FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

# ZADÁNÍ DIPLOMOVÉ PRÁCE

| | |
|---|---|
| **Název:** | BOZP portál III |
| **Student:** | Bc. Karolína Solanská |
| **Vedoucí:** | Ing. Jiří Chludil |
| **Studijní program:** | Informatika |
| **Studijní obor:** | Webové a softwarové inženýrství |
| **Katedra:** | Katedra softwarového inženýrství |
| **Platnost zadání:** | Do konce letního semestru 2018/19 |

## Pokyny pro vypracování

1. Analyzujte aktuální řešení portálu pro správu školení a testování BOZP na FIT ČVUT.
2. Ve spolupráci s pracovníky BOZP identifikujte stávající nedostatky systému.
3. Zrevidujte informační systém z pohledu nakládání s osobními daty uživatelů. Ve vztahu k GDPR (General Data Protection Regulation) vyhodnoťte rizika a navrhněte konkrétní bezpečnostní opatření.
4. Na základě výsledků penetration testů a etického hackování analyzujte stav zabezpečení portálu s ohledem na možné právní a finanční důsledky.
5. Do vývojového procesu BOZP portálu zaveďte průběžnou integraci pomocí verzovacího systému a zhodnoťte možné časové nároky. Do portálu také přidejte automatickou synchronizaci dat se systémem MOODLE.
6. Implementujte navržené změny a funkcionality do portálu BOZP.
7. Zhodnoťte finanční a časovou náročnost zavedení GDPR do projektu tohoto typu.
8. Zhodnoťte finanční a časový přínos průběžné integrace.

## Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 30. listopadu 2017

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

# BOZP portál III

***Bc. Karolína Solanská***

Katedra softwarového inženýrství
Vedoucí práce: Ing. Jiří Chludil

27. dubna 2018

# Poděkování

Chtěla bych poděkovat své rodině a svému příteli za podporu v průběhu mých studií a psaní této diplomové práce, svému vedoucímu Ing. Jiřímu Chludilovi za cenné rady a pečlivé vedení a svým kolegům a nadřízeným za podporu k dokončení studií.

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 27. dubna 2018 .....

### Odkaz na tuto práci

Solanská, Karolína. *BOZP portál III*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

# Abstrakt

Tato diplomová práce se zabývá analýzou stávajícího portálu BOZP. Tento portál slouží k evidenci a správě školení studentů, pedagogů a zaměstnanců ČVUT FIT. Analýza portálu je provedena především z pohledu funkčnosti a zabezpečení stávající aplikace. Z této analýzy jsou pak vyvozena doporučení a návrhy úprav. Tyto úpravy rozšiřují stávající funkcionality aplikace a vylepšují celkové zabezpečení aplikace a to jak z pohledu samotného kódu, tak i nasazení. Zabezpečení je navrženo tak, aby odpovídalo standardům používaným v praxi. Práce se také zabývá integrací všech komponent, které dosud nebyly přidány. Práce pak také analyzuje dopad směrnice GDPR na práci s uživatelskými daty a bezpečností aplikace a to jak z ekonomického a manažerského hlediska, tak i z pohledu technologického zabezpečení. Výstupem práce je upravená aplikace, její dokumentace a zhodnocení finanční a časové náročnosti. K dalšímu rozvoji aplikace pak může sloužit soubor doporučení, který je také součástí výstupu.

**Klíčová slova** BOZP,školení,GDPR,bezpečnost,osobní data,aplikace

# Abstract

This thesis is concerned by analysis of BOZP web portal. This web portal server for recording and administration of trainings of students, teachers and employees of CTU FIT. Analysis is focusing primarily on functionality and security of current application. This analysis results into recommendations and modifications. These modifications expand current functionalities of the application and improve overall security of the app from the point of view of the code itself and of the deployment. Security is designed to meet current standards. Thesis also covers integration of all components that weren't added to the application yet. Thesis also analyses impact of GDPR directive on work with user data and security of application from the point of economical and management view and as a technology. Thesis output is modified application, documentation and evaluation of financial and time impact. For next development of the application there is a set of recommendations, that is also part of the output.

**Keywords** BOSP,training,GDPR,security,personal data,application

# Obsah

# Seznam obrázků

# Seznam tabulek

# Úvod

Pojem „školení BOZP“ se nevyskytuje v žádném právním předpisu. Přesto je ale hojně používán mezi veřejností i v odborných kruzích v oblasti bezpečnosti práce. Toto slovní spojení se používá jako zkratka pro Školení bezpečnosti a ochrany zdraví při práci. Patří mezi nejzákladnější školení, která každý zaměstnavatel poskytuje svým zaměstnancům. V případě ČVUT Fakulty informačních technologií jej poskytuje škola svým studentům.
Kromě školení BOZP je také FIT povinen poskytnout studentům školení požární ochrany (PO) a také školení vyhlášky 50/78Sb paragrafu 4 o elektromontážích a revizích.
Pro účel zjednodušení administrace okolo těchto školení a usnadnění přihlašování na ně je na fakultě zřízen a již čtvrtý rok provozován portál BOZP. Jeho návrhem, implementací a dodatečnými úpravami se již zabývalo několik závěrečných prací, je tedy nutné celý koncept portálu sjednotit a upravit tak, aby vše fungovalo jak má a s portálem se uživatelům dobře pracovalo.
Zároveň byly v loňském roce uskutečněny bezpečnostní testy metodou etického hackování, které odhalily některé nedostatky v zabezpečení portálu. Je tedy třeba výstupy z těchto testů analyzovat, navrhnout jejich řešení a to do portálu implementovat.
V neposlední řadě bude v roce 2018 uveden v platnost zákon Evropské unie GDPR (General Data Protection Regulation), který se zabývá ochranou osobních údajů. Je třeba analyzovat povinnosti, které z tohoto zákona vyplynou a opět k nim navrhnout řešení a implementovat je.
Je důležité analyzovat také finanční a právní důsledky, které by mohly nastat pokud by portál nebyl dostatečně zabezpečen a nesplňoval legislativní požadavky.
Vzhledem k mnohým změnám a pravděpodobnému rozšiřování portálu v blízké budoucnosti se zavedení průběžné integrace do vývoje jeví jako užitečný krok, který pomůže zrychlit testování i nasazení portálu.

Kapitola 1

# Cíl práce

Cílem této práce je analyzovat BOZP portál na Fakultě Informačních Technologií po stránce funkční i uživatelské a identifikovat jeho stávající nedostatky. Tato analýza proběhne s pomocí pracovníků školení BOZP, kteří tento portál využívají ke své práci.
V další část analýzy stávajícího stavu portálu bude zkoumat výsledky etického hackování. Zde je třeba se zaměřit především na opravu stávajících nedostatků na poli zabezpečení aplikace. Je třeba kromě bezpečnostních problémů samotné aplikace také analyzovat zabezpečení celého webového serveru a použít k tomu správné nástroje.
Zbytek analytické části je pak analýza povinností vyplývajících z GDPR - General Protection Data Regulation, tedy směrnice Evropské unie, kterou se musí od května řídit všechny aplikace, které se zabývají zpracováním osobních dat.
Cílem této rozsáhlé analýzy je objevit a pojmenovat nedostatky portálu na všech frontách, tedy v oblasti funkcionalit, zabezpečení i nasazení. Následně je cílem práce navrhnout řešení, které by nejlépe pomohlo vyřešit objevené problémy aplikace.
V části implementace je pak cílem tyto nedostatky opravit podle navržených řešení a otestovat jejich vhodnost. Výstupem práce by pak měla být plně funkční a zabezpečená aplikace odpovídající moderním standardům a právním normám.

Kapitola 2

# Analýza stávající situace

V této kapitole se zabývám analýzou stávajícího řešení školení na FIT ČVUT, tedy portálem BOZP. V první části bych ráda shrnula momentální stav portálu. Následně se budu zabývat analýzou testování pomocí etického hackování, které na portálu proběhlo v minulém roce. V poslední části první kapitoly bude krátké shrnutí nové evropské směrnice GDPR, která se zabývá ochranou osobních údajů a jíž bude nutné portál přizpůsobit.

## 2.1 BOZP portál

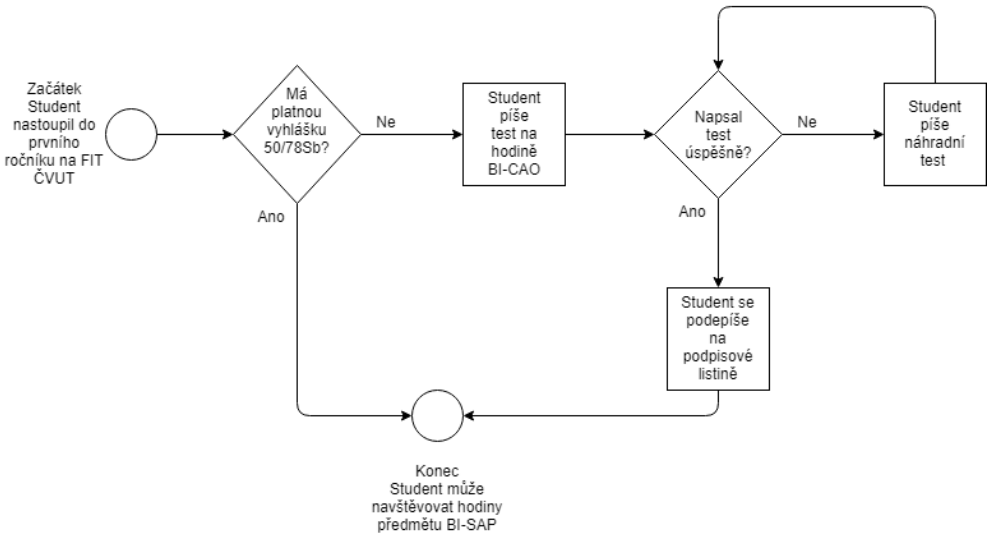
V této části první kapitoly se budu zabývat podrobnou analýzou aktuálního stavu BOZP portálu. Portál byl nasazen do ostrého běhu v roce 2013. Za pět let jeho provozu se ukázaly některé nedostatky původního řešení. Bylo také třeba přidat další moduly, se kterými původní návrh kompletně nepočítal.
Těmito úpravami se zabývalo již deset závěrečných prací několika studentů, z nichž dvě byly týmové.
V roce 2012 tým pod vedením Štefana Pinďáka[3] vytvořil aplikaci BOZP. Hlavní podíl práce na jejich modulech měli Martin Humeník s prací pokrývající téma správy školení[4] a Lukáš Jeschke s modulem e-learningu[5].
O rok později na ně navázali Kamil Falta a Dominik Jančík s pracemi zabývajícími se evidencí přístrojů[6][7]. Zároveň s nimi probíhalo testování portálu v rámci bakalářské práce Jiřího Kopeckého[8] a vývoj modulu pro správu Martina Náhlovského[9].
Kamil Falta se BOZP portálem zabýval i ve své diplomové práci, kdy doplnil modul pro správu školení[10]. Stejně tak Martin Náhlovský, který vylepšil modul e-learningu[11]. Jako poslední se ve své bakalářské práci portálem zabývala Hana Kozáková, která implementovala nové funkcionality do modulu pro správu školení[12].

Z předchozího seznamu bývalých studentů je jasné, že portál BOZP byl mno-

hokrát měněn a vylepšován. Byly zde přidávány moduly a přepisovány ty staré. Veškeré modifikace je třeba zhodnotit z pohledu celkové architektury a funkčnosti.
V této analytické části své práce budu navazovat na poslední dvě závěrečné práce a jejich výstupy.

### 2.1.1 Životní cyklus školení na portálu BOZP

Pracovníci BOZP mohou pomocí portálu vytvářet, editovat a mazat školení pro studenty a zaměstnance fakulty. Parametry školení jsou například typ školení, kapacita, termín uzávěrky a podobě.
Po přihlášení na BOZP portál pak uživatelé mohou vidět plánovaná školení na obrazovce Dostupné akce, kde se na ně také mohou přihlásit nebo odhlásit. Před školením si pak pracovník BOZP vygeneruje podpisovou listinu se seznamem přihlášených účastníků školení, kam se všichni účastníci školení podepíší. Následně může pověřený pracovník BOZP v sekci Správa školení jednotlivým účastníkům školení udělit. Podpisová listina je archivována.[12]
Existuje více různých druhů školení, které mají různé cykly. Například pro splnění školení vyhlášky 50/78Sb paragrafu 4 je třeba navíc splnit test, který studenti píší v rámci předmětu BI-CAO v prvním semestru prvního ročníku. Bez splnění tohoto testu pak nelze tento předmět absolvovat.
Test prověřující znalosti nabyté na tomto školení píší studenti v systému moodle, který je pro skládání testů přímo určený. Data odtud pak může pracovník BOZP exportovat a importovat je do portálu BOZP, kde se výsledky následně automaticky uloží jako záznam k jednotlivým studentům. Z portálu BOZP je pak možné vygenerovat podpisovou listinu. K podpisu této listiny se musí dostavit všichni studenti, kteří test úspěšně splnili.
Pokud je student u testu neúspěšný, musí se dostavit na náhradní termín, který je vypisován hromadně mimo běžné vyučovací hodiny. Zde pak test píše opakovaně.
Pokud již student má splněné školení i test z vyhlášky 50/78Sb, například ze střední školy, z minulých studií, nebo od zaměstnavatele, nemusí se účastnit školení ani testu a vše je mu uznáno.
Na následujícím diagramu můžeme tento proces vidět jasně ucelený.




Obrázek 2.1: Procesní diagram školení vyhlášky 50/78Sb

### 2.1.2 Analýza aktuálního řešení

Vzhledem k tomu, že podrobnou analýzou se zabývaly již předchozí práce a jejich nasazením se změnily jen některé implementační detaily, uvádím pouze krátké shrnutí aktuálního stavu BOZP portálu.

**Databázový model** Datová struktura modulu pro správu školení portálu BOZP je navržena jednoduše, přehledně a tak, aby na ni bylo možno v budoucnu navázat. Atributy tabulek i vazby jsou pojmenovány výstižně a stručně. Jako zdroj dat je používáno fakultní KOS REST api.
Databáze sestává celkem z 22 tabulek. Ukládají se do ní jak data, tak také nastavení aplikace a to konkrétně složky pro různé ukládání dat, které si může administrátor sám nastavit.
Databázově jsou také definované role uživatelů používané v systému. Tuto variantu framework Nette nevyžaduje, je ale přehlednější a vzhledem k tomu, že se na projektu podílelo mnoho vývojářů, je to užitečné.
Podrobný popis databáze včetně podrobného relačního modelu je možné najít v práci Kamila Falty[6]. Tento model také přikládám v elektronické podobě k této práci.

**Uživatelské role** Uživatelské role jsou v systému BOZP použity k rozlišení, které obrazovky smí uživatel zobrazit a které ne. Role, které mohou uživatelé mít jsou: správce, supervizor, zaměstnanec a student. Jediná role správce má plná práva k zobrazení všech funkcí portálu. Ostatní role mají práva omezená.

**Použité technologie** BOZP portál běží v PHP frameworku Nette verze 2.3.0.[13] Pro práci s databází je aktuálně použita knihovna Nette Database (NDBT)[14] jako další jsou využívány dvě různé knihovny pro zobrazení tabulkových gridů a to Nextras Datagrid[15] a Nette Grido[16]. Použití dvou různých typů datagridů má původ v historii a kontinuálním vývoji aplikace, protože Nextras Datagrid byl běžně používán především ve starších verzích aplikací v Nette. Aplikace také používá knihovnu PDFResponse pro generování PDF dokumentů.
Celá aplikace běží nad MySQL databází[17], což je doporučený databázový systém pro aplikace v Nette.
Na serveru na kterém aplikace běží je nainstalovaný linuxový operační systém Debian verze 6.0[18], což je poměrně starý systém - aktuální verze Debianu je 9[19]. Jako webový server je zde použit standardně Apache[20] verze 2.2.16, která je také velmi zastaralá a má několik známých zranitelností[21]. Pro správný běh Nette je tu pak nainstalováno PHP verze 5.4.45[22].

**Import dat** V modulu pro správu školení existuje možnost importovat data o studentech a vyučujících do své databáze s využitím KOS API. Tuto akci lze vyvolat ručně a oprávnění na její provedení má pouze správce portálu. Správce systému pak může automaticky importované údaje zkontrolovat a aplikovat. Stejný import funguje i pro přednášky, cvičeních a studenty na nich zapsaných. Import dat je poněkud komplikovaný co se týká chronologie. Data v KOS API totiž nemusí být nutně zadána ve správném pořadí. Problém nastává pokud byl například student nejprve zapsán do magisterského studia a teprve po té byl odebrán ze studia bakalářského. KOS API pak hlásí pouze poslední stav, přestože samo bere v úvahu všechny změny. Do portálu BOZP se pak již dostává pouze zpráva, že student ukončil bakalářské studium.

**Správa školení a přihlašování na ně** Správa školení je reprezentována jako interaktivní databázový výstup v podobě datagridu. Běžní uživatelé pak mohou v portálu vidět dostupné akce a u nich vždy možnost přihlásit se na konkrétní termín, případně se z něj odhlásit anebo si prohlédnout jeho detailní informace.

**Podpisové listiny** Samostatnou sekcí BOZP portálu jsou pak podpisové listiny, které je možné podle nahraného vzoru vygenerovat do formátu PDF. Na této vygenerované listině je pak seznam účastníků školení a místo pro jejich podpisy.
Podpisové listiny jsou plně lokalizovatelné, mohou být vydány v jakémkoliv jazyce, aktuálně je k dispozici čeština a angličtina. Jejich úpravami se zabývala především práce Hany Kozákové[12]. Ukázka podpisové listiny je přiložena k elektronické verzi této práce.

**Statistiky** Pro každý typ školení si může administrátor portálu v obrazovce statistik zobrazit tabulku, kde je shrnut pro každý typ studia počet proškolených uživatelů, přihlášených účastníků přihlášených na školení a procentuální statistiky proškolených uživatelů vůči neproškoleným.
V sekci statistik je pak možné rozsáhle filtrovat všechny skupiny studentů i zaměstnanců. To pak usnadňuje práci při shromažďování údajů. Je zde možné vidět tabulku studijních programů a jim přiřazených studentů. Tito se pak dají dále seřadit podle studijního roku v daném programu. Statistiky také rozlišují jestli je student z Fakulty Informačních Technologií, nebo z jiné fakulty a v jakém je ročníku.

## 2.2 Etické hackování a bezpečnost

V letním semestru školního roku 2016/17 proběhl v rámci předmětu BI-EHA (Etické hackování) kompletní rozbor. Tato analýza byla zpracovávána dvěma týmy studentů tohoto předmětu. Studenti se zabývali především zranitelností portálu z pohledu možnosti proniknutí a změny dat uvnitř.
Vzhledem k ne příliš uspokojivým výsledkům (co se bezpečnosti týká) bylo následně rozhodnuto, že je třeba přikročit k systematickému řešení zabezpečení portálu. Ochrana citlivých a osobních údajů uživatelů musí být implementována a nasazena také kvůli nové směrnici EU GDPR - General Data Protection Regulation, která začíná platit od května 2018.
V následující části shrnu jakým způsobem testy zranitelnosti probíhaly a krátce vysvětlím, co která metoda znamená. Ze závěrečné zprávy studentů není bohužel možné přesně citovat, protože není zveřejněna a podléhá NDA (Dohodě o mlčenlivosti), kterou autoři zprávy podepsali.

### 2.2.1 Účel testování

Testování probíhalo od poloviny března do června roku 2017. Pro toto testování použili studenti lokální kopii webové aplikace, neboť není možné tyto testy provozovat nad živými daty. K tomuto opatření bylo přistoupeno z důvodu bezpečnosti a také proto, že během etického hackování by mohlo dojít k nenávratnému poškození dat nebo databáze.
Účelem pokusu bylo vykonat kompletní testování zranitelnosti webové aplikace BOZP. Vzhledem k tomu, že studenti dostali k dipozici zdrojový kód, měli také možnost zkontrolovat systém zevnitř z pohledu programátora a identifikovat jeho slabiny.
Cílem celého testování pak bylo toto:

- Najít nějaká citlivá data
- Objevit způsob jak editovat data, ke kterým nemá být běžně přístup
- Najít mezery v zabezpečení

- Otestovat schopnost systému reagovat na hrozbu

### 2.2.2 Použité nástroje

V následující části se zabývám popisem a rozborem způsobů testování, které byly použity při etickém hackování v letním semestru 2017.

#### 2.2.2.1 Aktivní a pasivní průzkum - Reconnaissance

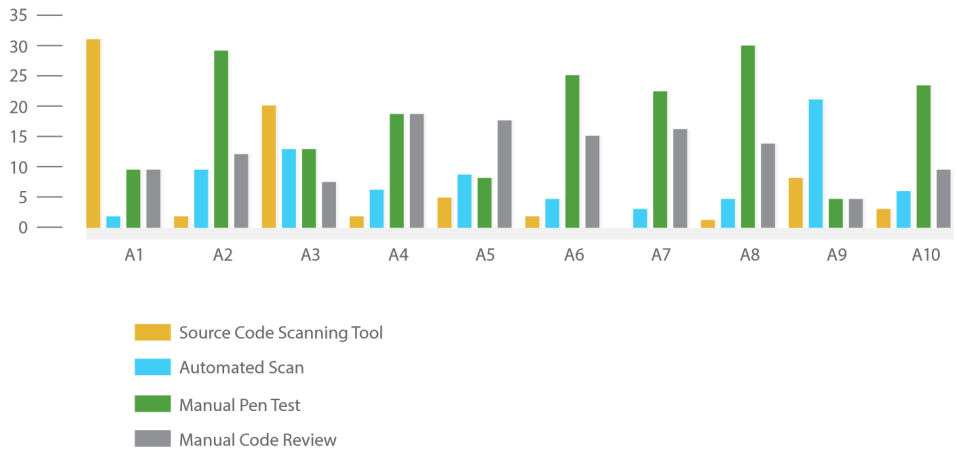
V etickém hackování patří prvotní průzkum k základním metodám zkoumání zabezpečení aplikací. Anglický výraz „Reconnaissance“ je první fází každého hackerského útoku na aplikaci. Hacker se pokouší získat tolik informací, kolik jen může o cíli svého útoku. A to především identifikaci cíle, zjištění rozsahu jeho IP adres, sítě, DNS záznamy, nebo také verzi systému a informace o jeho správcích. Obvykle je to také fáze nejdelší.[23]
Metody průzkumu jsou různé od jednoduchého vyhledávání na internetu, přes sociální inženýrství a prohledávání popelnic až po nenásilné skenování sítě.
Vzhledem k povaze těchto aktivit se proti nim velmi špatně brání. Informace o organizaci jsou na internetu dohledatelné a obvykle ani nebývá žádoucí je skrývat. Uživatelé systému i jeho správci o sobě na internetu zanechávají stopu, ze které se po čase dají vytěžit velmi hodnotné a kompletní informace. Z těchto informací se následně dá utvořit poměrně přesný obrázek o společnosti kde pracují, o procesech, které jsou tam uplatňovány i o případných slabých místech.[24]
Nicméně existují opatření, která mohou hackerům zabránit v proniknutí k systému, nebo alespoň velmi ztížit sběr informací. Mezi tato opatření patří[23]:

- Ujištění se, že ze systémů a aplikací neunikají žádné informace na internet, mimo jiné
  - Verze softwaru a patch levely
  - Emailové adresy
  - Jména a pozice klíčových zaměstnanců
- Ujištění se, že papírové dokumenty jsou pečlivě skartovány
- Poskytování generických informací registrátorům domén
- Zabránění LAN/WAN zařízením odpovídat na pokusy o skenování

#### 2.2.2.2 Kontrola kódu

Vzhledem k tomu, že autoři zprávy měli k dispozici celý zdrojový kód aplikace, měli tak možnost projít si její kód a najít případné bezpečnostní problémy.
Komplexní kontrola kódu není vyloženě standardní metodou etického hackování, obvykle se provádí na novém softwaru před uvedením do produkce.

Nicméně ke komplexní analýze zranitelnosti aplikace nepochybně patří.
Bezpečnostní kontroly kódu, neboli security code reviews, se mohou lišit především úrovní formality. Kontrolou je pozvání kolegy, aby se na kód podíval ale stejně tak i to, když se někdo velmi snaží objevit slabost. Ve větších společnostech mají pro kontrolu bezpečnosti kódu obvykle zřízený samostatný tým, kde pracuje několik lidí s různými znalostmi a kvalifikacemi. Obvykle jsou také zavedeny standardizované metriky měření kvality kódu. Bezpečnostní kontrola kódu je pravděpodobně tou nejefektivnější technikou identifikace bezpečnostních problémů a chyb v softwaru.[1]
Zlaté pravidlo kontroly kódu je, že kód, který prošel security code review, by následně měl projít i testy penetrace a to buď úplně bez problémů, nebo jen s menšími chybami.
Dle výzkumu sdružení OWASP (Open Web Application Security Project) můžeme na následujícím grafu vidět, jak jsou důležité jednotlivé fáze testování kódu pro odhalení různých druhů bezpečnostních problémů.




Obrázek 2.2: Výzkum spojující jednotlivé způsoby kontroly kódu s odhalením různých typů zranitelností[1]

**Zranitelnosti kódu podle OWASP** Na grafu 5.2 můžeme vidět různé druhy zranitelnosti kódu a metody, kterými je nejsnadněji odhalíme. Tyto zranitelnosti řadí OWASP do jednotlivých kategorií:

- **A1: Injection**
  Útoky pomocí injection dovolují útočníkům téměř doslova vpichovat obsah a příkazy do běžící aplikace a měnit tak její chování. Tento typ útoků

je běžný a široce rozšířený a umožňuje útočníkovi lehce zjistit zranitelná místa aplikace. Velmi často se používají proti starším aplikacím, které nebyly po delší dobu updatované[1].
Pokud se mluví o Injection útocích jde obvykle o SQL Injection. Takovéto útoky probíhají tak, že na vstupu programu místo obyčejného stringu SQL kód, který je následně serverem nechtěně spuštěn. Základní způsob tohoto druhu napadení je přímé vložení kódu do uživatelského vstupu. Méně přímá verze je pak postupné vkládání škodlivého kódu do vstupů, které se postupně zapisují do tabulky nebo jako metadata a spustí se až v momentě, kdy jsou jednotlivá políčka spojena do dynamického SQL příkazu.[25]
Jednou z nejefektivnějších metod obrany proti tomuto typu útoků je důsledná validace vstupu na principu „podezřelý dokud není prověřen".[1]

- **A2: Nespolehlivá autentifikace uživatele a session management**
  Autentifikace je důležitá především proto, že je hlavním vstupem k chráněným funkcionalitám a datům. Vzhledem k tomu, že není možné kontrolovat jak uživatel nakládá se svými přihlašovacími údaji, je třeba ujistit se, že bez řádného přihlášení nelze v aplikaci tyto chráněné funkcionality využívat. Nejčastější formou autentifikace je heslo, dále například klientské certifikáty, biometrické údaje, jednorázová hesla nebo přihlašovací frameworky, jako OAUTH nebo SSO.[1]
  Obvyklou obranou proti zranitelnosti autentifikace je například vynucení komplexity hesla, zamknutí účtu po několika špatných přihlášeních, Captcha, přihlašovací formulář pouze přes TLS (například HTTPS) nebo více faktorové ověření (MFA).
  MFA obvykle obsahuje alespoň dvě ze tří základních ověřovacích metod[1].

  - Něco, co uživatel zná (detaily účtu a heslo)
  - Něco, co uživatel má (bezpečnostní token, certifikát, nebo mobilní číslo)
  - Něco, co uživatel je (biometrika)

- **A3: Cross-site scripting** (XSS)
  Tento typ útoků je do jisté míry podobný Injection útokům. Do jinak validního a důvěryhodného webu jsou vloženy útočné skripty obvykle ve formě skriptu běžícího v prohlížeči. Chyby, které těmto útokům napomáhají jsou poměrně rozšířené. Je to například používání dat vložených uživatelem mezi kódem bez validace.
  Útočník může XSS použít k poslání skriptu nic netušícímu uživateli. Jeho prohlížeč skript spustí, protože nemá jak ho rozlišit od důvěryhodných skriptů. Následně pak útočník získá přístup ke cookies, session tokenům nebo k jiným citlivým informacím uchovávaným prohlížečem.[26]
  Bezpečnou obranou proti XSS je, stejně jako u Injection, důsledná kont-

rola vstupu, „escapování“ dat v případě jejich vkládání do skriptů a nasazení Content Security Policy[27].

- **A4: Insecure Direct Object Reference**
  Velmi běžným porušením bezpečnosti u aplikací, které poskytují různé úrovně přístupu (nepřihlášený, přihlášený uživatel, přihlášený administrátor, ...) je insecure direct object reference.
  Uživatel může být autorizován k přístupu ke stránce, ale už ne k určitému objektu v databázi. Typickým příkladem je profil jiného uživatele. Dostat se k nim dá pomocí manipulace s URL adresou.
  Jedinou opravdu účinnou obranou je pečlivá a důsledná autorizace všech akcí a kontrola toho, kdo je provádí. Toho lze dosáhnout například kontrolou rolí[1].

- **A5: Špatně nastavené zabezpečení (Security misconfiguration)**
  Mnoho moderních aplikací je vyvinuto nad frameworky, které se starají o mnoho činností, které už následně programátor nemusí řešit. Typickým příkladem je počítač na kterém běží Apache server. Tyto aplikace bývají jen zřídkakdy spouštěny v izolovaném prostředí. Naopak obvykle jich běží na jednom počítači více, popřípadě spolu spolupracují.
  Všeobecná ochrana pomocí dalších síťových prvků je velké téma, ale netýká se kontroly kódu, není v OWASP manuálu tedy příliš rozvinutá[1].

- **A6: Vystavení citlivých dat**
  Nedostatečná ochrana citlivých dat uživatelů je velkým problémem mnoha webových stránek. Ať už se jedná čísla kreditních karet, rodná čísla nebo přihlašovací údaje. Útočník, který tato data získá pak může snadno ukrást uživateli identitu, nebo zneužít jeho platební údaje. S blížícím se zákonem o GDPR a začátkem jeho platnosti v celé Evropské unii je toto téma aktuálnější než dříve.[1]
  Obrana proti krádeži citlivých dat se týká dvou úrovní:

  - **Ochrana při přenosu**
    Zpravidla se používá SSL/TSL vrstva pro šifrování dat putujících přes HTTP protokol, může se také jednat o FTPS. Veškerá rozhodnutí ohledně přenosu by měla být učiněna důkladně informovaným člověkem.
  - **Ochrana při skladování**
    To zahrnuje šifrování kreditních karet v databázi, hashování hesel a používání MAC (message authentication code) pro znemožnění modifikace zprávy mezi dvěma počítači.
    K šifrování dat v databázi je možné použít různé typy a druhy šifer. Mezi nejběžnější patří solení, hashing nebo PKI (Public-key Encryption)[1].

- **A7: Špatná kontrola access levelu (ACL - úrovně přístupu)**
  Aplikace, které poskytují přístup k nějaké funkcionalitě ji velmi často zpřístupňují uživatelům až po autentifikaci. Málokteré ale poté kontrolují, zda má uživatel skutečné právo k provádění daných akcí.
  Autorizace prováděné akce je stejně důležitá jako autentifikace uživatele. Aplikace by vždy měla kontrolovat, zda data, ke kterým uživatel přistupuje jsou opravu jeho a zda k nim má mít přístup. V moderních aplikacích je tento problém zpravidla vyřešen přidělením role uživateli po prvním přihlášení. Systém pak následně při každé jeho akci kontroluje, zda na ni má uživatel podle své role právo.
  Důležité je také autorizovat samotný bod, ze kterého se mohou uživatelé přihlašovat a co nejvíce ho omezit. V případě, že autorizace selže, systém by neměl uživateli dovolit v zadané akci pokračovat. Zlatým pravidlem je také to, že každá stránka by měla v základu odmítat přístup všem a následně jej pouze povolit příslušným rolím.

- **A8: Cross-site request Forgery (CSRF)**
  Tento typ útoku donutí koncového uživatele spustit několik nechtěných akcí na webové aplikaci, kde je aktuálně přihlášený. Nejde zde obvykle o krádež dat, protože útočník nemá šanci zachytit odpověď serveru. S pomocí sociálního inženýrství může své oběti odeslat odkaz emailem, nebo přes chat. Po kliknutí na tento odkaz je provedeno několik akcí automaticky v prohlížeči daného uživatele. Může tak být donucen například převést peníze, změnit si svou emailovou adresu a podobně. Pokud je oběť navíc administrátorem systému, může ohrozit celou aplikaci.[28]
  Účinnou obranou proti tomuto útoku je kontrola standardních hlaviček requestu pro ověření, že mají stejný původ a kontrola CSFR tokenu.[29]
  Další možností je použití takzvané Challenge-response. Přestože je to velmi silný nástroj, zásadně ovlivňuje prožitek uživatele a je proto doporučován především u aplikací s kritickými daty, jako je například bankovnictví. Challenge-response obranou je například CAPTCHA, jednorázový token nebo vynucení opakovaného přihlášení.[1]

- **A9: Používání komponent se známou zranitelností**
  Mnoho aplikací používá při svém běhu komponenty, nebo knihovny, které jsou buď open-source, nebo placené. Každopádně jsou to však komponenty vyvinuté třetí stranou. Použití cizích knihoven a modulů dává smysl, jsou již vyvinuty, fungují a byly obvykle dobře otestovány. Nicméně tento přístup má i své nevýhody a to především v oblasti bezpečnosti.
  Pokud je v nějaké knihovně nebo komponentě objevena bezpečnostní chyba, ocitne se velmi brzy pod drobnohledem, protože útočník ví, že tato chyba bude fungovat u kohokoliv, kdo tuto knihovnu používá.
  V rámci kontroly kódu se tyto zranitelnosti v komponentách objevují

jen velmi těžko. Nicméně by používání komponent třetích stran mělo být v organizaci pečlivě sledováno, aby mohli bezpečnostní experti případně rychle reagovat na problémy s jejich zranitelností.[1]

- **A10: Chybějící kontrola přesměrování**
  Webové aplikace přesměrovávají uživatele pravidelně, a to buď v rámci jedné aplikace, nebo i směrem k jiným. Bez důkladné kontroly může ale útočník přesměrování uživatele využít a převést ho na stránku, která používá phishing nebo malware. Neošetřené přesměrování také může pomoci získat přístup k neautorizovanému obsahu.
  Aplikace může bránit chybnému přesměrování tím, že přesměrovává jen na stránky, které má ve svém seznamu povolené, případně používá relativní url, aby se ujistila, že uživatel zůstane na důvěryhodné stránce.

### 2.2.3 OWASP testování

V další části se autoři zprávy etického hackování řídili dokumentem OWASP Web Application Testing Guide[30]. Následuje krátký výčet nejzajímavějších metod, které ke zkoumání použili. Kompletní zprávu lze najít v příloze této práce.

#### 2.2.3.1 Shromažďování informací

Jako první použili autoři zprávy vyhledávací nástroje k dohledání úniku informací ze stránky. Použili pro to Google Hacking Diggity Project, což je projekt umožňující vytěžování informací o vlastním webovém serveru z vyhledávačů Google a Bing. Dle zprávy nebylo touto metodou nalezeno nic, co by jakkoliv ohrožovalo bezpečnost serveru BOZP (například cach konfiguračních souborů).

**Fingerprint web serveru** Web server fingerprinting je kritickým úkolem každého, kdo testuje server kvůli bezpečnosti. Znalost verze a typu běžícího web serveru dává testerům do ruky velkou zbraň v podobě známých zranitelností a příslušných „holých míst“, na která se mohou v průběhu etického hackování zaměřit[31].
Testeři BOZP použili k tomuto účelu program httprint a Netcraft. Oba nástroje vrátily stejný výsledek, a to běžící Apache server verze 2.2.16. Po krátkém hledání je možné bez problémů najít seznam zranitelností této verze systému, která je navíc již zastaralá. Nejnovější bezpečnostní problém této verze byl nalezen v červnu 2017[21].

**Výčet aplikací na webserveru** Po použití programu nmapp bylo testery objeveno mnoho informací, které se dají využít k následnému proražení bezpečnosti serveru. Na různých portech na serveru běží hned několik různých

verzí aplikace BOZP a dalších webových aplikací. Starší verze jsou navíc ještě méně zabezpečené než ta nová, především proto, že nepoužívají k autentifikaci uživatelů Shibboleth. Bežících aplikací na různých portech serveru je celkem devět.

**Identifikace vstupních bodů aplikace** Všechny vstupní body aplikace jsou přehledně shrnuty v přiložené excelové tabulce. Aplikace má celkem 59 vstupních bodů a z toho v době testování bylo zranitelných nějakým útokem celkem devět, což je poměrně vysoké číslo.

#### 2.2.3.2 Testování konfigurace a nasazení

**Zacházení s citlivými daty a soubory** Vstupním bodem k aplikaci je také veřejně přístupný adminer, který slouží k pohodlnějšímu ovládání databáze. Přestože tvrdí, že k němu nelze přistoupit jinak než z lokální sítě, lze se do něj dostat šikovně zvoleným URL.
Také většina složek odmítá přístup, ale některé soubory ho mají volný. Pokud tester uhodne jméno a cestu, lze si tedy otevřít nebo stáhnout všechny soubory s těmito koncovkami: .lock, .json, .js, .ico, .config, .css, .gif, .md, .sql. Ještě větším problémem je však to, že soubory s přípojkou .php mohou být i spuštěny.

**Kontrola starých, zálohových dat** Přístupných dat je na serveru BOZP mnoho. Velká část z nich by být přístupná neměla, a to především:

- Starý dump databáze na adrese: `<https://bozp.fit.cvut.cz/sql/bozp3.sql>`
- Soubor composer, který odhaluje všechny pluginy, které aplikace používá `<https://bozp.fit.cvut.cz/composer.json>`
- Složky s nahranými soubory a obrázky uživatelů - k těmto je třeba znát přesné jméno souboru, nicméně vzhledem k předchozímu máme k dispozici plnou databázi.

#### 2.2.3.3 Testování autentifikace a autorizace

Protože server BOZP používá pro autentifikaci uživatelů SSO systém, není možné zkoumat způsob registrace uživatelů a volbu jejich rolí. Framework Nette samotný je pak poměrně dobře zabezpečen proti neautorizovanému přístupu. Testeři tedy nebyli schopní donutit aplikaci myslet si, že jsou přihlášení, přestože nebyli. Je to zásluha také SSO, která ke každému requestu přidává cookie pro identifikaci přihlášeného uživatele.
Jako další byl testován přístup k funkcím určeným pro jiné role. Vyjma jediného presenteru nebyli testeři schopní k nim přistoupit. Opět to lze přičíst na vrub frameworku Nette a jeho zabezpečení ACL.

Poslední, o co se autoři zprávy pokusili, byla změna rolí, které docílili pouhou změnou několika parametrů v url. Běžný uživatel portálu je tedy schopen si sám změnit roli na administrátora.
Vzhledem k absenci kontroly rolí u některých funkcí si také může uživatel jakékoli role prohlížet a upravovat profily ostatních uživatelů bez větších omezení, pouze uhádnutím správného id.

#### 2.2.3.4 Testování validace vstupu

Jako poslední z testování proběhly testy validace vstupu. Ukázaly, že Nette je poměrně odolné proti XSS (Cross Site Scripting). Naproti tomu je poměrně snadné použít některé stránky pro SQL Injection, včetně smazání celých tabulek.

### 2.2.4 Navrhovaná bezpečnostní opatření

Po testování přirozeně následovaly návrhy, jak opravit největší problémy serveru. Zde jsou návrhy testerů:

- **Skrýt všechny soubory, které nemají být veřejně přístupné**
  Je možné uschovat je do nepřístupné složky, to se týká například souboru composer.json. Dobrou praxí v oblasti bezpečnosti je nemít na serveru nic, co tam není potřebné. Například tedy starý sql dump nemusí být vůbec online.
- **Změnit a otestovat všechna bezpečnostní nastavení**
  Aplikace by měla kontrolovat roli při každé uživatelské akci, a to i při běžném zobrazování souborů. Ty by pak neměly být přístupné všem uživatelům.
- **Pečlivější kontrola při akci s id v url**
- **Nikdo kromě administrátora by neměl mít přístup ke stránce pro změnu role**
- **Anglická přihlašovací obrazovka by měla být buď aktualizována, nebo odstraněna**
- **Rozhraní adminera databáze by nemělo být přístupné ze své složky**

### 2.2.5 Souhrn výsledků etického testování BOZP portálu

V následující tabulce pak můžete vidět přehledně souhrn bezpečnostních problémů serveru BOZP objevených při etickém hackování a jejich kritičnost. Kritičnost je hodnocena na stupnici od 1 do 4, přičemž 4 značí závažný problém a 1 označuje nepříliš velký problém.

| **Závažnost (1-4)** | **Krátký popis** | **Podrobný popis** |
|---|---|---|
| 4 | Uživatel si může změnit roli | K URL <https://bozp.fit.cvut.cz/www/user/change-role/266350> mají přístup všechny uživatelské role, kdokoliv si tedy může svou roli změnit. Formulář sice nabízí pouze roli „Student“, ale při změně id role z 11 (Student) na 2 (Admin) změna rolí proběhne a ze studentského účtu je administrátorský. |
| 4 | Je možné nahrát .php soubor a spustit ho | Toto je jedna z těch nejnebezpečnějších zranitelností. Lze si pomocí toho pustit rozhraní shell, kde je následně možné vidět například všechny skryté soubory a prohlédnout si je. Týká se to také souborů kde jsou přihlašovací údaje. Pomocí tohoto rozhraní pak také není problém změnit některé soubory na serveru nebo celý jeho obsah vymazat. Tento problém je způsoben špatnou validací typu souboru při nahrávání obrázku certifikátu. Pokud je jméno ukončeno středníkem (například image.pgp;), obrázek si ponechá svou původní koncovku. Stále je třeba použít i další, složitější metody, pro to, aby v obrázku byl obsažen kód, nicméně není to složité například s použitím exif data. Největší hrozbou u této bezpečnostní mezery je, že zaměstnanec by mohl omylem spustit PHP skript, který by mohl následně dále poškodit server. |

| | | |
|---|---|---|
| 3 | Problémy presenteru na hlavní stránce | Signál ke smazání není na hlavní stránce blokován kontrolou přístupových práv. Odkaz je schovaný, ale url je velmi předvídatelná <https://localhost/bozp/www/homepage/default/1?do=articleRemove>. Kdokoliv bez práv (i bez přihlášení) může pomocí hádání id článků z databáze vše vymazat. Editace a přidávání článků je zabezpečená. |
| 3 | Problémy uživatelského presenteru | V akci User:profile je v url jasně viditelné id a bez kontroly je změnitelné. Každý, kdo je přihlášený může nejen vidět citlivé informace ostatních uživatelů (soukromý telefon a email) ale zároveň může tyto údaje i změnit. |
| 3 | Lze použít ukradené PHPSESSID | PHPSESSID je jediná kontrola k aktuální session uživatele, která je používaná. Pokud se útočníkovi podaří tuto aktivní session ukrást, není problém stát se uživatelem. |
| 3 | Stránka s editací kurzu nepoužívá anti-CSRF token | Většina portálu je dobře zabezpečená proti CSFR útoku. Jedinou výjimkou je stránka <https://bozp.fit.cvut.cz/www/course/>, která token nepoužívá. |
| 3 | Existuje SQL Injection v některých parametrech orderBy | Do téměř každého parametru této filtrace se dá vložit SQL příkaz. Pomocí této zranitelnosti se dá i úplně vymazat celá databáze. |
| 2 | Problémy presenteru události | Uživatel může ručně změnit id události, při jejím zobrazení, může tedy zobrazit všechny události uložené v databázi. |

| | | |
|---|---|---|
| 2 | Problémy s typem tréninku | Toto není ve skutečnosti problém bezpečnosti, spíše funkčnosti portálu.TrainingType presenter vrací Server Error. |
| 2 | Problémy s přihlašovací stránkou v angličtině | Další problém ve funkcionalitě. |
| 2 | Na serveru jsou zastaralé administrátorské nástroje | Jsou k nalezení na adrese <https://bozp.fit.cvut.cz:8090/>. Většina z nich již pravděpodobně nefunguje. I tak by však mohly být bezpečnostní hrozbou pro portál. |
| 2 | Do nástroje Adminer lze přistupovat z internetu | Pomocí přímého přístupu přes adresu souboru lze přeskočit podmínku přístupu z lokálního počítače k nástroji pro správu databáze Adminer. |
| 2 | Na serveru je záloha databáze přístupná z internetu | <https://bozp.fit.cvut.cz/sql/bozp3.sql> K nalezení tohoto souboru je sice třeba znát jeho přesnou adresu, ale ta je poměrně snadno uhodnutelná. Záloha databáze na stejném serveru, na kterém databáze běží, nedává z hlediska zabezpečení před ztrátou dat smysl. |
| 2 | PHPSESSID nemá bezpečnostní tag | Vzhledem k tomu, že celá stránka běží na HTTPS, by měl být PHPSESSID security tag přítomný. |
| 2 | Skript k vytváření uživatele | Skript spustitelný z příkazové řádky, který vytváří nového uživatele je přístupný z prohlížeče na adrese <https://bozp.fit.cvut.cz/bin/create-user.php>. Dle všeho je již zastaralý a nepoužívaný. |

| | | |
|---|---|---|
| 1 | Na web serveru je možné provést fingerprint | Na web serveru je velmi snadné identifikovat verzi systému pomocí fingerprint nástrojů. |

Tabulka 2.1: Tabulka popisující bezpečnostní hrozby v systému BOZP

## 2.3 GDPR

V následující části se zabývám analýzou GDPR (General data protection regulation) a jeho dopadem na informační systémy, jako takové. Zároveň analyzuji dopad na portál BOZP a údaje uživatelů v něm.

### 2.3.1 Shrnutí

Nařízení Evropského parlamentu a Rady (EU) 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (obecné nařízení o ochraně osobních údajů) známé jako GDPR je v platnosti a účinnost nastává 25. května 2018 po dvouleté periodě určené k přípravě. Nahrazuje tak současný zákon č. 101/2000 Sb., o ochraně osobních údajů[2].
V současné době lze konstatovat, že naprostá většina firem, organizací i státních institucí či samospráv „zaspala“ a tomuto nařízení nevěnovala pozornost. Přitom se nejedná o směrnici, nebo doporučení, ale o nařízení EU platné v rámci všech zemí Evropské unie bez možnosti zásadních úprav. Od data účinnosti tedy může organizace dostat pokutu v plné výši[2].
GDPR stanovuje firmám a organizacím mnohem větší požadavky na zabezpečení osobních dat, kontrolu procesů a výrazně zvyšuje odpovědnost pod vysokými sankcemi až do výše 550 milionů Kč nebo 4 % celosvětového obratu skupiny. Počítá se vyšší sankce. Znalost, jak dodržet předpisy a prakticky aplikovat nařízení za rozumných cenových podmínek se tak stává klíčovou kompetencí[2].

#### 2.3.1.1 Proč je GDPR důležité?

Jako jednotlivci jsme často nuceni předávat osobní informace jako součást online transakcí. Příkladem toho může být rezervace letenky, přístup do bankovnictví nebo komunikace na sociálních sítích. Akceptujeme to jako součást moderního života, neznamená to, že by se uživatelé neměli zajímat, co se stane po té, co jsou data předaná dál. GDPR je navrženo tak, aby měli občané EU větší kontrolu nad svými daty. Základním principem GDPR je, že osobní data „mohou být shromažďována legálně pouze pod striktními podmínkami a z legitimních důvodů“.[32]

#### 2.3.1.2 Kdo se bude muset Obecným nařízením řídit?

Obecným nařízením se bude především, pokud jde o povinnosti, řídit subjekt, který provádí uložení osobních údajů. Takový subjekt se nazývá **správce** osobních údajů. Obecným nařízením se řídí i zpracovatel, což je subjekt, který pro správce osobní údaje zpracovává. Pokud jde o práva, ta vyplývají fyzické osobě, což je subjekt údajů. Dále se obecným nařízením budou řídit i dozorové

úřady, tj. Úřad pro ochranu osobních údajů, který bude uplatňovat svěřené pravomoci za účelem plnění stanovených úkolů.[2]

### 2.3.2 Zásady a principy GDPR

GDPR stanovuje soubor zásad ochrany osobních údajů, podle kterých by organizace měly spravovat osobní údaje. Šest základních zásad GDPR lze nalézt v článku 5 nařízení. V následující části práce se jimi zabývám podrobněji.[33]

#### 2.3.2.1 Zásada 1. - Zákonnost, korektnost a transparentnost

*„Osobní údaje musí být ve vztahu k subjektu údajů zpracovávány korektně a zákonným a transparentním způsobem“*[33].
**Transparentnosti** je dosaženo informováním subjektu, a to ještě před shromažďováním údajů, nebo před případnými následnými změnami. Údaje nemusí být získány jen na přímo od jedinců, ale mohou být také získány z jiných datových souborů, sledováním, nebo pomocí algoritmů. GDPR stanovuje povinný seznam informací, které musí být poskytnuty subjektům údajů, ať už jsou osobní údaje získány přímo, nebo nepřímo. Nutná je poté možnost výběru, což je nezbytná podmínka pro spravedlivé zpracování.[2]
**Korektností** se rozumí popis zpracování. Relevantním příkladem transparentního a spravedlivého zpracování je situace, kdy zákazník uzavře smlouvu se svým operátorem a ten uchovává jméno a telefonní číslo pro účely fakturace služeb. Nemůže však - bez výslovného souhlasu - použít tyto informace například k oslovení sesterskou společností.[34]
**Zákonnost** je soulad zpracování osobních údajů s článkem 6 nařízení:

1. Subjekt údajů udělil souhlas se zpracováním osobních údajů pro jeden či více konkrétních účelů

2. Zpracování je nezbytné pro splnění smlouvy, jejíž smluvní stranou je subjekt údajů, nebo pro provedení opatření přijatých pře uzavřením smlouvy na žádost tohoto subjektu údajů

3. zpracování je nezbytné pro splnění právní povinnosti, která se na správce vztahuje

4. zpracování je nezbytné pro ochranu životně důležitých zájmů subjektu údajů, nebo jiné fyzické osoby

5. zpracování je nezbytné pro splnění úkolu prováděného ve veřejném zájmu nebo při výkonu veřejné moci, kterým je pověřen správce

6. zpracování je nezbytné pro účely oprávněných zájmů příslušného správce či třetí strany, kromě případů, kdy před těmito zájmy mají přednost zájmy, nebo základní práva a svobody subjektu údajů vyžadující ochranu osobních údajů, zejména pokud je subjektem údajů dítě[33]

#### 2.3.2.2 Zásada 2. - Omezení účelem

Tato druhá zásada si klade za cíl zajistit, aby byly organizace dostatečně sdílné ohledně důvodů, pro které chtějí získat osobní údaje a že zpracování informací je v souladu s rozumným očekáváním dotyčných jednotlivců[34].
V praxi druhá zásada znamená:

- Od prvopočátku musí být jasné, proč jsou údaje shromažďovány a co se s nimi následně bude dít
- Organizace musí dodržovat požadavky na spravedlivé zpracování, včetně povinnosti zaslat subjektům údajů informace o shromažďování údajů
- Je třeba dodržet požadavek o informování orgánu dohledu
- Je nutné zajistit, aby v případě, že organizace chce zpracovávat nebo zpřístupnit osobní údaje za jakýmkoliv účelem, který je dodatečný, nebo odlišný od původně stanoveného účelu, bylo nové použití, nebo zveřejnění spravedlivé a v souladu

Druhou zásadu lze shrnout následně. Osobní údaje mohou být shromažďovány pouze na základě konkrétního, jednoznačného a legitimního účelu - účelová specifikace a nesmí být dále zpracovávány způsobem neslučitelným s těmito účely - kompatibilní použití.[2]
Pokud jsou osobní údaje zpracovávány k jinému účelu, musí být tento nový účel specifikován a musí být zajištěno, že všechny požadavky na kvalitu údajů jsou rovněž splněny pro nové účely[33].

**Transparentnost** Existuje silná vazba mezi transparentností a specifikací účelu. Pokud je účel zřetelný a sdílený se zúčastněnými stranami, je ochrana omezení účelem skutečně účinná[2].

**Předvídatelnost** Pokud je cíl dostatečně specifický a jasný, mají subjekty údajů jasno v tom, co mohou očekávat. To přináší právní jistotu subjektům údajů a také osobám, které zpracovávají osobní údaje jménem správce údajů.[2]

**Kontrola jedince** Subjekt údajů může mít zpracování svých údajů pod kontrolou pouze za předpokladu, že je účel zpracování dat dostatečně jasný a předvídatelný. Pokud subjekty plně chápou účely zpracování, mohou uplatňovat svá práva, například vznesením námitky proti zpracování[2].

#### 2.3.2.3 Zásada 3. - Minimalizace dat

*„Osobní údaje musí být přiměřené, relevantní a omezené na nebytný rozsah ve vztahu k účelu, pro který jsou zpracovávány.“*[33].
Správce by měl tedy určit minimální množství osobních dat, které potřebuje k tomu, aby správně naplnil svůj cíl podnikání nebo zřízení. Měl by mít přesně tolik informací, kolik potřebuje, ale ne více. To je osvědčený postup známý jako „minimalizace dat“[2].
Nařízení GDPR hovoří o přiměřenosti, relevantnosti a omezení na nezbytný rozsah, ale blíže tyto termíny nespecifikuje. Tyto termíny jsou vždy v kontextu účelu, pro který probíhá zpracování a pro každý subjekt údajů, jehož osobní informace má organizace k dispozici[34].

#### 2.3.2.4 Zásada 4. - Přesnost

Tento požadavek úzce souvisí s třetí zásadou o minimalizaci dat. Zajištění přesnosti osobních údajů rovněž vede ke splnění tohoto požadavku. Přestože tato zásada zní jednoznačně, nařízení uznává, že nemusí být prakticky proveditelné zkontrolovat správnost každé položky osobních údajů, kterou organizace obdrží a následně uchovává.[2]
Ke splnění tohoto požadavku by organizace měla:

- realizovat přiměřené kroky k zajištění přesnosti všech osobních údajů, které získá a zpracovává
- zajistit, aby zdroj osobních údajů byl jasný a nezpochybnitelný
- pečlivě zvážit všechny problémy a nejasnosti ohledně přesnosti informací
- zvážit, zda je nutné a jak často informace aktualizovat[33]

Vyjádření názoru nebo mínění o jednotlivci je považováno za osobní údaj. Dva lidé mohou mít velice odlišné názory na schopnosti nebo osobnost jedince. Osobní zážitky, preference nebo předsudky mohou ovlivnit názory člověka do té míry, že nemůže být možné rozhodnout, který ze dvou protichůdných názorů odpovídá skutečnosti. Při zaznamenávání informací o jednotlivci je tedy důležité rozlišit, zda se jedná jen o názor nebo mínění a případně kdo tento názor vyjádřil.[34].

#### 2.3.2.5 Zásada 5. - Omezení uložení

*„Osobní údaje musí být uloženy ve formě umožňující identifikaci subjektů údajů po dobu ne delší, než je nezbytné pro účely, pro které jsou zpracovávány. Osobní údaje lze uložit po delší dobu, pokud se zpracovávají výhradně pro účely archivace ve veřejném zájmu, pro účely vědeckého, či historického*

*výzkumu nebo pro statistické účely podle čl. 89 odst. 1, a to za předpokladu provedení příslušných technických a organizačních opatření požadovaných tímto nařízením s cílem zaručit práva a svobody subjektu údajů.*"[33].
Tato zásada má velmi úzké vazby se dvěma předchozími principy. Zajištění likvidace osobních údajů, když už nejsou potřeba, snižuje riziko, že se stanou nepřesnými nebo zastaralými. Nařízení nestanoví žádné konkrétní minimální, nebo maximální lhůty. Místo toho hovoří o tom, že osobní údaje zpracovávané pro jakýkoliv účel nebo účely nesmí být uchovávány déle, než je nezbytně nutné.
V praxi to znamená, že organizace musí kontrolovat dobu uchovávání osobních údajů a zvážit účely, pro které uchovává informace při rozhodování o tom, zda a na jak dlouho si je uchová. Dále pak musí data bezpečně smazat či zlikvidovat informace, které již pro tento účel nejsou potřebné a aktualizovat, archivovat nebo bezpečně smazat informace, pokud jsou zastaralé.[2]
Ukládání osobních dat na příliš dlouhou dobu s sebou přináší různé problémy:

- zvýšené riziko, že budou informace zastaralé a budou tím páde používané chybně
- vlivem stárnutí informace je obtížné zajistit její přesnost
- přestože nejsou osobní údaje potřebné, je stále třeba zajistit jejich bezpečné použití
- organizace musí být schopná reagovat na žádosti o přístup k těmto osobním údajům, což může být při velkém množství dat obtížnější. Obzvlášť pokud jsou některá z těchto dat nepoužívaná

**Mazání údajů v IT systémech** Nařízení GDPR samo o sobě nikde nedefinuje pojem „vymazání" dat. Jednoduchá interpretace pojmu znamená jejich zničení. Situace u elektronického úložiště je poněkud složitější, protože informace, které byly smazány v jednom systému, mohou existovat v nějaké jiné podobě.
K dobré praxi patří dát jasně najevo, co se stane s daty uživatelů. Například při zrušení elektronického účtu, zda bude tento účet pouze zablokován a archivován, nebo budou jejich data opravdu trvale vymazána. V souvislosti s archivací je třeba mít na paměti, že se fakticky jedná o další formu zpracování a uchování dat.
Dobrá praxe připouští několik různých způsobů vymazání osobních dat. Patří mezi ně například fyzická destrukce, software zabezpečující vymazání, obnovení továrního nastavení, odborný zásah specialisty a formátování[2].

#### 2.3.2.6 Zásada 6. - Integrita a důvěrnost

Šestá zásada hovoří o integritě a důvěrnosti, tedy vlastně o zabezpečení dat a jejich ochraně. Nařízení GDPR v souvislosti se zabezpečením dat hovoří následovně:
*„Osobní data musí být zpracována způsobem, který zajistí náležité zabezpečení osobních údajů, včetně jejich ochrany pomocí vhodných technických nebo organizačních opatření před neoprávněným či protiprávním zpracování a před náhodnou ztrátou, zničením nebo poškozením ('integrita a důvěrnost')"*[33].
V praxi toto znamená, že organizace musí mít zajištěnou dostatečnou bezpečnostní ochranu, aby zabránila nechtěným, nebo záměrným ohrožením osobních údajů, které zpracovává. Musí tak především navrhnout a zajistit své zabezpečení tak, aby odpovídalo povaze osobních údajů, které společnost uchovává a zpracovává a musí zabránit škodám, nebo je alespoň minimalizovat, pokud už k porušení bezpečnosti dojde.
Organizace také musí jasně stanovit, kdo odpovídá za zajištění bezpečnosti informací a ověřit, zda má správné fyzické, nebo technické zabezpečení, které je podpořeno kvalitními a robustními zásadami a postupy nakládání s daty a řádně proškoleným a spolehlivým personálem. Důležitým bodem, který GDPR zmiňuje, je také připravenost reagovat na jakékoliv narušení bezpečnosti rychle a efektivně, tedy mít zpracovány náležité procesy pro tyto případy[2].
Porušení informační bezpečnosti může mít za následek způsobení skutečné škody a vyvolání obav u osob, které takováto událost ovlivní. Není výjimkou, že následkem porušení bezpečnosti dat byly:

- Podvodné transakce s kreditními kartami
- Zastrašování svědků, kterým hrozí fyzická újma a vydírání
- Zveřejnění citlivých dat, která mohou vést k těžkým následků (například osobní adresy policistů, vězeňského personálu nebo žen ohrožených domácím násilím)
- Falešné žádosti o splacení dluhu
- Hypoteční a úvěrové podvody

Ne všechny bezpečnostní incidenty mají tak závažné následky, nicméně mnohdy znamenají třeba nepříjemnosti pro postižené osoby. Typickým příkladem je zneužívání emailových adres ke spamu.[2]
Bezpečnostní opatření uvnitř organizace by pak měla zajistit, že k informacím budou moci přistupovat, opravovat je a mazat pouze oprávněné osoby a ty navíc jednají jen v rozsahu svých pravomocí. Bezpečnost by měla odpovídat povaze dotyčných informací a výši škody, která by mohla být důsledkem nesprávného použití, zneužití, náhodné ztráty nebo zničení.[34]

**Manažerská a organizační opatření** Organizace by se měla celkově zaměřit na budování firemní kultury v oblasti bezpečnosti a zvyšovat povědomí o ochraně osobních údajů v rámci celé organizace. Za jeden z klíčových prvků organizační bezpečnosti GDPR označuje osoby nebo oddělení v organizaci, které nesou každodenní odpovědnost za vhodná bezpečnostní opatření a za zabezpečení informací. Dotyčná osoba, nebo osoby by měly mít zároveň se zodpovědností i dostatečné pravomoci, včetně personálních a finančních zdrojů k zajištění bezpečnosti[2].

**Fyzická bezpečnost** Mnoho bezpečnostních incidentů souvisí s krádeží, nebo ztrátou zařízení samotného nebo k nim dochází při likvidaci staré a nepotřebné techniky. Případně jsou záznamy získány z jiných hardwarových prostředků. Do problematiky fyzické bezpečnosti je třeba zahrnout i takové oblasti, jakou je kvalita dveří a zámků, zda jsou prostory chráněny alarmem, bezpečnostním osvětlením nebo kamerovým systémem. Dále pak zahrnuje přístup do budovy, dohled nad pohybem návštěvníků nebo to, jak je v organizaci likvidován papírový odpad.

**Kybernetická bezpečnost** Bezpečnost počítačů, mobilů a tabletů se neustále vyvíjí a patří k těm nejsložitějším technickým oblastem. K zásadám kybernetické bezpečnosti patří především:

- **Počítačová bezpečnost** do této kategorie spadají firewally a antivirová kontrola, anti-spyware, automatické bezpečnostní aktualizace počítače, stahování nejnovějších oprav (záplat), přístup zaměstnanců k informacím na základě toho, jaké potřebují k výkonu své práce. Je také důležité nedovolit sdílení hesel, šifrovat veškeré osobní údaje a provádět pravidelné zálohování informací v počítačích a data se zálohami uschovat na jiném místě. Před likvidací starých počítačů a disků je pak třeba pečlivě vymazat všechny osobní informace.
- **Bezpečnost emailu** zde jde především o uživatelskou opatrnost. Zaměstnanci organizace by měli být řádně proškoleni v zacházení s podezřelými emaily.
- **Zabezpečení faxu** pokud to jde, neměl by fax být používán. Tato technologie je již zastaralá a nezabezpečená.[2]

**Porušení zabezpečení dat** Pokud navzdory opatření přijatým k ochraně osobních údajů dojde k porušení bezpečnosti dat, měla by firma postupovat podle standardního procesu pro narušení bezpečnosti. Standardní proces obsahuje čtyři základní prvky při postupu řešení bezpečnostního incidentu:

1. **Zachycení a zotavení** Důležité je si bezpečnostního incidentu všimnout a zabránit jeho pokračování.

2. **Posouzení rizik** Analýza rizik spojená s porušením bezpečnosti by měla v každé organizaci existovat ještě před jakýmkoliv incidentem. Dle tohoto dokumentu se následně postupuje.

3. **Oznámení o porušení zabezpečení** Firma by měla mít jasně stanoveno, koho je třeba o případném incidentu informovat. Ke zvážení je například oznámení osobám, kterých se incident týká, nebo Úřadu pro ochranu osobních údajů, či jiným dohledovým a regulačním orgánům.

4. **Hodnocení a reakce** Základním krokem je vyšetření příčiny a následné zhodnocení účinnosti reakce na vzniklou událost. V případě nutnosti by mělo dojít k aktualizaci bezpečnostních protokolů.[2]

#### 2.3.2.7 Zásada 7. - Zodpovědný přístup a prokázání souladu

Na rozdíl od opatření, která při ochraně osobních údajů platila dosud, se GDPR zabývá také odpovědností správce údajů. Ustanovení článku 5 je sice krátké ale z právního hlediska velmi důležité:
*„Správce odpovídá za dodržení odstavce 1 a musí být schopen toto dodržení souladu doložit.“*[33].
Tato zásada správci ukládá povinnost a zodpovědnost za zajištění souladu s předcházejícími šesti zásadami a schopnost tuto shodu dokázat, a to v každé fázi zpracování. Organizace tedy bude muset prokázat, že skutečně dělá to, co teoreticky říká ve svých pravidlech sepsaných na papíře[2].

### 2.3.3 Práva a odpovědnosti

GDPR dává občanům Evropské unie nová práva a ta stará posiluje. Subjekty osobních údajů mají především tato práva:

- **Právo být informován** - toto právo zahrnuje povinnost správce poskytnout pravdivé informace o zpracování.

- **Právo přístupu** - důvodem k tomu je, že jednotlivci mají mít možnost si ověřit zákonnost zpracování a právo být si vědom, že takové zpracování probíhá.

- **Právo na opravu** - pokud jsou údaje nepřesné nebo neúplné.

- **Právo na výmaz (být zapomenut)** - lze uplatnit pouze za určitých okolností, které dále stanovuje vyhláška GDPR.

- **Právo na omezení zpracování** - pokud je zpracovávání pozastaveno, má správce povoleno pouze ukládat osobní údaje, ale ne je nadále zpracovávat.
- **Právo přenositelnosti** - subjekty údajů mohou požádat o kopie svých osobních údajů v běžném elektronickém formátu. Smyslem je umožnit jejich snadný přenos a opětovné využití.
- **Právo vznést námitku** - pokud konkrétní osoba nedostane možnost uplatnit některé ze svých práv, má právo vznést námitku proti zpracování.
- **Práva spojená s automatizací rozhodování a profilováním** - pokud jsou data zpracovávána automaticky a osobní údaje jsou následně použity k hodnocení některých osobních aspektů (typicky hodnocení solventnosti).
- **Právo nebýt předmětem automatizovaného rozhodnutí** - navazuje na předchozí bod. Subjekt údajů by měl být informován, že může dojít k profilování jeho osoby a může to odmítnout.[2]

### 2.3.4 Role a zodpovědnosti

V následující části se zabývám popisem rolí klíčových pro GDPR a jejich zodpovědností.

#### 2.3.4.1 Správci

Role správce údajů byla klíčová i v předchozí legislativě. Nově nařízení GDPR dává důraz i na povinnosti a roli zpracovatele, správce však stále zůstává jako hlavní zodpovědná osoba. Fakticky také určuje činnosti spojené se zpracováním. Rozhoduje o tom, jaké osobní údaje budou shromažďovány, kdo je bude shromažďovat, o době shromažďování dat, účelu jejich zpracování, zabezpečení i míře rizika[34].
Povinností správce plynoucí z nařízení je ujistit se o kvalitách a seriózności svých dodavatelů, kteří budou mít k datům přístup. Součástí jeho povinností je také implementace kontrol a jejich pravidelné opakování a vyhodnocování. Jinými slovy, nelze zavést požadavky GDPR a tím konstatovat, že jsou všechny povinnosti splněny. V praxi se jedná o kontinuální a stálý proces, jehož průběh je třeba pravidelně dokladovat a mnoho činností opakovat[2].

#### 2.3.4.2 Společní správci

Pokud se jedná o společné zpracování dvou a více správců, je třeba před zahájením zpracování jasně stanovit a zdokumentovat vzájemné zodpovědnosti a povinnosti mezi správci[34].

#### 2.3.4.3 Zpracovatelé

Zpracovatelé jsou osoby, které zpracovávají osobní údaje, nebo mají k údajům shromážděným správcem přístup. Každý zpracovatel musí splňovat kritéria určená správcem podle požadavků GDPR, správce však nemůže zpracovateli určit každý krok průběhu zpracování, ale musí se spolehnout na jeho ujištění, že zpracování osobních dat je bezpečné a spolehlivé. Zpracovatel pak nese zodpovědnost za:

- IT systém a jeho architekturu zajišťující dostatečnou ochranu dat
- formu uložení a zabezpečení dat
- fyzickou bezpečnost dat
- bezpečný způsob transferu dat mezi organizacemi
- personál, který má k údajům přístup
- dodržení skartačního řádu
- zajištění práv subjektu údajů, jako právo výmazu, pozastavení apod.

Zpracovatel nesmí bez vědomí správce využít služeb jiného zpracovatele[34].

#### 2.3.4.4 Pověřenec pro ochranu osobních údajů

Pověřenec pro ochranu osobních údajů je pozice v rámci organizace, v níž působí zaměstnanec, nebo externí pracovník jako ochránce osobních údajů zákazníků, klientů, pacientů, zaměstnanců a podobně. V souladu s nařízením GDPR jsou povinny všechny podniky, které nabízejí zboží nebo služby zákazníkům v EU a shromažďují jejich údaje, jmenovat pověřence pro ochranu osobních údajů. Pověřenec sleduje zákony a postupy týkající se ochrany osobních údajů a provádí interní posuzování vlivu zpracování dat na soukromí subjektů[2].

### 2.3.5 Záznamy zpracování

GDPR jasně stanovuje povinnost organizace uchovávat záznamy prokazující shodu zpracování osobních údajů s nařízením a v případě potřeby vše dozorovému orgánu zdokumentovat. Existence zásad zpracování je klíčová, jedná se o jediný dokument, který by mělo být možné kdykoliv předložit. Zákon navíc ukládá povinnost správci údajů i zpracovateli uchovávat konkrétní údaje o svých zpracovatelských činnostech[2].

| **Záznamy správce** | **Záznamy zpracovatele** |
|---|---|
| Jméno a kontaktní údaje správce a/-nebo jmenovaného zástupce a pověřence | Jméno a kontaktní údaje zpracovatele a každého správce, jehož jménem zpracovávám osobní údaje, jmenovaného zástupce správce, nebo zpracovatele a pověřence |
| Účely zpracování | Kategorie zpracování vykonávaných jménem každého správce |
| Popis kategorií subjektů údajů a kategorií osobních údajů | Detaily všech přesunů údajů do třetích zemí nebo mezinárodních organizací, včetně jejich jasné identifikace a dokumentace vhodných záruk |
| Kategorie příjemců, kteří mají k údajům přístup včetně těch, kteří jsou mimo EU | Obecný popis technických a organizačních opatření v oblasti bezpečnosti |
| Detaily všech přesunů údajů do třetích zemí nebo mezinárodních organizací, včetně jejich jasné identifikace a dokumentace vhodných záruk | |
| Dobu pro výmaz různých kategorií osobních údajů | |
| Obecný popis technických a organizačních opatření v oblasti bezpečnosti | |

Tabulka 2.2: Tabulka povinných dokumentů správců a zpracovatelů[2]

### 2.3.6 Školení

Kybernetické útoky, které vedou k narušení dat, jsou obvykle způsobeny lidskou chybou v důsledku toho, že zaměstnanci dělají něco, co by neměli. Typickým příkladem je nevhodné zacházení s paměťovou kartou, posílání emailů, jejichž data nejsou šifrovaná, špatně zabezpečená Wi-Fi, ztracený telefon nebo tablet bez ochrany PINem či nedodržování zásad při práci s listinnými dokumenty v kanceláři, jako je zapomenutý důvěrný dokument v tiskárně.
Doposud byla školení bezpečnosti zaměstnanců pouze vhodnou doplňkovou aktivitou, od května 2018 ji však nařízení GDPR chápe jako nutnou a povinnou součást opatření směřujících k ochraně dat.
Zaměstnanci musí chápat GDPR, školení by měla být relevantní a osobní a zaměstnanci by měli být po jeho absolvování schopní rozpoznat porušení GDPR[2].

### 2.3.7 Informační technologie

Jednou z nejproblematičtějších oblastí v ochraně osobních dat je jejich přenos, a to ať už transfer pomocí sítí a internetu, nebo fyzický přenos v mobilních zařízeních. Této problematice se věnuje celá část GDPR, a to jak přenosu v datové podobě, tak přenosu do listinné podoby.

**Tiskárny** Ve všech firmách, které drží krok s dobu se prosazuje trend centralizace tiskových zařízení, zejména z důvodu minimalizace nákladů na údržbu a nákup tiskáren. V kanceláři je tedy umístěný velký multifunkční stroj, zpravidla na chodbě, nebo v samostatné místnosti. Pokud si však uživatel vytiskne citlivý dokument a zapomene si ho z tiskárny vyzvednout, může snadno dojít k úniku informací, když si ho - byť jen omylem - přečte někdo jiný.
Praktickým řešením, které mnohé firmy zavádějí, je tisk až na vyžádání. Tiskárna tedy čeká na zadání PINu, nebo otisku prstu, případně přihlášení uživatele a tiskne dokument až v jeho přítomnosti.
Další ne příliš zabezpečenou částí je zpravidla přenos dokumentů do tiskárny, jež probíhá přes interní síť a dokument není nijak zašifrován a může být zachycen. Otázkou je pak také samotná tiskárna. Velká multifunkční zařízení mívají samostatný disk a mohou ukládat informace o tištěných dokumentech. Je tedy třeba na to dbát, pokud se tiskárna například posílá k opravě[2].

**Bezpečí přenosných zařízení** Pokud se mluví o přenosných zařízeních, v dnešní době jsou obvykle myšleny především mobilní telefony, které uchovávají nezměrné množství informací o majiteli. Základním bezpečnostním prvkem je nastavení zamykání obrazovky. Dalším krokem je šifrování dat uložených v mobilu[34].

#### 2.3.7.1 Politika hesel

V dnešní době je zcela běžné, že jeden uživatel má několik desítek internetových účtů. Jsou to například přístupy do bankovnictví, e-shopů, emailů, sociálních sítí, cloudových úložišť a dalších. Každý z těchto účtů obsahuje přístupové osobní údaje. Delší doby k prolomení hesla lze dosáhnout nejen co nejsložitější kombinací různých znaků ale také délkou samotného hesla. Doporučeno je používat delší věty složené z několika slov[34].

#### 2.3.7.2 Kamerové systémy

Kamerové systémy jsou v České republice běžnou součástí života. Jsou téměř na každé ulici a veřejnost si na ně zvykla, přesto však představují velký zásah do života běžných lidí. Nařízení GDPR chápe použití kamerových systémů jako sběr osobních dat, pokud ze záznamu lze rozpoznat tváře jednotlivých osob. V současné době je kvalita kamer na takové úrovni, že se to týká prakticky každého provozovatele CCTV.
Použití mechanismů získání údajů bez vědomí osob je obecně nezákonné. Skryté sledování je obvykle povoleno v individuálních případech, kdy se zpravidla jedná o vyšetřování trestných činů. Bezpečnostní agentury, které umístí a provozují kamery ve prospěch klienta jsou pak považovány za zpracovatele osobních údajů.[2]

#### 2.3.7.3 Online

Riziko zneužití osobních údajů při pohybu online je nesrovnatelně vyšší než při běžném offline použití počítačové techniky. Skutečnost, že data jsou na internetu prakticky nejzranitelnější, klade na organizace mimořádné požadavky z hlediska bezpečnosti. Při zpracování a ukládání dat online je třeba dbát především na tyto zásady:

- Zvážit, zda k činnosti organizace je skutečně třeba informace o lidech shromažďovat. Je přijatelné požadovat osobní informace, pokud se lidé dotazují, nebo vstoupí s organizací do obchodního vztahu.
- Pokud jsou shromažďovány informace o fyzických osobách, měly by vědět, kdo je shromažďuje a za jakým účelem. Na webových stránkách organizace by mělo existovat jasné a přesné vysvětlení.
- Správce má povinnost chránit informace poskytnuté zákazníkem, je třeba tedy zvážit šifrování dat. Zaměstnanci s přístupem k těmto datům by pak měli být vyškolení, aby byli schopni zajistit dostatečnou bezpečnost.
- Při použití subdodavatele je třeba mít písemnou smlouvu, v níž jsou stanoveny jasné podmínky zajištění bezpečnosti dat.

- Při použití osobních údajů k zasílání marketingových sdělení, by měl zákazník mít jednoduchou a srozumitelnou možnost tato sdělení zrušit.
- Je důležité mít na webových stránkách uvedený jeden konkrétní kontakt, na který mohou zákazníci směřovat své dotazy a stížnosti, na něž následně bude někdo reagovat.
- Organizace by měla shromažďovat pouze informace, které skutečně používá.
- Lidé mají právo na přístup k informacím, které o nich organizace má.
- Organizace by měla v pravidelném časovém intervalu zákazníky vyzývat k aktualizaci informací, které o nich uchovává, aby se zajistila jejich důvěryhodnost.[2]

#### 2.3.7.4 Oznámení o ochraně osobních údajů

Každá organizace, která musí implementovat GDPR, by měla mít i oznámení o ochraně osobních údajů. Výchozím bodem pro toto oznámení by mělo být informování subjektů údajů. Takové oznámení by mělo obsahovat informace o organizaci, co bude organizace s daty uživatelů dělat a s kým bude organizace tato data sdílet[34].

#### 2.3.7.5 Šifrování

GDPR doporučuje pro ochranu citlivých osobních informací dva klíčové technologické prvky: anonymizaci a šifrování. Za určitých okolností se v této souvislosti dá hovořit také o pseudonymizaci nebo generalizaci.
S rozvojem technologie vznikly šifry, které patří do kategorie moderního kryptování. Lze je rozdělit na symetrické a asymetrické. Hlavní rozdíl spočívá v tom, zda se k zašifrování a rozšifrování používá stejný, nebo jiný klíč[2].

**Symetrické šifry** Symetrická šifra je taková šifra, která používá ke kryptování i dekryptování stejný klíč. Příkladem takových šifer mohou být například populární AES-256 nebo Twofish či DES[35].

**Asymetrické šifry** Asymetrické šifry oproti tomu potřebují klíčový pár. Ten se skládá ze dvou klíčů, veřejného klíče a soukromého klíče, které nejsou shodné, jsou asymetrické. Příkladem může být například hojně používaná RSA a její moderní nástupce Eliptické křivky[36].
V posledních letech došlo k řadě případů odcizení, ztráty, nebo zneužití osobních údajů. U mnoha z těchto případů byl tento incident způsoben nedostatečnou ochranou dat nebo zařízení, kde byly údaje uloženy.
V praxi by měly informace být zašifrovány a dešifrovány pomocí tajného klíče. Bez tohoto klíče k informacím nelze získat přístup a ty jsou tak chráněny

před neoprávněným nebo protiprávním zpracováním. Útočník se může pokusit o prolomení klíče metodou „brute force“, kdy vyzkouší všechny možné kombinace klíče. Prakticky to ale bude trvat velmi dlouho, než správný klíč najde. Vše záleží na kapacitě výpočetní techniky a její rychlosti.
Organizace by měly zvážit výhody, které šifrování nabízí, stejně jako zbytková rizika a zavedení dalších bezpečnostních opatření, která mohou být vhodným doplňkovým řešením. Posouzení vlivu na ochranu osobních údajů pomůže zdokumentovat a zdůvodnit všechna případná rozhodnutí.
Správci i zpracovatelé osobních údajů by si měli uvědomit, že může existovat směrnice, která upravuje doporučené způsoby šifrování pro konkrétní odvětví a druh osobních údajů.[2]

Kapitola 3

# Vyhodnocení analýzy a návrh

V následující části diplomové práce se věnuji zpracování analýzy, která vyplývá z předchozí teoretické části práce. Toto zpracování je opět rozděleno na tři části: BOZP portál, etické hackování a bezpečnost a GDPR. Vyhodnocuji zjištěné skutečnosti a navrhuji konkrétní řešení problémů.

## 3.1 BOZP portál

BOZP portál aktuálně postrádá systematický přístup. Vzhledem k počtu uskutečněných bakalářských i diplomových prací byly na portálu použity velmi rozdílné programovací styly. Přestože portál běží již pátým rokem nad poměrně dynamicky se rozvíjejícím frameworkem, neproběhlo dosud mnoho aktualizací kódu.
Velkou bolestí portálu je také nedostatečná automatizace. Z konzultací s pracovníky BOZP vyplynuly tři oblasti, které by si zasloužily úpravy. Jedná se o Statistiky, Školení a Reklamace. Kromě těchto úprav je pak také třeba udělat změny v aktuální implementaci cyklu školení, a to především u vyhlášky 50/78Sb.
Automatizovat bude třeba také samotné nasazování kódu na server (deployment) a balíčkování, které aktuálně využívá pouze verzovací systém GIT.

### 3.1.1 Statistiky

V aktuálním stavu slouží statistiky na portálu opravdu jako pouze čistě statistický nástroj. Správce systému si může zobrazit všechny uživatele k danému školení a má k dispozici informaci o tom, kdo ono školení již absolvoval a kdo ještě ne, to se pak následně přepočítá na procenta. Tento výsledek se zobrazí správci systému spolu s celkovým seznamem uživatelů.

**Problém** Neexistuje však zatím žádný způsob hromadné komunikace s uživateli. Pokud tak chce správce oznámit například nové vypsané termíny školení BOZP, musí buď psát hromadný email všem uživatelům portálu, nebo je vybírat po jednom, což je v počtu okolo 800 - 1000 nových studentů bez BOZP každý rok neúnosné a nerealizovatelné.

**Řešení** Řešením tohoto problému by byla možnost filtrování uživatelů podle splnění školení. Následně by pak měl mít pracovník BOZP možnost vygenerovat seznam těch, kteří zatím školení nemají, případně jim rovnou ze systému odeslat hromadný email.
Vzhledem k tomu, že přes portál je organizováno vetší množství různých školení, je třeba, aby bylo možné také uživatele filtrovat podle typu školení, která mají, nebo nemají splněna.

### 3.1.2 Školení

Momentální stav portálu vyžaduje mnoho zásahů člověka - správce. Zásadním ulehčením by tedy byla automatizace procesů souvisejících se správou školení.

**Problém** V momentálním řešení neexistuje možnost přímo z portálu odesílat hromadné emaily a také není téměř žádná možnost filtrování dle absolvovaných školení. Dalším nedostatkem je to, že portál nebere v úvahu, zda student, který je v prvním ročníku už školení vyhlášky 50/78Sb například absolvoval.

**Řešení** Pro potřeby vyhlášky 50/78Sb i ostatních typů školení je třeba, aby portál začal rozlišovat mezi několika stavy, do kterých se mohou uživatelé portálu dostat. A to sice:

- Student je v 1. ročníku a nemá školení
- Student je v 1. ročníku, již byl proškolen, ale v průběhu tohoto školního roku mu školení vyprší (jedná se obvykle o ty studenty, kterým bylo po 2. ročníku ukončeno studium a nastoupili znovu)
- Student je ve 2.,3. a vyšším ročníku (včetně studentů magisterského studia) a v průběhu tohoto školního roku mu vyprší platnost školení
- Student je v 1. ročníku magisterského studia, před tím však nikdy nestudoval na FIT ČVUT a musí být tedy řádně proškolen

Školení vyhlášky 50/78Sb se provádí jen u studentů 1. ročníků bakalářské etapy studia, a to ještě navíc jen u těch, kteří nemají absolvovaný předmět BI-SAP. Informace o absolvování tohoto předmětu by se dala získat použitím stávajícího KOS API, pomocí něhož probíhá import studentů do systému.

### 3.1.3 Reklamace

Studenti by měli mít možnost reklamovat stávající stav všech svých školení.

**Problém** V systému BOZP neexistuje momentálně žádná zpětná vazba, kterou by mohli jeho uživatelé využít ke kontaktování správce. Pokud tak student například chce uznat školení BOZP, nebo ho reklamovat, musí psát správci email.

**Řešení** Pokud studentovi přijde email, že je neproškolený, měl by mít možnost toto reklamovat pomocí BOZP portálu. Může také reklamovat, že proškolený není, pokud mu bude následkem chyby školení přiděleno.
Podobné reklamace by se pak měly týkat také studentů, kteří jsou na stáži nebo v zahraničí. Aktuálně student nemá možnost sám si tuto informaci do systému zanést. Tato možnost by měla být přidána.

### 3.1.4 Ostatní návrhy na změnu

V rámci dalších požadavků na změny v implementaci byly vzneseny tyto návrhy, které budou následně zakomponovány do výsledného programu.

- Při editaci profilu by měly být ověřovány všechny parametry, které uživatel vkládá.
- Drobná změna logiky při školení BOZP, přidání nového stavu - Test absolvován, nepodepsáno. Do tohoto stavu se student dostane v momentě, kdy bude mít absolvován test vyhlášky 50/78Sb, ale nebude mít podepsanou prezenční listinu. Stav ho bude vyzývat, aby se dostavil k podepsání.
- Na portálu by měly být spuštěné již dříve implementované nové gridy pro zobrazování dat. Je třeba jejich funkcionalitu ověřit a následně je nasadit.
- Udělení školení by mělo ukládat datum proběhlé akce, nikoliv aktuální datum.
- Pokud student napíše test znovu a nemá ještě propadlé datum dříve napsaného testu, test se aktuálně jeví, jakože je již podepsaný. To však není správný postup. Student se musí na podpisovou listinu podepsat znovu. Tuto funkcionalitu je třeba upravit tak, aby odpovídala realitě.
- Student, který by měl splnit kterékoliv školení, ale je aktuálně na stáži nebo zahraničním výjezdu, má mít možnost tuto skutečnost uvést ve svém profilu. Dosud měl možnost toto změnit pouze správce BOZP.

### 3.1.5 Průběžná integrace

Součástí zadání mé diplomové práce je také zavést do vývoje portálu principy průběžné integrace. Průběžná integrace neboli Continuous Integration je souhrn různých vývojářských metod a nástrojů sloužících k urychlení vývoje softwaru a spolupráce týmů. Je také jednou ze součástí metodik extrémního programování. Průběžná integrace má několik základních principů, které mohou být zavedeny jednotlivě, nebo v kombinaci. Jedná se především o:

- **Centralizované úložiště kódů** Každý softwarový projekt má mnoho různých souborů a součástí, na kterých typicky spolupracuje mnoho lidí najednou. Proto je doporučeno použití repository (sdíleného úložiště zdrojových kódů). Zpravidla se používá ve spojení s verzovacím systémem, jako jsou GIT nebo SVN. Tento systém pak zajistí konzistenci souborů a zároveň zálohu každodenní práce. K pokročilejším funkcím tohoto úložiště patří například globální přehled změn nebo reportování toho, kdo toto úložiště nejvíce používá, či která část kódu je nejčastěji upravována. Mezi nevýhody tohoto přístupu patří potřeba přísné disciplíny celého týmu a závislost na centrálním systému.

- **Automatizovaný build** Každé vydání nové verze systému vyžaduje kompilaci zdrojových kódů, kopírování a přesouvání souborů, nastavování a přesun výsledného balíčku na určené místo, případně spuštění, nebo restart systému. Z důvodu prevence lidské chyby je vhodné tento proces automatizovat, a tím ho i usnadnit.

- **Automatizované testy** Po úspěšném buildu nemusí být ještě jisté, že program pracuje správně jak technologicky, tak logicky. Ke kontrole tohoto slouží automatické testování. Základním principem je přehození fází vývoje. Programátor nejprve napíše testy a pak až následně samotnou funkci, která má testy projít.

- **Kontrola kvality kódu** Pro rychlejší a efektivnější změny je nutné kontrolovat kvalitu zdrojového kódu. Například se to vztahuje k počtu komentářů, používání funkcí a neopakování kódu. Ve společnostech zabývající se vývojem zpravidla existují předpisy pro kontrolu kódu.

- **Zpřístupnění poslední verze** Pokud je to možné, je dobré k poslední verzi buildu dát přístup co největšímu okruhu uživatelů, aby ji mohli používat a tedy i testovat.

Hlavními výhodami průběžné integrace jsou tedy především rychlé nalezení chyb v kódu a jeho automatická kontrola. Důležitou součástí je také přehled o jednotlivých verzích a rychlý přístup k poslední verzi softwaru. V neposlední řadě je to také ušetření času a financí při kompilaci a vydávání nové verze a šetření času testerů použitím automatických testů.

**Řešení průběžné integrace** Pro realizaci průběžné integrace je možné využít různé systémy i implementace. Pro portál BOZP jsem se po konzultaci s vedoucím práce rozhodla pokračovat na stávající infrastruktuře.
Testovací server, který byl již nainstalovaný studentem přede mnou, by měl být i nadále používán pro testovací běh, portál běžící pod serverem BOZP pak zůstává jako produkční server. Jako vývojové prostředí tedy bude nadále využívaná verze, kterou bude mít programátor sám u sebe na počítači.
Aktuálně je v projektu používán verzovací systém GIT, pomocí serveru GITLab je možné řešit průběžnou integraci nástrojem GitLab Runner. Ten funguje na základě YAML souborů, které mu stanovují úkoly při operacích gitu commit a push. V tomto konfiguračním souboru se dá následně nastavit celá průběžná integrace.
Toto řešení volím především proto, že správce aplikace má k dispozici GitLab Runner, který není běžně bezúplatně dostupný.
Pro nasazení průběžné implementace do projektu bude třeba udělat nový testovací server, který bude možné používat pro testování aplikace. Dále také bude potřebné napojení na produkční server a jeho nastavení, aby bylo možné bez obav nasazovat upravenou aplikaci do produkce.

## 3.2 Etické hackování a bezpečnost

Po přečtení zprávy, která vzešla z předmětu BI-EHA (Etické hackování) je jasné, že aplikace BOZP má poměrně mnoho „děr“ po stránce bezpečnosti. Mnoho těchto problémů by se jen obtížně hledalo bez znalosti zdrojového kódu, nicméně pro zkušeného hackera není problém tyto slabiny objevit.
Proto, aby bylo možné nadále portál BOZP vůbec provozovat, je třeba především opravit kritické chyby, a to sice:

**Uživatel si může změnit roli** Změna role je možná pomocí změny argumentu v url. Pro opravu tohoto problému je třeba důsledně kontrolovat právo přístupu. Následně je třeba zakázat přístup k této funkci jinými rolím, než je správce.

**Lze nahrát a spustit php soubor** S pomocí této chyby je možné následně odhalit například soubory s přihlašovacími údaji, editovat lokální data, nebo všechno ze serveru vymazat. Problém je způsoben při nahrávání obrázku s certifikátem. Zabránit útoku přes tento bod by měla kontrola nahrávaného souboru a jeho přípony.

Jako další jsou tu pak chyby, které jsou považovány za podstatné, nejsou sice kritické, ale mohou také vážně ohrozit bezpečnost serveru a dat, která na něm jsou.

**Problémy presenterů** Presenter na hlavní stránce nijak neomezuje role. Kdokoliv tak může vymazat články z databáze. Funkce editace a přidávání jsou obě zabezpečené. Presenter, který zobrazuje profil uživatele, má v url jasně zobrazené id. Kdokoliv přihlášený se může bez jakýchkoliv zábran podívat na profily dalších uživatelů. Co je ale horší, kdokoliv může editovat profil komukoliv, jehož id uhodne. Opravě obou těchto chyb opět pomůže kontrola rolí uživatele.

**Je možné používat ukradené PHPSESSIONID** Neexistuje žádná jiná kontrola session, kromě PHPSESSIONID. Po ukradení aktivní session není problém vystupovat jako daný uživatel. Řešením tohoto problému je přidání další kontroly session.

**Jedna ze stránek nepoužívá anti-CSRF token** Stránka editace kódu nepoužívá tento token, jako jediná. Ostatní stránky ho používají. Řešením je začít tento token používat i na této stránce.

**Lze použít SQL ijection** Tato chyba je v některých parametrech u orderBy. Do tohoto SQL Query je možné vložit téměř jakýkoliv skript, včetně smazání celé databáze. Je třeba pečlivěji kontrolovat všechny vstupy i adresy zadávané do portálu.

**Na serveru je dostupný dump databáze** Dump databáze by všeobecně měl být jako záloha skladován jinde než na stroji, na kterém běží samotná databáze. Řešením tohoto problému je tento soubor buď smazat a přesunout archiv jinam, nebo jej alespoň skrýt, aby nebyl veřejně přístupný.

Kromě těchto chyb jsou pak ještě na portálu chyby, které nejsou příliš podstatné, ale není vhodné je z hlediska bezpečnosti nechávat na běžícím serveru. Jedná se především o tyto problémy:

**Další problémy presenterů** Pomocí změny id při změně id je možné zobrazit jakoukoliv akci v databázi. Opět pomůže důslednější implementace kontroly rolí uživatelů, případně přístupů.

**Staré administrátorské nástroje na serveru** Na portu 8080 běží několik nástrojů určených k administraci serveru. Většina jich již nefunguje, ale z pohledu bezpečnosti by bylo vhodné je odstranit, neboť by mohly vést k bezpečnostnímu problému.

**Administrátorské nástroje jsou přístupné z internetu** Zkratka pro administrátorský nástroj k databázi nefunguje, ale přímá cesta ano. Pro opravu tohoto problému by bylo vhodné tento soubor skrýt.

**PHPSESSIONID nemá secure tag** Pro tento problém neexistuje žádné racionální zdůvodnění. Celá stránka již běží na HTTPS, není tedy důvod secure tag nepoužívat.

**Skript k vytváření uživatelů je přístupný z internetu** Skript, pomocí kterého se přidávali uživatelé, je přístupný. Tento skript je pravděpodobně zastaralý a již s ním nelze nové uživatele vytvářet. Nicméně pro bezpečnost serveru by měl být odstraněn.

**Web server vrací svůj fingerprint** Tento problém nalezený na předmětu BI-EHA byl již administrátorem odstraněn.

Revize kódu také přinesla dva problémy, které by měly být vyřešeny, přestože nejsou přímo porušením bezpečnosti. Jedná se o tyto problémy:

**Problém presenteru TrainingType** Presenter má problém s pojmenováním a při pokusu o jeho zobrazení vrací Server Error.

**Problémy přihlašovací stránky v EN** Stránka by měla být úplně odstraněna, nebo by měla přesměrovávat na SSO stejně jako stránka v češtině.

### 3.2.1 Všeobecná kontrola zabezpečení SSL

Protože zpráva vzešlá z Etického hackování se příliš nezabývala celkovým zabezpečení stránky pomocí SSL, analyzovala jsem jej pomocí nástroje SSL labs. Jedná se o službu poskytovanou zdarma za účelem zvýšení zabezpečení veřejně přístupných serverů. Služba provede hloubkovou analýzu konfigurace nastavení SSL na jakémkoliv webovém serveru vystaveném veřejně do internetu. SSL labs také uvádí, že informace, které získá nikam neukládá a nepoužívá je ani výsledky testů pro žádné jiné účely, než informování majitele serveru.
Nástroj uděluje serverům známky A - F, podle úrovně zabezpečení a nastavení jejich SSL. Je také možné porovnat svou úroveň s ostatními majiteli, kteří se rozhodnou výsledky zveřejnit.
SSL Labs je aktuálně jednou z nejpopulárnějších služeb tohoto druhu. Mimo jiné kontroluje:

- Vydavatele certifikátu
- Validitu certifikátu a algoritmus použitý k jeho podpisu

- Detaily povolených protokolů
- Dovolené šifry
- Simulovaný handshake

Výsledkem testu pak je poměrně přesná analýza s doporučeními, kterými se může řídit buď administrátor nebo bezpečnostní expert při nápravě chyb v zabezpečení serveru.
Služba SSL labs vyhodnotil původní zabezpečení serveru jako třídu C. Podrobnějším popisem tohoto výsledku a jeho důvodů se zabývám v části této práce Implementace a nasazení.

## 3.3 GDPR

GDPR je velkým oříškem i pro znalce v oboru. Především českými odbornými internetovými médii probíhají v poslední době poplašné zprávy ve snaze vzbudit paniku. Většina právníků se ale snaží tyto nálady mírnit. GDPR pouze upravuje stávající normy a standardizuje to, co bylo dosud v zákonech mnoha členských zemí EU, do centrální podoby zákona. Pokuty jsou také uváděny v maximální výši, přičemž příslušné úřady je mohou nařídit v jakékoliv výši.
Před nastudováním problematiky GDPR nebylo úplně jasné, jak klasifikovat portál BOZP. Nyní bych ho zařadila mezi zpracovatele dat. Portál BOZP zpracovává data pro ČVUT, potažmo pro Fakultu informačních technologií, kteří jsou tedy zodpovědnými správci. K této klasifikaci lze přistoupit, protože portál BOZP sám o sobě osobní data od uživatelů neshromažďuje, pouze přebírá z centrálních systémů a dále je zpracovává. Přesto je třeba, aby portál zavedl a dodržoval podmínky GDPR a předešel tak případné žalobě, která by mohla být vznesena vůči správci, tedy ČVUT.

### 3.3.1 GAP analýza

Tato analýza je v podstatě prvním krokem v rámci implementace požadavků GDPR do firemních procesů. Používá se k analýze a definování rozdílu mezi stavem v současnosti a stavem požadovaným, tedy „Kde jsme“ a „Kde chceme být“. Stav, kde chceme být, pak poměrně jednoznačně stanovuje směrnice. Cílem analýzy je především zjistit odpovědi na následující otázky:

- **Kde jsou v organizaci sběrné uzly osobních dat?**
  Tato otázka se BOZP portálu netýká, data nesbírá, pouze je přebírá a zpracovává.
- **Jaká je struktura shromažďovaných dat?**
  Data, která aplikace BOZP shromažďuje, jsou citlivého charakteru. Jedná

se o jména, personální čísla, přihlašovací jména, telefonní čísla a fakultní i soukromé emailové adresy. Kromě toho jsou v aplikaci také uložené informace o tom, kterou přednášku student navštívil a jestli absolvoval test.

- **Pomocí jakých nástrojů jsou data shromažďována a jak byl získán souhlas k jejich zpracování?**
  Tato položka se BOZP opět týká jen nepřímo. Data jsou shromažďována školou, která si od osob, kterých se to týká, vyžádá informovaný souhlas o shromažďování a zpracování osobních údajů.

- **Kdo má k datům přístup a na základě jakého oprávnění?**
  V optimálním případě má k datům uloženým v portálu BOZP přístup pouze správce portálu a pracovníci BOZP. Ti mají přístup na základě nutnosti znalosti těchto údajů ke zpracování školení BOZP. V aktuálním stavu ale vzhledem k bezpečnostním problémům portálu má k cizím citlivým datům přístup prakticky kdokoliv, kdo se může přihlásit. Toto je z hlediska GDPR nepřípustné a je nutné to napravit.

- **Jak jsou data uchovávána a chráněna?**
  Data jsou uchovávána v databázi PostgreSQL, která je zabezpečená přístupem pouze z localhostu a jménem a heslem. Ochrana dat je ale velmi vážně narušena vystaveným dumpem databáze na serveru, který je třeba z pohledu GDPR odstranit. Pouhé zamezení přístupu k datům z venčí nestačí, je třeba, aby se k této záloze skutečně nedostal ani správce systému, který by k datům přístup mít neměl.

- **V jakých systémech se s daty pracuje a v jakých procesech figurují? Jsou tyto procesy v souladu s GDPR?**
  BOZP portál kromě svého vlastního systému využívá školní SSO Shiboleth. Data, která jsou předávána mezi těmito dvěma systémy, jsou šifrována pomocí HTTPS, což je dostatečná ochrana pro přenos z pohledu GDPR. Na následné zacházení s daty na Shibolethu nemá tato práce žádný vliv. Samotný portál BOZP je zabezpečen také pomocí HTTPS a PHPSESSIONID. Některá bezpečnostní rizika, která byla v oblasti zabezpečení objevena při etickém hackování, jsou problémem i pro GDPR a z právního pohledu je třeba je odstranit.

- **Vazby a smlouvy třetích stran**
  BOZP portál nemá aktuálně vazby na žádné třetí strany, nebylo by to v souladu s politikou školy, kdy se přistupuje k větší centralizaci jednání. Tento bod se tedy portálu netýká.

- **Přístup k hodnocení dopadu na soukromí**
  V aktuálním stavu nelze tento přístup hodnotit, protože zatím nikdo neprováděl hodnocení dopadu na soukromí. Během mé práce na BOZP byli

všichni, kterých se to týkalo, vstřícní a chtěli problémy se soukromými daty vyřešit.

- **Proces řízení incidentů a schopnost reagovat** Není zaveden žádný proces řízení incidentů, protože není zavedeno žádné monitorování incidentů. Pokud tedy dojde k nějakému úniku dat, je nasnadě, že se o tom správce portálu ani nedozví. Aby bylo vše v souladu s legislativou, doporučovala bych zavést nějaký systém monitoringu portálu, nebo alespoň pravidelné kontroly. Co se týká reakce, je možné, v případě, že by došlo k narušení bezpečnosti, portál celý jednoduše na omezenou dobu vypnout. Nejedná se totiž o kritickou službu, především v letním semestru. V případě problémů je to i postup, který bych doporučila.

GDPR jasně ukládá, aby zpracovatelské činnosti byly podchyceny smluvně. Je také třeba ověřit platnost a relevantnost udělených souhlasů se zpracováním osobních dat. Tato položka bohužel není v mé kompetenci a musím se tedy omezit na pouhé doporučení pracovníkům BOZP aby tyto věci ověřili.

### 3.3.2 Posouzení vlivu na ochranu osobních údajů (DPIA)

Smysl DPIA (Data Protection Impact Assessment), tedy posouzení vlivu na ochranu osobních údajů, vychází ze základního principu GDPR, tedy zodpovědnosti správce. Organizace by měly na základě sestavené DPIA identifikovat a zhodnotit rizika plynoucí ze zpracování osobních údajů.
GDPR nevyžaduje provedení DPIA pro každou operaci zpracování, ale pouze pro takové, kde je pravděpodobné, že zpracování bude mít za následek vysoké riziko pro práva a svobody fyzických osob. Zejména je pak důležité, pokud se zavádí nová technologie nebo nový způsob zpracování dat.
Z toho vylívá, že DPIA pro BOZP portál není a pravděpodobně v budoucnu nebude nutné. Informace, které shromažďuje, nejsou totiž vysoce rizikové a jejich únik by sice uživatelům mohl způsobit nepříjemnosti, nicméně nijak významně by je neohrozil. Největším problémem by byl pravděpodobně únik telefonních čísel.

### 3.3.3 Analýza zabezpečení aplikací ČVUT

Vzhledem k tomu, že pro zabezpečení aplikací v rámci celé ČVUT je nutné postupovat jednotně, vydalo Výpočetní a informační centrum ČVUT metodiku pro implementaci požadavků GDPR. Zároveň s touto metodikou poskytlo také svým oddělením podrobný formulář pro analýzu aktuálního stavu GDPR v aplikaci. Před implementací změn byla aplikace zabezpečeně z 54,36 % dle sledovaných oblastí. Z celkového počtu 11 povinných opatření byla 4 plněna, 2 plněna částečně a 5 neplněna.

### 3.3.4 Návrh vyplývající z analýzy zabezpečení aplikací ČVUT

V této části lze najít návrhy, které pro aplikaci doporučuji z hlediska organizačního zabezpečení splnění GDPR.

#### 3.3.4.1 Přidělení odpovědností

Z pohledu GDPR je velmi důležité rozdělení a přidělení odpovědností za činnosti týkající se aplikace. Jde především o rozhodování o aplikaci, určování podmínek jejího běhu a rozšiřování a technickou správu. Jsou to tedy role jako business vlastník, vlastník rizik, technický správce a podobné. Pro splnění této podmínky bych doporučovala mít obsazeny alespoň ty nejdůležitější role a mít zveřejněné materiály obsahující jména, funkce a kontaktní údaje pověřených osob.

#### 3.3.4.2 Pravidla pro přidělování přístupů k aplikaci

V aktuálním systému dostane uživatelská práva každý, kdo má platný záznam v KOSu. Vyšší, tedy například administrátorská práva, dostávají uživatelé na základě individuálního posouzení správce po žádosti. Zde by bylo třeba jasně definovat, kdo dostane jaká práva a jak. Nejlepší způsob definice by byl interní dokument s pravidly.

#### 3.3.4.3 Formalizovaný proces odebírání přístupů

K odebrání přístupu dojde vymazáním uživatele z KOSu a následnou synchronizací dat. Stejně jako u přidělování přístupu k aplikaci i zde by měl být implementován nějaký interní dokument definující kdo a za jakých přesných podmínek přístup k aplikaci dostane.
Tato práva také v současnosti není možné přezkoumat a uživateli je odebrat. Pokud má stále platný záznam v KOSu, bude při dalším importu dat jeho účet obnoven.

#### 3.3.4.4 Privilegovaná přístupová oprávnění

Stejně jako u odebírání a přidělování celkového přístupu k aplikaci, je třeba také mít formalizovaný postup pro přidělování a odebírání privilegovaných přístupových práv. V aktuální verzi aplikace jsou privilegovaná přístupová práva přidělovaná na základě rozhodnutí správce aplikace.
Pro to, aby privilegovaná data přístupu měla smysl, je třeba také zamezit neoprávněným přístupům k datům uživatelům, kteří tato práva nemají. Toto je v aktuální verzi aplikace implementováno pouze z části.
Následně je třeba také zavést proces pravidelného přezkoumání přidělených přístupových oprávnění tak, aby odpovídala potřebám a aktuální situaci a neumožňovala přístup neoprávněným osobám.

#### 3.3.4.5 Nikdo nemůže revidovat a znovu schválit svá vlastní přístupová oprávnění

Zde existuje chyba v implementaci aplikace. Uživatel si s trochou šikovnosti může sám změnit svá přístupová práva. Tato část je třeba revidovat a upravit.

#### 3.3.4.6 Záloha dat a jejich šifrování

V současnosti se aplikace zálohuje na stejný server, kde je záloha databáze s citlivými údaji volně přístupná. Tento stav je velkým porušením bezpečnosti a je třeba jej napravit.
V aplikaci by mělo být zavedeno pravidelné a nejlépe automatické zálohování dat do bezpečného a šifrovaného úložiště.
Po té, co bude aplikace zálohována, je třeba také provádět pravidelné testy obnovy aplikace ze zálohy.

#### 3.3.4.7 Zabezpečení prostor, kde je aplikace fyzicky umístěna

Z důvodu lepšího monitoringu přístupů je důležité zavést a vést evidenci vstupů do místnosti, kde je aplikace fyzicky umístěna. Tato evidence by měla být zabezpečena proti možnému poškození či neoprávněné manipulaci.

#### 3.3.4.8 Postupy pro bezpečnou likvidaci

V okamžiku, kdy dojde k vyřazení datového média, které na sobě nese data z aplikace BOZP, je třeba jej bezpečně zlikvidovat. Pro tyto případy by bylo dobré zavést postupy pro bezpečnou likvidaci paměťových nosičů.

#### 3.3.4.9 Monitoring aplikace

S ohledem na využití kapacit by aplikace měla být alespoň částečně monitorována. Pro tuto potřebu je třeba zavést na server monitorovací nástroje, které by se o monitoring aplikace staraly a na případné bezpečnostní incidenty by byly schopny reagovat.
Tyto informace by pak měly být shromažďovány a uchovávány po dobu stanovenou zákonem. Měla by být také snadno prokazatelná jejich úplnost.

#### 3.3.4.10 Přezkoumávání bezpečnostních incidentů

Po každém zaznamenaném bezpečnostním incidentu a jeho úspěšném vyřešení by měl být tento incident přezkoumán. Na základě tohoto přezkoumání by pak mělo dojít k zavedení dodatečných bezpečnostních opatření, aby se předešlo opakování tohoto incidentu.
Z hlediska bezpečnosti by pak také měly být nastaveny a zavedeny postupy pro řešení bezpečnostních událostí a incidentů.

Kapitola 4

# Analýza dopadů

V této kapitole bych ráda zpracovala analýzu dopadů zavedení a nezavedení změn, které navrhuji. Analýzou se zabývám jak po finanční stránce, tak také po stránce právní a manažerské.

## 4.1 Finanční dopady

Součástí zadání mé diplomové práce je také analýza finančních dopadů zavedení a nezavedení mnou navrhovaných úprav portálu BOZP. V následující sekci se zabývám dvěma variantami, a to sice, jaké důsledky by mohlo mít nezavedení doporučení, které jsem navrhla v předchozí části. Také zde uvádím ekonomické zhodnocení zavedení změn.

### 4.1.1 Sankce GDPR

Velký strašákem všech firem jsou sankce vyplývající z nezavedení GDPR. V případě nezavedení, porušení nebo nepřipravenosti na toto nařízení hrozí organizacím velmi vysoké pokuty, které samo nařízení přesně definuje. V mnoha případech pak tyto pokuty mohou být pro firmy až likvidační.
V České republice je již několik let možné udělit pokuty za podobné prohřešky, které trestá GDPR. Jsou to především předpisy na ochranu soukromí a osobních údajů. GDPR hranici pokuty zvedá a zavádí podstatně vyšší možné sankce, než byly doposud možné. Maximální je výše 20.000.000 EUR, nebo 4 % z celkového ročního obratu. Uplatněna by pak měla být vyšší z obou možností a bude záviset na řadě faktorů, jako jsou především povaha, závažnost a délka porušování, počet poškozených subjektů osobních údajů a míra škody, kroky podniknuté správcem či zpracovatelem ke zmírnění škod, kategorie osobních údajů dotčené porušením a řada dalších.
Kromě těchto pokut mohou být správci či zpracovatelé osobních údajů navíc vystaveni žalobám podaným fyzickými osobami s nárokem na náhradu škody

v případě hmotné či nehmotné újmy.

#### 4.1.1.1 Provedení DPIA

Pro GDPR vyhlášku je velmi důležité provést DPIA. Její neprovedení, pokud se jedná o zpracování předmětu povinného DPIA nebo provádění nesprávným způsobem, může znamenat správní pokutu až do výše 10.000.000 EUR, nebo 2 % celkového ročního obratu, podle toho, která z hodnot je vyšší. V aktuálním českém ekvivalentu zákona je uvedena pokuta pouze 10.000.000 Kč, čímž se hrozba vysokého trestu prakticky vytrácí. Dosud však není jasné, jestli bude GDPR přijato s omezující výší, nebo tak jak je.

### 4.1.2 Náklady na vývoj a zavedení

Práce na implementaci úprav, a to včetně zkoumání možných variant a testování, činí zhruba 400 hodin. Práce na analýze právních důsledků i bezpečnosti a také zkoumání problémové domény a konzultace s pracovníky BOZP zabralo přibližně 300 hodin. Při poměrně nízké sazbě 400 korun na hodinu práce by tedy náklady na úpravu aplikace stály přibližně 280 000 korun. V těchto nákladech však nejsou zahrnuta zařízení, na kterých byl systém vyvíjen, elektrická energie, připojení k internetu a licence, což by si běžná firma účtovala. V sazbě jsou pak započteny náklady běžné pro OSVČ, tedy daň z příjmu a zdravotní a sociální pojištění.
Zavedení a následnou údržbu aplikace nelze vyčíslit. Systém je open-source, nicméně jeho předchozí verze již běží nad infrastrukturou Fakulty informačních technologií. Proto nelze vyčíslit přesnou částkou provoz a údržbu této aplikace.

## 4.2 Právní dopady

Velkou otázkou z pohledu GDPR jsou právní dopady. Pro většinu starších aplikací vstoupí GDPR v platnost oficiálně až od května 2018, zatím tedy neexistují právní precedenty. GDPR se stará především o práva uživatele. Jedním z takových práv je například možnost kdykoliv požádat o úpravu nebo kompletní vymazání svých zpracovávaných dat.
Další regulací je pak nemožnost sbírat údaje, ke kterým nebyl zpracovatel oprávněn a/nebo nejsou nezbytná pro účely aplikace.
Za velký právní dopad je možno považovat povinnost informovat o bezpečnostním incidentu do 24 hodin od zjištění kontrolní orgán státu a dále bezodkladně informovat všechny uživatele, jejichž data se ocitla v nebezpečí.
Většina právních dopadů byla shrnuta v předchozí kapitole této práce. Pokud by nebyla pravidla GDPR dodržena, mohla by Fakulta informačních technolo-

gií očekávat stížnosti a právní žaloby na porušení této regulace od dotčených osob.

## 4.3 Manažerské dopady

Dopady GDPR na management a řízení budou poměrně značné. Nadále již nebude možné předávat osobní údaje, a to ani v interních dokumentech, třetím osobám, které nebyly proškoleny v zacházení s nimi. Klíčovým dopadem pak bude samotné řízení bezpečnosti a zabezpečení informací.
Vzhledem k tomu, že za aplikací BOZP nestojí žádný projektový tým, který by se o ni dlouhodobě staral, nebudou manažerské dopady na ní nijak marginální. Pro další vývoj aplikace však již nebude možné používat zastaralá data. Tato data stále obsahují citlivé a osobní údaje bývalých a současných zaměstnanců a studentů školy. Pro účely dalšího vývoje bude třeba tato data anonymizovat a až po té je používat v lokálních kopiích aplikace.

### 4.3.1 Log management

Log management řeší přístup k zacházení s velkým množstvím logů vygenerovaných počítačem. Mezi ně například patří event logy, audit logy atp. Log management se pak stará o jejich sběr, agregaci, dlouhodobé skladování analýzu a reporting.
V případě aplikace BOZP jsou logy shromažďovány ve dvou složkách, jako u všech standardizovaných aplikacích psaných ve frameworku Nette. Těmito složkami jsou složky temp a log, které se nachází v aplikační složce. Ve složce temp je shromažďována cache aplikace a ve složce log pak logy rozdělené do několika souborů - exception.log, info.log. Kromě těchto souborů je při každém pádu aplikace na fatální chybě zaznamenána tato chyba do zvláštního html souboru.
Pro analýzu logů BOZP lze použít textový prohlížeč, pro případ html chyb pak webový prohlížeč. Ke každému logu se zaznamenává časová známka, pád aplikace je tak snadno dohledatelný podle času. Automatická analýza logů pro framework Nette není implementována a bylo by nutné si ji udělat celou od začátku. Pro účely aplikace BOZP je to zbytečné, aplikace generuje jen omezené množství logů, které je běžný člověk schopen zpracovat.

### 4.3.2 Řízení bezpečnostní dokumentace a rozhodování

Důležitou součástí managementu GDPR je řízení bezpečnostní dokumentace a rozhodování. Tento management se zabývá především bezpečností dat klientů a jejich právy ve vztahu ke zpracování osobních údajů.
Pro splnění podmínek GDPR z hlediska řízení bezpečnosti je třeba zavést organizační a technická opatření. Zde se jedná především o interní směrnice ochrany osobních údajů, vhodná bezpečnostní opatření, vedení záznamů

o zpracování osobních údajů a posuzování vlivu na soukromí. Je také třeba informovat úřady pro ochranu údajů o porušení bezpečnosti osobních údajů. Vzhledem k tomu, že Fakulta Informačních technologií spadá pod organizaci ČVUT, která řeší otázky bezpečnosti informací jednotně, není třeba - a bylo by to dokonce bezúčelné - vytvářet tyto interní dokumenty samostatně. Posouzení vlivu na soukromí bude také provedeno centrálně. Z hlediska rozhodování je pak třeba, aby správce aplikace podstoupil školení o bezpečnosti a zacházení s osobními údaji a řídil se doporučeními, které vydá Výpočetní a informační centrum ČVUT.

### 4.3.3 Role DPO

Po všech organizacích, které musí splňovat požadavky GDPR, tato vyhláška vyžaduje zřízení pozice DPO - Data Protection Officer. Pracovní náplní takového člověka pak bude usilování o dodržení obecného nařízení EU o ochraně osobních údajů a zajištění trvalého souladu s tímto nařízením ve všech klíčových činnostech organizace. Tato role ale není nová, GDPR jí pouze přidává nové činnosti a okruh vědomostí.
Povinností pověřence DPO je tedy především monitoring dodržování GDPR v organizaci. Dále by pak měl radit svým kolegům a spolupracovat s dozorovými úřady pro ochranu osobních údajů. Velmi důležitá je jeho počáteční povinnost nastavit jasné způsoby komunikace s interními i externími zúčastněnými osobami.
Vzhledem k tomu, že - stejně jako u předchozího bodu - FIT spadá v tomto ohledu pod ČVUT, měla by touto rolí pověřit zaměstnance přímo škola. Určený pracovník by pak měl mít přehled o systémech jak používaných na celé škole, tak i na jednotlivých fakultách a tedy i o aplikaci BOZP.

Kapitola 5

# Implementace

Tato kapitola se zabývá implementací mnou navržených změn v BOZP portálu. Změny jsou jak po stránce funkčnosti aplikace, tak také po stránce zabezpečení serveru samotného i zajištění jeho ochrany v případě útoku.

## 5.1 Funkční úpravy BOZP portálu

V této sekci popisuji změny, které jsem implementovala a nasadila v rámci funkčních úprav portálu BOZP. Jedná se především o bezpečnostní úpravy na základě analýzy GDPR a výstupu výsledků etického hackování. Dále také uvádím funkční úpravy, které byly vyžadovány pracovníky BOZP na konzultacích.

### 5.1.1 Update frameworku

Některé nově instalované moduly Nette vyžadují vyšší verzi tohoto frameworku. Aktuální aplikace byla napsaná ve verzi 2.3, jejíž podpora byla také ukončena v únoru 2017 a byla kompatibilní s verzí PHP 7.1. Pro modul emailu ale bylo třeba přejít na aktuální verzi Nette, kterou je 2.4. Ta vyžaduje PHP alespoň verze 5.6. Před implementací změn jsem tedy musela provést update Nette na novou verzi. Tento update vytvořil problémy v kompatibilitě, které bylo třeba odstranit. Jednalo se zejména o problémy s konfigurací, která se s přechodem na novou verzi poměrně výrazně měnila.
Další problémy a nefunkční stránky pak vznikaly například proto, že se metoda $this->invalidateControl('rows') s novou verzí měnila na $this->redrawControl('rows'). Tato metoda byla hojně používaná v gridech na mnoha stránkách aplikace.
Tyto a další problémy jsem vyřešila a aplikace BOZP tak získala náskok tím, že je implementovaná v nejaktuálnější verzi frameworku a může tak využívat jeho podporovanou a stále vyvíjenou verzi.

### 5.1.2 Statistiky a hromadné emaily

Problémem statistik byly především nedostatečné možnosti filtrování uživatelů. V původní verzi portálu nebylo možné uživatele vybrat na základě údajů o tom, kdo už daný kurz úspěšně dokončil. Tento problém byl vyřešen úpravami staré funkčnosti statistik tak, aby dávaly větší smysl.
Požadavkem pracovníků BOZP bylo také zavedení možnosti hromadné komunikace s uživateli. To bylo vyřešeno implementací nové funkce „Hromadný email“. Uživatelům, kteří ještě školení nesplnili, je možné poslat hromadný email přímo ze systému aplikace BOZP.

**Implementace** Pro hromadné odesílání emailů jsem zvolila nejjednodušší variantu, která se nabízela. Vzhledem k pokročilosti filtrování spamů a hromadných emailů, které je implementované do většiny serverů, které emaily přijímají, nemá v dnešní době smysl zakládat si vlastní mailový server. Vzhledem k tomu, že by nebyl zanesen jako důvěryhodný do žádného z anti-spam filtrů, pravděpodobně by emaily z něj téměř bez výjimky končily ve spam schránce.
Z tohoto důvodu jsem se po konzultaci se svým vedoucím práce rozhodla využít školní SMTP server pro odesílání emailů. Pro aktuální stav lze využít účet pracovníka BOZP, do budoucna by bylo ale vhodné zřídit pro BOZP na školním serveru individuální schránku, například bozp@fit.cvut.cz.
Velkým oříškem při implementaci hromadných emailů do portálu byl způsob, jakým by bylo možné zobrazit všechny příjemce a zároveň dát pracovníkovi BOZP možnost tento list modifikovat. V konečném řešení se sada emailů zobrazuje jako hodnota typu string, kde jsou maily oddělené čárkou. PHP7 pak má funkci, která umožňuje rychlé a efektivní rozdělení stringu do pole. Jedná se o metodu explode.

```
1  $emails = explode(',', $values['emails']);
2
```

Ukázka kódu 1: Použití hromadných emailů

Nette samotné pak má svou třídu, která umožňuje efektivně sestavovat a posílat emaily - Nette Mail.

### 5.1.3 Školení

Nedostatkem portálu v rámci sekce školení je především nedotažená logika. Pokud byl například student v prvním ročníku, portál nebral v úvalu, jestli již absolvoval školení vyhlášky padesát, která má platnost dva roky. Studenti, kteří nastupovali do prvního ročníku podruhé, tak museli školení vyhlášky 50/78Sb absolvovat znovu.
Mým původním záměrem bylo využít informace o absolvování předmětu, které se dají získat pomocí stávajícího KOS API ze školního informačního serveru.

Pomocí stejného systému probíhá import uživatelů do aplikace BOZP. Po prozkoumání dokumentace KOS API jsem ale bohužel zjistila, že nic, co by se dalo pro tento účel využít v něm není dosud implementováno.
Nejen, že se nedá zvenku zjistit, jaké předměty již daný student absolvoval. Z KOSu se dá sice zjistit, zda je konkrétní student přihlášený na konkrétní termín zkoušky předmětu, nedá se to však zjistit zpětně a také není možné zjistit známku.
V dokumentaci popisující KOS API je zmíněno, že v budoucnu bude možné pomocí těchto requestů získat více informací. Je tedy pravděpodobné, že zde stále probíhá vývoj a při příští úpravě aplikace BOZP by se již takovéto informace daly z KOSu vytáhnout.

### 5.1.4 Reklamace

V původním řešení portálu nebyla možnost reklamovat výsledek školení. Tato možnost byla přidána jako formulář v profilu studenta. Student tedy může jednoduchým kliknutím na tlačítko a následným potvrzením vyskakovacího okna vznést reklamaci na výsledek školení. Po této reklamaci je správci portálu nově odeslán automatický email s žádostí.
Další možnost interakce studenta s portálem byla přidána tak, že si student sám může zanést do systému to, že je aktuálně na stáži, nebo v zahraničí. Není tedy již třeba psát složitě emaily pracovníkům BOZP v případě, že se student nemůže z tohoto důvodu na školení dostavit.

### 5.1.5 Ostatní změny

V rámci dalších požadavků na změny v implementaci byly vzneseny tyto návrhy, které budou následně zakomponovány do výsledného programu.

- Při editaci profilu by měly být ověřovány všechny parametry, které uživatel vkládá.

- Drobná změna logiky při školení BOZP, přidání nového stavu - Test absolvován, nepodepsáno. Do tohoto stavu se student dostane v momentě, kdy bude mít absolvován test vyhlášky 50/78Sb, ale nebude mít podepsanou prezenční listinu. Stav ho bude vyzývat, aby se dostavil k podepsání.

- Na portálu by měly být spuštěné již dříve implementované nové gridy pro zobrazování dat. Je třeba jejich funkcionalitu ověřit a následně je nasadit.

- Udělení školení by mělo ukládat datum proběhlé akce, nikoliv aktuální datum.

- Pokud student napíše test znovu a nemá ještě propadlé datum dříve napsaného testu, test se aktuálně jeví, jakože je již podepsaný. To však nelze, student ho musí podepsat znovu. Tuto funkcionalitu je třeba upravit tak, aby odpovídala realitě.
- Student, který by měl splnit kterékoliv školení ale je aktuálně na stáži, nebo na zahraničním výjezdu má mít možnost tuto skutečnost uvést ve svém profilu. Dosud měl možnost toto změnit pouze správce BOZP.

## 5.2 Úpravy zabezpečení portálu

Následující sekce se zabývá úpravami zabezpečení portálu. Po nasazení těchto úprav by nemělo být již tak snadné narušit jeho bezpečnost. Vzhledem ke stále se měnící bezpečnostní situaci a novým hrozbám na internetu, nemůže být zabezpečení nikdy 100%, nicméně lze s jistotou říci, že portál BOZP není kritická infrastruktura a při náznaku jakýchkoliv úniků dat, nebo ohrožení je možné jej kompletně odstavit z provozu bez větších dopadů na zpracovávanou agendu.

### 5.2.1 Úpravy dle zjištění etického hackování

Problémy nalezené v rámci analýzy etického hackování jsou poměrně závažné a mohl by na ně být snadno proveden útok s cílem získat citlivá data. Zde jsou tedy implementované změny, které byly v rámci portálu zavedeny a nasazeny na nový server. Implementované změny vychází z mých návrhů uvedených v předchozí části.

**Uživatel si může změnit roli** V původní aplikaci bylo možné si změnit svou roli pomocí výměny parametrů ve formuláři. Framework Nette, použitý pro vývoj BOZP, implementuje celé zacházení s uživatelskými rolemi. Tyto role jsou používány i v této aplikaci. U některých akcí však kontrola oprávnění zcela chybí. Proto bylo relativně nekomplikované zavést kontrolu rolí i pro tuto funkci. Níže můžete vidět jednoduché ověření uživatelových práv k zacházení dané akci.

```
1   $roleChangeAllowed = $this->getUser()->isAllowed('User', '
      changeRole');
2   if($roleChangeAllowed ){
3     if(!$user){
4       $this->redirect('Homepage:');
5       return;
6     }
7   }
8
```

Ukázka kódu 2: Kontrola uživatelských rolí

**Je možné nahrát .php soubor a spustit ho** Nette má svou vlastní kontrolu nahrávaných souborů. Tu bylo třeba použít a aplikovat na jakékoliv nahrávané soubory, především pak na ty, které byly nahrávány uživateli stránky. Touto důslednou kontrolou je pak zakázána možnost nahrání jiného souboru než obrázku.

**Smazání článku na hlavní stránce může provést kdokoliv** Zde podobně jako u problému se změnou rolí uživatele stačilo ošetřit přístupová práva k odstraňování článků. Tato kontrola se již prováděla u všech ostatních akcí se články ale z nějakého důvodu byla u mazání pozapomenuta.

```
1   $articleRemoveAllowed = $this->getUser()->isAllowed('Homepage',
      'articleRemove');
2     if($articleRemoveAllowed ) {
3     if ($id) {
4       $this->articleModel->delete($id);
5     }
6   }
7
```

Ukázka kódu 3: Kontrola uživatelských rolí

**Uživatelé si mohou zobrazit a upravovat profily, které jim nepatří** V akci User:profile bylo v url jasně viditelné id a v případě, že jej přihlášený uživatel přepsal, mohl vidět profily ostatních a dokonce je i upravovat. Zde úprava tohoto problému spočívala jednak v kontrole, zda uživatel zobrazuje svůj profil a nebo, zda má právo zobrazovat profily ostatních (to může pouze administrátor).

```
1   $currentUserId = $this->getUser()->getId();
2   $renderUserAllowed = $this->getUser()->isAllowed('User', '
      otherProfiles');
3   if (($id == $currentUserId) || $renderUserAllowed) {
4     ...
5
```

Ukázka kódu 4: Úprava a zobrazení profilu

**Lze použít ukradené PHPSESSID** Bezpečnostní riziko tohoto problému nespočívalo v tom, že PHPSESSID je jediná kontrola pro identifikaci session. Hlavním problémem bylo, že se po celou dobu spojení uživatele se serverem nemění a tudíž je platné, dokud trvá přihlášení. Jednoduché zabezpečení tohoto atributu jsem zajistila tím, že se obnoví při každém novém načtení jakékoliv stránky a i kdyby ho někdo ukradl, bude mít velmi omezené možnosti práce. Další ochranou je automatické odhlášení, které na uživatele aplikuje školní SSO Shiboleth.

**Stránka s editací kurzu nepoužívá anti-CSRF token** Zde byl problém v zabezpečení formuláře. Framework Nette má funkci zajišťující bezpečnost formulářů. Všechny ostatní formuláře toto zabezpečení implementovaly, nicméně u formuláře s editací kurzu chybělo, tudíž ho stačilo doplnit.

**Existuje SQL Injection v některých parametrech orderBy** Tento problém se mi nepovedlo na lokální kopii aplikace replikovat. Vzhledem k tomu, že testování probíhalo v loňském roce nad původní aplikací se domnívám, že byly ještě použity staré datagridy k zobrazování dat. Nové datagridy, které byly do aplikace implementovány Hanou Kozákovou[12], nebyly v té době ještě nasazeny na serveru. Nový datagrid se od starého liší mimo jiné také tím, jak ošetřuje vstupy a SQL Injection. Tento problém by tedy měl být eliminován nasazením nové verze portálu.

**Uživatel může zobrazit všechny události uložené v databázi jen uhádnutím id** V tomto případě se nejedná přímo o bezpečnostní riziko, ale spíše o jakési vytěžování dat. Uživatel může ručně změnit id události při jejím zobrazení, může tedy zobrazit všechny události uložené v databázi. To není žádoucí, pokud je událost uživateli skryta, je k tomu jistě důvod (například se na termín již nemá jít přihlašovat).
Tento problém vyžadoval poněkud komplexnější změnu logiky zobrazování událostí, protože tento problém dosud nebyl nijak ošetřován. Běžný uživatel si po úpravě smí zobrazit pouze ty události, které odpovídají jeho skupině, druhu školení a jsou aktuální. Jediná role, která může zobrazit všechny akce, je administrátor.

**Problémy s prezenterem typu tréninku** Toto nebyl ve skutečnosti problém bezpečnosti, ale funkcionality portálu. Byl opraven a funguje bez větších zádrhelů.

**Problémy s přihlašovací stránkou v angličtině** Přihlašovací stránka v angličtině nebyla správně přesměrovaná na přihlašování pomocí systému Shibboleth. Po opravě tohoto problému začalo vše v pořádku fungovat.

**Na serveru jsou zastaralé administrátorské nástroje** Na adrese `<https://bozp.fit.cvut.cz:8090/>` byly k nalezení celkem čtyři staré administrátorské nástroje. Ani jeden z nich již nebyl funkční vzhledem k nekompatibilitě a tomu, že nebyly na aplikaci nijak napojeny. Pro jistotu jsem je všechny odstranila.

**Do nástroje Adminer lze přistupovat z internetu** Pomocí zadání přesné adresy souboru se na starém serveru dal spustit Adminer - nástroj pro přihlá-

šení do databáze. Všechny soubory, ke kterým nepotřebuje uživatel z venku přistupovat, byly uschovány.

**Na serveru je záloha databáze přístupná z internetu** <https://bozp.fit.cvut.cz/sql/bozp3.sql> Tento soubor byl odstraněn. Nová záloha databáze byla vytvořena a je skladována na jiném, bezpečnějším místě. Nedávalo smysl ji uchovávat na stejné struktuře ani jako zálohu pro případ pádu serveru.

**PHPSESSID nemá bezpečnostní tag** Pravděpodobně z toho důvodu, že byla stránka vyvíjena na lokáním počítači bez https, nebyl přidaný bezpečnostní parametr pro PHPSESSIONID. Stačilo doplnit cookie_secure true parametr do konfigurace, aby byl tento bezpečnostní problém odstraněn.

**Skript k vytváření uživatele** Skript pro vytvoření uživatele je zastaralý a nepoužívaný. Netvořil bezpečností riziko, ale byl k ničemu, byl tedy odstraněn.

**Na web serveru je možné provést fingerprint** Pro bezpečnost serveru není dobré, pokud je možné provést na serveru zjištění verze systému a frameworku apache. Tomu bylo třeba zamezit. Na serveru Apache ve verzi dva je možné zakázat zobrazování citlivých informací pomocí tohoto nastavení.

```
1  ServerSignature Off
2  ServerTokens Prod
3
```

Ukázka kódu 5: Nastavení serveru

### 5.2.2 Úpravy pro splnění GDPR

V samotném programu nebylo třeba v souvislosti s GDPR upravovat příliš mnoho věcí. Většina problémů se vyřešila zapracováním všeobecných pravidel bezpečnosti. Mnoho dalších bylo splněno také opravou bezpečnostních děr, které byly objeveny během bezpečnostního auditu etického hackování.

### 5.2.3 Vyhodnocení bezpečnosti před a po úpravách

Před začátkem zpracování úprav pro splnění GDPR jsem provedla analýzu zabezpečení aplikace dle návodů obsažených v metodice, kterou poskytlo Výpočetní a informační centrum ČVUT všem vlastníkům aplikací na FIT ČVUT. Analýza nedopadla pro aplikaci příliš pozitivně, pro požadavky GDPR byla aplikace zabezpečena jen z 54,36 %. Z celkového počtu 11 povinných opatření byla 4 plněna, 2 plněna částečně a 5 neplněno.
Tu samou analýzu jsem provedla i po implementaci zabezpečení a výsledek

byl výrazně lepší než před úpravami. Aplikace byla zabezpečena z 76,36 % pro splnění GPDR a z počtu 11 povinných opatření bylo 11 plněno. Velký význam měla především formalizace procesů, kterou vyhláška vyžaduje.

### 5.2.4 Formalizace procesů pro splnění GDPR

Pro splnění GDPR bylo také nutné formalizovat procesy, které se týkaly práce s daty. Zde jsou vyjmenovaná doporučená opatření, která byla dotažena do finální podoby ve spolupráci s BOZP pracovníky.

#### 5.2.4.1 Přidělení odpovědností

Zde se jednalo především o rozhodování o aplikaci, určování podmínek jejího běhu a rozšiřování a technickou správu. Bylo rozhodnuto o přiřazení těchto rolí jednotlivým pracovníkům BOZP:

- **Business vlastník**: Tuto roli bude vždy vykonávat nejvyšší pověřený pracovník BOZP. Vzhledem k tomu, že na FIT ČVUT je takový pracovník vždy pouze jeden, je tato osoba určena jednoznačně. Kontaktní údaje na něj jsou uvedeny v systému BOZP.
- **Technický správce**: Tato role bude vždy určovaná vedením fakulty. Platnost role je přezkoumávána vždy na začátku školního roku, kdy se řeší všechny úvazky. V současné době to bude vedoucí projektů, které měly za úkol vývoj aplikace BOZP.

#### 5.2.4.2 Pravidla pro přidělování přístupů k aplikaci

Přidělování práv na základě dobré vůle správce aplikace není ideální přístup. Z pohledu GDPR v aplikaci BOZP chybí jakákoliv formální definice rolí a jejich přidělování.
Zde je tedy popis rolí a míst, kam mají přístup:

- **Admin** - administrátorský účet, má přístup ke všem funkcím všech obrazovek.
- **Guest** - role nepřihlášeného hosta na stránce, má přístup pouze k přihlášení, hlavní stránce a FAQ.
- **LoggedIn** - přihlášený uživatel, nadrole, která se stará o společné přístupy všech následujících rolí.
- **Student** - student, má přístup jen do sekcí týkajících se jeho akcí na BOZP portálu.

- **Employee** - zaměstnanec fakulty, může přistupovat k obdobným funkcím jako student.
- **Supervisor** - pracovník BOZP, má přístup ke schvalování a plánování kurzů a k dalším funkcím potřebným k vykonávání svých povinností.

#### 5.2.4.3 Formalizovaný proces odebírání přístupů

Vzhledem k povaze aplikace BOZP, aktuální proces, při kterém přístup dostane ten, kdo má platný účet v KOSu - školním informačním systému, plně dostačuje k manipulaci s účty.
Vzhledem k tomu, že není možné někomu jen tak odebrat přístupová oprávnění, pravděpodobně by bylo vhodné zavést pro případy kritické nouze - kdy byl například účet ukraden a použit k útokům - novou roli. Tato role by se měla jmenovat „blocked“ a mít přístup pouze k hlavní stránce aplikace. V aktuální implementaci může tuto roli bez problémů nahradit role „guest“, která funguje obdobně. Účtu, který by bylo třeba zablokovat, je pak možné tuto roli změnit přímo v administračním prostředí databáze.
Vzhledem k tomu, o jak jednoduchý a velmi řídký případ užití aplikace se jedná, stačí pro aktuální stav tento postup. Do budoucích verzí by bylo příjemné zabudovat možnost blokovat a odblokovávat uživatele přímo z aplikace pomocí administrátorského, nebo správcovského účtu.

#### 5.2.4.4 Privilegovaná přístupová oprávnění

V aktuální verzi aplikace jsou privilegovaná přístupová práva přidělovaná na základě rozhodnutí správce aplikace.
V implementaci jsem opravila některá povolení, která byla chybná, nedávala smysl, nebo nebyla vůbec obsažena. Tedy například už nikdo kromě administrátora nemůže zobrazit a upravovat profil nikoho jiného.
Dále bych navrhovala, aby byla každý rok po té, co vedení fakulty rozhodne o úvazcích pracovníků BOZP, přezkoumána privilegovaná přístupová práva k aplikaci. Takto by se mělo zamezit tomu, aby nepovolané osoby stále měly přístup k aktuálním datům uživatelů. O každém tomto přezkoumání by měl vzniknout záznam, který by udával, komu byla privilegovaná práva odebrána, přidělena a ponechána a proč.

#### 5.2.4.5 Záloha dat, jejich šifrování a postupy pro bezpečnou likvidaci

Server aplikace je virtuálním serverem, který běží ve fakultním cloudu. Je tedy poměrně snadné jej celý zazálohovat a následně ze zálohy opět spustit. Vzhledem ke skutečnosti, že data obsažená v aplikaci nejsou kritická a mají i svou papírovou podobu, není třeba je zálohovat velmi často. Vzhledem k omezené kapacitě serveru by to mohlo znamenat jeho pravidelné zablokování.

Po posouzení dopadů ztráty dat na aplikaci by bylo nejvhodnější celý server zálohovat přibližně jednou za semestr, a to před tím, než začnou probíhat kurzy BOZP, tedy na konci září a na konci února. Záloha databáze by pak měla probíhat každý týden.
Protože data v aplikaci se mění poměrně pravidelně, a to nejvíce především v průběhu zimního semestru, pro aktuální stav postačí roční zálohování. K záloze databáze by v budoucích implementacích bylo vhodné zavést skript, který se o ni postará automaticky.
Zhruba jednou za půl roku by také bylo vhodné po zazálohování systému provést test jeho obnovy ze zálohy. Ten by měl být prováděn ručně a administrátor by měl následně otestovat, že aplikace kompletně funguje.
Pokud bude záloha dále skladovaná ve fakultním cloudovém systému, mělo by být zaručeno, že po její likvidaci skutečně proběhne vymazání všech dat. Pokud by bylo nutné zálohu někam přenést, je třeba, aby byla přenášena pomocí šifrovaného média a to bylo následně buď zničeno, nebo byla data velmi pečlivě odstraněna.

#### 5.2.4.6 Zabezpečení prostor, kde je aplikace fyzicky umístěna

Vzhledem k tomu, že aplikace je umístěna v cloudu, není v silách provozovatele aplikace zajistit monitoring prostor, kde je fyzicky umístěna. V tomto případě je tedy nutné se přesvědčit, že provozovatel cloudové služby tyto přístupy monitoruje a prostory umístění serverů jsou zabezpečeny proti případnému neoprávněnému vstupu.

#### 5.2.4.7 Monitoring aplikace a bezpečnostní incidenty

Po domluvě s pracovníky BOZP a se správcem aplikace bylo rozhodnuto, že zatím nebude přistoupeno k aplikaci monitorovacích nástrojů na server. K tomuto kroku bylo přistoupeno zejména proto, že tyto nástroje jsou výpočetně náročné a málokdy mají opensource licenci.
K aktuálnímu monitorování aplikace stačí logy, do kterých je zaznamenávána veškerá aktivita, která se na serveru děje. Tak jako má samotná aplikace své logy, tak má také webový server Apache své podrobné logy, kde zaznamenává činnosti, které se na něm dějí. K podrobnému monitorování dění v aplikaci a na serveru lze použít tyto logy.
Vzhledem k tomu, že tyto logy jsou shromažďovány na serveru a uchovávány zatím bez jakéhokoliv mazání, je snadno prokazatelná jejich úplnost. S ohledem na zákon by logy o všech činnostech v aplikaci měly být uchovávány alespoň rok.
Pokud se pomocí těchto monitorovacích nástrojů povedlo zachytit bezpečnostní incident a bude úspěšně zabráněno narušení bezpečnosti, je třeba informace o tomto incidentu přezkoumat. Na základě tohoto přezkoumání pak musí správce aplikace přijmout opatření, která zabrání opakování incidentu

v budoucnu.
Bezpečným a jistým řešením incidentů na serveru BOZP je jeho dočasné odstavení z provozu. Pokud by tedy došlo k jeho napadení, je třeba zablokovat k němu přístup pomocí bezpečnostních protokolů serveru Apache, pod kterým aplikace běží.

## 5.3 Nastavení průběžné integrace

Dle analýzy, kterou jsem provedla v předchozích částech práce, se pro průběžnou integraci portálu BOZP nejlépe hodí použití nástroje GitLab Runner. Pro jeho použití je třeba, aby byl celý projekt verzován v systému GIT. Tento je aktuálně používán a verze zrojového kódu jsou ukládány na školní server GITLab.
Pro správné fungování nástroje GitLav Runner je třeba pečlivě nastavit YAML soubory, které se budou spouštět pokaždé, když některý z vývojářů nahraje novou verzi na GitLab pomocí operací git commit a git push.
Jedná se tedy především o soubor .gitlab-ci.yml, který celou průběžnou integraci řídí. Dále pak v projektu ve složce scripts tři skripty, before-install.sh, after-install.sh a makedeb.sh. Tyto tři skripty se spustí před instalací softwaru na server, po ní a následně vytvoří balíček, který je možné použít dále.
Pro správné fungování těchto skriptů je třeba mít na testovacím i produkčním serveru uživatele gitb, který se bude moci přihlašovat pomocí ssh klíče bez hesla. Bezpečnější přihlašování přes ssh klíč s heslem není pro tento automatický skript možné a zároveň by nemělo význam, protože heslo by muselo být tak jako tak někde uloženo.

### 5.3.1 Před instalační skript

Skript, který se spouští před instalací balíčku, je pro účely BOZP jen čistící skript. Vytvoří příslušnou složku pro aplikaci a přejde do ní. Následně pak vymaže staré logy, cache aplikace a webovou cache.

```
1  set -e
2  mkdir -p /var/www/bozp
3  cd /var/www/bozp
4  rm -rf temp/ log/ www/webtemp/
5
```

Ukázka kódu 6: Skript před instalací

### 5.3.2 Po instalační skript

Skript spouštěný po instalaci balíčku je o něco složitější než ten, který se pouští před, nicméně, protože portál BOZP nemá žádné zvláštní služby, které by byly potřeba pro jeho běh, není třeba mít tento skript složitý. Níže je možné vidět výňatek toho nejzajímavějšího.

```
1  a2enmod ssl
2  a2enmod rewrite
3
4  mkdir -p temp/cache/ log/ www/webtemp/
5  chown -R www-data:www-data temp/ log/ www/webtemp/
6
7  rm -f /tmp/bozp*
8  rm -f /tmp/package-*
9
10 systemctl restart apache2
11
```

Ukázka kódu 7: Skript po instalaci

Vzhledem k tomu, že při průběžné integraci není nikdy jisté, jak vypadal server před instalací, je nutné počítat i s možností, že je aplikace na server instalována poprvé, nebo že na něm běží její jiná, konfliktní verze. Proto je třeba nejprve nastavit mód SSL a Rewrite pro webový server Apache.
Dále je pak důležité ve složce aplikace vytvořit nové složky pro ukládání dočasných dat, jako jsou logy a cache.
Následujícím krokem je pak promazání potencionálně problematických složek a souborů, které by mohly způsobit nefunkčnost aplikace.
Vzhledem k tomu, že jsme v předchozích krocích měnili nastavení serveru Apache, je třeba jej také restartovat, aby si načetl novou konfiguraci.

### 5.3.3 Skript pro vytvoření balíčku

Skript pro vytvoření instalačního balíčku má několik důležitých součástí. V této části textu ho podrobněji rozebírám, ukázka je pak pouze výňatek z celého kódu.

```
1  cp -r !(dist|external|services|tests|makedeb.sh) dist/var/www/
     $projectname
2
3  find dist/var/www/$projectname -depth -name '.git' -exec rm -rf
     '{}' \;
4
5  cp app/config/example.bozp.conf dist/etc/apache2/sites-available
     /
6
7  version=$(cat app/config/config.neon | grep -P -o '(?<=version:
     '\'')[0-9.]+')
8  iteration=$(date +%s)
9
10 fpm -s dir -t deb -C dist -n $projectname \
11 -m "email" \
12 --before-install scripts/before-install.sh \
13 --after-install scripts/after-install.sh \
14 -v "$version" \
15 --iteration "$iteration" \
16  -d "systemd" \
```

```
17    -d "apache2" \
18    -d "php" \
19    -d "mysql-server" \
20    -d "curl" \
21    -d "composer" \
22    ...
23
```

Ukázka kódu 8: Skript pro vytvoření instalačního balíčku

Skript pro vytváření instalačního balíčku si nejprve vytvoří složky, do kterých pak nakopíruje projekt. V kódu výše můžeme vidět, že následně z projektu odebere to, co není v instalačním balíčku potřeba, tedy například testy, distribuční balíček, nebo sám sebe.
Následně je třeba také odstranit soubory gitovského repositáře, protože ty jsou většinou velké a je naprosto zbytečné aby byly součástí instalačního balíčku. Z pohledu bezpečnosti nasazení na produkční server je dokonce nežádoucí, aby byly tyto soubory v projektu zachovány, protože prozrazují historii vývoje a změny kódu.
Jako následující krok je zkopírován konfigurační soubor pro webový server Apache. Ten je důležitý zejména proto, že je třeba počítat s tím, že balíček může být nasazován i na úplně čerstvě instalovaný server. Je pak tedy také důležité, aby se všechna nastavení provedla automaticky a nebylo třeba zásahu člověka, při kterém může dojít k chybám.
Po shromáždění všech důležitých souborů a pročištění balíčku pak skript balíček vytvoří, přidá mu příslušnou verzi, číslo sestavení a především závislosti, které je třeba na serveru mít pro správné fungování.

### 5.3.4 Nasazení průběžné integrace

Pro to, aby byla průběžná aplikace nasazena na server, je třeba provést jednoduchá opatření. V GitLabu je třeba tuto integraci povolit a nastavit adresu testovacího a produkčního serveru. Dále je třeba tam také přidat ssh klíč, kterým se bude uživatel gitb na server přihlašovat.
Pro samotné správné fungování a běh integrace po každém git push je pak třeba ke zdrojovým souborům adresáře přidat také soubor .gitlab-ci.yml. Tento soubor popisuje nastavení GitRunneru při nasazování na server. Zejména vytvoří složky pro budoucí instalaci. Také zajišťuje připojení na server pomocí klíče a zkopírování správných složek ze svého repositáře.
Přidáním tohoto souboru a nastavení příslušných služeb GitRunneru je tak nasazení průběžné integrace dokončeno a funkční.

## 5.4 Nasazení a zabezpečení serveru

Součástí instalace bylo pro mě také nasazení aplikace na server a jeho zabezpečení. Tato část zabezpečení je velmi důležitá jak z pohledu počítačové

bezpečnosti, tak z pohledu GDPR.
Vzhledem k předchozím zkušenostem s nasazováním aplikací do produkčního běhu jsem se rozhodla nejprve provést úplný čerstvý build serveru na testovacím stroji a nasazení aplikace na něj. Následně pak bylo možné postupovat stejně při nasazení a reinstalaci serveru na produkční server.
Testovací server bylo třeba postavit tak jako tak, protože bylo nutné mít testovací prostředí pro potřeby průběžné integrace, kterou podrobněji rozebírám v předchozích částech práce.

### 5.4.1 Problémy zabezpečení předchozího serveru

V této části se zabývám těmi nejvýraznějšími problémy zabezpečení starého serveru. Byly to především zastaralý operační systém, nezabezpečené SSH a starý web server.

#### 5.4.1.1 Zastaralý operační systém

Na předchozím produkčním serveru byla hlavním problémem, který způsoboval bezpečnostní chyby, zastaralost. Byl zde nasazen linuxový operační systém Debian GNU verze 6.0.10 - Squeeze. Tato verze Debianu byla vydána 6. ledna 2011. Takzvané datum „End of life“ - kdy přestává tato verze dostávat důležité, především bezpečnostní updaty - pak nastalo 31. května 2014. Příčinu toho, proč na serveru běžel takto starý operační systém, bych viděla především v tom, že je jen velmi obtížné jej updatovat na aktuální verzi. Pro tento krok je nutné celý systém přeinstalovat a zabere to i několik hodin uvést znovu vše do provozuschopného stavu.

#### 5.4.1.2 Nezabezpečené přihlášení

Velkou bezpečnostní zranitelností z pohledu síťového byla možnost přihlásit se na server jako root přes ssh, a to pouze pomocí hesla. Důležitou zásadou síťové bezpečnosti je nenechávat nikdy možnost přihlašovat se jako root přes ssh.
K běžné práci na serveru bohatě stačí mít uživatelský účet, který má přístup ke složkám aplikace a může je upravovat. Pokud je to nutné, pak by tento měl mít buď tzv. „sudo právo“, tedy že po opětovném zadání hesla je možné provést jednorázové činnosti, které vyžadují vyšší oprávnění - například restart apache. Ani to však není ideální varianta. Pokud správce serveru potřebuje provést nějakou činnost s právy roota, měl by se jako již přihlášený na serveru na tento účet přehlásit.
Přistupovat v dnešní době k serveru přes ssh pouze s použitím hesla je zbytečně riskantní. Heslo může být velmi snadno prolomeno, zvláště pokud je slabé, nebo slovníkové.

#### 5.4.1.3 Zastaralý Apache server

Na starém serveru běžel Apache server verze 2.2.16 - edice pro operační systém Debian. Tento server byl poprvé vydán 1. prosince 2005 a jeho podpora byla definitivně ukončena 7. listopadu 2017. Tento server tedy již nedostává žádné nové záplaty na své zranitelnosti. Existuje také několik známých zranitelností, které byly již v této verzi nalezeny a nejsou opraveny. Mohly by tak ohrozit celou aplikaci a její bezpečnost.

### 5.4.2 Postup buildu serveru

Pro build nového serveru bylo třeba udělat následující kroky:

1. **Instalace operačního systému** Jako první bylo třeba vybrat a nainstalovat nový operační systém. Vzhledem ke snaze udržet kontinuitu mezi servery, jsem vybrala linuxový operační systém Debian GNU, verze 9.4 - Stretch. Je to nejčerstvější stabilní verze tohoto operačního systému. Tato verze byla vydána 17. června 2017 a ukončení její podpory se plánuje přibližně na červen 2020. Po té bude třeba server znovu přeinstalovat. Instalace tohoto systému nevyžadovala žádná zvláštní nastavení.

2. **Instalace Open SSH serveru** Pro snadný vzdálený přístup k serveru bylo třeba na něm nainstalovat a zprovoznit Open SSH server. Následně jsem na něm zakázala přístup pouze pomocí hesla a přístup pomocí root účtu, a to úpravou soubory /etc/ssh/ssh_config, kde jsem zakomentovala řádky RSAAuthentication a PasswordAuthentication. Následně bylo nutné vygenerovat klíč, který by bylo možné použít pro autentizaci uživatele. Ke generování jsem zvolila PuTTy Key Generator a vytvořila jsem zde klíč šifrovaný pomocí RSA s počtem bitů 2048 a zabezpečený heslem. Veřejnou část klíče jsem následně vložila do složky /.ssh/ do souboru authorized_keys. Pro přihlášení se k serveru vzdáleně je tedy teď nutné použít tento SSH klíč.

3. **Instalace Apache serveru** Na server bylo dále třeba nainstalovat důležité komponenty pro správný běh webu. Nainstalovala jsem tam tedy Apache server verze 2.4.25. Pro tento server je klíčové nastavení, kterému se věnuji v jedné z dalších částí.

4. **Instalace PHP** Pro běh frameworku Nette je nutné PHP. Vzhledem k tomu, že jsem se během implementace rozhodla o update Nette na projektu na nejnovější verzi, není třeba se obávat i nejnovější verze PHP. Protože na operačním systému Debian si nelze vybírat verzi instalovaného balíčku, nainstalovala jsem tedy verzi, která byla k dispozici a to PHP 7.0.27.

5. **Instalace databáze** Aplikace BOZP používá pro svůj běh databázi MySQL. Po té, co byla nedávno rozdělena majitelem (Oracle), již MySQL nefunguje pod svým původním názvem, ale jmenuje se MariaDB. Tato databáze je nadále open source a může být tedy využívána bez omezení. Na serveru jsem nainstalovala její verzi 10.1.26, což je nejaktuálnější verze.

6. **Nastavení SSL** Jako poslední krok před samotným nasazením aplikace na server bylo třeba uvést do chodu SSL zabezpečení. Pro aplikaci, kam se uživatelé přihlašují, je to v dnešní době nutnost. Je to také nutné, protože zde dochází ke komunikaci se školním systémtm identit Shibboleth a mohlo by tedy dojít k narušení tohoto spojení a podvrhnutí přihlašovacích údajů, či jejich zcizení.

#### 5.4.2.1 Nastavení Apache serveru

Server Apache bylo třeba nastavit tak, aby veškerá příchozí spojení přesměrovával na HTTPS a zabránil tak spojení přes HTTP, které je zranitelné. Z pohledu bezpečnosti je pak zajímavá především následující část konfigurace HTTPS VirtualHost (zde uvedená neúplná).

```
1  <VirtualHost *:443>
2    DocumentRoot /var/www/bozp
3    SSLEngine on
4    SSLProtocol all -SSLv3 -SSLv2
5    SSLHonorCipherOrder on
6    SSLCipherSuite HIGH:!aNULL:!MD5
7
8    Header always set Strict-Transport-Security "max-age=31536000;
     includeSubDomains; preload"
9
10   <Directory /var/www/bozp>
11     Options FollowSymLinks
12     AllowOverride all
13     Require all granted
14   </Directory>
15 </VirtualHost>
16
```

Ukázka kódu 9: Nastavení Apache serveru

V konfiguraci můžeme vidět poměrně striktní nastavení přístupů ke složce s aplikací. Původní nastavení bylo ještě přísnější, byl zde AllowOverride nastaven na none. To však bránilo aplikaci ve fungování a muselo být opět povoleno.
Dalším zajímavým nastavení je pak Strict-Transport-Security. Toto zabezpečení funguje tak, že po prvním připojení k serveru si prohlížeč uloží cookie, která mu udává, že tato aplikace běží pouze na HTTPS. Nikdy se tak již na dané doméně nepokusí připojit na HTTP, vždy půjde přímo na HTTPS,

aniž by se musel dotazovat Apache serveru. Toto opatření zmenšuje riziko útoků typu „man in the middle“, kdy je napadena HTTP verze stránky, která může například přesměrovávat uživatele na jiný web.
Je také dobré si všimnout zakázaných zastaralých protokolů SSL verze 2 a 3. Povoleny tak zůstaly TLS protokoly 1.0, 1.2 a 1.3. Protokol verze 1.0 je také již překonaný a nepatří k nejbezpečnějším, jeho odstraněním bychom ale ztratili možnost komunikovat s uživateli, kteří používají starší prohlížeče, například Internet Explorer 8 pod Windows XP, nebo Android až do verze 4.3. Pro tuto chvíli je ještě tento protokol ponechán. Do budoucna bych doporučila jej zakázat, protože zařízení s prohlížeči, které ho používají bude ubývat.
V neposlední řadě také můžeme vidět, že jsou zde nastaveny moderní šifry, které má SSL na serveru Apache používat.

#### 5.4.2.2 Nastavení SSL

Pro vydání certifikátu je zažitý proces potvrzující autority. Aktuálně se ale čím dál více do praxe dostávají certifikáty služby Let's encrypt. Tato služba je automatizovaný proces, který pravděpodobně v budoucnu částečně nahradí nutnost složité ruční tvorby, ověřování, podepisování a obnovování certifikátů autoritou. Služba poskytuje zdarma ověřené doménové certifikáty. Pro testovací server tento certifikát stačí. Pro produkční pak bude použit certifikát vydaný pro starý server certifikační autoritou používanou FIT ČVUT.

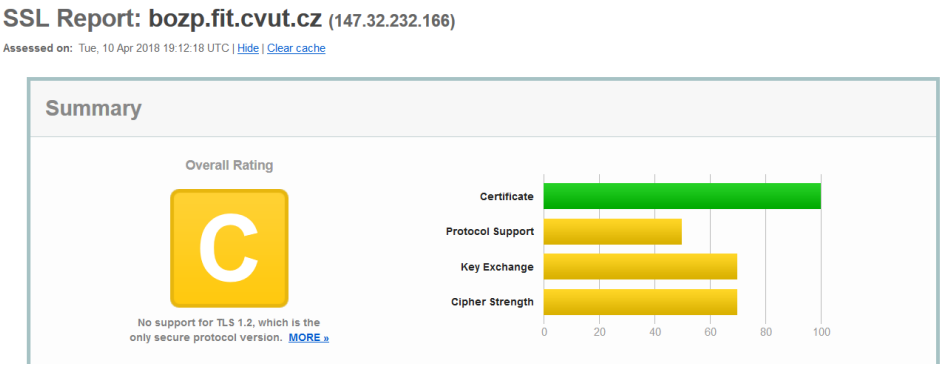
### 5.4.3 Srovnání starého a nového serveru

Ke srovnání úrovně zabezpečení před reinstalací a po ní jsem použila především nástroj SSL labs. Ten testuje zabezpečení všech prvků přístupu k serveru pomocí SSL. Také testuje přístup z různých zařízení a různé známé typy útoků na server.

#### 5.4.3.1 Výsledky SSL labu před reinstalací

Původní server nevyšel z testu na SSL labs příliš dobře, dostal známku C.
Hlavními důvody, proč získal tak nízké hodnocení, byly:

- Server podporuje slabé parametry pro výměnu klíčů při použití Diffie-Hellman šifry.
- Server podporuje pouze staré protokoly ale ne aktuálně nejlepší TLS 1.2
- Server přijímá šifru RC4 pouze pomocí starých protokolů
- Server nepodporuje Forward Secrecy u použitý prohlížečů
- Server nepodporuje šifrovací sady Authenticated encryption (AEAD)




Obrázek 5.1: Výsledné hodnocení SSL Labu původního serveru

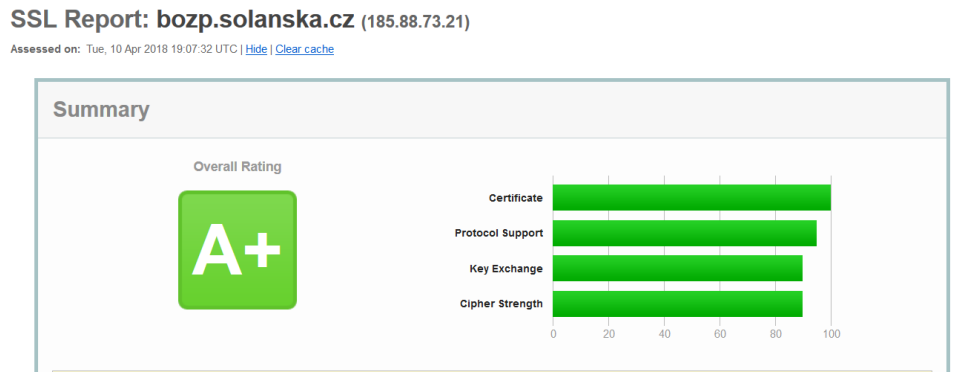
Tyto problémy byly způsobeny především zastaralým operačním systémem i serverem Apache. Aplikace na tyto součásti nemá žádný vliv.
Nedostatečná ochrana forward secrecy je poměrně důležitým problémem. Pokud by totiž někdy v budoucnu bylo ukradeno a rozšifrováno heslo či klíč, které server používal pro komunikaci, mohla by být veškerá komunikace rozšifrována zpětně.

#### 5.4.3.2 Výsledky SSL labu po reinstalaci

Jak je možné vidět na následujícím obrázku, po reinstalaci serveru a zavedení nových komponent dostal server hodnocení A+, což je nejvyšší možné ohodnocení, které SSL labs udělují.
Toto hodnocení lehce narušuje povolení šifer RSA s AES 256 a 128, které jsou všeobecně považovány za slabé, nicméně jsou v rámci protokolu TLS 1.2 používány i jiné silnější šifry.
Jediný prohlížeč, kterým se již nadále nebude možné k aplikaci připojit, je Internet Explorer na operačním systému XP a klienti používající Javu 6u45. Tyto technologie jsou již však tak zastaralé, že je možné je zanedbat. Mezi uživateli BOZP bude jejich procento opravdu mizivé, ne-li nulové.

## 5.5 Build a nasazení na produkční server

Předchozí část nasazení a buildu aplikačního serveru se týkala pouze testovacího serveru, který byl zaveden jako virtuální server sloužící pro testování aplikace. Všechny kroky bude potřeba replikovat také na produkčním serveru, a to po jeho plném zazálohování. Toto nasazení musí proběhnout před tím, než vstoupí v platnost vyhláška GDPR a je tedy plánované na začátek května letošního roku.




Obrázek 5.2: Výsledné hodnocení SSL Labu po reinstalaci serveru

## 5.6 Doporučení

Pro nasazení nové aplikace, její údržbu a další rozvoj mám doporučení především v těchto oblastech:

### 5.6.1 Zabezpečení aplikace

Při psaní kódu v budoucnu je vhodné dodržovat jednotný styl. Je také důležité aby byla dodržována všechna doporučení Nette. K frameworku Nette je při jeho vydání přiložen také soubor doporučení, kterých by se měl každý vývojář držet. Jedná se především o zásady, jejichž dodržením se vyhnete zbytečným problémům. Příkladem těchto problémů je pak například velká část kapitoly o výstupech z etického hackování.
V oblasti internetové bezpečnosti je třeba dbát také na aktuální zabezpečení serveru na kterém aplikace běží. Často jsou na nových i starých systémech nacházeny zranitelnosti, které mohou vést ke ztrátě kontroly nad nimi, nebo k únikům dat. Proto je třeba provádět aktualizace všech komponent celého systému na pravidelné bázi.
Z těchto důvodů bych pro budoucí běh aplikace navrhovala zavést proces, díky kterému bude správce serveru pravidelně kontrolovat zabezpečení (například pomocí SSL labs) a zavádět nová opatření ve prospěch bezpečnosti serveru. Také by měl na pravidelné bázi dělat aktualizace všech komponent.

### 5.6.2 Zabezpečení GDPR

Pro zabezpečení plného splnění GDPR by bylo do budoucna třeba především anonymizovat data, se kterými se pracuje při vývoji nových komponent aplikace. Aktuálně jsou používána data pouze zastaralá, ale reálná, což je v rozporu s legislativními změnami, které GDPR zavádí.
Pro další účely GDPR by bylo třeba také zavést přesné postupy pro reakci

na bezpečnostní incidenty a lepší management logování událostí v aplikaci.
Ne příliš dostatečné je také zálohování, které je aktuálně na serveru prováděno. Bylo by třeba zálohovat jak data, skladovaná v databázi, tak také samotný server. Vzhledem k tomu, že se jedná o cloudové řešení serveru, neměla by pravidelná záloha představovat větší problém.

### 5.6.3 Testování

Aplikace BOZP obsahuje automatické testy zavedené při jejím vývoji. Před dalším rozvojem je důležité tyto testy revidovat, opravit a doplnit.
Během vývoje předchozí verze aplikace také probíhalo uživatelské testování. Pro aktuální verzi toto testování nebylo nutné vzhledem k tomu, že z pohledu uživatele nebyl způsob ovládání aplikace nijak změněn.

### 5.6.4 Zabezpečení serveru

Pro budoucí nasazení aplikace do produkce doporučuji držet se předchozí kapitoly, která pojednává poměrně podrobně o nastavení serveru a jeho zabezpečení.
Průběžná integrace, která byla v aplikaci zavedena usnadní v budoucnu její vývoj a testování.
Pro přístup k serveru je aktuálně používáno SSH, které je veřejně přístupné na TCP portu 22. Toto je místo možné slabiny serveru a bylo by do budoucna vhodné jej uzavřít. V tomto případě bych doporučila povolit sem přístup pouze z DMZ - Demilitarizované zóny, nebo z VPN. Pro tyto účely by bylo možné použít školní VPN, kam mají standardně přístup všichni zaměstnanci. Tímto omezením přístupu k SSH se pak zamezí možným útokům na něj.
Pro úplně korektní a bezpečné provozování serveru by bylo vhodné jej umístit pod ochranu firewallu. Tento firewall by pak měl z internetu propouštět pouze povolený provoz, tedy uživatele připojující se přes HTTP, HTTPS a ICMP.

### 5.6.5 Další rozvoj

Aplikace BOZP by si zasloužila další rozvoj a to především v podobě automatizace procesů. Aktuální import dat z KOS API probíhá ručně a nepravidelně. To bylo zavedeno především proto, že data z KOSu nejsou řazena chronologicky a je tak možné, že některá si navzájem odporují. V aplikaci by bylo třeba naprogramovat systém, který by byl schopen rozhodnout tyto problémy. Hlubší automatizace procesů by pracovníkům BOZP velmi usnadnila práci.
Pro usnadnění práce by pak bylo třeba nalézt způsob, jakým by systém byl schopný rozlišit detailnější informace o jednotlivých uživatelích, aniž by bylo nutné je zadávat ručně. Jedná se například o rozlišení nových studentů a těch, kteří opakují první ročník. Druhá skupina by pak nemusela být školena dvakrát.

Pro potřeby pracovníků BOZP by pak bylo dobré upravit část statistik do atraktivnější podoby, kdy by mohlo být zajímavé například grafické zobrazení nasbíraných dat.
Během dalšího rozvoje aplikace by pak bylo vhodné celkově revidovat procesy a funkcionality. Mělo by se zkoumat především, zda jsou nutné, nebo jestli by nebylo možné je navrhnout a implementovat efektivněji a použitelněji.

### 5.6.6 Zavedení e-learningu

V praxi se aktuálně ve většině firem používá školení BOZP pomocí e-learningu. Toto řešení je nejen ekologické, ale má i ekonomické výhody. Především však šetří čas všech zúčastněných.
Pro budoucnost BOZP na FIT ČVUT by bylo dobré zvážit možnost zavedení školení pomocí e-learningu. Nebylo by pak již třeba organizovat hromadné přednášky a sbírat na nich podpisy účastníků. Tento způsob školení BOZP je již poměrně běžný a jeho zavedení by neměla brání žádná právní nařízení.

# Závěr

Cílem této diplomové práce bylo analyzovat BOZP portál na Fakultě Informačních Technologií především po stránce bezpečnosti, ale také po stránce funkční a uživatelské. Tato analýza proběhla za asistence pracovníků školy zabývajících se školení BOZP, kteří tento portál využívají ke své práci. Další část analýzy stávajícího stavu pak měla zkoumat výsledky etického hackování, které proti portálu proběhly v minulém roce. Také jsem měla analyzovat nařízení GDPR a jeho dopady na správu portálu. Cílem této rozsáhlé analýzy bylo objevit a pojmenovat komplexní nedostatky portálu, navrhnout řešení a implementovat jej.

Během shromažďování informací o vyhlášce GDPR jsem zjistila, jak komplikované téma to je a že do světa informační bezpečnosti přináší mnoho změn. Také jsem shromáždila dostatek materiálů pro to, aby bylo možné toto nařízení v portálu BOZP zavést.

Při analýze problémů zabezpečení aplikace a výstupů etického hackování jsem pak zjistila, že problémy se netýkají pouze samotné aplikace, ale také serveru na kterém aplikace běží.

Při provádění analýzy funkčností portálu jsem narazila na několik oblastí, které bylo možné vylepšit a ulehčit tak práci pracovníkům BOZP. Jednalo se především o oblast hromadné komunikace se studenty.

Vzhledem k bezpečnosti, není možné systém BOZP publikovat jako open-source, nicméně je v něm několik oblastí, které bych doporučila do budoucna dále rozvíjet. Jedná se například o automatizaci komunikace se systémem Moodle.

Během implementace mých návrhů se objevilo několik drobných problémů, které jsem ale byla schopna úspěšně vyřešit. Výsledná aplikace tedy obsahuje všechny navržené funkčnosti. Aplikaci jsem následně nasadila na testovací server a na něm také zavedla průběžnou implementaci. Systém je plně funkční na testovacím serveru a v době odevzdání této diplomové práce je naplánované také uvedení na produkční server, a to na polovinu května.

Manažerské a ekonomické zhodnocení celé implementace a především zavedení

GDPR pak ukázalo, že tyto úpravy byly nezbytné a Fakulta Informačních Technologií se tak vyhne případným problémům v budoucnu.
Všechny cíle mé práce tak byly úspěšně splněny, systém je nasazen, čeká na praktické využití a případný další rozvoj.

# Literatura


[1] OWASP. OWASP Code Review Guide v2. 2017. Available from: <https://www.owasp.org/images/5/53/OWASP_Code_Review_Guide_v2.pdf>

[2] Nezmar, L. *GDPR: praktický průvodce implementací*. Grada Publishing, 2017, ISBN 978-80-271-0668-4.

[3] Pinďák, Š. *BOZP portál - vedení studentského týmu a specifikace požadavků při jeho tvorbě*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2012.

[4] Humeník, M. *BOZP portál - správa školení*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2012.

[5] Jeschke, L. *BOZP portál - e-learning*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2012.

[6] Falta, K. *BOZP portál II - Evidence přístrojů*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.

[7] Jančík, D. *BOZP portál II - Evidence přístrojů*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.

[8] Kopecký, J. *BOZP portál - Testování*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.

[9] Náhlovský, M. *BOZP portál II - Správa úrazů*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.

[10] Falta, K. *BOZP portál - modul pro správu školení*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

[11] Náhlovský, M. *BOZP portal - eLearning module*. Master's thesis., Czech Technical University in Prague, Faculty of Information Technology, 2016.

[12] Kozáková, H. *BOZP portál - modul pro správu školení*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

[13] Dokumentace Frameworku Nette. 2016. Available from: <https://doc.nette.org/cs/2.3/>

[14] Dokumentace knihovny Nette Database. 2017. Available from: <https://doc.nette.org/cs/2.4/database>

[15] Dokumentace knihovny Nextras Datagrid. 2016. Available from: <https://componette.com/nextras/datagrid/>

[16] Dokumentace knihovny Nette Grido. 2016. Available from: <http://o5.github.io/grido-examples/documentation.cs.html>

[17] Dokumentace databáze MySQL. 2018. Available from: <https://www.mysql.com/>

[18] Dokumentace operačního systému Debian 6.0. 2015. Available from: <https://www.debian.org/releases/squeeze/debian-installer/>

[19] Debian 9 „Stretch“ uvolněn. 2017. Available from: <https://www.debian.org/News/2017/20170617>

[20] Apache HTTP Server Documentation. 2018. Available from: <https://httpd.apache.org/docs/>

[21] Apache. Apache httpd 2.2 vulnerabilities. 2017. Available from: <http://httpd.apache.org/security/vulnerabilities_22.html>

[22] PHP 5.4.45 Release Announcement. 2014. Available from: <http://php.net/releases/5_4_45.php>

[23] Olzak, T. The five phases of a successful network penetration. 2008. Available from: <https://www.techrepublic.com/blog/it-security/the-five-phases-of-a-successful-network-penetration/>

[24] Graves, K. *CEH: Official Certified Ethical Hacker Review Guide: Exam 312-50*. Wiley, second edition, 2 2017, ISBN 978-0782144376.

[25] Microsoft. SQL Injection. 2016. Available from: <https://technet.microsoft.com/en-us/library>

[26] OWASP. Cross-site Scripting (XSS). 2018. Available from: <https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)>

[27] OWASP. XSS (Cross Site Scripting) Prevention Cheat Sheet. 2018. Available from: <https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet#A_Positive_XSS_Prevention_Model>

[28] OWASP. Cross-Site Request Forgery (CSRF). 2018. Available from: <https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)>

[29] OWASP. Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet. 2018. Available from: <https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet>

[30] OWASP. Web Application Penetration Testing. 2017. Available from: <https://www.owasp.org/index.php/Web_Application_Penetration_Testing>

[31] OWASP. Fingerprint Web Server. 2017. Available from: <https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002)>

[32] Winder, D. Getting ready for GDPR. 2016. Available from: <https://search.proquest.com.ezproxy.techlib.cz/docview/1843263696?pq-origsite=summon>

[33] Parliament, E.; of the Council. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016, and repealing Directive 95/46/EC. 2016. Available from: <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.119.01.0001.01.ENG&toc=OJ:L:2016:119:TOC>

[34] Žůrek, J. *Praktický průvodce GDPR*. ANAG, 2017, ISBN 978-80-7554-097-3.

[35] Ministerstvo školství, m. a. t. Projekt: 1.5, Registrační číslo: CZ.1.07/1.5.00/34.030. 2016. Available from: <https://coptkm.cz/portal/reposit.php?action=0&id=40940&revision=-1&instance=1>

[36] Ministerstvo školství, m. a. t. Projekt: 1.5, Registrační číslo: CZ.1.07/1.5.00/34.0304. 2016. Available from: <https://coptkm.cz/portal/reposit.php?action=0&id=40950&revision=-1&instance=1>

Příloha A

# Seznam použitých zkratek

**GDPR** General Data Protection Regulation

**BOZP** Bezpečnost a ochrana zdraví při práci

**OWASP** Open Web Application Security Project

**oAUTH** Open Authorization

**SSO** Single Sign On

**TLS** Transport Layer Security

**DPIA** Data Protection Impact Assessment

**DPO** Data protection officer

**FIT** Fakulta informačních technologií

**SSL** Secure Sockets Layer

**SSH** Secure Shell

**TLS** Transport Layer Security

**DMZ** Demilitarized zone

**VPN** Virtual private network

**ICMP** Internet Control Message Protocol

**HTTP/S** Hypertext Transfer Protocol (Secure)

Příloha **B**

# Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF
- `materials` ..... materiály přiložené k práci
  - `ethical_hacking.pdf` .... výstup etického hackování ve formátu PDF
  - `database_model.png` ..... databázový model aplikace
  - `podpisova_listina.pdf` ..... ukázková podpisová listina
- `zadani.pdf` ..... zadání diplomové práce