Návrh metodiky pro penetrační testování webových aplikací

Bc. Oliver Polka

Diplomová práce
2018

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Oliver Polka**
Osobní číslo:	**A15183**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Forma studia:	**prezenční**
Téma práce:	**Návrh metodiky pro penetrační testování webových aplikací**
Téma anglicky:	**The Design of a Methodology for the Penetration Testing of Web Applications**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na dané téma se zaměřením na bezpečnost webových aplikací.**
2. **Popiště možnosti a nástroje pro penetrační testování webových aplikací.**
3. **Navrhněte a vytvořte metodiku penetračního testu.**
4. **Realizujte penetrační test na reálné webové aplikaci.**
5. **Vhodně vyhodnoťe a reprezentujte získané výsledky včetně doporučení pro odstranění zjištěných zranitelností.**
6. **Vytvořte šablonu jako podklad pro budoucí testování.**

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **SELECKÝ, Matúš. Penetrační testy a exploitace. Brno: Computer Press, 2012. ISBN 978-80-251-3752-9.**
2. **ALLEN, Lee a Kevin CARDWELL. Advanced penetration testing for highly-secured environments: employ the most advanced pentesting techniques and tools to build highly-secured systems and environments. Second edition. Birmingham: Packt Publishing, 2016, xiii, 400. Community experience distilled. ISBN 978-1-78439-581-0.**
3. **FAIRCLOTH, Jeremy. Penetration tester's open source toolkit. 3rd ed. Waltham, MA: Elsevier/Syngress, c2011. ISBN 978-1-59749-627-8.**
4. **LONG, Johnny. Google hacking for penetration testers. Burlington, MA: Syngress Pub., c2008. ISBN 978-1-59749-176-1.**
5. **KLEVINSKY, T. J., Scott. LALIBERTE a Ajay. GUPTA. Hack I.T.: security through penetration testing. Boston: Addison-Wesley, c2002. ISBN 0-201-71956-8.**
6. **OWASP Top 10 2017 [online]. OWASP, 2017 [cit. 2017-12-01]. Dostupné z: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf**

Vedoucí diplomové práce: **Ing. Petr Žáček**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **8. prosince 2017**

Termín odevzdání diplomové práce: **28. května 2018**

Ve Zlíně dne 8. prosince 2017

doc. Mgr. Milan Adámek, Ph.D.
děkan

L.S.

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky

doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor;
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne …….

podpis diplomanta

ABSTRAKT

Táto diplomová práca sa zaoberá penetračným testovaním webových aplikácií. V úvode teoretickej časti popisuje problematiku testovania softvéru. Ďalšia časť je zameraná na používané metodiky, techniky a postupy penetračného testovania, vrátane rozboru najznámejších aplikácií využívaných pre penetračné testovanie. Na základe poznatkov z teoretickej časti, bola navrhnutá metodika pre realizáciu penetračného testu webovej aplikácie v praktickej časti diplomovej práce. Ďalej praktická časť obsahuje popis uskutočneného penetračného testu webovej aplikácie. Zahrňuje jednotlivé fázy penetračného testu od plánovania a prípravy, cez realizáciu a analýzu výsledkov testu. V závere praktickej časti diplomovej práce sú prezentované a zdokumentované výsledky.

Klíčová slova: penetračný test, testovanie softvéru, webová aplikácia, OWASP,

ABSTRACT

This Master‘s thesis is on penetration testing of web applications. In the begining of theoretical part, the field of testing software is described. Next part is focused on techniques, processes and methodologies of penetration testing including the introduction of most common aplications and tools for penetration testing. In practical part, a design of methodology for penetration testing of web application is created based on the information gathered in theoretical part. Practical part also includes description of a penetration test of a real web application, which contains of planning and preparations, conducting the testing and analysis of the results. In the end of the practical part the results are documented and presented.

Keywords: penetration test, software testing, web application, OWASP

Tu, na tomto mieste, by som chcel poďakovať mojej rodine a blízkym za podporu počas celého môjho štúdia.

Vďaka patrí aj môjmu vedúcemu diplomovej práce Ing. Petrovi Žáčkovi za odborné vedenie a trpezlivosť pri písaní tejto práce.

OBSAH

ÚVOD

Otázka bezepečnosti počítačových systémov a sietí je v dnešnej dobe veľmi naliehavá. Rýchly technologický pokrok so sebou neprináša len výhody ale taktiež riziká pre všetkých užívateľov, či už sa jedná o fyzické alebo právnické osoby. Každý z nás používa minimálne mobilný telefón a počítač. Často môžeme podľahnúť pocitu falošného bezpečia a tak volíme jednoduché heslá, nepoužívame antivírusové programy, využívame voľne dostupné aplikácie alebo nakupujeme produkty na e-shopoch, kde poskytujeme informácie o našich kreditných kartách. Každá takáto činnosť znamená potenciál pre tzv. kyberkriminalitu. Dôsledkom môže byť „len“ nevyžiadaná pošta alebo telefonáty na rôzne ankety. Môže ale dôjsť aj k závažnejším trestným činom ako je krádež finančných prostriedkov z nášho účtu alebo krádež identity. V obchodnom prostredí nedostatočná bezpečnosť používaných systémov môže znamenať únik citlivých informácií o činnosti spoločnosti, know-how alebo informáciách o zákazníkoch. Finančné straty sú v takomto prípade rozsiahle a môžu ohroziť existenciu organizácie.

Ak hovoríme o bezpečnosti webových aplikácií, nedostatočne zabezpečené aplikácie sú ľahkým terčov hackerov, ktorý môžu mať rôzne úmysly. Či už to je osobné obohatenie alebo poškodenie prevádzkovateľa aplikácie alebo jej užívateľov. Takéto incidenty znamenajú veľké riziko pre konkurencieschopnosť a dôveru užívateľov. Preto sa na bezpečnostnú politiku kladie v organizáciách čoraz väčší dôraz. Neoddeliteľnou súčasťou opatrení v rámci bezpečnostnej politiky je aj pravidelný bezpečnostný audit k odhaleniu prípadných zraniteľných miest softvéru. V spolupráci s testermi sú realizované penetračné testy, ktoré simulujú možné chovanie hackerov pri útoku na systém.

Táto diplomová práca sa zaoberá práve problematikou penetračných testov v súvislosti s testovaním softvéru ako prostriedku k ochrane pred kyberkriminalitou.

Cieľom diplomovej práce je poskytnúť ucelený pohľad na možnosti a nástroje penetračných testov webových aplikácií a návrh metodiky pre reálny penetračný test, ktorý je prevedený v rámci praktickej časti.

Penetračný test je v závere praktickej časti vyhodnotený a sú navrhnuté odporúčania pre odstránenie zistených zraniteľností.

Poznatky a závery z tejto diplomovej práce je možné použiť ako podklad pre budúce testovanie webových aplikácií.

I. TEORETICKÁ ČÁST

1 BEZPEČNOSŤ WEBOVÝCH APLIKÁCIÍ

S rozvojom informačných technológií sa veľká časť kriminality presúva do sveta počítačov, Internetu. Často je označovaná termínom kybernetická kriminalita. Tento pojem je odvodený od pojmu kybernetický priestor, ktorým sa označuje virtuálne prostredie, bez začiatku a konca, nemá hranice národných štátov a nie je možné určiť aké je rozsiahle. Polícia ČR definuje kybernetickú kriminalitu, ako trestnú činnosť, ktorá je páchaná v prostredí informačných a komunikačných technológií vrátane počítačových sietí. Oblasť informačných alebo komunikačných technológií je buď predmetom útoku, alebo je páchaná trestná činnosť za výrazného využitia týchto technológií. [1]

Zo štatistiky Polície ČR vyplýva, že počet trestných činov kybernetickej kriminality za posledných sedem rokov narástol takmer štvornásobne čo je možné vidieť v nasledujúcej tabuľke (Tab. 1).V tabuľke vidíme najviac zastúpené skupiny trestných činov.

Tab. 1: Skupiny trestných činov kyberkriminality a ich podiel za obdobie 2011-2017 [1]

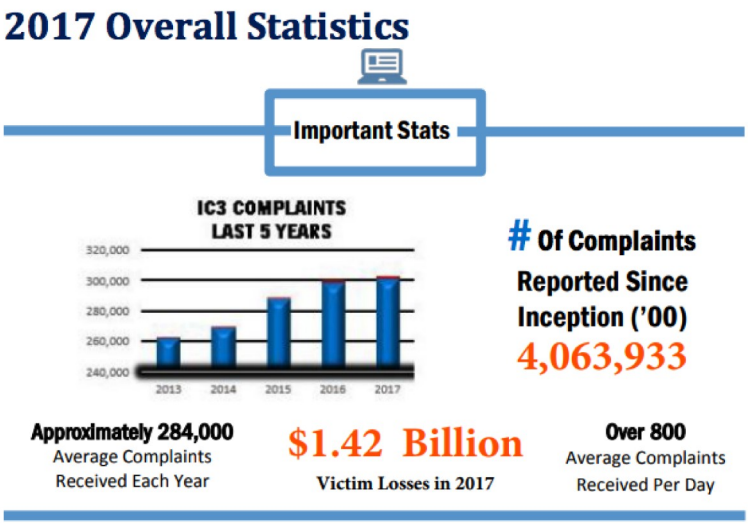
Štruktúra útokov	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**
podvodné jednanie	917	1303	1863	2478	2932	3235	3140
tj. %	61.05%	59.36%	59.94%	56.99%	58.37%	60.54%	55.54%
hacking	66	112	220	555	578	534	608
tj. %	4.39%	5.10%	7.08%	12.76%	11.51%	9.99%	10.75%
marvnostné delikty	134	161	261	314	351	344	561
tj. %	8.92%	7.33%	8.40%	7.22%	6.99%	6.44%	9.92%
autorskoprávne delikty	155	241	181	262	315	237	296
tj. %	10.32%	10.98%	5.82%	6.03%	6.27%	4.43%	5.24%
násilné prejavy a hate crime	86	111	155	202	230	265	318
tj. %	5.73%	5.06%	4.99%	4.65%	4.58%	4.96%	5.62%
ostatné	146	267	428	537	617	729	731
tj. %	9.72%	12.16%	13.77%	12.35%	12.28%	13.64%	12.93%
Celkom útokov v IT	**1502**	**2195**	**3108**	**4348**	**5023**	**5344**	**5654**

1.1 Internet Crime Report

Závažnosť témy kybernetickej kriminality z celosvetového hľadiska dokazuje aj report americkej bezpečnostnej organizácie Internet Crime Complaint Center (IC3), ktorá spadá pod FBI. Organizácia IC3 poskytuje verejnosti spoľahlivý mechanizmus na oznamovanie podozrivých aktivít v kyberpriestore, ktoré by mohli byť predmetom trestnej činnosti. Zároveň zbiera dáta, vyhodnocuje hrozby a následne sa stará o zvýšenie povedomia

verejnosti o hrozbách internetovej kriminality. [2]

V každoročnej správe pre širokú verejnosť IC3 informuje o svojej činnosti, úspechoch a tiež analyzuje trendy v oblasti internetovej kriminality.



Obr. 1. Štatistiky podnetov IC3 za rok 2017 [3]

IC3 obdržala od roku 2000 viac než 4 milióny podnetov, priemerne 284 tisíc ročne. Finančné straty spôsobené kyberkriminalitou za rok 2017 dosiahli hodnoty 1,42 miliárd dolárov (Obr. 1). Z reportu IC3 za rok 2017 máme taktiež možnosť získať informácie o typoch kriminálnych činov v kyberpriestore (Obr. 2). [3]

2017 Crime Types

By Victim Count			
Crime Type	**Victims**	**Crime Type**	**Victims**
Non-Payment/Non-Delivery	84,079	Misrepresentation	5,437
Personal Data Breach	30,904	Corporate Data Breach	3,785
Phishing/Vishing/Smishing/Pharming	25,344	Investment	3,089
Overpayment	23,135	Malware/Scareware/Virus	3,089
No Lead Value	20,241	Lottery/Sweepstakes	3,012
Identity Theft	17,636	IPR/Copyright and Counterfeit	2,644
Advanced Fee	16,368	Ransomware	1,783
Harassment/Threats of Violence	16,194	Crimes Against Children	1,300
Employment	15,784	Denial of Service/TDoS	1,201
BEC/EAC	15,690	Civil Matter	1,057
Confidence Fraud/Romance	15,372	Re-shipping	1,025
Credit Card Fraud	15,220	Charity	436
Extortion	14,938	Health Care Related	406
Other	14,023	Gambling	203
Tech Support	10,949	Terrorism	177
Real Estate/Rental	9,645	Hacktivist	158
Government Impersonation	9,149		
Descriptors*			
Social Media	19,986	*These descriptors relate to the medium or tool used to facilitate the crime, and are used by the IC3 for tracking purposes only. They are available only after another crime type has been selected.	
Virtual Currency	4,139		

Obr. 2. Typy kriminálnych činov podľa počtu obetí za rok 2017 [3]

1.1.1 Kategórie kybernetických zločinov podľa IC3

IC3 rozdeľuje jednotlivé druhy kyberkriminality na nasledovné kategórie:

- **419/overpayment**: jedná sa o podvodný email, v ktorom odosielateľ navrhuje príjemcovi podiel na zisku . Podmienkou je uhradenie manipulačného poplatku.
- **Dodatočný poplatok (Advanced Fee):** dodatočný poplatok. Páchateľ informuje obeť o veľkej finančnej výhre. Najprv ale musí zaplať daň alebo poplatok aby mu výhra bola vyplatená. Obeť poplatok zaplatí, ale sľúbené peniaze neobdrží.
- **Aukcie (Auction):** falošná transakcia, ku ktorej dochádza na aukčných online stránkach.
- **Zneužitie obchodného emailu (Business Email Compromise)**: je zameraný na spoločnosti, ktoré obchodujú s externými dodávateľmi alebo spoločnosťami, ktoré pravidelne realizujú online peňažné prevody. Tieto sofistikované podvody majú na svedomí podvodníci, ktorí dokážu kompromitovať emailové účty pomocou sociálneho inžinierstva alebo nabúrania sa do počítačového systému.

- **Falošná charita (Charity)**: páchatelia vytvoria falošné charitatívne organizácie, väčšinou s tématikou chorôb, a profitujú z jednotlivcov, ktorý veria, že finančne podporujú legálnu organizáciu.

- **Zneužitie dôvery (Confidence/Romance Fraud)**: podvodník si s obeťou vytvorí dôverný vzťah, na základe toho požiada o finančnú pomoc, ktorú následne zneužije.

- **Únik obchodných dát (Corporate Data Breach)**: únik obchodných informácií zo zabezpečeného miesta do nedôveryhodného prostredia. Taktiež môže ísť o únik dát v rámci organizácie, keď sú citlivé alebo tajné informácie kopírované, prenášané, ukradnuté alebo použité neautorizovanou osobou.

- **Zneužitie kreditnej karty (Credit Card Fraud):** s kreditnou kartou je vykonaná neautorizovaná transakcia.

- **Zločiny proti deťom (Crimes against children):** čokoľvek spojené so zneužitím detí.

- **Kriminálne fóra (Criminal Forums):** médiá, kde sa vymieňajú nápady a postupy, ktoré je možné použiť ku kriminálnej činnosti.

- **Zamietnutie služby (Denial of Service)**: prerušenie prístupu autorizovanej osoby do systému alebo siete, typicky spôsobené nezákonným úmyslom.

- **Podvodné zamestnávanie (Employment):** jedinec verí, že je legálne zamestnaný a prichádza o peniaze vykonávaním práce, za ktorú následne nie je zaplatený.

- **Vydieranie (Extortion)**: nezákonné požadovanie peňazí alebo majetku pod hrozbou napr. násilia. Môže zahrňovať hrozby, fyzické napadnutie, verejné poníženie atď..

- **Gambling**: online gambling, obecne všeobecný pojem pre gambling s využitím internetu.

- **Vydávanie sa za štátneho úradníka (Government Impersonation):** podvodník sa vydáva za úradníka s cieľ získať peňažných prostriedkov od obete.

- **Hacktivist**: počítačový hacker, ktorého aktivity slúžia k propagácii spoločenského alebo politického problému.

- **Obťažovanie/ vyhrážanie (Harassment/Threats of Violence)**: obťažovanie vzniká keď páchateľ použije falošné obvinenie alebo vyjadrenie k zastrašovaniu obeti. Hrozba násilia odkazuje na vyjadrenie zámeru spôsobiť bolesť, zranenie v prípade, že obeť nezaplatí požadovanú sumu.

- **Poistné podvody (Health Care Related)**: trestné činy, ktoré súvisia so zneužitím súkromnej alebo verejnej zdravotnej starostlivosti.

- **Porušenie autorského práva (IPR/Copyright and Counterfeit)**: krádež alebo ilegálne využitie cudzích nápadov, vynálezov alebo umeleckého prejavu, vrátane obchodného tajomstva, patentov produktov, časti filmov, hudby alebo softvéru.

- **Krádež identity/krádež účtu (Identity Theft/Account Takeover):** krádež osobných identifikačných údajov ako je meno, číslo sociálneho poistenia. V prípade krádeže účtu páchateľ získa prístupové údaje k účtu (prihlasovacie meno, heslo)

- **Investičné podvody (Investment)**: podvodná technika, pri ktorej páchateľ získa investície na základe klamných informácií. Podvodník sľubuje veľký zisk s malým rizikom. Zahrnuje i tzv. pyramidové hry.

- **Lotérie/Stávky (Lottery/Sweepstakes)**: jedinec je kontaktovaný ohľadne výhry v lotérii alebo stávke, ktorej sa ale nezúčastnil. „Výhru“ získa až po zaplatení poplatku – po zaplatení ale páchateľ preruší kontakt.

- **Maleware/Scareware**: softvér navrhnutý so zámerom poškodiť ale znefunkčniť počítačový systém. Niekedy využíva zastrašujúcu taktiku pri domáhaní sa finančnej sumy.

- **Skreslenie skutočnosti (Misrepresentation)**: produkt alebo služba je predaná online, po doručení ale kvalita neodpovedá popisu ako bol produkt/služba ponúkaná.

- **Nedoručenie platby/ produktu (Non Payment/Non Delivery):** platba za produkt alebo službu nie je uhradená alebo zaplatený produkt nie je doručený.

- **Únik osobný údajov (Personal Data Breach)**: únik obchodných informácií zo zabezpečeného miesta do nedôveryhodného prostredia. Taktiež môže ísť o únik dát

v rámci organizácie, keď su citlivé alebo tajné informácie kopírované, prenášané, ukradnuté alebo použité neautorizovanou osobou.

- **Phishing/Vishing/Smishing/Pharming**: nevyžiadané emaily, textové správy alebo telefonáty od legálnych organizácií, ktoré vyžadujú osobní, finančné alebo prístupové informácie.
- **Ransomware**: typ poškodzujúceho softvéru navrhnutého tak aby zablokoval prístup k počítačovému systému dokým nie sú vyplatené peniaze páchateľovi.
- **Re-shipping**: jedinec obdrží zásielku, ktorú si neobjednal a musí následne zaplatiť za poštovné „správnemu“ majiteľovi, väčšinou do zahraničia.
- **Majetkové podvody (Real Estate)**: podvody súvisiace s nehnuteľnosťami, prenájmom alebo vlastníctva nehnuteľností.
- **Social Media**: trestný čin s využitím sociálnych médií ako napr. Facebook, Twitter, Instagram atď.
- **Tech support**: pokus o získanie prístupu k eletektronickému zariadeniu obeti pod zámienkou technickej podpory, zvyčajne ako dobre známa spoločnosť. Podvodníci si vyžiadajú vzdialený prístup k zariadeniu, ktoré následne zneužijú.
- **Terorizmus (Terrosism)**: násilný čin s úmyslom vytvoriť strach na základe náboženských, politických alebo ideologických cieľov.
- **Vírus (Virus):** kód schopný skopírovať sám seba, ktorý má škodlivý efekt ako napr. skompromitovanie systému alebo zničenie dát.
- **Podvody s virtuálnou menou (Virtual Currency)**: trestný čin v oblasti kryptomeny ako je Bitcoin, Litecoin alebo Potcoin. [3]

Z vyššie uvedeného vidíme, že podvodníci dokážu s využitím moderných technológií oklamať tisícky ľudí a získať nelegálnym spôsobom veľké finančné čiastky. Preto dnes viac ako kedykoľvek v minulosti je nutné venovať pozornosť zabezpečeniu počítačových systémov a sietí proti neautorizovaným prístupom. V bežnom živote každý z nás používa až stovky aplikácií a softvéru, ktoré pracujú s našimi osobnými údajmi a preto sme veľmi zraniteľný ak by poskytovateľ týchto služieb zanedbal otázky bezpečnosti. Nasledujúca

kapitola pojednáva o testovaní softvéru, ktorý je neoddeliteľnou súčasťou bezpečnosti webových aplikácií.

1.2 OWASP TOP 10 PROJECT

Open Web Application Security Project (OWASP) je projekt a online komunita zaoberajúca sa tvorbou zadarmo dostupných materiálov, dokumentov, metodológií či nástrojov v oblasti bezpečnosti webových aplikácií. Projekt vznikol v roku 2001, neskôr bol preformovaný do nadácie OWASP Foundation ako nezisková organizácia v USA a pod rovnakým názvom vystupuje aj v Európe. Medzi známe projekty patria napríklad OWASP Top Ten, OWASP Software Assurace Maturity Model, OWASP Testing Guide, a iné. [4]

OWASP Top Ten je projekt cielený pre zvýšenie povedomia o bezpečnosti webových aplikácií. Reprezentuje všeobecné znalosti o najzávažnejších bezpečnostných hrozbách webových aplikácií. Do projektu sú zapojení bezpečnostní experti z celého sveta, ktorí sa podieľajú na vytvorení tohto zoznamu. Dokument je pravidelne aktualizovaný aby reflektoval aktuálny stav hrozieb na Internete. V dokumente je popísaných 10 najzávažnejších hrozieb, ich priebeh, prevencia, a riziko aké so sebou nesú. Adopcia postupov priblížených v projekte predstavuje prvé kroky k vytvoreniu bezpečnostnej kultúry v organizácií a teda naštartovanie procesu minimalizácie týchto rizík. Za rok 2017 definuje projekt OWASP TOP 10 nasledujúce hrozby:

- **Injection** - chyby umožňujúce injekciju a spustenie kódu v SQL, OS, LDAP, keď sú nedostatočne ošetrené dáta odoslané ako súčasť príkazu alebo dotazu. Útočník dokáže týmto spôsobom spustiť príkaz naviac pre prístup a manipuláciu s dátami ku ktorým nemá oprávnenie.

- **Broken Auth. and Session Management** - nesprávna implementácia funkcií aplikácie pre autentifikáciu a správu relácií môže umožniť útočníkom získať heslá, kľúče, tokeny, alebo inak zneužiť problém v implementácií pre dočasné alebo trvalé prebratie užívateľovej identity.

- **Sensitive Data Exposure** - webové aplikácie a API často nedostatočne chránia osobné a citlivé údaje, či už finančné alebo informácie o zdravotnej starostlivosti, atď.. Útočníci sú schopný získať, ukradnúť alebo pozmeniť takéto nedostatočne

chránené dáta, ukradnúť identitu, vytvoriť kreditný podvod, alebo spáchať iné kriminálne činnosti s využitím týchto dát. Citlivé dáta si vyžadujú nadštandardnú ochranu, šifrovanie nie len dát samotných ale aj šifrovanie prenosov medzi internými a externými službami ale aj medzi webovým serverom a koncovým užívateľom.

- **XML External Entities (XXE)** - staršie ale nesprávne konfigurované XML procesory vyhodnocujú externé referencie na entity vo vnútri XML dokumentov. Tieto referencie môžu odhaliť interné súbory, sieťové disky, vykonať interné skenovanie portov, spustenie vzdialeného kódu alebo pomôcť pri vytvorení útoku DoS.

- **Broken Access Control** - neautentifikovaní užívatelia nemajú prístup do všetkých častí systému, avšak tieto obmedzenia sú často nedostatočne vynucované a chyby sa dajú zneužiť pre prístup do častí systému ktoré nemajú byť prístupne neautentifikovaným užívateľom alebo nemajú byť prístupné užívateľom bez dostatočných prístupových práv.

- **Security Misconfiguration** - nesprávne nastavené bezpečnostné opatrenia, kvôli vlastnej nesprávnej manuálnej konfigurácií, kvôli externej ad-hoc konfigurácií alebo žiadnej konfigurácií (webové aplikácie bez SSL). Známa nedostatočne zabezpečená defaultná konfigurácia, alebo využívanie nastavení pre uľahčenie developmentu na produkčných stránkach býva taktiež častým problémom - chybové hlášky obsahujúce citlivé informácie alebo informácie o infraštruktúre, nesprávne nastavené HTTP hlavičky, neaktualizované systémy a SW, framework, komponenty atď.

- **Cross-Site Scripting (XSS)** - chyby umožňujúce XSS vznikajú, keď webová aplikácia narába s neznámymi dátami bez správnej validácie, odstraňovania neprijateľných znakov alebo pri obnovovaní webových stránok z užívateľských dát použitím API prehliadača. XSS útoky dovoľujú útočníkovi spustenie skriptov v prehliadači obete, odchytenie dát, zneužitie prihlásenia alebo presmerovanie obete na podvodné stránky bez vedomia užívateľa.

- **Insecure Deserialization** - nezabezpečená deserializácia často vedie k vzdialenému spusteniu škodlivého kódu. Aj v prípadoch kedy problémy deserializácie nevedú k spusteniu kódu, môžu byť zneužité pre ďalšie útoky alebo eskaláciu

užívatľských práv.

- **Using Components with Known Vulnerabilities** - komponenty webových aplikácií, ako sú knihovne, frameworky alebo iné softvérové moduly sú spúšťané s rovnakými právami ako webová aplikácia. Ak určitý komponent obsahuje známu zraniteľnosť, ktorá je zneužitá, útok môže zapríčiniť stratu dát alebo dokonca prevzatie kontroly na serverom. Využívanie komponentov so známymi zraniteľnosťami môže znížiť celkovú bezpečnosť aplikácie alebo API a umožniť rôzne formy útokov.

- **Insufficient Logging and Monitoring** - Nedostatočné logovanie a monitorovanie, spojené s neexistujúcou alebo neefektívnou integráciou reakcie na útoky umožňuje útočníkom prenikať hlbšie do systému, udržať si prístup na dlhší čas, získať prístup do ďalších pripojených systémov, upraviť a pozmeňovať dáta či dokonca ich úplne zničiť. [5]

Bezpečnosť webových aplikácií však nekončí týmito 10 skupinami hrozieb. Stovky ďalších problémov môžu ovplyvniť celkovú bezpečnosť aplikácie. Prevedením jednorazového bezpečnostného auditu práca na aplikácií nekončí. Tak ako sa hrozby každoročne menia, menia sa aj techniky a metódy útokov, sú objavené nové zraniteľnosti. Je preto nutné aby sa bezpečnosť stala dôležitou súčasťou stratégie organizácie a odstraňovanie hrozieb bolo kontinuálnym procesom.

2 TESTOVANIE SOFTVÉRU

Vývoj softvéru je z pohľadu použitých technológií, s možnosťami nasadenia na rôzne platformy, alebo z pohľadu na rôznorodosť programov veľmi rozsiahly pojem. Testovanie softvéru, je neodeliteľná súčasť vývoja. Či už sa jedná o manuálne prechádzanie aplikáciou, či písanie automatizovaných testov, tieto, a ďalšie činnosti overovania správneho fungovania aplikácie sú každodenné činnosti nie len vývojárov, ale aj špecializovaných pracovníkov – testerov.

2.1 Chyby v softvéri

Pri popisoch situácií so zlyhaním softvéru, sa používajú rôzne pojmy ako sú:

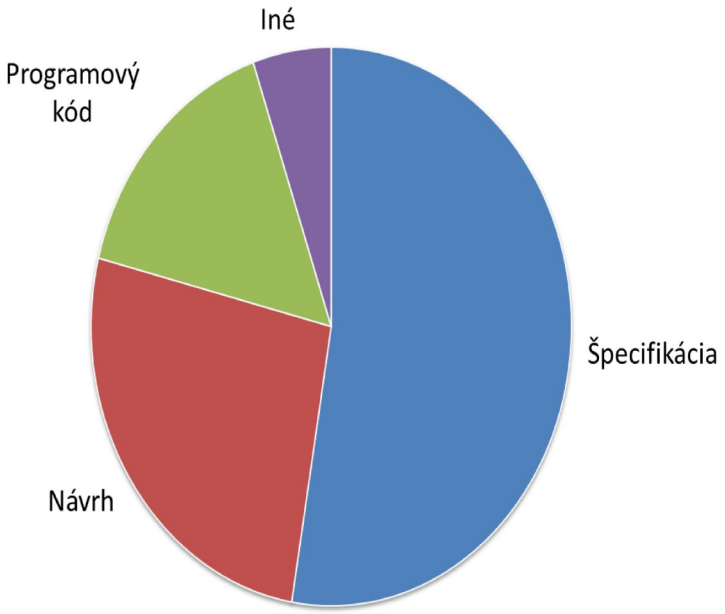
- vada
- závada
- problém
- zlyhanie
- udalosť
- anomália
- chyba
- nekonzistencia [6]

Jednotlivé pojmy sú používané s inou frekvenciou a ich význam sa môže mierne odchyľovať, avšak pre jasné porozumenie je vhodné používať jeden výraz a držať sa ho. Preto aj v nasledujúcich kapitolách bude využívané slovo chyba alebo anglický originál „bug“. Pre definíciu chyby je vhodné vyhradiť pojem „špecifikácia produktu“, jedná sa o popis ako bude výsledný produkt(softvér, aplikácia) vyzerať, správať sa, čo bude schopný robiť a naopak nerobiť. Následne je teda možné hovoriť o chybe v prípade, že aspoň jedna z daných podmienok je pravdivá:

- Softvér nerobí niečo, čo by podľa špecifikácie mal.
- Softvér robí niečo, čo by podľa špecifikácie robiť nemal.
- Softvér robí niečo, o čom sa špecifikácia nezmieňuje.
- Softvér nerobí niečo, o čom sa síce špecifikácia nezmieňuje, ale mala by sa.

- Softvér nie je zrozumiteľný, nedá sa s ním pracovať, alebo by ho mohol koncový zákazník považovať za nesprávny. [6]

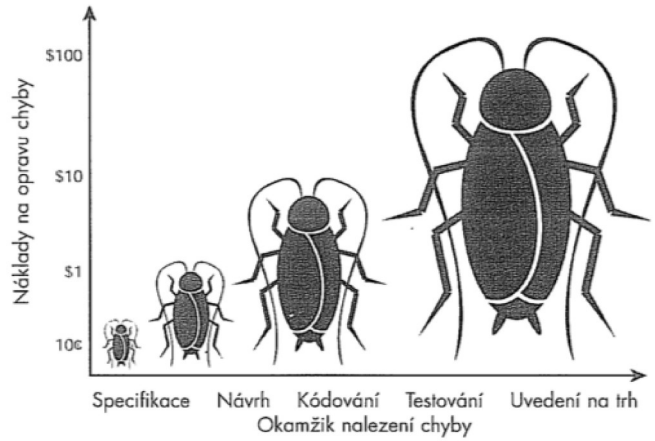
K efektívnemu testovaniu softvéru, je nutné porozumieť prečo a kde chyby vznikajú. Prekvapivým štatistickým údajom je, že väčšina z chýb nie spôsobená programátormi pri písaní zdrojového kódu softvéru, ale už pri nedostatočne dobre spracovanej špecifikácií produktu alebo pri jeho návrhu (Obr. 3). [6]



Obr. 3. Príčiny chyby [6]

Dôvodov prečo je to tak, je hneď niekoľko, stručná alebo žiadna špecifikácia nie je napísaná, špecifikácia nie je dostatočne podrobná, špecifikácia sa neustále mení. Je teda nutné v rámci vývoja softvéru a jeho plánovania tento krok nezanedbávať ani v prípade moderného agilného vývoja SW. Keďže sa požiadavky na bezpečnosť neustále zvyšujú je nutné aj tieto dostatočne špecifikovať. [6]

Z tohto vyplýva, že je možné odhaliť chyby už na začiatku vývojového procesu aplikácie dokonca vo fáze špecifikácie parametrov budúceho softvéru. Náklady na opravu chýb v rannom štádiu projektu sú niekoľkonásobne nižšie ako v prípade, že daná chyba sa prejaví až v momente kedy aplikáciu používa koncový užívateľ. To znamená, že čím skôr je chyba odhalená tým menšie straty spôsobí(Obr. 4). [6]



Obr. 4. Náklady na opravu chyby v priebehu času rastú [6]

Ak sa napríklad bezpečnostná chyba objaví až po uvedení produktu na trh, jej zneužitie môže spôsobiť nie len finančné straty, ale aj ohroziť meno organizácie či poškodiť meno danej organizácie na trhu. Pravé z tohto dôvodu, je význam testovania bezpečnosti vrátane penetračných testov veľmi veľký. [6]

2.2 Definícia testovania

Pojem „testovanie softvéru“ má veľa definícií, je ťažké vybrať jednu, ktorá by popisovala presne to, čo za týmto termínom stojí, testovanie softvéru sa v priebehu rokov mení rovnako ako sa menia programy a spôsoby ich vývoja.

Bill Hetzel vo svojej knihe „The Complete Guide to Software Testing“ uvádza: *„Testing is the process of establishing confidence that program or system does what it is supposed to“*. Testovanie popisuje ako proces vytvárania, nadobúdania dôvery, že program alebo systém, ktorý vyvíjame naozaj robí to čo od neho očakávame, pre čo bol vytvorený. [7]

Iný pohľad na testovanie priniesol Mayers „*Testing is the process of executing a program or system with the intent of finding errors*“. Privádza nás k myšlienke, že cieľom testovania je hľadanie chýb. Často býva toto testovanie označované ako „dynamické“, pre potrebu mať bežiacu aplikáciu alebo systém. Je však nutné si uvedomiť, že dosiahnutie stavu, kedy program nemá žiadne chyby je takmer nemožné. Ďalším z nedostatkov tejto definície je jej konkrétnosť. Testovanie je totiž oveľa rozsiahlejšie, ako len spúšťanie aplikácie. [7]

Je nutné pozerať sa na testovanie ako na širokú a nepretržitú činnosť, počas vývoja softvéru. Hetzel uvádza súhrn pohľadov na testovanie SW, ktoré pozbieral za niekoľko rokov spolu s kolegami počas seminárov, od účastníkov a patrí sem napríklad:

- kontrola oproti špecifikácií programu,
- hľadanie „bugov“ v programe,
- vytváranie akceptačných kritérií,
- uistenie sa, že systém je pripravený na používanie,
- získavanie dôvery vo funkčnosť programu,
- chápanie limitov výkonu aplikácie,
- učenie sa čo systém nedokáže,
- overovanie dokumentácie,
- dokazovanie ukončenia práce. [7]

Testovanie môže byť teda chápané ako proces zbierania informácií o programe alebo systéme. Patria sem aktivity, ktoré nám dávajú odpovede na otázky typu:

- Je SW pripravený na používanie?
- Aké sú riziká?
- Čo všetko je SW schopný urobiť?
- Aké sú limitácie daného SW? [7]

Analýza všetkých týchto bodov priviedla Hetzela k vytvoreniu novej definície testovania: *„Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets the required results*“. Táto definícia teda zahrňuje nie len dynamické testovanie, ale aj tzv. „statické“ testovanie, analýzu špecifikácie, dokumentácie, zdrojového kódu. Statické testovanie, na rozdiel od dynamického, nevyžaduje bežiacu aplikácie, je možné teda so statickým testovaním začať od prvej špecifikácie produktu, ešte pred napísaním prvého riadku zdrojového kódu. Takéto

testovanie môže mať za výsledok spresnenie špecifikácie, zlepšenie odhadov alebo ušetrenie nákladov pre následný vývoj. [7]

Ďalšia definícia, ktorá je často spomínaná hovorí o testovaní SW ako o procese overovania kvality, popisuje testovanie softvéru ako empirické hľadanie vykonávané za účelom poskytnutia všetkých zainteresovaným stranám informácie o kvalite testovaného programu alebo systému. [7]

Definícia podľa Institute of Electrical and Electronics Engineers(IEEE) hovorí, že softvérové testovanie je proces analýzy SW k odhaleniu rozdielov medzi existujúcimi a požadovanými podmienkami, odhaleniu bugov a k zhodnoteniu vlastností programu alebo systému. [8]

2.3 Kategórie testovania

Testovanie softvéru prebieha skúmaním produktu na niekoľkých úrovniach a následným reportovaním výsledkov. Proces testovania je možné rozdeliť na tieto kategórie:

- Black-box testing a White-box testing
- Statické a dynamické testovanie
- Testy splnením a zlyhaním
- Funkcionálne a nefunkcionálne testovanie [6]

2.3.1 Black-box a White-box testovanie

Jedno zo základných rozdelení testovania SW je:

- Black-box testing (Testovanie čiernej skrinky)
- White-box testing (Testovanie bielej skrinky) [6]

Testovanie čiernej skrinky vychádza z predpokladu, že tester, vie a „vidí“ len to čo daný softvér má robiť, nedokáže sa do vnútra skrinky pozrieť a nemôže teda vedieť ako daný

softvér pracuje. Z toho dôvodu nie je možné vytvoriť testovacie scenáre na základe toho, ako softvér pracuje a musí sa pri testovaní opierať len o špecifikáciu. [6]

Testovanie bielej skrinky vychádza z predpokladu, že tester má plný prístup ku zdrojovému kódu softvéru, je schopný sa pozrieť „do vnútra“, kde môže skúmať vnútornú logiku zdrojového kódu, a navrhnúť tak špecifické testovacie scenáre pre otestovanie správneho fungovania. [6]

Tieto prístupy majú svoje opodstatnenie a miesto vo vývoji softvéru vo všetkých fázach testovania. Počas jednotlivých fáz sa ich podiel mení. V literatúre je možné sa stretnúť s pojmom „grey-box testing“, táto stredná cesta býva častá v tímoch, kde sú jednotlivé testovacie role rozdelené medzi vývojárov a špecializovaného testera. Tester má možnosť pozrieť „pod pokrievku“ vyvíjaného softvéru pre zlepšenie efektivity testovania na úrovni užívateľskej. Dobrý príkladom testovania „šedej skrinky“ je testovanie webových aplikácií, kedy má tester možnosť prechádzať jednotlivé webové stránky, ale zároveň aj nahliadnuť do zdrojového HTML kódu a získať tak určitú znalosť štruktúry webovej aplikácie. [6]

2.3.2 Statické a dynamické testovanie

Z pohľadu, či je k prevedeniu testu nutné softvér spustiť, sa testovanie delí na:

- statické testovanie
- dynamické testovanie [6]

Statické testovanie nevyžaduje spustenie aplikácie (softvéru), používa sa pre včasné odhalenie chýb, pre lepšie porozumenie kódu a dokumentácie. Výsledkom statického testovania môže byť napríklad spresnenie odhadov náročnosti pre ďalší vývoj softvéru.

Pri dynamickom testovaní, je predmetom záujmu spustený softvér, na základe zmeny vstupov je možné analyzovať výstupy testovaného programu. [6]

2.3.3 Testy splnením a zlyhaním

V priebehu vývoja aplikácie sa v jednotlivých fázach prevádzajú testy splnením(test-to-pass). Cieľom takéhoto testu je dokázať, že softvér spĺňa základnú minimálnu funkcionalitu. Tieto testy sú prevažne jednoduchého charakteru – hľadaná sa akákoľvek úspešná cesta k požadovanému výsledku. Opakom testu splnením sú tzv. testy zlyhania(test-to-fail), ktoré sa prevádzajú po dosiahnutí minimálnej funkcionality za účelom skúmania hraníc možnosti softvéru. Takéto testovanie si vyžaduje časový rámec, keďže hľadanie neexistujúcej chyby by mohlo byť teoreticky nekonečné. [6]

2.3.4 Funkcionálne a nefunkcionálne testovanie

Funkcionálne testovanie je zamerané na tzv. funkčné požiadavky, súhrn funkcionality programu, systému vyvíjaného pre zákazníka. Zákazník pomocou týchto požiadaviek popisuje spôsob, akým bude softvér používaný a na základe týchto požiadaviek dokážeme posúdiť kedy je časť alebo systém ako celok hotový. Pri definovaní funkčných požiadaviek odpovedáme na otázky ako je možné niečo urobiť, alebo ako bude daná funkcionalita implementovaná. [6][9]

Všetky podmienky sú väčšinou spísané do dokumentu špecifikácie požiadaviek alebo funkcionálnej špecifikácie. Niekedy bývajú zapísané priamo vo formáte „Use Case“ - príkladov používania danej časti systému alebo funkcionality. Zo špecifikácie alebo z use case sú vytvorené testovacie scenáre, pomocou ktorých je možné overiť správne fungovanie aplikácie. Testy môžu byť manuálne, ale aj automatizované či už čiastočne alebo úplne pomocou nástrojov. Cieľom je vytvoriť scenáre, na základe znalosti softvéru a znalosti ako bude daný softvér využívať reálny koncový užívateľ. [6][9]

Nefunkcionálne testovanie je typ testovania pre kontrolu aspektov ako sú výkon, efektivita, spoľahlivosť, použiteľnosť, bezpečnosť a iných aspektov softvéru. Nefuncionálne testovanie je špecificky navrhnuté aby overilo „ako“ systém pracuje, zmeralo jeho charakteristiky a vytvorilo pohľad na celkovú pripravenosť softvéru pre užívanie. [6][9]

Porovnanie funkcionálneho a nefukcionálneho testovania uvádzam je uvedené v tabuľke č.2. [9]

Tab. 2: Porovnanie funkcionálneho a nefunkcionálneho testovania [9]

	Funkcionálne testovanie	**Nefunkcionálne testovanie**
Realizácia	- pred nefunkcionálnym testovaním	- po funkcionálnom testovaní
Oblasť záujmu	- požiadavky zákazníka	- očakávania zákazníka
Požiadavky	- ľahko definovateľné	- ťažko definovateľné
Využitie	- pomáha potvrdiť chovanie aplikácie	- pomáha potvrdiť výkon aplikácie
Cieľ	- potvrdenie funkcie softvéru	- potvrdenie výkonu softvéru
Podmienky	- špecifikácia funkcionalít	- špecifikácia výkonu
Manuálny test	- jednoducho realizovateľný	- ťažko realizovateľný
Funkčnosť	- popisuje čo aplikácia robí	- popisuje jako aplikácia pracuje
Príklad testu	Otestovanie možnosti prihlásenia.	Úvodná stránka sa načíta za 2 sekundy.
Typy testov	- unit testy - smoke testy - akceptačné testy - integračné testy - regresné testy	- performance testy - záťažové testy - testy škálovateľnosti - testy použiteľnosti - stress testy - security testy

Security testing

Dôležitým typom nefunkcionálneho testovania je testovanie bezpečnosti tzv. security testing. Bezpečnostné testovanie zisťuje, či systém alebo aplikácia neobsahuje chyby, ktoré by mohli spôsobiť straty dát alebo poškodenie funkcionality. Typickými požiadavkami z pohľadu bezpečnosti sú napríklad:

- dôvernosť dát
- integrita dát
- autenticita dát
- dostupnosť dát
- autorizácia[10][11]

Typy testov bezpečnosti

Podľa Open Source Security Testing Methodology Manual(OSSTMM) je možné rozdeliť bezpečnostné testovanie do siedmych kategórií:

- Skenovanie zraniteľností – je zvyčajne realizované pomocou automatizovaného softvéru pre skenovanie systému proti známym zraniteľnostiam.
- Skenovanie bezpečnosti – zahrňuje identifikáciu slabých sieťových alebo systémových miest a následne poskytuje riešenia pre zníženie rizík ich zneužitia. Skenovanie kombinuje manuálne a automatizované techniky.
- Penetračné testovanie – jedná sa o simulovanie útoku hackera, testovanie zahŕňa analýzu čiastkových systémov za účelom kontroly potenciálnych zraniteľností k externým hackerským útokom.
- Vyhodnotenie rizík – testovanie zahrňuje analýzu bezpečnostných rizík pozorovaných v organizácií, riziká sa klasifikujú ako nízke, stredné a vysoké. Testovanie odporučuje kontrolu a opatrenia k znižovaniu rizík.
- Bezpečnostný audit – preskúmanie aplikačných a operačných systémov z pohľadu bezpečnostných nedostatkov, audit môže byť vykonaný formou kontroly celého kódu.
- Vyhodnotenie bezpečnostnej pozície – kombinácia bezpečnostného skenovania, etického hackovania, a vyhodnocovania rizík za účelom získania kompletného obrazu organizácie z pohľadu bezpečnosti.
- Etické hackovanie – cieľom hackerov nie je profit, ale odhaľovanie bezpečnostných slabín v systémoch.[12][11]

Penetračné testy budú bližšie rozobrané v nasledujúcej kapitole.

3 PENETRAČNÉ TESTY

Penetračný test je simulovanie útoku hackera pre overenie súčasnej odolnosti testovaného systému s využitím všetkých dostupných techník, taktík a znalostí. Na penetračné testovanie je možné pozerať sa ako na formu bezpečnostného auditu na zákazku, môžeme ho zaraďovať do oblasti etického hackingu. Cieľom je nájdenie bezpečnostných slabín pomocou manuálnych alebo automatizovaných testov, prienik do aplikácie, systému alebo do celej firemnej infraštruktúry. Na základe výsledkov penetračného testovania je možné identifikované slabiny analyzovať a znemožniť ich využitie alebo opraviť bezpečnostné chyby. [13]

3.1 Hacking

Neoprávnený prístup k počítačovému systému a nosiču informácií podľa ust. §230 trestného zákonníku je trestným činom, ktorý je využiteľný pre väčšinu jednania označovaného ako tzv. hacking, narušovanie dát, narušovanie systému a neposlednej rade i zneužívanie zariadení. Najtypickejším príkladom, ktorý býva prešetrovaný, je jednanie páchateľa, ktorý prekoná zabezpečenie počítačového systému a získa prístup k dátam obete, s ktorými môže ďalej ľubovoľne nakladať. Súčasťou týchto činností býva mimo iného aj šírenie škodlivého kódu, implementácia tzv. backdooru do voľne prístupného softvéru atď. Stále častejšou formou je napadanie emailových účtov, účtov na sociálnych sieťach, účtov internetového bankovníctva, ktoré má za následok prienik do súkromia, získavanie citlivých informácií s možnosťou ich poškodeniu, zničeniu alebo získaniu finančného prospechu. S tým súvisí aj ďalšia nadväzujúca trestná činnosť (vydieranie, prenasledovanie, krádeže z účtov, podvody) [14]

Súčasťou tohto druhu trestnej činnosti sú aj kybernetické útoky (napr. DDoS) alebo vydieranie prostredníctvom ransomware. Ďalšou formou môže byť aj porušenie tajomstva dopravovaných správ podľa ust. §182 trestného zákonníku, ktorého najčastejší prejav býva označovaný ako sniffing, keď páchateľ zachytáva prebiehajúcu komunikáciu v sieti a získava tak citlivé údaje nielen o prevádzke ale aj obsahu. Deje sa tak často na nezabezpečených wifi pripojeniach, na strane zmanipulovaných emailových serverov a v poslednej dobe i napádaním aj domácich routerov. Páchatelia sa tak dostávajú k citlivým údajom ako sú heslá, platobné údaje, alebo citlivý či intímny obsah, ktorý následne

využívajú k nátlaku na obeť so snahou o vlastné finančné obohatenie alebo poškodenie povesti obete. [14]

3.1.1 Typy hackerov

Hackeri nie sú automaticky kriminálnici. Definícia slova „hacker“ je kontroverzná, označovať niekoho kto kompromituje počítačovú bezpečnosť, ale taktiež označuje aj skúseného vývojára softvéru. Často sa v IT komunite rozdeľujú hackeri na:

- Black Hat Hacker
- White Hat Hacker
- Gray Hat Hacker. [15]

Takzvaný „black hats“ sú typmi hackerov známy hlavne z médií, svojou činnosťou sledujú osobný prospech (krádeže kreditných kariet, osobných dát, a iné), alebo ich cieľom je poškodiť spoločnosť a jednotlivcov (DDoS útoky). Rizikom u tejto skupiny hackerov je, že informácie získane ich nelegálnou činnosťou sú ochotný predať zločineckým organizáciám, ktoré sú schopné spôsobiť oveľa väčšie škody ako hackeri samotní. [15]

Protikladom „black hats“ sú tzv. white hats hackeri, inak nazývaní aj etickí hackeri. Jedná sa o IT expertov, ktorí ovládajú hackerské techniky prieniku do systémov. Robia tak s dobrým úmyslom, často na žiadosť samotných firiem pri testovaní ich systémov. Reportujú svoje nálezy organizáciám, pre ktoré pracujú za cieľom zlepšenia ich obrany proti reálnym útokom. Ich skúsenosti sú často využívané aj pri vývoji softvéru pre odhalenie slabých miest zabezpečenia. Mnohé organizácie vyhlasujú verejnú výzvu na odhalenie zraniteľností v ich SW za značnú finančnú odmenu. [15]

Posledná skupina zapadá niekde medzi. Gray hat hacker pri odhalení slabiny SW túto informáciu nevyužije k vlastnému obohateniu, ale na druhú stranu neposkytne ju ani organizácií aby bola schopná chybu odstrániť. Takéto zistenia verejne zdieľajú, čím vzniká riziko zneužitia kriminálnymi skupinami. [15]

Toto rozdelenie je spojené s povahovými rysmi človeka a je známe aj medzi širokou verejnosťou. Na základe technických znalostí môžeme rozdeliť hackerov na 3 skupiny:

- First-tier Hackers
- Second-tier Hackers
- Third-tier Hackers. [16]

Prvá skupina pozostáva z hackerskej elity, programátorov s extrémnymi schopnosťami pri hľadaní zraniteľností a exploitov. Jednotlivci nehľadajú publicitu, často pracujú v úzadí a sú známe len ich internetové aliasy. Okrem týchto schopnosti sú ďalej schopný vytvárať špecializované nástroje pre zneužívanie exploitov alebo testovanie. Ich činnosť je ťažko vysledovateľná. Môžu pracovať či už pre súkromé alebo vládne bezpečnostné zložky, ale aj pre iné záujmové skupiny. [16]

Druhú skupiuú tvoria tzv. „Second-tier Hackers“, ktorí disponujú schopnosťami na úrovni systémových administrátorov. Majú skúsenosti s viacerými OS, rozumejú sieťovej komunikácií a dokážu využiť exploity. Majoritným podielom bezpečnostných konzultantov spadá do tejto kategórie. Často sa zapájajú do projektov IT open-source komunity, pri tvorbe nových nástrojov. [16]

Third-tier hackers sú často označovaný ako „script kiddies“ pojem vznikol z faktu, že členovia tejto skupiny sa spoliehajú na skripty alebo hackerské nástroje získané z Internetu a zároveň sa často jedná o neplnoletých inšpirovaných filmami. Takéto osoby, väčšinou nemajú dostatočné technické znalosti, pri ich činnosti si neuvedomujú reálne riziká, ktorým vystavujú ciele alebo aj seba. Pre organizácie predstavujú veľký problém, pretože dôsledky ich činnosti sú nepredvídateľné. [16]

3.1.2 Dôsledky reálnych útokov

Ako sme uviedli v kapitole 1, kyberkriminalita neznamená len únik dát, ale aj finančné straty spôsobené krádežou firemných informácií. IC3 vyčíslila stratu spôsobenú kyberkriminalitov za rok 2017 na približne 1,42 miliardy amerických dolárov. Pre porovnanie v Európe, spoločnosť Ponemon Institute realizovala analýzu v štyroch európskych krajinách, kde analyzovala výšku finančných strát spojenú s únikom dát v komerčných firmách (Tab. 3). [13]

Tab. 3: Cena straty dát [13]

Krajina	Nemecko	Veľká británia	Francúzsko	Taliansko
Podnikateľské finančné straty	1,33 mil. €	780 tis. £	782 tis. €	474 tis. €
Priemerné finančné straty na jednotku	146 €	79 £	122 €	78 €
Percento zákazníkov, ktorí opustia spoločnosť po strate dát	3.5%	2.9%	4.4%	3.5%
Štatistika príčin straty dát				
Kriminálne útoky a krádeže	42.0%	31.0%	43.0%	28.0%
Nedbalosť zamestnancov a dodávateľov	38.0%	36.0%	30.0%	39.0%
Zlyhanie IT a biznis procesov	19.0%	33.0%	26.0%	33.0%

Z prezentovaných dát vyplýva, že straty nie sú zanedbateľné, z toho dôvodu je otázka penetračných testov a potreba IT bezpečnosti na mieste. Testy by mali overiť nie len odolnosť sietí voči útokom z vonka, ale aj proti útokom vlastných zamestnancov s nekalými úmyslami. [13]

Okrem priamych finančných strát, môže dôsledkom hackerského útoku nastať:

- nedostupnosť služby,
- neoprávnený prístup,
- strata dôvery zákazníka/užívateľov. [13]

3.2 Ciele penetračných testov

Cieľom penetračných testov môže byť čokoľvek u čoho hrozí riziko nežiadúceho prieniku do systému, krádeže dát alebo spôsobenie škody obchodnej aktivity. V podstate môžeme zjednodušene povedať, že sa jedná o rovnaké ciele, ako majú hackeri. Patria sem:

- webové stránky (verejné a neverejné webové aplikácie),
- bezdrôtové siete,

- databázové servery,
- doménové radiče,
- vnútorné informácie o zamestnancoch, firemných klientoch a firemným know-how,
- e-mailové servery a schránky,
- prístupové heslá,
- úložisko dát a FTP servery,
- softvérové aplikácie a informačné systémy. [16]

3.3 Typy penetračných testov

V oblasti informačných technológií môžeme penetračné testy softvérov rozdeliť podľa spôsobu prevedenia. Jedná sa o:

- manuálne testy – sofistikované testy a testovacie postupy na mieru, sú však časovo náročné.
- Automatizované testy – rýchle a ľahko opakovateľné testy, neobsiahnu ale všetky možné varianty (je ťažšie ich prispôsobiť podľa situácie).
- Semiautomatické testy – ideálna kombinácia automatických a manuálnych testov s maximálnym využitím predností oboch spôsobov. [13]

Penetračné testy môžeme taktiež rozdeliť obdobne ako testy SW (kap.2) vo všeobecnosti na základe úrovne znalostí testovaného systému:

- black – box testovanie
- white – box testovanie
- grey – box testovanie. [13]

Okrem vyššie spomenutej kategorizácie môžeme hovoriť o rôznych formách penetračných testov:

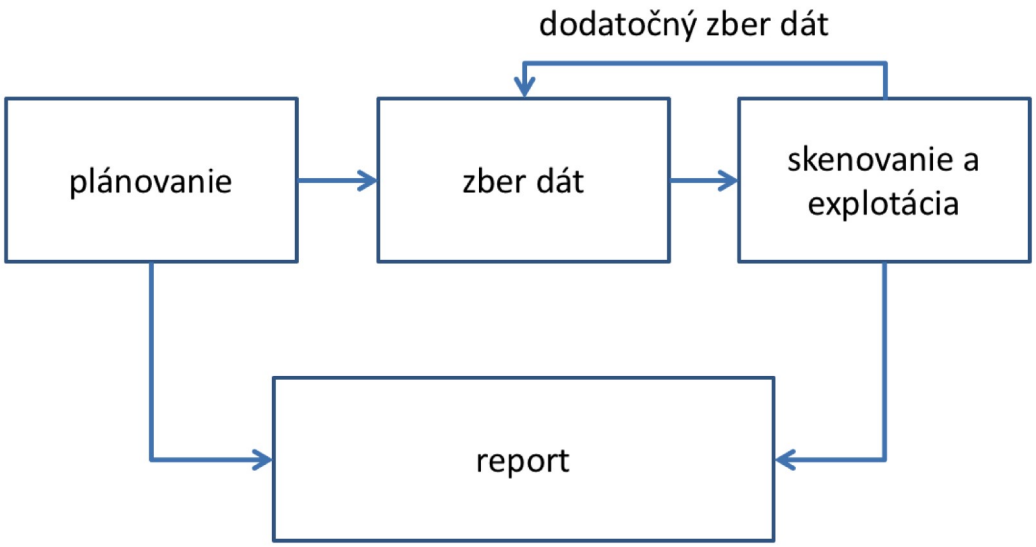
- ohlásený penetračný test – cieľom je otestovať celkovú odolnosť informačného systému a bezpečnostnej politiky v organizácii. O priebehu testu sú informovaný všetci zamestnanci firmy.
- Neohlásený penetračný test – cieľ je rovnaký ako u ohláseného testu, avšak informovaný je len vrcholový management, aby bolo možné otestovať aj dodržovanie bezpečnostných smerníc organizácie v reálnych (nepripravených) podmienkach.
- Externý penetračný test.
- Interný penetračný test.
- Penetračný test s využitím techník sociálneho inžinierstva. [16]

3.4 Priebeh penetračných testov

Postup penetračných testov popisuje priebeh penetračného testovania. Existujú viaceré prístupy, ktoré využívajú buď voľne dostupné nástroje – metodika je teda verejne známa. V mnohých prípadoch ale softvérové firmy testujú na základe vlastného know-how, ktoré si vybudovali za roky svojej činnosti a preto si postupy chránia.

Penetračné testovanie má vo všeobecnosti 4 fázy (Obr. 5):

- Fáza 1: Cieľ a rozsah penetračného testu (Plánovanie)
- Fáza 2: Zber dát
- Fáza 3: Exploitácia
- Fáza 4: Report [13][17]



Obr. 5: Proces penetračného testu [17]

3.4.1 Cieľ a rozsah penetračného testu

V prípravnej fáze je nutné zodpovedať niekoľko otázok.

- Čo budeme testovať?
- Ako budeme testovať (nástroje, typy testov)?
- Kedy budú testy vykonané?

Dôležité je hneď v prípravnej fáze vymedziť čo je obsahom testovania a čo nie. Celkový cieľ testovania sa rozdelí do dielčích cieľov aby sme dokázali určiť, či všetko má byť otestované a na čo sa musíme zamerať. [13]

3.4.2 Zber dát

Po dôkladnom plánovaní v prvej fáze penetračného testovania nasleduje fáza zberu dát. Informácie získavame za účelom pochopiť ako systém pracuje a lepšie zistiť, kde môžu byť zraniteľné miesta. Z testovanej siete / systému dokážeme získať informácie pomocou

rôznych techník, a to aktívne alebo pasívne. [18][17]

Pasívny zber dát je získavanie dát bez možnosti detekcie činnosti objektom testovania. Jedná sa o kombináciu získavania údajov z verejne dostupných zdrojov, a získavanie informácií predstieraním činnosti bežného užívateľa. Príkladom môže byť návšteva webovej stránky, použitie aplikácie. [18]

Aktívny zber dát je v určitých prípadoch možné rozoznať, jedná sa napríklad o skenovanie portov, ip adries, odchytávania sieťovej komunikácie a pod. V prípade, že cieľ disponuje IDS, IPS, firewall dokáže takýto zber dát odhaliť. [18]

Pomocou rôznych techník získavame nasledovné informácie:

- Informácie o hostiteľovi a IP adrese – získané napríklad pomocou DNS odpočúvaním, dotazy InterNIC (WHOIS), odpočúvanie sieťovej komunikácie
- Informácie o menách zamestnancov a kontakty – získavame pomocou prehľadávania firemných webových severov alebo doménových serverov
- Systémové informácie – získavané metódami ako sú enumerácia NetBIOS a NIS (Network Information System)
- Informácie o aplikáciách a services - napríklad verzie používaného operačného systému[17]

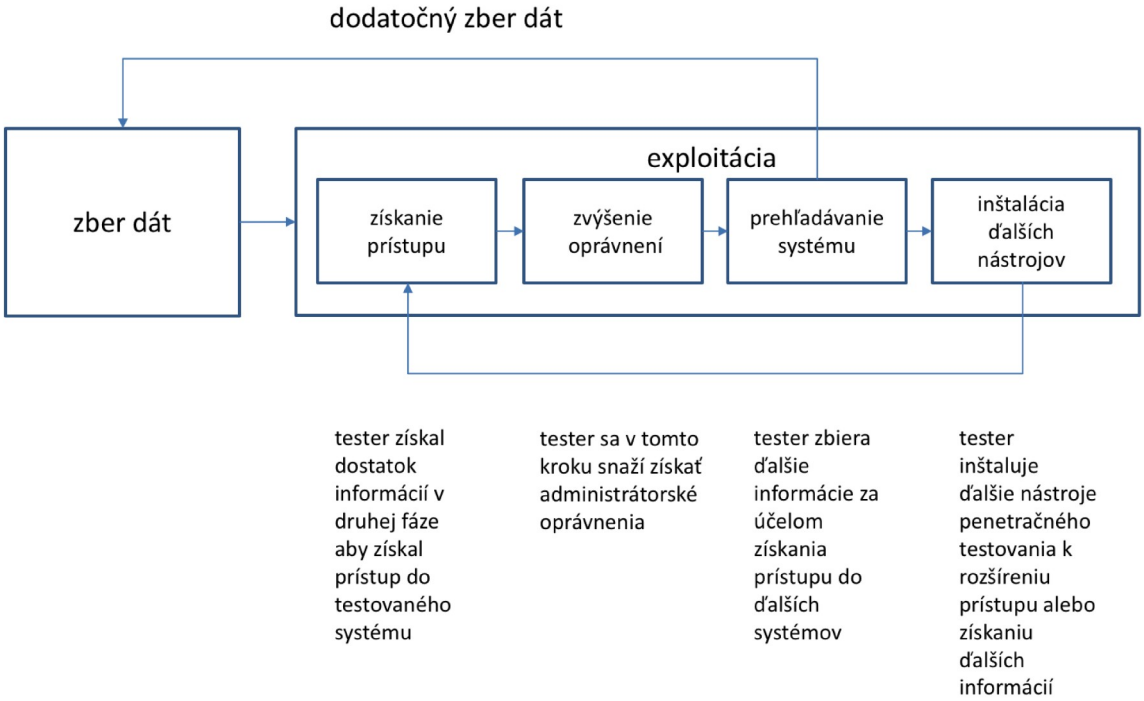
Súčasťou zberu informácií môžu byť použité i techniky ako je „dumpster diving“ tj. Prehľadávanie odpadkov alebo reálna prechádzka po firme. Takto dokážeme odhaliť ďalšie informácie o cieli penetračného testu ako je napr. heslo napísané na papieriku.[17]

V druhej časti zberu dát vyhodnotíme už získané informácie o systéme a tie využijeme pre analýzu zraniteľnosti systému s využitím databáz zraniteľnosti, ktoré sú verejne dostupné napr. National Vulnerability Database (NVD). Zraniteľnosti identifikujete manuálne z verejne dostupných zdrojov, táto časť je preto časovo náročná ale dokáže odhaliť aj tie slabé miesta testovaného systému, ktoré automatické skenery prehliadnu. [17]

3.4.3 Exploitácia

Samotná realizácia penetračného testu je proces, ktorý overuje identifikované potenciálne zraniteľné miesta systému/aplikácie. K tomu sa využívajú tzv. exploity. Exploit programy alebo exploit skripty sú nástroje, ktoré využijú zraniteľnosť systému. Ak je takýto útok úspešný, zraniteľnosť je potvrdená a nasleduje návrh mitigačného opatrenia k zamedzeniu bezpečnostného rizika. [13] [17]

Niektoré exploity prinesú nové informácie o systéme, pomocou ktorých tester identifikuje ďalšie zraniteľné miesta. Pomocou exploitov môže tester získať väčšie oprávnenia v systéme a to využiť k inštalácií nástrojov priamo v systéme a tím podporovať testovací proces (Obr. 6). [17]



Obr. 6: Schéma exploitácie s náväznosťou na predchádzajúcu fázu [17]

Najčastejšie zraniteľnosti exploitované počas penetračného testu spadajú do týchto kategórií:

- Nesprávna konfigurácia – nesprávne nastavené bezpečnostné prvky, často defaultné nastavenia, ktoré sú ľahto napadnuteľné

- Chyby kernelu – chyby v kóde OS, ktoré ohrozujú celý systém
- Pretečenie bufferu – možnosť spustenia nežiadúceho kódu s eskalovanými privilégiami
- Nedostatočná validácia vstupov
- Zneužitie symbolických linkov
- File descriptor útoky
- Race condition – chyba v systéme či procese, vznik nepredvídateľných výsledkov pri nesprávnom poradí alebo načasovaní vykonaných operácií
- Nesprávne práva súborov a priečinkov. [17]

3.4.4 Report

Poslednou fázou je vytvorenie reportu, ktorý by mal sumarizovať výsledky jednotlivých testov, pridať zistenia a poznatky, ktoré boli pri teste získané. Súčasťou reportu môžu byť v niektorých prípadoch aj odporúčania pre vyriešenie bezpečnostných problémov. [13]

Základné kritéria pre report:

- Zámer penetračného testovania – špecifikovanie cieľov penetračného testu,
- Technický report – pojednáva o technických detailoch testovania,
- Zhodnotenie – finálny prehľad testov, demonštrovanie dopadu zraniteľností na bezpečnosť systému, môže obsahovať stratégie pre nápravné opatrenia a plán pre ďalšie kroky zabezpečenia. [18]

Výsledky testovania v reporte musia byť prezentované v zrozumiteľnej forme ako pre technických, tak pre riadiacich pracovníkov.

3.5 Nástroje penetračného testovania

Počas celého procesu penetračného testovania testeri využívajú celú radu nástrojov ku skenovaniu systému, zberu informácií, dokumentáciu.

Kali Linux od spoločnosti Offensive Security zastrešuje veľkú skupiny nástrojov, jedná sa o linuxovú distribúciu, ktorá poskytuje testerom nie len nástroje ale zároveň aj prostredie pre testovanie. Jednotlivé nástroje je možné deliť do skupín podľa technickej náročnosti na obsluhu, podľa možnosti automatizácie alebo podľa spôsobu získavania informácií.

Google – najpoužívanejší vyhľadávač na svete, poskytuje mnoho možností pre pokročilé vyhľadávanie informácií. Pomocou operátorov a špeciálne skonštruovanými hľadanými výrazmi je možné získať verejne dostupné informácie o objekte testovania bez jeho vedomia. Základné informácie o webovej stránke, osobách a zákazníkovi môžu pomôcť plánovaní penetračného testu. [19]

Sociálne siete - dnešné trendy a doba nabádajú k zapojeniu sa do sociálnych sietí, aby si užívatelia, ale aj firmy jednoducho a efektívne oznamovali a vymieňali svoje informácie. Veľa ľudí si stále neuvedomujú riziká spojené so sociálnymi sieťami, informácie poskytujú širokej verejnosti a k dispozícií potenciálnym útočníkom. Prienik do systému môže byť zjednodušený pri zneužití týchto informácií. [20]

WHOIS – protokol pre získanie záznamov o majiteľovi domény, IP adresy alebo čísla ASN. Informácie vrátené pomocou WHOIS obsahujú údaje ako emailová adresa, kontaktné číslo, a ďalšie metadáta. [21]

Nslookup – aplikácia pre dotazovanie názvu serveru pre IP adresy špecifickej domény alebo hostiteľa na doméne. Je možné v určitých prípadoch získať detailné dáta o doméne. [21]

Dig – nástroj pre dotazovanie DNS serveru za účelom získavania informácií o cieli, jednoduchým povelom dig za ktorým nasleduje meno domény je možné získať základné informácie ako IP adresa atď.. [21]

Ping – jednoduchý test dostupnosti a časovej odozvy serveru, možné využiť na odhaľovanie IP adries v sieti. V určitých prípadoch a konfiguráciách je možné príkaz

zneužiť pre vytvorenie ping flood alebo ping of death útoku na DoS serveru. [21]

Traceroute – najjednoduchší spôsob identifikácie routeru je poslať traceroute na webovú stránku cieľa alebo známy server. V prípade, že spoločnosť má dostatočné zabezpečenie je použitie tohto nástroja limitované. [21]

Nmap – jedná sa o skener portov, pomocou rôznych parametrov je možné oskenovať veľký rozsah portov v krátkom čase. [21]

Metasploit Framework – nástroj navrhnutý na testovanie bezpečnosti systémov, využíva databázu známych chýb a exploitov. Na jeho vývoji sa podieľa rada bezpečnostných odborníkov. Nástroj je možné ovládať cez príkazový riadok, alebo cez radu GUI nástrojov. Pomocou neho dokážeme nájdenú chybu opakovane zneužiť. [20]

Armitage – nástroj na vizualizáciu útokov pomocou Metasploit frameworku alebo iných techník, nástroj je navrhnutý aby užívateľa viedol k ďalším krokom, podporuje pokročilé funkcie. [20]

Nccrack – jeden z nástrojov na „krekovanie“, prekonanie autentifikácie sieťových prvkov. Testuje nie len počítače ale aj sieťové zariadenia pomocou útokov hrubou silou, používaním tabuliek alebo iných techník. [20]

Burpsuite – grafický nástroj na testovanie bezpečnosti webových aplikácií, dostupný v platenej a free verzií. Zastrešuje kompletný proces testovania od skenovania, cez detekciu a identifikáciu zraniteľností či databázu exploitov. [22]

THC-Hydra – paralizovaný nástroj na prekonávania prihlasovacích protokolov s podporou veľkého množstva techník pre vytvorenie útoku. [23]

3.6 Hlavné prínosy penetračných testov:

- Nástroj pre zlepšenie bezpečnostného povedomia firmy.
- Preverenie informačnej bezpečnosti v praxi.
- Zdokumentovanie slabých miest a prienikov do informačného systému.

- Ilustrácie ako ľahko sa útok môže odohrať v praxi.
- Posúdenie pripravenosti a reakcie IT pracovníkov.
- Zhodnotenie odhalených bezpečnostných nedostatkov podľa stupňa ich závažnosti.
- Prevencia finančných strát.
- Eliminácia nedostupnosti služby.
- Eliminácia neoprávnených prístupov - zamedzenie neautorizované zmeny v konfigurácii servera či pracovnej stanice.
- Eliminácia získanie dôverných a citlivých informácií.
- Ochrana dobrého mena značky (zneužitie informácií v obchodnom styku).
- Eliminácia straty dôvery (napr. U dodávateľov).
- Identifikácia zraniteľnosti.
- Obraz reálneho posúdenia bezpečnostného zabezpečenia informačnej infraštruktúry firmy.
- Detailný technická správa opisujúca a hodnotiace zistené nedostatky a stupeň ich nebezpečnosti.
- Manažérska správa s odporúčanými krokmi pre nápravu nedostatkov a optimalizáciu prevádzky systému. [13][16][20]

II. PRAKTICKÁ ČÁST

4 NÁVRH METODIKY PENETRAČNÉHO TESTU

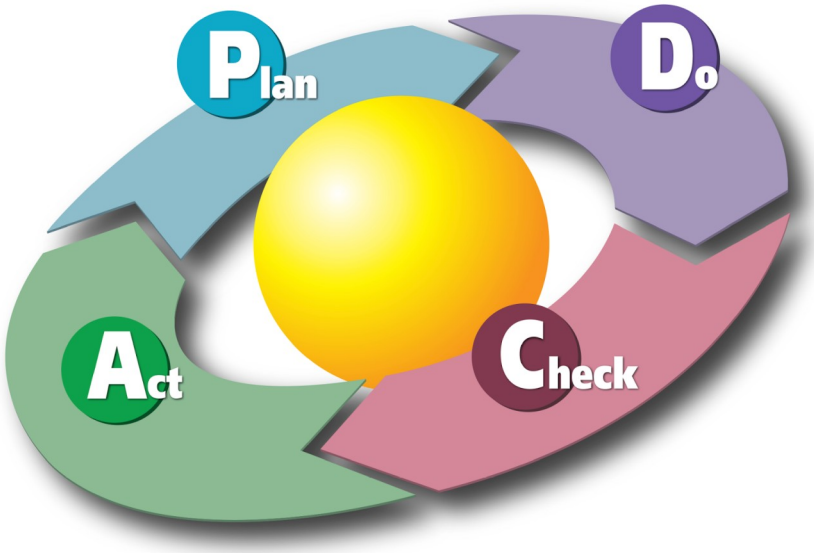
Z teoretickej časti vyplýva, že bezpečnosť webových aplikácií je veľmi komplexná problematika. Penetračné testy sú dôležitou súčasťou v boji proti kybernetickej kriminalite, je však na prevádzkovateľoch systémov a aplikácií ako pristupujú k otázke zabezpečenia svojich produktov a služieb. Neexistuje univerzálny návod, ako k penetračným testom pristupovať, je možné využívať open-source metodológie avšak testeri si na základe svojich skúseností vytvárajú vlastné know-how a postupy prispôsobujú na mieru danému produktu, ktorý testujú.

V praktickej časti tejto diplomovej práce sa budeme venovať návrhu metodiky, ktorú je možné aplikovať na penetračné testovanej webových aplikácií.

Pri tvorbe vychádzame z literárnej rešerše realizovanej v teoretickej časti a taktiež z vlastných poznatkov a skúseností s práce testera SW v IT.

4.1 PDCA

Pri zostavení metodiky penetračného testu budeme vychádzať z manažérskeho prístupu používaného ku kontinuálnemu zlepšovaniu a kontrole procesov a produktov. Táto metóda sa nazýva PDCA(Plan-do-check-act) inak známa aj ako Demingov kruh (Obr. 7). [24]



Obr. 7: PDCA cyklus [24]

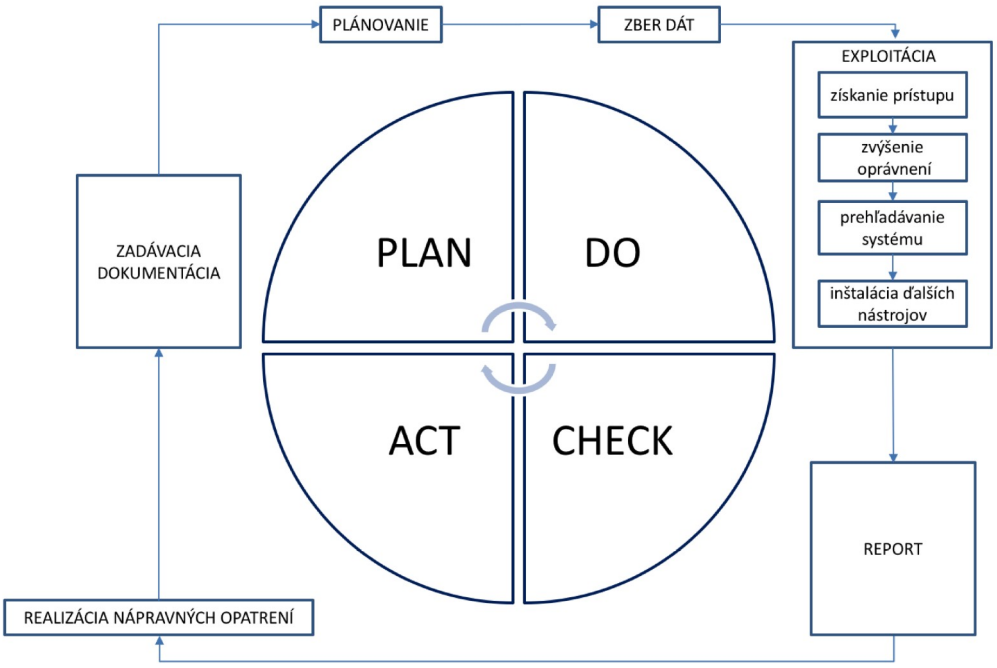
PDCA pozostáva zo štyroch navzájom nadväzujúcich činností:

- Plánovanie (Plan) – detailný plán projektu, identifikácia cieľov, rozdelenie úloh a akčný plán spoločne s definovanými milestonami
- Vykonávanie (Do) – realizácia akčného plánu, dosahovanie milestonov
- Kontrola (Check) – zhromaždenie a analýza výsledkov, identifikácia problémov a vypracovanie nápravných opatrení
- Akcia (Act) – realizácia nápravných opatrení. [24]

Vzhľadom na to, že sa jedná o kontinuálne vylepšovanie po realizácií fáze akcie, okamžite prechádzame znovu do ďalšieho štádia projektu a nachádzame sa znovu vo fáze plánovania. Získané informácie z „prvého kola“ využijeme v ďalšom cykle pre zlepšenie celého procesu. Táto metóda je využívaná v projektovom managemente. K overeniu bezpečnosti webových aplikácií väčšinou pristupujeme ako k projektu a preto sme ju zvolili ako vhodný postup pri penetračnom teste. Navyše na bezpečnosť aplikácií sa nedá pozerať ako na jednorazovú záležitosť, ale taktiež sa jedná o kontinuálne zlepšovanie bezpečnostných prvkov a preto sa metóda PDCA javí ako viac než vhodná.

4.2 PDCA ako metodika pre penetračné testovanie

Tak ako metóda PDCA tak aj navrhnutá metodika pozostáva zo štyroch častí. Na obrázku 8. vidíme základnú schému a aktivity v jednotlivých častiach.



Obr. 8: Základná schéma návrhu metodiky penetračného testovania

4.2.1 Fáza plánovania

V prvej časti projektu je nutné vytvoriť zadávaciu dokumentáciu na základe ktorej bude penetračné testovanie realizované. Zadávacia dokumentácia by mala obsahovať:

- zmluvu medzi zadávateľom a vykonávateľom (v prípade, že sa jedná o testovanie externé)
- stanovenie rozsahu testovaného systému
- stanovenie zodpovednej osoby za penetračné testovanie
- vyhradenie časového obdobia kedy bude penetračné testovanie vykonané
- popis použitých techník a nástrojov
- stanovenie ochrany osobných údajov
- dodatočné informácie od zadávateľa

Šablóna zadávacej dokumentácia bola vytvorená pre uľahčenie fázy plánovanie. (Príloha P1)

Cieľom prvej fázy je zdokumentovať, pripraviť podklad a naplánovať realizáciu penetračného testu, nastaviť pravidlá pre jeho vykonanie a definovať kedy, kto a akým spôsobom bude testovať.

Okrem vykonania plánovania je vhodné, aby zvolený zodpovedný pracovník prichystal počas tejto fázy HW a SW vybavenie na základe zvolených techník a nástrojov pre možnosť zahájenia penetračného testovania hneď po ukončení časti plánovania.

4.2.2 Fáza vykonávania

V tejto fáze má tester 3 základné úlohy spojené s vykonaním penetračného testu:

- zber dát a skenovanie,
- exploitácia,
- zhromaždenie výsledkov.

Na základe zvolených techník a nástrojov tester zbiera dáta pasívne z verejných zdrojov alebo aktívne použitím techník skenovania. Tester dokumentuje nie len informácie o nájdených zraniteľnostiach ale aj celý proces pre možnosti reprodukcie potenciálnych chýb alebo možnosti zlepšenia procesu v budúcnosti. Po nájdení zraniteľnosti a úspešnom zneužití môže znova nastať proces zbierania dát na základe nových prístupných informácií alebo systémov.

Pri zbieraní dát nás zaujímajú informácie ako štruktúra organizácie, zamestnanci, emailové adresy, jednotlivé webové adresy a sub-domény, IP adresy serverov, technológie na ktorých webové stránky bežia, služby, ktoré na serveroch bežia, otvorené sieťové porty, poskytovateľ hostingu, internetu a podobne.

Počas testovania je nutné aby testy boli dôsledné, čo najviac sa približovali reálnemu útoku hackera, využívali aktuálne techniky a nástroje a hlavne aby boli opakovateľné. V prípade odhalenia zraniteľností a aplikovaní protiopatrení v ďalších častiach procesu je nutné aby mohol byť test zopakovaný a aby bolo protiopatrenie potvrdené ako účinné. Ďalej by mali

byť výsledky testov merateľné, zodpovedať realite a zároveň aktuálne k danému obdobiu kedy proces penetračného testovania prebieha.

Zhromaždenie nazbieraných informácií, informácií o procesoch, testoch a výsledkoch testovania sú predpokladom pre ukončenie druhej fázy.

4.2.3 Fáza kontroly

Nazhromaždené informácie je nutné v rámci tejto časti zanalyzovať. Analyzované sú dáta, ktoré boli nájdené pri zbere dát, proces testovania ako taký a výsledky testovania. Nájdené bezpečnostné zraniteľnosti a exploity sú ohodnotené stupňom závažnosti a stupňom pravdepodobnosti zneužitia. Jednotlivé stupne by mali byť definované a zrozumiteľné. V tejto fáze je výstupným produktom report, kombinujúci dokumentáciu pre vedenie spoločnosti (zadávateľa) tak pre technických užívateľov (technická dokumentácia). V prípade nájdenia bezpečnostných zraniteľností a slabín by mali byť popísane a súčasťou reportu aj možnosti mitigácie a ochrany proti daným útokom.

Report by mal obsahovať nasledujúce údaje:

- súhrn cieľov a špecifikácie zo zadávacej dokumentácie
- technický report (časový záznam, použitý HW, SW a nástroje, výsledky, ...)
- zhodnotenie výsledkov pre netechnických pracovníkov
- odporúčania pre mitigáciu rizík
- odporúčania pre ochranu proti demonštrovaným útokom (ak boli objavené a zneužité zraniteľnosti)
- identifikáciu a podpis pracovníka vykonávajúceho penetračné testovanie

Všetky tieto body boli zahrnuté do šablóny pre výsledný report penetračného testovania, ktorá môže slúžiť pomôcka pri vyhodnocovaní výsledkov testov. (Príloha P2)

4.2.4 Fáza akcie

Poslednou fázou jedného cyklu PDCA je fáza akcie. Na strane vykonávateľa penetračného testu (testera) sú povinnosti jasné:

- prezentácia výsledkov penetračného testovania (reportu)
- predanie všetkých nazbieraných dát a dokumentácie
- konzultácia mitigačných opatrení a nápravných opatrení pre nájdené zraniteľnosti

Zadávateľ má v tejto časti dôležitú úlohu, prebrať jednotlivé dokumenty a porozumieť výsledkom testovania. Akékoľvek nejasnosti či nezrovnalosti je nutné okamžite vyriešiť. Ďalším krokom, ktorý by mal zadávateľ vykonať je realizácia nápravných opatrení, v tomto kroku môže tester alebo bezpečnostný konzultant ďalej zohrávať rolu.

Keďže hlavným dôvodom zvolenia metódy PDCA bol jej kontinuálny charakter, po fáze akcie začína automaticky príprava ďalšieho cyklu. Zadávacia dokumentácia sa môže aktualizovať na novú verziu s prihliadnutím na výsledky testovania predchádzajúceho cyklu a taktiež na realizované nápravné opatrenia.

5 PENETRAČNÝ TEST WEBOVEJ APLIKÁCIE

Test bol vykonaný na reálnej webovej aplikácie, z dôvodu ochrany identity spoločnosť a informácií osôb, ktoré boli počas penetračného testovania získané budú dáta v diplomovej práci anonymizované. Doména spoločnosti je zamenená za doménu „http://testovana-domena.dk/“. Takáto doména nie je nikde zaregistrovaná a preto by použitie tohto názvu nemalo spôsobiť žiadny problém. Mená zamestnancov sú zamenené za náhodné mená vytvorené generátorom. Telefónne čísla a IP adresy, alebo iné údaje, ktoré by mohli viesť k identifikácií budú čiastočne v dátach nahradené znakmi „X“ - napr. z IP adresy 127.0.0.1 sa stane XXX.XXX.0.1. Všetky namerané dáta sú reálne a odpovedajú skutočnosti v čase vykonávania penetračného testovania.

Jedná o interné penetračné testovanie spoločnosti, firma s vykonávaným testovaním súhlasila avšak nie je dostupný súhlas vo forme zmluvy. V čase písania diplomovej práce sú nájdené problémy už odstránené, ale aj napriek tomu sú informácie pre ochranu firmy a osôb anonymizované.

5.1 Fáza plánovania

V rámci plánovania penetračného testovania boli stanovené ciele penetračného testu, rozsah testovania, časový harmonogram, použité techniky a nástroje. Ďalej boli zadefinované obmedzenia penetračného testovania pre zabezpečenie neporušenej dostupnosti webových stránok počas pracovného týždňa. Kompletný popis je možné vidieť v zadávacej dokumentácií penetračného testu (Tab. 4).

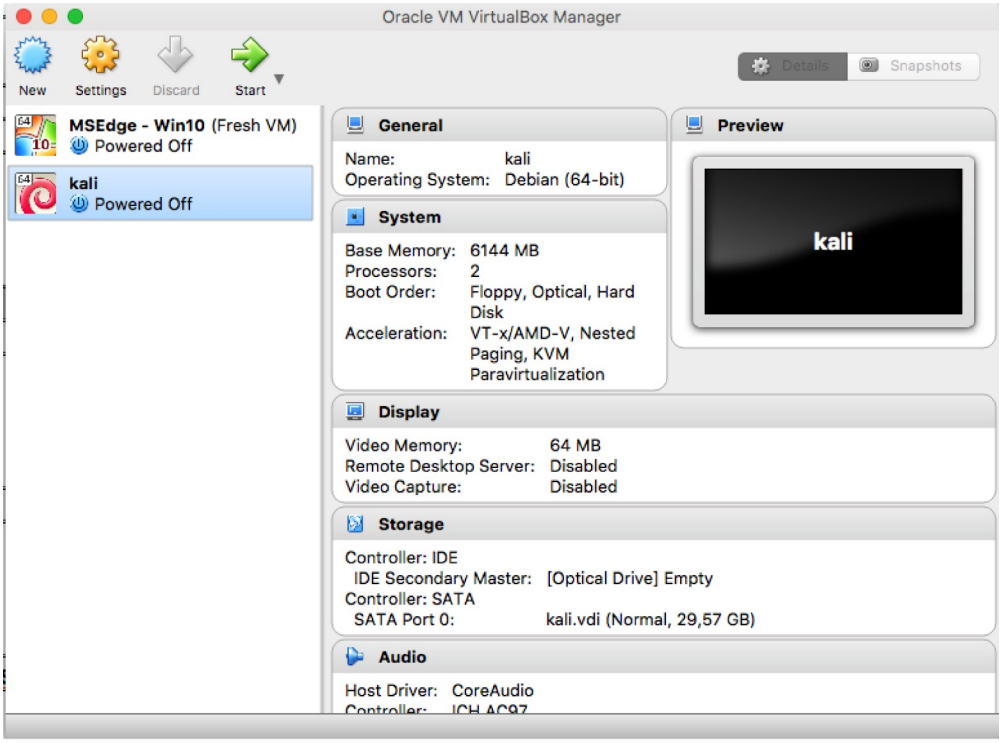
Ďalším krokom vo fáze plánovania bola príprava HW a SW pre začatie testovania vo fáze 2. Pre vytvorenie izolovaného prostredia a zároveň prostredia, ktoré sa dá preniesť a jednoducho znovu použiť (dajú sa reprodukovať testy) bol využitý virtualizačný softvér spoločnosti ORACLE® - Virtual Box. Pomocou tohto nástroje je možné virtualizovať OS. Virtualizovaýn OS sa navonok javí ako štandardný systém, avšak užívateľ má plnú kontrolu nad pripojenými zariadeniami, nastaveniami virtuálnych sieťových prvkov a pod. Zároveň je možné „virtuálny stroj“ preniesť a spustiť na inom hostiteľskom PC (Obr. 9).

Tab. 4: Zadávacia dokumentácia reálneho penetračného testu

Zadávacia dokumentácie penetračného testovania			
Zadávateľ		**Vykonávateľ**	
Názov:	Testovaná-doména	**Názov:**	Oliver Polka
Adresa:	Smetanova 2, Martin	**Adresa:**	Štefánikova 17/23, Zlín
Zodpovedná osoba:	Justin Case	**Zodpovedná osoba:**	Oliver Polka
Kontakt:	+420 5X XX XX X5	**Kontakt:**	+420 607 501 291
Ďalšie informácie:		**Ďalšie informácie:**	
Popis projektu			
Cieľ penetračného testovania:			
Cieľom penetračné testovania je overenie bezpečnosti webovej aplikácie na doméne http://testovana-domena.dk a serveru, na ktorom je daná aplikácia hostovaná			
Rozsah penetračného testovania:			
Webová aplikácia na url addrese http://testovana-domena.dk, ďalšie webové aplikácie dostupné na rovnakom servery. Hlavným základom je testovanie zraniteľností podľa OWASP TOP 10 projektu, testovanie pomocou automatizovaných nástrojov, odolnosť proti malým až stredným útokom DoS.			
Časový harmonogram penetračného testovania:			
10.01.2017 - Zahájenie penetračného testovania, príprava prostredia HW a SW 11.01.2017 – 21.01.2017 – Realizácia penetračného testovania 21.01.2017 – 25.01.2017 – Analýza výsledkov a vytvorenie reportu 25.01.2017 – Odovzdanie reportu, konzultácia výsledkov s IT oddelením spoločnosti			
Typy a zameranie penetračného testovania:			
- kombinácia automatizovaného a manuálneho testovania, využitie nástrojov Kali Linux pre odtestovanie aplikácie podľa zadaného rozsahu penetračného testovania			
Použitý HW a SW:			
- HW bezpečnostného pracovníka (vykonávateľ), linuxová distribúcia Kali Linux, nástroje Google, WHOIS, DIG, Nmap, Uniscan a Metasploit a iné			
Ďalšie požiadavky:			
- penetračné testovanie, ktoré by mohlo negatívne ovplyvniť prevádzku webových stránok musí byť vykonané vo večerných hodinách alebo počas víkendu - tester môže použiť deštruktívne metódy, prípadné vážne problémy (nedostupnosť aplikácie) musí okamžite hlásiť zodpovednej osobe zadávateľa			
Zoznam príloh:			
Dátum:	10.1.2017	**Podpis zadávateľa:**	
		Podpis vykonávateľa:	

Dôležitým nástrojom je OS, pre penetračné testovanie bol zvolený operačný systém Kali Linux od spoločnosti Offensive Security. Kali Linux – linuxová distribúcia špecializovaná

na forenznú analýzu a penetračné testy, predstavuje nástupcu populárneho systému BackTrack. Je možné ju využívať ako tzv. „Live“ CD, kedy nie je nutné vo virtuálnom stroji inštalovať túto distribúciu na disk alebo ako klasickú linuxovú distribúciu po nainštalovaní (Obr. 10).



Obr. 9: GUI programu Virtual Box

Pre plynulejšiu a stabilnú prácu bol systém nainštalovaný do virtuálneho stroja. Kali Linux poskytuje veľké množstvo nástrojov z verejne známych, alebo možnosť doinštalovať akékoľvek ďalšie balíčky (systém je postavený na populárnej distribúcií Debian, s ktorou je plne kompatibilný). Medzi najznámejšie nástroje Kali Linux patria – SQLMap, Nmap, Hydra, Wireshark, Burpsuit, Aircrack-ng, John the ripper, či framework Metasploit.

Možnosti jednotlivých nástrojov boli preskúmané a niekoľko z nich bolo zvolených pre využitie v testovaní.

Obr. 10: Užívateľské prostredie Kali Linux

5.2 Fáza vykonávania

Realizácia penetračného testu, v prvom rada začína so zberom informácií z verejných zdrojov.

Google

Pomocou vyhľadávacieho nástroja Google boli získané kontaktné informácie na niekoľko zamestnancov spoločnosti, ich tel. čísla, emailové adresy a pozícia v spoločnosti. Prehliadaním výsledkov vyhľadávania kombináciou rôznych permutácií mena domény boli objavené ďalšie webové stránky patriace pod rovnakú spoločnosť. Vzhľadom na fakt, že dané webové adresy neboli hostované na rovnakom webovom servery, boli z ďalšieho penetračného testovania vylúčené.

Ďalšie informácie z verejných zdrojov boli získané pomocou jednoduchých webových nástrojov spoločnosti eset na stránke www.paranoia.cz. [25]

WHOIS

Výsledky dotazu WHOIS nám poskytujú informácie o registrantovi danej domény, mene administrátora a jeho kontaktné informácie. Taktiež nám dávajú informácie o sídle firmy (Obr. 11).

```
paranoia.cz > # Hello 82.99.191.151. Your session has been logged.
Domain: testovana-domena.dk
DNS: testovana-domena.dk
Registered: 2011-01-21
Expires: 2017-01-31
Registration period: 1 year
VID: no
Dnssec: Unsigned delegation
Status: Active

Registrant
Handle: CRIC1-DK
Name: Testovana Domena
Address: Ve XXXXX 20
Postalcode: 11000
City: Praha 1
Country: CZ
Phone: +420 29 XX XX XX 1

Administrator
Handle: IL1605-DK
Name: Justin Case
Address: Ve XXXXX 20
Postalcode: 11000
City: Praha 1
Country: CZ
Phone: +420 29 XX XX XX 1

Nameservers
Hostname: ns.testovana-domena.cz
Handle: CRIC1-DK
Hostname: ns3.testovana-domena.cz
Handle: AAS119-DK
```

Obr. 11: Odpoveď na dotaz WHOIS

DIG

Nástroj DIG, niekedy označovaný aj ako Nslookup slúži k nájdeniu DNS záznamov pre doménu alebo IP adresu. Je možné ho pustiť v troch módoch A, MX, PTR. V rámci testovania boli otestované všetky varianty z výsledkov dotazov neboli získané žiadne zaujímave údaje. Výsledok dotazu DIG typu A môžeme vidieť na (Obr. 12).

```
paranoia.cz >
; <<>> DiG 9.9.5-9+deb8u9-Debian <<>> testovana-domena.dk
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52693
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;testovana-domena.dk. IN A

;; ANSWER SECTION:
testovana-domena.dk. 86187 IN A XXX.XXX.192.3

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Jan 18 15:55:23 CET 2017
;; MSG SIZE rcvd: 55
```

Obr. 12: Odpoveď na DIG dotaz typu A

Traceroute

Pomocou trace sme schopný mapovať cestu, ktorou cestujú pakety k zadanému cieľu, z výsledku (Obr. 13) môžeme vydedukovať, že daný cieľ sa s veľkou pravdepodobnosťou nachádza v Prahe a stojí za jedným z veľkých internetových poskytovateľov.

```
paranoia.cz > traceroute XXX.XXX to XXX.XXX.192.3 (XXX.XXX.192.3),
20 hops max, 60 byte packets
1 82-99-191-1.static.bluetone.cz (82.99.191.1) 1.558 ms 1.524 ms
1.502 ms
2 et-0-3-0-100-mx-site4.bluetone.cz (84.244.124.40) 1.484 ms 1.468
ms 1.450 ms
3 nix4.t-mobile.cz (91.210.16.50) 1.476 ms 1.479 ms 1.787 ms
4 ph3918-ea1-tge0-4-0-0.cz.net (213.29.169.134) 3.164 ms ph3918-ea1-
tge0-2-0-0.cz.net (213.29.169.130) 3.701 ms ph3918-ea1-
tge0-4-0-0.cz.net (213.29.169.134) 3.317 ms
5 ws146.zzz.invex.cz (XXX.XXX.37.146) 3.469 ms 3.457 ms 3.691 ms
6 * * *
7 * * *
```

Obr. 13: Výsledok trasovania cesty paketov

Nástroj Traceroute je posdledný, ktorý ešte v rámci procesu zíkavania dát môžeme zaraďovať do tzv. „pasívneho“ zberu. Ďalšie nástroje už sú charakteru „aktívnych“, teda aktívne skenujú alebo zasahujú do testovaného cieľa. Je možné v určitých prípadoch zaznamenať, že útočník mapuje aplikáciu alebo, že penetračný test je v priebehu.

Uniscan

Nástroj Uniscan v sebe kombinuje niekoľko nástrojov skenovania webových stránok. Nmap, WHOIS, Nslookup, ale aj nástroje pre hľadanie zraniteľností typu Remote File Include, Local File Include, Remote Command Execution.

Vo výsledkoch skenovania nmap pluginu nástroja Uniscan môžeme vidieť niekoľko na prvý pohľad otvorených portov:

```
 Nmap scan report for www.testovana-domena.dk (XXX.XXX.192.3)
| Host is up (0.0054s latency).
| Not shown: 938 filtered ports
| PORT      STATE SERVICE    VERSION
| 33/tcp    open  tcpwrapped
| 70/tcp    open  tcpwrapped
| 80/tcp    open  http       nginx
| 161/tcp   open  tcpwrapped
| 443/tcp   open  ssl/http   nginx
| 625/tcp   open  tcpwrapped
| 646/tcp   open  tcpwrapped
| 720/tcp   open  tcpwrapped
| 787/tcp   open  tcpwrapped
| 873/tcp   open  tcpwrapped ...
```

Až stovky ďalších portov v reporte označených ako otvorené „*873/tcp open tcpwrapped*“, v skutočnosti nie sú prístupné a je nutné situáciu zanalyzovať v ďalšej časti cyklu PDCA.

Ďalšou zaujímavou časťou reportu je sekcia – Interesting Strings in HTML – môže často odhaliť chybné HTML stringy, naozaj zaujímavé časti kódu, ktoré identifikujú používané služby a podobne. V tomto prípade plugin zreportoval nasledujúce útržky kódu:

```
| INTERESTING STRINGS IN HTML
|
| li>Databases
| small>Assistant Vice President, Commerzbank, Frankfurt
|        a        href="https://twitter.com/testovana-domena"
class="twitter-icon"
onclick="ga('send','event','Footer','Click','Social        -
Twitter')">
|                                                          a
href="https://www.facebook.com/pages/TestovanaDomena/13357094
6685052"                                  class="facebook-icon"
onclick="ga('send','event','Footer','Click','Social        -
Facebook')">
```

Sada vulnerabilitít otestovaná pomocou nástroja Uniscan:

- FCKeditor,
- Timthumb < 1.33,
- SQL Injection,
- Local a Remote files include,

- PHP CGI Argument Injection,
- Remote Command Execution,
- XSS,
- Web shell finder.

Všetky uvedené testy prešli v poriadku, žiadna zraniteľnosť z vyššie uvedených nebola potvrdená.

Zaujímavá informácia bola odhalená v súbore robots.txt:

```
| Check robots.txt:
| [+] User-agent: *
| [+] Disallow: /a/
| [+]
| [+] Sitemap: /sitemap.xml
```

Z tohto dokumentu sa dozvedáme o existencií stránky s url „http://testovana-domena.dk/a/“, ktorá je odstránená z crawlerov vyhľadávacích nástrojov. Otvorením danej stránky bol nájdený prihlasovací formulár inak nedostupný pre bežných užívateľov a ktorý sa nenachádza v mapách stránky. Prihlasovací formulár môžeme využiť pre ďalšie testy, napríklad brute force útok.

Poslednou sekciou, ktorá prináša výsledky je nástroj Email Detection, ktorý bol schopný získať z webovej stránky množstvo emailových adries zamestnancov:

```
| E-mails:
| [+] E-mail Found: fXXXXXa@testovana-domena.dk
| [+] E-mail Found: nXXXXXn@testovana-domena.dk
| [+] E-mail Found: jXXXXXs@testovana-domena.dk
| [+] E-mail Found: vXXXXXa@testovana-domena.dk
| [+] E-mail Found: pXXXXXa@testovana-domena.dk
| [+] E-mail Found: iXXXXXo@testovana-domena.dk
| [+] E-mail Found: eXXXXXy@testovana-domena.dk
| [+] E-mail Found: jXXXXXa@testovana-domena.dk
| [+] E-mail Found: bXXXXXa@testovana-domena.dk
| [+] E-mail Found: sXXXXXa@testovana-domena.dk
| [+] E-mail Found: rXXXXXr@testovana-domena.dk
| [+] E-mail Found: sXXXXXa@testovana-domena.dk
| [+] E-mail Found: sXXXXXa@testovana-domena.dk
| [+] E-mail Found: dXXXXXs@testovana-domena.dk
| [+] E-mail Found: sXXXXXy@testovana-domena.dk
```

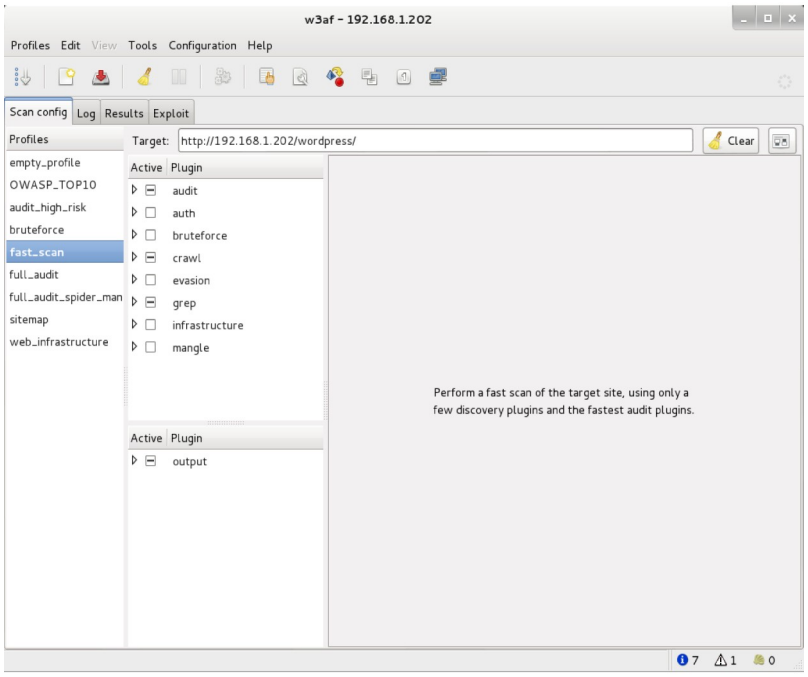
```
| [+] E-mail Found: jXXXXXn@gmail.com
| [+] E-mail Found: kXXXXXk@testovana-domena.dk
| [+] E-mail Found: vXXXXXu@testovana-domena.dk
| [+] E-mail Found: cXXXXXj@testovana-domena.dk
| [+] E-mail Found: uXXXXXy@testovana-domena.dk
| [+] E-mail Found: jXXXXXa@testovana-domena.dk
| [+] E-mail Found: sXXXXXa@testovana-domena.dk
```

Veľké množstvo emailových adries zvyšuje množstvo SPAMu, ktorý môžu zamestnanci dostávať, ale predstavuje aj nebezpečenstvo vytvorenia sofistikovaného phishingového útoku.

W3af

Tento framework sa zameriava na vytváranie bezpečnostných auditov pre identifikáciu zraniteľností webových aplikácií a hľadanie exploitov. Balíček v OS Kali Linux obsahuje GUI prostredie pre jednoduché ovládanie (OBR). Framework obsahuje viac ako 130 pluginov pre identifikáciu a exploitáciu zraniteľností od SQL injections až po XSS. [26]

Jedna z významných testovacích sád obsahuje testy zamerané priamo na testovanie zraniteľností OWASP TOP 10 a preto sme ho použili.



Obr. 14: Užívateľské prostredie frameworku w3af [26]

Pomocou nástroja bol spustený profil OWASP TOP 10, bola nájdená jedna zraniteľnosť:

```
[Št 19. leden 2017, 21:36:27 CET - vulnerability] The whole
target has no protection (X-Frame-Options header) against
Click-Jacking attacks. This vulnerability was found in the
requests with ids 17, 51, 57, 58, 74, 81, 132, 180 to 190.
```

Metasploit

Nástroj Metasploit bol užitočný pri vytváraní DoS útokov na webový server, kde aplikácia bežala. Boli prevedené útoky typu SYN flood z rôznymi parametrami. Webové stránky počas útoku reagovali spomalene, avšak k úplnému výpadku služby nedošlo.

Postup spustenia jednoduchého syn flood útoku je nasledovný:

1. *msf > use auxiliary/dos/tcp/synflood* – výber typu exploitu.
2. msf *auxiliary(synflood) > set RHOST XXX.XXX.192.3* – zvolenie cieľovej adresy.
3. msf *auxiliary(synflood) > set RPORT 80* – zvolenie sieťového portu.
4. msf *auxiliary(synflood) > exploit.*

SQLmap

Open-source nástroj na automatizované penetračné testovanie, detekciu SQL injection chýb a získavanie kontroly nad databázovými servermi. SQLmap je aplikovateľný na stránkach s odosielacími formulármi. V rámci testovanej aplikácie bolo vybraných niekoľko:

- http://testovana-domena.dk/kontakty
- https://testovana-domena.dk/a/
- http://testovana-domena.dk/clanok

Ani na jednej stránke nebola objavená zneužiteľná zraniteľnosť.

Hydra

Pri testovaní bol taktiež vyskúšaný nástroj hydra na bruto force slovníkový útok proti prihlasovaciemu formuláru na adrese „https://testovana-domena.dk/a/“. Ako slovník bol použitý známy zoznam používaných hesiel rockyou.txt. Pokus o získanie prístupu do

systému touto technikou nebolo úspešné.

5.3 Fáza kontroly

Analyzovali sme nazbierané dáta a proces exploitácie. Na základe výsledkov analýzy bol vytvorený report penetrčaného testovania ku ktorému sme využili pripravenú šablónu (Príloha 2). (Tab. 5)

5.4 Fáza akcie

Vytvorený report bol predaný zadávateľovi. Ďalšie kroky pre implementáciu nápravnych opatrení sú na uvážení zadávateľa projektu penetračného testovania.

V rámci diskusie o nájdených zraniteľnostiach so zadávateľom a pracovníkmi IT oddelenia bola zistená skutočnosť, že zdanlivo „otvorené“ porty naskenované pri penetračnom testovaní nie sú v skučtonosti chybou a zároveň nie sú v skutočnosti otvorené. Technika, ktorú používa firewall spoločnosti CISCO, ktorej produkty firma využíva vytvára falošný dojem nezabezpečenia sieťových prvkov. To slúži k odlákaniu aktivity hackera, ktorý sa snaží porty využiť pri ďalších útokoch.

Tab. 5: Report penetračného testovania

Report penetračného testovania			
Zadávateľ		**Vykonávateľ**	
Názov:	Testovaná-doména	**Názov:**	Oliver Polka
Adresa:	Smetanova 2, Martin	**Adresa:**	Štefánikova 17/23, Zlín
Zodpovedná osoba:	Justin Case	**Zodpovedná osoba:**	Oliver Polka
Kontakt:	+420 5X XX XX X5	**Kontakt:**	+420 607 501 291
Ďalšie informácie:		**Ďalšie informácie:**	
Špecifikácia penetračného testovania:			
- Cieľom penetračné testovania je overenie bezpečnosti webovej aplikácie na doméne http://testovana-domena.dk a serveru, na ktorom je daná aplikácia hostovaná - základom testovania je zoznam bežných zraniteľností zhrnutých v projekte OWASP TOP 10 - testovanie kombináciou automatizovaných nástrojov a manuálneho testovania			
Technický report:			
Testovanie prebehlo podľa odhadovaného časového harmonogramu Využité nástroje: Kali Linux, WHOIS, DIG, Traceroute, Google, Uniscan, Hydra, Metasploit Framework, w3af, SQLmap Známe odtestované útoky patria: - SQL Injections, - Cross-Site Scripting, - DoS útok, - Click-jacking, - Remote code execution test, - Local File Inclusion test, - Remote File Inclusion test, - brute-force útok na prihlasovaciu stránku administrácie webového serveru, - overenie známych zraniteľností použitých technológií, - sensitive data exposure and harvesting. Boli odhalené zraniteľnosti: 1. Click-jacking – nebola detekovaná žiadna ochrana proti tomuto druhu útoku. 2. Na serveroch sa nachádza množstvo otvorených portov – potenciálna hrozba pre DdoS útoky. 3. Z webových stránok je možné jednoducho harvestovať emailové adresy – SPAM.			
Odporúčané opatrenia:			
1. Zmena zdrojového kódu aplikácie pre odstránenie zraniteľnosti click-jackingu napr. na základe návodu projektu OWASP “https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet” 2. Uzatvorenie nepotrebných portov 3. Identifikovať a pokúsiť sa zmenšiť množstvo informácií o zamestnancoch dostupných na webovej stránke			
Zhodnotenie výsledkov:			
Webová aplikácia odolala penetračným testom OWASP TOP 10, a nebola nájdená žiadna zraniteľnosť zo zoznamu najzávažnejších zraniteľností projektu OWASP. Nájdené zraniteľnosti nie sú kritického charakteru, preto môžeme hodnotiť stav bezpečnosti testovanej webovej aplikácie za dobrý. Ďalším kladným bodom je, že webová aplikácia odolala jednoduchým DoS útokom, snahe o injekciu stránky SQL kódom alebo XSS. Prihlasovací formulár naviac odolal aj slovníkovému útoku hrubou silou.			
Dátum:	25.1.2017	**Podpis zadávateľa:**	
		Podpis vykonávateľa:	

ZÁVER

Penetračné testy sú neodeliteľnou súčasťou testovania softvéru, či už pri jeho vývoji alebo po uvedení na trh, pri aktualizáciách alebo pri bezpečnostných auditoch organizácií, ktoré softvér používajú. Pre efektívne prevedenie penetračných testov je nutné dobre poznať cieľ – teda ako daná aplikácia / softvér pracuje, aké má funkcie a môže mať slabé stránky.

Teoretické princípy testovaniu softvéru s dôrazom na penetračné testy boli popísané formou literárnej rešerše v teoretickej časti tejto diplomovej práce. Práca testerov si vyžaduje veľmi dobré technické znalosti, skúsenosti ale aj značnú dávku kreativity, aby dokázali intuitívne prispôsobiť použité nástroje a techniky pre efektívne otestovanie softvéru.

Neexistuje univerzálny prístup k penetračným testom, pretože každá aplikácia alebo systém si vyžaduje individálne posúdenie. Napriek tomu je možné vytvoriť všeobecnú metodiku k prevedeniu penetračného testu ako je prezentované v praktickej časti diplomovej práce. Z vlastných skúseností testera sa osvedčil prístup PDCA, ktorý využíva projektový prístup kontinuálneho zlepšovania procesov alebo produktov.

Cieľom uskutočneného penetračného testu bolo overenie bezpečnosti webovej aplikácie a serveru, na ktorom je daná aplikácia hostovaná. Základom testovania bol zoznam bežných zraniteľností v projekte OWASP TOP 10. Testovanie prebehlo kombináciou automatizovaných nástrojov a manuálneho testovania. Výsledky boli spracované formou reportu, ktorý zahŕňal nástroje, odhalené zraniteľnosti a odporúčané opatrenia k náprave bezpčenostných slabín. Na základe prístupu PDCA bola vytvorená šablóna a metodika, ktorú je možno využiť pri iných penetračných testoch.

S ohľadom na rýchly technologický pokrok s oblasti IT sa vynárajú stále nové riziká pre zabezpečenie webových aplikácií, ktoré testeri pri realizácií penetračných testov musia odhaliť. Práve preto je téma penetračných testov a testovania bezpečnosti IT systémov veľmi aktuálna a predstavuje potenciál pre ďalšií výskum.

SEZNAM POUŽITÉ LITERATURY

[1] Policie ČR. Kyberkriminalita [online]. 2018 [cit. 2018-05-20]. Dostupné z: http://www.policie.cz/clanek/kyberkriminalita.aspx

[2] Internet Crime Complaint Center. About IC3 [online]. 2018 [cit. 2018-05-20]. Dostupné z: https://www.ic3.gov/about/default.aspx

[3] Internet Crime Compliant Center. Internet Crime Report 2017 [online]. 2018 [cit. 2018-05-20]. Dostupné z: https://pdf.ic3.gov/2017_IC3Report.pdf

[4] OWASP. Open Web Application Security Project [online]. 2018 [cit. 2018-05-20]. Dostupné z: https://www.owasp.org/index.php/Main_Page

[5] OWASP: The Open Web Application Security Project. OWASP Top Ten Project [online]. 2018 [cit. 2018-05-20]. Dostupné z: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[6] PATTON, Ron. Testování softwaru. Praha: Computer Press, 2002. ISBN 80-7226-636-5

[7] HETZEL, William C. he complete guide to software testing. 2nd ed. . Wellesley, Mass.: QED Information Sciences, 1988. ISBN 978-0894352423

[8] IEEE: Institute of Electrical and Electronics Engineers. ISO/IEC/IEEE 29119-1:2013(E) - ISO/IEC/IEEE International Standard - Software and systems engineering —Software testing —Part 1:Concepts and definitions [online]. 2013. Dostupné z: http://standards.ieee.org/findstds/standard/29119-1-2013.html

[9] Guru99. Functional Testing Vs Non-Functional Testing: What's the Difference? [online]. 2018 [cit. 2018-05-20]. Dostupné z: https://www.guru99.com/functional-testing-vs-non-functional-testing.html

[10] US-CERT: United States Computer Emergency Readiness Team. Introduction to Information Security [online]. 2013. Dostupné z: https://www.us-cert.gov/security-publications/introduction-information-security

[11] Guru99. What is Security Testing: Complete Tutorial [online]. 2018 [cit. 2018-05-

20]. Dostupné z: https://www.guru99.com/what-is-security-testing.html

[12] ISECOM: Institute for Security and Open Methodologies. OSSTMM 3 – The Open Source Security Testing Methodology Manual [online]. 2010 [cit. 2018-05-20]. Dostupné z: http://www.isecom.org/mirror/OSSTMM.3.pdf

[13] SELECKÝ, Matúš. Penetrační testy a exploitace. Brno: Computer Press, 2012. ISBN 978-80-251-3752-9

[14] Policie ČR. Jednotlivé druhy kyberkriminality [online]. 2018 [cit. 2018-05-20]. Dostupné z: http://www.policie.cz/clanek/jednotlive-druhy-kyberkriminality.aspx

[15] Chris Hoffman. Hacker Hat Colors Explained: Black Hats, White Hats, and Gray Hats [online]. 2013. Dostupné z: https://www.howtogeek.com/157460/hacker-hat-colors-explained-black-hats-white-hats-and-gray-hats/

[16] KLEVINSKY, T. J., LALIBERTE, Scott a GUPTA, Ajay. Hack I.T.: security through penetration testing. Boston: Addison-Wesley, 2002. ISBN 0-201-71956-8

[17] NIST: National Institute of Standards and Technology. TGISTA: Technical Guide to Information Security Testing and Assessment [online]. 2008 [cit. 2018-05-20]. Dostupné z: https://csrc.nist.gov/publications/detail/sp/800-115/final

[18] ALLEN, Lee a CARDWELL, Kevin. Advanced penetration testing for highly-secured environments: employ the most advanced pentesting techniques and tools to build highly-secured systems and environments. Second edition.. Birmingham: Packt Publishing, 2016. ISBN 978-1-78439-581-0

[19] LONG, Johny. Google hacking for penetration testers. Burlington, MA: Syngress, 2008. ISBN 978-1-59749-176-1

[20] 20: SLEZÁK, Zbyněk, Bezpečnostní audity a pentesting ve firemním prostředí, 2013

[21] FAIRCLOTH, Jeremy. Penetration tester's open source toolkit. 3rd ed.. Waltham, MA: Elsevier/Syngress, 2011. ISBN 978-1-59749-627-8

[22] Portswigger. Burp Suite Editions [online]. 2018 [cit. 2018-05-20]. Dostupné z: https://portswigger.net/burp

[23] Offensive Security. THC-Hydra [online]. 2014 [cit. 2018-05-20]. Dostupné z: https://tools.kali.org/password-attacks/hydra

[24] Karn Bulsuk. Taking the First Step with the PDCA (Plan-Do-Check-Act) Cycle [online]. 2009. Dostupné z: https://www.bulsuk.com/2009/02/taking-first-step-with-pdca.html

[25] Eset. Paranoia - Internetové nástroje [online]. 2018. Dostupné z: http://www.paranoia.cz/tools

[26] Andres Riancho. w3af [online]. 2014 [cit. 2018-05-20]. Dostupné z: https://tools.kali.org/web-applications/w3af

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

API	Application Programming Interface
atď.	a tak ďalej
ASN	Autonomous System Number
ČR	Česká Republika
DoS	Denial of Service
$	americký dolár
DDoS	Distributed Denial of Service
DNS	Domain Name System
€	Euro
FBI	Federal Bureau of Investigation
FTP	File Transfer Protocol
GUI	Graphical User Interface
HW	hardware
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IT	Information Technology
IEEE	Institute of Electrical and Electronics Engineers
IPR	Intellectual Property Rights
IP	Internet Protocol
IC3	Internet Crime Complaint Center
£	britská libra
LDAP	Lightweight Directory Access Protocol
napr.	napríklad
NIS	Network Information System
NVD	National Vulnerability Database
OS	operčaný systém
OSSTMM	Open Source Security Testing Methodology Manual
OWASP	Open Web Application Security Project
PDCA	Plan, Do, Check, Act
SSL	Secure Sockets Layer
SQL	Structured Query Language
SW	software
tzv.	takzvaný
tj.	to je, to jest
ust.	ustanovenie
USA	United States of America

wifi	Bezdrôtová sieť
XSS	Cross-Site Scripting
XML	Extensible Markup Language
XXE	XML External Entities

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P 1: ŠABLÓNA ZADÁVACEJ DOKUMENTÁCIE

Šablóna - Zadávacia dokumentácie penetračného testovania			
Zadávateľ		**Vykonávateľ**	
Názov:		**Názov:**	
Adresa:		**Adresa:**	
Zodpovedná osoba:		**Zodpovedná osoba:**	
Kontakt:		**Kontakt:**	
Ďalšie informácie:		**Ďalšie informácie:**	
Popis projektu			
Cieľ penetračného testovania:			
Rozsah penetračného testovania:			
Časový harmonogram penetračného testovania:			
Typy a zameranie penetračného testovania:			
Použitý HW a SW:			
Ďalšie požiadavky:			
Zoznam príloh:			
Dátum:		**Podpis zadávateľa:**	
		Podpis vykonávateľa:	

PŘÍLOHA P 2: ŠABLÓNA REPORTU PENETRAČNÉHO TESTU

Šablóna – Report penetračného testovania			
Zadávateľ		**Vykonávateľ**	
Názov:		**Názov:**	
Adresa:		**Adresa:**	
Zodpovedná osoba:		**Zodpovedná osoba:**	
Kontakt:		**Kontakt:**	
Ďalšie informácie:		**Ďalšie informácie:**	
Špecifikácia penetračného testovania:			
Technický report:			
Odporúčané opatrenia:			
Zhodnotenie výsledkov:			
Dátum:		**Podpis zadávateľa:**	
		Podpis vykonávateľa:	