Česká zemědělská univerzita v Praze

Technická fakulta

Katedra technologických zařízení staveb

Diplomová práce

Penetrační testy webových aplikací

Bc. Jana Sladká

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Technická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jana Sladká

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Penetrační testy webových aplikací

Název anglicky

Penetration tests of web applications

Cíle práce

Cílem práce je shrnout problematiku penetračních testů a analýzy e-shopu ve vztahu k legislativním a normativním předpisům. Posouzena bude vhodnost nasazení konkrétního e-shopového systému. Výsledkem práce bude konkretizace a doporučení pro testovanou platformu.

Metodika

1. Úvod
2. Cíl práce
3. Metodika
4. Platformy pro realizaci e-shopů
5. Legislativní a normativní předpisy vztahující se k bezpečnosti webových aplikací
6. Penetrační testy
7. Výběr vhodné platformy pro testování
8. Implementace metodiky testování pro vybranou platformu
9. Diskuze výsledků a jejich hodnocení
10. Návrhy pro další zabezpečení
11. Zhodnocení a cenová kalkulace platformy splňující definovaná kritéria

Doporučený rozsah práce

50 – 60 stánek včetně obrázků a grafů

Klíčová slova

Penetrační testy, IT bezpečnost, webové aplikace, normy

Doporučené zdroje informací

DARIE, C. *AJAX a PHP : tvoříme interaktivní webové aplikace profesionálně*. Brno: Zoner Press, 2006. ISBN 80-86815-47-1.

HEROUT, P.: Testování pro programátory, 2016, České Budějovice: Kopp, ISBN 978-80-7232-481-1

Odpovídající normy a vyhlášky

POKORNÝ, J.: Hacking – praktický průvodce penetračním testováním, 2015, Brno: Zoner Press, ISBN: 978-80-7413-313-8

SELECKÝ, M.: Penetrační testy a exploitace, 2012, Brno: Computer Press, ISBN 978-80251-3752-9

SKLAR, D.: PHP 7 – Praktický průvodce nejrozšířenějším skriptovacím jazykem pro web, 2017, Brno: Zoner Press, ISBN: 978-80-7413-363-3

VRÁNA, J.: 1001 tipů a triků pro PHP, 2013, Brno:Computer Press, ISBN: 978-80-251-2940-1

Předběžný termín obhajoby

2020/2021 LS – TF

Vedoucí práce

Ing. Jan Lešetický, Ph.D.

Garantující pracoviště

Katedra technologických zařízení staveb

Konzultant

Ing. Martin Olmr

Elektronicky schváleno dne 3. 3. 2020

doc. Ing. Jan Malaťák, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 10. 2. 2021

doc. Ing. Jiří Mašek, Ph.D.

Děkan

V Praze dne 10. 05. 2021

Čestné prohlášení

„Prohlašuji, že jsem diplomovou práci na téma: Penetrační testy webových aplikací vypracovala samostatně a použila jen pramenů, které cituji a uvádím v seznamu použitých zdrojů. Jsem si vědoma, že odevzdáním diplomové práce souhlasím s jejím zveřejněním dle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů, ve znění pozdějších předpisů, a to i bez ohledu na výsledek její obhajoby. Jsem si vědoma, že moje diplomová práce bude uložena v elektronické podobě v univerzitní databázi a bude veřejně přístupná k nahlédnutí. Jsem si vědoma 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, ve znění pozdějších předpisů, především ustanovení § 35 odst. 3 tohoto zákona, tj. o užití tohoto díla.“

V Praze dne 14. 5. 2021

_____

Sladká Jana

Poděkování

Ráda bych touto cestou poděkovala svému vedoucímu práce panu Ing. Janu Lešetickému, Ph.D. za výběr tématu a možnost pracovat pod jeho vedením. Díky tomu jsem získala spoustu znalostí, které mohu uplatnit v praxi. Chtěla bych také poděkovat celé své rodině, která mi byla nejen při psané této práce, ale během celého studia mou největší oporou.

Penetrační testy webových aplikací

Abstrakt

Diplomová práce se zabývá penetračním testováním webových aplikací. Na úvod teoretické části jsou popsány možné platformy pro realizaci e-shopů a normativní a legislativní předpisy vztahující se k webovým aplikacím. Zbývající teoretická část popisuje problematiku penetračního testování. Jsou zde uvedeny používané techniky, metodiky a postupy penetračního testování. Na základě poznatků z teoretické části, byla navrhnuta metodika pro realizaci penetračního testování webových aplikací. Praktická část obsahuje popis uskutečněného testování, s jednotlivými fázemi od plánování až po report. Cílem práce je porovnání webových aplikací z hlediska bezpečnosti. V závěru jsou popsány výsledky a porovnání vybraných webových aplikací.

Klíčová slova: webová aplikace, e-shop, penetrační testování, OWASP

Penetration tests of web applications

Abstract

The diploma thesis deals with penetration testing of web applications. At the beginning of the theoretical part, are describe possible platforms for the implementation of e-shops and normative and legislative regulations related to web applications. The remaining theoretical part describes the issue of penetration testing. Here are presented used techniques, methodologies and procedures of penetration testing. Based on the knowledge from the theoretical part, a methodology for the implementation of penetration testing of web applications was proposed. The practical part contains a description of the performed testing, with individual phases from planning to report. The aim of the work is to compare web applications in terms of security. Finally, the results and comparison of selected web applications are described.

Keywords: web application, e-shop, penetration testing, OWASP

Obsah

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek

MySQL	– My Structured Query Language
FTP	– File Transport Protocol
PHP	– Hypertext Preprocessor
HTTP	– HyperText Transfer Protocol
HTTPS	– HyperText Transfer Protocol Secure
SEO	– Search Engine Optimization
XML	– eXtensible Markup Language
VPN	– Virtual Private Network
Wi-Fi	– Wireles Fidelity
B2B	– Business to Business
B2C	– Business to Client
IT	– Informační Technologie
ISMS	– Information Security Management System
SMART	– Specific, Measurable, Achievable, Realistic, Time Specific
PCDA	– Plan Do Check Act
GDPR	– General Data Protection Regulation
IP	– Ingress Protection
HTML	– Hyper Text Markup Language
JS	– JavaScript
OWASP	– Open Web Application Security Project
SQL	– My Structured Query Language
OS	– Operační Systém
LDAP	– Lightweight Directory Access Protocol
API	– Appliaction Programming Interface
DoS	– Disk operating System
XXE	– XML eXternal Entity
SSL	– Secure Socket Layer
XSS	– Cross-Site Scripting
IDS	– Intrusion Detection System
IPS	– Intrusion Prevention Systems
DNS	– Domain Name System

SW – Software

HW – Hardware

1 Úvod

Otázka bezpečnosti webových aplikací a sítí je v dnešní době velmi naléhavá. Rychlý technologický pokrok s sebou přináší nejen výhody, ale také možná rizika pro všechny uživatele. Každý z nás denně používá mobilní telefon nebo počítač. Poměrně často je lehké podlehnout pocitu falešného bezpečí, a tak se volí jednoduchá hesla, nepoužívají se antivirové programy, stahují se volně dostupné aplikace nebo se nakupuje na neověřených e-shopech. Toto téma od roku 2020 nabírá na obrátkách. Dotýká se každé jednotlivé osoby, protože vzhledem ke koronavirové krizi je možné nakupovat pouze online. Právě zde zákazníci zadávají své osobní údaje a údaje z platebních karet. Každá taková činnost znamená potenciál pro tzv. kyberkriminalitu. Důsledkem toho může dojít k závažným trestním činům jako je krádež finančních prostředků z účtu nebo krádež identity.

Rozhodne-li se někdo pro e-shop, pak si může vybrat platformu, na které bude podnikání realizovat. K dispozici je řešení typu open-source, komerční nebo na míru. Každá platforma má své výhody a také jistá úskalí. U komerčního řešení a řešení na míru by měl za bezpečnost nést zodpovědnost majitel platformy. Předpokládá se, že u těchto řešení je bezpečnost v pořádku. V případě open-source je za to zodpovědný provozovatel webové aplikace.

V online obchodování nedostatečná bezpečnost používaných systémů může znamenat únik citlivých informací o společnosti, získání know-how nebo informace o zákaznících. O GDPR pojednává zákon 110/2019 Sb., ten obsahuje několik odkazů na individuální ochranu. Před jakýmkoli zpracováním osobních údajů musí být subjekt informován mimo jiné o účelech, pro které budou údaje zpracovány, o totožnosti správce údajů, o příjemcích jeho osobních údajů a také o době uchování údajů.

Nedostatečně zabezpečené webové aplikace jsou lehkým terčem hackerů, kteří mohou mít nekalé úmysly. Ať už se jedná o poškození provozovatele aplikace nebo osobní obohacení. K bezpečnosti webový aplikací se vztahuje norma ISO/IEC 27001, která specifikuje rámec systému řízení bezpečnosti informací. Ústředním bodem této normy je požadavek na plánování, implementaci, provoz, monitorování a zlepšování bezpečnosti. ISMS se zavádí dle normy ISO/IEC 27002 a certifikace se uděluje podle normy ISO/IEC 27001.

Jakékoliv incidenty znamenají velké riziko pro konkurenceschopnost a důvěru uživatelů. Proto se na bezpečnostní politiku klade čím dál větší důraz v organizacích. Neoddělitelnou součástí opatření v rámci bezpečnostní politiky je i pravidelný bezpečnostní audit, který vede k odhalení případných zranitelných míst software. Ve spolupráci s testery jsou realizované penetrační testy, které simulují možné chování hackerů při útoku na systém. Pro penetrační testování je možné využít přístupy jako white box, black box nebo gray box. Rozhodují zde úroveň a rozsah informací, které jsou testerovi poskytnuty. Testované prostředí může být interní nebo externí podle toho, jestli se útočník snaží proniknout z vnitřního nebo vnějšího prostředí. Penetrační testování lze provést manuálně, automaticky nebo semiautomatizovaně. Na zlepšení zabezpečení software pracuje nezisková organizace OWASP, která pravidelně vydává top 10 nejčastějších chyb zabezpečení webových aplikací. Dle toho bude realizována praktická část.

2 Cíl práce

Cílem diplomové práce je definovat problematiku penetračního testování a možnosti testování webových aplikací. Prvním dílčím cíl je teoreticky popsat problematiku penetračního testování včetně prostředí, provedení nebo možných přístupů testerů a vybrat si vhodnou metodiku pro testování vybraných webových aplikací. Vhodnou metodikou pro penetrační testování bude nezisková organizace OWASP, která nabízí nástroj pro testování webových aplikací.

Druhým dílčím cílem bude otestovat tímto nástrojem 2 vybrané platformy pro realizaci e-shopů a výsledky porovnat. Z výsledků by mělo být jasně patrné, jakou platformu si pro realizaci e-shopu zvolit a jak vysoce zabezpečené tyto platformy jsou. Ve výsledcích by nemělo chybět ani doporučení na další zabezpečení.

3 Metodika

Teoretická část diplomové práce je zpracovaná pomocí deskriptivních metod založených na studiu dostupné a odborné literatury. Dochází zde k základnímu rozdělení platforem pro realizaci e-shopu a detailnímu popisu jejich zástupců. K bezpečnosti webových aplikací se vztahují normy ISO/IEC 27001, ISO/IEC 27002 a zákon z Evropské Unie o ochraně osobních údajů neboli GDPR.

Blíže specifikované jsou především penetrační testy webových aplikaci, kde jsou detailně popsány možná prostředí, provedení, typy testů nebo přístupy testerů. Významnou metodikou penetračního testování je nezisková organizace OWASP, která pravidelně vydává top 10 chyb zabezpečení webových aplikací. Tato metodika byla použita pro praktickou část, kde dochází k penetračnímu testování komerční platformy a open-source platformy.

Průběh testování se rozvrhuje do 4 fází – plánování, shromažďování dat, exploitace a skenování a report. Všechny tyto fáze bylo nutné vykonávat shodně pro obě platformy. Nalezená rizika jsou pak rozepsána pro každou platformu zvlášť. V závěru práce dochází ke komparaci výsledku z praktické části a návrhu na zlepšení bezpečnosti

4 Platformy pro realizaci e-shopu

Online obchodování je moderní a stále vyhledávanější forma obchodování. Oblíbenost je získaná zejména pro jednoduchost nákupu, trvalou dostupnost a možnost nákupu z pohodlí domova. Obchodování na internetu se také rozděluje na B2B (Business to Business) nebo B2C (Business to Client). B2C – jedná se o prodej produktů a služeb od obchodníka ke konečnému spotřebiteli. Jinak řečeno jde o maloobchodní prodej. B2B – Je to způsob prodeje produktů a služeb mezi podnikateli. Nejedná se o konečné spotřebitele, ale slouží k dalšímu prodeji. Obchod je založený na určitých smlouvách / dohodách o odběru mezi jednotlivými obchodníky.

Základní výhody pro provozovatele e-shopů jsou:

- Výrazně nižší investice oproti klasickým kamenným prodejnám.
- Funkčnost obchodu je 24 hodin, 7 dní v týdnu.
- Produkty na internetu se nabízí i těm zákazníkům, které by v kamenné prodejně mnoho neupoutalo. Je snadné v dnešní době oslovit potencionální klienty z jakékoliv části republiky / světa.

4.1 Komerční e-shopové řešení

Komerční řešení neboli pronájem e-shopu je vhodný pro malé společnosti nebo ty, který teprve na internetu začínají podnikat a nemají s tím žádné zkušenosti. K založení není potřeba znalost IT úkonů, databáze či FTP. Pronájem se platí zpravila měsíčně, max. ročně, kdy pro roční platbu bývá výhodnější částka.

Většina e-shopových řešení nabízející pronájem mají testovací verze (dle společnosti od 15 do 30 dnů), na kterých je možné si vyzkoušet základní funkčnost. Za tu dobu je možné si vyzkoušet veškeré nastavení jak z pohledu zákazníka, tak z pohledu administrátora.

Každá platforma má podnikání založené jinak. Některá nabízí veškeré funkce v e-shopu v rámci pronájmu. Jiná nabízí pouze základní funkce a další se musí zakoupit za měsíční poplatek nebo jednorázový odkup.

Mezi nejznámější komerční řešení nabízející pronájem se řadí shoptet, eshop-rychle.cz, oxyshop nebo shop5. [4]

Výhody

- Možnost okamžitě začít podnikat na internetu.
- Není potřeba vlastní server.
- Zákaznická a technická podpora (u většiny platforem je zdarma).
- Základní grafická šablona v ceně (u většiny platforem).

Nevýhody

- Není možnost FTP přístupu.
- Limit pro počet položek a kategorií v e-shopu.
- Není možnost individuálních úprav (pouze u některých řešení).
- Možnost výběru grafiky z připravených šablon, bez možnosti úprav (pouze u některých řešení).[1][2]

4.1.1 Shoptet

Platforma shoptet je na tuzemském trhu již od roku 2009. Zakladatelem je Miroslav Uďan, který je dodnes v čele shoptetu. Neustálým přibýváním e-shopů roste i tato platforma, která se řadí mezi jedničku v České republice. Po 7 letech od jejich prvního spuštění se stali jedničkou trhu, kdy překonali hranici 9900 e-shopů. V roce 2019 překonali hranici 19 000 internetových obchodů. Zhruba polovina českých e-shopů využívá platformu Shoptet.

Velkou výhodou je responzivní administrace e-shopu a mobilní aplikace, která je však už za poplatek. Tímto je možné přehledně a jednoduše přidávat zboží, obsluhovat objednávky a další funkce z chytrého telefonu. Avšak práci na počítači to zcela nenahradí a taková obsluha by měla být pouze výjimečná.

Podpora pro zákazníky

Nevynikají pouze nejvyšším počtem e-shopu, ale mají i nejlépe zpracovanou podporu. Na webových stránkách shoptet.cz v sekci podpora mají hned 10 odkazů pro začínající, ale i pokročilé uživatele.

Významným pomocníkem je shoptet školka, kde je krok po kroku vysvětlené, na co by se nemělo při přípravě e-shopu zapomenout. V článcích se zákazník dočte, jaká je vhodná základní literatura (knížky, články pro začínající online podnikatele), příprava produktů, vhodná propagace a vylepšení základních funkcí. V administraci platformy s tím perfektně poradí průvodce „Shoptetrix“, který pomůže nejen s nejdůležitějším nastavením, ale i přidá užitečné rady do začátku.

V pozadí však nesmí zůstat shoptet blog, který přináší články ze světa e-shopové scény. Najdou se zde novinky z e-commerce, rozhovory s velmi úspěšnými marketingovými specialisty a tipy, díky kterým nezůstane žádný e-shop pozadu. Rady a tipy jsou v úspěšnému podnikání potřeba, ale nesmí se zapomenout na inspiraci a motivaci. I na takové téma je platforma připravená. Články v blogu jsou založené na skutečných příbězích úspěšných, českých a předních podnikatelů. [14][15]

Tarify a ceník

U shoptetu je možné si vybrat z 5 různých tarifů. Od tarifu zdarma až po ENTERPRISE. Verze jsou různě omezené a záleží o jaký sortiment se jedná. Základní omezení nastává u počtu produktů, doplňků a e-mailů. Platba za tarif je možná měsíční, roční nebo na 2 roky, kdy samozřejmě je výhodnější si předplatit delší období.

Jednotlivé tarify se mění v závislosti na množství produktů, počtem e-mailových schránek nebo doplňky. Platí zde základní rovnice a to, že čím více toho zákazník bude potřebovat, tím vyšší bude cena za pronájem e-shopu. Ke každému zakoupenému tarifu nabízejí 1 000 Kč pro Google Ads a 2 000 Kč na Sklik. Tyto nástroje se používají na propagaci e-shopu. [27]

Funkce a zakázkové práce

Funkce v e-shopu jsou dostupné dle jednotlivých verzí a záleží tedy na výběru. Shoptet nabízí obrovské množství doplňků a je možné si tyto doplňky do e-shopu vybrat v rámci tarifu nebo si jej v průběhu kdykoliv dokoupit. V nabídce nyní nabízejí více než 190 doplňků ze všech oblastí:

a) Produkty – slouží pro lepší prezentaci produktů a zajistí lepší prodejnost (související, parametrické filtrování, varianty, hodnocení atd.)
b) Dopravy – usnadňují zejména expedici objednávek a zjednodušují práci s odesíláním objednávek (export pro PPL a Českou poštu, balíkobot, zásilkovna, uloženka atd.)
c) Platby – umožňují co nejjednodušší platební systémy (Comgate, Gopay, Essox, Pays, PayPal atd.)
d) Marketing – nástroje, které zvyšují prodejnost produktů a otevírá nové cesty zákazníkům (Glami, Google, pokročilé SEO, Aukro, Colabim atd.)

e) Vzhled – odlišuje e-shopy od konkurence a zajišťuje originalitu vzhledu (šablony, bannery atd.)
f) Zákazníci – umožňuje skvělou komunikaci se zákazníky, aby byl zajištěný perfektní servis (Smartsupp, Mailchimp, Retino atd.)
g) Účetnictví – usnadňují práci s účetními systémy (Profit, Vario Konektor, Money S3 atd.)
h) Reporting – zajišťují viditelnost e-shopu ve vyhledávačích (SEOwebmarter, Statistiky, MonkeyData statistiky atd.)
i) Ostatní – doplňující funkce (Foxentry, Lemonero atd.)

Výše zmíněné doplňky jsou už připravené a kdykoliv se mohou do e-shopu spustit. Shoptet nabízí web, kde si každý může vybrat svého specialistu, který mu může poradit v daném odvětví. Najdou se zde specialisté od úpravy vzhledu, přes marketing, SEO, účetnictví, právo až po programování na míru. Je možné si vybrat ke spolupráci specializovanou firmu nebo fyzickou osobu. Ceny se odvíjí od jednotlivých profesí a základní hodinovou sazbu si stavuje každý sám. [24]

4.2 Open-source řešení

Open-source je software, kde mají uživatelé přístup ke zdrojovému kódu. Tím se zejména odlišuje od komerčního řešení. Většina software s otevřeným kódem je zdarma distribuován, ale s omezenými pravidly, jak jej používat. Jedná se o prototyp decentralizovaného, samo organizujícího se procesu. Neexistuje žádné centrální plánování nebo řízení.

Software je chráněn autorskými právy a distribuován s licenčními podmínkami navrženými tak, aby zdrojový kód byl vždycky k dispozici. Podmínky jsou však zcela rozdílné oproti komerčnímu využití. Uvádí se tam zejména jakým způsobem se software nakládat, upravovat a distribuovat. Licence s otevřeným zdrojovým kódem všeobecně udělují uživatelům oprávnění používat software pro jakýkoliv účel. Některé open-source organizace (nazývané „copyleft“) stanovují, že každý, kdo uvede upravený program s otevřeným zdrojovým kódem, musí také zveřejnit celý zdrojový kód. Kromě toho si také určité organizace stanovují, že každý, kdo změní a sdílí program s ostatními si nesmí účtovat jakýkoliv poplatek.

Zdrojový kód je část software, kterou většina uživatelů počítačů nikdy nevidí. Je to kód, se kterým mohou počítačoví programátoři manipulovat a upravovat funkčnost aplikace. Programátoři, kteří mají ke kódu přístup, většinou program doplní o vlastní funkce nebo jen doplní správnost syntaxe, která nemusí vždy fungovat správně.

Mezi nejznámější open-source aplikace se řadí WordPress. Tento systém byl vytvořený jako redakční systém. Stal se natolik oblíbeným, že byl doplněn o plugin WooCommerce a tím došlo k rozšíření základního redakčního systému o e-commerce funkce. Celý WordPress je napsaný v čistém PHP jazyce a MySQL.

České open-source řešení nabízí firma Shopsys. Jedná se o kvalitní zpracování open-source e-shopového řešení, které vzniklo právě pro tvorbu e-shopu. Oproti WordPressu je Shopsys řešení postavené na Symfony.

Výhody

- Nulová pořizovací cena systému v základní formě.
- Aktualizace systému.
- Možnost vlastních úprav a vývoje.
- Další uživatelé využívají stejný systém (vhodné při řešení problémů).

Nevýhody

- Vyšší riziko napadnutí hackerem
- Není zde podpora.
- Může být finančně náročnější (koupě pluginů, funkce navíc atd.) [3]

4.2.1 Wordpress

Redakční publikační open-source systém vytvořený společností Automattic. V současné době patří mezi nejpoužívanější systém, díky němu lze vytvořit webovou stránku, e-shop nebo jen blog bez znalosti programování a kódování. Systém vyvíjený pod licencí GNU GPL a je napsaný v PHP a veškerá data ukládá do MySQL databáze, případně MariaDB.

Platforma původně vznikla jako redakční systém, avšak se časem rozšířila o řadu dalších funkcí. Jednoduše lze vytvořit základní internetovou prezentaci, blogovací systém, fórum nebo e-shop.

Obsah, vzhled i funkce je možné měnit v administračním rozhraní Wordpressu. Aktuálně nejnovější verze je 5.5 kde se nachází více než 50 různých jazyků. Pro tyto jazyky je systém kompletně přeložen. Celkově platforma nabízí přes více než 170 jazyků, ale veškeré moduly nejsou přeložené.

Mezi nejznámější weby postavené na této platformě se řadí CNN, Mercedes Benz nebo webové stránky společnosti Walt Disney. Z českých webů na této platformě nalezneme obuvnickou společnost Baťa.

Každá platforma má své základní požadavky na správný chod. U Wordpressu při verzi 5.5 jsou následující požadavky:

- PHP verze 7.4 a vyšší
- MySQL verze 5.6 nebo novější, případně MariaDB verze 10.1 nebo novější
- Zabezpečené připojení HTTPS

Funkce

Přes více než 38 % všech webů na internetu je postaveno na platformě Wordpress. Každým dnem přibývají další. Kombinace jednoduchosti pro uživatele a možnosti si jednotlivé funkce upravit nebo doprogramovat, dává Wordpress flexibilitu a snadnou ovladatelnost všem svým zákazníkům.

K dispozici je více něž tisíc pluginů, díky kterým je možné web přizpůsobit svým požadavkům. Tyto pluginy dávají tzv. neomezené množství výsledných funkcí. Pokud by bylo potřeba, tak každou jednotlivou funkci je možné rozšířit, změnit nebo odebrat. [16][17]

4.2.2 Prestashop

Jedná se o platformu e-shopu typu open-source, která nabízí tisíce dostupných doplňků a funguje pod licencí General Public Licence. V současnosti eviduje více než 300 000 uživatelů. Platforma Prestashop je naprogramovaná v programovacím jazyce Symphony a obsahuje více než 600 funkcí pro správu internetového obchodu. Jelikož se jedná o open-source, tak kdokoliv může číst, upravovat a dále nabízet software.

Projekt vznikl v roce 2007 v Paříži a už v roce 2010 byl vyhlášen jako vítěz v Packt 2010 jako nejlepší open-source software pro tvorbu e-commerce aplikací. V roce 2011 platforma zvítězila ve stejné kategorii na Open Source Awards.

Platforma PrestaShop je nabízena ve více než 40 jazycích, ale je plně přeložena je pouze do anglického a francouzského jazyka. V roce 2019 veškerý součet e-shopů postavených na této platformě dosahoval cca 300 000 ve 190 zemích světa. Celkový obrat z těchto e-shopů byl více než 17 miliard EUR.

Aby se platforma zprovoznila a fungovala správně, musí být zajištěný hosting s těmito základními požadavkami:

- Apache na webovém serveru musí být min. verze 2.2 a vyšší
- Verze PHP min. 7.4 a vyšší
- Verze MySQL min. 5.6 a vyšší

Funkce

I když platforma PrestaShop není rozšířená natolik jako Wordpress, tak v základní verzi nabízí o mnoho více funkcí. K dispozici je pak více než 10 000 modulů, které je možné si stáhnout přímo z oficiálních zdrojů. Jsou tam moduly jak ve verzi zdarma, tak placené. Tyto moduly může naprogramovat kdokoliv, pak jen musí projít testováním, které zabere několik dnů. Společnost pak může tento modul přidat a programátor z něj může profitovat. O tyto moduly je možné e-shop kdykoliv rozšířit a vylepšit. V základní verzi platforma obsahuje:

- Různé platební metody (zejména platební karty)
- Možnost dopravy včetně poplatků (hmotnost, balení)
- Možnost vytvoření z e-shopu katalog
- Statistiky (návštěvnost, prodej – obrat, zisk), SEO
- Zasílání newsletterů
- Marketingové nástroje (slevové kupóny, věrnostní slevy, partnerský program)
- Základní nebo rozšířená analytika v administraci

Stejným způsobem je to i se šablonami, které jsou také dostupné na oficiálním webu. Podle náročnosti je možné si vybrat základní, složitější nebo robustní šablonu. Možné je samozřejmě si šablonu naprogramovat, dle svých vlastních přestav a tu pak je možné nabídnout ostatním na oficiálních stránkách. [18]

4.3 E-shop na míru

Vytvoření e-shopu na míru je vhodné pro podniky, které mají nějakou zkušenost s obchodováním na internetu, dlouhodobou vizi a zejména finanční prostředky. Takové řešení je totiž poměrně časově náročné. Jde o řešení, které se snaží maximálně přizpůsobit požadavkům klienta a zároveň implementovat vše, co zákazník ke svému podnikání potřebuje. Jedná se o úzkou spolupráci mezi dodavatelem řešení a zákazníkem. Dodavatelem je ve většině řešení tým odborníků, který se skládá z programátora, grafika a marketingového specialisty.

Realizace tohoto řešení je ze všech dostupných řešení nejdražší, protože realizaci e-shopu předchází nespočet různých aktivit. Značný rozdíl je mezi přesným zadáním pro realizaci a poptávkou. Zatím co poptávka přichází od zákazníka, tak zadání má již na starosti dodavatel. Tým odborníků specializující se na tvorbu e-shopu má za úkol zpracovat zadání. To by mělo být vypracované na základě dat od zákazníka, funkcionality, obsahu webu a také veškerý postup realizace s přesným popisem práce pro vývojářský tým. Konečná cena může být v případě zadání již odhadnutelná, ale v případě poptávky se jedná pouze o velmi orientačn. Cenová relace za e-shop je u každého developera rozdílná, protože každý využívá jiné techniky a postupy.

Výhody

- Unikátní grafika a vzhled e-shopu.
- Funkce naprogramované dle představ.
- Podpora a správa od dodavatele.

Nevýhody

- Vysoké pořizovací náklady.
- Delší doba dodání, vzhledem k zadání. [1][4]

4.3.1 Shop5

Systém je postavený na čistém PHP a MySQL. Tvůrcem je Bc. Martin Březovský, který je stále jediným programátorem. Největší předností této platformy je, že dokážou e-shop přizpůsobit zákazníkovi na míru.

Platforma nabízí nejen e-shop na míru, ale i pronájem nebo odkoupení licence. Nejedná se o open-source aplikaci a není tedy možné software si upravovat nebo jej dále šířit. Momentálně na této platformě běží cca 800 e-shopů. Síla tkví v tom, že funkce e-shopu nejsou nijak omezené podle tarifu. Nevýhodou této platformy může být neresponzivní administrace. I když spravovat e-shop z telefonu není úplně vhodné, tak je spousta zákazníků, kteří právě toto vyžadují.

Podpora pro zákazníky

Oporou pro zákazníky shop5 je web specializovaný pouze jako podpora pro zákazníky. Na tento web dostane zákazník odkaz po založení e-mailu a také se tam vždy dostane z administrace e-shopu. Zákazník tam nalezne většinu funkcí s podrobným popisem. Aby to pro zákazníka bylo snazší, tak u každé sekce nalezne otazník, který vysvětluje, k čemu tato sekce slouží a jaké jsou její možnosti. Stejné je to i funkcí, které si zákazník nastavuje.

Za zmínku stojí i youtube kanál Betulasoft s.r.o., kde je spoustu užitečných návodu na nastavení e-shopu v systému shop5. Najdou se tam základní návody na přidání kategorií, produktů, základní úpravy až po různé vychytávky.

Jako všechny platformy tak i tato nabízí testovací verzi po dobu 30 dní zdarma. Funkce ani na testovací verzi nejsou nijak omezené a je tedy možné si celkovou funkčnost vyzkoušet. [25]

Tarify a ceník

Platforma nabízí nejen pronájem e-shopu, ale i jeho odkoupení. Pronájem e-shopu je možné od verze MICRO až po ULTIMATE. Pronájem se vybírá podle počtu kategorií a produktů v e-shopu. Jiné omezení zde není. Všechny pronájmy je možné platit 3 způsoby, a to buď čtvrtročně, půlročně nebo ročně. Kdy roční platba je vždy ta nejvýhodnější.

Odkup e-shopu je možný hned ve 2 verzích, kde verze START je základní verze e-shopu bez dalších zakázkových úprav. K odkoupení e-shopu je zapotřebí mít zakoupenou doménu a hosting. Velikost serveru na hostingu se odvíjí zejména podle počtu produktů, jazykových verzí a přístupů za den. Vždy je tedy vhodné probrat hosting s podporou shop5. V rámci jakéhokoliv odkupu e-shopu jsou aktualizace e-shopu na 2 roky. Druhá nabízená verze odkupu e-shop PROFI. V této verzi je zakázková grafika, multijazyčná verze a několik hodin práce programátora. [26]

5 Legislativa a normy vztahující se k bezpečnosti webových aplikací

Bezpečnost v informatice se dnes nezaměřuje pouze na veřejný sektor, ale i na soukromý sektor ve smyslu chránit data a jejich informace. V minulosti byly veškeré informace uchovávané v papírové podobě, z toho důvodu nebylo potřeba dbát na zvýšenou informační bezpečnost. Proto byly vytvořené bezpečnostní kybernetické standardy, umožňující firmám vykonávat činnosti s cílem minimalizovat množství útoků směřovaných na sítě počítačů. Tyto normy vznikly, protože bylo potřeba v bezpečí uchovávat ukládaná data. Se vzrůstající závislostí na SMART technologiích dochází k nárustů hackerských útoků a hrozeb. Vyšší moc tak čím dál častěji vyžaduje implementaci těchto legislativ a norem v systémech. Je zcela nemožné eliminovat veškeré hrozby. Zlepšení zabezpečení využitím standardů přispívá k zajištění ochrany diskrétních informací a k řízení rizik.

5.1 ISO/IEC 27001 a 27002

V roce 2005 byl uveřejněn standard ISO/ IEC 27001 specifikující rámec systému řízení bezpečnosti informací (ISMS). Popisuje požadavky na zlepšování a udržování systému řízení rizik, bezpečnosti informací a samotnou implementaci. Norma je zaměřena na společnosti ze všech odvětví a všech velikostí. Jejím cílem je v organizacích chránit informační aktiva, aby nedošlo k úniku citlivých informací. Od uvedení normy se zcela zásadně změnily technologie ve všech sférách. Došlo k velkému nárustů zejména v mobilních technologiích a přibyly kybernetické hrozby. To byl jeden z podnětů pro organizaci ISO tuto normu modernizovat a vznikla norma ISO/IEC 27001:2013.

I když došlo k revitalizaci norem, tak se zachovalo pravidlo, že podle normy ISO/IEC 27001 se uděluje certifikace a ISMS se zavádí dle normy ISO/IEC 27002. Získáním certifikátu organizace prokazují schopnost aplikovat bezpečnostní opatření, které se vyznačuje dostatečným zabezpečením informací a ochranou informačních aktiv.

Ústředním bodem ISO/IEC 27001 je požadavek na plánování, implementaci, provoz, monitorování a zlepšování procesně orientovaného ISMS. Přístup by měl být sladěn s cyklem PDCA (Plan, DO, Check, Act), viz obrázek 1.

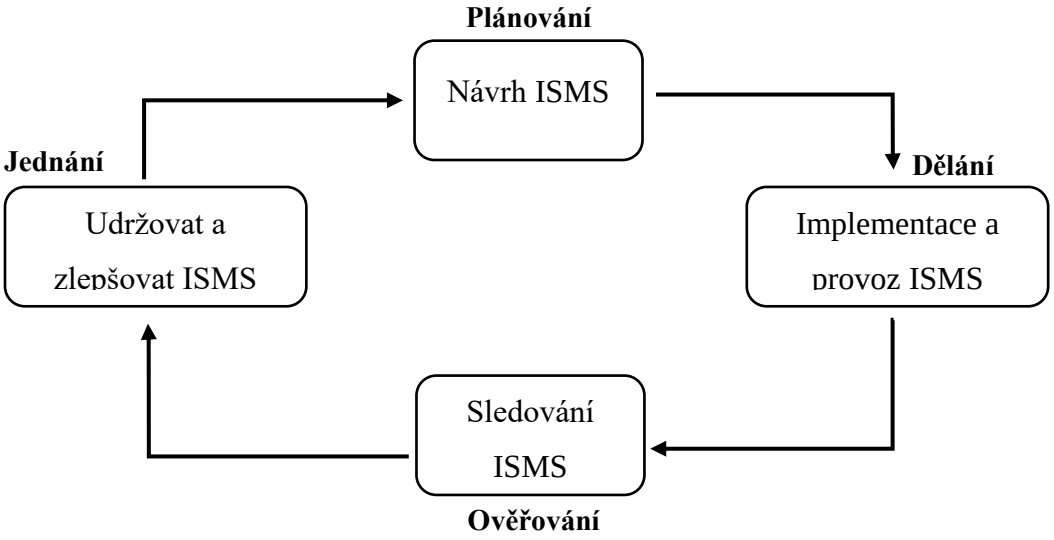
1) Plánování (Plan) – identifikování informačních aktiv a s mini spojené bezpečnostní požadavky

2) Dělání (Do) – provoz a implementace ISMS

3) Ověřování (Check) – posuzuje výkonnost

4) Jednání (Act) – provádí změny pro dosažení zlepšení [9][10]

Obrázek 1 Cyklus PDCA



Zdroj: vlastní zpracování na základě [9]

5.2 GDPR

Obecné nařízení o ochraně osobních údajů (GDPR) je základní zákon Evropské Unie o ochraně údajů a obsahuje několik odkazů na individuální ochranu. Před jakýmkoli zpracováním osobních údajů musí být subjekt informován mimo jiné o účelech, pro které budou údaje zpracovány, o totožnosti správce údajů, o příjemcích jeho osobních údajů a v neposlední řadě o době uchování údajů.

Osobní údaje jsou jedním z klíčových pojmů zákona o ochraně osobních údajů. Osobními údaji se rozumí všechny informace vztahující se k identifikované nebo identifikovatelné osobě. Za identifikovatelnou osobou se považuje fyzická osoba, kterou je možné přímo nebo nepřímo identifikovat, zejména odkazem na identifikátor. Tím může být identifikační číslo, jméno, on-line identifikátor, lokalizační údaje, dále pak jeden nebo více faktorů charakteristických pro genetickou, fyzickou, fyziologickou, ekonomickou, duševní, sociální nebo kulturní identitu dané fyzické osoby.

V případě anonymních údajů se pak jedná o antonymum k osobním údajům a týkají se informací, které se netýkají identifikované nebo identifikovatelné osoby. Subjekt údajů již není možné identifikovat. Zpracování anonymních údajů nepodléhá uplatňování zákona o ochraně osobních údajů. Pseudonymní údaje, tj. osobní údaje po pseudonymizaci, jsou stále informace týkající se identifikované osoby, a tedy podléhají zákonu o ochraně osobních údajů.

Článek 4(5) GDPR definuje pseudonymizaci jako zpracování osobních údajů takovým způsobem, že je již nelze připisovat konkrétnímu subjektu údajů bez použití dodatečných informací. S technickými a organizačními opatřeními, která zajistí, že nebudou připisovány identifikované fyzické osobě. Zákon definuje pseudonymizaci jako akt zpracování, nikoli jako kategorii osobních údajů. Odkazuje se na proces, který snižuje riziko přímé identifikace, ale nevytváří anonymní údaje. Pseudonymizace je označována jako prostředek ke snížení rizik pro subjekty údajů, a jako vhodná záruka pro veškeré osobní údaje používané pro vědecký, statistický nebo historický výzkum.

Správcem může být fyzická nebo právnická osoba, v některých případech i orgán správní moci (musí však zpracovávat osobní údaje). Hlavní náplní správce je odpovědnost za dodržení povinností kladených obecním zákonným nařízením. Mezi zákonné nařízení se řadí zpracování GDPR. Každý správce musí doložit důvod zpracování osobních údajů a zabezpečit osobní údaje natolik, aby nedošlo k jejich odcizení (následnému zneužití). GDPR neupřesňuje, jak by měli správci údajů plnit svou oznamovací povinnost. V souvislosti s webovou aplikací jsou informace obvykle poskytovány ve všeobecných podmínkách nebo v zásadách ochrany osobních údajů na stránkách poskytovatele.

Zpracovatel je fyzická nebo právnická osoba, agentura, orgán veřejné moci nebo jiný subjekt, který zpracovává pro správce osobní údaje. Mezi správcem a zpracovatelem musí být uzavřená smlouva, která určuje zpracovateli, za jakým účelem, jaké osobní údaje a na jak dlouhou dobu bude činnost vykonávat. Ve smlouvě také nesmí chybět jaké jsou povinnosti a práva správce.

Osobní údaje jsou zpracovávané pro konkrétní účely a jakmile dojde k vykonání tohoto účelu, tak poskytnuté informace již nebudou potřeba. Zákon to vymezuje jako dobu nezbytně nutnou. Až dojde k dokončení účelů, tak je povinnost údaje bezpečně zlikvidovat. Dle GDPR je nutné zabezpečit osobní údaje technicky a organizačně, aby nedošlo k jejich úniku, přepsání nebo zneužití. [11][12][13]

6 Penetrační testy

Webové aplikace poskytují svoje služby po síti a uživatel používá jako klienta webový prohlížeč. Penetrační testování slouží k funkčnímu prozkoumání sítí a počítačového systému s cílem nalézt slabá místa a odstranit je. Pokud se slabá místa povede odstranit, dojde ke zvýšení bezpečnosti daného informačního systému. Předmětem penetračního testování je detailní prozkoumání systému, externího prostředí a odhalení potencionálně slabých míst v informačním systému. Zkoušejí využít slabá místa s následným napadením a získáním přístupů do částí, kam daný uživatel nemá oprávnění přístupu. Představují tedy simulaci reálných útoků na systémy, přičemž při testování dochází k odhalení slabých míst a jejich zneužití. Podobně k tomu dochází při reálném útoku. Tyto testy se většinou provádějí při závěrečných kontrolách informačního systému. Důležité je stanovit si dosažitelný cíl před zahájením celého testování. Takovým dosažitelným cílem by mohlo být riziko, že hacker naruší zabezpečení určitých klíčových aktiv. Celý proces testování by se neměl podceňovat. Synonymem pro penetrační testování je například „white hat hackování“ nebo „etické hackování“. Antonymem penetračního testování je „black hat hackování“ nebo pouze „hacker“.

Etický hacker se při své práci snaží napodobit činnost opravdového hackera, čímž dokáže nejlépe imitovat reálný útok. I když etický hacker využívá stejné nástroje jako škodlivý hacker, tak jsou mezi nimi patřičně velké rozdíly, které je třeba si uvědomit. Základním rozdílem těchto dvou typů hackerů je úmysl, motivace a autorizace.

Hlavním a nejjednodušším znakem, kterým můžeme odlišit škodlivého hackera od etického je autorizace. Znamená to získání potřebných povolení k provedení jakýchkoliv činností pro potřeby provedení penetračních testů. Souhlas s testováním vždy vydává provozovatel daného informačního systému. Etický hacker v rámci autorizace musí mít jasně vymezený rozsah prováděných činností (včetně cílů), který je schválený oboustranně. I v případě, že by mohlo dojít k narušení systému během testování, které však není úmyslné. Škodlivý hacker jedná zásadně bez vědomí provozovatele.

Motivace etického hackera je pomoct dané společnosti, která provozuje informační systém, snížit riziko napadení škodlivým hackerem a pomáhá zvyšovat úroveň zabezpečení. Největší motivací škodlivého hackera je vidina velkého finančního zisku, který se snaží získat vyděračským chováním od provozovatele napadeného informačního systému. Avšak není pravidlo, že hackerovi jde pouze o finance. Škodlivý hacker může mít motivaci:

a) politickou – zahrnují ničení, narušování nebo převzetí kontroly nad cíli, špionáž, vytváření politických prohlášení, protesty nebo odvety
b) ekonomickou – zahrnují krádež duševního zdraví nebo ekonomicky cenných aktiv (např. informace o kreditních kartách), podvody nebo průmyslovou špionáž
c) sociokulturní – zahrnují útoky s filozofickými, teologickými, politickými nebo humanitárními cíli. Patří sem i touha a zvědavost po publicitě.

6.1 Typy testů

Vhodně zvolený typ testu k penetračnímu testování přinese nejvíce potřebných informací pro další zabezpečení webové aplikace Před výběrem správného přístupu je zapotřebí provést analýzu informačního systému, protože každý z přístupů má určité výhody a nevýhody. Testování podléhá vše, kde může hrozit cizí proniknutí do systému, zpronevěra dat nebo způsobení škody (podnikatelská aktivita). Jedná se například o:

- Veřejně dostupné webové stránky
- Interní informace o firemních klientech a zaměstnancích
- E-mailové schránky a servery
- Přístupové údaje (hesla)
- FTP servery a datová uložiště
- Informační systémy a softwarové aplikace

Nejčastější prováděné testy

- Síťové penetrační testování

  Tento typ testu se využívá pro testování infrastruktury počítačové sítě, jestli neobsahuje potencionální bezpečnostní hrozby či zranitelnosti. U testování počítačové sítě je možné rozlišovat interní a externí testování.

  - Interní – tester dostává přístup přímo do firemní sítě skrz VPN připojení, nebo je možné testovat na místě určení, kde se bude tester připojovat do vnitřní sítě dané společnosti přes Wi-Fi.
  - Externí – pro testování získá tester rozsah veřejných IP adres (případně pouze jednu adresu).

- Penetrační testování webových aplikací

  V dnešní době hojně používané testování, které se rozšiřuje zejména využíváním webových aplikací, které ukládají citlivé údaje uživatelů. Z tohoto důvodu je potřeba webové aplikace testovat z hlediska jejich bezpečnosti. Testování probíhá získáním informací formou URL adresy dané webové aplikace, jedná-li se o testování pomocí Black boxu. V případě White boxu by tester měl k dispozici i zdrojový kód aplikace.
- Penetrační testy mobilních aplikací

  Telefonní zařízení s operačním systémem Android, iOS a další nejsou využívané pouze pro osobní účely, ale také ve firemním prostředí. Je tedy důležité zajistit bezpečnost nejen operačního systému, ale i nainstalovaných aplikací, pracující s citlivými informacemi. Vhodnými penetračními testy je možné najít případná rizika sytému a aplikací. [6][7][8]

6.1.1 Přístupy

Před začátkem provádění penetračních testů je nutné se rozhodnout, jaký z klíčových přístupů se bude používat. Lze provádět tzv. Black box, White box nebo Gray box testování. Rozhodujícím faktorem je úroveň a rozsah informací, které se testerovi poskytnou o dané společnosti a testovanému systému před začátkem testování.

White box

Před začátkem testování jsou testerovi poskytnuté veškeré potřebné informace o testovaném informačním systému. Jedná se o informace sítě, nakonfigurovaných parametrech v rámci software, ale tester má také přístup ke zdrojovému kódu, ve kterém se musí umět orientovat. Základní data jsou vždy testerovi známá a má k nim přístup. Tím dokáže rychleji identifikovat slabá místa systému a je schopný testy cílit na skutečná předem zřejmá specifika daného systému. Používá se zejména pro detekci logických chyb v kódu programu, ale také pro ladění kódu, hledání náhodných chyb a odhalené nesprávného naprogramování. Provádí se v nízké úrovni návrhu a implementovaného kódu. Test se považuje za úspěšný v případě, že tester rozpozná, jak se má program chovat. Tím pak může tester zjistit, zda se program odlišuje od zamýšleného cíle.

Black box

Principem tohoto přístupu je, že tester zná pouze základní informace o vstupech a potencionálních výstupech dané aplikace. Nijak není informován o vnitřní struktuře aplikace ani síťové infrastruktuře. Vychází se ze zadaných požadavků na informační systém a ověřuje se, jestli software vyhovuje. Metoda black box se nejvíce podobá reálnému útoku, avšak je poměrně časově náročná. Je zde i riziko přehlédnutí zranitelného místa, které se může nacházet uvnitř sítě nebo ve vnitřních částech aplikace. Testování se provádí na již kompletně hotovém systému. Hlavním významem je testování platných a neplatných vstupů z pohledu zákazníka.

Gray box

Cílem tohoto testování je zjistit případná rizika z důvodu nesprávné struktury nebo špatného používaní aplikace. Jedná se o kombinaci white box a black box testování, které je vhodné zejména pro webové aplikace. Metodika je nezávislá na platformě a jazyce. Gray box je silně závislý na použití ladícího programu hostitelské platformy ke spuštění a ověření software před testováním. Z důvodu absence zdrojového nebo binárního kódu není možné použít white box testování. Testuje se především jestli software splňuje zadání a dané specifikace.

6.1.2 Prostředí

Penetrační testování lze rozdělit podle toho, jestli se útočící osoba snaží proniknout do systému z vnějšího nebo vnitřního prostředí. Testování se používá ve firemních sítích, kde vzniká velké riziko takových útoků.

Interní penetrační testování

Testy jsou prováděné ve vnitřním prostředí dané společnosti. Dochází k simulaci útoku od běžného nezvýhodňovaného uživatele z vnitřní sítě (např. návštěva, zaměstnanec), který dostal přístup do interních firemních systémů. Tester usiluje o získání přístupu k datům, ke kterým nemá oprávněný přístup. Cílem testu je ověřit veškeré možnosti spojené s manipulací se soukromými daty (např. kopírování, upravování, odstranění) neoprávněnou osobou. Testy tedy prověřují funkčnost interních bezpečnostních mechanismů, které by měly dostatečně chránit firemní data a systémy před neoprávněnými přístupy běžných uživatelů.

Externí penetrační testování

Prověřují úroveň zabezpečení služeb a prvků dostupných z externího prostředí (např. internet), které mají cíl na vnitřní systém zadávající firmy. Hlavní náplní testů je oblast ověřit bezpečnost firewall, případných dalších prvků, oddělující vnější síť od vnitřní. Tester může být informován o vnitřní struktuře, ale nemusí mít žádnou informaci. Záleží, jestli se rozhodne pro přístup white box nebo black box testování.

6.1.3 Provedení

Penetrační testování je možné také rozdělit podle způsobu provedení. Jednotlivé testování může ovlivnit nejen kvalitu testů, ale i časovou náročnost. Je možné z toho hlediska rozdělit testy na manuální, automatické a semiautomatizované.

Manuální penetrační testování

Tester provádí veškeré testy manuálně na základě svých bohatých zkušeností. Jedná se o časově náročné testování, kdy testující musí mít rozsáhlé znalosti v testované oblasti, tj. HTML, PHP, SQL, JavaScript atd. Výhodou je možnost vytvoření propracované procedury a testy na míru pro speciální podmínky. Tester by měl být schopný popsat a interpretovat své záměry a důvody zvoleného testování. Musí být také schopný to vysvětlit absolutně nezasvěcené osobě, která nemá dostatečné znalosti v této oblasti (management, jednatel atd.).

Automatizované penetrační testování

Testy vykonávané automatizovaně jsou rychlejší, nabízejí více možností a rozšiřitelnost dle potřeb. Zpravidla bývají jednoduché v reprodukovatelnosti a verifikovatelnosti. Používané nástroje pro penetrační testování byly stvořené profesionály, kteří mají několikaletou praxi v oboru. Výhodou oproti manuálnímu testování je významně kratší čas na učení a následné využití v praxi. Pro testera je jednodušší se naučit se obsluhovat již připravenou aplikaci pro testování než se učit princip manuálního penetračního testování. Testování probíhá zadáním vstupních podmínek a parametrů. Aplikace pak provede veškeré testy, ke kterým je určena. Mezi nevýhody automatizovaného testování lze považovat nemožnost otestovat všechna zranitelná místa. Hodí se pro skenování síťových portů, kde je zapotřebí naskenovat vyhodnotit a zpracovat nepřeberné množství dat.

Semiautomatizované penetrační testování

Představují kombinaci manuálního a automatizovaného penetračního testování. Snaží se o kompromis mezi manuálním a automatickým testováním s maximálním využitím výhod z každé formy. [7]

6.2 Metodiky penetračního testování

V každém odvětví informačních technologií lze provádět a řídit testování dle známých metodik. Jedná se tedy o zdokumentovaný formální souhrn standardů, nástrojů, doporučení, modelů, postupů a samostatných pracovních úkonů, které jsou před zahájením projektu k dispozici.

6.2.1 OWASP

Nezisková organizace Open Web Application Security Project pracující na zlepšení zabezpečení software byla založena v roce 2001 Markem Curphey a Dennisem Groves. Jedná se o otevřenou komunitu, která umožňuje organizacím získávat, vyvíjet, udržovat a provozovat aplikace, které jsou důvěryhodné. Veškeré dokumenty, nástroje, projekty, kapitoly, fóra jsou otevřené a zdarma pro veškeré zájemce. Zajímající se o zlepšení bezpečnosti webových aplikací.

OWASP se specializuje především na webové aplikace, ale nově se začal zabývat i mobilními aplikacemi. Úmysl organizace je nasbírat co největší množství testovacích technik, které poté vysvětlí a snaží se jejich stav udržet. Vše potřebné k otestování webové aplikace je sepsané v manuálu OWASP Testing Guide. Manuál ve verzi v4 je rozdělený do třech kapitol – Testovací rámec OWASP, Testování zabezpečení webových aplikací a report. Verze v4 je aktuální stabilní verze projektu OWASP. Oproti předchozí třetí verzi nabízí aktualizaci všech kapitol. [19]

Projekt OWASP Top 10 je seznam nečastějších chyb zabezpečení webových aplikací. Ukazuje také na jejich rizika, dopady a navrhuje jejich zabezpečení. Největší rizika vyhodnocené společností OWASP jsou:

1. **Injection** – chyby umožňující injekci a spuštění kódu v SQL, OS, LDAP, pokud nejsou dostatečně ošetřená odesílaná data jako součást dotazu nebo příkazu. Tímto způsobem dokáže útočník spustit příkaz navíc pro manipulaci a přístup k datům, ke kterým nemá žádná oprávnění.
2. **Broken Authentication** – nesprávná implementace funkcí aplikace související se správou relací a ověřováním umožňuje útočníkovi získat hesla, tokeny relací, klíče nebo jinak zneužít chyby v implemtentaci k trvalému nebo dočasnému převzetí uživatelské identity.
3. **Sensitive Data Exposure** – webové aplikace a rozhraní API často řádně neuchrání citlivá data, tj. zdravotnické informace, finanční nebo osobní údaje. Útočník je schopný takto málo chráněná data získat, ukrást nebo změnit tak, aby mohl provést podvod s kreditní kartou, ukrást identitu nebo spáchat jiný trestný čin s využitím získaných dat. Citlivá data si žádají nadstandartní šifrování a ochranu. Nejen však samotných dat, ale i šifrování přenosu mezi externími a interními službami a také mezi webovým serverem a koncovým uživatelem.
4. **XML External Entitiee (XXE)** – zastaralé nebo chybně nakonfigurované procesory XML vyhodnocující externí reference na entity uvnitř XML dokumentů. Tyto reference mohou odhalit vnitřní soubory, síťové disky, vykonávat interní skenování portů, spuštění vzdáleného kódu nebo pomoct při vytvoření útoku DoS.
5. **Broken Access Control** – neautentizovaný uživatel nemá přístup do veškerých částí systému, ale tyto omezení jsou často vynucované nedostatečně a dají se zneužít chyby pro přístup do části segmentu, které nemají být neautentizovanému uživateli vůbec přístupné. Stejné je to i s uživatelem s nedostatečnými přístupovými právy.
6. **Security Misconfiguration** – chybná konfigurace zabezpečení kvůli vlastní nedostatečné ruční konfiguraci, kvůli externí ad-hoc konfiguraci nebo žádné konfiguraci (bez certifikátu SSL nebo Let´s encrypt). Nedostatečně

zabezpečená výchozí konfigurace, využívání nastavení pro ulehčení developmentu na ostrých stránkách, otevřené cloudové uložiště, nesprávná konfigurace hlavičky HTTP a podrobné chybové zprávy obsahující citlivé informace. Veškeré operační systémy, knihovny a architektury musejí být nejen bezpečně nakonfigurované, ale musí být i zavčas opraveny nebo aktualizovány.

7. **Cross-Site Scripting (XSS)** – chyby XSS vznikají vždy, když aplikace pracuje na webové stránce s nedůvěryhodnými daty bez řádné validace při odstraňování nevhodných znaků, nebo při obnovování webových stránkách z uživatelských dat za použití API prohlížeče, které může vytvářet JavaScript nebo HTML. XSS útoky útočníkovi dovolují spustit skripty v napadeném prohlížeči, zneužití přihlášení, zachycení dat nebo přesměrování uživatele na podvodné stránky.
8. **Insecure Deserialization** – nedostatečně zabezpečená deserializace může vést ke vzdálenému spuštění nechtěného kódu. I přesto, že vady deserializace nevedou ke spuštění vzdáleného kódu, je možné je využít k vykonávání dalších útoků.
9. **Using Components with Known Vulnerabilities** – komponenty webových aplikací jako jsou frameworky, knihovny a jiné softwarové moduly jsou spouštěny s rovnocennými právy jako aplikace. Obsahuje-li nějaký komponent známku zranitelnosti a ta je zneužita, může takový útok zapříčinit ztrátu dat nebo celkové převzetí vedení serveru. Celkovou bezpečnost aplikace nebo API může zničit využívání komponentů se známou zranitelností, umožňující různé formy útoků.
10. **Insufficient Logging & Monitoring** – nepostačující logování a monitorování spolu s neefektivní nebo neexistující integrací reakcí na útoky umožňuje hlubší proniknutí do systému, získat přístup do dalších částí systémů, prodloužit si delší čas pro přístup, upravit, změnit nebo úplně odstranit data. Studie dokazují, že čas na odhalení narušení je déle než 200 dní a je spíše detekován externími stranami než interními.[20][22]

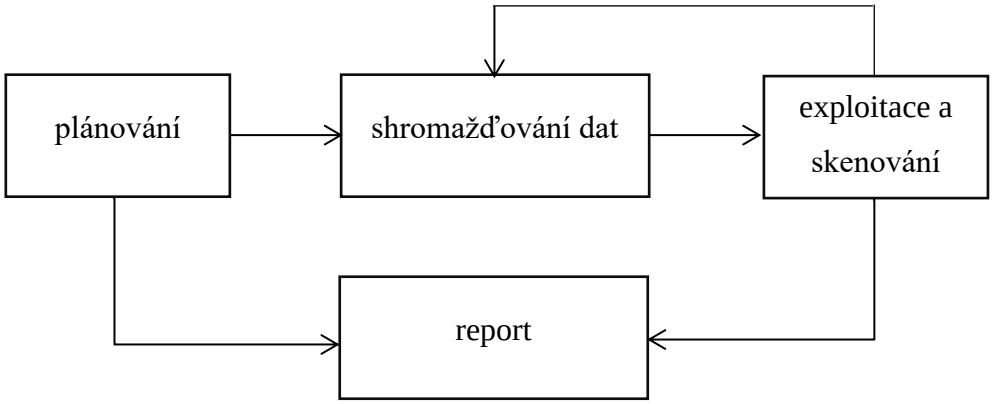
6.3 Průběh penetračního testování

Postup penetračních testů popisuje průběh penetračního testování, který je zobrazený na obr. 2. Existuje mnoho přístupů, které využívají volně dostupné nástroje, tzn. metodika je veřejně známá. V častých případech softwarové podniky testují na základě vlastního know-how, které si vybudovali za několika letou praxi a tím si svoje postupy chrání.

Penetrační testování lze rozvrhnout do 4 fází:

1. Rozsah a cíl penetračního testu (plánování)
2. Shromažďování dat
3. Exploitace a skenování
4. Report

Obrázek 2 Fáze penetračního testování



Zdroj: vlastní zpracování na základě [7]

Rozsah a cíl penetračního testu

V prvotní fázi je potřeba si určit obecné zadání a cíl, na které budou následně použity vybrané penetrační testy. Je vhodné se zaměřit na prioritní cíle, protože není možné otestovat vše na 100 %. Cíle se tedy rozdělí do několika dílčích cílů, aby se dokázalo určit, co všechno má být otestované a na co je potřeba se opravdu zaměřit. Každý tester by měl znát odpověď na základní otázky:

- Co se bude testovat?
- Jakým způsobem se bude testovat? (typy testů, nástroje)
- Kdy budou testy vykonávané?

Shromažďování dat

Po důkladném plánování průběhu a cílů penetračního testování následuje sběr dat, kde informace jsou sbírané za účelem pochopení, jakým způsobem systém pracuje. Tím je možné lépe zjistit, kde se nachází zranitelná místa. Z testovaného systému je možné získat informace prostřednictvím různých technik pasivně nebo aktivně.

Pasivní sběr dat je získávání dat bez jakékoliv možnosti detekce činnosti objektům testování. Jedná se o kombinaci získávání informací předstíráním činnosti běžného uživatele a získávání údajů z veřejně dostupných zdrojů. Může se jednat o použití aplikace nebo návštěva webové stránky.

Aktivní sběr dat je v nějakých situacích možný rozeznat. Jde např. o skenování portů, zachycování síťové komunikace, IP adresy atd. Pokud cíl disponuje systémem detekce vniknutí (IDS), systém prevence vniknutí (IPS), firewall dokáže odhalit tento sběr dat.

Pomocí různých technik lze získat zajímavé informace:

- IP adresa a informace o hostiteli – dotazy InterNIC (WHOIS), odposlechy síťové komunikace
- Kontakty a informace o jménech zaměstnanců – získávání pomocí doménových serverů nebo prohledávání firemních webových serverů
- Systémové informace – získávané metodami jako je výpočet síťového vstupně/výstupního systému (NetBIOS) a síťového informačního systému (NIS)
- Informace o službách a aplikacích – verze používaného operačního systému

Reálná procházka po firmě nebo prohledávání odpadků můžou být součástí sběru informací. Technika se nazývá „dumpster diving“. Tímto způsobem lze odhalit další informace o vybraném cíli penetračních testů, např. na papíru napsané heslo.

V další části sběru dat se vyhodnocují už získané informace o systému, které se využijí pro analýzu zranitelnosti systému s použitím databázové zranitelnosti. Jsou veřejně dostupné např. národní databáze zranitelností (NVD). Zranitelnosti se identifikují ručně z veřejných dostupných zdrojů, což je poměrně časově náročné, ale dokážou se tím odhalit ta slabá místa, která jsou přehlídnuta automatickými skenery.

Exploitace a skenování

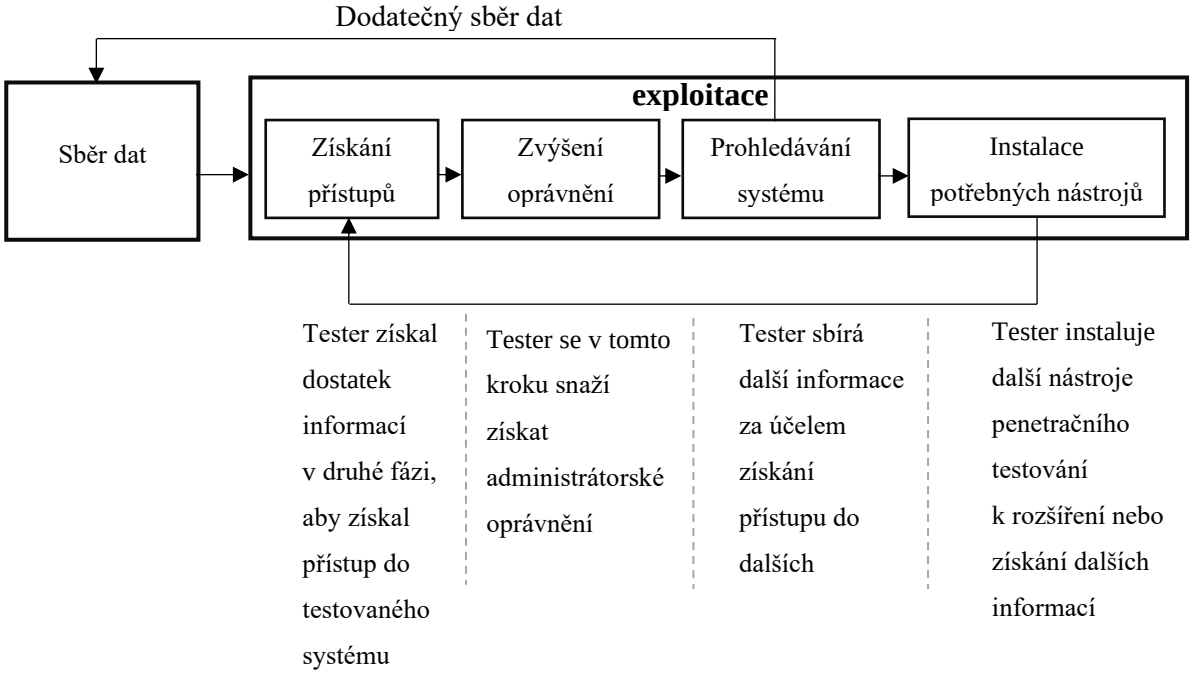
Samotné provedení penetračního testu je proces, který verifikuje identifikovaná potencionálně zranitelná místa aplikace. Využívají se k tomu tzv. exploity. Exploit programy nebo skripty jsou nástroje, které využívají zranitelnost systému. Je-li útok zdařil či potvrzená zranitelnost, tak následuje návrh zmírňujícího opatření k zamezení bezpečnostního rizika.

Některé exploity přinesou o systému nové informace, pomocí kterých je možné identifikovat další zranitelná místa. Pomocí exploitů je možné získat větší oprávnění v systému. To lze využít pro instalaci nástrojů přímo v systému a tím přispět k testovacímu procesu. Na obrázku 3 je zobrazené schéma exploitace s návazností na sběr dat.

Nejčetnější zranitelnosti exploitované v průběhu penetračního testu je možné rozdělit do těchto kategorií:

- **Chybná konfigurace** – defaultně nastavené prvky, které nejrychleji napadnutelné, nevhodně nastavené bezpečnostní prvky
- **Chyby v jádře operačního systému** – chyby v kódu OS, které mohou ohrozit celý systém
- **Přetečení bufferu** – možnost spuštění nechtěného kódu s gradujícími privilegii
- Nepřiměřená validace vstupů

Obrázek 3 Schéma exploitace s návazností na sběr dat



Zdroj: vlastní zpracování na základě [7]

Report

Závěrečnou fází je vypracování reportu, ve kterém by měly být se sumarizované výsledky z jednotlivých testů, doplněné poznatky a zjištění. Součástí reportu v některých případech mohou být i doporučení na vyřešení bezpečnostních problémů.

Každý report by měl obsahovat:

- Záměr testování – specifikování cílů
- Technická zpráva – popisuje technické detaily penetračního testování
- Zhodnocení – celkový přehled testů, demonstrace dopadu zranitelnosti na bezpečnost testovaného systému, může být doplněné o strategii pro zabezpečení a o následných krocích

Report by měl být psaný ve srozumitelné formě jak pro techniky, tak i pro manažerské pracovníky. [7]

6.4 Nástroje penetračního testování

Během celého penetračního testování je využíváno celé řady nástrojů pro sběr informací, skenování systému nebo dokumentaci.

WHOIS – webové rozhraní pro získávání záznamů o majiteli domény, DNS, name servery nebo IP adresy. Informace vrácené rozhraním WHOIS obsahují kontaktní údaje (email, telefon) a další metadata.[21]

Google – celosvětově nejpoužívanější vyhledávač, který poskytuje další možnosti pro pokročilé vyhledávání informací. Pomocí speciálně zkonstruovanému hledání výrazů a operátorů je možné získat veřejně dostupné informace bez vědomí testovaného objektu. Získání základních informací o zákaznícovi, zaměstnancích a webové stránce mohou pomoct přípravě ve fázi plánování.

Sociální sítě – zapojení firem do sociálních sítí je způsob, jakým si jednoduše a efektivně vyměnit svoje informace. Lidé si stále neuvědomují, jaká jsou možná rizika spojená s poskytováním informací široké veřejnosti a tím se mohou dostat až k potencionálním hackerům. Tyto informace mohou průnik do systému velmi zjednodušit.

Ping – jednoduchý test časové odezvy serveru a dostupnosti, může sloužit pro odhalování IP adres v síti. V méně častých případech může být příkaz zneužitý pro vytvoření ping of death nebo ping flood útoku na DoS serveru.

Kali Linux – tvůrcem systému je společnost Offensive Security, zajišťující velkou skupinu nástrojů skrz linuxovou distribuci, která testerům poskytuje nástroje a zároveň samotné testovací prostředí. Samostatné nástroje je možné rozdělit do několika skupin podle možností automatizace, technické náročnosti obsluhy a podle způsobu získávání informací.

ZAP – nástroj Zed Attact Proxy (ZAP) je volně stažitelný nástroj sloužící pro penetrační testování s open-source kódem. Je spravován pod záštitou OWASP a navržen pro testování webových aplikací podle OWASP TOP 10. Ve svém jádru je ZAP to, co je známé jako prostřední proxy server. Nachází se mezi webovou aplikací a prohlížečem, tím může zkontrolovat a zachytit odeslané zprávy. V případě potřeby upravit obsah a následovně pakety přeposlat na cíl. Aplikaci je možné spustit samostatně nebo přes daemon process. Je-li používaný jiný síťový proxy server, je možné nástroj nakonfigurovat pro připojení k tomuto proxy serveru.

Uniscan – kombinuje v sobě několik nástrojů skenování webových stránek – WHOIS, Nmap, Nslooup a další nástroje pro hledání zranitelnosti typu Remote File Include, Remote Command Execution nebo Local File Include. [23][7]

7 Testování platformy Shop5.cz

Pro penetrační testování byla vybraná platforma shop5.cz, na které k dnešnímu dni běží cca 800 e-shopů. Testování platformy shop5 bude probíhat se souhlasem majitele Bc. Martina Březovského. Celý e-shop je na testovací doméně *http://jana.testsXXX.XX*. Tato doména je vytvořená majitelem, a proto by použití tohoto názvu nemělo způsobit žádné problémy. Jména uživatelů, adresy, telefonní čísla, emaily, IP adresy nebo jiné údaje, které by mohli vést k identifikaci budou částečně nebo zcela nahrazené, protože je to podmínka majitele platformy. Veškerá naměřená data jsou reálná a odpovídají skutečnosti v čase vykonávání penetračního testování.

Jedná se o interní penetrační testování, kdy jednatel společnosti souhlasil s vykonáváním testování.

7.1 Plánování

V první fázi plánování byl stanoven rozsah a cíl testování, časový harmonogram, použité nástroje a techniky. Kompletní popis uveden v zadávacím protokolu penetračního testu – viz tabulka 1.

Následujícím krokem byla příprava software a hardware pro spuštění testování. Pro vytvoření izolovaného prostředí a zároveň prostředí, které se dá jednoduše přenést a znovu použít byl zvolený virtuální software od společnosti ORACLE – Virtual Box. Použitím tohoto nástroje je možné OS virtualizovat. Navenek se virtuální OS jeví jako standartní systém, ale uživatel má plnou kontrolu nad připojenými zařízeními, nastavením síťových virtuálních prvků atd. Je možné využít základní vlastnosti „virtuálního stroje“, přenést a spustit na jiném hostitelském počítači. Přenos může být např. pomocí flash disku nebo jiného úložného zařízení.

Tabulka 1 Zadávací protokol penetračního testování

Zadávací protokol penetračního testování webové aplikace shop5			
Zadavatel		**Vykonavatel**	
Jméno	Betulasoft s.r.o.	**Jméno**	Bc. Jana Sladká
Adresa	Aloise Jiráska 260, Příbram	**Adresa**	Drkolnovská 209, Příbram
Odpovědná osoba	Bc. Martin Březovský	**Odpovědná osoba**	Bc. Jana Sladká
Popis projektu			
Rozsah penetračního testování			
Otestovat zabezpečení webové aplikace umístěné na doméně http://jana.testsXXX.XX. Zejména zranitelnosti podle OWASP TOP 10 projektu a otestování pomocí automatizovaných nástrojů.			
Časový harmonogram			
25. 2. 2021 - Zahájení penetračního testování (příprava SW a HW)			
1. 3. 2021 - 15. 4. 2021 - Penetrační testování webové aplikace			
15. 4. 2021 - 18. 4. 2021 - Tvoření reportu a analýza výsledků			
20. 4. 2021 - Odevzdání reportu zadavateli			
Cíl penetračního testování			
Cílem penetračního testování je ověření bezpečnosti webové aplikace na doméně http://jana.testsXXX.XX.			
Zaměření a typy penetračního testování			
Kombinace manuálního a automatizovaného testování, využití nástrojů Kali Linux pro otestování aplikace podle zadaného rozsahu.			
Použitý SW a HW			
Hardware vykonávatele, Kali Linux, Nmap, Zap a další nezbytné nástroje.			
Další požadavky			
Nejsou ze strany zadavatele uvedeny.			

Zdroj: Autor práce

Důležitým nástrojem je operační systém. Pro penetrační testování byl vybraný operační systém od společnosti Offensive Security Kali Linux – linuxová distribuce specializovaná na penetrační testování a forenzní analýzu.

Pro stabilnější a plynulejší práci byl systém nainstalovaný do virtuálního stroje. Kali Linux nabízí velké množství veřejně známých nástrojů. Další balíčky je možné kdykoliv doinstalovat. Mezi nejznámější nástroje patří – SQLMap, Wireshark, Hydra, Nmap, Burpsuit, ZAP, Aircrack-ng nebo framework Metasploit.

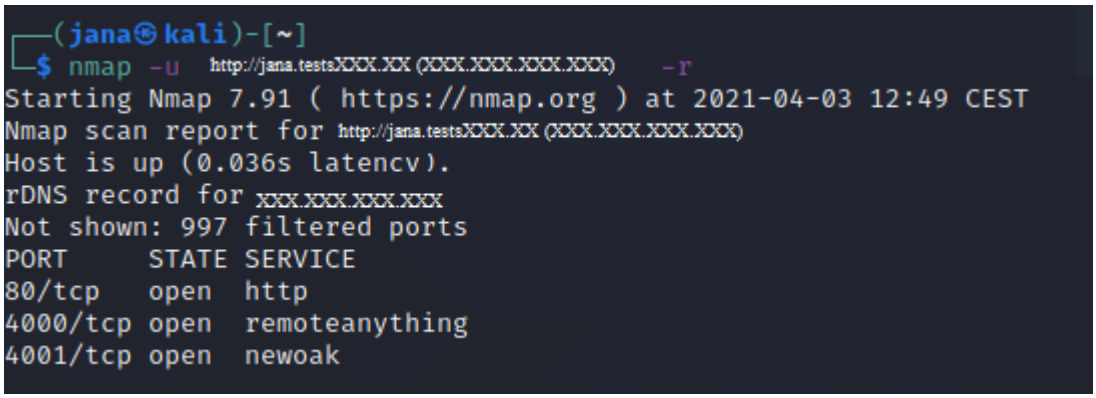
7.2 Shromažďování dat

Realizace penetračního testování začíná sbíráním veřejně dostupných informací. Pro shromažďování dat se využívají zejména veřejně dostupné nástroje jako je WHOIS, nmap a další. Procházení těchto všech nástrojů je poněkud zdlouhavé, tak byl vybraný nástroj Uniscan, který v sobě všechny tyto nástroje kombinuje.

Uniscan

Ve výsledcích skenování je možné vidět na první pohled několik otevřených portů. Z obrázku 4 v prvním řádku je patrná verze, datum a čas skenování. Na dalším řádku je obsažena cílová IP adresa (v tomto případě IPv4) a název DNS. Od nmap je vyžadováno, že zobrazí zajímavé porty, i když všechny skenované porty jsou započítávány. Porty, které jsou nejvíce zajímavé, protože jsou zřídka vídané nebo v otevřeném stavu, jsou rozepsány na jednotlivém řádku. Je-li mnoho portů v jednom neotevřeném stavu, jsou považovány za výchozí stav a agregovány na jeden řádek. Z pátého řádku je zřejmé, že nebylo zobrazeno 997 portů. Následuje tabulka „zajímavých portů“, kde jsou zobrazené 3 otevřené porty.

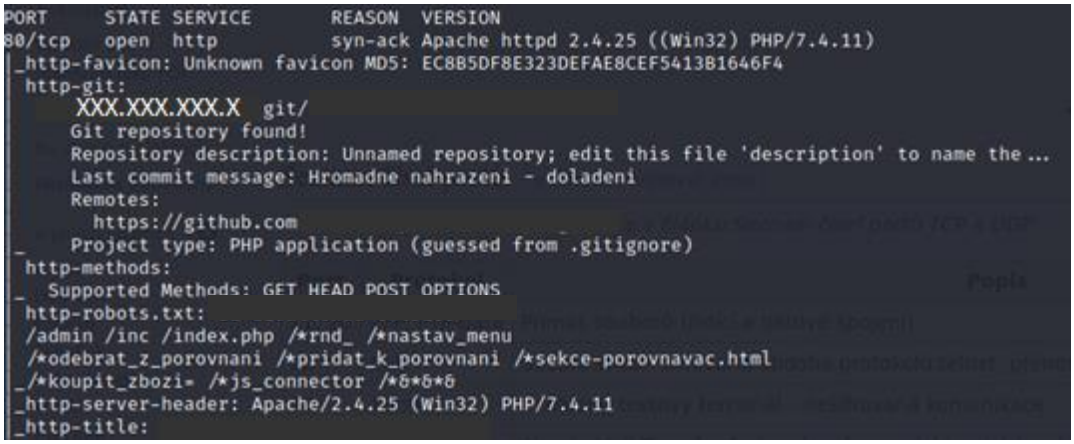
Obrázek 4 Zajímavé porty



Zdroj: Autorka práce

Příkaz uvedený na obrázku 4 odhaluje pouze zajímavé porty. Použitím příkazu „*nmap -vv -A jana.testsXXX.XX*“ se vypíše detailněji co se na jednotlivých portech nachází.

Obrázek 5 Protokol 80/tcp



Zdroj: Autorka práce

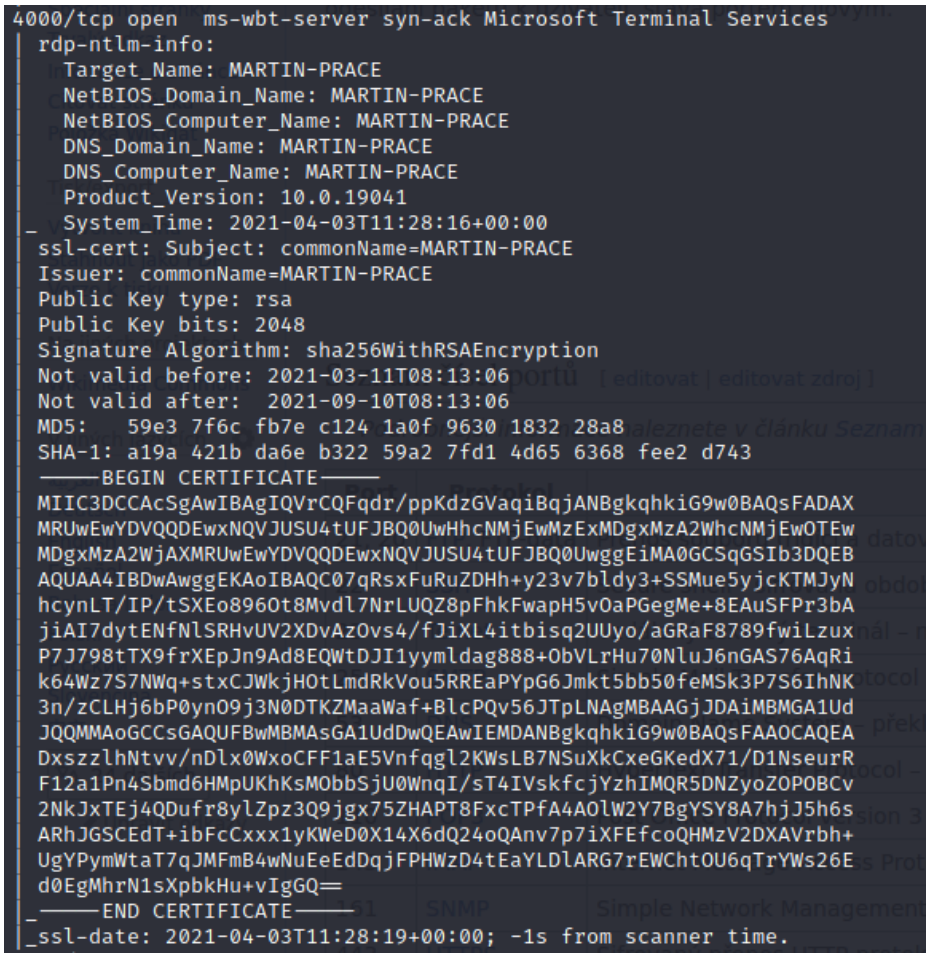
Na portu 80/tcp je nachází HyperText Transfer Protocol používaný pro přenos webových stránek a dalších dat. Verze je Apache HTTP 2.4.25 ((Win32) PHP/7.4.11). Je tedy zřejmé, že testovaná doména opravdu nevyužívá šifrovaný přenos přes protokol TLS, pro který se využívá port 443. Skenování tohoto protokolu také odhalilo favicon, git, method, soubor robot.txt, hlavičku serveru a titulek. Zajímavé v tomto nálezu je url adresa githubu, kde se nachází projekt shop5. Github slouží jako verzovací nástroj pro vývoj software. Může se jednat o bezplatnou verzi pro open-source projekty nebo placenou verzi pro soukromé projekty. Webová aplikace shop5.cz je na githubu umístěna jako soukromá a není ji tedy možné veřejně vyhledat. Kromě url adresy, kde je aplikace umístěna je také vidět jaký poslední commit (úpravený soubor) byl na server odeslán, a že je projekt typu PHP aplikace.

V souboru robot.txt je celkově zakázáno 11 položek – viz obrázek 5. Položka disallow robotovi říká, aby dané adresáře neindexoval. Pokud je tam pouze lomítko „/“, tak to znamená, že robot vůbec nesmí do celého adresáře. V případě shop5 se jedná např. o adresář admin. Je-li tam hvězdička a lomítko „*/“, pro robota to znamená, že nesmí indexovat stránku, jejíž url adresa kdekoliv obsahuje daný výraz. Dle obr. 5 to může být např. odebrat_z_porovnani, koupit_zbozi a další.

Na portu 4000/tcp a 4001/tcp se nacházejí 2 počítače využívající operační systém Microsoft (např. Windows 10). V prvním odstavci na obrázku 6 jsou základní informace o počítači. Zajímavější informace jsou však v odstavci 2, kde je možné vidět informace o certifikátu SSL využívající algoritmus RSA pro digitální podpisy nebo šifrování dat.

Veškeré certifikáty SSL/TLS, které se v dnešní době používají, mají velikost klíče 2048 bitů, čímž činí web bezpečným. Je možné vyčíst, že certifikát je platný od 11. 3. 2021 a jak takový veřejný certifikát vypadá.

Obrázek 6 Protokol 4000/tcp



Zdroj: Autorka práce

7.3 Exploitace a skenování

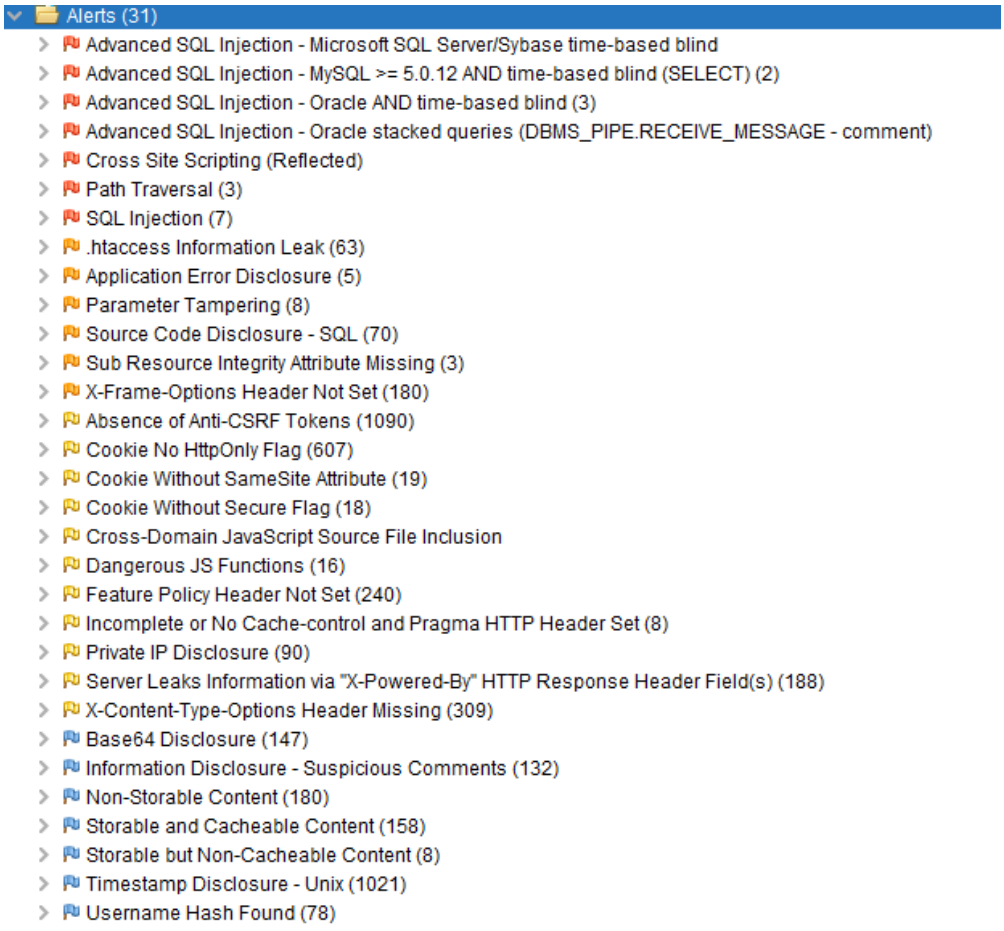
Po shromáždění veřejně dostupných dat přichází na řadu samotné testování aplikace, pro kterou je vhodné použít nástroj ZAP.

ZAP

Pro spuštění a instalaci je potřebné mít nainstalovanou Javu 8+ a vyšší. V Kali Linux je nástroj ZAP již v nainstalovaný, ale lze jej nainstalovat na jakýkoliv OS.

Testování je možné spustit jako aktivní nebo pasivní prohledávání protokolu HTTP, která najdou konkrétní chyby zabezpečení. Důležité je upozornit na fakt, že se jedná pouze o příklady upozornění a mnoho pravidel obsahuje různé podrobnosti v závislosti na přesném problému, ke kterému došlo. Jelikož nebylo možné dokončit po několika opakovaných pokusech automatický scan, tak byl použit manuální. Při manuálním testování aplikace bylo zaznamenáno několik upozornění, a to od těch největších zranitelností, po ty nejmenší (informativní). Z obrázku 7 je patrné, že v aplikaci bylo odhaleno celkově 31 výstrah, které jsou uspořádány sestupně.

Obrázek 7 Nalezené zranitelnosti v ZAP



Zdroj: Autorka práce

Advanced SQL Injection

Celkově 4 výstrahy byly zaznamenány u Advanced SQL Injection – Time-base Blind SQL. Jedná se o útok, kdy útočník (v tomhle případě aplikace ZAP) odešle do databáze SQL dotazy, které vynutí, aby databáze před odpovědí na určitou dobu čekala. Doba odezvy vrátí útočníkovi, zda je výsledek true nebo false. I když nejsou v tomto případě vrácená žádná data, tak to útočníkovi dokáže odvodit používanou hodnotu.

Cross Site Scripting (Reflected)

Cross-site scripting odhalilo problém v url adrese *http://jana.testsXXX.XX /cz-kategorie_0-0.html* - <script>alert(1);</script>. Tento test je typický pro skriptovací jazyky, protože funguje v každém prohlížeči a rychle se zapíná. Vložení JavaScritptu do webové aplikace může sloužit ke krádeži přihlašovacích údajů, falšování identity, phishingu nebo dokonce jen přeskakovat útok přes ně k jinému cíli.

Path Traversal

Cílem procházení adresářů je získání přístup k adresářům a souborům uloženým mimo kořenovou složku webu. Manipulace s proměnnými, které odkazují na soubory pomocí sekvence ve tvaru „tečka-tečka.lomítko (../)“ nebo absolutní cesty k souborům, je možné přistupovat k libovolným souborům nebo adresářům uložených v systému (včetně zdrojového kódu aplikace, konfigurace atd.). Jedná se o pozitivně falešnou výstrahu, protože při tomto útoku byla IP adresa zařazená na black list a e-shop se pro ni stal zcela neviditelný.

SQL Injection

Technika, která může zničit celou databázi. Jde o umístění škodlivého kódu do příkazu SQL prostřednictvím vstupní webové stránky. Jedná se o pozitivně falešnou výstrahu, protože při tomto útoku byla IP adresa zařazená na black list a e-shop se stal pro ni zcela neviditelný.

.htaccess Information Leak

Soubory .htaccess mohou být využívané ke změně konfigurace software Apache Web Server k povolení nebo zakázání dalších funkcí (případně i těch, které nabízí software Apache). Při útoku byla zablokovaná IP adresa - blacklist_file_check 10.161.116.84. Jedná se tedy o falešně pozitivní výstrahu.

Application Error Disclosure

Jedná se o varovnou / chybovou zprávu, která může obsahovat soukromé informace, např. umístění souboru. Informace je možné využít ke spuštění dalších útoků na webovou aplikaci. Může se jednat o falešně pozitivní výstrahu, pokud se chybová zpráva nalezne v dokumentaci.

Na stránce *http://jana.testsXXX.XX/admin/index.php?sekce=zbozi*

- Warning –„ count(): Parameter must be an array or an object that implements Countable in X:\XXX\XXX\XX5\xxx.php on line 11935
“

Na stránce http://jana. testsXXX.XX /admin/index.php

- Fatal error - Out of memory (allocated 394264576) (tried to allocate 157355280 bytes) in X:\XXX\XXX\XX5\xxx.php on line 827
- Warning - A non-numeric value encountered in X:\XXX\XXX\XX5\xxx.php on line 147
- Warning - Invalid argument supplied for foreach() in X:\XXX\XXX\XX5\xxx.php on line 295

Parameter Tampering

Neoprávněná manipulace s webovými parametry umožňuje manipulaci s vyměňovanými parametry mezi serverem a klientem za účelem úpravy dat aplikace. Zejména oprávnění uživatelů, množství a cena produktů apod. Nejčastěji bývají uložené informace v skryté formě pole, cookies nebo URL Query Strings. Výsledkem útoku je zobrazení chybové stránky.

Source Code Disclosure – SQL

Útok slouží k možnému odhalení zdrojového kódu aplikace, souboru a zobrazení konfiguračních souborů. Tím je možné získat citlivé informace o aplikaci (filtry, ověření vstupů, dotazy, připojovací řetězce databáze atd.).

Sub Resource Integrity Attribute Missing

Chybějící atribut integrity dílčího prostředku kontroluje, jestli chybí atribut integrity ve skriptu, nebo prvek propojení obsluhovaný externím prostředkem (např. CDN). Pomáhá zmírnit útok, kdy byla CDN ohrožena a obsah byl nahrazen škodlivým obsahem. Síť pro doručování obsahu (Content Delivery Network) je síť počítačů propojených vzájemně skrz internet, zvyšující dostupnost dat uživatelům.

X-Frame-Options Header Not Set

Hlavičku odpovědi HTTP X-Frame-Options lze použít k označení, zda prohlížeč má povolené vykreslování stránky v <iframe>, <frame>, <object> nebo <embed>. Webové aplikace to mohou použít, aby se zabránilo útokům „click-jacking“ tím, že zajistí, aby jejich obsah nebylo možné vložit do jiných webů. Přidané zabezpečení je vhodné pouze v případě, že uživatel přistupující k dokumentu, používá prohlížeč, který aplikace X-Frame-Option podporuje. Funguje pouze v případě, že je nastavený v hlavičce (např. v metaznačkách - <meta http-equiv=“X-Frame-Options“ content=“deny“>).

Absence of Anti-CSRF Tokens

Token CSRF je tajná, jedinečná a nepředvídatelná hodnota generovaná webovou aplikací na straně serveru. Ke klientovi je přenášena tak, že je již zanesena v následném požadavku HTTP provedeném klientem. Při pozdějším požadavku aplikace je ze strany serveru ověřen, zdali obsahuje požadovaný token. V opačném případě je požadavek zamítnutý. CSRF tokeny mohou zabránit útokům tím, že hackerovi neumožní vytvořit platný požadavek HTTP. Vzhledem k tomu, že útočník není schopen předpovědět hodnotu tokenu, nemůže vytvořit požadavek se všemi požadovanými parametry, které jsou nezbytné k tomu, aby aplikace požadavku vyhověla.

Dangerous JS Functions

V JavaScriptu je funkce eval() jedna z nejnebezpečnějších funkcí v JS. Funkce vezme řetězec a pokusí se ho spustit jako kód JavaScriptu. Může dojít k negativnímu ovlivnění vnitřního stavu aplikace. Je-li u webové aplikace vstup pomocí textového pole, na kterém běží eval(), útočník může snadno spustit skript a tím provede XSS útok založený na DOM (skriptování mezi weby). Nejnovější prohlížeče umožňují použití JSON.parse(), stále jsou však starší prohlížeče, které tuto funkci neumožňuje a je zapotřebí použít JavaScript eval() jako metodu pro vytváření objektu JSON. Podobným způsobem to lze zneužít k provedení XSS útoku na straně serveru.

Timestamp Disclosure – Unix

Časové razítko zveřejněné webovým nebo aplikačním serverem lze využít k načtení možných citlivých informací (např. token, během ověřování nebo šifrování). Jedná se o unixový čas (např. 1617894017 je čtvrtek 8. dubna 2021 17:00:17). Veškeré výstrahy tohoto typu by měly být ručně prozkoumány, zdali se jedná opravdu o Timestamp. Mělo by zde dojím k přezkoumání, že se jedná o opravdové úniky časového razítka serveru. Zveřejněná data timestamp nejsou citlivá, protože se v žádné formě nepoužívají ke generování citlivých informací. V tomto případě lze tedy výstrahu ignorovat.

Username Hash Found

Byl nalezený hash uživatelského jména což může znamenat, že chybí zabezpečení IDOR (Insecure Direct Object Reference). Je zapotřebí ruční testování majitelem aplikace, aby se dalo vyvrátit, že je to možné zneužít. Pro šifrování je použitý hash Md5, což je široce používaná kryptografická funkce, která vytváří hodnotu hash 16 B (128bit), obvykle vyjádřenou jako šestnáctkové číslo v kombinaci číslic a písmen. Byla nalezena hodnota – „21232f297a57a5a743894a0e4a801fc3“.

7.4 Report

Vytvořený report (viz tabulka 2) byl předaný zadavateli. Webová aplikace shop5 odolala většině penetračních testů v aplikace ZAP. Byly odhaleny celkově 2 nejzávažnější zranitelnosti, které bude potřeba ze strany majitele SW vyřešit. Vhodné řešení je uvedeno v protokolu. Dalo by se tedy říct, že aplikace je nedostatečně zabezpečená a bude potřeba zabezpečení vylepšit.

Tabulka 2 Report penetračního testování webové aplikace shop5

Report penetračního testování webové aplikace shop5			
Zadavatel		**Vykonavatel**	
Jméno	Betulasoft s.r.o.	**Jméno**	Bc. Jana Sladká
Adresa	Aloise Jiráska 260, Příbram	**Adresa**	Drkolnovská 209, Příbram
Odpovědná osoba	Bc. Martin Březovský	**Odpovědná osoba**	Bc. Jana Sladká
Specifikace penetračního testování			
Cílem penetračního testování je ověření bezpečnosti webové aplikace na doméně http://jana.testsXXX.XX. Základem pro testování je seznam běžných zranitelností vyjmenovaných v OWASP TOP 10.			
Technický report			
Testování probíhalo dle předem stanovených termínů. Využité nástroje: Kali Linux, Uniscan, ZAP **Nejzávažnější zranitelnosti:** Advance SQL Injection Cross Site Scripting **Méně závažné zranitelnosti** Aplication Tampering Source Code Disclosure Sub Resource Integrity Attribute Missing X-Frame-Options Header Not Set			
Doporučené opatření:			
Advance SQL Injection - https://www.owasp.org/index.php/Top_10_2010-A1, https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet			
Cross Site Scripting - http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html			
X-Frame-Options Header Not Set - https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options			
Další informace			
Zadavatel obdržel od vykonavatele k této zprávě i report z aplikace ZAP, který má 65 stránek. Dále také informace, které jsou zmíněné v kapitole 6.2 a 6.3			

Zdroj: Autorka práce

8 Testování platformy Wordpress

Instalace Open-source aplikace Wordpress je o něco náročnější než komerční řešení, u kterého stačí pouze nastavit správné DNS záznamy a u majitele aplikace vložit doménu. Doména s webhostingem byla zakoupena u společnosti Wedos, kde je možné Wordpress nainstalovat pomocí funkce „Instalátor aplikací“. Stačilo pár kliknutí a aplikace byla nainstalována a spuštěna na doméně http://sweet-soft.eu. Aby bylo možné na dané doméně provozovat e-shop, bylo zapotřebí nainstalovat plugin WooCommerce, kde jsou veškeré nezbytné funkce pro online obchodování.

Toto penetrační testování slouží pouze pro porovnání open-source aplikace a komerční aplikace z hlediska zabezpečení. Testování bude probíhat na stejném principu jako byla otestována platforma Shop5.

8.1 Plánování

Ve fázi plánování je zapotřebí si stanovit rozsah a cíl testování domény *http://sweet-soft.eu*, na které je umístěna platforma Wordpress a nainstalovaný základní plugin WooCommerce bez jakéhokoliv dalšího zabezpečení. Použité nástroje a techniky budou shodné jako u předchozí testované platformy shop5. Zadávací protokol je uvedený v tabulce 3.

Prostředí pro totožné otestování aplikace byl zvolen Virtual box od společnosti Oracle, kam byl nainstalovaný operační systém Kali Linux od společnosti Offensive Securiy.

Tabulka 3 Zadávací protokol pro testování open-source aplikace Wordpress

Zadávací protokol penetračního testování webové aplikace Wordpress	
Vykonavatel	
Jméno	Bc. Jana Sladká
Adresa	Drkolnovská 209, Příbram
Odpovědná osoba	Bc. Jana Sladká
Popis projektu	
Rozsah penetračního testování	
Otestovat zabezpečení webové aplikace umístěné na doméně http://sweet-soft.eu. Zejména zranitelnosti podle OWASP TOP 10 projektu a otestování pomocí automatizovaných nástrojů.	
Cíl penetračního testování	
Cílem penetračního testování je ověření bezpečnosti webové aplikace na doméně http://sweet-soft.eu.	
Zaměření a typy penetračního testování	
Kombinace manuálního a automatizovaného testování, využití nástrojů Kali Linux pro otestování aplikace podle zadaného rozsahu.	
Použitý SW a HW	
Hardware vykonavatele, Kali Linux, Nmap, Zap a další nezbytné nástroje.	
Další požadavky	
Testování open-source platformy musí probíhat stejně jako u platformy shop5.	

Zdroj: Autorka práce

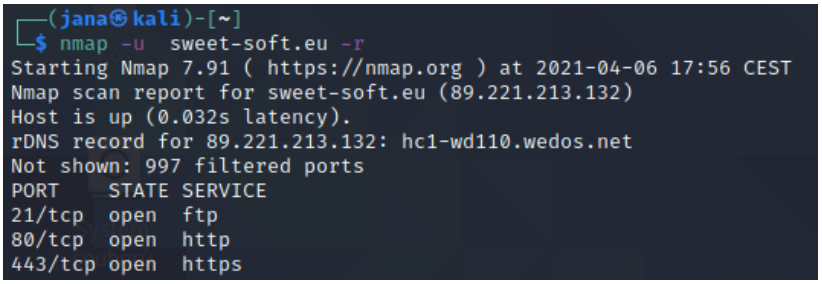
8.2 Shromažďování dat

Realizace každého penetračního testování začíná sbíráním veřejně dostupných informací.

Uniscan

Spuštěním příkazu *nmap -u sweet-soft.eu -r* došlo ke skenování 1000 portů. Z obrázku 8 je na prvním řádku možné vidět, že bylo zahájeno 6. 4. 2021 skenování portů. Doména je umístěna na IP adrese 89. 221. 213.132 a DNS záznamy jsou umístěny na Wedosu. Nezajímavé porty (celkem 997) nebyly vůbec zobrazeny, ovšem zbývající 3 porty byly objeveny ve stavu otevřeném. Tyto porty je zapotřebí podrobněji oskenovat a zjistit co se na nich nachází.

Obrázek 8 Zajímavé porty na doméně sweet-soft.eu



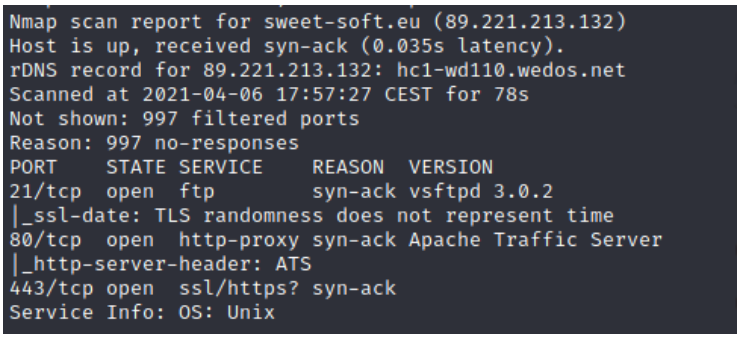
Zdroj: Autorka práce

Podrobné skenování je možné pomocí příkazu *nmap -vv -A sweet-soft.eu*, který odhalil více informací o zajímavých portech – viz obrázek 9. Na portu 21/tcp se nachází server ftp s licencí GPL pro UNIX systémy, včetně Linuxu. Tento server se vyznačuje tím, že je stabilní, rychlý a bezpečný. Verze vsftpd-3.0.2 byla vydaná v září 2019.

Port 80/tcp patří HTTP, který využívá Apache Traffic Server. Software Apache Traffic Server je rozšiřitelný, škálovatelný a rychlý proxy server, sloužící pro ukládání do mezipaměti. Kompatibilní je s protokoly http/1.1 a http/3.

Posledním zajímavým portem je 443/tcp, což je standartní port pro přenos HTTPS. Tento protokol zajišťuje, že poskytovatel internetového připojení (případně kdokoliv jiný v síti) nemůže manipulovat nebo číst konverzaci, probíhající mezi prohlížečem a serverem. V zásadě chrání veškeré citlivé transakce a poskytuje určitou úroveň ochrany osobních údajů. Jakmile prohlížeč naváže připojení HTTPS, odešle se TCP požadavek přes port 443. Pokud je však připojení navázáno, jsou data aplikace (zpráva vyměňovaná mezi serverem a klientem) šifrována. Kromě skutečné informace si může útočník přečíst informaci o IP adrese, velikosti zprávy, web, kterému je připojení navázáno nebo frekvence připojení. Je důležité zdůraznit, že použitím portu 443 HTTPS se neposkytuje anonymní procházení.

Obrázek 9 Detailnější skenování zajímavých portů



Zdroj: Autorka práce

8.3 Exploitace a skenování

Jakmile jsou nasbírané informace z veřejně dostupných zdrojů, tak přichází na řadu samostatné testování aplikace. Testování bude probíhat za využití nástroje ZAP v Kali linux.

ZAP

Open-source aplikace není tak datově náročná jako komerční aplikace. E-shop bylo možné otestovat automatickým skenem, který využívá jak aktivního prohledání protokolu, tak i pasivního. Oproti manuálnímu prohledávání je méně časově náročné a postačí pouze vložit odkaz url adresy a zahájit testování. Nástroj ZAP se pak postará o veškeré prohledávání. Celkově automatické penetrační testování trvalo něco málo přes 3 hodiny. Všechny odhalené výstrahy (celkem 17) jsou zobrazeny na obrázku 10. Blíže rozepsané jsou pouze ty, které nebyly objeveny při testování platformy shop5.

Obrázek 10 Výstrahy ze ZAP

Zdroj: Autorka práce

Source Code Disclosure – PHP

Útočník může získat na straně serveru zdrojový kód webové aplikace. Ten může obsahovat citlivá data, např. uživatelská jména a hesla, řetězce připojení k databázi atd. Může tím dojít k odhalení interního fungování a obchodní logiky aplikace. S těmito informacemi může útočník zahájit útok, kterým přistoupí k databázi nebo jiným datovým zdrojům. V závislosti na oprávnění účtů získaných ze zdrojového kódu může dojít ke čtení, mazání nebo aktualizaci dat z databáze.

Cookie No HttpOnly Flag

Příznak HttpOnly je zahrnutý v hlavičce odpovědi HTTP set-cookie. Použitím příznaku HttpOnly při generování souboru cookie, je možné zmírnit riziko přístupu scriptu na straně klienta k chráněnému souboru cookie (podporuje-li to prohlížeč). Pokud je příznak obsažený v hlavičce odpovědi HTTP, k souboru cookie nelze získat přístup prostřednictvím scriptu na straně klienta. V důsledku toho, i když je odhalena chyba skriptování mezi weby (XSS) a uživatel náhodně přistupuje k odkazu, který tuto chybu zneužívá, prohlížeč nezobrazí soubor cookie třetí straně. Nepodporuje-li prohlížeč HttpOnly a web se pokusí nastavit cookie soubor, bude příznak ignorován prohlížečem. Tím se vytvoří klasický soubor cookie přístupný scriptu. V důsledku toho se stane zranitelným vůči krádeži škodlivým skriptem.

X-Content-Type-Options Header Missing

Tato hlavička byla zavedena společností Microsoft jako způsob, kterým mohou webmasteři blokovat sledování obsahu. Testeři očekávají, že tato hlavička bude nastavena. Slouží ke zmírnění bezpečnostních rizik tím, že brání „hádání“ MIME typu souboru. Pokud se vloží do stránky odkaz na skript *<script type=“text/javascript“ src=“testScript.php“></script>*, dynamicky se generuje (koncovka souboru je .php) a nenastaví se správná Content-Type HTTP hlavička (ta, která odpovídá MIME typu JS souborů), ale zůstane jako výchozí text/html, prohlížeč JavaScript vykoná. Pokud se přidá HTTP hlavička X-Content-Type_Options: nosniff, Google Chrome a Internet Explorer skript zablokují, protože není nastavený MIME typ text/javascript, ale text/html.

Feature Policy Header Not Set

Zásady funkcí umožňují vývojářům webových aplikací selektivně povolit, upravit a zakázat chování určitých API rozhraní a webových funkcí v prohlížeči. Pomocí toho lze např.:

- změnit výchozí chování mobilních videí a videí třetích stran - *.autoplay*
- umožnit rámcům iframe používat API rozhraní - .fullscreen
- zablokovat použití zastaralých API rozhraní, jako je *.document.write()*

Jednoduše lze zablokovat veškerý obsah používaný rozhraní API pro geografickou správu na webu odesláním seznamu povolených položek pro tuto funkci – *Feature-Policy: geolocation ‚none'*.

Information Disclosure - Suspicious Comments

Zveřejněné komentáře nebo komentované fragmenty zdrojového kódu mohou útočníkovi pomoct s porozuměním základní logiky webové aplikace a najít funkční, zastaralé koncové body. Hacker může o aplikaci shromažďovat další informace tak, že se naučí fragmenty zdrojového kódu, které byly komentovány. Nefunkční logika zabezpečení či stále funkčních, ale nepoužívaných koncových bodů, které mohou vracet soukromá data nebo interní informace o společnosti (vnitřní struktura sítě, osobní jména vývojářů).

Storable and Cacheable Content

Cache-control je hlavička HTTP, která se používá k určení zásad ukládání prohlížeče do mezipaměti v odpovědích serveru i klientských požadavcích. Zásady zahrnují způsob ukládání prostředku do mezipaměti, kde je uložen do jeho vypršení. Např. „cache-control: public, max-age=360“, kde „max-age“ je maximální čas, po který je prostředek uložený v mezipaměti, definovaný v sekundách. Z příkladu tedy vyplývá, že vrácený prostředek je platný po dobu 360 sekund, po které musí prohlížeč požádat o novější verzi.

8.4 Report

Při testování Open-source platformy Wordpress byly odhaleny 2 závažné výstrahy, které jsou na seznamu nejzávažnějších zranitelností projektu OWASP. Výstrahy Path Traversal a SQL Injection byly detailně popsány v penetračním testování platformy Shop5. Wordpress nabízí několik pluginů, které těmto zranitelnostem dokážou předejít. Pluginy jsou jak zdarma, tak za poplatek.

Open-source platforma v základní verzi neodolala nejzávažnějším zranitelnostem, které jsou v projektu OWASP a pro jakékoliv používání na internetu je vhodné nainstalovat minimální zabezpečení. Veškeré bezplatné pluginy je zapotřebí stahovat pouze z oficiální stránky www.wordpress.org. Vyhledávač dostupných pluginů se nachází i v administraci. Ty placené lze pak stahovat pouze z webu výrobce nebo ověřených zdrojů, např. www.woocommerce.com. Zakoupit pluginy je možné i z neověřených zdrojů, ale dochází zde k velkému risku (krádež citlivých údajů, špatná funkčnost atd).

9 Diskuse výsledků a zhodnocení

Základní porovnání výsledků penetračního testování komerční platformy shop5 a open-source platformy Wordpress.

1) Při testovaní platformy shop5.cz byl již e-shop pro testování nainstalovaný a připravený na doméně *http://jana.testsXXX.XX* od majitele webové aplikace. Oproti tomu Wordpress bylo zapotřebí si nainstalovat. K instalaci bylo potřeba si koupit doménu a hosting. Aby bylo možné využít Wordpress k online obchodování musel se nainstalovat plugin WooCommerce. Open-source platforma byla tedy časově náročnější na instalaci než komerční řešení, u kterého se řeší pouze doména (u pronájmu není potřeba hosting).
2) V obou případech testování se jednalo o interní penetrační testování, protože dochází k simulaci útoku od běžného uživatele z vnitřní sítě, který dostal přístup do vnitřních firemních systémů. Ani u jedné platformy nebylo dokázáno, že by při testování unikly citlivé informace.
3) Nástroj Uniscan objevil u obou webových aplikací celkem 3 zajímavé porty. Zatímco v případě shop5 se jednalo o porty 80/tcp, 4000/tcp a 4001/tcp, tak u Wordpressu se jednalo o porty 21/tcp, 80/tcp a 443/tcp. Obě aplikace tedy využívají nezabezpečeného protokolu HTTP (80/tcp), což pro penetrační testování nepředstavuje žádný problém. V případě domény www.sweet-soft.eu, která byla využitá pro open-source aplikaci, je možné požádat hosting o zabezpečený protokol HTTPS (443/tcp). U komerční platformy v případě ostré verze je tento protokol zajišťovaný automaticky.
4) Penetrační testování probíhalo pomocí nástroje ZAP. Zatímco u open-source aplikace bylo možné spuštění automatického testování, tak u komerčního řešení se využilo manuální testování. Důvodem bylo, že platforma shop5 byla velmi náročná na celkové automatické testování a po několika hodinách došlo k zastavení nástroje. Automatické testování u Wordpressu trvalo celkem 3 hodiny. Manuální testování shop5 probíhalo několik dnů.
5) Výsledek testování ukázal, že u komerčního řešení se odhalilo 31 výstrah, zatímco u open-source 17 výstrah. Celkový přehled výstrah je zobrazen v tabulce 4. V kapitole 6 je uvedeno, že celkově se u platformy shop5 zobrazilo 31 výstrah. V tabulce jich je pouze

28, protože první čtyři (Advanced SQL Injection) byly shodné a lze je tedy považovat za jednu výstrahu.

Tabulka 4 Celkové výstrahy z nástroje ZAP

Výstrahy	**Komerční platforma**	**Open-source platforma**
závažné	2	2
středně závažné	5	3
méně závažné	11	6
informační	7	6
falešně pozitivní	3	0
celkem	**28**	**17**

Zdroj: Autorka práce

6) Závažné výstrahy jsou u každé platformy rozdílné. U shop5 se ukázalo, že je možné riziko v Cross Site Scripting a Advanced SQL Injection. U Wordpressu to byly Path Traversal a SQL Injection. Ačkoli se tyto výstrahy projevily i u komerční webové aplikace, tak v detailnějším zkoumání došlo při těchto útocích k zablokování IP adresy na straně e-shopu. Je tedy zřejmé, že platforma shop5 má již v základu integrovanou bezpečnostní funkci. Funkce „automatický BLACKLIST“ slouží tak, že v případě rozpoznání hrozby, zablokuje konkrétní IP adresu. V tu chvíli se pro zablokovanou IP adresu stane e-shop neviditelný.

7) Středně závažné výstrahy jsou u obou platforem v základu shodné. Open-source aplikace měla celkově 3 méně závažné, zatímco komerční aplikace celkem 6. U komerční platformy jsou však navíc Appliacation Error Disclosure, Parametr Tampering .htaccess Information Leak. Poslední zmíněná výstraha byla detekovaná jako falešně pozitivní, protože po tomto útoku došlo k umístění IP adresy na BLACKLIST. Shodné výstrahy jsou Source Code Disclosure, Sub Resource Integrity Attribute Missing a X-Frame-Options Header Not Set.

8) Méně závažné výstrahy se opět ve větším měřítku ukázaly u platformy shop5, kde bylo aplikací ZAP zaznamenáno 11 výstrah. Oproti tomu u Wordpress se objevilo 6 výstrah, které jsou společné s aplikací shop5. Společné výstrahy jsou Absence of Anti-CSRF Tokens, Cookie No HttpOnly Flag, Cookie Without SameSite Attribute, Dangerous JS Functions, Feature Policy Header Not Set a X-Content-Type-Options Header Missing.

Komerční aplikace je o dost náročnější. Je to možné vidět na tom, kolikrát byla celkově zaznamenána výstraha Absence of Anti-CSRF Tokens. U open-source to bylo 259krát a u komerční dokonce 1090krát, což je opravdu velký rozdíl. Zbývajících 5 výstrah, které byly zaznamenány u shop5 jsou Cookie without Secure Flag, Cross-Domain JavaScript Source File Inclusion, Incomplete or No Cache-control and Pragma http Header Set, Private IP Disclose, Server Leaks Information via „X-Powerered-By“ http Responce Header Field(s).

9) Informační výstrahy slouží zejména jako informativní. Neměly by se však nijak podcenit. Prvních 6 výstrah je pro obě platformy společných, ale u shop5 je zajímavé zjištění výstrahy Username Hash Found. Při detailnějším zkoumání nástroj ZAP odhalil, jaké je použité šifrování pro heslo, a dokonce detekoval konkrétní heslo, které bylo nastaveno. Tohle by neměla žádná aplikace objevit. Do vyhledávače stačí pouze napsat *Md5 21232f297a57a5a743894a0e4a801fc3* a už se zobrazí, co se po tímto hash skrývá – viz obrázek 11. Heslo do administrace je tedy „admin“.

Obrázek 11 Dešifrování hesla na internetu

Zdroj: www.google.com

10 Návrh na další zabezpečení

Dle provedených penetračních testů je vhodné každou platformu lépe zabezpečit, protože bylo prokázáno, že je možné provést nejzávažnější útoky uvedené v projektu OWASP TOP 10. Každá platforma však bude potřebovat jiné řešení. Základním krokem ke zvýšení bezpečnosti je využívat dostatečně silná hesla nebo náhodně generovaná.

Jakoukoliv webovou aplikaci je vhodné nainstalovat na ověřený hosting, protože v případě hacknutí bude možné obnovit poslední zálohovanou verzi (ftp + databáze). Kvalitní hosting pravidelně aktualizuje zabezpečení serverů a snaží se tak předejít známým hrozbám, jako jsou např. DDoS útoky. Je samozřejmě možné webovou aplikaci umístit i na vlastní server, ale uživatel by měl mít zkušenosti se správou serveru a zejména s jeho zabezpečením. Při doporučení vhodného zabezpečení se počítá s tím, že v dnešní době spousta uživatelů webových aplikací si je instalují na vlastní server, protože tím ušetří desítky až stovky korun za měsíc. Už si však vůbec neuvědomují, jaký to může mít na celou situaci dopad.

10.1 Open-source platforma

Celosvětově nejpoužívanější open-source aplikace Wordpress v základní verzi není zabezpečená natolik, aby odolala útokům skutečného hackera. Wordpress na svých oficiálních webových stránkách uvádí, že zabezpečení je vyvíjeno podle projektu OWASP – viz obrázek 12. Dle testování se dokázalo, že pro základní verzi toto tvrzení není pravdivé. Použití vhodných pluginů se zabezpečení webové aplikace určitě povede zvýšit. I když je na obrázku vidět, že se jedná o OWASP top 10 z roku 2013 a v této práci se používá OWASP top 10 2017, tak odhalené vysoce nebezpečné výstrahy jsou stejné.

Obrázek 12: OWASP Top 10 na Wordpress

Zdroj: https://cs.wordpress.org/about/security/

UpdraftPlus

K plánování záloh a úložného prostoru je možné využít velmi populární plugin UpdraftPlus. Plugin je komplexní a intuitivní na ovládání. Oproti jiným nabízí obnovu a zálohu, umožňuje nastavení automatických záloh v pravidelném čase a předností je, že testování proběhlo na více než 3 miliónech webech. Zálohování lze nastavit přímo na Google Disk, Dropbox, Amazon S3, FTP atd. V případě placené verze lze zálohovat na Microsoft Onedrive, Google Cloud Storage, SFTP a další. Oproti verzi zdarma je doplněná o přírůstkové zálohy, migraci webových stánek nebo jejich duplikaci, šifrování databáze nebo bezplatná odborná podpora. Placená verze se pohybuje od 70 $ do 399 $ za rok. Ceny se liší v závislosti na počtu licencí.

Wordfence

Světově nejoblíbenější bezpečnostní skener a firewall. Plugin obsahuje skener malware a firewall bránu koncového bodu, který byl od základu vymyšlený pro Wordpress. K dispozici je verze zdarma nebo placená. Firewall webových aplikací se využívá k blokaci a identifikaci škodlivého provozu. Oproti cloudovým řešením neporušuje šifrování a tím tedy nemohou unikat data. Zajišťuje i ochranu proti brute force omezením pokusů pro přihlášení. Placená verze je doplněna o aktualizaci podpisu malware za pomocí kanálu Threat Defense nebo o seznam blokovaných IP adres. Bezpečnostní skener kontroluje známe chyby zabezpečení a upozorňuje v případě nedostatku. Upozornění přijde i v případě nebezpečných url adres, podezřelého obsahu, zastaralého nebo zrušeného pluginu. V reálném čase lze sledovat pokusy o hackování, přístupy a další. Ceny se pohybují od 99 $ do 74,25 $ za rok, v závislosti na počtu licencí.

Securi Security

Jedná se o celosvětově uznávanou autoritu v záležitostech souvisejících se zabezpečením webových aplikací. Bezpečnostní sada je určena k doplnění stávajících zabezpečení. Plugin zajišťuje monitorování integrity souborů, vzdálené skenování malware, bezpečnostní zákrok po hacknutí, bezpečnostní oznámení a další. Cena se pohybuje od 199,99 $ - 499,99 $ za rok v závislosti na požadovaných funkcích.

iThemes Security

Bezpečnostní plugin, který pomáhá uživatelům Wordpress se zastaralým software nebo slabými hesly. Zvyšuje zabezpečení díky dvoufázovému ověřování, přidání soli k heslu, generování silných hesel, vypršení platnosti hesla pro uživatele, google reCAPTCHA, nastavení importu / exportu nebo online srovnávač souborů. Lze spravovat několik webů najednou. Zakáže problémové uživatele, roboty a další hostitele. Vynutí SSL pro stránky správce a pro jakoukoliv stránku nebo příspěvek. Zjišťuje skryté chyby 404, které mohou nepříjemně ovlivnit SEO a spoustu dalších užitečných funkcí. Cena za tento plugin je 749 $ a nejsou tam žádná omezení.

Automatická aktualizace pluginů

Použitím výše zmíněných pluginů bude přicházet upozornění v případě, že bude již zastaralý plugin. Aby však docházelo k pravidelné aktualizaci, tak je možné připsat 2 řádky do souboru „wp-config.php“ a tím bude docházet k pravidelné aktualizaci témat a pluginu na platformě Wordpress.

```
add_filter('auto_update_plugin', '__return_true');
add_filter('auto_update_theme', '__return_true');
```

Zabránění editovat kód pluginů a témat

V případě několika správců webové aplikace je vhodné zabránit těmto uživatelům v editování kódů. Pokud by některá z těchto osob měla plný přístup, tak může dojít k hacknutí nebo odcizení citlivých údajů. Do souboru *wp-config.php* se pouze připíše jeden řádek.

```
define('DISALLOW_FILE_EDIT', true);
```

.htaccess

Soubor .htaccess slouží k tomu, aby si uživatel webových stránek sám mohl upravit některé vlastnosti na serveru, aniž by o to musel žádat správce. V souboru lze nastavit přesměrování (přes 301), nastavení jiného výchozího souboru místo *index.php* a také ke skrytí souborů (nejčastěji těch konfiguračních). U Wordpress se konfigurační souboru nazývá *wp-config.php*. Jednoduchým příkazem lze tento soubor skrýt pro uživatele.

<files wp-config.php>

order allow, deny

deny from all

</files>

Pomocí příkazu *Options All – Indexes* lze ukrýt jednotlivé adresáře webu.

CLOUDFLARE

Globální síť navržená tak, aby všechno, co se připojuje k internetu bylo bezpečné, soukromé, spolehlivé a rychlé. Plugin se používá zejména pro ochranu proti DDoS útokům. Wordpress nenabízí nativní zabezpečení před tímto typem útoku. Funguje jako „síto“, přes které teče veškeré spojení vedoucí na webovou aplikaci. Služba umí zachytit nežádoucí pakety a tím odvrátit DDoS útok. V případě naprogramovaných funkcí na míru, služba umožňuje otestování funkcí na míru. Ceny se pohybují od 20 $ do 200 $ za měsíc.

Výše zmíněné pluginy jsou v celosvětovém žebříčku nejvyhledávanější a nejpoužívanější pro platformu Wordpress. Celkové cenové zhodnocení pro zabezpečení open-source systému je zobrazeno v tabulce 5. Dát v přepočtu cca 29 216,16 Kč / rok (počítáno kurzem 21,51 Kč) je sice vysoká částka, ale všechno záleží na tom, kolik pro každého znamená zabezpečení systému. Pro e-shopy, které mají zisk nad 1 mil ročně se toto zabezpečení určitě vyplatí.

Tabulka 5 Celkové cenové zhodnocení za bezpečnostní pluginy

Název pluginu	**cena od ($)**
UpdraftPlus	70,00
Wordfence	99,00
Securi Security	199,99
iThemes Security	749,00
Cloudeflare	240,00
celkem	**1 357,99**

Zdroj: Autorka práce

10.2 Komerční platforma

U komerční platformy zodpovídá za zabezpečení webové aplikace její majitel (není-li ve VOP uvedeno jinak). Shop5 má svého správce serveru, který se stará o jeho bezpečný a nepřerušovaný chod. Jelikož systém je vyvíjený již 17 rokem a jediným programátorem je stále Bc. Martin Březovský, byly očekávány nějaké možné útoky. Dříve totiž nebylo zapotřebí řešit tolik možných útoků na webovou aplikaci, jako je tomu nyní. Doporučení na další zabezpečení bude pouze pro vybrané výstrahy.

Advance SQL Injection

U vstupu uživatele je zapotřebí použít „whitelist“ povolených znaků nebo „blacklist“ nepovolených znaků. Shop5 má již funkci BLACKLIST připravenou, ale bude ji potřeba doplnit, aby se zabránilo tomuto typu útoku.

Jedná se o příkazy SQL, které databázový server odesílá a analyzuje odděleně od všech parametrů. Tímto způsobem se zamezí útočníkovi vložit škodlivý SQL dotaz. V případě využívání MySQLi a PHP jsou 2 možnosti, jak toho dosáhnout:

1) CHOP

```
$x = $pdo->prepare('SELECT * FROM tabulka WHERE jmeno = :jmeno');
$x->exec(array('jmeno' => $jmeno));
foreach ($x as $radek){
        //něco udělat s $radek
}
```

Třída CHOP v PHP pomocí funkce *$pdo->prepare* - připraví příkaz k provedení a vrátí objekt příkazu. Funkce *$x->exec* spustí příkaz SQL a vrátí počet ovlivněných řádků. Pomocí *foreach* pak programátor určí, co se má s každým řádkem stát.

2) MySQLi

```
$x = $dbConnection->prepare("SELECT * FROM tabulka WHERE jmeno ='name'");
$x->bind_param('s', $name);
$x->execute();
$vysledek = $x->get_result();
While ($radek = $vysledek->fetch_assoc()){
        //něco udělat s $radek
}
```

V prvním řádku je připravený SQL dotaz. Funkce *$x->bind_param('s', $name)* váže parametry na SQL dotaz a řekne databázi, jaké jsou parametry. Argument "s" udává, že se jedná o řetězec. Příkaz *$x->execute()* provede připravený příkaz a v následujícím řádku dochází k získání sady výsledků z připraveného příkazu. Dokud je podmínka splněná, tak se spouští požadovaný kód.

Cross Site Scripting

K neperzistentnímu XSS typu útoku nedochází při načtení webové aplikace nebo stránky, jako tomu je u perzistentních XSS. Uživatel software se stává obětí, kliknutím na pohybné URI (Uniform Resource Identifer), např. z emailu. Z pohledu webové aplikace zranitelné místo představuje zpětné zobrazování vloženého vstupu. Exite několik pravidel, které je zapotřebí dodržovat, aby se předešlo XSS útokům:

1) kódování HTML – kódování entit HTML, aby nedošlo ke spuštění skriptu, sstyle nebo nějaké události. Nahradit "&" za & "<" za &alt a další znaky
2) kódování JavaScriptu – týká se dymanicky generovaného kódu JS. Př. kódování bloku JS.

```
<script type="text/javascript">
var msg = "<%= Encode.forJSBlock(neduveryhodne)%>";
alert(msg);
</script>
```

11 Závěr

Penetrační testy jsou neoddělitelnou součástí testování software, buď už při jeho vývoji, uvedení na trh, při aktualizacích nebo během bezpečnostních auditů organizace. Pro efektivní provedení penetračních testů je nutné dobře odhadnout cíl, tedy jak daná webová aplikace pracuje, jaké má funkce a kde může mít slabé stránky.

Úvod diplomové práce je věnován platformám webových aplikací, které je možné rozdělit do několika řešení (komerční, open-source, na míru). Každé řešení má své výhody a nevýhody. I když je možné vložit na internet takřka vše, tak webové aplikace podléhají legislativním a normativním předpisům. Podle normy ISO/IEC 27001 se uděluji certifikace a systém řízení bezpečnosti informací se zavádí dle normy ISO/IEC 27002. Získáním certifikátu se společnost prokazuje tím, že její software je dostatečně zabezpečený. Již od roku 2018 je platný zákon z Evropské Unie o ochraně osobních údajů (GDPR) a definuje jakým způsobem je možné nakládat s osobními údaji. Tato povinnost se vztahuje na každého, kdo jakkoliv pracuje s osobními údaji.

Teoretické principy testování software s důrazem na penetrační testy byly vysvětleny formou literární rešerše v teoretické části této diplomové práce. Práce testerů vyžaduje velmi dobré technické znalosti, zkušenosti a techniky pro efektivní otestování aplikace. Neexistuje univerzální přístup k penetračním testům, protože každá aplikace nebo systém vyžaduje individuální posouzení. Oproti tomu je možné vytvořit všeobecnou metodiku k provedení penetračního testu jako je prezentované v praktické části. Totožná metodika byla provedena na obě testované aplikace.

Cílem uskutečněného penetračního testování bylo ověření zabezpečení komerční aplikace a open-source aplikace. Výsledky z obou testování porovnat a zjistit, kterou platformu je lepší si pro tvorbu e-shopu vybrat. Otestování komerční platformy je možné pouze se souhlasem majitele, a ten byl získán u platformy shop5. Wordpress byl ihned jasnou volbou pro porovnávání, protože je celosvětově nejpoužívanější a byl na něm postavený e-shop na dálniční známky za cca 310 mil korun. Základ pro testování byl seznam běžných zranitelností v projektu OWASP TOP 10. Testování proběhlo kombinací manuálního a automatizovaného testování pomocí nástroje uniscan a ZAP. Výsledky byly zpracované formou reportu, který zahrnuje odhalené zranitelnosti a doporučené opatření k nápravě bezpečnostních slabin.

V obou testovaných platformách byly odhaleny nejzávažnější výstrahy. Zatímco u komerčního řešení došlo k vyřešení během 1 dne majitelem platformy, tak u open-source řešení není jasně dané. Můžeme jen zkoušet, jaký plugin zabrání těmto zranitelnostem. Velké překvapení nastalo u platformy shop5, kde bylo odhaleno použité šifrování pro heslo, a dokonce konkrétní heslo. Správným vložením výrazu do vyhledávače se okamžitě rozklíčovalo dané heslo a tím bylo možné se dostat do administrace e-shopu.

I když se tedy zdá, že Wordpress je zdarma a celkově se tedy vyplatí na něm provozovat e-shop, tak to neplatí z pohledu bezpečnosti. U komerční platformy bylo odhaleno více výstrah, protože však majitel zná perfektně svůj systém, tak nebyl žádný velký problém v tom, aby díky tomuto reportu došlo během krátké doby k řádnému zabezpečení. Náklady za bezpečnost nese v tomto případě majitel platformy. V případě Wordpress za to má zodpovědnost majitel e-shopu, který ve většině případech je pouhým uživatelem a nikoliv programátorem. Je tedy vhodné, aby ten, kdo vůbec nerozumím těmto technickým věcem, využíval komerční řešení než open-source.

S ohledem na rychlý technologický pokrok v oblasti IT se vynořují stále nová rizika pro zabezpečení webových aplikací, které testeři při realizaci penetračních testů musí odhalit. Právě proto je téma penetračních testů a testování bezpečnosti IT systémů velmi aktuální a představuje potenciál pro další výzkum.

12 Seznam použitých zdrojů

1. **Luciano Manelli, Giulio Zambon.** *Eshop Application.* místo neznámé : Apress, Berkeley, CA, 2020. 978-1-4842-5866-8.

2. **Mgr. Marek Doleček.** Elektronický obchod. *businessinfo.cz.* [Online] https://www.businessinfo.cz/cs/clanky/elektronicky-obchod-ppbi-51052.html#!&chapter=1.

3. **Fuggetta, Alfonso.** www.sciencedirect.com. [Online] 15. Duben 2003. https://www.sciencedirect.com/science/article/abs/pii/S0164121202000651.

4. **Bc. Martin Březovský.** Rozdíl mezi pronajatým a odkoupeným e-shopem. *shop5.cz.* [Online] https://www.shop5.cz/rozdil-mezi-koupenym-a-pronajatym-eshopem.html.

5. **B. Arkin, S. Stender, G. McGraw.** ieeexplore.ieee.org. [Online] 14. únor 2005. https://ieeexplore.ieee.org/abstract/document/1392709.

6. **Matthew Denis, Carlos Zena, Thaier Hayajneh.** ieeexplore.ieee.org. [Online] 20. Duben 2016. https://ieeexplore.ieee.org/abstract/document/7494156. 978-1-4673-8490-2.

7. **Selecký, Matúš.** *Penetrační testy a exploitace.* Brno : Computer Press , 2012. ISBN 978-80-251-3752-9.

8. **Shivani Acharya, Vidhi Pandya.** www.ijecse.org. [Online] 2012. https://d1wqtxts1xzle7.cloudfront.net/52391978/Gray-box_testing.pdf?1490918749=&response-content-disposition=inline%3B+filename%3DBridge_between_Black_Box_and_White_Box_G.pdf&Expires=1620586777&Signature=LIVpFBUv-9tfAbImINa~v4Js4s4VGDGcFxBRlTQOVis4nxJJvPF. ISSN 2277-1956.

9. **Humbhreys, Edward.** *Implementing the ISO/IEC 27001 ISMS standard.* Norwood : Artech house, 2016. IBS 13: 978-1-60807-930-8.

10. **Disterer, Georg.** www.scirp.org. [Online] Duben 2013. https://serwiss.bib.hs-hannover.de/frontdoor/deliver/index/docId/938/file/ISOIEC_27000_27001_and_27002_for_Information_Security_Management.pdf.

11. **journals.sagepub.com.** journals.sagepub.com. [Online] 1. Listopad 2017. https://journals.sagepub.com/doi/abs/10.2501/IJMR-2017-050?journalCode=mrea.

12. **Nezmar, Luděk.** *GDPR praktický průvodce implementací.* Praha : GRADA Publishing a.s., 2017. ISBN 978-80-271-0668-4.

13. **I. van Ooijen, Helena U. Vrabec.** link.springer.com. [Online] 11. Prosinec 2019. https://link.springer.com/article/10.1007/s10603-018-9399-7.

14. **shoptet.** www.shoptet.cz. [Online] [Citace: 20. Říjen 2020.] https://podpora.shoptet.cz/hc/cs/?_ga=2.158458173.666566073.1620551873-1737209958.1620551873.

15. Shoptet. www.shoptet.cz. [Online] [Citace: 20. Říjen 2021.] https://www.shoptet.cz/jak-zacit/.

16. **wordpress.** wordpress.org. [Online] [Citace: 21. Říjen 2020.] https://wordpress.org/about/.

17. **Brazell, Aaron.** *WordPress Bible.* místo neznámé : Wiley, 2011. ISBN-13: 978-0470937815.

18. **Castaño, Arnaldo Pérez.** *PrestaShop Recipes.* místo neznámé : Apress, Berkeley, CA, 2017. ISBN 978-1-4842-2574-5.

19. **Li, Jinfeng.** papers.ssrn.com. [Online] 1. Červenec 2020. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3570446. ISSN: 2516-029X.

20. **Bach-Nutman, Matthew.** arxiv.org. [Online] 17. Prosinec 2020. https://arxiv.org/ftp/arxiv/papers/2012/2012.09960.pdf.

21. **Suqi Liu, Lawrence K. Saul, Geoffrey M. Voelker, Stefan Savage, Ian Foster.** dl.acm.org. *Who is .com?: Learning to Parse WHOIS Records.* [Online] Říjen 2015. https://dl.acm.org/doi/abs/10.1145/2815675.2815693.

22. **Yuma Makino, Vitaly Klyuev.** ieeexplore.ieee.org. [Online] 3. Září 2015. https://ieeexplore.ieee.org/abstract/document/7340766. 978-1-4673-8361-5.

23. **Gilberto Najera-Gutierrez, Juned Ahmed Ansari.** *Web Penetration Testing with Kali Linux.* místo neznámé : Packt Publishing Ltd, 2018. ISBN 978-1-78862-337-7.

24. **shoptet.** *Doplňky doplnky.shoptet.cz.* [Online] https://doplnky.shoptet.cz/.

25. **Březovský, Bc. Martin.** Podpora www.shop5.cz. [Online] https://www.shop5.cz/videa.html.

26. Bc. Martin Březovský—. www.shop5.cz. [Online] https://www.shop5.cz/cenik.html.

27. **shoptet.** Tarify www.shoptet.cz. [Online] [Citace: 21. Říjen 2020.] https://www.shoptet.cz/cenik/.

Přílohy

Příloha 1: Report z aplikace ZAP – jana.testsXXX.XX

ZAP Scanning Report

Summary of Alerts

Generated on ne, 28 bře 2021 11:39:48

Risk Level	Number of Alerts
High	6
Medium	9
Low	20
Informational	16

Alerts

Name	Risk Level	Number of Instances
Advanced SQL Injection - AND boolean-based blind - WHERE or HAVING clause	High	1
Advanced SQL Injection - MySQL >= 5.0.12 AND time-based blind (SELECT)	High	1
Advanced SQL Injection - Oracle AND time-based blind	High	2
Cross Site Scripting (Reflected)	High	1
Path Traversal	High	3
SQL Injection	High	7
.htaccess Information Leak	Medium	41
Application Error Disclosure	Medium	5
Parameter Tampering	Medium	6
Source Code Disclosure - SQL	Medium	71
Sub Resource Integrity Attribute Missing	Medium	3
X-Frame-Options Header Not Set	Medium	181
Absence of Anti-CSRF Tokens	Low	1093
Cookie No HttpOnly Flag	Low	613
Cookie Without SameSite Attribute	Low	19
Cookie Without Secure Flag	Low	18
Cross-Domain JavaScript Source File Inclusion	Low	1
Dangerous JS Functions	Low	16
Feature Policy Header Not Set	Low	241
Incomplete or No Cache-control and Pragma HTTP Header Set	Low	8
Private IP Disclosure	Low	92
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	189
X-Content-Type-Options Header Missing	Low	310
Base64 Disclosure	Informational	148
Information Disclosure - Suspicious Comments	Informational	134
Non-Storable Content	Informational	181
Storable and Cacheable Content	Informational	159

Alert Detail

High (Medium)	Advanced SQL Injection - AND boolean-based blind - WHERE or HAVING clause
Description	A SQL injection may be possible using the attached payload
URL	http://jana.tests XXX.XX ndex.php?sekce=kategorie&nadrazena=0&id=0+AND+4527%3D1964--+Ratt&md_1616922249=1616922249.1521
Method	GET
Parameter	id
Attack	0 AND 3584=3584-- BWnC
Instances	1
Solution	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input. Apply the privilege of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application.
Other information	The page results were successfully manipulated using the boolean conditions [0 AND 3584=3584-- BWnC] and [0 AND 4527=1964-- Ratt] The parameter value being modified was stripped from the HTML output for the purposes of the comparison. Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter.
Reference	https://www.owasp.org/index.php/Top_10_2010-A1 https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
CWE Id	89
WASC Id	19
Source ID	1

Medium (High)	Sub Resource Integrity Attribute Missing
Description	The integrity attribute is missing on a script or link tag served by an external server. The integrity tag prevents an attacker who have gained access to this server from injecting a malicious content.
URL	https://www.gstatic.com/charts/49/js/jsapi_compiled_ui_module.js
Method	GET
Evidence	<script type="text/javascript" src="https://www.google.com/jsapi">
Instances	1
Solution	Provide a valid integrity attribute to the tag.
Reference	https://developer.mozilla.org/en/docs/Web/Security/Subresource_Integrity
CWE Id	16
WASC Id	15
Source ID	3

Low (Medium)	**Cross-Domain JavaScript Source File Inclusion**
Description	The page includes one or more script files from a third-party domain.
URL	http://jana.tests XXX.XX js_connector.php
Method	POST
Parameter	https://www.google.com/jsapi
Evidence	<script src="https://www.google.com/jsapi"></script>
Instances	1
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Reference	
CWE Id	829
WASC Id	15
Source ID	3

Příloha 2: Report z aplikace ZAP – sweet-soft.eu

ZAP Scanning Report

Summary of Alerts **Generated on ne, 11 dub 2021 14:35:30**

Risk Level	Number of Alerts
High	2
Medium	3
Low	6
Informational	6

Alerts

Name	Risk Level	Number of Instances
Path Traversal	High	3
SQL Injection	High	1
Source Code Disclosure - PHP	Medium	1
Sub Resource Integrity Attribute Missing	Medium	172
X-Frame-Options Header Not Set	Medium	89
Absence of Anti-CSRF Tokens	Low	259
Cookie No HttpOnly Flag	Low	65
Cookie Without SameSite Attribute	Low	66
Dangerous JS Functions	Low	24
Feature Policy Header Not Set	Low	136
X-Content-Type-Options Header Missing	Low	194
Base64 Disclosure	Informational	4
Charset Mismatch	Informational	9
Information Disclosure - Suspicious Comments	Informational	127
Storable and Cacheable Content	Informational	170
Storable but Non-Cacheable Content	Informational	92
Timestamp Disclosure - Unix	Informational	27

High (Medium)	SQL Injection
Description	SQL injection may be possible.
URL	http://sweet-soft.eu/?p=9-2
Method	GET
Parameter	p
Attack	9-2
Instances	1
Solution	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them, Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input. Apply the principle of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application.
Other information	The original page results were successfully replicated using the expression [9-2] as the parameter value The parameter value being modified was stripped from the HTML output for the purposes of the comparison
Reference	https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
CWE Id	89
WASC Id	19
Source ID	1

Medium (High)	Sub Resource Integrity Attribute Missing
Description	The integrity attribute is missing on a script or link tag served by an external server. The integrity tag prevents an attacker who have gained access to this server from injecting a malicious content.
URL	http://sweet-soft.eu/obchod/?add-to-cart=1059
Method	GET
Evidence	<link rel='stylesheet' id='niva-store-fonts-css' href='https://fonts.googleapis.com/css?family=Open+Sans%3A300%2C400%2C600%2C700&subset=cyrillic%2Ccyrillic-ext' type='text/css' media='all' />
URL	http://sweet-soft.eu/kategorie-produktu/obleceni/
Method	GET
Evidence	<link rel='stylesheet' id='niva-store-fonts-css' href='https://fonts.googleapis.com/css?family=Open+Sans%3A300%2C400%2C600%2C700&subset=cyrillic%2Ccyrillic-ext' type='text/css' media='all' />
URL	http://sweet-soft.eu/produkt/kostkovana-sukne-vero-moda/?add-to-cart=1059
Method	GET
Evidence	<link rel="profile" href="https://gmpg.org/xfn/11">

Instances	172
Solution	Provide a valid integrity attribute to the tag.
Reference	https://developer.mozilla.org/en/docs/Web/Security/Subresource_Integrity
CWE Id	16
WASC Id	15
Source ID	3

Informational (Medium) Base64 Disclosure

Description	Base64 encoded data was disclosed by the application/web server. Note: in the interests of performance not all base64 strings in the response were analyzed individually, the entire response should be looked at by the analyst/security team/developer(s).
URL	http://sweet-soft.eu/wp-content/plugins/woocommerce/assets/js/selectWoo/selectWoo.full.min.js?ver=1.0.6
Method	GET
Evidence	select2/data/minimumInputLength
URL	http://sweet-soft.eu/wp-includes/css/wp-embed-template-ie.min.css?ver=5.7
Method	GET
Evidence	iVBORw0KGgoAAAANSUhEUgAAABQAAAAUCAQAAAAngNWGAAAAcElEQVR4AdXRVxmEMBAGwJMQCUhA/hKQECmRsFJwMFfp7HfP/E8pk0173CuKpt/0R+WaBaaZqogLagBMuh+DdoKbyRCwqZ/SnM0R5oQuZ2UHS8Z6k23qPxZCTrV5U/HMi8bsfHVXP7K/GXZHaTO7S54CWLdH/N2Y/wAAAABJRU5ErkJggg==

Instances	4
Solution	Manually confirm that the Base64 data does not leak sensitive information, and that the data cannot be aggregated/used to exploit other vulnerabilities.
Other information	±é^rÝżu«Zth§Ŝk¦"zn´·§,Ř
Reference	http://projects.webappsec.org/w/page/13246936/Information%20Leakage
CWE Id	200
WASC Id	13
Source ID	3