci potenciálních slabých míst a posílení bezpečnosti organizace a ochrany dat. (Synopsys, 2023)

Osoby provádějící etický hacking jsou také známé jako white-hat hackeři a mají oprávnění a souhlas od organizace, do které se pokoušejí proniknout. (Coursera, 2023) Metody a techniky etického hackingu mohou zahrnovat penetrační testování, skenování zranitelností, testování webových aplikací a další. (Javatpoint, 2023) Naproti etickému hacking je tu hacking, který je motivován finančním ziskem, sabotáží, špionáží nebo dalšími nelegálními aktivitami. (Sinha, 2017 str. 18)

3.1.1 Typy hackerů

V oblasti informační bezpečnosti a kybernetiky rozlišujeme tři hlavní typy hackerů: hackery black hat, white hat a grey hat. Tato barevná terminologie vychází z touhy hackerů vymezit se a odlišit eticky správné hackery od těch nesprávných. Nejzásadnější black hat a white hat mají svůj původ ve westernových filmech. (Shea, 2019)

Heroické a záporné postavy byli obvykle snadno rozlišitelní podle barvy jejich kovbojských klobouků. Statečný a poctivý protagonisté často nosili bílý klobouk, zatímco záporné postavy preferovaly tmavý či černý klobouk. Tento koncept se postupem času zakotvil v širším kulturním povědomí, a nakonec se objevil i v terminologii informační bezpečnosti. (Norton, 2018 stránky 10-11)

3.1.1.1 White Hat

Autorizovaní hackeři neboli také white-hat hackeři, jsou právě ti, co jsou v odvětví informační bezpečnosti nazývají etickými hackery. (Norton, 2018 str. 11) Etický hacker využívá své dovednosti a znalosti v oblasti hackingu k identifikaci zranitelností a slabých míst v počítačových systémech, sítích nebo aplikacích, aby pomohl společnostem zlepšit jejich zabezpečení před útoky. (Terra, 2023) Zatímco většina neautorizovaných hackerů se

neřídí zákony ani oprávněními k útokům na systémy, white hat hackeři se jimi řídí. (Comptia, 2023) (Dvořáková, 2023)

Očekává se od nich, že se budou řídit etickým kodexem a zároveň budou při své činnosti dodržovat stanovené zákony a přístupová oprávnění. (Comptia, 2023) Obvykle jsou najímáni přímo společnostmi nebo klienty, aby testovali operační systémy, hardware, software a zranitelnosti sítí. Pronikají do systémů, aby našli zranitelná místa a společnosti tak mohly své systémy opravit a zmírnit potenciální kybernetické hrozby. (Norton, 2018 stránky 11-12)

V rámci své role také provádějí penetrační testy. Penetrační testy odhalí slabá místa v síti a otestují její bezpečnostní opatření. Může také určit, jak je zranitelná vůči útokům hackerů. (Froehlich, 2021) White hat hackeři hledají zranitelnosti nebo exploity pouze tehdy, když jim to zákon dovoluje. Svůj výzkum mohou provádět v softwaru s otevřeným zdrojovým kódem a také v softwaru nebo systémech, které vlastní nebo k jejichž zkoumání byli oprávněni, včetně produktů a služeb. Tyto typy programů odměňují jednotlivce penězi za odhalení bezpečnostních chyb. White hat hackeři plně odhalují všechny nalezené zranitelnosti společnosti nebo vlastníkova produktu, který je zodpovědný za opravu těchto chyb, aby mohly být problémy vyřešeny dříve, než je zneužijí hackeři s jinými úmysly. (Coursera, 2023)

3.1.1.2 Black Hat

Black hat hackeři jsou známý pod pojmem „cracker“. Je to neautorizovaná osoba, která se záměrně podílí na činnostech narušujících bezpečnost počítačového systému, uzavřeného softwarového kódu a informačních sítí. Hlavním cílem hackera, který pracuje proti vůli vlastníka systému, je získat neoprávněný přístup do cílového systému, přičemž jeho záměry sahají od krádeže, zničení informací až po narušení provozu systému, znemožnění přístupu legitimním uživatelům nebo převzetí kontroly z osobních důvodů. Někteří hackeři mohou dokonce využít své kontroly nad systémem k vydírání vlastníka požadováním výkupného výměnou za vzdání se kontroly. (Norton, 2018 str. 11)

Neautorizovaní hackeři mohou jednat na vlastní pěst, jako součást větší kyberzločinecké organizace nebo jménem nepřátelského národního státu. (Comptia, 2023) Většinou jsou motivováni pověstí, peněžním ziskem nebo špionáží. Hacker je označen jako black hat bez ohledu na to, zda své záměry vnímá jako čestné. To znamená, že i hackeři, kteří se zapojují do hackerských útoků ze sociálních nebo politických důvodů, jsou stále považováni za black

hat hackery, protože jejich záměrem je zneužít všechny zranitelnosti, které objeví. Stejně tak státem sponzorované hackerské aktivity prováděné nepřátelskými státy pro válečné účely spadají do kategorie black hat bez ohledu na jejich odůvodnění nebo mezinárodní postavení jejich státu. (Norton, 2018 str. 12)

Neautorizovaní hackeři jsou často pachateli mnoha významných úniků a zneužití dat. (Comptia, 2023) Většina z nich běžně používá k provádění útoků na organizace malware, sociální inženýrství a taktiku odepření služby. Mohou také vypustit malware, který ničí soubory, drží počítače jako rukojmí nebo krade hesla, čísla kreditních karet a další osobní údaje. (Kaspersky, 2018)

3.1.1.3 Grey hat

Mezi autorizovanými a neautorizovanými hackery existuje další typ hackerů, který představuje kombinaci obou skupin. Grey hat hackeři jsou jednotlivci, kteří využívají bezpečnostních zranitelností k tomu, aby zvýšili povědomí veřejnosti o existenci dané zranitelnosti. Tito hackeři sice nemají zlé úmysly, které jsou obvykle připisovány neautorizovaným hackerům, ale zároveň nemusí nutně dodržovat etický kodex, jak je tomu u autorizovaných hackerů. (Comptia, 2023) Hacker, který nemá povolení vlastníka nebo správce systému, může být považován za neetického při hledání bezpečnostních chyb. Věnují se neautorizovanému penetračnímu testování, jehož hlavním cílem je upozornit vlastníka na případné nedostatky. Grey hat hackeři často pronikají do systémů, které sami používají nebo využívají, aby posílili jejich bezpečnost a zabránili budoucím zásahům ze strany škodlivějších aktérů. (Norton, 2018 str. 12)

Mohou se rozhodnout soukromě odhalit bezpečnostní zranitelnost společnosti nebo výrobci, aniž by zveřejnili výsledky. Nicméně mnoho z nich veřejně využívá nalezenou zranitelnost v hardwaru nebo softwarových programech bez svolení výrobce, aby upozornili na tento problém. V oblasti kybernetické bezpečnosti existuje častý obav, že když grey hat hacker zveřejní zranitelnost, usnadňuje záškodnickým hackerům krádež informací a dat ze systémů. (Comptia, 2023)

3.1.1.4 Další typy hackerů

V historii kybernetické bezpečnosti došlo k několika pokusům rozšířit tradiční terminologii označující hackery podle barvy klobouku, aby lépe reflektovala různé skupiny s odlišnými motivacemi a dovednostmi. Ačkoli většina hackerů je obvykle kategorizována jako black

hat, white hat nebo grey hat, v rámci odvětví existuje mnoho dalších nuancovaných typů a subkategorií hackerů. (Buxton, 2022) V oblasti kybernetické bezpečnosti lze rozlišit řadu různých typů hackerů, z nichž každý má specifické motivace a cíle. (Shea, 2019)

3.1.1.4.1 Red hat

Red hat hackeři využívají své expertní dovednosti k podpoře organizací a jednotlivců v boji proti kybernetické kriminalitě. V komunitě informační bezpečnosti jsou red hat hackeři známí jako „vigilanti“ neboli kybernetičtí dozorci. (Trevino, 2023) Často se střetávají s hackery black hat a mohou proti nim zahájit kybernetické akce s cílem odhalit a zveřejnit jejich data a činnost. (Kim, 2018 str. 16) Na rozdíl od hackerů white hat, kteří by obvykle nahlásili nelegální aktivity úřadům, red hat hackeři často zahájí agresivní protiútoky s cílem zničit infrastrukturu a prostředky black hat hackerů. (Shea, 2019)

Red hat hackeři se řadí do takzvaného red teamu, kde se zaměřují na ofenzivní stránku kybernetické bezpečnosti. Jejich úkolem je útočit na systémy a následné prolamování obrany. Po sérii simulovaných útoků tým vytvoří organizaci doporučení, jak posílit zabezpečení systému a sítě. (Praveen, 2023)

3.1.1.4.2 Green hat

Green hat hackeři jsou v oboru nováčky a stále se učí v oblasti průnikového testování a hackingu. Přestože jsou začátečníci, přispívají k diverzifikaci pohledů v oblasti kybernetické bezpečnosti, zejména v rámci hackerů (Trevino, 2023) V porovnání s ostatními kategoriemi hackerů nemají takovou hloubku zkušeností, ale projevují silný zájem o další vzdělávání. (Shea, 2019)

I přesto, že jejich úmysly nemusí být zlomyslné, nedostatek znalostí může vést k neúmyslným škodám, pokud nejsou dodržována příslušná bezpečnostní opatření. Naopak, jsou to amatérští hackeři, kteří se snaží využít již existujících skriptů ve svých pokusech o hackování. Tito nezkušení hackeři často nerozumí plně následkům svých činů ani tomu, jak by mohli být zneužiti škodlivými aktéry. (Privacysense, 2023)

3.1.1.4.3 Blue hat

Společnosti je často najímají k identifikaci bezpečnostní chyby v jejich systémech nebo zranitelnosti v produktech, které ještě nebyly uvedeny na trh. Blue hat hackeři jsou podobní white hat hackerům v tom, že využívají své hackerské dovednosti, aby pomohli zlepšit kybernetickou bezpečnostní infrastrukturu organizace. (Trevino, 2023) Konference BlueHat, organizovaná firmou Microsoft a určená pouze pro pozvané, byla zřízena za

účelem podporování komunikace mezi hackery a inženýry. V určitých odborných kruzích může být blue hat hacker vnímán jako ten, jehož hlavní motivací je pomsta. Blue hat hackeři mohou být podobní začátečníkům, tedy green hat hackerům, ale odlišuje je jejich primární motivace k pomstě, nikoli snaha o zdokonalování svých dovedností v oblasti hackingu. (Shea, 2019)

Blue hat hackeři se ve skupině řadí se do takzvaného blue teamu, který se naopak zaměřuje na obranu před kybernetickými útoky a hrozbami. Pod tento tým spadá školení zaměstnanců v oblasti kybernetické bezpečnosti, skenování zranitelnosti sítě, řízení rizik a taktika zmírňování následků. (Praveen, 2023)

3.1.2 Metodologie

Metodika etického hackingu popisuje postupy, nástroje a techniky používané při etickém hackingu. Při přípravě na hackování systému se útočníci řídí určitou metodikou. (Praveen, 2023) Metodika klade důraz na komplexní a systematický přístup, který je přizpůsoben konkrétnímu systému nebo prostředí a zajišťuje odhalení a odstranění potenciálních slabých míst. (DianApps, 2019)

K dosažení tohoto cíle využívají etičtí hackeři řadu nástrojů, technik a osvědčených postupů, přičemž často simulují reálné útoky, aby posoudili odolnost systému. (Tripathy, 2023)

3.1.2.1 Průzkum a sběr informací

Po obdržení výslovného, smluvního a právního souhlasu organizace nebo osoby může začít průzkumná část procesu. (Praveen, 2023) Ta zahrnuje shromáždění co nejvíce informací o cíli pomocí různých nástrojů, které má hacker k dispozici, včetně webových stránek společnosti, internetového průzkumu, a dokonce i sociálního inženýrství. (Sengupta, 2021) Průzkum je přípravná fáze, kdy hacker dokumentuje požadavek organizace, zjišťuje cenné konfigurační a přihlašovací údaje systému a sonduje sítě. Tyto informace jsou klíčové pro provedení útoků a zahrnují: pojmenovací konvence, služby v síti, servery zpracovávající pracovní zátěž v síti, adresy IP, jména a přihlašovací údaje uživatelů připojených k síti, fyzické umístění cílového počítače a další. (Singh, 2022 str. 22)

V procesu musí být zahrnut rozsah penetračního testu, shromažďování informací pomocí vyhledávačů nebo sociálních sítí, techniky hackingu pomocí vyhledávače a zjištění otisků webových stránek. (Singh, 2022 str. 22) Do shromažďování informací patří penetračního testování 679, pod které spadá shromažďování informací WHOIS, shromažďování

informací o systému doménových jmen (DNS), shromažďování informací o síti, a nakonec sociální inženýrství. Stejný postup by totiž použil útočník při svém pokusu o narušení organizace. (Praveen, 2023)

3.1.2.2 Skenování

V další fázi skenování se přechází od pasivního k aktivnímu shromažďování informací tím, že se hledají způsoby, jak proniknout do sítě a obejít všechny zavedené systémy. (Praveen, 2023) Musí se provést několik kroků pro samotné skenování. Nejprve je nutné provést prověřování sítě při kterém se zjišťuje počet hostitelů v síti, skenování portů za účelem zjištění služeb, „grabování" bannerů cílových operačních systémů a portů, skenování zranitelností a vytvoření síťové topologie cílové sítě. (Singh, 2022 str. 22)

Do skenování se zahrnuje shromažďování informací o všech strojích, uživatelích a službách v síti pomocí automatizovaných skenovacích nástrojů. Při penetračním testování se obvykle provádějí tři typy skenování: (Sengupta, 2021)

a) Mapování sítě – Jedná se o zjišťování topologie sítě, včetně informací o hostitelích, serverech, směrovačích a firewallech v rámci hostitelské sítě. Po zmapování mohou white hat hackeři vizualizovat a strategicky plánovat další kroky procesu etického hackingu. (Sengupta, 2021)
b) Skenování portů – Etičtí hackeři používají automatizované nástroje k identifikaci všech otevřených portů v síti. Jedná se tak o účinný mechanismus, který umožňuje vyjmenovat služby a živé systémy v síti a způsob, jak s těmito komponentami navázat spojení. (Sengupta, 2021)
c) Skenování zranitelnosti – Použití automatizovaných nástrojů k odhalení slabých míst, která mohou být zneužita k organizování útoků. (Sengupta, 2021)

Mezi populární nástroje etického hackingu, které se běžně používají ve fázi skenování patří například SNMP Sweepery, prověřování pomocí pingu, mapovače sítě, skenery zranitelností a další. (Sengupta, 2021)

3.1.2.3 Zneužití (získání přístupu)

Následuje třetí fáze, během které hacker získá přístup k cíli, zjistí, kde se nacházejí různé zranitelnosti, a vyhodnotí, jak velké škody by mohl způsobit, když už má přístup. (Praveen, 2023) Třetí fáze zahrnuje pokus o odeslání škodlivé zátěže do aplikace prostřednictvím sítě, přilehlé podsítě nebo fyzicky pomocí připojeného počítače. (Sabih, 2018 str. 52) Hackeři

obvykle používají mnoho hackerských nástrojů a technik k simulaci pokusů o neoprávněný přístup, včetně například: phishing, útoky typu injection, zpracování externích entit XML, používání komponent se známými zranitelnostmi a další. (Sengupta, 2021)

Pokud jsou útoky úspěšné, hacker získá kontrolu nad celým systémem nebo jeho částí a může simulovat další útoky, jako je narušení dat a distribuované odepření služby (DDoS). (Sengupta, 2021) Získání přístupu do sítě/systému, lze dosáhnout několika způsoby jako je sociální inženýrství, shoulder surfing (surfování přes rameno), různé útoky na hesla, sniffing sítě, útoky typu Man-in-the-Middle (MiTM), používání různé techniky ke zneužití cílových systémů a k získání přístup přes příkazový řádek. A ty vedou ke zneužití služeb, objevování dalších zařízení a ke zvýšení oprávnění v napadeném systému nebo síti. (Singh, 2022 str. 23)

3.1.2.4 Post-exploatace (udržování přístupu)

Čtvrtá fáze procesu etického hackingu zahrnuje procesy, které hackeři používají pro zajištění přístupu k aplikaci pro budoucí použití. White-hat hacker průběžně zkoumá systém na další zranitelnosti a posouvá oprávnění tak, aby pochopil, jak velkou kontrolu mohou útočníci získat, jakmile projdou bezpečnostním prověřením. (Sengupta, 2021) Někteří útočníci se také mohou snažit skrýt svou identitu tím, že odstraní důkazy o útoku a nainstalují zadní vrátka pro budoucí přístup. (Grimes, 2017 str. 29) Lze tedy předpokládat, že průměrný útočník není hned uvnitř a není hned venku. Jakmile se nějakému útočníkovi podaří úspěšně proniknout do sítě nebo systému, zůstává tam co nejdéle. (Praveen, 2023)

Aby se tedy hackeři vyhnuli jakýmkoli důkazům, které vedou zpět k jejich škodlivé činnosti, provádějí úkoly, které vymažou všechny stopy po jejich činnosti. Mezi ně patří odinstalování skriptů a aplikací používaných k provádění útoků, úprava hodnot registru, vymazání protokolů a odstranění složek vytvořených během útoku. (Sengupta, 2021) Od normálního vloupání se tato fáze moc neliší, neboť pachatel může věnovat čas tomu, aby odstranil veškeré důkazy o svém činu. V této fázi bude hacker hledat jakékoli stopy po své činnosti a odstraňovat je. (Praveen, 2023) Hackeři, kteří si chtějí zachovat nepozorovaný přístup, mají tendenci skrývat svou identitu pomocí technik, jako jsou například takzvané tunelování. (Sengupta, 2021)

3.1.2.5 Poskytnutí závěrečné zprávy

V poslední části etický hacker shromáždí všechny poznatky získané během svého úkolu a podá o nich organizaci zprávu, včetně doporučení, jak se vyhnout budoucím bezpečnostním incidentům. (Praveen, 2023) Nedílnou součástí je tedy vypracování zprávy a předání výsledků. Musí být vytvořen oficiální dokument, ve kterém jsou uvedeny informace. A to všechny zranitelnosti nalezené na cílech, všechna rizika, rozdělená do kategorií na stupnici vysoká, střední a nízká, kalkulátor CVSS (Calculator Vulnerability Scoring System) a doporučené metody nápravy a zmírnění zjištěných zranitelností. (Singh, 2022 str. 680)
Závěrečnou zprávu je nutné napsat tak, že ji pochopí každý, kdo si ji přečte včetně netechnického publika, jako je vrcholový management a vedoucí pracovníci členy. Zpráva by měla obsahovat informace v pořadí titulní list, shrnutí, shrnutí zranitelností, podrobnosti o testech, nástroje použité při testování, původní rozsah práce, hlavní část zprávy. (Singh, 2022 str. 681)

3.1.3 Typy etického hackingu

Etičtí hackeři využívají k posouzení bezpečnosti digitální infrastruktury společnosti širokou škálu odborných znalostí a různých nástrojů a strategií. Operace kybernetické bezpečnosti často využívají typy etického hackingu. (Kanumilli, 2022)

3.1.3.1 Black-box testování

Při black-box testování hacker o systému předem nic neví. Místo toho testuje software zvenčí a poté se do systému dostane pomocí hrubé síly. (Poston, 2020) Tento typ etického hackingu je jedním z nejnebezpečnějších v oblasti kybernetické bezpečnosti, protože se používá také k identifikaci bezpečnostních mezer v síti nebo systému, které by mohl útočník využít. aby mohli spáchat trestné činy. (Kanumilli, 2022)
Black-box testování lze využít například k ověření přihlášení uživatele, k procházení profilu uživatele, změně hesla a další. Návrh takového testu nevyžaduje znalost kódu aplikace. (Kanumilli, 2022)

3.1.3.2 White-box testování

White-box testování je proces, při kterém hacker disponuje kompletními znalostmi o systému, včetně jeho vnitřního fungování a potenciálních zranitelností, a pokouší se o průnik do něj. (Poston, 2020) Vývojáři často využívají white-box testování k tomu, aby

ověřili funkčnost svých systémů při zátěži před jejich nasazením do produkčního prostředí. Informace o interním dění organizace jsou koordinovány s IT oddělením a dodržují se stanovená pravidla. (Kanumilli, 2022)
Přijímají se také opatření s cílem zabránit neoprávněnému přístupu hackerů do firemní sítě. (Kanumilli, 2022) White-box testování zahrnuje i revizi návrhu, kontrolu kódu, analýzu toku dat a ověření, zda jsou splněny stanovené požadavky. (Poston, 2020)

3.1.3.3 Grey-box testování

Hybrid mezi testováním white-box a black-box, při kterém má tester určité, ale ne úplné znalosti systému a musí použít deduktivní uvažování a technické znalosti, aby odhalil bezpečnostní chyby v cílovém systému nebo síti. (Kanumilli, 2022) Grey hat hackeři využívají své znalosti k dobrým i špatným účelům, například když pomocí počítačových virů kradou peníze z bank a jiných podniků. (Poston, 2020)
Mezi grey-box testování patří testování použitelnosti nebo výkonu. (Poston, 2020) Znalost toho, jak dobře bude vaše aplikace fungovat i v reálných podmínkách, je zásadní pro její úspěšný vývoj a tato metoda v tom může pomoci. (Kanumilli, 2022)

3.1.3.4 Vniknutí do webové aplikace

Nabourat se do webové aplikace znamená využít její bezpečnostní nedostatky. K vytváření webových aplikací se nejčastěji používají jazyky HTML, CSS a JavaScript, PHP a další. Vzhledem ke způsobu, jakým webové prohlížeče tyto jazyky čtou, je možné vydávat se za oprávněného uživatele a provádět na webu určité činnosti. (Kanumilli, 2022)
Webové aplikace, sdílené přes síť jako je například internet a často založené na prohlížeči, mohou být zranitelné vůči útokům skriptů. (Skillmea, 2021) Příkladným útokem je Cross-site scripting (XSS), který modifikuje HTML webové stránky tak, aby obsahoval škodlivý obsah. Při dobře provedeném útoku XSS může být špatná osoba také schopna převzít relaci prohlížeče oběti se serverem, aniž by získala její přihlašovací údaje. (Kanumilli, 2022)

3.1.3.5 Zneužití bezdrátové sítě

V tomto typu se využívají bezpečnostní díry v systému pro získání přístupu do počítačové sítě bez povolení. Používá se pro to technika známá jako wardriver, při níž útočník jezdí po otevřených nebo nedostatečně chráněných bezdrátových sítích pomocí notebooku nebo jiného zařízení schopného zachytit bezdrátový signál. (Kanumilli, 2022)

Využití bezdrátových sítí je dalším aspektem etického hackování, který se zaměřuje na detekci a eliminaci potenciálních zranitelností v bezdrátovém prostředí. Tato oblast je často vyhledávána útočníky z důvodu existujících slabých míst, která mohou umožnit neoprávněný přístup. Tyto slabiny mohou být způsobeny špatnou konfigurací sítě, nedostatečným šifrováním a použitím zastaralých bezpečnostních protokolů. (Rau, 2023)

3.1.4 Etika a právní aspekty

Hacking se stal celosvětovou kybernetickou kriminalitou a obtěžuje národy v oblasti bezpečnosti, narušení bezpečnosti dat, finančních narušení, podvodů atd. Neetické hackerství je v očích každého národa jednoznačně trestným činem. Na celém světě byly vytvořeny různé právní předpisy a zákony, které chrání práva jednotlivce ve virtuálním světě a které musí etický hacker dodržovat. (Shukla, 2019)

3.1.4.1 Etika

1. Dodržování předpisů: Organizace musí respektovat platné zákony a předpisy týkající se ochrany dat, soukromí a duševního vlastnictví. Například v USA je důležité dodržovat zákon o počítačových podvodech a zneužití (CFAA), který trestá neoprávněný přístup k počítačům. V Evropské unii je pak důležité dodržovat obecné nařízení o ochraně osobních údajů (GDPR) a zákon o ochraně soukromí v elektronických komunikacích (ECPA). (Gillam, 2023)
2. Odpovědnost: Organizace musí zvážit potenciální odpovědnost spojenou s penetračním testováním, včetně možných škod způsobených testováním a reálných důsledků pro systémy. Je třeba najímat kvalifikované a důvěryhodné penetrační testery a zajistit pojistné krytí pro případnou odpovědnost. (Gillam, 2023)
3. Dokumentace: Organizace musí vést podrobnou dokumentaci o penetračním testování, včetně rozsahu, metod a výsledků. Tato dokumentace slouží k prokázání profesionálního postupu organizace a splnění právních požadavků. Důkladná dokumentace může pomoci organizacím minimalizovat potenciální odpovědnost a prokázat náležitou péči v případě narušení bezpečnosti. (Gillam, 2023)

3.1.4.2 Právní aspekty

Právní aspekty hackingu se neustále vyvíjí, například evropský parlament ratifikoval nové legislativní akty, označované jako směrnice NIS2, s cílem zesílit kybernetickou bezpečnost ve strategických sektorech v rámci Evropské unie. Tato směrnice rozšiřuje spektrum odvětví, jež podléhají jejím nařízením, zahrnujíc sektory energetiky, dopravy, bankovnictví, zdravotní péče a digitální infrastruktury. Dále stanovuje pro členské státy EU přísnější regulační povinnosti v oblasti kybernetické bezpečnosti a intenzifikuje spolupráci mezi nimi. Evropská unie rovněž vypracovává Legislativní akt o kybernetické rezilientnosti, jenž definuje bezpečnostní normy pro digitální produkty, a Legislativní akt o digitální operativní rezilientnosti, jehož primárním účelem je ochrana finančního sektoru před kybernetickými hrozbami. (European Parlament, 2023)

V globálním kontextu Evropský parlament nabízí komplexní právní rámec v oblasti hackingu, primárně zaměřený na aktivity vynucovacích orgánů. (Gutheil, a další, 2017) Tato pravidla ovlivňují formování politik kybernetické bezpečnosti v mnoha zemích. (Mishra, 2022) Ačkoliv etický hacking je klíčovým prvkem v oblasti kybernetické bezpečnosti, je nezbytné nalézt rovnováhu mezi bezpečnostními požadavky a právními omezeními. Různé státy proto upravují svou legislativu tak, aby bylo zajištěno, že tato praxe je prováděna v souladu s etikou a zákonem. (Gutheil, a další, 2017)

V České republice se v průběhu posledních tří let věnuje zvýšená pozornost etickému hackingu v reakci na rostoucí potřebu kybernetické bezpečnosti. Zpráva o kybernetické bezpečnosti z roku 2020 zdůrazňuje důležitost etického hackingu jako nástroje k posilování národní digitální infrastruktury a zabezpečení dat. Je důležité poznamenat, že právní normy vyžadují, aby etičtí hackeři dodržovali stanovené směrnice a zabezpečovali ochranu dat proti kyberkriminalitě. Byly také zahájeny vzdělávací programy pro etické hackery, což potvrzuje význam této disciplíny v zemi. (Řehka, 2020)

Existuje několik zásadních právních předpisů týkajících se kybernetické kriminality:

- Článek 84 - Mezi hackery black-hat a white-hat existuje jemná hranice. Tento článek stanovuje ochranu, kterou poskytuje vláda, správci a dalším osobám, které jednají v dobré víře jménem těchto autorit. Etický hacker jmenovaný vládou nebo správcem jedná v souladu se zákony a předpisy. (Shukla, 2019)
- Zákon § 43 - Podle tohoto zákona se jedná o trestný čin, pokud osoba bez souhlasu vlastníka nebo jiné osoby upravuje, poškozuje nebo narušuje počítačovou síť, stahuje, kopíruje nebo získává z ní data. Pokud osoba jedná

na základě pověření nebo v dobré víře, nenese odpovědnost za způsobenou škodu. (Shukla, 2019)

- Článek 43-A – Tento článek stanoví, že nedostatečná ochrana dat zavazuje osobu k náhradě škody. Etický hacker, který nedodrží ochranu údajů, nese odpovědnost podle tohoto článku. (Shukla, 2019)
- Oddíl 66 - Tento oddíl zákona upravuje trestné činy související s počítači. Každá osoba, která se dopustí nepoctivého a podvodného jednání podle článku 43, může být potrestána trestem trvajícím až 3 roky. (Shukla, 2019)
- Zákon § 66-F – Vládní agentury mohou najímat kybernetické odborníky, aby se bránily kybernetickému terorismu podle § 66-F, kde je definováno neoprávněné nebo překročení oprávněného přístupu. (Shukla, 2019)

Je nezbytné, aby organizace před provedením penetračního testování zvážily etické a právní aspekty. (Gillam, 2023) Tímto způsobem lze zajistit, že testování bude prováděno zodpovědně a profesionálně a minimalizují se potenciální rizika. (Shukla, 2019)

3.2 Webové aplikace

Webová aplikace je počítačový program uložený na vzdáleném serveru a spouštěný uživateli prostřednictvím webového prohlížeče. Je to aplikační program, který využívá mnoho webových technologií k provádění úkolů na internetu. (Javatpoint, 2023) Webové aplikace mají několik výhod oproti klasickým desktopovým aplikacím. Použití prohlížečů umožňuje kompatibilitu aplikace s většinou standardních počítačů a operačních systémů. Dále webové aplikace nezabírají paměť na pevném disku počítače a jsou přístupné téměř z jakéhokoli počítače nebo zařízení, které má uživatel k dispozici. (Volle, 2022)

Další výhodou webových aplikací je možnost současného používání více uživateli, což umožňuje současnou účast a spolupráci. Ačkoli webové aplikace vyžadují připojení k síti, s rozšířením internetu toto omezení ztrácí na významu. Webové aplikace mohou mít různé účely, od internetových obchodů, webových e-mailů, kalkulaček, až po platformy sociálních médií. Většina webových aplikací je přístupná pomocí běžného webového prohlížeče, ale některé speciální typy vyžadují specifické prohlížeče. (Volle, 2022)

Webové aplikace často využívají skripty na straně serveru, jako je PHP nebo ASP, pro zpracování informací a ukládání dat. Klient strana může využívat skripty jako JavaScript a HTML pro reprezentaci dat před uživateli. Některé webové aplikace kombinují skripty na straně serveru i klienta. Webové aplikace umožňují uživatelům komunikovat s organizacemi nebo firmami pomocí online formulářů, fór, nákupních košíků, systémů správy obsahu a dalších nástrojů. (Javatpoint, 2023)

Uživatelé mohou také vytvářet a sdílet dokumenty a data pomocí webových aplikací. Díky nim mohou uživatelé spolupracovat na projektech bez ohledu na geografickou polohu. Webové aplikace poskytují uživatelům flexibilitu a snadný přístup k funkcím a informacím prostřednictvím internetu. (Javatpoint, 2023)

3.2.1 Bezpečnostní hrozby a potencionální rizika

Webové aplikace mohou čelit řadě typů útoků v závislosti na cílech útočníka, povaze práce cílové organizace a konkrétních bezpečnostních nedostatcích aplikace. (Cloudflare, 2023) Zabezpečení webových aplikací je klíčové pro ochranu důležitých dat a bránění se proti kybernetickým hrozbám. Bez adekvátních bezpečnostních opatření mohou webové platformy padnout za oběť hackerům, což může mít za následek finanční ztráty, krádeže identit a škody na jednotlivcích či firmách. (Wallis, 2023)

3.2.1.1 Zranitelnosti nultého dne

Jedná se o zranitelnosti, které tvůrci aplikace neznají, a které tedy nemají k dispozici opravu. Útočníci se pokouší tyto zranitelnosti rychle zneužít ideálně v den vydání aplikace. (Cloudflare, 2023) Tyto zranitelnosti mohou být využity útočníky k provádění neoprávněných operací, jako je například neautorizovaný přístup k citlivým datům, implantace škodlivého softwaru nebo ovládnutí systému. (Eset, 2023)

Útočníci mohou tyto zranitelnosti využít k infikování systému různými typy malware, jako je spyware, trojský kůň, ransomware, keylogger nebo jiné formy škodlivého softwaru. Tím může dojít k úniku dat nebo dokonce k ovládnutí cílového zařízení. (Eset, 2023)

3.2.1.2 Cross site scripting (XSS)

XSS je zranitelnost, která útočníkovi umožňuje injektovat skripty na straně klienta do webové stránky s cílem získat přímý přístup k důležitým informacím. (Cloudflare, 2023) Vydává se za uživatele nebo se uživatele pokouší oklamat, aby odhalil důležité informace. (Wallis, 2023) Pokud uživatel navštíví stránku napadenou XSS, může dojít k automatickému spuštění škodlivého kódu ve webovém prohlížeči, což může mít za následek různé útoky a nežádoucí aktivity. (KirstenS, 2024)

Existuje několik druhů XSS útoků, jako je Reflected XSS, Stored XSS a DOM-based XSS, které se liší v metodách vstřikování a spouštění škodlivého kódu. Útočníci mohou tuto zranitelnost využít k úniku citlivých dat, přesměrování uživatelů na phishingové stránky, krádeži cookies, manipulaci obsahu stránek a dalším nežádoucím aktivitám. (KirstenS, 2024)

3.2.1.3 SQL injection (SQLi)

SQLi je metoda, kdy útočník zneužívá zranitelnosti ve způsobu, jakým databáze provádí vyhledávací dotazy. (Wallis, 2023) Útočníci pomocí SQLi získávají přístup k neoprávněným informacím, upravují nebo vytvářejí nová uživatelská oprávnění, manipulují s citlivými daty nebo je ničí. (Cloudflare, 2023)

Existuje mnoho způsobů, jak provádět SQL injection, včetně vkládání SQL kódu do formulářů, URL parametrů, HTTP hlaviček a dalších uživatelských vstupů. Útočník může vložit SQL příkazy, aby získal citlivá data, jako jsou hesla a uživatelská jména uložená v databázi. Je důležité, aby organizace přijaly opatření k prevenci těchto útoků a chránily tak bezpečnost svých systémů a dat. (Acunetix, 2024)

3.2.1.4 Útoky typu DoS (Denial-of-service) a DDoS (distributed denial-of-service)

Představují formy kybernetických útoků, které zneužívají zdroje k zablokování přístupu uživatelů k cílovému prostředku, jako jsou webové stránky, e-mailové servery, telefonní sítě a další komunikační infrastruktura. (Thefastcode, 2023)
Prostřednictvím různých vektorů jsou útočníci schopni přetížit cílový server nebo jeho okolní infrastrukturu. (Wallis, 2023) Když server přestane být schopen efektivně zpracovávat příchozí požadavky, začne se zpomalovat, a nakonec odmítne obsluhu příchozích požadavků od legitimních uživatelů. (Cloudflare, 2023)

3.2.1.5 Přetečení vyrovnávací paměti

Přetečení vyrovnávací paměti je anomálie, ke které dochází, když software zapisuje data do definovaného prostoru v paměti známého jako vyrovnávací paměť. Toto chování lze zneužít k vložení škodlivého kódu do paměti a potenciálně tak vytvořit zranitelnost v cílovém počítači. (Cloudflare, 2023)
Útočníci mohou přebytečná data, uchovávaná v sousedních adresách paměti, použít jako výchozí bod pro zahájení útoku. Je důležité implementovat bezpečnostní opatření, jako je kontrola vstupů a výstupů, a pravidelně aktualizovat softwarové systémy, aby se minimalizovala možnost využití této zranitelnosti útočníky. (Imperva, 2023)

3.2.1.6 Cross site request forgery (CSRF):

Cross site request forgery spočívá v podvedení oběti, aby provedla požadavek, který využívá jejího ověření nebo autorizace. (Cloudflare, 2023) Využitím oprávnění účtu uživatele je útočník schopen odeslat požadavek vydávající se za uživatele. (Glas, a další, 2021)
Po kompromitaci uživatelova účtu může útočník zničit nebo změnit důležité informace. Útočníci se běžně zaměřují na vysoce privilegované účty, jako jsou správci nebo vedoucí pracovníci. (Glas, a další, 2021)

3.2.1.7 Zneužití rozhraní API

Rozhraní API neboli rozhraní pro programování aplikací je software, který umožňuje dvěma aplikacím vzájemně komunikovat. (Cloudflare, 2023) Stejně jako každý typ softwaru mohou obsahovat zranitelnosti, které útočníkům umožňují odeslat do jedné z aplikací škodlivý kód nebo zachytit citlivá data při jejich přenosu z jedné aplikace do druhé. S rostoucím

využíváním rozhraní API se jedná o stále častější typ útoku. Rizika, kterým dnes organizace čelí jsou shrnuta v seznamu OWASP API. (Glas, a další, 2021)

3.2.1.8 Zneužití kódu třetích stran

Mnoho moderních webových aplikací využívá různé nástroje třetích stran jako například weby elektronického obchodu používající nástroj třetí strany pro zpracování plateb. Útočníci mohou být schopni tento nástroj kompromitovat a ukrást data, která zpracovává, zabránit jeho fungování nebo jej použít k injektáži škodlivého kódu na jiné místo v aplikaci. (Cloudflare, 2023)

3.2.1.9 Chybná konfigurace útočné plochy

Útočná plocha organizace je celá její IT stopa, která může být náchylná ke kybernetickým útokům: servery, zařízení, SaaS a cloudová aktiva, která jsou přístupná z internetu. Tato útočná plocha může zůstat zranitelná vůči útokům v důsledku přehlédnutí nebo chybné konfigurace některých prvků. (Cloudflare, 2023) Při průzkumu v roce 2021 až 90 % aplikací bylo testováno na nějakou formu chybné konfigurace. (Glas, a další, 2021)

3.2.2 Způsoby ochrany webových aplikací

Zabezpečení webových aplikací je rozsáhlá a neustále se měnící disciplína. (Xu, 2021) Ochrana se proto odvíjí od toho, jak se objevují nové útoky a zranitelnosti. V dnešní době je internetových hrozeb je tolik aktivních, že se žádná organizace neobejde bez určitých bezpečnostních služeb. (Cloudflare, 2023)

Webových aplikace jsou častým terčem hackerských útoků a mohou být napadnuty odkudkoli na světě. Přesto existují standardizované metody a nástroje, které umožňují vývojářům vytvářet bezpečné aplikace. (Xu, 2021)

1. **Zmírňování útoků DDoS – Služby** pro zmírnění DDoS se nacházejí mezi serverem a veřejným internetem a používají specializovanou filtraci a extrémně vysokou kapacitu pásma, aby zabránily nárazovému škodlivému provozu zahltit server. Dnes se bez této služby nedá obejít, protože moderní útoky DDoS dokážou zahltit i nejodolnější servery. (Cloudflare, 2023)
2. **Brána WAF (Web Application Firewall)** – Brána filtrující provoz, o němž je známo nebo existuje podezření, že využívá zranitelnosti webových aplikací. Nové

zranitelnosti se objevují příliš rychle a tiše na to, aby je téměř všechny organizace dokázaly zachytit samy. (Cloudflare, 2023)

3. **Brány API** – Pomáhají spravovat a monitorovat provoz API, identifikovat přehlížená rozhraní API a blokují provoz, o němž je známo, že se zaměřuje na zranitelnosti rozhraní API, nebo je podezřelý z jejich zneužití. (Cloudflare, 2023) (Xu, 2021)
4. **DNSSEC** – Protokol, který zaručuje, že provoz DNS webové aplikace je bezpečně směrován na správné servery, takže uživatelé nejsou napadeni útočníkem na cestě. (Cloudflare, 2023)
5. **Správa šifrovacích certifikátů** – Třetí strana spravuje klíčové prvky šifrovacího procesu SSL/TLS, jako je generování soukromých klíčů, obnovování certifikátů a odvolávání certifikátů z důvodu zranitelnosti. To má za následek odstranění rizika přehlédnutí těchto prvků a odhalení soukromého provozu. (Cloudflare, 2023)
6. **Správa botů** – Využívá strojové učení a další specializované metody detekce k rozlišení automatizovaného provozu od lidských uživatelů a zabrání těm prvním v přístupu k webové aplikaci. (Cloudflare, 2023)
7. **Zabezpečení na straně klienta** – Kontroluje nové závislosti na JavaScriptu třetích stran a změny kódu třetích stran, což pomáhá organizacím dříve zachytit škodlivé aktivity. (Cloudflare, 2023) (Xu, 2021)
8. **Správa útočného povrchu** – Akční nástroje pro správu útočného povrchu by měly poskytovat jediné místo, kde lze zmapovat útočný povrch, identifikovat potenciální bezpečnostní rizika a několika kliknutími je zmírnit. (Cloudflare, 2023)

3.2.3 Metody a technologie pro ochranu webových aplikací

Existuje několik metod a technik pro ochranu webových aplikací mezi které patří kontroly zabezpečení aplikací. (Crowdstrike, 2023) To jsou techniky, které zlepšují zabezpečení aplikací na úrovni kódu a snižují zranitelnost. Jsou navrženy tak, aby reagovaly na neočekávané vstupy, jako jsou vstupy způsobené vnějšími hrozbami. Pomocí kontrolních mechanismů zabezpečení aplikací mají programátoři větší kontrolu nad reakcemi na neočekávané vstupy. Zabezpečení aplikací pomáhá podnikům odvrátit hrozby pomocí nástrojů a technik určených ke snížení zranitelnosti. (Krishna, 2021)

Jsou to kroky přidělené vývojářům k implementaci bezpečnostních standardů. Jedním z hlavních předpisů, které musí podniky dodržovat, je speciální publikace Národního

institutu pro standardy a technologie (NIST SP), která poskytuje pokyny pro výběr bezpečnostních kontrol. (Crowdstrike, 2023)

Existují různé typy kontrol zabezpečení aplikací určené pro různé přístupy k zabezpečení, které zahrnují: (Crowdstrike, 2023)

1. Autentizace: Potvrzení, zda je identita uživatele platná; nezbytné pro vynucení přístupu založeného na identitě. (Crowdstrike, 2023)
2. Šifrování: Převod informací nebo dat do kódu, aby se zabránilo neoprávněnému přístupu; může se týkat jednotlivých souborů nebo celého projektu. (Crowdstrike, 2023)
3. Protokolování: Zkoumání činnosti uživatele za účelem kontroly případů podezřelé činnosti nebo narušení. (Crowdstrike, 2023)
4. Kontroly platnosti: Kontrola, zda zadaná a zpracovávaná data splňují určitá kritéria. (Crowdstrike, 2023)
5. Kontrola přístupu: Omezení přístupu k aplikacím na základě IP adres nebo jinak oprávněných uživatelů. (Crowdstrike, 2023)

3.2.4 Bezpečnostní opatření

Vytvoření plánu zabezpečení webové aplikace – Provozovat webovou aplikaci bez plánu zabezpečení je nebezpečné. Proto je dobré si připravit plán, co dělat, pokud bude webová aplikace napadena hackery. Plán by měl obsahovat konkrétní údaje o osobách, které webovou aplikaci chrání, a o tom, která aplikace by měla být zabezpečena jako první, pokud společnost čelí krizi. (Krishna, 2021) Důležitou částí je provádění neustálého testování prostřednictvím manuálního, cloudového řešení, softwaru nebo poskytovatelů spravovaných služeb. (Beatrice, 2020)

Sledování aktiv – Zaměstnanec nebo vlastník webové aplikace nemůže znát každý detail ve své organizaci. Je ale důležité mít základní povědomí o tom, které servery organizace používá pro konkrétní funkce nebo aplikace. Sledování softwarového majetku zachraňuje před katastrofou, která by mohla přijít v budoucnu. Tento proces by měl být co nejvíce automatizován, aby organizace mohly škálovat svůj vývoj. (Beatrice, 2020)

Posouzení hrozeb – Sestavením seznamu toho, co je třeba ve webové aplikaci chránit pomůže zjistit, jaký druh bezpečnostních problémů hrozí a co by se dalo udělat jako proaktivní opatření k jeho zakrytí. Je nutné se soustředit na možnosti jaké cesty by mohli

hackeři použít k prolomení aplikace, jaká jsou stávající bezpečnostní opatření, která by útok odhalila nebo mu zabránila, jaké jsou zapotřebí nástroje a další. (Beatrice, 2020)

Prioritizace zabezpečení – Organizace si již nemohou dovolit ponechat kybernetickou bezpečnost pouze na bezpečnostních specialistech. Zabezpečení webových aplikací by mělo být integrováno do všech fází procesu vývoje, provozu a testování. (Beatrice, 2020)

Zálohování dat webových stránek – Zálohování všech informací na webových stránkách je povinný proces. Zálohování může pomoci v případě napadení malwarem nebo narušení bezpečnosti, protože organizaci bude stačit obnovit webovou aplikaci a získat přístup ke svým dříve uloženým datům. (Beatrice, 2020) (Krishna, 2021)

Zavedení programu odměn – Skvělým způsobem, jak získat zpětnou vazbu od komunity ohledně potenciálních problémů se zabezpečením webových aplikací, je zavedení programu odměn. Nabídnutím odměny a peněžních hodnot lze podpořit komunitu v oblasti bezpečnostních rizik. (Beatrice, 2020)

Šifrování dat – Šifrování je základní proces kódování informací, který je chrání před kýmkoli, kdo k nim nemá oprávněný přístup. Samotné šifrování nezabraňuje zásahům do přenosu dat, ale zastírá srozumitelný obsah pro ty, kteří k němu nemají oprávnění. Lze použít i k zabezpečení dat v klidu. (Beatrice, 2020)

Skenování zranitelností – Pravidelné kontroly a skenování zabezpečení mohou udržet riziko webových aplikací. Skenování je nutné provádět na určité bázi. Odborníky je doporučováno alespoň jednou týdně nebo když dojde ke změně. (Beatrice, 2020)

Automatizace a integrace bezpečnostních nástrojů – V dnešní době většina IT institucí poskytuje automatizovaná a integrovaná bezpečnostní řešení, neboť manuální kroky jsou náchylné na větší riziko hrozby. Zprávy o automaticky ověřených zranitelnostech se načítají přímo do sledovačů chyb vývojářů a rovnou do fáze opravy. (Beatrice, 2020)

Školení zaměstnanců – Neustálé školení zaměstnanců a jejich pravidelné testování v oblasti zabezpečení webových stránek. Někteří mají buď jen základní znalosti o této problematice, nebo vůbec žádné. Díky tomu budou sami odhalovat zranitelná místa. (Beatrice, 2020) Je důležité organizovat vzdělávací kurzy i pro vývojářský tým v oboru zabezpečení, což posílí jejich znalosti o nejlepších praktikách v oblasti zabezpečení webových aplikací. Díky tomu budou schopni vytvářet bezpečnější kód a eliminovat typické chyby vedoucí k zranitelnostem. (Krishna, 2021)

3.3 Prostředí

Vytvoření virtuální laboratoře pro penetrační testování umožní vytvořit bezpečné prostředí pro zdokonalování dovedností, škálovat prostředí a přidávat nové zranitelné systémy, a odstraňovat starší systémy, které nejsou potřeba, a dokonce vytvářet virtuální sítě pro přesun útoků z jedné sítě do druhé. (Sinha, 2017 str. 18)

Koncept vytvoření vlastní virtualizované laboratoře pro penetrační testování umožní maximalizovat prostředky na stávajícím počítači, bez nutnosti nákupu online laboratoře od různých poskytovatelů služeb nebo dokonce kupovat další počítače a služby. (Singh, 2022 str. 33)

3.3.1 Virtual Box

VirtualBox představuje robustní virtualizační řešení pro architektury x86 a AMD64/Intel64, vhodné jak pro podnikové, tak pro domácí prostředí. Nabízí širokou škálu funkcí a vysoký výkon pro korporátní klienty, a zároveň je jedním z mála profesionálních virtualizačních nástrojů s otevřeným zdrojovým kódem dostupným pod licencí GNU General Public License (GPL) verze 3. (Hasenmueller, 2023)

VirtualBox aktuálně podporuje operační systémy Windows, Linux, MacOS, Solaris a další. Podporuje i širokou paletu hostovaných operačních systémů, včetně různých verzí Windows, DOS/Windows 3.x, různých verzí Linuxu, Solaris a OpenSolaris, OS/2 a OpenBSD. (Hasenmueller, 2023)

Vývoj VirtualBoxu je dynamický a jeho aktualizace jsou pravidelně zveřejňovány. Seznam podporovaných funkcí, hostovaných operačních systémů a platforem je kontinuálně rozšiřován. VirtualBox je komunitním projektem, který je podporován společností Oracle, jež se zavázala udržovat produkt na profesionální úrovni kvality. (Hasenmueller, 2023)

VirtualBox nabízí hodnotu pro správce IT, koncové uživatele a vývojáře tím, že jim poskytuje flexibilní nástroj pro práci s virtuálními počítači. VirtualBox umožňuje vývoj aplikací na mnoha různých platformách a simulaci virtuálních sítí pro důkladné testování svých produktů. (Wallen, 2017)

A i když mnoho odborníků vnímá VMware jako vedoucí podnikové virtualizační řešení, VirtualBox má své pevné místo v podnikovém prostředí díky své schopnosti efektivně a spolehlivě provozovat virtuální počítače. (Wallen, 2017)

3.3.1.1 Využití a výhody VirtualBoxu

Spuštění více, často různých, operačních systémů na jednom počítači za účelem testování různých typů útoků nebo scénářů. To poskytuje nákladově efektivní metodu, jak to provést bez nutnosti používat více počítačů, a také poskytuje prostředí, které lze snadno obnovit a znovu spustit nebo přehrát, pokud se něco poškodí nebo je třeba provést demonstraci klientovi. (Zerodaysnoop, 2020)

Provozování jiného operačního systému pro účely penetračního testování, než jaký se používá na běžném počítači pro e-mail, psaní dokumentů atd., navíc virtuální počítače chrání hostitelský počítač před činnostmi prováděnými ve virtuálním počítači. (Wallen, 2017)

Mezi hlavní výhody tohoto přístupu patří:

- Nákladová efektivita: Umožňuje výrazné úspory, jelikož odpadá potřeba pořizovat a spravovat více fyzických strojů pro každý operační systém.
- Bezpečnost: Pracovní prostředí ve virtuálním stroji je izolováno, čímž chrání hostitelský systém před potenciálně škodlivými akcemi prováděnými v rámci virtuálního počítače.
- Flexibilita a pohodlí: Snadná migrace, kopírování a spouštění různých operačních systémů podle aktuálních potřeb.
- Snadné zálohování a obnovení: Umožňuje rychlé obnovení systémů do původního stavu, což je ideální pro testovací a vývojové účely.
- Kompatibilita: Virtuální stroje zajišťují konzistentní prostředí bez ohledu na fyzický hardware.
- Stabilita: Nehavaruje-li jeden virtuální stroj, nijak neovlivní ostatní provozované virtuální stroje ani hostitelský systém.
- Licencování: V určitých případech může umožnit efektivnější správu licencí softwaru.
- Podpora tenkých klientů: V kombinaci s výkonnými vzdálenými servery lze efektivně využívat tenké klienty, což snižuje celkové provozní náklady.

Výsledkem je, že virtuální počítače nabízejí moderním IT týmům výraznou flexibilitu a efektivitu, přičemž minimalizují rizika a zároveň zvyšují produktivitu. (Zerodaysnoop, 2020)

3.3.1.2 Rizika používání VirtualBoxu

VirtualBox technologicky separuje virtuální počítač od hlavního, hostujícího operačního systému. I když se VirtualBox používá právě se záměrem zamezení jakéhokoliv rizika na vlastním zařízení, přesto existují situace, kdy může interakce s virtuálním prostředím ohrozit bezpečnost hostitelského systému. (Das, 2022) Oracle VirtualBox funguje jako koordinátor virtuálních počítačů z více operačních systémů a může zlepšit výkon hostovaných virtuálních počítačů. Jedná se o bezplatný software s otevřeným zdrojovým kódem, který si může stáhnout každý. (Tight, 2023)

- Nedostatečná aktualizace virtuálního počítače: Přesto že se jedná o jednouchou věc, tak u virtuálních strojů dochází právě k opomíjení aktualizace. (Campbell, 2022) Ať již je na virtuálním stroji nainstalován Windows, Linux či jiný operační systém, pravidelné aktualizace jsou nezbytné pro jeho bezpečné fungování. Při záměru uchovávat konkrétní verzi operačního systému pro testování, stroj by se měl po dokončení testů buďto aktualizovat, nebo jej kompletně odstranit. (Das, 2022)
- Sdílení souborů mezi virtuálním a hostujícím systémem: Když je povolen přístup k souborům a složkám mezi těmito dvěma prostředími, zvyšuje se riziko ohrožení hlavního systému. (Das, 2022) Mnozí uživatelé často ve virtuálních strojích nemají nainstalovány antivirové řešení, což může při stahování souborů z neověřených zdrojů zvyšovat riziko zabezpečení. (Das, 2022) (Campbell, 2022)
- Instalace potenciálně nebezpečných aplikací ve virtuálním prostředí: Virtuální stroje často slouží k testování různého softwaru, který by uživatelé nemuseli chtít instalovat přímo na svůj primární systém. Pokud ale tato aplikace obsahuje zranitelnosti a sdílení souborů je povoleno, může to ohrozit i hostující operační systém. (Das, 2022)

Ačkoli VirtualBox poskytuje izolované prostředí pro virtuální stroje a většina uživatelů ho používá právě k účelům, při kterých nechtějí ohrozit svůj hostitelský systém, je nezbytné dbát na bezpečnostní opatření, aby se minimalizovalo potenciální riziko. (Das, 2022)

3.3.2 Kali Linux

Kali Linux, který byl dříve známý jako BackTrack Linux, je open-source linuxová distribuce založená na Debianu. Je specificky navržená pro pokročilé penetrační testy a audit informační bezpečnosti. Distribuce nabízí uživatelsky přívětivé nástroje, přednastavené konfigurace a automatizační skripty. (Wilson, 2023)

V rámci Kali Linuxu lze nalézt řadu specializovaných úprav a stovky specializovaných nástrojů, které pokrývají široké spektrum oblasti informační bezpečnosti – od penetračních testů, přes bezpečnostní výzkum, průzkumu sítí, identifikace a využití zranitelností až po forenzní analýzy a reverzní inženýrství, stejně jako nástroje pro správu zranitelností a testování v rámci Red Team akcí. (Singh, 2022 str. 38) V nabídce jsou různé nástroje, skripty a frameworky pro činnosti jako sběr informací o cílovém systému, skenování sítě, identifikace zranitelností nebo jejich využití. (Wilson, 2023)

Ačkoliv byl Kali Linux primárně navržen pro specialisty v oblasti kybernetické bezpečnosti, je vhodný i pro IT administrátory a profesionály zaměřené na síťovou bezpečnost. (Hertzog, a další, 2017) Tento systém je vybaven mnoha funkcemi a aplikacemi, které usnadňují práci penetračním testerům a bezpečnostním specialistům. (Singh, 2022 str. 38) Kali Linux je univerzální řešení dostupné a vhodné jak pro profesionály v oblasti kybernetické bezpečnosti, tak pro entuziasty a amatéry. (Wilson, 2023)

3.3.2.1 Funkce systému Kali Linux

Kali Linux je vybaven bohatou paletou nástrojů pro penetrační testování, která zahrnuje více než 600 specializovaných aplikací. Tento operační systém je navíc zcela bezplatný, což znamená, že uživatelé za něj nemusí platit. Jeho otevřený zdrojový kód je volně přístupný, což umožňuje každému provádět úpravy nebo přizpůsobení dle svých potřeb. (Wilson, 2023) Dále je třeba zdůraznit, že Kali Linux dodržuje standard Filesystem Hierarchy Standard (FHS), což usnadňuje uživatelům Linuxu lokalizaci binárních souborů, knihoven a dalších esenciálních komponent. Systém byl navržen s ohledem na kompatibilitu s co největším počtem bezdrátových zařízení, což zaručuje jeho funkčnost na širokém spektru hardware. Jádro Kali Linuxu je optimalizováno pro vstřikování a obsahuje aktuální záplaty pro tento účel. (Wilson, 2023) Bezpečnost je v centru pozornosti vývojového týmu, který pracuje v omezeném a zabezpečeném prostředí, čímž je zajištěna integrita a bezpečnost balíčků a repositářů. Vývojáři rovněž podepisují všechny balíčky pomocí protokolu GPG, což zvyšuje důvěru v jejich autenticitu. Kali Linux je multijazyčný a umožňuje uživatelům pracovat v jejich mateřském jazyce. Systém je navíc plně přizpůsobitelný, takže uživatelé mohou snadno upravit své rozhraní a nastavení dle vlastních preferencí. (Wilson, 2023)

Tento operační systém také nabízí podporu pro zařízení na platformách ARMEL a ARMHF, včetně oblíbených systémů jako Raspberry Pi a BeagleBone Black. Repositáře pro tyto platformy jsou integrovány přímo do hlavní distribuce, což zaručuje aktuálnost nástrojů. Kali

Linux navíc obsahuje nástroje a materiály vhodné pro vzdělávání v oblasti kybernetické bezpečnosti, díky čemuž je vhodný jak pro odborníky, tak pro začátečníky. (Wilson, 2023) Operační systém je pravidelně aktualizován s novými bezpečnostními opatřeními a funkcemi, což ho udržuje v čele průmyslového vývoje. Silná komunita za ním zaručuje přístup k rozsáhlým fórům, návodům a řešením. Uživatelé mají možnost instalovat Kali Linux na různá zařízení, od stolních počítačů a notebooků, přes mobilní zařízení, až po cloudové servery. Pro ty, kteří preferují virtuální prostředí, Kali Linux nabízí optimalizace a nástroje pro snadnou virtualizaci. Modulární struktura systému pak umožňuje uživatelům vybrat si, které aplikace a služby chtějí mít nainstalované. (Wilson, 2023)
Kali Linux rovněž poskytuje bezpečné prostředí pro testování a analýzu a minimalizuje tak riziko poškození cílových systémů. Uživatelé mohou navíc snadno integrovat tento systém s populárními bezpečnostními frameworky, jako jsou Metasploit, Nmap a další. Závěrem je třeba zdůraznit, že Kali Linux byl navržen s důrazem na etické hackování a kybernetickou bezpečnost, což uživatelům poskytuje nástroje a praxe v souladu s právními normami a profesní etikou. (Wilson, 2023)

3.3.2.2 Oblíbenost Kali Linux mezi hackery

Distribuce Kali Linux je v bezpečnostní komunitě vysoce ceněna díky svému specializovanému zaměření. Nabízí širokou škálu nástrojů pro penetrační testování, bezpečnostní analýzu, počítačovou forenziku a reverzní inženýrství. Vývoj systému Kali Linux probíhá v maximálně zabezpečeném prostředí a je svěřen omezenému počtu ověřených jednotlivců. Ti mají oprávnění přidávat a aktualizovat softwarové balíčky. Každý z těchto balíčků je navíc digitálně podepsán svým vývojářem pro zajištění autenticity. (Techspot, 2023)

3.3.3 DVWA

Damn Vulnerable Web App zkráceně DVWA je webová aplikace vytvořená v PHP/MySQL s cílem být záměrně zranitelná. Byla navržena primárně pro vzdělávací účely v oblasti kybernetické bezpečnosti. (Bisson, 2023) Tato platforma je ideálním nástrojem pro specialisty v oblasti kybernetické bezpečnosti a hackery, kteří chtějí procvičit a rozšířit své dovednosti v bezpečném a legálním prostředí. (Comptia, 2020) Umožňuje specialistům v oblasti bezpečnosti testovat a zdokonalovat své dovednosti v legálním a kontrolovaném prostředí. Navíc je nástrojem pro webové vývojáře k lepšímu pochopení bezpečnostních

praxí a zabezpečení webových aplikací. (Ken, 2023) Vzdělávacím institucím poskytuje nástroj pro výuku zabezpečení webových aplikací. (Garcia, 2023)

DVWA byla navržena tak, aby pokryla širokou škálu webových zranitelností s různými úrovněmi obtížnosti, a to prostřednictvím uživatelsky přívětivého rozhraní. DVWA obsahuje jak zdokumentované, tak nezdokumentované zranitelnosti, což je záměrné, aby uživatelé mohli prohloubit své dovednosti v detekci a nápravě. (Garcia, 2023) Obsahuje různé typy zranitelností s různými stupni obtížnosti, což uživatelům umožňuje postupně zvyšovat úroveň svých schopností. Primárním cílem je poskytnout bezpečnostním profesionálům platformu pro testování a zdokonalení jejich technik a nástrojů, zatímco webovým vývojářům pomáhá lépe rozumět a implementovat bezpečné postupy při vývoji webových aplikací. (Bisson, 2023)

Přestože je DVWA užitečným vzdělávacím nástrojem a je záměrně zranitelná, neměla by tedy být hostována na veřejně dostupných serverech nebo v jakémkoli prostředí s přístupem k internetu, aby se předešlo reálným bezpečnostním rizikům. Nejlepším postupem je provozování DVWA v izolovaném virtuálním prostředí, například v nástrojích jako VirtualBox či VMware, s nastavením na interní síťový režim. (Garcia, 2023)

3.3.3.1 Funkce DVWA

- DVWA poskytuje rozhraní pro simulaci různých typů webových zranitelností s různou úrovní obtížnosti. (Tse, 2021)
- Řeší zranitelnosti, jako je SQL Injection, kdy útočníci mohou manipulovat s příkazy SQL a získat tak přístup k neautorizovaným datům. (Tse, 2021)
- DVWA má také funkce související s ověřováním a správou relací, které jsou často náchylné k chybám, což činí systémy náchylnými ke kompromitaci. (Whittly, 2020)
- Aplikace nabízí také vizuální průvodce, protože je k dispozici diagram znázorňující, jak spolu jednotlivé funkce komunikují. (Whittly, 2020)

3.4 Nástroje pro etický hacking

Hackerský software a nástroje představují soubor počítačových aplikací a skriptů vyvinutých k identifikaci a využití slabých míst v operačních systémech, webových aplikacích, serverech a sítích. V dnešní době řada organizací využívá nástroje etického hackingu k posílení své kybernetické obrany a ochraně dat před potenciálními hrozbami. Tyto nástroje mohou být dostupné jako open-source řešení nebo jako komerční produkty. Ačkoli mohou být legálně využity pro defenzivní účely, mohou být také zneužity k škodlivým útokům. (Simplilearn, 2023) Nástroje pro penetrační testování jsou softwarové aplikace a skripty určené k identifikaci a využití zranitelností v operačních systémech, webových aplikacích, serverech a síťových infrastrukturách. (Williams, 2023)

Na trhu existuje široká škála nástrojů, které umožňují uživatelům provádět etické hackování a ověřovat zabezpečení svých systémů. Zatímco některé z těchto nástrojů jsou nabízeny jako open-source řešení, dostupné k bezplatnému stáhnutí a úpravám, jiné jsou komerční produkty poskytující specifické vlastnosti a podporu. (Williams, 2023) Bezpečnostní experti často využívají nástroje etického hackingu k analýze a zabezpečení IT infrastruktury. Zatímco původně byly tyto systémy primárně určeny pro monitorování sítě, nyní mohou být použity i pro řízení firewallových řešení, systémů detekce průniků (IDS), VPN, antivirových programů a filtrů proti spamu. (Simplilearn, 2023)

Mezi nejznámější nástroje v oblasti kybernetické bezpečnosti patří Nmap, Nessus, Nikto, Kismet, NetStumbler, Acunetix, Netsparker, Intruder, Metasploit Aircrack-Ng a další. (Simplilearn, 2023)

3.4.1 Volně dostupné nástroje pro etický hacking

Mnoho nástrojů, které kybernetičtí specialisté využívají, jsou dostupné zdarma v rámci open-source licencí. K provádění většiny penetračních testů a útoků není nutná vysokovýkonná hardwarová infrastruktura s vysokými náklady. Obyčejný notebook či stolní počítač s adekvátním množstvím paměti RAM, úložištěm a dostatečně rychlým procesorem může často plně postačovat. Historie kybernetické bezpečnosti je plná příběhů o jednotlivcích, kteří s minimálním rozpočtem dosáhli významných úspěchů, ačkoli se každý musí sám rozhodnout, jakou kombinaci hardwaru a softwaru potřebuje pro své konkrétní cíle. (Norton, 2018 str. 24)

3.4.2 Burp Suite

Burp Suite je přední nástroj v oblasti zabezpečení webových aplikací vytvořený firmou PortSwigger, který je široce využíván specialisty na penetrační testování a experty v oblasti kybernetické bezpečnosti. (Portswigger, 2023) Jedná se o aplikaci založenou na proxy serveru, což umožňuje uživatelům monitorovat a manipulovat komunikací mezi webovým prohlížečem a cílovým webovým serverem. (Singh, 2022 str. 593)

V praxi penetrační tester používá Burp Suite jako prostředek pro průhledné zachytávání komunikace z webového prohlížeče. Tím se otevírá možnost upravovat různé části požadavků předtím, než jsou směrovány k cílové webové aplikaci, což umožňuje odhalení a využití potenciálních zranitelností. (Singh, 2022 str. 593) Tato platforma kombinuje několik nástrojů, které pokrývají celý spektrum testovacího procesu od prvotní analýzy a mapování aplikace po identifikaci a využití potenciálních zranitelností. (Portswigger, 2023)

Vývoj Burp Suite započal v roce 2004, kdy Dafydd Stuttard identifikoval potřebu vysoce efektivního nástroje pro penetrační testování webových aplikací. Během následujících 16 let prošel tento software výraznou evolucí, rozšířil své funkce a stal se esenciálním nástrojem v arzenálu specialistů na kybernetickou bezpečnost. Dnes je Burp Suite uznáván jako přední nástroj pro testování zabezpečení webových aplikací, a co víc, byl rozšířen o funkce, které umožňují analyzovat zranitelnosti v API a mobilních aplikacích. (Rahalkar, 2021 str. 5)

Sada Burp Suite, přední nástroj pro testování zabezpečení webových aplikací, nabízí tři různé edice přizpůsobené různým potřebám uživatelů: Community Edition pro bezplatné využití, profesionální verze pro expertní penetrační testování a podniková edice nabízející automatizované skenování a možnost integrace do CI/CD procesů. (Rahalkar, 2021 str. 5) Díky svým rozsáhlým schopnostem je Burp Suite často označován za průmyslový standard v oblasti penetračních testů. (Portswigger, 2023). Různé varianty Burp Suite pokrývají širokou škálu potřeb – od nováčků v oboru, přes odborníky až po korporátní segment. Tím je zajištěno, že každý uživatel získá nejlepší dostupné nástroje pro svoje specifické potřeby v oblasti testování zabezpečení webových aplikací. (Rahalkar, 2021 str. 8)

3.4.2.1 Community Edition

Burp Suite Community Edition je bezplatná verze nabízející základní nástroje pro penetrační testování webových aplikací. Tato edice je ideální pro ty, kteří se teprve začínají seznamovat s oblastí zabezpečení aplikací. Poskytuje uživatelům základní prostředky pro vstup do světa

testování, včetně proxy serveru pro monitorování komunikace, opakovače pro upravení a odeslání požadavků a nástrojů pro kódování či dekódování dat. (Rahalkar, 2021 str. 7)
Sada Burp Suite funguje jako webový proxy server. Balíček pracuje s webovým prohlížečem a penetrační tester zachycuje provoz mezi webovým serverem a prohlížečem. Všechny tři prvky mohou být umístěny na stejném počítači. (Cooper, 2023)
Community Edition nabízí hackerům a ostatním uživatelům snadný přístup k funkcím pro analýzu a výzkum systému a umožňuje jim efektivně organizovat své pracovní postupy. Uživatelé mají také možnost kopírovat relevantní data z výzkumné obrazovky do modulu pro provedení útoku. Mezi klíčové nástroje patří: (Jethva, 2023)

- Proxy: Engine zodpovědný za veškerý výzkum a přípravu scénářů útoku. Má integrované procesory pro přesměrování síťového provozu a provádění analýz.
- Repeater: Umožňuje uživatelům injektovat provoz do sítě a následně testovat aplikace na přítomnost slabých míst. Umožňuje také vytváření a modifikaci HTTP hlaviček dle specifikace.
- Decoder: Slouží pro dekódování šifrovaných dat a převod zdrojových dat do požadovaného formátu.
- Sequencer: Analyzuje shromážděná data s cílem detekovat náhodnost, kontroluje vzor a hodnotu každé varianty v testovacích strategiích.
- Comparer: Využíván k rozlišení a srovnání odpovědí, které mohou být komplexní k dekódování. (Jethva, 2023)

3.4.2.2 Professional Edition

Burp Suite Professional Edition je pro odborníky specializující se na zabezpečení webových aplikací. Tato edice nabízí rozsáhlou sadu pokročilých nástrojů, které výrazně zvyšují efektivitu odhalování zranitelností. (E-spin, 2021)Je to ideální řešení pro profesionály, kteří vyžadují sofistikované prostředky jak pro manuální, tak pro automatizované penetrační testy. K pokročilým funkcím patří testování zranitelností, možnost automatizovaného skenování zranitelností a integrovaná rozšíření, sofistikované metody prolamování ochran, fuzzing, automatická generace exploitů pro CSRF, clickjacking a další. (Rahalkar, 2021 stránky 7-8)

3.4.2.3 Enterprise Edition

Burp Suite Enterprise Edition je primárně zaměřena na korporátní sektor místo jednotlivých specialistů. Tato verze je klíčová pro integraci bezpečnostního skenování do vývojových pipelinů a je navržena pro organizace, které implementují DevSecOps přístup. Ačkoliv nemusí mít stejnou šíři manuálních testovacích nástrojů jako profesionální edice, její hodnota spočívá v poskytování automatizovaných řešení pro zabezpečení vývojových procesů. (Rahalkar, 2021 str. 8)

3.4.3 Sqlmap

SQLmap je renomovaný open-source nástroj určený k penetračnímu testování, který se zaměřuje zejména na identifikaci a využívání zranitelností SQL injection ve webových aplikacích. Pochází z jazyka Python a je vytvořen tak, aby pomáhal při odhalování těchto zranitelností a jejich případném zneužití. (Choudhary, 2023)

Hlavním cílem nástroje SQLmap je automatizovat náročný proces vyhledávání a zneužívání slabých míst SQL injection ve webových aplikacích. Jeho automatizační schopnosti zefektivňují proces identifikace, čímž se stává neocenitelným přínosem pro testery, jejichž cílem je určit zranitelnost webové aplikace vůči SQL injection. (Fagun, 2023)

Z funkčního hlediska je SQLmap velmi všestranný nástroj. Dokáže ověřit, zda v systému existují zranitelnosti typu SQL injection. Kromě toho je schopen získat důležité údaje, jako jsou názvy databází, tabulek a sloupců, a v některých případech dokonce i přístupové pověření. Pokud jsou zjištěné zranitelnosti závažné, SQLmap může dosáhnout úplného přístupu do systému. (Singh, 2018)

3.4.4 Commix

Commix je nástroj, zaměřený na identifikaci a využití zranitelností spojených s injekcí příkazů, jež často vznikají ve webových stránkách a aplikacích. Tato konkrétní zranitelnost typicky postihuje webové aplikace a její účinnost spočívá v odhalování slabých míst v bezpečnosti. (Shariq, 2022)

Commix je vytvořený v programovacím jazyce Python a disponuje rozmanitými moduly, jež umožňují uživatelům aktivně vyhledávat potenciální slabiny v cílových aplikacích. Pro útok na danou URL využívá Commix různé strategie, a to jak pomocí datových řetězců, tak i skrze HTTP hlavičky či cookies, a to včetně útoků na autentizační parametry. V rámci

Commixu mají uživatelé přístup k rozmanitým možnostem enumerace, což zvyšuje jeho efektivitu při analýze bezpečnosti. (Shariq, 2022)

Jedním z významných prvků Commixu jsou dvě odlišné techniky injekce příkazů, přičemž první z nich využívá výsledky, zatímco druhá pracuje s tzv. slepou technikou injekce příkazů (BlindSQL). Tato kombinace technik poskytuje uživatelům široké možnosti při zkoumání a využívání potenciálních bezpečnostních chyb. (Stasinopoulos, 2023) Commix, jako open-source nástroj, je volně dostupný na platformě GitHub, což podporuje komunitní přístup k jeho vývoji a zajišťuje transparentnost a sdílení znalostí v oblasti kybernetické bezpečnosti. (Stasinopoulos, 2023)

3.5 Testovací scénáře

Ve světě softwarového testování je klíčové rozumět pojmu "testovací scénář". Testovací scénář obsahuje soubor testovacích případů, ať již manuálně prováděných nebo automatizovaných, který pomáhá vyhodnotit funkčnost a spolehlivost daného softwarového řešení. Poskytuje tak roadmapu pro týmy nebo uživatele odpovědné za zajištění kvality, kterou by měli sledovat při ověřování softwaru. (Qamadness, 2023)

Primárním účelem testovacího scénáře je zkontrolovat, jak celý systém reaguje na různé uživatelské vstupy a situace. Tester má za úkol vcítit se do role koncového uživatele, aby tak mohl identifikovat a předvídat reálné situace, s nimiž se software může setkat po svém nasazení v provozu. (Qamadness, 2023)

3.5.1 Scénářové penetrační testování

Penetrační testování je často omezeno na specifické IT aktiva, například webové či mobilní aplikace, a nezaměřuje se celkově na IT infrastrukturu organizace. Tím pádem nemůže poskytnout komplexní pohled na veškeré bezpečnostní rizika, která mohou vznikat z různých směrů. (ECQ, 2023) Penetrační testovací scénáře jsou navrženy k simulaci reálných útoků na IT systémy či sítě s cílem odhalit potenciální zranitelnosti. Při přípravě takového testování je nezbytné nejprve stanovit rozsah a hlavní cíle, specifikovat zaměření na konkrétní systémy a rozhodnout o metodologii testování. Tester poté sbírá relevantní informace o testovaném cíli, což mu pomáhá lépe rozumět jeho fungování a možným slabým místům. (ptsecurity, 2017)

Scénářový penetrační test se liší od tradičních penetračních testů. Zatímco tradiční testy se typicky soustředí na identifikaci zranitelností v konkrétním a omezeném kontextu IT prostředí, scénářové penetrační testy pokrývají širší spektrum organizace a zaměřují se na hodnocení odolnosti a připravenosti organizace vůči útokům, jak by se mohly vyskytnout v reálném světě. (securify, 2022) Služba scénářového penetračního testování, kterou nabízí společnosti řeší tento nedostatek tím, že simuluje útoky jak z perspektivy vnějších útočníků, tak z pohledu interních či nepoctivých uživatelů. Využívá přitom metod a postupů skutečných kybernetických hrozeb a sofistikovaných útočníků. Scénářové penetrační testy tak poskytují realistický pohled na možné útoky, kdy jsou simulace prováděny dle specifických scénářů a jsou efektivnější z hlediska časové náročnosti. (ECQ, 2023)

Na základě definovaného rozsahu a cílů testování lze poté vytvářet konkrétní testovací případy. Ty mohou simulovat různé typy útoků – od využití již známých zranitelností po

hledání nových slabých míst pomocí technik jako je fuzzing či brute-force útoky. Specifické testovací scénáře jsou závislé na cílech penetračního testování a charakteristikách testovaného systému. Například testování webové aplikace může obsahovat scénáře simulující útoky typu SQL injection, cross-site scripting nebo zneužití chyb v nahrávání souborů. (ptsecurity, 2017)

Při využití red teamu se simuluje postup konkrétního kybernetického útočníka nebo skupiny útočníků, aplikující se jejich specifické taktiky a metody. V rámci scénářového penetračního testu je možné se zaměřit na specifický útokový scénář, přičemž může dojít k přeskočení některých kroků. Například, pokud je cílem prověřit zranitelnost konkrétního serveru, nemusí se provádět phishingový útok, ale test může začít přímo zkoumáním zabezpečení tohoto serveru. (securify, 2022)

3.5.2 Výhody testování podle scénářů

Ve srovnání s tradičními penetračními testy, které se primárně zaměřují na identifikaci co nejvíce zranitelností, je scénářová metodologie testování orientována na hodnocení a srovnání efektivity kontrolních mechanismů v oblasti kybernetické bezpečnosti. (honeyteksystems, 2023) Scénářové penetrační testování je zaměřený přístup k ofenzivnímu hodnocení bezpečnosti. (Redscan, 2022)

Na rozdíl od standardního penetračního testování, které se orientuje na identifikaci zranitelností, se scénářové testování koncentruje na simulační hodnocení efektivity bezpečnostních opatření vůči specifickým taktikám a postupům potenciálního útočníka. (Redscan, 2022)

Testování na bázi scénářů je pokročilá a specializovaná metoda ofenzivního hodnocení kybernetické bezpečnosti s klíčovými otázkami, na které se snaží najít odpovědi (honeyteksystems, 2023):

- Jak úspěšně naše bezpečnostní opatření čelí, detekují a reagují na hrozby?
- Existují v našem zabezpečení oblasti, které by mohly být terčem zkušených útočníků?
- Mohou naši bezpečnostní experti z týmu Blue Team efektivně odolat pokročilým hrozbám?
- Dokážou naši analytici správně identifikovat autentické bezpečnostní incidenty a odlišit je od falešných poplachů?
- Jsou vytvořeny a správně aplikovány postupy reagování na bezpečnostní události?

- Disponují naše interní bezpečnostní týmy potřebnými znalostmi a dovednostmi pro řešení a nápravu incidentů? (Redscan, 2022)
- Jaká je účinnost a správná konfigurace našich monitorovacích systémů ve vztahu k detekci a prevenci?
- Existují v našem zabezpečení slabá místa, jež by mohli vytrvalí útočníci využít?
- Disponují analytici z "modrého týmu" potřebnými schopnostmi k identifikaci a odražení pokročilých útoků?
- Dokážou bezpečnostní analytici spolehlivě rozeznat mezi skutečnými hrozbami a falešnými poplachy?
- Existují a jsou řádně implementovány postupy reakce na incidenty, zahrnující eskalaci detekovaných hrozeb a řešení možných narušení?
- Mají vnitřní bezpečnostní týmy esenciální znalosti a dovednosti pro řešení a nápravu incidentů? (honeyteksystems, 2023)

3.5.3 Tvorba

Pro efektivní sestavení testovacích scénářů je zásadní důkladná analýza a porozumění všem relevantním dokumentům, včetně funkčních a technických specifikací. (Coursera, 2023) S detailním přehledem požadavků lze identifikovat klíčové funkce a potenciální uživatelské interakce podle specifikací. Tyto izolované funkce tvoří základ pro další vytváření testovacích scénářů. (Fruhlinger, 2021)

Vizuální reprezentace aplikace poskytuje jasnější kontext, což může výrazně podpořit tvorbu komplexnějších a více krycích testovacích scénářů. Takový přístup zároveň zajišťuje, že každý sestavený testovací scénář odpovídá specifikovaným požadavkům. (Olaogun, 2023)

3.6 Komparace

Komparativní neboli srovnávací metoda, představuje výzkumnou strategii zaměřenou na analýzu různých aspektů různých jevů za účelem identifikace podobností, rozdílů a zákonitostí. Aplikuje se k ověřování hypotéz, charakterizaci jevů, klasifikaci, typologizaci a zkoumání historických a kontextuálních proměn. Efektivní využití této metody předpokládá teoretickou a metodologickou způsobilost. (Linhart, a další, 2017)

Podle autora článku software testing je pro testera jakéhokoliv softwaru nejdůležitější zajistit, aby výsledky testů odpovídaly očekávanému chování systému. Faktory, jako jsou konfigurace, data, zátěž a závislosti, však mohou výsledky zkreslit. (2023) Zpráva o testu zobrazuje shrnutí zprávy a výsledky porovnání se zvýrazněním úspěšných a neúspěšných testovacích případů. (Oracle, 2016)

3.6.1 Postup komparace

- Stanovení cílů testování: Před zahájením je nezbytné zdokumentovat cíle. Je nutné si položit otázky. Jaký je hlavní cíl testů? Mělo by být možné určit funkční i nefunkční požadavky. Kromě toho je zásadní stanovit metriky kvality a výkonnosti. (Software Testing, 2023)
- Základní a srovnávací úroveň: Aby bylo možné provádět smysluplná hodnocení, mělo by se stanovit základní linie, která představuje požadovaný stav systému na základě požadavků. (Software Testing, 2023) Použití výchozího i srovnávacího stavu pomůže přesně určit nesrovnalosti a zjistit jejich příčiny. (Coursera, 2023)
- Využití vhodného nástroje: Volba nástrojů, které souzní s cíli testování, může být přínosná. Tyto nástroje, ať už pro automatizaci testování, analýzu dat nebo reporting, by měly napomoci efektivnímu sběru a interpretaci výsledků testování. Takové nástroje nejen zefektivňují proces testování, ale také nabízejí využitelné poznatky a zajišťují kvalitu aplikace. (Software Testing, 2023)
- Konzistentní testovací dat: Variabilita testovacích dat a případů může způsobit značné nesrovnalosti ve výsledcích. Proto je zásadní udržovat konzistenci testovacích dat a případů v testovacím i produkčním prostředí. Tato konzistence zajišťuje, že získané výsledky jsou srovnatelné, a snižuje možnost vzniku anomálií. (Software Testing, 2023)

- Kontrolní testovací proměnné: Různé prvky, od konfigurace systému a síťových podmínek až po chování uživatelů, mohou zásadně ovlivnit výsledky testů. Je nezbytné zajistit, aby v testovacím i produkčním prostředí byly zachovány podobné podmínky. Tím lze omezit vliv vnějších rušivých vlivů a zajistit, že výsledky testů zůstanou konzistentní. (Software Testing, 2023)

4 Vlastní práce

V praktické části se diplomová práce zaměřuje na podstatu etického hackingu a ochrany webových aplikací. Po teoretickém pozadí a odborném literárním přezkumu z předchozích částí následují konkrétní kroky a postupy, které byly definovány v zadání práce.

Začíná se vytvořením specifikovaného testovacího prostředí, což je klíčový krok pro zajištění důkladných a standardizovaných výsledků. Následně se zabývá procesem výběru a návrhu testovacích scénářů, které byly zvoleny na základě identifikovaných zranitelností aplikace a relevantní literatury. Dále se práce zaměřuje na aplikaci těchto scénářů, konkrétní provedení testů a jejich následné vyhodnocení. Na základě získaných dat jsou navrženy způsoby, jakými lze webové aplikace nejlépe chránit před rozpoznanými útoky.

V závěrečné fázi se nachází komplexní zhodnocení různých způsobů ochrany webových aplikací, aby bylo možné vybrat a doporučit nejefektivnější metody ochrany. Toto hodnocení poskytuje pevný základ pro závěrečné doporučení, které vychází z naměřených výsledků a provedených testů.

4.1 Tvorba prostředí

Začíná se výběrem vhodného virtuálního prostředí, kde volba padla na VirtualBox. Toto rozhodnutí je odůvodněno několika klíčovými faktory, včetně jeho uživatelské přívětivosti, kompatibility a schopnosti vytvářet izolované virtuální stroje. V této části jsou také základní informace o tomto nástroji a jeho významu pro testy.

Následuje instalace a konfigurace operačního systému Kali Linux. Tento OS je zvolen pro jeho širokou škálu nástrojů pro penetrační testování a zabezpečení. Klíčové nastavení operačního systému je detailně popsáno, aby bylo zajištěno, že vše bude fungovat optimálně pro testovací účely. Dále tato část popisuje proces instalace a nastavení testovací aplikace DVWA. Tato aplikace je zvolena pro její specializaci na etické hackingové testy a je detailně nastavena v rámci Kali Linuxu pro optimální výkon.

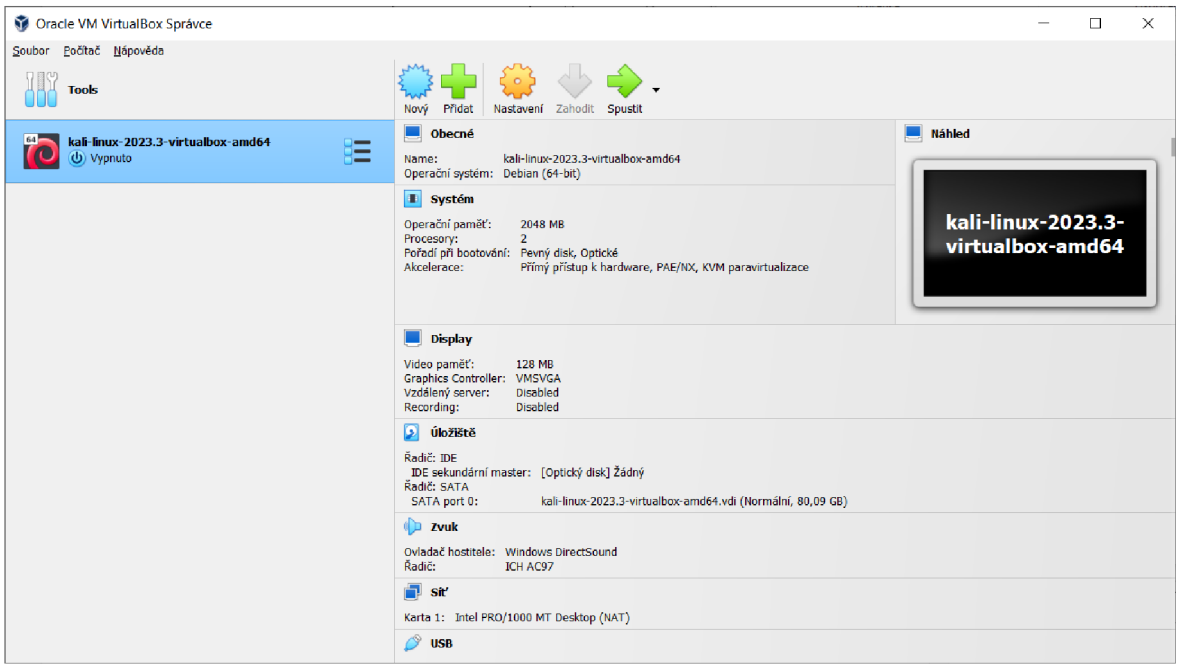
V závěrečné části jsou důležité aspekty zabezpečení testovacího prostředí. Je nezbytné zajistit, aby hostitelský počítač zůstal v bezpečí při provádění těchto testů, a proto jsou zde popsány kroky, které byly podniknuty k zajištění tohoto bezpečí.

4.1.1 Volba virtuálního prostředí

Jak bylo zmíněno v teoretické části, VirtualBox je oblíbený nástroj pro virtualizaci, který je navíc open-source. Pro účely etického hacking a testování je virtualizace obzvláště vhodná. VirtualBox poskytuje izolované prostředí, což znamená, že všechny akce prováděné v rámci virtuálního stroje nemají vliv na hlavní operační systém. Toto je pokládáno za klíčové při práci s potenciálně škodlivým kódem nebo při testování nebezpečných aplikací. Verze VirtualBoxu během testování byla verze 7.0.10, jehož rozhraní lze vidět na obrázku 1.

Pokud se něco pokazí během testování, VirtualBox umožňuje snadnou obnovu virtuálního stroje do původního stavu. Toto je ideální pro experimentování, protože pak mizí obavy z trvalých následků. Kromě toho VirtualBox podporuje širokou škálu operačních systémů a je dostupný na mnoha platformách, což z něj činí univerzální řešení.

Oproti jiným virtuálním emulátorům, jako jsou VMware nebo Microsoft Hyper-V, nabízí VirtualBox jednoduchost a uživatelskou přívětivost díky svému intuitivnímu rozhraní. Dále má za sebou silnou online komunitu, která je připravena poskytnout pomoc, návody nebo řešení různých problémů. A konečně, významnou výhodou je skvělá integrace s Kali Linuxem, což je jedním z nejoblíbenějších nástrojů pro etický hacking. VirtualBox nabízí předpřipravené obrazy pro tento operační systém, což usnadňuje jeho instalaci a nastavení. Vzhledem k všem těmto vlastnostem a výhodám je VirtualBox často preferovanou volbou pro etické hackery a výzkumníky v oblasti bezpečnosti.



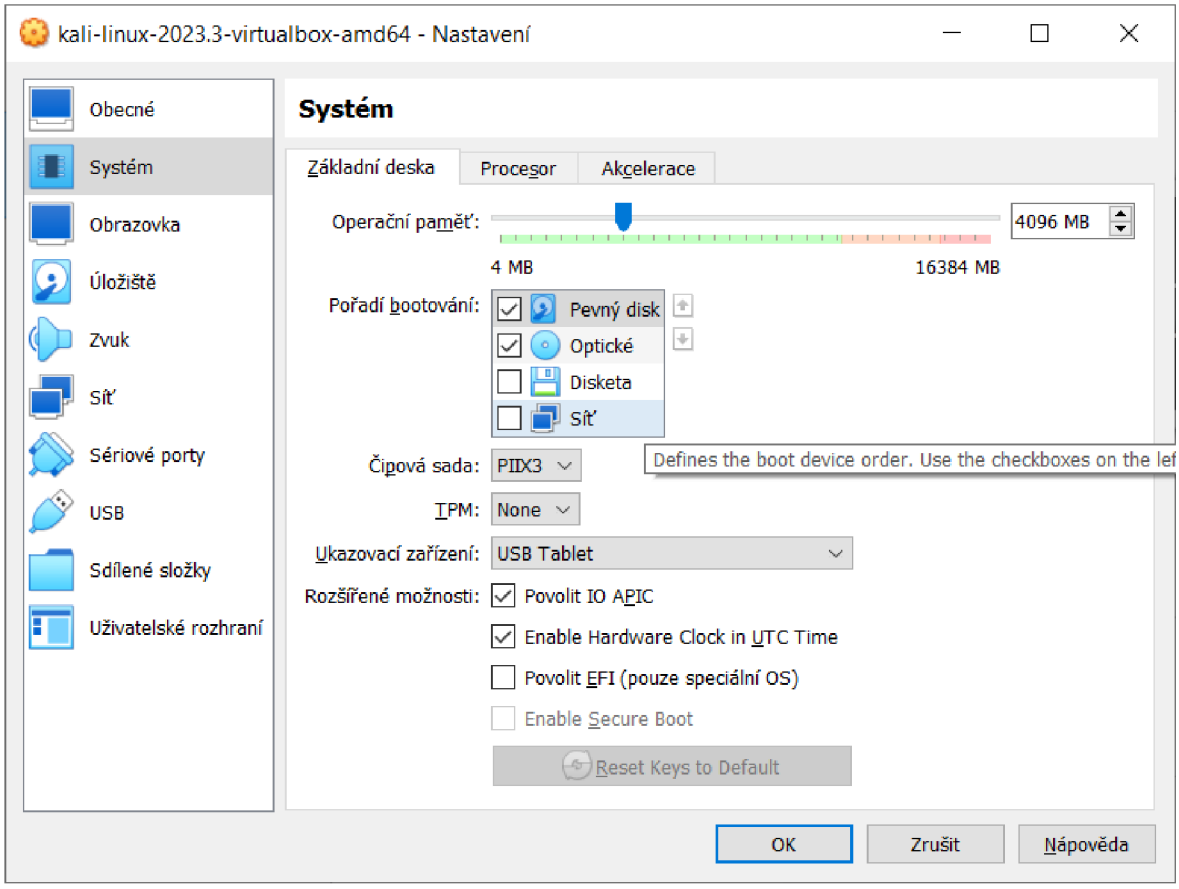
Obrázek 1: Prostředí VirtualBoxu, Zdroj: Vlastní

4.1.2 Instalace a konfigurace operačního systému Kali Linux

V rámci diplomové práce bylo rozhodnuto využít operační systém Kali Linux, konkrétně verzi 2023.3. Volba tohoto operačního systému byla motivována jeho rozsáhlým souborem nástrojů specializovaných na bezpečnostní testování a penetrační zkoušky, což jej činí ideálním kandidátem pro účely této diplomové práce. Podrobnější informace o charakteristikách a výhodách Kali Linuxu jsou popsány v teoretické části práce.
Správná instalace a konfigurace operačního systému jsou klíčové pro dosažení optimálního výkonu a bezpečnosti, zejména v kontextu testování webových aplikací a analyzování zranitelností. V následujících krocích je popsán postup instalace a doporučené nastavení pro Kali Linux 2023.3 ve virtuálním prostředí.

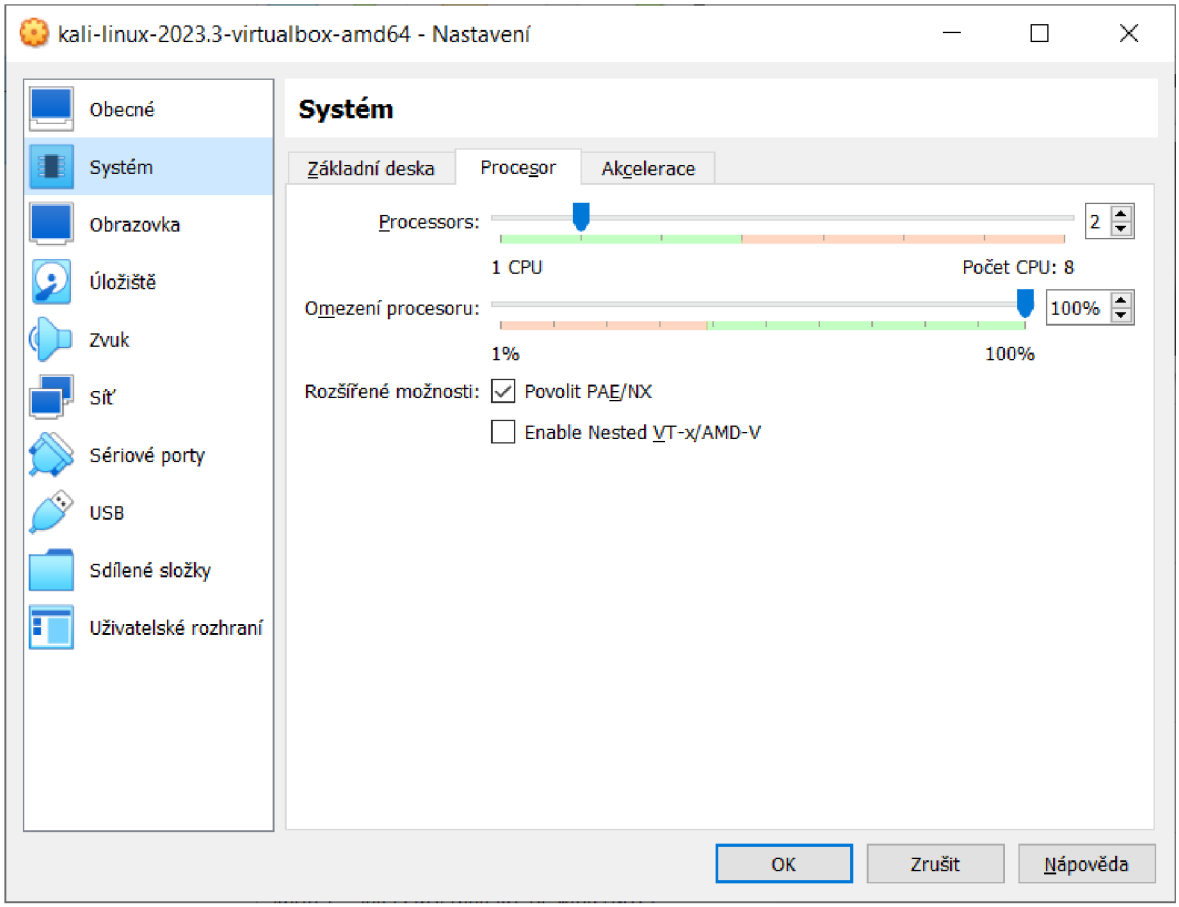
4.1.2.1 Konfigurace

Při konfiguraci virtuálního prostředí je jedním z klíčových parametrů alokace paměti RAM. Pro optimální výkon operačního systému Kali Linux je důležité zohlednit dostupné množství paměti RAM v hostitelském počítači. Oficiální dokumentace Kali Linuxu doporučuje pro základní provoz alespoň 1 GB RAM. Avšak pro komplexní bezpečnostní analýzy a výkonnější operace je vhodné alokovat větší množství paměti, v případě mého počtu RAM 16 GB je ideální alokovat až 4 GB viz. obrázek 2.
Pokud se během provozu objeví známky nedostatečné paměti, jako je zpomalený výkon nebo nestabilita systému, doporučuje se přidělit více RAM. Nicméně je důležité dbát na to, aby bylo stále dostatečné množství paměti ponecháno pro hostitelský operační systém, aby nedošlo k jeho zpomalení nebo nestabilitě.



Obrázek 2: Upravení množství paměti RAM, Zdroj: Vlastní

Dalším krokem v optimalizaci nastavení virtuálního prostředí je konfigurace procesorových jader. I když Kali Linux je schopen běžet stabilně i na jednom procesorovém jádře, pro náročnější úlohy a hladký multitasking je výhodné přidělit více jader. Tímto nastavením lze zajistit, že systém bude reagovat pružněji na více vláknové operace a zlepší se celkový výkon. Nastavení procesoru lze vidět na obrázku 3.



Obrázek 3: Upravení množství procesoru, Zdroj: Vlastní

Po úspěšném startu operačního systému Kali Linux systém, jehož rozhraní lze vidět na obrázku 4, vyzve k zadání uživatelského jména. Výchozí uživatelské jméno pro tento operační systém je "kali". Heslo je na tom obdobně, tedy opět "kali". Toto výchozí nastavení je určeno pro základní používání, avšak pro běžný provoz je důrazně doporučeno změnit heslo z důvodů bezpečnosti.

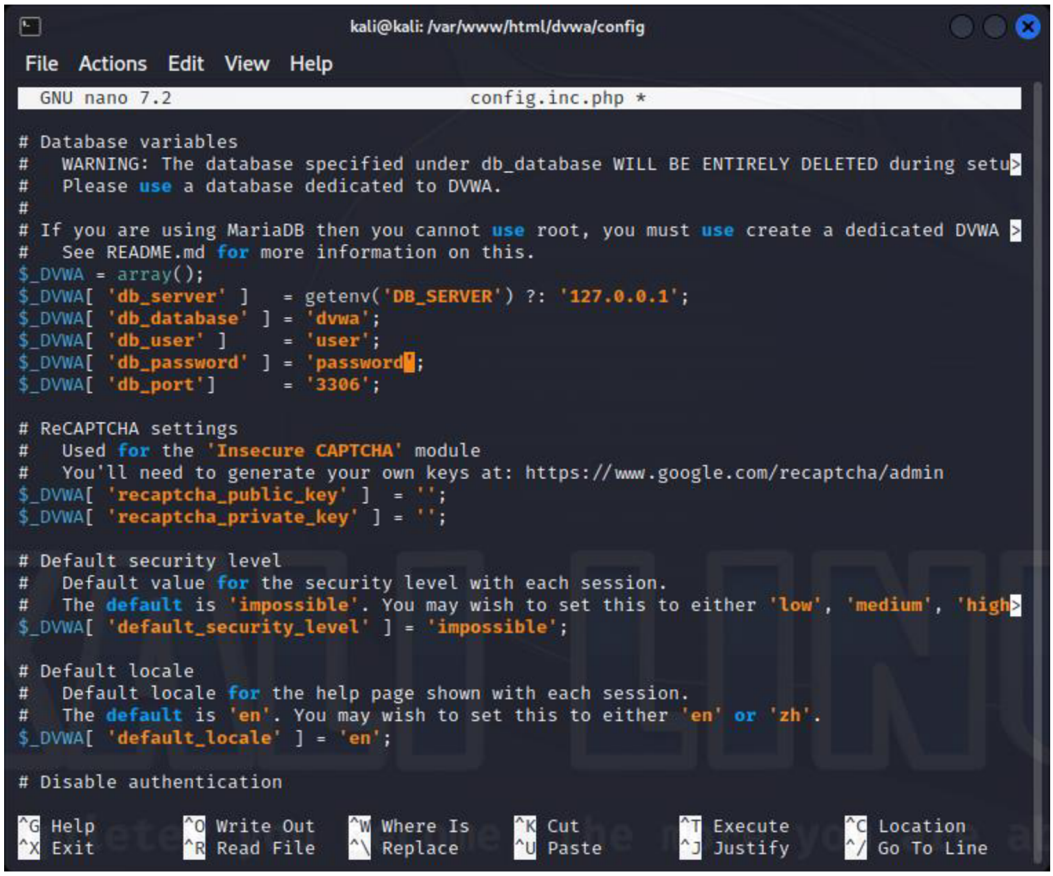
Obrázek 4: Operační systém Kali Linux, Zdroj: Vlastní

4.1.3 Testovací aplikace

Inicializace projektu probíhala stahováním zdrojového kódu aplikace DVWA z jejího oficiálního repositáře na GitHubu. K tomu byl využit systém Git s příkazem git clone ve znění: https://github.com/digininja/DVWA.

Následovala alokace přístupových práv pro složku DVWA za účelem zajištění adekvátních operací nad soubory. Toto bylo provedeno terminálovým příkazem chmod -R 777 dvwa/.

Konfigurační soubor config.inc.php byl následně upraven tak, aby reflektoval specifické parametry. Především specifikování uživatelského jména a hesla pro aplikaci, jak je vidět na obrázku 5. V tomto případě byl použit příkaz: sudo nano config.inc.php.



Obrázek 5: Nastavení přihlašovacích údajů

Následoval start mysql příkazem: sudo service mysql start. Chápaje architektonickou závislost DVWA, bylo imperativní nainstalovat webový server Apache spolu s databázovým serverem MariaDB. K tomuto účelu bylo využito systému apt.

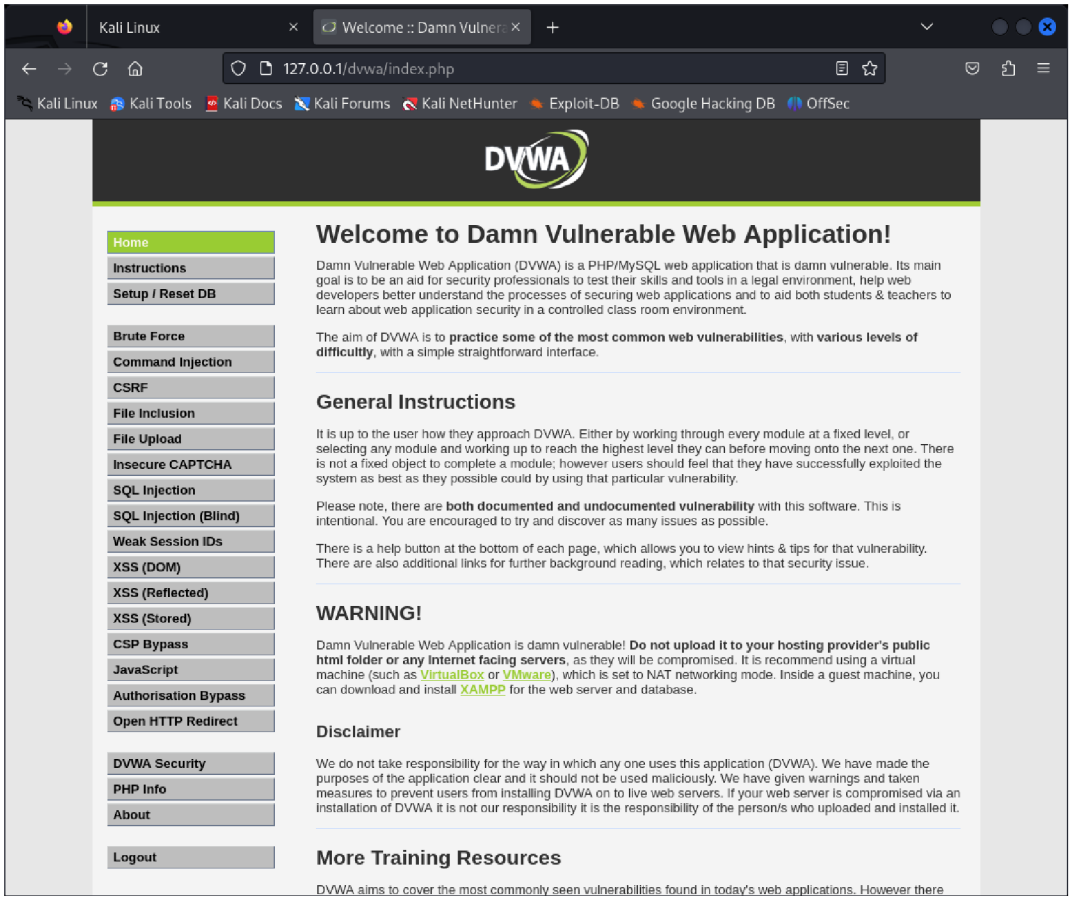
Pro inicializaci a konfiguraci databáze MySQL bylo nutné službu spustit, což se provedlo standardním způsobem. Po spuštění byli provedeny administrativní změny v databázi s cílem připravit prostředí pro DVWA.

Vzhledem k specifickým požadavkům na PHP v kontextu DVWA bylo potřeba přidat surry php ppa repozitář. To umožnilo optimalizovat a rozšířit funkcionalitu PHP pro tento projekt. Příkazy byly použity v následujícím znění a pořadí:

```
sudo apt update
sudo apt -y install lsb-release apt-transport-https ca-certificates
sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
echo "deb https://packages.sury.org/php/ buster main" | sudo tee /etc/apt/sources.list.d/php.list
```

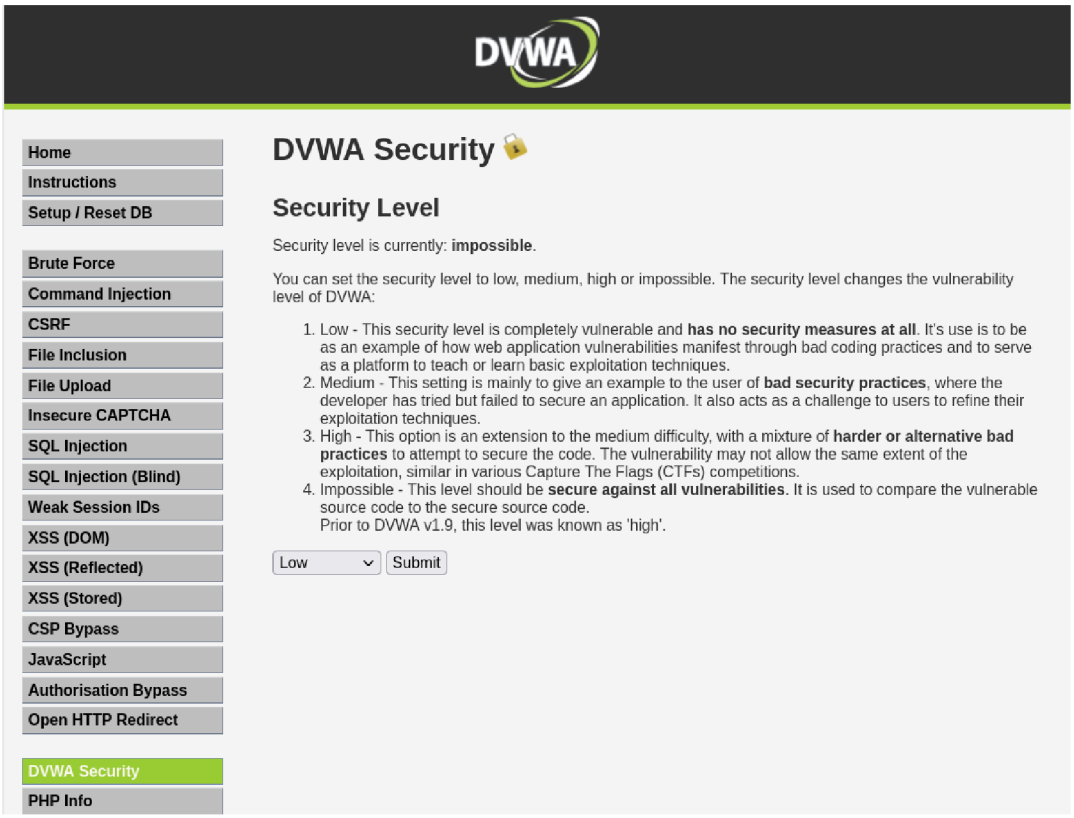
Dalším krokem byla úprava konfiguračního souboru php.ini pomocí příkazu sudo nano php.ini. Byly specifikovány parametry allow_url_fopen a allow_url_include tak, aby splňovaly požadavky aplikace DVWA.

Nakonec, po všech předchozích konfiguracích, lze spustit Apache server příkazem sudo service apache2 start a následné ověření funkčnosti aplikace DVWA v prohlížeči na lokální adrese, kde je možné se přihlásit dříve zadaným jménem a heslem. Toto potvrzuje obrázek 6, kde lze vidět rozhraní aplikace DVWA po úspěšném přihlášení.



Obrázek 6: Rozhraní aplikace DVWA

Pro zajištění plné funkčnosti DVWA byl proveden reset databáze skrze uživatelské rozhraní aplikace. Následné znovu přihlášení, tentokrát pomocí uživatelského jména admin a hesla password.



Obrázek 7: Nastavení úrovni

Jako poslední věc je nutné nastavit úroveň zabezpečení na „low“ viz. obrázek 7 neboť při prvotním spuštění je tato úroveň nastavená na „imposible“ neboli nemožné.

4.1.4 Zabezpečení testovacího prostředí

Při nastavování systému Kali Linux v prostředí VirtualBox pro účely testování a zabezpečení je třeba zvážit několik klíčových aspektů:

1. Instalace: VirtualBox jak je výše zmíněno poskytuje izolované prostředí, které uživatelům umožňuje instalovat Kali Linux, aniž by měnili svůj primární operační systém.
2. Nastavení sítě: Síťové nastavení: Je potřeba se ujistit, že síťové nastavení virtuálního počítače je nakonfigurováno na "NAT" nebo "Host-Only", aby se zabránilo náhodnému vystavení internetu nebo místní síti během testování.

3. Funkce snímků (snapshot): VirtualBox poskytuje funkce snímků. Před provedením jakýchkoli změn nebo spuštěním potenciálně škodlivých nástrojů se doporučuje pořídit snímek.
4. Izolace malwaru: V případě testování malwaru je potřeba zajistit, aby virtuální počítač neměl přístup ke kritickým systémům ani k internetu, a předejděte tak náhodné infekci nebo úniku dat.
5. Hardwarová omezení: Jednou z nevýhod provozování systému Kali Linux ve VirtualBoxu jsou možná hardwarová omezení. Konfigurace nastavení virtual boxu a kali linux je sepsána v předešlé kapitole.
6. Aktualizace a zabezpečení: Pravidelná aktualizace operačního systému Kali Linux a VirtualBoxu, během celého testování. Získají se tím nejnovější bezpečnostní záplaty a funkce a předejte se tak dalšímu nebezpečí.

Použití VirtualBoxu pro testovací prostředí Kali Linux nabízí bezpečné a flexibilní řešení. Pro zajištění maximální bezpečnosti a funkčnosti je však nezbytné správně nakonfigurovat nastavení a uvědomit si omezení platformy.

4.2 Tvorba testovacích scénářů

Při přípravě testovací strategie pro webovou aplikaci je esenciální vybrat vhodné testovací scénáře. Práce se zaměřuje na bezpečnostní testování, což znamená vytvoření scénářů, které nejen ověří funkcionalitu a výkon aplikace, ale především odhalí potenciální zranitelnosti a slabiny. Hlavní zájem je upřen na aplikaci DVWA, která je známá svými úmyslně navrženými zranitelnostmi a poskytuje ideální platformu pro učení.

Kali Linux, uznávaný v bezpečnostních kruzích jako jeden z nejkomplexnějších nástrojů pro etický hacking, nabízí širokou škálu nástrojů a utilit, které lze využít při testování aplikací jako je DVWA. Proto byly vybrány testovací scénáře pro tuto aplikaci tak, aby využily možností a nástrojů dostupných v Kali Linuxu.

V následujících částech se práce věnuje konkrétním testovacím scénářům pro DVWA, navrženým tak, aby plně využily možností a nástrojů, které Kali Linux nabízí pro etický hacking a testování bezpečnosti.

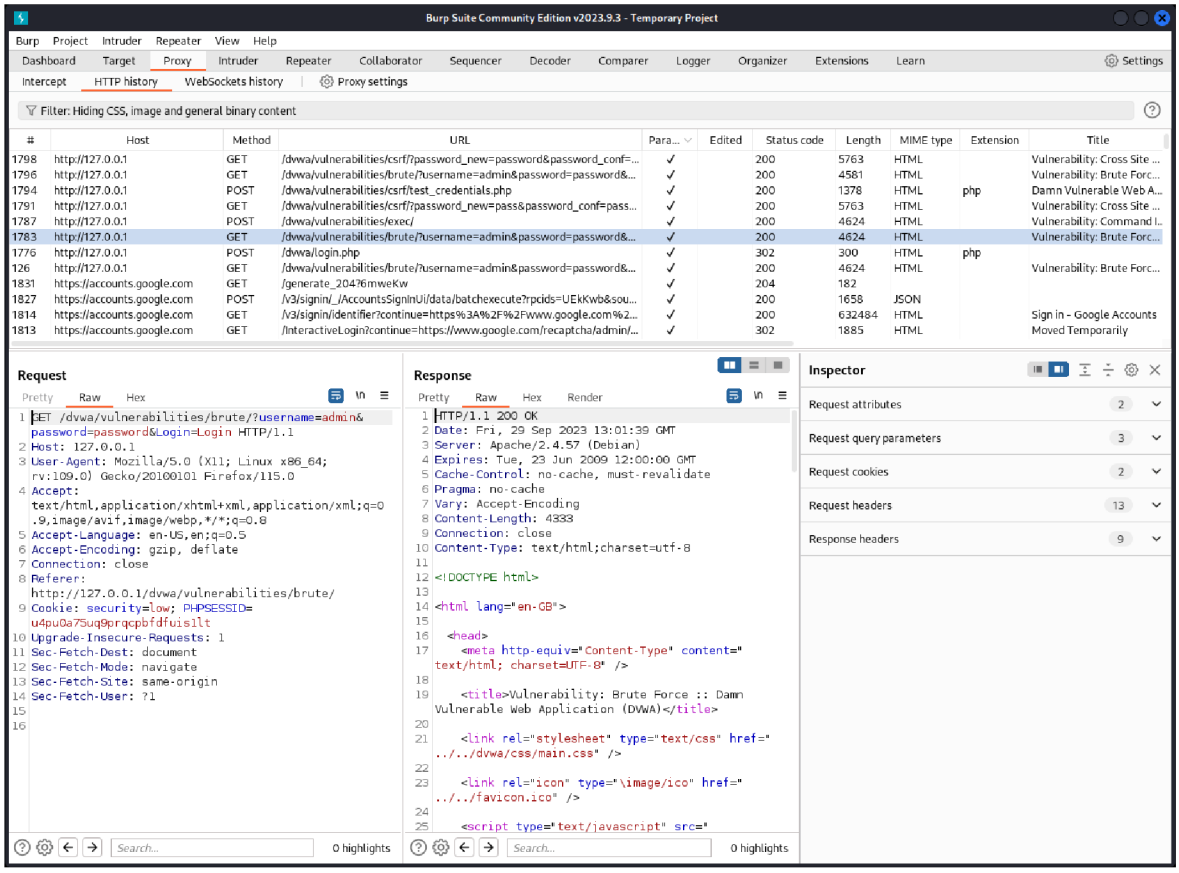
4.2.1 Nalezení zranitelností aplikace

Pro nalezení zranitelností bude použit nástroj Burp Suite, který je detailněji popsán v části 3.4 Nástroje pro etický hacking. Burp Suite je nástrojem pro zjišťování zranitelností webových aplikací a je široce používán odborníky na bezpečnost. Je ideální pro analýzu komunikace mezi webovým klientem a serverem, což umožňuje detailní testování různých aspektů webových aplikací.

Nejprve je potřeba připravit prohlížeč a nastavit správné porty. Firefox mění svůj port, a proto je nutné rozšíření FoxyProxy, kde lze změnit port tak, aby odpovídal tomu, co je v aplikaci Burp Suite. Důležité je také stáhnout certifikaci Burp CA.

Následuje ruční procházení neboli manual crawling, které nahrazuje funkci Spider, jelikož není dostupná v community verzi. Jedná se tedy ruční procházení vybraných funkcí v aplikaci DVWA, které je možné otestovat dle dostupných nástrojů. Tímto způsobem se generují a zachycují požadavky v Burp Suite.

Jedná se o časově náročnější úkol, ale zajistí to, že jsou pokryty všechny vybrané funkce. Zranitelnosti jsou identifikovány dle požadavku a odpovědi, jak je zobrazeno na obrázku 8.



Obrázek 8: Burp Suite Request/Response, Zdroj: Vlastní

4.2.1.1 Brute force

Potenciální hrozbu útoků hrubou silou lze klasifikovat jako střední až vysokou. Toto hodnocení se liší v závislosti na povaze chráněných informací a struktuře hesel z hlediska jejich délky a složitosti. Úspěšné prolomení hesla útočníkem prostřednictvím správného uhodnutí hesla může otevřít cestu k neoprávněnému přístupu do systému, následnému narušení dat a dalším souvisejícím bezpečnostním hrozbám.

Při analýze byl pozorován požadavek GET s parametry, a to uživatelské jméno=admin a heslo=password. Systém odpověděl stavem "200 OK", který byl doprovázen zprávou "Vítejte v oblasti chráněné heslem admin". Toto chování naznačuje, že v aplikaci chybí ochranná opatření proti útokům hrubou silou, což je patrné z absence mechanismů, jako jsou postupy pro zablokování účtu nebo zavedení ověření CAPTCHA po několika po sobě jdoucích neúspěšných pokusech o přihlášení.

4.2.1.2 Command injection

Útoky typu Command Injection představují značné bezpečnostní riziko, které je často klasifikováno jako vysoké. Při těchto útocích může útočník prostřednictvím zranitelné

aplikace spustit libovolné příkazy v hostitelském operačním systému, čímž získá potenciální plnou kontrolu nad hostitelským systémem. Rozsah hrozby závisí na oprávněních zneužívané aplikace. Úspěšné injektování příkazů může ohrozit integritu, dostupnost a důvěrnost aplikace a dat.

Byla provedena analýza požadavku POST zaslaného na výzvu "Command Injection" v nástroji DVWA. Parametry požadavku obsahovaly ip=8.0.0.1, což naznačovalo potenciální operaci "Ping a device" se zadanou IP adresou. Výsledná odpověď obsahovala standardní výstup příkazu ping provedeného se zadanou IP adresou. To ukazuje, že aplikace přijímá uživatelský vstup a přímo jej používá v příkazu na úrovni systému bez dostatečné sanitace. Nedostatek validačních nebo sanitačních opatření otevírá cestu k tomu, aby škodliví aktéři připojili nebo vložili libovolné příkazy, které server provede.

Vzhledem k pozorovanému chování může útočník tuto zranitelnost zneužít ke spuštění libovolných příkazů na serveru, které mohou sahat od získání systémových informací až po škodlivější akce, jako je manipulace s daty nebo dokonce kompromitace serveru, v závislosti na konfiguraci a oprávněních serveru.

4.2.1.3 CSRF

Poskytnutý požadavek HTTP označuje metodu GET použitou k odeslání požadavku na změnu hesla. Zásadní je, že tento požadavek GET vkládá do adresy URL nové heslo (password_new=pass) a jeho potvrzení (password_conf=pass). To je problematické, protože adresy URL lze snadno sdílet, přihlásit nebo náhodně odhalit.

Odpověď "Password Changed" znamená, že aplikace povolila změnu hesla prostřednictvím požadavku GET bez jakékoli formy tokenu CSRF nebo ověření, které by zajistilo, že požadavek byl skutečně iniciován uživatelem. To znamená, že pokud útočník dokáže uživatele oklamat a přimět ho k návštěvě škodlivě vytvořeného odkazu nebo webu, který tento požadavek spustí, může být heslo uživatele změněno bez jeho vědomí.

Dalším aspektem, kterého je třeba si všimnout, je absence tokenu anti-CSRF v požadavku. Bezpečně navržené aplikace obvykle obsahují takový token, který server ověří, aby se ujistil o legitimitě požadavku. Bez této ochrany se aplikace stává potenciálním cílem útoků CSRF.

4.2.1.4 File inclusion

Změna hesla prostřednictvím GET: První sdílený požadavek GET vykazuje zjevnou chybu. Aplikace umožňuje změnu hesla prostřednictvím požadavku GET, přičemž nové heslo je

vloženo přímo do adresy URL. Tato konstrukční chyba umožňuje útočníkovi snadno vytvořit škodlivý odkaz a přimět uživatele k nevědomé změně hesla.

Chybějící tokeny Anti-CSRF: Klíčovým prvkem, který v požadavku chybí, je token anti-CSRF. Webové aplikace obvykle takový token vkládají do formulářů, aby ověřily pravost požadavku. Absence takového mechanismu v této aplikaci je znepokojující a činí ji zranitelnou vůči útokům CSRF.

Funkčnost testovacích pověření (test credentials): Požadavek POST na /dvwa/vulnerabilities/csrf/test_credentials.php potvrzuje, zda jsou zadaná pověření správná. Úspěšné ověření hesla pro "admin" v odpovědi naznačuje možnou sekundární zranitelnost. Útočník by mohl vytvořit útok CSRF a použít tuto funkci k uhodnutí nebo potvrzení hesla, což by usnadnilo zneužití dalších zranitelností nebo získání neoprávněného přístupu.

Aplikace odkazuje na atribut SameSite cookie jako na opatření k zabránění CSRF. Tento atribut může při správném nastavení skutečně zmírnit CSRF. Zranitelná povaha aplikace však naznačuje, že tento atribut nemusí být vhodně implementován.

Zranitelnost v podobě začlenění souborů: Stránky (include.php, file1.php, file2.php a file3.php), vykazují zranitelnost LFI (Local File Inclusion), protože aplikace používá parametr "page" k zahrnutí souborů bez validace.

- Soubor 1: Zobrazuje podrobnosti o uživateli včetně uživatelského jména a IP adresy.
- Soubor 2: Obsahuje vtip, který demonstruje schopnost přepínat obsah na základě parametru "page".
- Soubor 3: Poskytuje více personalizovaných údajů o uživateli a demonstruje schopnost aplikace zpracovávat soubory PHP a zobrazovat na jejich základě data. Zahrnuje položky Uživatelské jméno admin, IP adresa 127.0.0.1, Řetězec User-Agent, URL odkazu, IP adresa hostitele.

Pokud tato zranitelnost LFI existuje v reálném scénáři, může být zneužita ke čtení citlivých souborů ze serveru, což může vést k vyzrazení informací nebo jiným škodlivým aktivitám.

Riziko pro soukromí může představovat také zobrazování podrobností o uživateli a personalizovaných údajů.

4.2.1.5 Sql injection

Zatížení '1'='1' je tautologický příkaz SQL. Vždy se vyhodnotí jako true a jeho primární použití je testování, zda je aplikace zranitelná vůči SQL injection. Pokud není vstup správně sanitizován nebo parametrizován, může injektování takového payloadu často vést k

nechtěnému chování aplikace, například ke zveřejnění všech řádků tabulky. Na to navazují důsledky zranitelnosti, které z toho plynou:

- Odhalení dat: Pokud je aplikace náchylná k SQL injection, mohou záškodníci potenciálně získat citlivá data z databáze.
- Manipulace s daty: Útočníci mohou data v databázi nejen prohlížet, ale mohou být schopni je i měnit nebo mazat.
- Potenciální provádění příkazů: V určitých scénářích SQL injection (jako jsou zásobníkové dotazy) mohou útočníci dokonce spustit libovolné příkazy na hostitelském serveru.

4.2.1.6 Sql injection (blind)

Úroveň rizika je vysoká. Ačkoli je zneužití „slepého“ SQLi časově náročnější než zneužití klasického SQLi, stále může vést k úplnému ohrožení databáze.
Z poskytnutého požadavku GET s použitím parametru id=sda vrátila odpověď stav HTTP "404 Not Found" doprovázený zprávou "User ID is MISSING from the database". To naznačuje, že vstup přímo ovlivňuje výsledek dotazu SQL. Ačkoli není viditelná žádná přímá chybová zpráva SQL, tento rozdíl v chování na základě vstupu naznačuje potenciální slepou injektáž SQL.

4.2.1.7 Ověřování pomocí JavaScriptu

Úroveň rizika je mírná. Útočník může potenciálně zpětně analyzovat proces generování tokenů, padělat tokeny a obejít validační mechanismy. Poskytnutý zdroj stránky odhaluje funkci generate_token() v jazyce JavaScript. Tato funkce přijme vstup uživatele (frázi), provede na něm kódování rot13 a poté vypočítá jeho hash MD5. Výsledek je nastaven jako hodnota tokenu ve formuláři.
Zpětným inženýrstvím tohoto procesu generování tokenu může útočník padělat platné tokeny bez použití poskytnuté funkce a potenciálně tak obejít jakoukoli validaci na straně serveru, která se na tento token spoléhá nebo manipulovat s funkcemi aplikace, které jsou hlídány tímto mechanismem tokenu.

4.2.1.8 XSS (Reflected)

Vysoká úroveň rizika. XSS (Reflected) mohou útočníci využít ke spuštění škodlivých skriptů v kontextu relace oběti, což může vést k odcizení tokenů relace, manipulaci s obsahem webové stránky nebo k dalším útokům na uživatele.

V daném požadavku se z užitečného zatížení po dekódování adresy URL stane <script>alert('test');</script>. Tento skript se přímo promítne a provede v prohlížeči uživatele, jak je vidět v odpovědi, uvnitř značky pre: <pre>Hello <script>alert('test');</script></pre>. Útočník může vytvořit škodlivou adresu URL obsahující užitečné zatížení. Když na něj oběť klikne, skript se spustí v kontextu její relace.

4.2.1.9 XSS (Stored)

XSS (stored) může mít významnější důsledky než XSS (reflected), protože škodlivý skript je trvale uložen a může ovlivnit každého uživatele, který navštíví stránku obsahující tento škodlivý skript.

Poskytnutý POST požadavek ukazuje, že uživatel se pokusil uložit škodlivý skript do polí "Jméno" (txtName) a "Zpráva" (mtxMessage).

Odpověď odráží uložený skript v sekci návštěvní knihy, konkrétně:

Jméno: <script>al
Zpráva: <script>alert('test');</script>

To ukazuje, že škodlivý vstup byl uložen a je bez jakékoli sanitace odrážen zpět, čímž potvrzuje zranitelnost uloženého XSS.

Útočník může do návštěvní knihy vložit škodlivé skripty. Když jiní uživatelé navštíví stránku, uložený skript se spustí, což může vést k potenciálnímu převzetí relace, krádeži dat nebo dokonce k exploitaci prohlížeče, v závislosti na charakteru uloženého skriptu.

4.2.2 Návrhy testovacích scénářů

V této sekci představuji návrhy testovacích scénářů pro různé typy útoků, které nabízí aplikace DVWA. Testovací scénáře popisují celý průběh daného útoku za použití určitého nástroje pro útok. V rámci testovacích scénářů musí být zapsány všechny kroky, a to včetně nastavení.

Mé testovací scénáře zahrnují široké spektrum útoků, které lze na aplikaci DVWA uskutečnit, a to za využití nástrojů obsažených v distribuci Kali Linux. Scénáře jsou sestaveny na základě nalezených zranitelností, které byly identifikovány v předešlé části mé

práce, dokumentací nástrojů a návodů vytvořených autory nástrojů. Na základě těchto scénářů budou bude probíhat testování, podle kterých budou navrženy způsoby ochrany a komparace změn po aplikování ochrany.

Scénáře se tedy zaměřuji na jednotlivé zranitelnosti v následujícím pořadí:

1. Brute Froce – Příloha 1
2. Command Injection – Příloha 2
3. CSRF – Příloha 3
4. File inclusion – Příloha 4
5. SQL injection – Příloha 5
6. Javascript – Příloha 6
7. XSS – Příloha 7

4.3 Testování

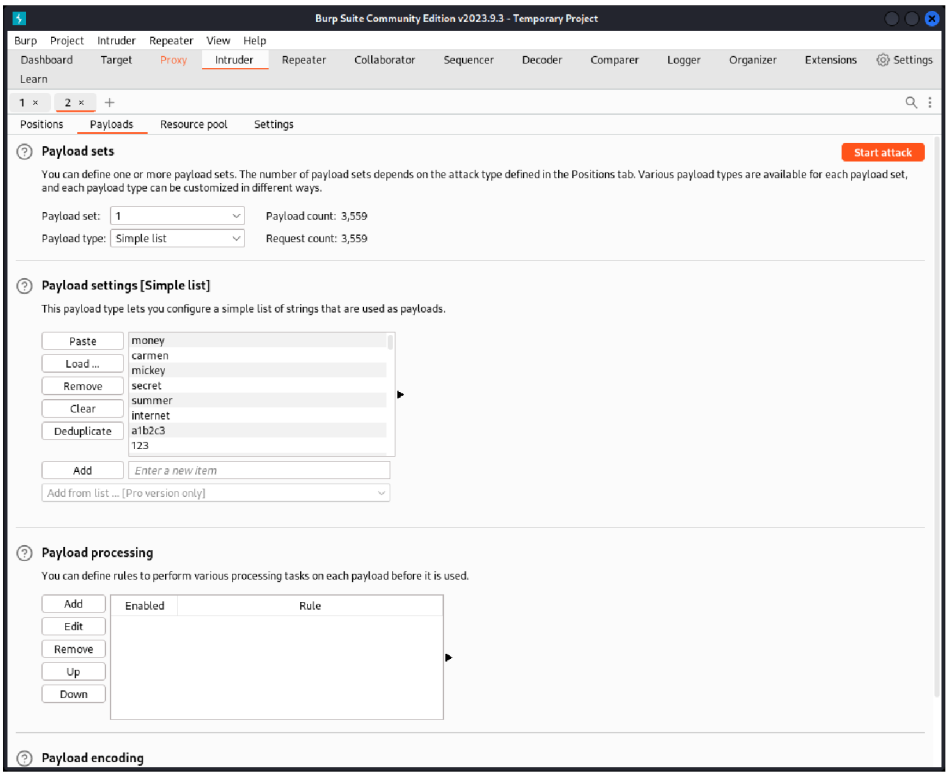
Tato část se zaměřuje na samotné testování na základě návrhů testovacích scénářů. Je rozdělena na provedení testů, které se podrobně zaměřují na každou zranitelnost, na kterou byl vytvořen testovací scénář. Po provedení testů jsou sepsány výsledky této části.
V druhé části se zaměřuje na návrhy způsobů ochrany, které se odvíjí od prvotních testů při nízkém zabezpečení. Opět na pouze zranitelnosti, které byly podrobeny otestování.

4.3.1 Provedení testů

Jako první je potřeba nastavit zabezpečení aplikace DVWA na obtížnost „Low“ pro demonstraci té nejzranitelnější stránky a jednodušší zjištění správnosti testovacích scénářů. V další kroku je potřeba nastavit proxy v prohlížeči, čímž bylo docíleno použitím rozšíření do prohlížeče nazvané FoxyProxy, které je bezplatné a volně dostupné. Tím tak dojde k úspěšnému spojení hlavního nástroje BurpSuite a webové aplikace DVWA.

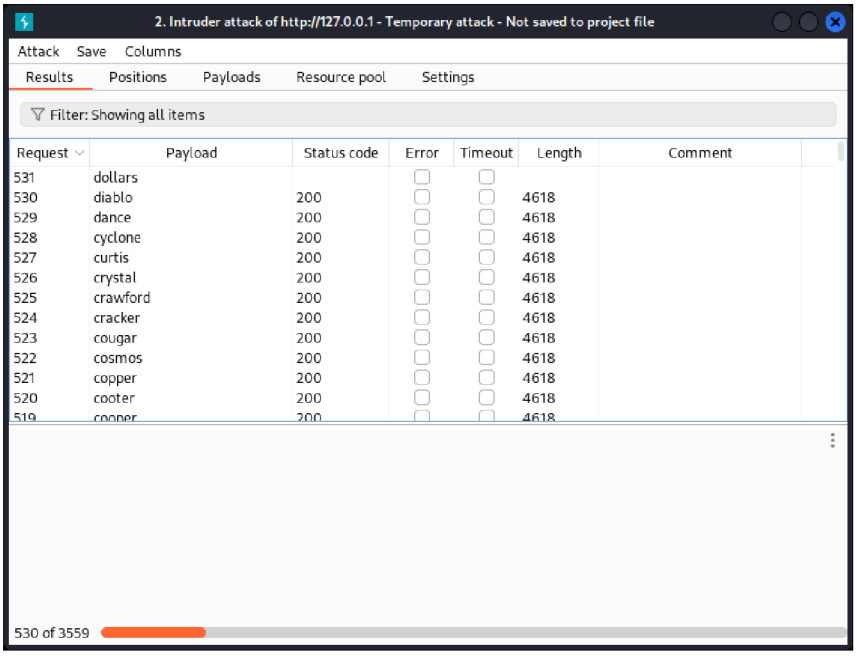
4.3.1.1 Provedení brute force útoku dle testovacího scénáře 1

Pomocí nástroje Burp Suite a přes možnost proxy je nejprve nutné najít zranitelnost. Po jejímž objevení se označí cílené místo. Následuje nahraní takzvaného „paylodu“ neboli seznamu slov, která se budou používat na prolomení hesla. Je několik možností, jak seznam vytvořit ať už ručně nebo přes nějaký generátor, který lze najít kdekoliv na internetu. Naštěstí Kali Linux disponuje hned několika seznamy ve složce wordlists. Pro testování je vybrán seznam john.lst. Tento soubor obsahuje 3559 běžně používaných hesel a je tak dobrým startem pro testování. Po vložení paylodu a správním nastavení, jak je vidět na obrázku 9, lze spustit útok.



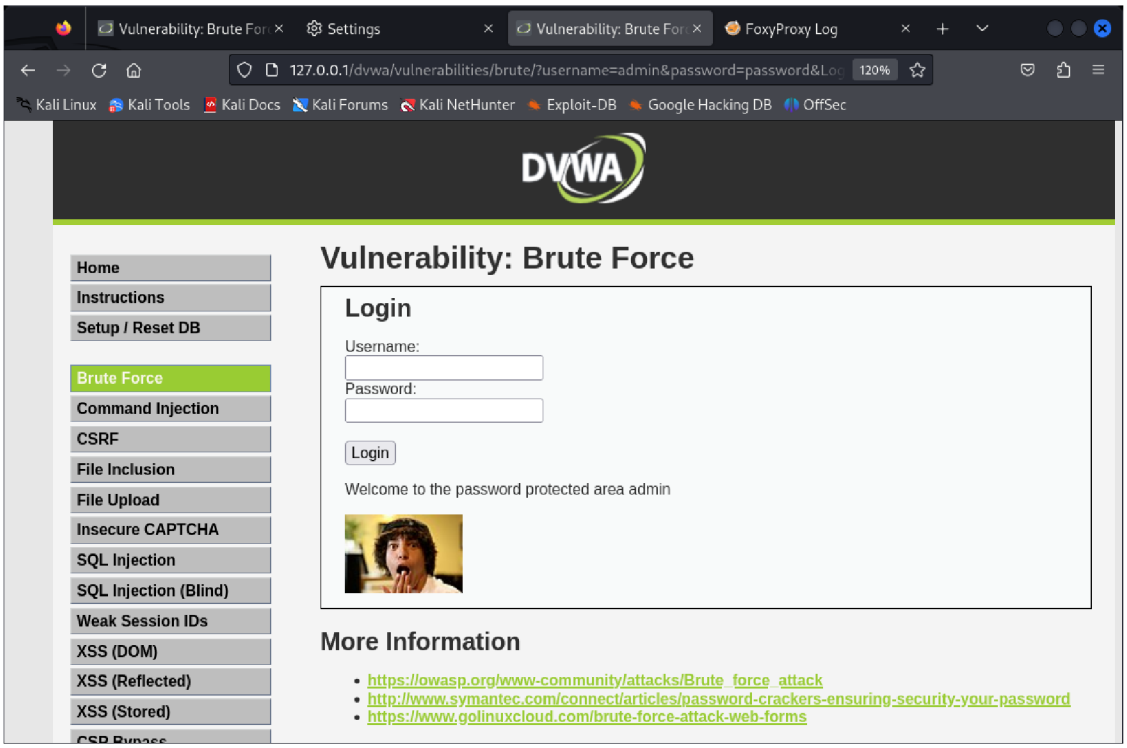
Obrázek 9: Burp Suite Intruder, Zdroj: vlastní

Burp suite následně projíždí jedno slovo za druhým, jak je vidět na obrázku 10. A v tento okamžik se čeká na změnu sloupce lenght, jehož hodnoty by měli mírně narůst z čehož bude zřejmé, že došlo k nalezení hesla. I přes to, že přihlašovací údaje jsou známy, musí se útok provést jako útočník. Proto útok zabral něco okolo tří hodin, jelikož community verze programu Burp Suite je značně omezená a test trvá déle.



Obrázek 10: Burp Suite Intruder – průběh útoku, Zdroj: Vlastní

Po nalezení hesla, jehož délka je 4661 oproti ostatním slovům s délkou prodlevy 4618 nezbývá než dané heslo vyzkoušet v přihlášení na obrázku 11.



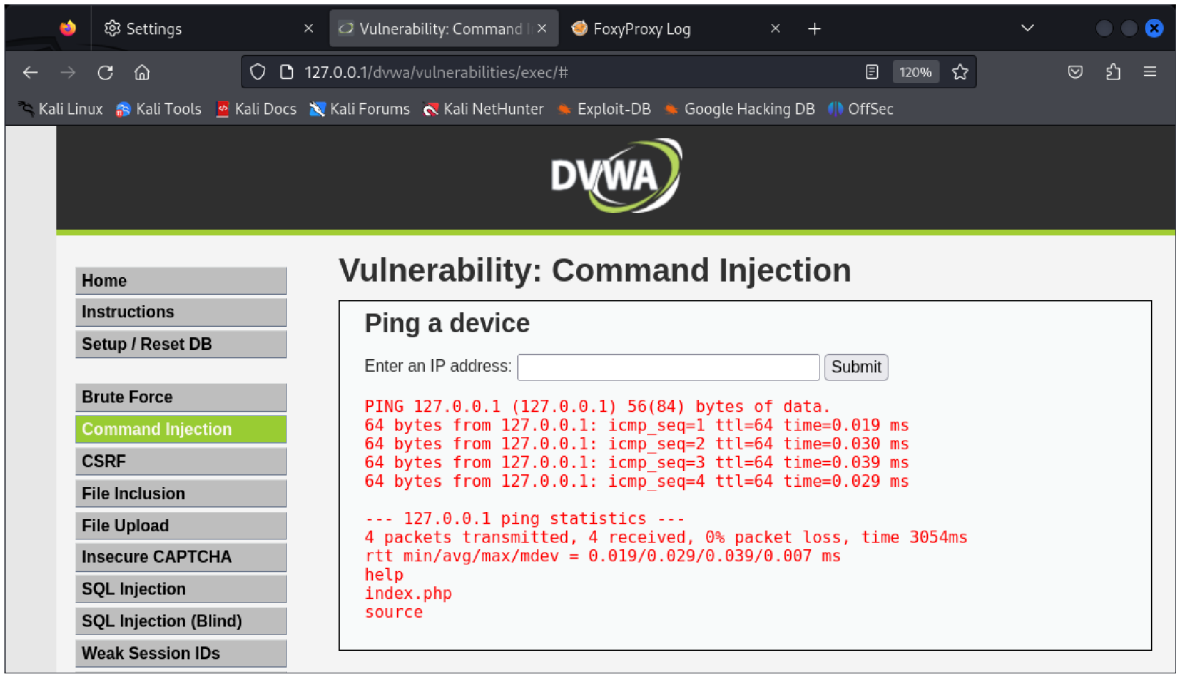
Obrázek 11: Prolomení hesla, Zdroj: vlastní

Po kliknutí na obrázek po přihlášení je možné zjistit, počet uživatelů zaevidovaných v této aplikaci. Je jich celkem 5 a ke každému lze nyní když je známo uživatelské jméno tak stejným způsobem nebo pomocí nástroje hydra zjistit heslo.

4.3.1.2 Provedení command injection dle testovacího scénáře číslo 2

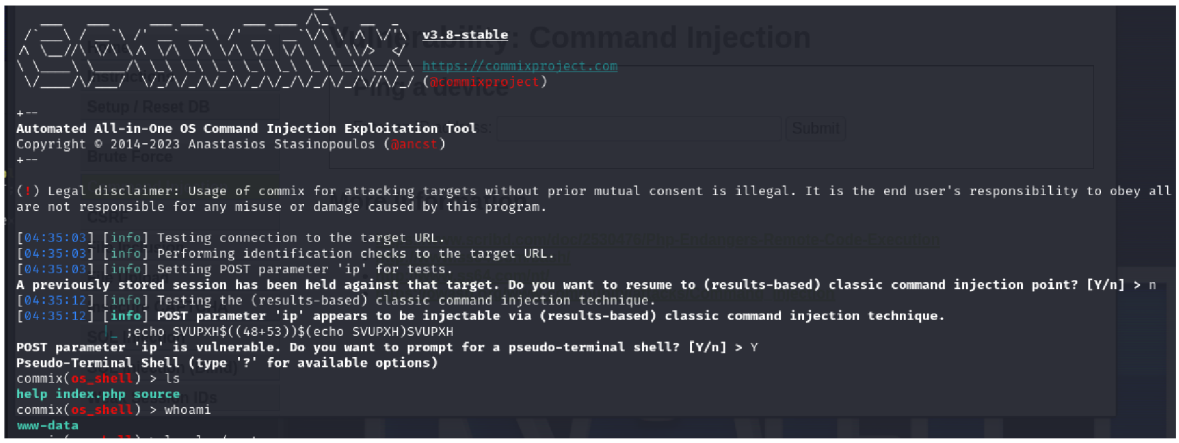
Z průzkumu zranitelností na zabezpečení „Low“ je známo, že uživatelský vstup není nijak kontrolovaný proti škodlivým příkazům. Je možné zadat více příkazů, ty spojovat dohromady a nemusí se jednat o IP adresu, což vstup požaduje. Příkazy fungují i bez mezer a dá se podle nich zobrazit zdrojový kód, který slouží k pochopení stránky.

Z jednoduchých příkazů jako je ls, pwd, whoami a dalších je možné zjistit téměř vše od výpisu souborů, jejich prohlížení, informace o uživateli, jeho systému a mnoho dalších, jak je vidět na obrázku 12. a to jen pomocí jednoho nezabezpečeného pole.



Obrázek 12: Výstup po zadání příkazu ls, Zdroj:Vlastní

K jednoduššímu použití slouží nástroj commix, který se napojí pomocí url odkazu, identifikátoru PHPSESSID a data vstupu. To vše je získatelné z nástroje Burp Suite jak tomu bylo v předešlém testu. Zde opět fungují stejné příkazy tak jako v rozhraní aplikace.



Obrázek 13: Commix výstup, Zdroj: Vlastní

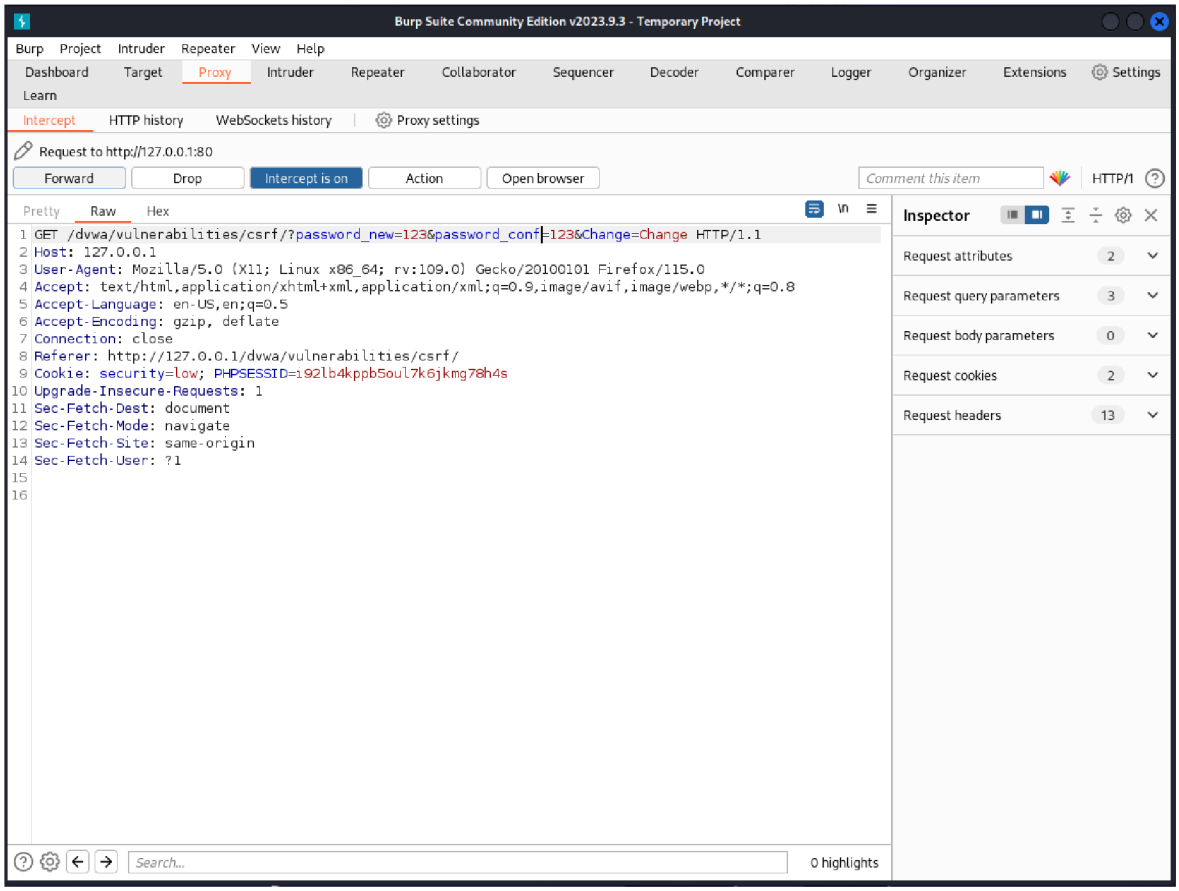
4.3.1.3 Provedení CSRF dle testovacího scénáře číslo 3

V případě nulového zabezpečení lze z odkazu stránky získat url „http://127.0.0.1/dvwa/vulnerabilities/csrf/?password_new=123&password_conf=123&Change=Change#“, který lze využít na změnu hesla prostým posláním uživateli a po jeho kliknutí. Pro větší skrytí lze vytvořit jednoduchou stránku s tlačítkem viz obrázek 14. Jedná se pouze o demonstraci a k docílení úspěchu by byla zapotřebí propracovanější stránka. Záleží tedy na úsilí útočníka, jakou si dá práci s jejím vytvořením.

```html
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Document</title>
</head>
<body>
        <h3>Zmena hesla</h3>
        <a href="http://127.0.0.1/dvwa/vulnerabilities/csrf/?
password_new=123&password_conf=123&Change=Change#">Pro zmenu kliknete sem</a>
</body>
</html>
```

Obrázek 14: HTML kód k vytvoření falešné stránky, Zdroj: Vlastní

Další možností je využít Burp Suite který zachytí změnu hesla a může dojít k jejímu změnění, jak je vidět na obrázku 15.



Obrázek 15: Burp Suite – zachycení hesla, Zdroj: Vlastní

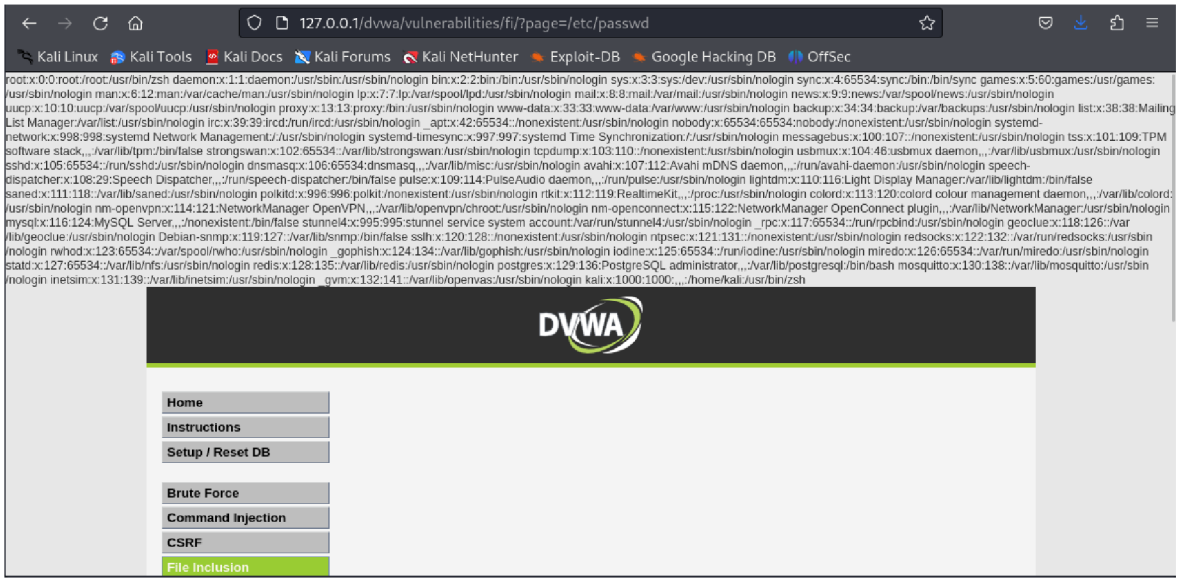
Kde program zachytí změnu hesla a je možné změnit password_new a password_conf z původně zadávaného hesla password na 123 a po ozkoušení je vidět na obrázku 16, že změna funguje.

Obrázek 16: Potvzení hesla, Zdroj: Vlastní

4.3.1.4 Provedení File Inclusion dle testovacího scénáře číslo 4

Po proklikáváni dostupných souborů je možné si všimnout ve vyhledávacím poli změně url adresy, takže při různých pokusech se jde dostat například do souboru 4, který na stránce ani není zobrazen. Po průzkumu zdrojového kódu lze odhalit skryté řádky jako například doplněním za page ../../hackable/flags/fi.php

Ze zdrojového kódu je také možné se dočíst o složce passwd a po vložení 127.0.0.1/dvwa/vulnerabilities/fi/?page=/etc/passwd, ze které lze získat přihlašovací údaje a hesla, jak je vidět na obrázku 17.

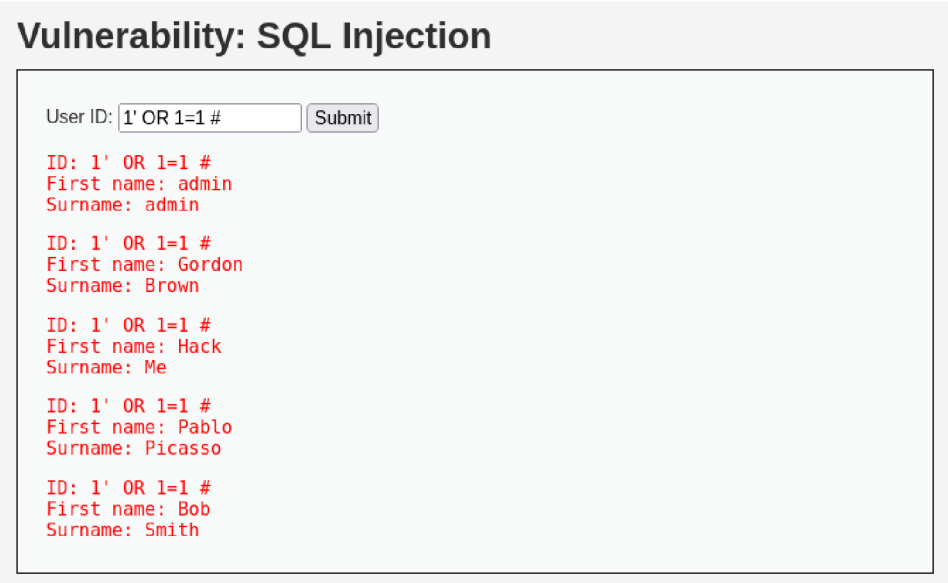


Obrázek 17: Výstup File Inclusion, Zdroj: Vlastní

4.3.1.5 Provedení SQL injection dle testovacího scénáře číslo 5

Po průzkumu zdrojového kódu lze zjistit že u atributu id není žádná validace, to znamená že není nutné psát do pole to co stránka chce, ale to, co chce útočník. Tedy lze například udělat následující 1' OR 1=1 # což znamená, pokud se uživatelské id = 1 nebo 1 = 1, lze získat

jména registrovaných uživatelů, jak je vidět na obrázku 18. Opět se dá tímto způsobem získat vše co je zde uložené.

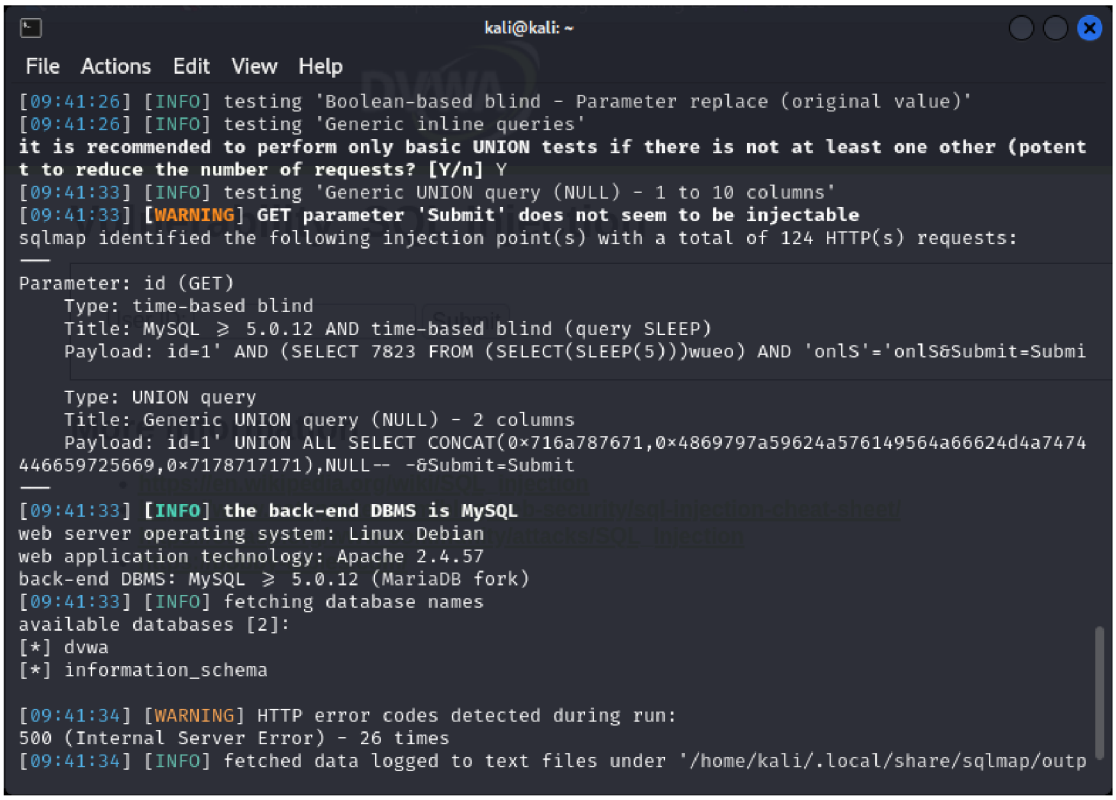


Obrázek 18: Výstup aplikace DVWA po zadání vstupu, Zdroj: Vlastní

Tohle ale není jediný způsob. Další možností je použitím nástroje Sqlmap, který je detailněji popsán v části o Sqlmap. Kdy poslední verze Sqlmap nabádá uživatele nebo útočníka k tomu co potřebuje jako je adresa, cookies, PHPHSESSID a další. Nebo je možné zadat následující příkaz do Sqlmap:

sqlmap -u "http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=aq33pfpbp5685830jqu1ftudob" –dbs

A ten pak projde stránku a nalezne hlavní soubory, jak je vidět na obrázku 19.

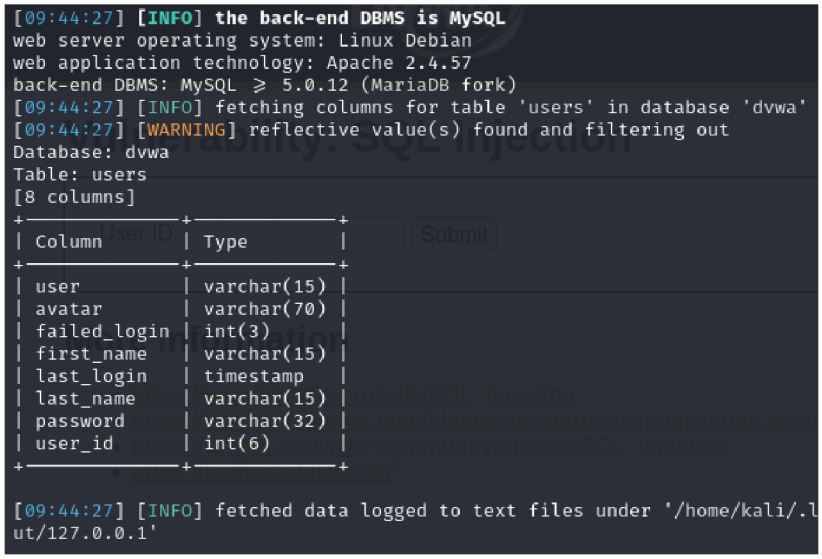


Obrázek 19: Sqlmap – výstup po prolomení, Zdroj: Vlastní

Je zřejmé, že se zde nachází složka dvwa a po jejím dalším průzkumu se lze dostat až k uživatelům pomocí:

sqlmap -u "http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=aq33pfpbp5685830jqu1ftudob" -D dvwa -T users –columns

jehož výsledek je vidět na obrázku 20.



Obrázek 20: Sqlmap – nalezení souborů, Zdroj: Vlastní

Je známo, že jsou zde uživatelé, které lze zobrazit buď podle toho co požaduje sqlmap nebo použitím následujícího příkazu:

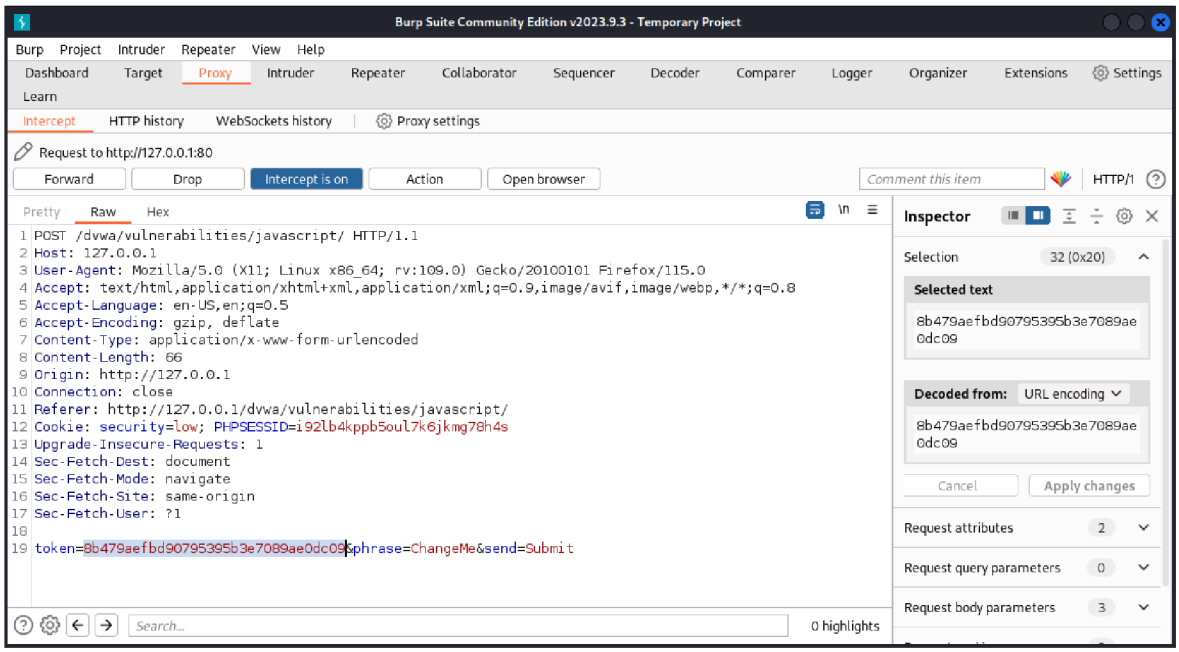
sqlmap -u "http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=aq33pfpbp5685830jqu1ftudob" -D dvwa -T --dump-all

Po zadání je zobrazen kompletní seznam uživatelů i s jejich jmény, hesly, posledních přihlášeních a další, zobrazených na obrázku 21.

Obrázek 21: Sqlmap – nalezení hesel, Zdroj: Vlastní

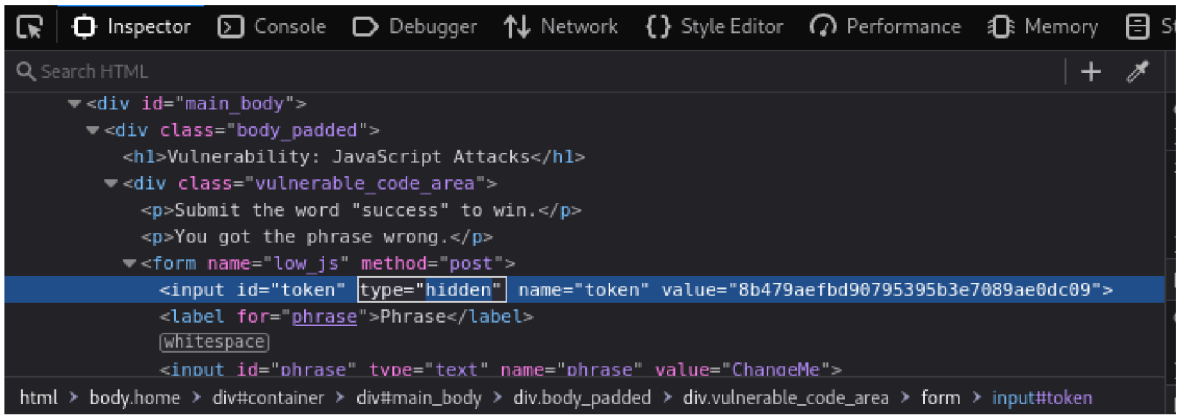
4.3.1.6 Provedení ověření pomocí Javascript dle testovacího scénáře číslo 6

Ve zdrojovém kódu je vidět že systém pracuje s tokenem pro ověření slova, ten lze získat například nástrojem Burp Suite, který přes službu Proxy zachytí žádost a zobrazí ji, jak je vidět na obrázku 22.



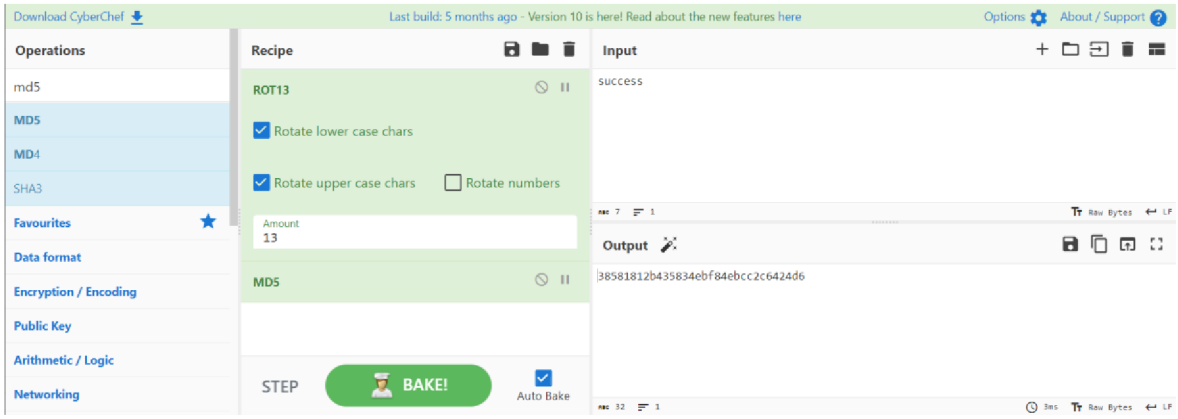
Obrázek 22: Burp Suite – Nalezení tokenu, Zdroj: Vlastní

Pole pro vložení tokenu je ale skryté, nicméně lze odkrýt jednoduchou úpravou, jak je vidět na obrázku 23 smazáním slova „hidden“. Po průzkumu je vidět, že token se nemění se změnou slova success, které stránka požaduje. Token je tedy pouze pro slovo „change me“.



Obrázek 23: Odkrytí skrytého pole, Zdroj: Vlastní

Pro to slouží například stránka CyberChef (https://gchq.github.io/CyberChef/), kde použitím hashovacích operací ROT13 a MD5, lze vytvořit token pro slovo success, jak je vidět na obrázku 24.

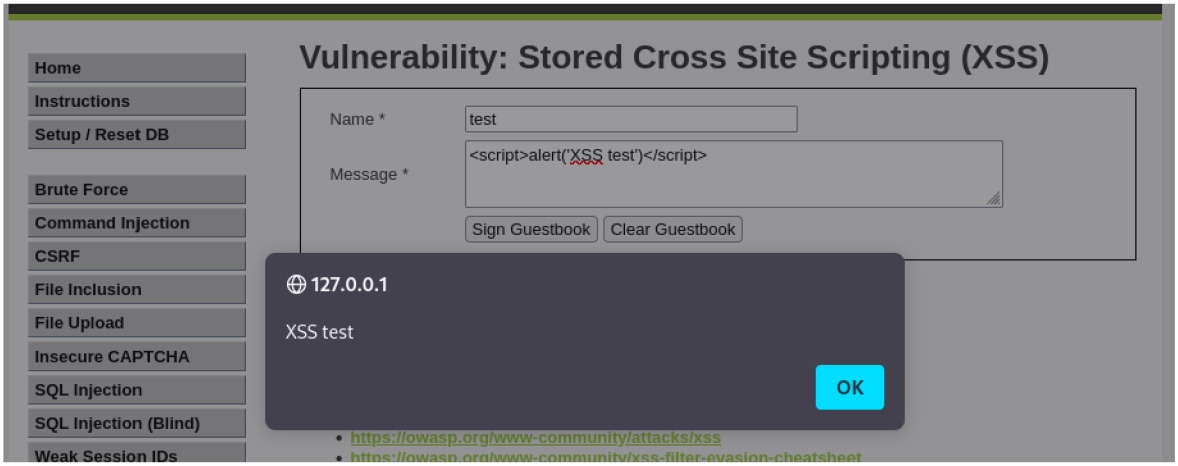


Obrázek 24: Vygenerovaný token ze stránky CyberChef, Zdroj: (The Cyber Swiss Army Knife, 2016)

Po vložení správného tokenu, který pasuje ke slovu „success“, tlačítko pro vložení funguje a je obdržen nápis „Well Done“.

4.3.1.7 Provedení XSS dle testovacího scénář číslo 7

Stored XSS v tomto případě slouží jako knihovna návštěv, kde je vše uloženo v databázi a zobrazeno bez jakéhokoli očištění. Proto posláním jednoduchého skriptu: <script>alert(`XSS test`)</script>, kdykoli kdo navštíví stránku, obdrží upozornění zobrazené na obrázku 25.



Obrázek 25: Zpráva při vstupu, Zdroj: Vlastní

V tomto případě je zobrazena jen zpráva, ale pomocí skriptu útočník tak může získat citlivá data uživatele nebo systému.

4.3.2 Výsledky

V provedení všech útoků dle testovacích scénářů došlo k úspěšnému prolomení zabezpečení. Proto v následující sekci budou navrženy způsoby ochrany, které jsou možné udělat v aplikaci DVWA. Dále v průběhu testu došlo k úplnému zhroucení celého systému pod náročností několika běžících útoků, konktrétně u SQL injection. Pro další testy bude potřeba navýšit výkon.

4.3.3 Návrh způsobů ochrany proti daným útokům na webové aplikace

Způsoby ochrany jsou sestavovány dle dostupných dokumentací k nástrojům a informací dostupných v aplikaci DVWA. Každý návrh se dělí na dvě části: první kde je sepsán návrh v číselném pořadí podle míst, která byla použita k napadnutí. Tato část neslouží k zamezení prolomení bezpečnosti ale pouze k jejímu zpomalení. Druhá část naopak obsahuje výpis dalších ochran, které by mohli být použity k úplnému zamezení provedení útoku.

4.3.3.1 Návrh ochrany proti brute force útoku

Brute force útoku se dá definitivně zamezit, pokud někdo chce. Zároveň z důvodu této práce jsou zohledněny možnosti aplikace DVWA a ostatní možnosti jsou uvedeny v dalších zabezpečení.

1. Omezení počtu žádostí na server: Nastavení omezení pro počet žádostí na server za určité období. To může pomoci zabránit rychlým Brute-Force útokům.

2. Použití anti Cross-site Request Forgery tokenu – Při každém pokusu o přihlášení se odesílá CSRF token. Hodnota tohoto tokenu se mění při každém načtení stránky.
3. Náhodné rozmezí po vložení neúspěšného hesla.

Další možnosti zabezpečení: zabezpečení proti hrubé síle neboli zablokování IP adres, dvoufázová autentizace (2FA), logování a sledování, monitorování aktivit uživatelů, zákaz používání nebezpečných hesel, použití bezpečnostních knihoven a frameworků.

4.3.3.2 Návrh ochrany proti útoku command injection

1. Filtrování speciálních znaků – Filtrace speciálních znaků, které by mohly být zneužity při command injection, jako jsou:, &, |, atd.
2. Omezení systémových příkazů – Omezený seznam povolených systémových příkazů, které lze spouštět.
3. Whitelist znaků – Použití whitelistu znaků znamená, že aplikace povoluje pouze určený seznam bezpečných znaků v uživatelských vstupech.
4. Detekce neautorizovaných příkazů – Detekce a odmítnutí pokusů o spuštění neautorizovaných příkazů a zaznamenávat je pro sledování a analýzu.

Mezi další možnosti zabezpečení patří: Ověření syntaxe IPv4, kontrola vstupů, používání parametrizovaných dotazů, implementace firewallu, pravidelná analýza kódu s pravidelné bezpečnostní testy z různých perspektiv.

4.3.3.3 Návrh ochrany proti CSRF útoku

1. Ochrana před UNION operátorem – je zakázáno používání UNION operátoru v SQL dotazech, což může zabránit některým typům SQL injection útoků.
2. Nemožnost měnit typ dotazu - např. SELECT, INSERT, UPDATE pomocí SQL injection. To znamená, že není možné přepínat mezi různými druhy SQL dotazů.
3. Omezení délky vstupních dat – omezení délky vstupních dat, což může snížit riziko některých druhů SQL injection útoků.
4. ANTI-CSRF Token: Aplikace generuje ANTI-CSRF token pokaždé, když požadujete stránku pro změnu hesla

Mezi další možnosti zabezpečení patří: Použití více než jednoho synchronizačního tokenu nebo složitějších algoritmů pro tokeny, implementace striktních politik využívajících několik vrstev ověřování pro odolnost vůči sofistikovanějším útokům.

4.3.3.4 Návrh ochrany proti File Inclusion útoku

1. Ochrana proti nulovému bajtu – Zabraňuje vkládání nulového bajtu do cesty k souboru, což by mohlo být využito k obcházení některých kontrol.
2. Omezení na povolené protokoly (např. http, https) - Může být nastaveno omezení na povolené protokoly, což minimalizuje možnosti útočníka manipulovat s protokolem při zahrnutí souborů.
3. Ochrana před URL dekódováním – Implementace ochrany proti dekódování URL může zabránit některým trikům, které útočníci mohou použít k obcházení kontrol.
4. Ochrana proti znakům specifickým pro operační systém – Filtrace nebo omezení určitých znaků specifických pro operační systém (např. "/", "", ":")

Mezi další možnosti zabezpečení patří: Úplné omezení možností zahrnutí souborů a implementace logování pro sledování pokusů o neoprávněné zahrnutí, sledování a analýza datových toků pro identifikaci anomálií v zahrnování souborů.

4.3.3.5 Návrh ochrany proti SQL injection

1. Ochrana před dvojitým URL kódováním (Double URL Encoding): aplikace odolávala pokusům o dvojité URL kódování, což může být využito k obcházení některých filtrů.
2. Nemožnost měnit typ dotazu: Nemožnost měnit typ dotazu pomocí SQLi. To znamená, že nelze přepínat mezi SELECT, INSERT, UPDATE a DELETE dotazy.
3. Omezení délky vstupních dat: Omezení délky vstupních dat může zabránit některým typům SQL injection útoků, které závisí na dlouhých vstupních řetězcích.
4. Filtrování některých klíčových slov: Filtrování některých klíčových slov, která jsou často spojena s SQL injection útoky.

Další zabezpečení: parametrizované dotazy, validace vstupních dat, princip nejmenších oprávnění (Principle of Least Privilege), escapování znaků, whitelist pro vstupní data, monitorování a logování, Web Application Firewall (WAF), aktualizace a patche, bezpečnostní testování, ochrana před time-based útoky.

4.3.3.6 Návrh ochrany proti Javascript útoku

1. Implementace CSP – Omezí, které zdroje mohou být spouštěny na stránce
2. Ochrana proti znakům speciálním pro JavaScript – Omezení nebo filtrace speciálních znaků, které mohou být zneužity k injekci škodlivého kódu.

3. HTTP Only Cookies – Používání atributu "HttpOnly" pro cookies, což zabraňuje JavaScriptu v přístupu k nim, což snižuje riziko útoků typu Session Hijacking.
4. Aktivace bezpečnostních hlaviček – Používání bezpečnostních hlaviček, jako je "X-Content-Type-Options", "X-Frame-Options" a "X-XSS-Protection", může posílit bezpečnost webové aplikace proti různým hrozbám.
5. Bezpečné přípojky (Safe eval) - Omezení používání eval() funkce nebo podobných konstrukcí, které by mohly způsobit injekci kódu.

Mezi další možnosti zabezpečení patří: Aktualizace knihoven a závislostí, CORS (Cross-Origin Resource Sharing) omezení, používání "strict mode", zakázání eval(), omezení používání Function constructoru, bezpečné správce hesel, omezení používání globálních proměnných, monitoring bezpečnostních událostí, používání bezpečného spojení (HTTPS), omezení přístupu k senzitivním API, bezpečné práce s cookies, bezpečné pracovní postupy s třetími stranami.

4.3.3.7 Návrh ochrany proti XSS útoku

1. Ochrana před vložením skriptu do databáze – Je implementována kontrola a validace vstupních dat, která jsou ukládána do databáze, aby se minimalizovala možnost vložení nebezpečného skriptu.
2. Omezení HTML tagů a atributů – Omezení na povolené HTML tagy a atributy může zabránit vkládání nebezpečných skriptů nebo atributů do stránky.
3. Content Security Policy (CSP) – Implementace CSP může omezit, které zdroje mohou být spouštěny na stránce, což snižuje riziko XSS útoků.
4. Aktivace bezpečnostních hlaviček – Používání bezpečnostních hlaviček, jako je "X-Content-Type-Options", "X-Frame-Options" a "X-XSS-Protection", může posílit bezpečnost webové aplikace proti různým hrozbám, včetně XSS.
5. Omezení oprávnění uživatelských účtů – Omezení oprávnění uživatelských účtů může minimalizovat možnosti úspěchu útoku a omezit potenciální dopady.

Mezi další možnosti zabezpečení patří: Sanitizace vstupů před ukládáním do databáze, escapování dat při zobrazen, Content Security Policy (CSP), aktivace bezpečnostních hlaviček, ochrana proti Reflective XSS, logování a sledování, automatické odstranění škodlivých dat, bezpečné práce s cookies, omezení oprávnění uživatelských účtů.

4.4 Aplikace ochran

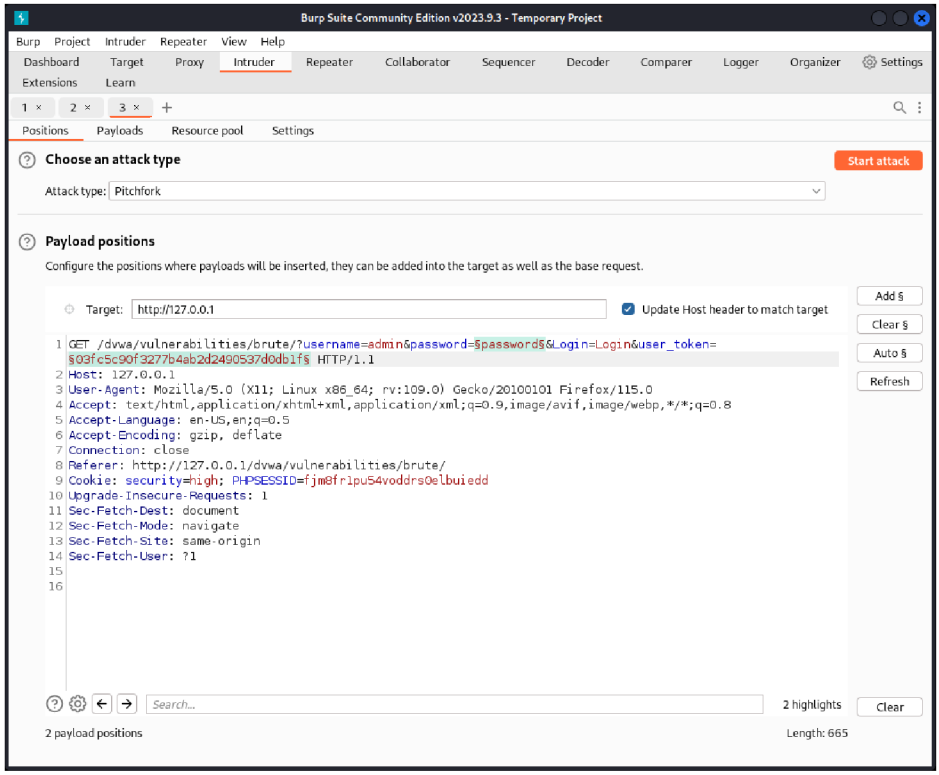
V této části se provádí opětovné útoky na webovou aplikaci DVWA s již použitým návrhem zabezpečení. V jednotlivých částech každé zranitelnosti jsou opět provedeny testy, nyní s již větším zabezpečení, které byly navrženy v části předchozí. Testují se zde i testovací scénáře zda, lze podle nich opět testy provést nebo je potřeba jejich úprava k úspěšnému dokončení testu.

4.4.1 Aplikace ochran webové aplikace

Po aplikaci ochran se opět provádí testy, které by mohli vést k úspěšnému prolomení při zvednutém zabezpečení. Každá zranitelnost, která byla objevena v části zkoumání zranitelností, je zde opět prověřena pokusu o prolomení. V aplikaci DVWA se jedná o možnost zvednutí zabezpečení z „Low“ na „High“.

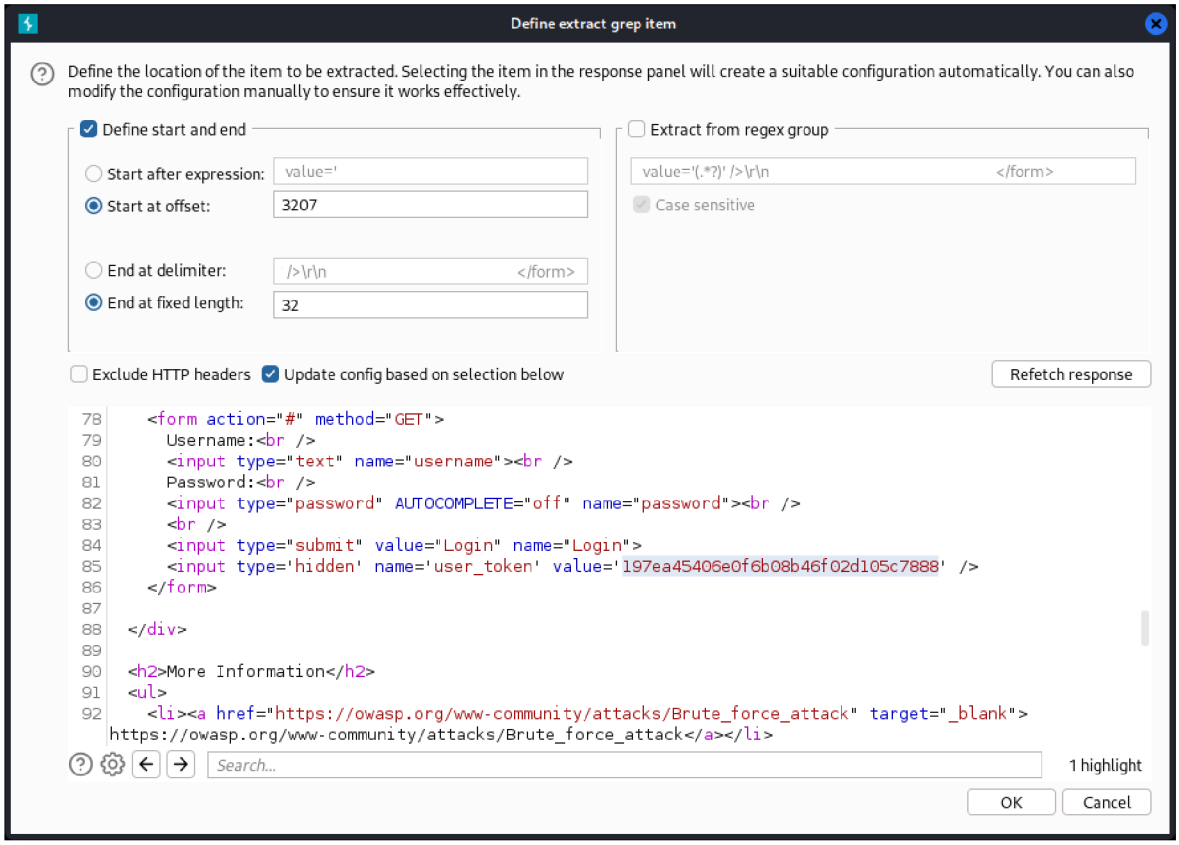
4.4.1.1 Brute force útok po aplikaci ochrany

Po aplikování ochrany některé nástroje jako Hydra nedokáží zabezpečení prolomit, proto je nutné opět použít Burp Suite, který dokáže pracovat s CSRF tokeny. Nejdříve je nutné vybrat heslo a token a přidat je. Zároveň je potřeba přepnout útok na útok typu „Pitchfork“. Jak je vidět na obrázku 26.



Obrázek 26: Burp Suite vyběr míst útoků, Zdroj: vlastní

Po výběrů míst útoku je nutné opět připravit „paylody“. Pro první heslo zůstává postup stejný, tedy opět se budou data sbírat ze souboru john.lst. Jiný postup připadá na token, u kterého je potřeba v nastavení nalézt místo tokenu na webové aplikaci, jak je vidět na obrázku 27.



Obrázek 27: Burp Suite nastavení tokenu, Zdroj: vlastní

Jako poslední je potřeba v záložce Resource pool nastavit maximální počet souběžných žádostí na 1 a spustit útok. Po nějaké době Burp Suite objeví heslo „password“ a identifikátor „03fc5c90f3277b4ab2d2490537d0db1f“ což je vidět opět délkou útoku 4772 oproti ostatním 4734.

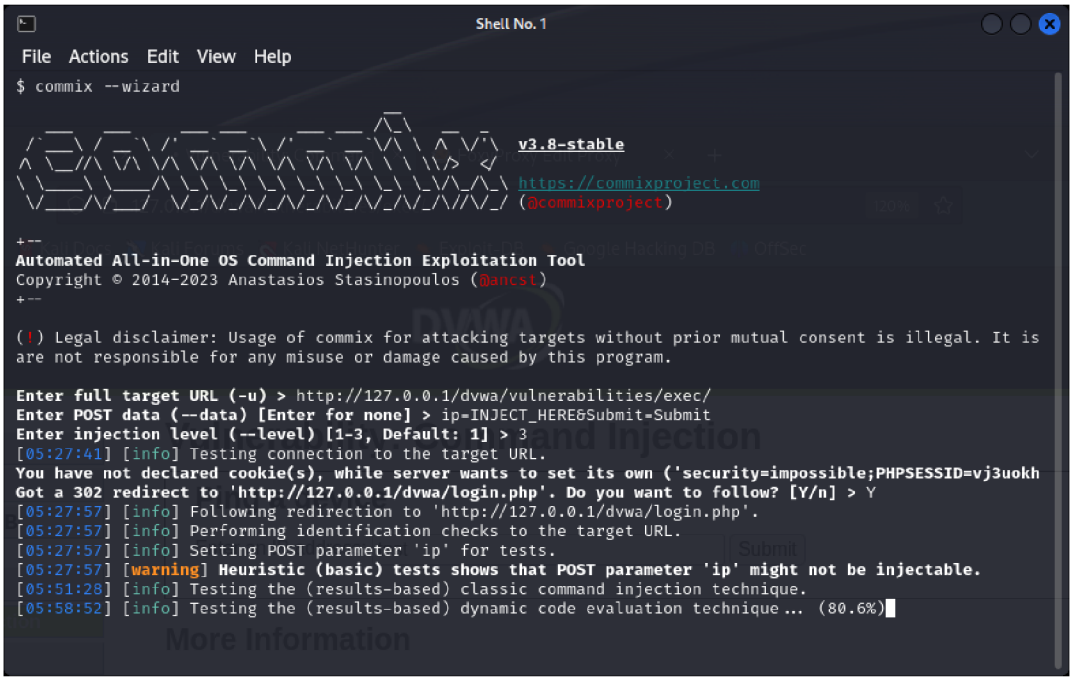
4.4.1.2 Útok Command injection po aplikaci ochrany

Po aplikování ochrany je stále možné vložit IP adresu ale po vložení stejných příkazů nepřichází žádná odpověď. Po prozkoumání zdrojového kódu je ale nalezena zranitelnost v blacklistu, kde parametr ‘| ‘ obsahuje mezeru za znakem, tudíž znak | bez mezery není nijak limitován, jak je vidět obrázku 28.

```php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = trim($_REQUEST[ 'ip' ]);

    // Set blacklist
    $substitutions = array(
        '&'  => '',
        ';'  => '',
        '| ' => '',
        '-'  => '',
        '$'  => '',
        '('  => '',
        ')'  => '',
        '`'  => '',
        '||' => '',
    );

    // Remove any of the characters in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );
```

Obrázek 28: Úryvek zdrojového kódu, Zdroj: Vlastní

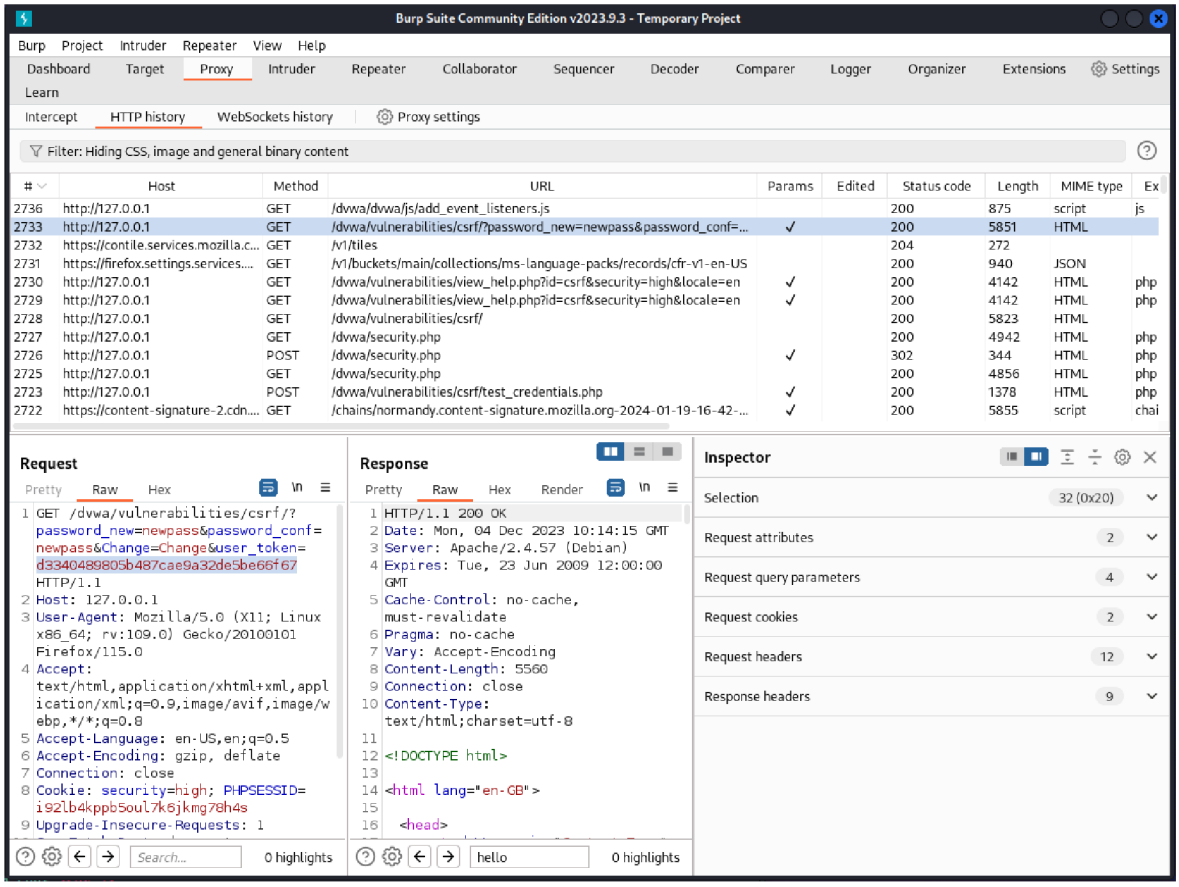
Tudíž po vložení například 127.0.0.1|ipconfig lze příkaz provést. Nebo je tu opět možnost použít nástroj Commix, který je zobrazen na obrázku 29, ale zde se drasticky prodlouží čas proniknutí, nejspíše z limitů samotného nástroje.



Obrázek 29: Průběh prolomení pomocí nástroje Commix, Zdroj: Vlastní

4.4.1.3 Útok CSRF po aplikaci ochran

Po aplikaci ochran už nelze provést útok stejným způsobem, neboť stránka aplikovala několik ochran. Důležitým aspektem útoku je teď získat CSRF token, který lze najít ve zdrojovém kódu nebo pomocí nástroje Burp Suite přes proxy jak je na obrázku 30. Token je vyznačený a lze ho najít pod user_token.



Obrázek 30: Burp Suite – Nalezení tokenu

Nyní pomocí zdrojového kódu lze opět sestavit html stránku s tlačítkem pomocí následujícího formuláře.

```
<form action="http://127.0.0.1/dvwa/vulnerabilities/csrf/" method="GET">
        <br />
        <input type="hidden" value="123" name="password_new"><br />
        <br />
        <input type="hidden" value="123" name="password_conf"><br />
        <br />
        <input type="submit" value="Change" name="Change">
        <input type='hidden' name='user_token' value='d3340489805b487cae9a32de5be66f67' />
</form>
```

Obrázek 31: Formulář, Zdroj: Vlastní

4.4.1.4 Útok File inclusion po aplikaci ochran

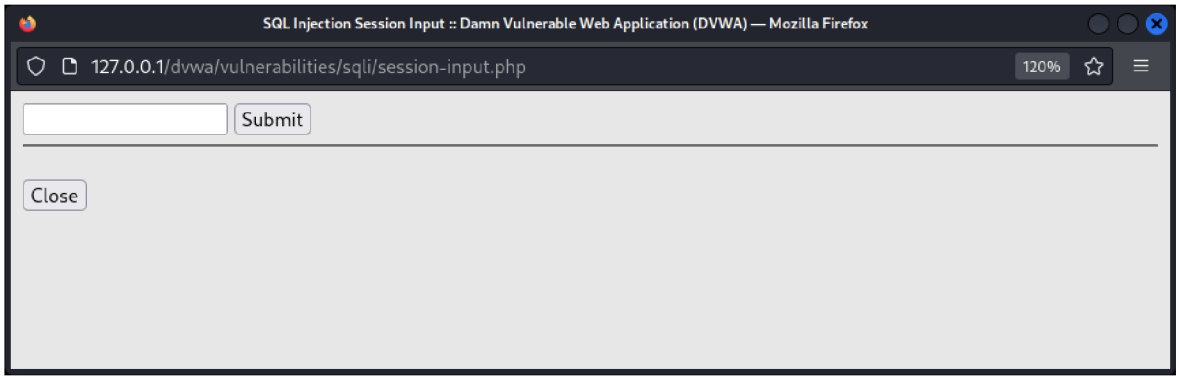
V případě útoku pomocí file inclusion je možné postupovat podobně, proto jsou opět vyzkoušeny předešlé pokusy zakončené: 127.0.0.1/dvwa/vulnerabilities/fi/?page=../../hackable/flags/fi.php. Nicméně po všech pokusech je pokaždé obdržen stejný výsledek, a to ERROR FILE NOT FOUND. Po prozkoumání zdrojového kódu a několika manuálních pokusech je opět možné se dostat

k heslům pomocí správného formulování, které je následující 127.0.0.1/dvwa/vulnerabilities/fi/?page=file:///etc/passwd, viz obrázek 32.

Obrázek 32: Výstup File Inclusion po aplikaci ochran, Zdroj: Vlastní

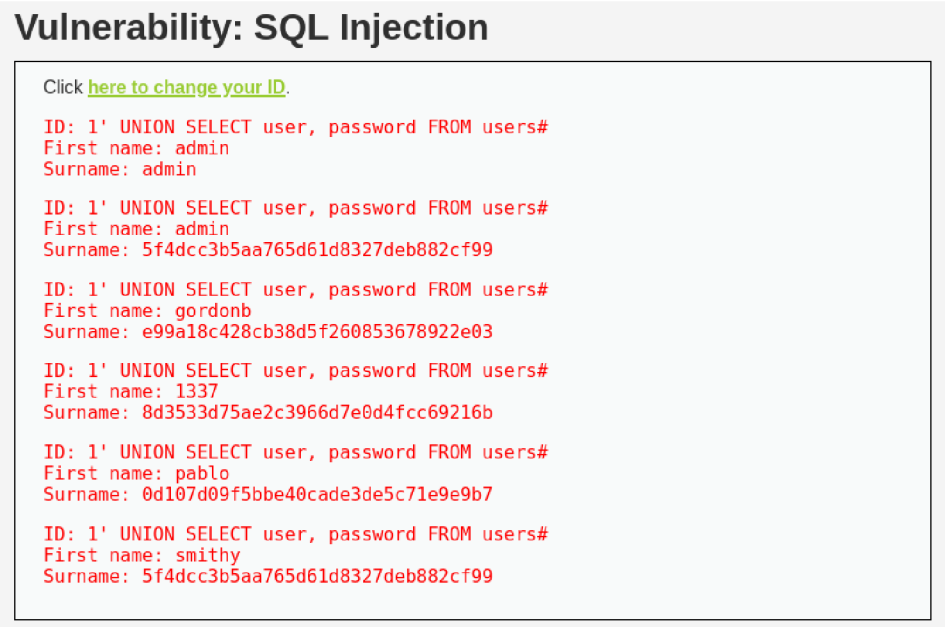
4.4.1.5 SQL injection po aplikaci ochran

Po aplikaci ochran se používá funkce mysql_real_escape_string(). Tato funkce nepovoluje uvozovky v payloadu. Ty ale nejsou potřeba, protože ve zdrojovém kódu, lze zjistit, že pole Id je zpracováváno bez uvozovek.



Obrázek 33: Nové okno pro vyplnění, Zdroj: Vlastní

Další ochranou je otevření nového okna viz obrázek 33. Minulý příkaz zde nefunguje ale je možné vložit následující payload: 1' UNION SELECT user, password FROM users#. Lze to učinit zde, přímo vložit do zdrojového kódu na kartě inspect nebo použitím nástroje Burp Suite.



Obrázek 34: Výstup po vložení příkazi UNION, Zdroj: Vlastní

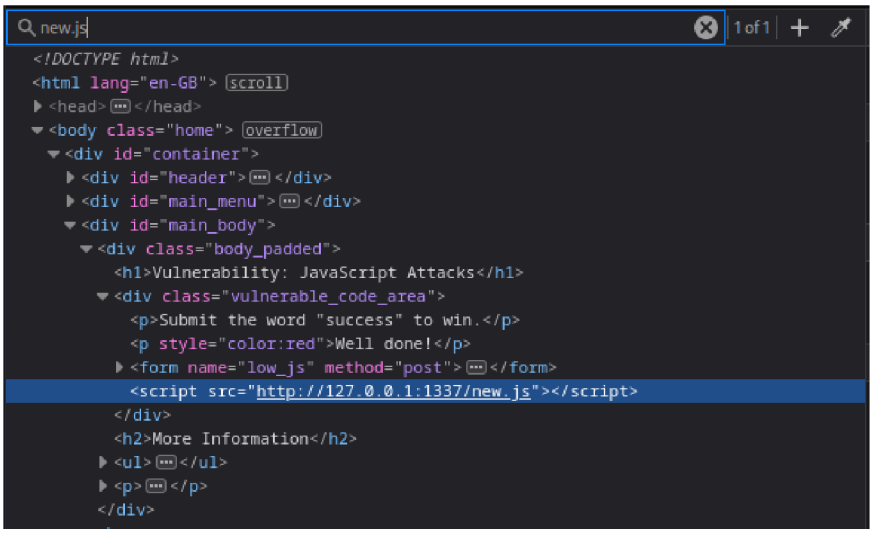
Jak zobrazuje obrázek 34. Po vložení došlo k vypsání loginů a hesel všech uživatelů uložených v databázi. Správným formulováním lze tímto způsobem získat další informace, které jsou uloženy v databázi této webové aplikace.

4.4.1.6 Útok pomocí Javascriptu po aplikaci ochran

Pomocí nástroje Burp Suite je možné odkrýt skrytý token, při změně nastavení Proxy v aplikaci. Zamítnutím File Extension a upravení modifikačních pravidel na odhalení skrytých formulářových polí, jejich okamžité zobrazení, povolení zamítnutých formulářových polí a odstranění limitu pole. Po úpravách a znovu načtení stránky Burp Suite zobrazí skryté pole viz. Obrázek x.

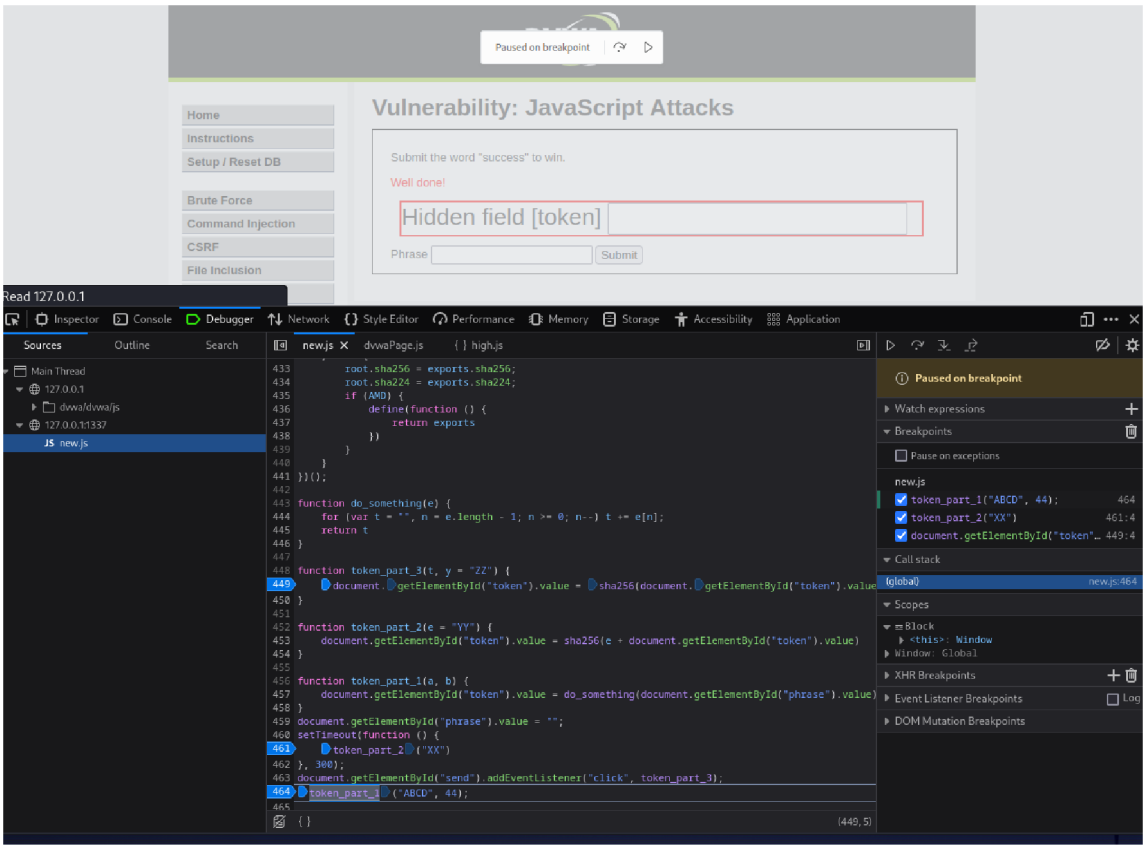
Obrázek 35: Odhalení skrytého pole, Zdroj: Vlastní

Použitím python serveru a aplikace Burp Suite je nahrazen původní js soubor nazvaný high.js novým souborem nazvaným new.js.



Obrázek 36: Zobrazení nahrazeného souboru new.js, Zdroj: Vlastní

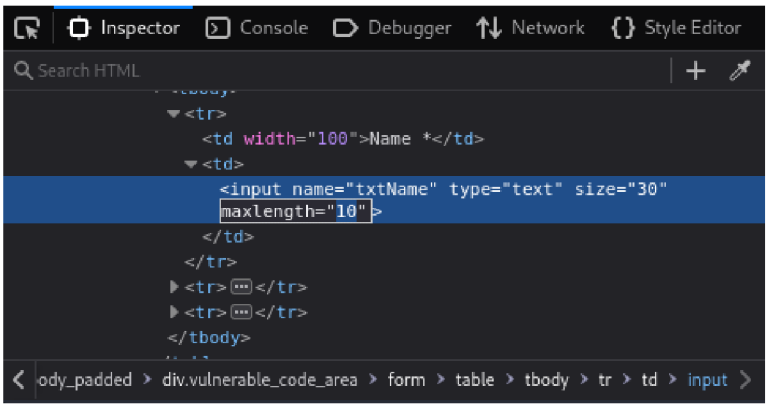
Pomocí vloženého javascriptu, který je velmi podobný tomu, co je na stránce, debuggeru v prohlížeči zobrazeným na obrázku 37 byly nastaveny takzvané breakpointy neboli body přerušení, pomocí kterých byl zjištěn skrytý token ke slovu success a po projití celého procesu došlo k prolomení a je obdržena odpověď „Well Done“



Obrázek 37: Debugger, Zdroj: Vlastní

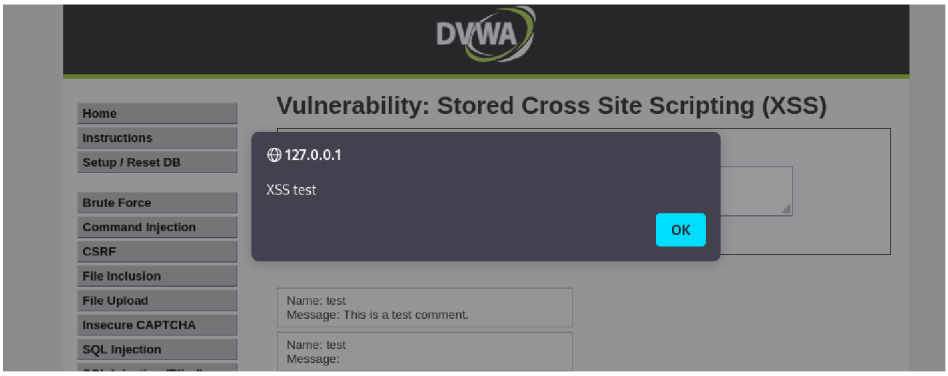
4.4.1.7 XSS útok po aplikaci ochran

První ochranou je limit znaků na pole, který lze jednoduše změnit, jak je vidět na obrázku x nebo pomocí nástroje Burp Suite, který dokáže toto udělat automaticky.



Obrázek 38: Změna délky pole, Zdroj: Vlastní

Skripty jsou kompletně zakázány a používání speciálních znaku taktéž. Proto je nutné to napsat jinak a je tu možnost pomocí následující příkazu: <img src=x onError (`XSS test`)> a výsledek je pak stejný jako v případě žádné ochrany, jak zobrazuje obrázek 39. Zpráva se úspěšně zobrazí po vkročení na onu stránku.



Obrázek 39: Zpráva při vstupu na stránku, Zdroj: Vlastní

4.5 Komparace účinnosti útoků před a po aplikování ochran

Tato část se zaměřuje na porovnání útoků před aplikací ochrany, po ní a po aplikaci dodatečných ochran, které by měli útoku definitivně zamezit. Komparace je prováděna na základě naměřených informací z dřívějších dvou pokusů o prolomení.

Tato část se nejprve zaměřuje na stanovení testovaných aspektů a jejich možnosti. Po tomto následuje samotná komparace, kde jsou tyto aspekty použity. Následuje výběr nejlepších postupů ochrany v každé testované a komparované zranitelnosti, podle kterého se v poslední části zpracovávají výsledky.

4.5.1 Testované aspekty

- Úroveň zabezpečení aplikace DVWA – Jednotlivá zabezpečení jsou rozepsána v průběhu práce při každém útoku nebo návrh ochrany. Testovací aspekt zabezpečení reprezentuje jednotlivá zabezpečení aplikace DVWA, která jsou následující:
    - Low: Aplikace DVWA je zabezpečena minimálně, tedy s nízkou úrovní odolnosti vůči běžným útokům. Bezpečnostní mechanismy jsou buďto úplně zanedbané nebo slabé.
    - Medium: Střední úroveň zabezpečení zahrnuje lepší ochranu než u nízké úrovně, ale mohou zde stále existovat známé zranitelnosti.
    - High: Aplikace DVWA je dobře zabezpečena s důrazem na odolnost vůči širokému spektru útoků.
    - Impossible: Tato úroveň znamená, že je nemožné provést útok kvůli vysoké úrovni zabezpečení aplikace.
- Použitý nástroj/prostředí – Specifikace konkrétního nástroje nebo prostředí, například „Burp Suite“, „Sqlmap“, nebo „Commix“. Každý z těchto nástrojů má své vlastní schopnosti a zaměření na různé typy testování bezpečnosti.
- Úspěšnost – Tento aspekt značí úspěšnost útoku pomocí nominální proměnné:
    - Úspěšný: Útok byl úspěšný a útočník získal neoprávněný přístup nebo nějaký jiný nežádoucí výsledek.
    - Neúspěšný: Obranná opatření aplikace byla úspěšná a útok nebyl proveden nebo byl odražen.
- Doba trvání – Časový rámec, který udává, jak dlouho trval útok. Délka může naznačovat efektivnější obranná opatření nebo vyšší obtížnost útoku.

- Druh/variace útoku – Seznam konkrétních útoků nebo jejich variací, například „SQL injection“, „Cross-Site Scripting (XSS)“, „Cross-Site Request Forgery (CSRF)“, a další.
- Ochranná opatření – Výčet implementovaných bezpečnostních opatření, jako jsou firewally, šifrování dat, silné autentizační metody, filtrování, a další.
- Obtížnost z pohledu testera – Zhodnocení obtížnosti provedení útoku z pohledu osoby tedy testera. Obtížnost je ovlivněna mými schopnostmi, časem věnovaným přípravě, samotného provedení testu, tak i kvalitou implementovaných bezpečnostních opatření. Možné výstupy jsou:
    - Snadná: Útok je snadno proveditelný s minimálním úsilím a znalostmi.
    - Střední: Provedení útoku vyžaduje určitou úroveň znalostí a dovedností, ale není extrémně obtížné.
    - Vysoká: Útok je náročnější a vyžaduje pokročilé dovednosti a znalosti, které jsou potřeba dopředu znát, aby bylo možné test dokončit.
    - Nedosažitelná: Útok nelze provést, protože jsou implementována vysoká ochranná opatření nebo jsou znalosti a dovednosti testera nedostatečné.

4.5.2 Komparace jednotlivých útoků

4.5.2.1 Brute Force

Testovaný aspekt	Výsledky před ochrannými opatřeními	Výsledky po ochranných opatřeních	Výsledky po aplikování dodatečných ochran
Úroveň zabezpečení aplikace DVWA	Low	High	Impossible
Použitý nástroj/prostředí	Burp Suite	Burp Suite	Burp Suite
Úspěšnost	Úspěšný	Úspěšný	Neúspěšný
Přibližná doba trvání	126 minut	205 minut	Test nebyl dokonán z důvodu nezdaru
Druh/variace útoku	Brute force	Brute Force (Pitchfork)	Brute Force (Pitchfork)
Ochranná opatření	N/A	Omezení žádostí, CSRF token, náhodné rozmezí pro chybném pokusu	Zablokování IP adres, dvoufázová autentizace (2FA)
Obtížnost z pohledu testera	Snadná	Střední	Nedosažitelná

Tabulka 1: Komparace Brute Force

4.5.2.2 Command Injection

Testovaný aspekt	Výsledky před ochrannými opatřeními	Výsledky po ochranných opatřeních	Výsledky po aplikování dodatečných ochran
Úroveň zabezpečení aplikace DVWA	Low	High	Impossible
Použitý nástroj/prostředí	Burp Suite a Commix	Burp Suite a Commix	Burp Suite a Commix
Úspěšnost	Úspěšný	Úspěšný	Neúspěšný
Přibližná doba trvání	7m 36s	11m 3s	Test nebyl dokonán z důvodu nezdaru
Druh/variace útoku	Command injection	Command injection	Command injection
Ochranná opatření	N/A	Filtrování speciálních znaků, Whitelist znaků, Omezení příkazů	Ověření syntaxe IPv4, používání parametrizovaných dotazů, implementace firewallu
Obtížnost z pohledu testera	Snadná	Snadná	Nedosažitelná

Tabulka 2: Komparace Command Injection

4.5.2.3 CSRF

Testovaný aspekt	Výsledky před ochrannými opatřeními	Výsledky po ochranných opatřeních	Výsledky po aplikování dodatečných ochran
Úroveň zabezpečení aplikace DVWA	Low	High	Impossible
Použitý nástroj/prostředí	Burp Suite, HTML kód	Burp Suite, HTML kód	Burp Suite, HTML kód
Úspěšnost	Úspěšný	Úspěšný	Neúspěšný
Přibližná doba trvání	6m 12s	15m 8s	Test nebyl dokonán z důvodu nezdaru
Druh/variace útoku	CSRF	CSRF	CSRF
Ochranná opatření	N/A	Ochrana před UNION operátorem, změna typu dotazu, Omezení délky vstupních dat, ANTI-CSRF Token	Použití více než jednoho synchronizačního tokenu, striktní politiky
Obtížnost z pohledu testera	Snadná	Střední	Nedosažitelná

Tabulka 3: Komparace CSRF

4.5.2.4 File inlusion

Testovaný aspekt	Výsledky před ochrannými opatřeními	Výsledky po ochranných opatřeních	Výsledky po aplikování dodatečných ochran
Úroveň zabezpečení aplikace DVWA	Low	High	Impossible
Použitý nástroj/prostředí	DVWA, url	Burp Suite	Burp Suite
Úspěšnost	Úspěšný	Úspěšný	Neúspěšný
Přibližná doba trvání	8m 32s	16m 24s	Test nebyl dokonán z důvodu nezdaru
Druh/variace útoku	File inclusion	File inclusion	File inclusion
Ochranná opatření	N/A	Omezení na povolené protokoly, Filtrace nebo omezení určitých znaků, Ochrana proti nulovému bajtu	Úplné omezení možností zahrnutí souborů
Obtížnost z pohledu testera	Snadná	Snadná	Nedosažitelná

Tabulka 4: Komparace File inclusion

4.5.2.5 SQL injection

Testovaný aspekt	Výsledky před ochrannými opatřeními	Výsledky po ochranných opatřeních	Výsledky po aplikování dodatečných ochran
Úroveň zabezpečení aplikace DVWA	Low	High	Impossible
Použitý nástroj/prostředí	DVWA, Sqlmap	DVWA, Sqlmap	DVWA, Sqlmap
Úspěšnost	Úspěšný	Úspěšný	Neúspěšný
Přibližná doba trvání	10m 48s	15m 14s	Test nebyl dokonán z důvodu nezdaru
Druh/variace útoku	SQL injection	SQL injection	SQL injection
Ochranná opatření	N/A	Double URL Encoding, Nemožnost měnit typ dotazu, Omezení délky vstupních dat, Filtrování klíčových slov	Parametrizované dotazy, validace vstupních dat, whitelist pro vstupní data
Obtížnost z pohledu testera	Střední	Střední	Nedosažitelná

Tabulka 5: Komparace SQL injection

4.5.2.6 Javascript

Testovaný aspekt	Výsledky před ochrannými opatřeními	Výsledky po ochranných opatřeních	Výsledky po aplikování dodatečných ochran
Úroveň zabezpečení aplikace DVWA	Low	High	Impossible
Použitý nástroj/prostředí	Burp Suite	Burp Suite a Firefox debugger	Burp Suite a Firefox debugger
Úspěšnost	Úspěšný	Úspěšný	Neúspěšný
Přibližná doba trvání	19m 35s	31m 40s	Test nebyl dokonán z důvodu nezdaru
Druh/variace útoku	Javascript	Javascript	Javascript
Ochranná opatření	N/A	Content Security Policy, Filtrace speciálních znaků, HTTP Only Cookies	CORS, zakázání eval(), omezení používání globálních proměnných
Obtížnost z pohledu testera	Střední	Vysoká	Nedosažitelná

Tabulka 6: Komparace Javascript

4.5.2.7 XSS

Testovaný aspekt	Výsledky před ochrannými opatřeními	Výsledky po ochranných opatřeních	Výsledky po aplikování dodatečných ochran
Úroveň zabezpečení aplikace DVWA	Low	High	Impossible
Použitý nástroj/prostředí	DVWA	Burp Suite a DVWA	Burp Suite a DVWA
Úspěšnost	Úspěšný	Úspěšný	Neúspěšný
Přibližná doba trvání	6m 15s	11m 37s	Test nebyl dokonán z důvodu nezdaru
Druh/variace útoku	XSS stored	XSS stored	XSS stored
Ochranná opatření	N/A	Ochrana před vložením skriptu do databáze, Omezení HTML tagů a atributů, Aktivace bezpečnostních hlaviček	Content Security Policy (CSP), aktivace bezpečnostních hlaviček, automatické odstranění škodlivých dat
Obtížnost z pohledu testera	Snadná	Střední	Nedosažitelná

Tabulka 7: Komparace XSS

4.5.3 Výběr nejlepších postupů pro ochranu

Při výběru optimálních postupů pro ochranu webových aplikací je nezbytné detailně se zaměřit na každou identifikovanou zranitelnost v rámci testovaného prostředí DVWA navrhnout specifická bezpečnostní opatření. Výběr je prováděn na základě testovacích scénářů, samotného testování a navržených ochran.

4.5.3.1 Brute Force

Brute force útoky jsou častým nástrojem útočníků, kteří se snaží získat neoprávněný přístup na účet uživatele prostřednictvím opakovaných pokusů o uhodnutí hesla. V případě omezení žádostí a nutnosti CSRF tokenu je sice brute force útok zpomalen ale jeho úplnému zablokování nejlépe zabrání zablokování IP adres po překročení určitého počtu neúspěšných pokusům, které je první linií obrany. Nicméně, účinnější ochranou je implementace dvoufázové autentizace (2FA), která vyžaduje od uživatele poskytnutí dvou nezávislých forem identifikace, čímž značně komplikuje úkoly útočníkům a v mém testovaném prostředí naprosto zabraňuje jakékoliv možnosti útoku.

4.5.3.2 Command Injection

Zranitelnost command injection umožňuje útočníkovi spustit neautorizované příkazy na serveru. Aplikace filtrace znaků a použití whitelistu opět útočníka pouze zpomalí. Mnohem ideálnější ochranou je ověření syntaxe IPv4, tedy že útočník může vkládat pouze IP adresu a nic jiného. Posílením použitím parametrizovaných dotazů zamezí dalšímu postupu. Dalšími důležitými opatřeními může být ve firmách rovněž konfigurace firewallu k blokování neoprávněných příkazů, sledování datových toků umožňující rychlou identifikaci potenciálních hrozeb a reakcí na ně.

4.5.3.3 Cross-Site Request Forgery (CSRF)

CSRF útoky mohou vést k provedení nežádoucích akcí v rámci webové aplikace prostřednictvím autorizovaného uživatele. Omezení délky a použití anti CSRF tokenu slouží pro zpomalení útoku. K zamezení se více hodí implementace synchronizačních tokenů, které jsou zahrnuty do formulářů a ověřují legitimnost požadavků. Dalším užitečným posílením může být použití striktní politiky pro ověření požadavků posilující obranu proti CSRF útokům.

4.5.3.4 File Inclusion

Zranitelnost file inclusion může umožnit útočníkovi zahrnout a vykonat škodlivý kód nebo soubory. Pro zpomalení útoku funguje filtrace nebo omezení určitých znaků a ochrana proti nulovému bajtu. Pro úplné zamezení útoku je nezbytné úplně omezit možnosti zahrnutí souborů. Dalšími kroky může být důsledné sledování datových toků a přijmout preventivní opatření k zabránění neoprávněnému přístupu k souborům.

4.5.3.5 SQL Injection

Zranitelnost SQL injection umožňuje útočníkovi injektovat škodlivý SQL kód do dotazů na databázi, což může vést k neoprávněnému získání či modifikaci dat. Omezení délky vstupních dat a filtrování některých klíčových slov jsou opět metody, jak efektivně útok zpomalit. Nicméně implementace parametrizovaných dotazů, striktní validace vstupních dat a escapování znaků jsou klíčovými prvky ochrany. Vytvoření a udržování whitelistu pro vstupní data ztěžuje manipulaci s dotazy a snižuje riziko úspěšného SQL injection útoku.

4.5.3.6 JavaScript

Zranitelnosti spojené s JavaScript mohou být využity pro provádění různých útoků, včetně Cross-Site Scripting (XSS). Pro účinné zpomalení je dobré použít například filtraci speciálních znaků. Ale pro ochranu proti těmto hrozbám je důležité implementovat Content Security Policy (CSP), zakázat rizikové funkce jako eval() a omezit používání globálních proměnných. Tato opatření předcházejí injekci škodlivého kódu do webové stránky.

4.5.3.7 Cross-Site Scripting (XSS)

XSS útoky jsou zaměřeny na vložení škodlivého kódu do webové stránky, který je poté spuštěn v prohlížeči uživatele. Pro účinné zpomalení funguje omezení HTML tagů a atributů a aktivace bezpečnostních hlaviček. Pro účinnou ochranu je nezbytné implementovat Content Security Policy (CSP), aktivovat bezpečnostní hlavičky a automaticky odstraňovat škodlivá data. Tato opatření společně zajišťují, že webová aplikace efektivně odolává různým formám XSS útokům.

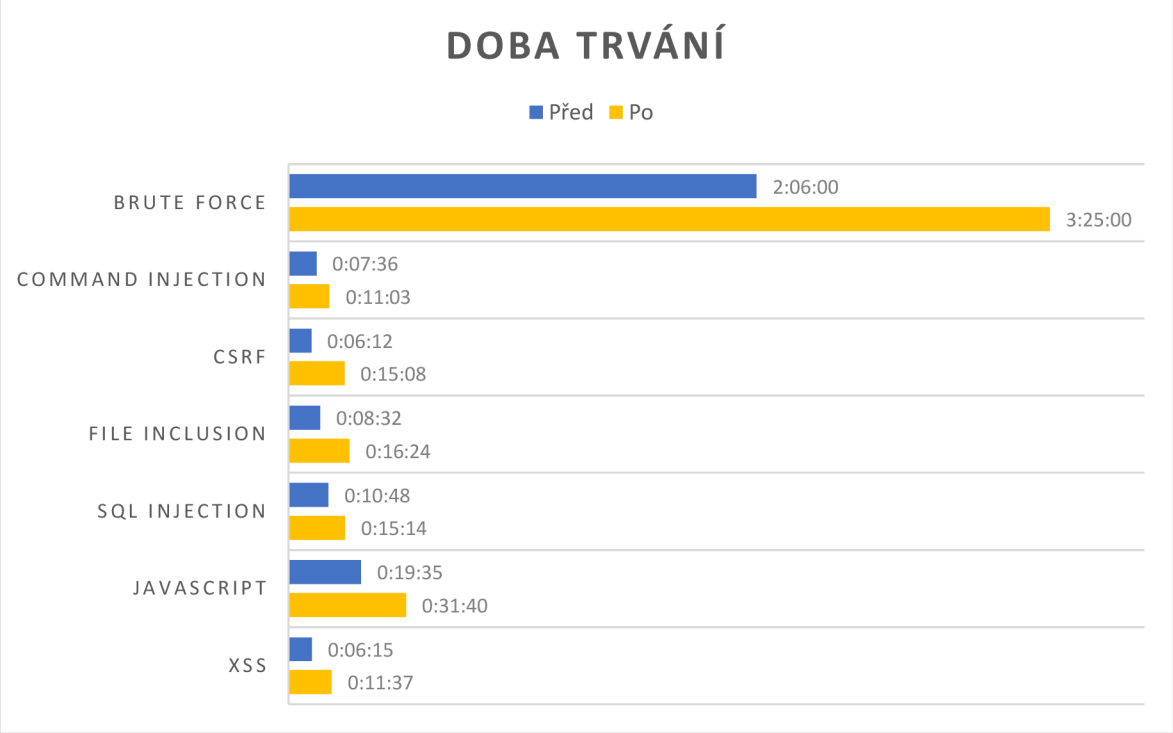
4.5.4 Vyhodnocení

Vyhodnocení a implementace těchto bezpečnostních opatření by měla být součástí celkové strategie zabezpečení webové aplikace. Pravidelné aktualizace, testování a sledování bezpečnostních událostí jsou klíčové pro udržení účinnosti ochranných mechanismů před aktuálními a novými hrozbami. Je nutné ale přijmout to, že všechny tyto metody a návrhy jsou pouze v mezích, které byli dostupné při zpracovávání práce a je tedy možné, ne-li pravděpodobné, že by zkušení odborníci v praxi dokázali tyto ochrany obejít.
Použití těchto postupů by mělo být neustále zdokonalováno na základě vývoje bezpečnostního prostředí a nově identifikovaných zranitelností. Implementace komplexních bezpečnostních strategií je klíčem k udržení integritní a spolehlivé webové aplikace v době neustálého vývoje hrozeb a technologií.

5 Výsledky a diskuse

V této části je provedeno zhodnocení výsledků a následná diskuse, která poskytuje vhled do účinnosti testovaných scénářů během simulace různých typů útoků na webovou aplikaci následované možnostmi pro další testy, které uvádí, co vše by bylo možné v rámci práce dále zpracovávat. Během provádění testů byly získány výsledky, které pomohly lépe porozumět, jak jednotlivé zabezpečovací opatření ovlivňují odolnost aplikace vůči potenciálním hrozbám.

Jak lze sledovat z grafu 1, první zvolený cíl implementace zabezpečení byl úspěšně dosažen. I přes prolomení všech zabezpečovacích mechanismů útočníkům trvalo prolomení systému déle, což svědčí o účinnosti navržených opatření. Časy po aplikaci dodatečných ochran již nejsou psány z důvodů selhání po několika minutách.



Graf 1: Doba trvání útoků

Z analýzy časových údajů před a po provedení různých metod útoků vyplývají některé klíčové závěry týkající se jejich účinnosti a možných dopadů. Prvním významným zjištěním je, že útok metodou brute force má největší časovou náročnost, přičemž dobu jeho provedení zvyšuje o 79 minut. Toto zjištění poukazuje schopnost způsobit významné zpoždění ve funkčnosti systému díky použitým zabezpečení. Je důležité si uvědomit, že ačkoliv může být brute force útok časově náročný, relativně snadno se provádí, pokud nejsou zavedena adekvátní bezpečnostní opatření.

Naopak, metody jako SQL injection a JavaScript také způsobují výrazné prodloužení doby trvání útoku, což naznačuje jejich schopnost ovlivnit funkčnost systému a zranitelnost. Tyto techniky jsou známé pro svou schopnost manipulovat s daty a vyvolávat nežádoucí akce na cílovém systému.

Přestože brute force útok dominuje časově s uvedeným zvýšením o 79 minut, je nutné brát v potaz také složitost útoku jako celek. Nejvyšší složitost představuje útok typu JavaScript, vyžadující pečlivou pozornost a znalost fungování webových stránek a JavaScriptu. Ačkoliv i brute force útok po aplikaci ochranných opatření není triviální, dostupnost penetračních testovacích nástrojů jej značně usnadňuje. Je rovněž důležité zdůraznit, že doba trvání útoku je závislá na použitém zásobníku slov.

Po kombinaci testovaných aspektů s kladeným důrazem na složitost daného útoku a doby trvání vyplývá že nejefektivnější aplikovaná zabezpečení úrovně „high" jsou případě útokem typu JavaScriptu, následovaná Brute force útokem a CSRF. V případě aplikování dodatečných ochran úrovně „impossible" došlo k selhání ve všech případech s použitím stejné metody jako tomu bylo v předešlé obtížnosti, s tím že o útok typu Javascript byl pokoušen nejdéle, a naopak brute force nejmenší dobu.

Nejúčinnější zabezpečení u jednotlivých zranitelností:

- Brute force – Zablokování IP adres, dvoufázová autentizace
- Command injection – Ověření syntaxe IPv4
- CSRF – Použití synchronizačního tokenu
- File Inclusion – Úplné omezení možností zahrnutí souborů
- SQL Injection – Použití parametrizovaných dotazů a whitelistu pro vstupní data
- Javascript - CORS, zakázání eval(), omezení používání globálních proměnných
- XSS – Content Security Policy (CSP) a aktivace bezpečnostních hlaviček

Zjištění 100% účinnosti testovacích scénářů je v případě Command injection, File Inclusion a SQL injection. Ve všech zmíněných variantách toto představuje značný úspěch v rámci zabezpečení těchto konkrétních zranitelností, to znamená, že scénář mohl bát použit v neupravené podobě jak v době před aplikováním zabezpečení, tak i po.

Dále je třeba zdůraznit, že nejefektivnější aplikovaná zabezpečení byla v případě útoků typu Javascript, Brute Force a CSRF představující klíčový aspekt v oblasti prevence kybernetických hrozeb. Konkrétněji, v případě útoku typu Javascript jsou metody jako CORS, zakázání eval() a omezení používání globálních proměnných nezbytné k omezení

rizika zneužití této zranitelnosti. Pro útoky typu Brute Force je implementace opatření jako zablokování IP adres a dvoufázová autentizace nezbytná k efektivní ochraně proti neoprávněným pokusům o přihlášení. V případě CSRF je klíčovou ochrannou vrstvou použití synchronizačního tokenu, který zabrání neautorizovanému odesílání žádostí ze strany útočníka.

Celkově lze tedy konstatovat, že úspěšnost testovacích scénářů a efektivnost aplikovaných zabezpečení představuje kritický prvek v boji proti kybernetickým hrozbám a zachování integrity a bezpečnosti informačních systémů. V případě udržování nějaké webové aplikace je tedy nezbytné tyto opatření aktuální a reagovat na nové typy hrozeb a zranitelností, aby byla zajištěna co nejvyšší míra ochrany a bezpečnosti dat.

5.1.1 Možnosti dalších testů

V průběhu této studie byly identifikovány zranitelnosti, z nichž část byla vyhodnocena jako kritická pro bezpečnost webových aplikací. Tato kritéria byla důkladně analyzována a následně bylo rozhodnuto, které konkrétní zranitelnosti budou dále zkoumány v rámci testování. Tento selektivní přístup byl zdůvodněn několika faktory. Za prvé, existovala podobnost některých typů útoků, což umožnilo koncentrovat se na specifické hrozby. Za druhé, samotná aplikace DVWA poskytuje řadu možností pro testování zranitelností, avšak z důvodů efektivity bylo nutné vybrat konkrétní scénáře a nezkoumat každou potenciální slabost zvlášť, což by vedlo k zdvojení rozsahu práce. Mezi vybranými možnostmi dalších testů patří File Upload, Insecure CAPTCHA, Blind SQL Injection, Weak Session IDs, XSS (Doomed), XSS (Reflected), CSP Bypass, Authorization Bypass a Open Redirect. Tato selekce není omezena pouze na aplikaci DVWA, ale může být aplikována i na další podobné aplikace či webové platformy.

Dalším významným krokem v rámci této práce by mohlo být rozšíření testování na reálnou aplikaci nějaké firmy. To by umožnilo aplikovat získané znalosti a metodiky na konkrétní pracovní prostředí a ověřit jejich praktickou účinnost v kontextu firemní bezpečnosti. Takový přístup by mohl poskytnout užitečné informace pro organizace, které chtějí proaktivně posilovat své bezpečnostní opatření a minimalizovat rizika spojená se zranitelnostmi webových aplikací.

6 Závěr

Tato práce se zaměřila na problematiku etického hackingu, bezpečnosti webových aplikací a jejich ochranu před různými formami útoků. Byla provedena analýza rizik, jimž jsou uživatelé vystaveni, a bezpečnostních opatření webových aplikací.

V teoretické části práce byly pečlivě rozebrány klíčové aspekty etického hackingu. Důraz byl kladen na pochopení různých typů útoků a jejich metodiky, jakož i na identifikaci strategií ochrany a prevence. Zvláštní pozornost byla věnována problematice bezpečnosti webových aplikací a technologiím používaným k jejich ochraně.

Praktická část práce zahrnovala tvorbu prostředí pro testování a vytváření testovacích scénářů. Prostředí bylo postaveno na platformě Virtual Box s operačním systémem Kali Linux. Následovalo navržení a implementace testovacích scénářů. Pro testování byly využity volně dostupné nástroje pro etický hacking, jako je Burp Suite, Sqlmap a Commix. Provedené testy byly detailně analyzovány a porovnány s cílem vybrat nejefektivnější postupy pro ochranu webových aplikací. Díky praktické demonstraci bylo dosaženo hlubšího porozumění zranitelnostem webových aplikací a efektivním způsobům, jak jim předcházet.

Výsledky práce jasně ukázaly, že správně navržené testovací scénáře a adekvátní ochranná opatření mohou výrazně zvýšit úroveň bezpečnosti webových aplikací a minimalizovat riziko úspěšného útoku. Závěry z této práce poskytují cenný přínos pro začátečníky v oblasti kybernetické bezpečnosti, kteří mohou využít získané poznatky.

Nicméně je důležité si uvědomit, že kybernetická bezpečnost je dynamické pole a nové hrozby a techniky útoků se neustále vyvíjejí. Proto je nezbytné nadále sledovat aktuální trendy a inovace v oblasti bezpečnosti webových aplikací a pružně reagovat na nové výzvy. Což je i vidět v aplikaci DVWA, jejíž možnosti se každý rok rozšiřují a poskytují nové možnosti pronikávat do systému. Diplomová práce poskytuje pevný základ pro další výzkum a rozvoj v této klíčové oblasti informačních technologií.

8 Seznam použitých zdrojů

Acunetix. 2024. What is SQL Injection (SQLi) and How to Prevent It. *Acunetix.* [Online] 9. 1 2024. https://www.acunetix.com/websitesecurity/sql-injection/.
Beatrice, Adilin. 2020. Top 10 Proactive Web Application Security Measures. *analyticsinsight.* [Online] 25. 11 2020. https://www.analyticsinsight.net/top-10-proactive-web-application-security-measures/#:~:text=Ten%20ways%20to%20ensure%20web%20application%20security%201,Scan%20your%20website%20for%20vulnerabilities%20...%20Dal%C5%A1%C3%AD%20polo%C5%BEky.
Bisson, Marilyn. 2023. 2 Ways To Install and Use DVWA On Windows 10. *eldernode.* [Online] 1. 1 2023. 2 Ways To Install and Use DVWA On Windows 10.
Buxton, Oliver. 2022. What are the three main types of hackers? *Avast.* [Online] 12. 10 2022. https://www.avast.com/c-hacker-types.
Campbell, Kern. 2022. Is It Safe To Install VirtualBox? *thegadgetbuyer.* [Online] 9. 11 2022. https://thegadgetbuyer.com/potential-risks-of-virtualbox/.
Cloudflare. 2023. What is web application security? *Cloudflare.* [Online] 2023. What is web application security?.
Comptia. 2020. DVWA - Manual SQL Injection and Password. *Cybersecurityhoy.* [Online] 8 2020. https://cybersecurityhoy.files.wordpress.com/2020/08/lab16-dvwa-manual-sql-injection-and-password-cracking.pdf.
—. 2023. What Is Ethical Hacking? *Comptia.* [Online] 2023. https://www.comptia.org/content/articles/what-is-ethical-hacking.
Cooper, Stephen. 2023. Burp Suite Review & the Best Alternatives. *Comparitech.* [Online] 9. 2 2023. https://www.comparitech.com/net-admin/burp-suite-review/.
Coursera. 2023. What Is Ethical Hacking? *Coursera.* [Online] 17. 5 2023. https://www.coursera.org/articles/what-is-ethical-hacking.
Crowdstrike. 2023. APPLICATION SECURITY: CHALLENGES, TOOLS & BEST PRACTICES. *crowdstrike.* [Online] 28. 3 2023. https://www.crowdstrike.com/cybersecurity-101/application-security/.
Das, Ankush. 2022. Is VirtualBox Safe or Is It a Security Risk? *makeuseof.* [Online] 31. 8 2022. https://www.makeuseof.com/is-virtualbox-safe-or-security-risk/.
DianApps. 2019. Ethical Hacking and its Methodology. *Medium.* [Online] 18. 10 2019. https://medium.com/@DianApps/ethical-hacking-and-its-methodology-41468bc2ea67.
Dvořáková, Lucie. 2023. Co je to etický hacking? Poznejte rozdíl mezi White Hat a Black Hat. *l-a-b-a.* [Online] 20. 4 2023. https://l-a-b-a.cz/blog/565-co-je-to-eticky-hacking-poznejte-rozdil-mezi-white-hat-a-black-hat.
ECQ. 2023. Scenario-based Penetration Test. *e-cq.* [Online] 2023. https://www.e-cq.net/scenario-based-penetration-test.html.
Eset. 2023. Zero day útok. *Eset.* [Online] 2023. https://www.eset.com/cz/zero-day/.
E-spin. 2021. Burp Suite Pro vs Enterprise what the differences. *e-spincorp.* [Online] 19. 11 2021. https://www.e-spincorp.com/burp-suite-pro-vs-enterprise-what-the-differences/.
European Parlament. 2023. Fighting cybercrime: new EU cybersecurity laws explained. *European Parlament.* [Online] 13. 9 2023. https://www.europarl.europa.eu/news/en/headlines/security/20221103STO48002/fighting-cybercrime-new-eu-cybersecurity-laws-explained.
Fagun, Mejbaur Bahar. 2023. SQLmap: A Powerful Tool for Hackers and Essential for SQA Testers. *https://www.linkedin.com/pulse/sqlmap-powerful-tool-hackers-essential-sqa-*

testers-fagun. [Online] 14. 7 2023. https://www.linkedin.com/pulse/sqlmap-powerful-tool-hackers-essential-sqa-testers-fagun.
Froehlich, Andrew. 2021. White hat hacker. *Techtarget.* [Online] 29. 12 2021. https://www.techtarget.com/searchsecurity/definition/white-hat.
Fruhlinger, Josh. 2021. Penetration testing explained: How ethical hackers simulate attacks. *csoonline.* [Online] 10. 12 2021. https://www.csoonline.com/article/571697/penetration-testing-explained-how-ethical-hackers-simulate-attacks.html.
Garcia, Ramiro. 2023. DVWA. *github.* [Online] 29. 7 2023. https://github.com/digininja/DVWA.
Gillam, Jason. 2023. What are the ethical and legal considerations for penetration testing? *secureideas.* [Online] 9. 3 2023. https://www.secureideas.com/knowledge/what-are-the-ethical-and-legal-considerations-for-penetration-testing.
Glas, Brian, a další. 2021. Top 10 Web Application Security Risks. *owasp.* [Online] 2021. https://owasp.org/www-project-top-ten/.
Grimes, Roger A. 2017. *Hacking the Hacker: Learn from the Experts Who Take Down Hackers.* Indianapolis : John Wiley & Sons, Inc., 2017. ISBN: 978 1 119 39621 5.
Gutheil, Mirja , a další. 2017. Policy Department C: Citizens' Rights and Constitutional Affairs. *europarl.europa.* [Online] 3 2017. https://www.europarl.europa.eu/RegData/etudes/STUD/2017/583137/IPOL_STU(2017)583137_EN.pdf.
Hasenmueller, Achim. 2023. Welcome to VirtualBox.org! *VirtualBox.* [Online] Oracle, 18. 7 2023. https://www.virtualbox.org/.
Hertzog, Raphaël, Gorman, Jim a Aharoni, Mati. 2017. *Kali Linux Revealed.* místo neznámé : Offsec Press, 2017. 978-0-9976156-0-9.
honeyteksystems. 2023. What is Scenario-Based Testing? *Honeyteksystems.* [Online] 2023. https://honeyteksystems.com/services/assess/scenario-based-testing/.
Choudhary, Anshul. 2023. SQLmap — A Comprehensive Guide For Begineers. *Medium.* [Online] 5. 8 2023. https://medium.com/@anshulchoudhary227/sqlmap-a-comprehensive-guide-for-begineers-f0ecd75f11ad.
Imperva. 2023. Database Security. *Imperva.* [Online] 20. 12 2023. https://www.imperva.com/learn/data-security/database-security/.
Javatpoint. 2023. Ethical Hacking Tutorial. *Javatpoint.* [Online] 2023. https://www.javatpoint.com/ethical-hacking.
—. 2023. What is a Web Application? *Javatpoint.* [Online] 2023. https://www.javatpoint.com/web-application.
Jethva, Hitesh. 2023. Burp Suite Review and Alternatives. *Netadmintools.* [Online] 23. 1 2023. https://www.netadmintools.com/burp-suite-review#wbounce-modal.
Kanumilli, Prathima. 2022. What is Ethical Hacking? Definition, Basics, Types, Attacks Explained. *henryharvin.* [Online] 25. 11 2022. https://www.henryharvin.com/blog/what-is-ethical-hacking/.
Kaspersky. 2018. What is a Black-Hat hacker? *Kaspersky.* [Online] 31. 8 2018. https://www.kaspersky.com/resource-center/threats/black-hat-hacker.
Ken, Leon. 2023. DVWA Penetration Testing Report. *Systemweakness.* [Online] 1. 1 2023. https://systemweakness.com/dvwa-penetration-testing-report-2c7beb3395ff.
Kim, Peter. 2018. *The hacker playbook 3.* místo neznámé : Secure Planet LLC, 2018. 978-1980901754.
KirstenS. 2024. Cross Site Scripting (XSS). *Owasp.* [Online] 2024. https://owasp.org/www-community/attacks/xss/.

Kost, Edward. 2023. What is Wireshark? The Free Network Sniffing Tool. *Upguard.* [Online] 6. 4 2023. https://www.upguard.com/blog/what-is-wireshark.
Krishna, Ananda. 2021. 7 Web Application Security Practices You Can Use. *thesslstore.* [Online] 27. 9 2021. https://www.thesslstore.com/blog/web-application-security-practices-you-can-use/.
Legal Desire. 2017. Ethical Hacking and It's Legality. *legaldesire.* [Online] 10. 10 2017. https://legaldesire.com/ethical-hacking-legality/.
Linhart, Jiří a Vodáková, Alena. 2017. Metoda srovnávací. *encyklopedie.* [Online] 11. 12 2017. https://encyklopedie.soc.cas.cz/w/Metoda_srovn%C3%A1vac%C3%AD.
Mishra, Alok. 2022. Attributes impacting cybersecurity policy development: An evidence from seven nations. *Sciencedirect.* [Online] 9 2022. https://www.sciencedirect.com/science/article/pii/S0167404822002140.
Norton, Alan T. 2018. *Computer Hacking Beginners Guide: How to Hack Wireless Network, Basic Security and Penetration Testing, Kali Linux, Your First Hack.* San Francisco : Independently published, 2018. 9781980390978.
Olaogun, Daniel. 2023. Creating Effective Test Scenarios: Best Practices and Tips for Successful Testing. *saucelabs.* [Online] 19. 5 2023. https://saucelabs.com/resources/blog/creating-effective-test-scenarios-best-practices-and-tips-for-successful.
Oracle. 2016. Compare test case results with expected results. *Oracle.* [Online] 6. 10 2016. https://docs.oracle.com/html/E79061_01/Content/Test%20cases/Compare_test_case_results_with_expected_results.htm.
Portswigger. 2023. Burp Suite Enterprise Edition. *Portswigger.* [Online] 2023. https://portswigger.net/burp/enterprise.
—. 2023. What do you want to do with Burp Suite? *Portswigger.* [Online] 2023. https://portswigger.net/burp.
Poston, Howard. 2020. What are black box, grey box, and white box penetration testing? *infosecinstitute.* [Online] 11. 8 2020. https://resources.infosecinstitute.com/topics/penetration-testing/what-are-black-box-grey-box-and-white-box-penetration-testing/.
Praveen. 2023. Ethical Hacking: Understanding the Basics. *eccouncil.* [Online] 12. 5 2023. https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/ethical-hacking-understanding-basics/.
Privacysense. 2023. Green Hat Hacker. *Privacysense.* [Online] 14. 1 2023. https://www.privacysense.net/terms/green-hat-hacker/.
ptsecurity. 2017. Corporate information system penetration testing: attack scenarios. *ptsecurity.* [Online] 26. 5 2017. https://www.ptsecurity.com/ww-en/analytics/corporate-information-system-penetration-testing/.
Qamadness. 2023. Test Scenario: Definition, Purpose, and How to Create. *Qamadness.* [Online] 3. 4 2023. https://www.qamadness.com/knowledge-base/test-scenario-definition-purpose-and-how-to-create/.
Rahalkar, Sagar. 2021. *A Complete Guide to Burp Suite:.* Pune : Apress, 2021. 978-1-4842-6402-7.
Rau, Thomas. 2023. Pět nejnebezpečnějších útoků na bezdrátové sítě. Jak se před nimi ochránit? *Computerworld.* [Online] 27. 11 2023. https://www.computerworld.cz/clanky/pet-nejnebezpecnejsich-utoku-na-bezdratove-site-vite-jak-se-pred-nimi-ochranit/.

Redscan. 2022. SCENARIO-BASED TESTING. *Redscan.* [Online] 8. 7 2022. https://www.redscan.com/services/scenario-based-testing/.
Řehka, Ing. Karel. 2020. 2020 REPORT ON CYBER SECURITY IN THE CZECH REPUBLIC. *Nukib.* [Online] 2020. https://www.nukib.cz/download/publications_en/2020_report_on_cyber_security_in_the_czech_republic.pdf.
Sabih, Zaid. 2018. *Learn Ethical Hacking.* Birmingham : Packt Publishing Ltd., 2018. 978-1-78862-205-9.
securify. 2022. SCENARIO BASED PENTEST. *securify.* [Online] 26. 8 2022. https://www.securify.nl/en/services/scenario-based-pentest/.
Sengupta, Sudip. 2021. What are the Five Steps of Ethical Hacking. *crashtest-security.* [Online] 22. 10 2021. https://crashtest-security.com/five-steps-of-ethical-hacking/.
Shariq, Mohd. 2022. Commix – OS Command Injection and Exploitation Tool. *Geeksforgeeks.* [Online] 21. 11 2022. https://www.geeksforgeeks.org/commix-os-command-injection-and-exploitation-tool/.
Sharpe, Richard, Warnicke, Ed a Lamping, Ulf. 2023. Wireshark User's Guide. *Wireshark.* [Online] 2023. https://www.wireshark.org/docs/wsug_html_chunked/.
Shea, Sharon. 2019. 6 different types of hackers, from black hat to red hat. *Techtarget.* [Online] 10 2019. https://www.techtarget.com/searchsecurity/answer/What-is-red-and-white-hat-hacking.
Shivanandhan, Manish. 2020. What is Nmap and how to use it. *Freecodecamp.* [Online] 2. 11 2020. https://www.freecodecamp.org/news/what-is-nmap-and-how-to-use-it-a-tutorial-for-the-greatest-scanning-tool-of-all-time/https://www.freecodecamp.org/news/what-is-nmap-and-how-to-use-it-a-tutorial-for-the-greatest-scanning-tool-of-all-time/.
Shukla, Ankit. 2019. Laws you need to know as an Ethical Hacker. *ipleaders.* [Online] 4. 1 2019. https://blog.ipleaders.in/laws-need-know-ethical-hacker/.
Simplilearn. 2023. Top 30+ Ethical Hacking Tools and Software You Need to Be Aware of in 2023. *Simplilearn.* [Online] 2. 8 2023. https://www.simplilearn.com/top-5-ethical-hacking-tools-rar313-article.
Singh, Glen D. 2022. *The Ultimate Kali Linux Book: Second Edition.* Birmingham : Packt Publishing, 2022. 978-1-80181-893-3.
Singh, Satyam. 2018. Important SQLMap commands. *Resources.* [Online] 17. 7 2018. https://resources.infosecinstitute.com/topics/penetration-testing/important-sqlmap-commands/.
Sinha, Sanjib. 2017. *Beginning Ethical Hacking with Python.* Howrah : Apress, 2017. 978-1-4842-2540-0.
Skillmea. 2021. Co je etický hacking? *Skillmea.* [Online] 20. 12 2021. https://skillmea.cz/blog/co-je-eticky-hacking.
Software Testing. 2023. How do you evaluate and compare test results from test and production environments? *Linkedin.* [Online] 22. 6 2023. https://www.linkedin.com/advice/1/how-do-you-evaluate-compare-test-results-from.
Stasinopoulos, Anastasios. 2023. Commix. *Github.* [Online] 11. 9 2023. https://github.com/commixproject/commix.
Synopsys. 2023. Ethical Hacking. *Synopsys.* [Online] 29. 3 2023. https://www.synopsys.com/glossary/what-is-ethical-hacking.html.
Techspot. 2023. Kali Linux 2023.2. *techspot.* [Online] 2023. https://www.techspot.com/downloads/6738-kali-linux.html#certified.

Terra, John. 2023. White Hat Hacker: The What, Why and How. *Simplilearn.* [Online] 7. 8 2023. https://www.simplilearn.com/white-hat-hacker-article.
The Cyber Swiss Army Knife. 2016. *CyberChef.* [Online] 2016. https://gchq.github.io/CyberChef/.
Thefastcode. 2023. Co jsou útoky typu Denial of Service a DDoS? *Thefastcode.* [Online] 2. 4 2023. https://www.thefastcode.com/cs-czk/article/what-are-denial-of-service-and-ddos-attacks.
Tight, Ben. 2023. The Invisible Risk of Oracle VirtualBox. *Metrixdata360.* [Online] 23. 6 2023. https://metrixdata360.com/license-series/oracle-virtualbox-risks/.
Trevino, Aranza. 2023. The Three Different Types of Hackers. *Keepersecurity.* [Online] 26. 6 2023. https://www.keepersecurity.com/blog/2023/06/26/the-three-different-types-of-hackers/.
Tripathy, Susnigdha. 2023. What Is Ethical Hacking in Cybersecurity? Ultimate Guide. *Enterprisenetworkingplanet.* [Online] 22. 8 2023. https://www.enterprisenetworkingplanet.com/security/what-is-ethical-hacking/.
Tse, David. 2021. Damn Vulnerable Web Application(DVWA) — SQL Injection Walkthrough. *Medium.* [Online] 14. 1 2021. https://dtwh.medium.com/damn-vulnerable-web-application-dvwa-sql-injection-walkthrough-9c486a2178be.
Volle, Adam. 2022. Web application. *Britannica.* [Online] 6. 10 2022. https://www.britannica.com/topic/Web-application.
Wallen, Jack. 2017. VirtualBox: A cheat sheet. *techrepublic.* [Online] 16. 10 2017. https://www.techrepublic.com/article/virtualbox-everything-the-pros-need-to-know/.
Wallis, Jerry. 2023. 10 Web Application Security Risks & How To Deal With Them. *webo.digital.* [Online] 8. 2 2023. https://webo.digital/blog/10-web-application-security-risks/.
Whittly, Michael. 2020. OWASP Top 10 and DVWA. *Medium.* [Online] 30. 9 2020. https://levelup.gitconnected.com/ethical-hacking-part-1-owasp-top-10-and-dvwa-3f2d55580ba8.
Williams, Lawrence. 2023. 20 Best Ethical Hacking Tools & Software (Jul 2023 Update). *Gru99.* [Online] 8. 7 2023. https://www.guru99.com/learn-everything-about-ethical-hacking-tools-and-skills.html.
Wilson, Ben. 2023. What is Kali Linux? *Kali.* [Online] 6. 3 2023. https://www.kali.org/docs/introduction/what-is-kali-linux/.
Xu, Tammy. 2021. 13 Best Practices for Improving Web Application Security. *Builtin.* [Online] 25. 1 2021. https://builtin.com/software-engineering-perspectives/web-application-security.
Yadav, Ajay. 2020. NMAP Cheat Sheet. *Tutorialspoint.* [Online] 18. 3 2020. https://www.tutorialspoint.com/nmap-cheat-sheet.
Zerodaysnoop. 2020. Virtual Machines & Cyber Security. *zerodaysnoop.* [Online] 22. 5 2020. https://www.zerodaysnoop.com/tools/virtual-machines/virtual-machines-cyber-security/.

9 Seznam obrázků, tabulek, grafů a zkratek

9.1 Seznam obrázků

Obrázek 1: Prostředí VirtualBoxu, Zdroj: Vlastní 48
Obrázek 2: Upravení množství paměti RAM, Zdroj: Vlastní 50
Obrázek 3: Upravení množství procesoru, Zdroj: Vlastní 51
Obrázek 4: Operační systém Kali Linux, Zdroj: Vlastní 51
Obrázek 5: Nastavení přihlašovacích údajů 52
Obrázek 6: Rozhraní aplikace DVWA 53
Obrázek 7: Nastavení úrovni 54
Obrázek 8: Burp Suite Request/Response, Zdroj: Vlastní 57
Obrázek 9: Burp Suite Intruder, Zdroj: vlastní 64
Obrázek 10: Burp Suite Intruder – průběh útoku, Zdroj: Vlastní 64
Obrázek 11: Prolomení hesla, Zdroj: vlastní 65
Obrázek 12: Výstup po zadání příkazu ls, Zdroj:Vlastní 66
Obrázek 13: Commix výstup, Zdroj: Vlastní 66
Obrázek 14: HTML kód k vytvoření falešné stránky, Zdroj: Vlastní 67
Obrázek 15: Burp Suite – zachycení hesla, Zdroj: Vlastní 67
Obrázek 16: Potvzení hesla, Zdroj: Vlastní 68
Obrázek 17: Výstup File Inclusion, Zdroj: Vlastní 68
Obrázek 18: Výstup aplikace DVWA po zadání vstupu, Zdroj: Vlastní 69
Obrázek 19: Sqlmap – výstup po prolomení, Zdroj: Vlastní 70
Obrázek 20: Sqlmap – nalezení souborů, Zdroj: Vlastní 70
Obrázek 21: Sqlmap – nalezení hesel, Zdroj: Vlastní 71
Obrázek 22: Burp Suite – Nalezení tokenu, Zdroj: Vlastní 71
Obrázek 23: Odkrytí skrytého pole, Zdroj: Vlastní 72
Obrázek 24: Vygenerovaný token ze stránky CyberChef, Zdroj: (The Cyber Swiss Army Knife, 2016) 72
Obrázek 25: Zpráva při vstupu, Zdroj: Vlastní 73
Obrázek 26: Burp Suite vyběr míst útoků, Zdroj: vlastní 77
Obrázek 27: Burp Suite nastavení tokenu, Zdroj: vlastní 78
Obrázek 28: Úryvek zdrojového kódu, Zdroj: Vlastní 79
Obrázek 29: Průběh prolomení pomocí nástroje Commix, Zdroj: Vlastní 79
Obrázek 30: Burp Suite – Nalezení tokenu 80
Obrázek 31: Formulář, Zdroj: Vlastní 80
Obrázek 32: Výstup File Inclusion po aplikaci ochran, Zdroj: Vlastní 81
Obrázek 33: Nové okno pro vyplnění, Zdroj: Vlastní 81
Obrázek 34: Výstup po vložení příkazi UNION, Zdroj: Vlastní 82
Obrázek 35: Odhalení skrytého pole, Zdroj: Vlastní 82
Obrázek 36: Zobrazení nahrazeného souboru new,js, Zdroj: Vlastní 83
Obrázek 37: Debugger, Zdroj: Vlastní 83
Obrázek 38: Změna délky pole, Zdroj: Vlastní 84
Obrázek 39: Zpráva při vstupu na stránku, Zdroj: Vlastní 84

9.2 Seznam tabulek

Tabulka 1: Komparace Brute Force 87
Tabulka 2: Komparace Command Injection 87
Tabulka 3: Komparace CSRF 88
Tabulka 4: Komparace File inclusion 88
Tabulka 5: Komparace SQL injection 89
Tabulka 6: Komparace Javascript 89
Tabulka 7: Komparace XSS 90

9.3 Seznam grafů

Graf 1: Doba trvání útoků 93

Přílohy

Příloha 1 - Testovací scénář 1 pro Brute force útok

- Výběr nástroje:
  - Otevřete nástroj Burp Suite, který je k dispozici v Kali Linux.
- Definice cíle:
  - Otevřete aplikaci DVWA v prohlížeči.
  - Zjistěte URL a port, kde DVWA běží. Obvykle je to http://localhost:80/dvwa/.
  - Ověřte dostupnost cílového systému pomocí nástrojů jako ping nebo curl, například: ping localhost
- Příprava seznamů
  - Připravte seznam uživatelských jmen. Můžete použít seznamy dostupné v prostředí jako je Kali Linux nebo nástroje pro generování seznamů, jako je například Seclists nebo Cewl.
  - Použijte seznam často používaných hesel, například "password", "123456", "admin", a další.
- Konfigurace nástroje Burp Suite:
  - Otevřete Burp Suite.
  - Nastavte Burp Suite, aby fungoval jako proxy server pro prohlížeč a zachycoval provoz mezi prohlížečem a DVWA.
  - Nastavte proxy v prohlížeči, aby používal Burp Suite (adresa a port, na kterém Burp Suite běží).
  - V Burp Suite konfigurujte průchodový bod (Upstream Proxy) pro provoz směřující na cílovou aplikaci DVWA.
- Provedení útoku:
  - Spusťte Burp Suite a sledujte provoz mezi prohlížečem a DVWA.
  - Zaznamenejte požadavky na přihlášení, které obsahují pole pro uživatelské jméno a heslo.
  - S použitím nástroje Intruder v Burp Suite nastavte slovníkový útok s vašimi seznamy uživatelských jmen a hesel.
  - Spusťte útok a monitorujte výstup, hledaje úspěšné přihlášení.
- Ověření výsledků:

- Pokud Burp Suite najde platné přihlašovací údaje, ověřte je manuálně přihlášením se do aplikace DVWA.

Příloha 2 - Testovací scénář pro Command injection

- Výběr nástroje:
  - Otevřete nástroj Commix, který je k dispozici v Kali Linux.
- Identifikace potenciálně zranitelných míst:
  - Identifikujte místa v aplikaci DVWA, kde může dojít k injektáži příkazu.
  - Ověřte, zda tyto vstupy nejsou již satinovány nebo omezeny na konkrétní formáty.
- Testování zranitelností:
  - Použijte Commix k testování identifikovaných vstupních bodů s běžnými injekčními příkazy, např.; ls nebo dir.
  - V případě potřeby upravte parametry Commix pro specifické testování, jako je obcházení obranných mechanismů nebo útoky skryté za autentizací.
- Analýza výsledků:
  - Monitorujte výstup aplikace a Commix, abyste zjistili, zda je možné provádět příkazy na serveru.
  - Pokud je injekce úspěšná, zkuste další příkazy pro získání více informací o serveru, např. získání seznamu uživatelů, prohlížení konfiguračních souborů nebo spuštění systémových diagnostik.

Příloha 3 - Testovací scénář pro CSRF útok

- Výběr nástroje:
  - Otevřete nástroj Burp Suite, který je k dispozici v Kali Linux.
- Zachytávání požadavků:
  - Nastavte Burp Suite tak, aby fungoval jako proxy pro váš prohlížeč.
  - Proveďte průchod aplikací DVWA a zachytávejte všechny HTTP požadavky.
- Analýza požadavků:
  - Identifikujte místa, kde aplikace mění stav nebo data bez použití CSRF tokenu. Klíčová místa mohou zahrnovat změnu hesla, aktualizaci uživatelského profilu nebo jakoukoli jinou akci, která mění stav nebo data.
- Vytvoření škodlivého obsahu:

- Na základě analyzovaných požadavků vytvořte škodlivý odkaz nebo stránku, která replikuje požadavek. Tato stránka může být hostována na jiném serveru a měla by simulovat legitimní výzvu k akci, kterou oběť může sledovat (např. „Klikněte sem pro aktualizaci svého profilu“).
- Použijte Burp Suite nebo jiný nástroj k vytvoření CSRF payloadu, pokud je to nutné.

- Simulace útoku:
  - Pošlete tento škodlivý odkaz oběti. To může být simulováno otevřením odkazu v prohlížeči, kde je uživatel již přihlášen do aplikace DVWA.
  - Sledujte, zda je požadavek úspěšně proveden.
- Ověření výsledků:
  - Ověřte dopady takového útoku, jako je změna hesla oběti nebo úprava uživatelských dat.

Příloha 4 - Testovací scénář pro File inclusion

- Identifikace potenciálních souborů/složek:
  - Poznamenejte si všechny soubory a složky, které mohou být zajímavé pro další zkoumání.
- Analýza aplikace:
  - Prozkoumejte aplikaci DVWA a identifikujte parametry nebo funkce, které mohou zahrnovat soubory, např. funkce pro načítání šablon, konfiguračních souborů nebo jiných zdrojů.
- Testování zranitelnosti:
  - Vložte různé hodnoty do identifikovaných parametrů. Pokusy mohou zahrnovat:
  - Systémové soubory, např. /etc/passwd
  - Jiné soubory aplikace, které jste identifikovali v prvním kroku.
  - Odkazy na externí zdroje, např. http://malicious.example.com/malicious_script.php, abyste ověřili Remote File Inclusion (RFI).
- Ověření zranitelnosti:
  - Sledujte výstupy aplikace a hledejte známky neoprávněného začlenění souborů, jako jsou obsahy systémových souborů nebo výstupy z externích zdrojů.

- Zkoumání dopadu:
    - Pokud bylo možné vložit soubory, zkoumejte, co lze pomocí této zranitelnosti dosáhnout. To může zahrnovat získání citlivých informací, spuštění nežádoucích skriptů nebo změnu chování aplikace.

Příloha 5 - Testovací scénář pro SQL Injection

- Výběr nástroje:
    - Spusťte nástroj sqlmap, který je k dispozici v Kali Linux.
    - Identifikace potenciálně zranitelných bodů:
    - Identifikujte parametry a body v aplikaci DVWA, kde lze vložit SQL kód. To může zahrnovat vstupy formulářů, parametry URL nebo jakékoli jiné vstupní body, kde se očekávají uživatelské vstupy.
- Automatické testování:
    - Použijte sqlmap k automatickému testování identifikovaných vstupních bodů. To může zahrnovat použití různých technik SQL injection, aby byla zajištěna úplnost testování.
- Analýza výsledků:
    - Sledujte výstupy sqlmap a identifikujte zranitelné body.
    - Pro všechny identifikované zranitelné body použijte sqlmap k extrakci dat z databáze, což může zahrnovat získání seznamu tabulek, obsahu těchto tabulek nebo dokonce hesel a jiných citlivých informací.
- Další zkoumání:
    - Pokud je zranitelnost potvrzena, zkuste provádět další útoky přes sqlmap nebo manuálně. To může zahrnovat čtení citlivých dat z databáze, zápis nových dat do databáze, mazání dat z databáze nebo pokusy o získání vyšších oprávnění nebo přístupu k dalším částem systému.

Příloha 6 - Testovací scénář pro Ověřování pomocí JavaScriptu

- Zachycení komunikace:
    - Spusťte Burp Suite a nastavte jej tak, aby fungoval jako proxy pro váš prohlížeč, což umožní zachytit veškerou HTTP a HTTPS komunikaci mezi prohlížečem a aplikací DVWA.

- Identifikace ověřovacích funkcí:
    - Prozkoumejte aplikaci DVWA a identifikujte funkce a části aplikace, které používají JavaScript pro ověřování nebo validaci dat. To může zahrnovat formuláře, interaktivní prvky nebo jakékoli jiné komponenty, které reagují na uživatelské akce.
- Deaktivace JavaScriptu a zkouška:
    - Deaktivujte JavaScript v prohlížeči.
    - Zkuste provést akci nebo funkci v aplikaci DVWA, která by měla být ověřena nebo validována pomocí JavaScriptu.
- Analýza výsledků:
    - Sledujte chování aplikace a zaznamenejte, zda je možné obejít JavaScriptové ověřování. To může zahrnovat odesílání neplatných dat, obejití omezení nebo provádění akcí, které by měly být blokovány.
    - Pokud je ověřování obejito, analyzujte, jaké informace nebo funkce jsou nyní přístupné a jak to může ovlivnit celkovou bezpečnost a funkčnost aplikace.

Příloha 7 - Testovací scénář pro XSS

- Zachycení komunikace:
    - Spusťte Burp Suite a nastavte jej tak, aby fungoval jako proxy pro váš prohlížeč, což umožní zachytit veškerou HTTP a HTTPS komunikaci mezi prohlížečem a aplikací DVWA.
- Identifikace potenciálně zranitelných bodů:
    - Prozkoumejte aplikaci DVWA a identifikujte vstupní body, kde by mohl být vstup odrážen zpět v odpovědi. To může zahrnovat vstupy formulářů, parametry URL, vyhledávání, komentáře nebo jakékoli jiné místo, kde se uživatelský vstup zobrazuje výstupu.
- Testování XSS payloadů:
    - Vložte různé běžné XSS payloady do identifikovaných vstupních bodů. Payloady mohou zahrnovat kód, jako je <script>alert('XSS')</script>, nebo složitější skripty pro získání cookie nebo manipulaci s DOM.
    - Pro komplexní testování můžete využít seznam XSS payloadů dostupných na internetu nebo v databázi nástroje Burp Suite.

- Analýza výsledků:
    - Po odeslání XSS payloadů sledujte chování prohlížeče a zaznamenejte, zda je některý z nich spuštěn.
    - Pokud je některý z payloadů úspěšný, zkoumejte jeho chování a zjistěte, jaký má potenciální dopad na uživatele nebo aplikaci.