Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Systém pro správu bezpečnostních incidentů v síti WEBnet

Plzeň 2021 Bc. Martin Šebela

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Martin ŠEBELA**
Osobní číslo:	**A19N0046P**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Softwarové inženýrství**
Téma práce:	**Systém pro správu bezpečnostních incidentů v síti WEBnet**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s postupy reakce CSIRT na bezpečnostní incidenty.
2. Proveďte analýzu bezpečnostních incidentů a na základě provedené analýzy identifikujte nejzávažnější a nejčastější bezpečnostní incidenty.
3. Navrhněte asistenční systém pro podporu řešení bezpečnostních incidentů.
4. Navržený systém implementujte.
5. Systém otestujte v reálném provozu.
6. Proveďte analýzu výsledků a doporučte další možná rozšíření.

Rozsah diplomové práce: **doporuč. 50 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Ing. Jiří Čepák**
Centrum informatizace a výpočetní techniky

Konzultant diplomové práce: **Doc. Ing. Pavel Král, Ph.D.**
Katedra informatiky a výpočetní techniky

Datum zadání diplomové práce: **11. září 2020**
Termín odevzdání diplomové práce: **20. května 2021**

L.S.

Doc. Dr. Ing. Vlasta Radová
děkanka

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 24. září 2020

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 20. května 2021

Bc. Martin Šebela

Abstract

This master's thesis deals with cybersecurity, specifically the description of incident response teams CERT/CSIRT and the description of procedures used in the incident response process. The most common and major security incidents at the University of West Bohemia (UWB) are discussed in the following analysis. Based on the results of the analysis, the requirements for the software that is the aim of the master's thesis are specified. The implemented CAIH (*Computer-aided Incident Handling*) system is designed to be intuitive and modular and integrates several existing information systems. The inputs of the implemented system are the date and time and any IP address from the UWB IP range. Based on the inputs, the system automatically searches for related records in the network logs and blocks or unblocks the detected user or device at the UWB network (*WEBnet*). Results of the master's thesis show that when the implemented CAIH system was deployed, the time required to resolve a security incident was reduced. The time required to resolve an incident was reduced by an average of up to five times compared to the manual control and up to three times compared to the software currently in use.

Abstrakt

Diplomová práce se zabývá problematikou kyberbezpečnosti, a to konkrétně popisem bezpečnostních týmů typu CERT/CSIRT a popisem postupů a metod používaných při řešení bezpečnostních incidentů. Následuje analýza nejčastějších a nejzávažnějších bezpečnostních incidentů, které jsou řešeny univerzitním bezpečnostním týmem na ZČU. Na základě analýzy a současného stavu jsou specifikovány požadavky na software, který je výstupem diplomové práce. Implementovaný systém typu CAIH (*Computer-aided Incident Handling*) je navržen jako intuitivní a modulární a integruje několik existujících informačních systémů. Systém umožňuje po zadání data a času a libovolné IP adresy z rozsahu ZČU, vyhledat související záznamy v logu a provést blokaci, či následné odblokování konkrétního uživatele nebo zařízení v síti ZČU. Výsledky diplomové práce ukazují, že po nasazení implementovaného systému do ostrého provozu byla snížena doba nutná k vyřešení bezpečnostního incidentu v průměru až pětinásobně oproti manuálnímu postupu a až trojnásobně oproti dosud používanému nástroji.

Poděkování

Chtěl bych poděkovat vedoucímu práce, panu *Ing. Jiřímu Čepákovi* za vedení diplomové práce a za všechny připomínky s ní související. Dále pak konzultantovi práce, panu *Ing. Aleši Padrtovi, Ph.D.* z *Forenzní laboratoře CESNET* za připomínky a rady týkající se analytické části.

Obsah

1 Úvod

Internet a počítačové sítě obecně jsou součástí běžného života již desítky let, jejich důležitost a provázanost se společností se ovšem rok od roku zvyšuje. Nepostradatelnost *Internetu* potvrdila i pandemie v roce 2020, která zvýšila nároky na bezpečnost a potřebu využívat online prostředí i tam, kde to donedávna nebylo zvykem. Do online prostředí se ovšem dávno předtím přesunuli i útočníci, jejichž jediným cílem je škodit a zneužívat *Internet* k nelegálním aktivitám. Různé formy útoků, pokusů o narušení nebo o proniknutí do počítačových systémů a sítí byly totiž zaznamenány již v roce 1988 [19].

Vzhledem k tomu, že kybernetických hrozeb neustále přibývá a techniky útočníků jsou čím dál více sofistikovanější, je nutné, aby se řešením bezpečnostních incidentů někdo zabýval. Státy a instituce, které jsou si kybernetických hrozeb vědomy, tak začaly budovat specializované bezpečnostní týmy typu CERT/CSIRT, jejichž úkolem je okamžitá reakce na kybernetické bezpečnostní incidenty a minimalizace případných škod. Na *Západočeské univerzitě v Plzni* (ZČU) působí bezpečnostní tým WIRT[1], který má ve správě univerzitní počítačovou síť označovanou názvem *WEBnet*[2].

Každý bezpečnostní tým řeší všechny incidenty ve své síti, tedy nejen ty, na které přijde sám, ale i ty, které jsou mu nahlášeny. Kromě běžných uživatelů se tak na bezpečnostní tým obrací i další bezpečnostní týmy z jiných institucí, potažmo států. Bezpečnostnímu týmu jsou například zasílána hlášení o problémových uživatelích nebo hlášení na kompromitovaná zařízení v jím spravované počítačové síti, která jsou zdrojem phishingových útoků nebo jsou součástí botnetu. Podobně může bezpečnostní tým řešit požadavky od orgánů činných v trestním řízení.

Pro řešení bezpečnostních incidentů existuje standardizovaný postup, který se aplikuje na všechny incidenty stejně, ale implementace dílčích kroků může být odlišná. Prvotní kroky musí být především rychlé a účinné, aby kompromitované zařízení nebo podezřelý uživatel nezpůsobili další škody a nezhoršili celkovou reputaci organizace.

Řešení každého bezpečnostního incidentu je navíc časově náročné. Je třeba důkladně analyzovat různé logy, sbírat data, a ta následně vyhodnocovat a komunikovat se všemi účastníky incidentu. Proces řešení incidentu

[1] WEBnet Incident Response Team – `https://wirt.zcu.cz`

[2] WEBnet je zkratka slov **We**st **B**ohemia **net**work

je zároveň náchylný na případné chyby. Podobné je to se školením nových členů bezpečnostního týmu, kterým je nutné postupně představit všechny postupy používané při řešení různých typů incidentů. Situaci by tak bezpečnostnímu týmu usnadnil software, který by členovi s řešením bezpečnostního incidentu pomohl, a zefektivnil tak jeho práci.

Cílem diplomové práce je vytvořit modulární systém pro správu bezpečnostních incidentů, který celý proces od nahlášení incidentu po jeho uzavření automatizuje, urychlí a zjednoduší tak řešení celého incidentu. Systém členovi bezpečnostního týmu poskytne jednoduchý automat s jasně definovanými stavy, kterými je nutné při řešení bezpečnostního incidentu postupně projít. Vytvořený systém bude díky jednotlivým modulům napojen na dosud existující systémy, které jsou na ZČU nasazeny a jejichž cílem je například evidování bezpečnostního incidentu v interním systému pro správu požadavků nebo blokování uživatele a zařízení v určité oblasti počítačové sítě.

2 Kyberbezpečnost a bezpečnostní týmy

Bezpečnost je u elektronicky poskytovaných služeb definována tzv. *triádou kybernetické bezpečnosti* označovanou též jako CIA podle počátečních písmen následujících tří vlastností [11, 21, 24]:

- *Confidentiality* (důvěrnost) – přístup je umožněn jen oprávněným uživatelům,
- *Integrity* (integrita) – s daty nebylo neoprávněně manipulováno,
- *Availability* (dostupnost) – služba je funkční a přístupná.

Situace, během které mohlo dojít k narušení některé z výše uvedených vlastností CIA, je označována jako tzv. *bezpečnostní událost* [24]. Typickým příkladem je, že se útočníkovi podařilo zjistit heslo uživatele, které ale zatím nebylo nikde zneužito.

Bezpečnostní incident pro změnu označuje stav, kdy už došlo k narušení některé z výše uvedených vlastností CIA [24]. Jako příklad lze uvést stav, kdy útočník zjištěné heslo již zneužil.

2.1 Bezpečnostní týmy

Za řešení bezpečnostních událostí a incidentů jsou v organizacích zodpovědná samostatná oddělení – tzv. *bezpečnostní týmy* označované jako CERT (*Computer Emergency Response Team*) nebo CSIRT (*Computer Security Incident Response Team*), viz podkapitola 2.5 o *Pojmenování bezpečnostních týmů*. Jedná se o centralizovaná a specializovaná oddělení, jejichž cílem je efektivní reakce na bezpečnostní incidenty [18].

Počet osob zabývajících se řešením bezpečnostních incidentů se může v průběhu času v každé organizaci vyvíjet. V počátku se řešení bezpečnostních incidentů mohou věnovat převážně jednotlivci. S přibývajícím počtem incidentů a zvyšujícím se nárokům na bezpečnost, začínají vznikat konkrétní pracovní skupiny, případně dochází k vybudování bezpečnostních týmů typu CERT/CSIRT. U větších organizací se také může jednat o celá samostatná oddělení. [24]

Některé organizace ovšem nemusí mít v důsledku své politiky a podmínek bezpečnostní týmy zřízeny, a jejich činnost je například svěřena správci počítačové sítě [24].

Základní službou, která je každým bezpečnostním týmem poskytována, je řešení bezpečnostních incidentů (*incident handling*). Pokud jsou k dispozici dostatečné zdroje nebo je v zájmu organizace poskytovat i další služby, pak jsou často předmětem bezpečnostních týmů i následující činnosti [24]:

- vydávání varování a upozornění o možných zranitelnostech (např. o novém druhu podvodného e-mailu – tzv. *phishing*),
- správa zranitelností (udržování přehledu o běžících systémech a jejich verzích v organizaci a mapování zranitelností na verze systémů v organizaci),
- analýza artefaktů (binární soubory, logy, *NetFlow* apod.),
- forenzní analýza (zjišťování důkazů, z jakého důvodu k incidentu došlo),
- sledování nových technologií a definování doporučených postupů,
- audity a penetrační testy,
- vývoj bezpečnostních nástrojů,
- detekce průniků v počítačové síti,
- sdílení dat o bezpečnosti a incidentech,
- *Disaster Recovery* (plán obnovy po vážné havárii s ohledem na co nejrychlejší znovuobnovení systému),
- *Business Continuity Management* (alternativní plán obnovy při selhání systému),
- vzdělávání uživatelů a administrátorů,
- konfigurace a údržba nástrojů, aplikací a infrastruktury,
- bezpečnostní konzultace,
- analýza rizik,
- vyhodnocení bezpečnosti produktů (např. prováděním analýzy zdrojového kódu – tzv. *code review*).

Každý bezpečnostní tým má pod správou určité pole působnosti (tzv. konstituence) [24]. Typicky se jedná o konkrétní IP rozsah a o konkrétní DNS (*Domain Name System*) domény.

Například v případě univerzitního bezpečnostního týmu WIRT, který má ve správě počítačovou síť *WEBnet* na *Západočeské univerzitě v Plzni*, se jedná o:

- IPv4 síť `147.228.0.0/16`,
- IPv6 síť `2001:718:1801::/48`,
- doménové jméno `zcu.cz` a další domény ve vlastnictví ZČU.

2.2 Typy bezpečnostních týmů

Bezpečnostní týmy lze rozdělit podle možností zásahu do následujících tří typů [24]:

- *koncové* (nebo také *výkonné*),
- *koordinační* (nebo také *přeposílací*),
- *kombinované*.

Všechny bezpečnostní týmy jsou na stejné úrovni, neboť mezi nimi neexistuje žádná hierarchie [22, 24]. Důvěryhodnost je tak závislá především na osobních vazbách mezi jednotlivými CSIRT týmy [24]. Každý z bezpečnostních týmů ovšem může pro zvýšení důvěryhodnosti projít certifikací u mezinárodně uznávaných organizací sdružujících CSIRT týmy (viz kapitola 2.6).

2.2.1 Koncový (výkonný) bezpečnostní tým

Koncový bezpečnostní tým má svou konkrétní síť, kterou spravuje a v níž má právo a možnosti aktivně zasahovat, tedy omezovat a blokovat jednotlivá zařízení nebo uživatele.

Příkladem koncového týmu je univerzitní bezpečnostní tým WIRT na ZČU.

2.2.2 Koordinační (přeposílací) bezpečnostní tým

Jedním z cílů koordinačního, resp. přeposílacího týmu je fungovat jako jednotný kontakt do rozsáhlé sítě (např. do sítě CESNET2) nebo jako jednotný kontakt státu se zahraničím (například se zahraničními CSIRT týmy) [24].

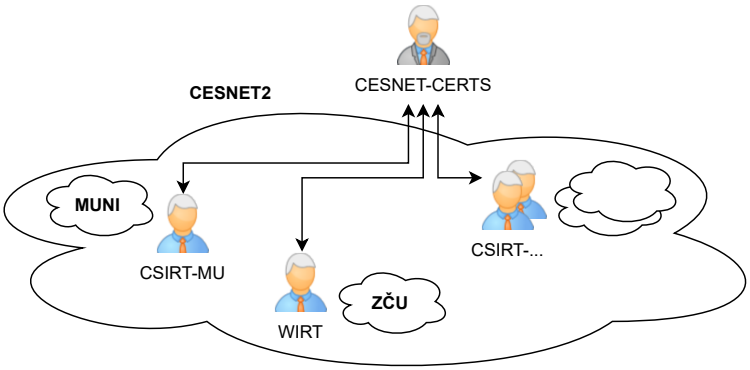
Koordinační tým může také pomáhat ostatním CSIRT týmům při řešení rozsáhlejšího bezpečnostního incidentu nebo správně nasměrovat začínající bezpečnostní týmy, kterým pomůže nalézt konkrétní CSIRT tým (protistranu), který by incident mohl vyřešit a stížnost mu předá [24].

Koordinační bezpečnostní tým typicky nemá svou vlastní síť, a nemá tak výkonné pravomoci [18]. Obdržená hlášení z jiných sítí na základě vlastního seznamu předává dalším bezpečnostním týmům, jejichž počítačové sítě nebo uživatelů se incident týká – například koncovým, univerzitním týmům nebo týmům v komerční sféře. Koordinační tým také pro ostatní bezpečnostní týmy pořádá školení nebo je informuje o aktuálních bezpečnostních hrozbách a zranitelnostech [24].

Jako příklad lze uvést bezpečnostní tým CSIRT.CZ, který plní roli národního koordinačního CSIRT týmu *České republiky*, a to na základě dohody s NÚKIB (*Národní úřad pro kybernetickou a informační bezpečnost*) [18].

2.2.3 Kombinovaný bezpečnostní tým

Jedná se o bezpečnostní tým s vlastní sítí, která je týmu svěřena (typicky se jedná o páteřní síť) s tím, že zbytek požadavků je týmem koordinován [24]. Uvnitř páteřní sítě totiž mohou existovat i další podsítě, které jsou ale ve správě jiných bezpečnostních týmů. Pokud je tak problém detekován v některé z podsítí, za kterou je zodpovědný jiný bezpečnostní tým, pak jsou související požadavky předávány právě konkrétnímu, výkonnému týmu.



Obrázek 2.1: Koordinační role bezpečnostního týmu CESNET-CERTS v síti CESNET2 se znázorněním koordinace koncových (v tomto případě dvou univerzitních) bezpečnostních týmů. Pokud je incident detekován v některé z podsítí, CESNET-CERTS hlášení předává konkrétnímu, výkonnému týmu, jehož podsítě se incident týká.

Příkladem může být bezpečnostní tým CESNET-CERTS, který spravuje akademickou síť CESNET2, jejíž součástí je i podsíť `147.228.0.0/16` určená pro potřeby ZČU. V případě, že bezpečnostní tým CESNET-CERTS obdrží hlášení o problému uvnitř sítě ZČU, požadavek automaticky předává k vyřešení univerzitnímu bezpečnostnímu týmu WIRT (viz obr. 2.1).

2.3 Kontaktování bezpečnostního týmu

Cílem kontaktování je informovat bezpečnostní tým, že byl v jeho konstituenci detekován konkrétní problém [24]. Smyslem zasílaných hlášení je

- upozornit bezpečnostní tým na problémového uživatele nebo na kompromitované zařízení, které způsobuje v jiné síti abnormální nebo podezřelé, neoprávněné chování a požadovat nápravu (například napadené zařízení skenuje a útočí na síť jiného CSIRT týmu, který hlášení zasílá),
- upozornit bezpečnostní tým na nalezenou zranitelnost nebo bezpečnostní chybu, a zabránit tak zneužití chyby v konstituenci týmu.

V případě kontaktování bezpečnostního týmu se očekává věcnost sdělení a podložení informací daty, resp. doložení konkrétních důkazů – například výpisy z logů, výpisy z *NetFlow* a dalších. [24]. Příjemce sdělení (tj. bezpečnostní tým) má tak možnost porovnat obdržené výpisy se svými vlastními logy a zjistit, zdali k uvedené události v daný okamžik skutečně došlo a případně provést adekvátní kroky (např. zablokovat napadené zařízení, které rozesílá phishingové e-maily).

Bezpečnostní týmy mezi sebou komunikují standardizovanou formou. Očekává se vždy funkční e-mailová adresa ve tvaru `abuse@domain.tld` (např. `abuse@zcu.cz`), která slouží ke kontaktování bezpečnostního týmu včetně hlášení bezpečnostních incidentů [24]. K bezpečnostním incidentům dochází nezávisle na čase nebo dni, a je tak vhodné kontaktovat tým právě pomocí e-mailové adresy, která je navíc typicky svázána se systémem pro správu požadavků, ve kterém je incident následně evidován (viz podkapitola 2.4) [24]. Pro nahlášení konkrétního problému tak stačí zjistit, jaký bezpečnostní tým má daný IP rozsah nebo doménu ve správě a kontaktovat CSIRT tým na zmíněnou e-mailovou adresu.

Oficiální kontakt na bezpečnostní tým může být nalezen například pomocí služby `whois` nad konkrétní doménou nebo IP adresou [24]. Mezi informacemi o doméně nebo IP adrese je pak uveden tzv. *abuse contact*. Ten běžně obsahuje již výše zmíněnou e-mailovou adresu ve tvaru `abuse@domain.tld`

(např. `abuse@zcu.cz`), která slouží k nahlášení bezpečnostního incidentu. Část výstupu nástroje služby `whois` pro IP rozsah ZČU (`147.228.0.0/16`) ukazuje obr. 2.2.

```
% Information related to '147.228.0.0 - 147.228.255.255'
% Abuse contact for '147.228.0.0 - 147.228.255.255' is 'abuse@zcu.cz'

inetnum:        147.228.0.0 - 147.228.255.255
netname:        ZCU-TCZ
descr:          Plzen
country:        CZ
org:            ORG-UOWB1-RIPE
admin-c:        UOWB1-RIPE
tech-c:         UOWB1-RIPE
status:         LEGACY
mnt-by:         RIPE-NCC-LEGACY-MNT
mnt-by:         TENCZ-MNT
remarks:        Please report network abuse -> abuse@zcu.cz
...
abuse-mailbox:  abuse@zcu.cz
```

Obrázek 2.2: Část výstupu služby `whois` pro IP rozsah `147.228.0.0/16` (IPv4 rozsah ZČU). Ve výstupu je uvedena e-mailová adresa `abuse@zcu.cz`, která slouží pro nahlášení bezpečnostního incidentu.

Alternativou může být i soubor `security.txt`, který se u webových projektů umisťuje do adresáře `/.well-known`. V souboru je běžně uváděn kontakt, resp. kontaktní e-mail na bezpečnostní tým (direktiva `Contact`, viz obr. 2.3), popř. další informace jako například PGP (*Pretty Good Privacy*) klíč, kterým bude možné obsah zasílané zprávy zašifrovat.

```
Contact: https://support.zcu.cz/index.php/WIRT
Contact: mailto:abuse@zcu.cz
```

Obrázek 2.3: Obsah souboru `security.txt` umístěného na webu[2] ZČU s informacemi ke kontaktování bezpečnostního týmu WIRT.

2.4 Evidence komunikace

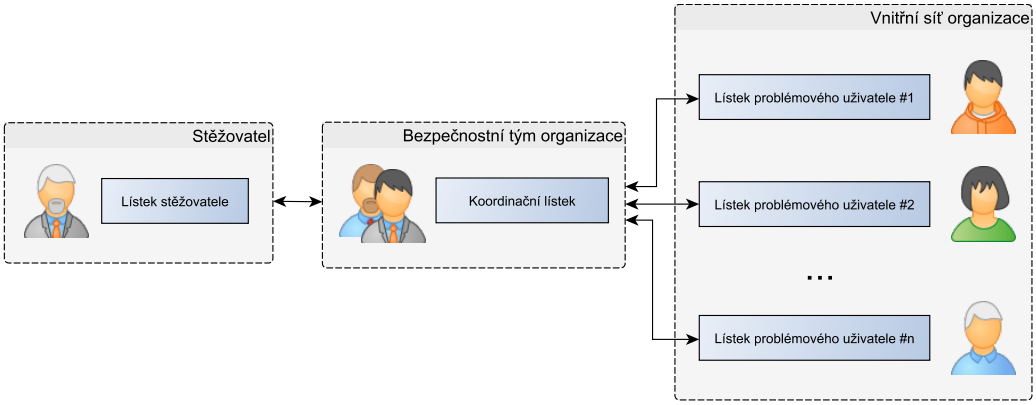
Veškerá komunikace bezpečnostního týmu by měla být z důvodu uchování historie evidována v interním systému pro správu požadavků (tzv. *ticketing system*), kde každý požadavek, dotaz nebo hlášení představuje jeden tzv. *lístek* (*ticket*) [20, 24]. Například dotazy směřující na e-mailovou adresu `abuse@zcu.cz` jsou tak směrovány přímo do interního systému pro správu

[3]Příklad souboru `security.txt` na webu ZČU – `https://www.zcu.cz/.well-known/security.txt`

požadavků. Mezi zástupce takových systémů lze zařadit např. RT (*Request Tracker*) nebo OTRS (*Open-Source Ticket Request System*).

Každému hlášení (lístku), které je v systému pro správu požadavků založeno, je automaticky přiřazen jedinečný identifikátor. Pokud je součástí incidentu více napadených počítačů nebo více problémových uživatelů, vytváří se pro každého z nich vlastní lístek. U rozsáhlejších incidentů je doporučeno vytvářet i interní tzv. *koordinační lístek*, v rámci nějž bezpečnostní tým shromažďuje poznatky z řešení celého incidentu (viz obr. 2.4) [24].

Zároveň je důležité oddělit komunikaci stěžovatele (např. jiný bezpečnostní tým) od toho, na koho je stížnost mířena (např. problémový uživatel v organizaci) [24]. Pro jedno hlášení incidentu jsou tak vedeny alespoň dva zvláštní lístky – každý pro jednu ze stran. Uživatelé, vůči kterým je stížnost vznesena, by také nikdy neměli komunikovat přímo se stěžovatelem – komunikace s druhou stranou by nemusela být příliš efektivní z důvodu nepochopení požadavků nebo neznalosti postupů [24]. Kromě toho by se uživatelé mohli během komunikace k něčemu neúmyslně přiznat, a situace by tak mohla mít právní dohru.



Obrázek 2.4: Příkladové schéma řešení rozsáhlého bezpečnostního incidentu s využitím koordinačního lístku. Je důležité oddělení komunikace mezi stěžovatelem a možným problémovým uživatelem.

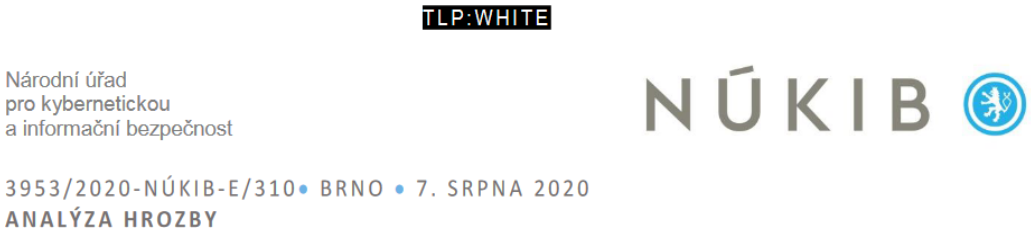
V každém lístku je uchovávána kompletní historie komunikace související s daným incidentem. Mezi lístky lze navíc vytvářet vazby – např. lístek má spojitost s jiným bezpečnostním incidentem, který tým již v minulosti řešil, případně je to prerekvizita pro jiný z lístků. Tím je zároveň zajištěn přehledný přístup k historii incidentu všem členům bezpečnostního týmu, jsou vyřešena přístupová práva s ohledem na pravidlo minimálního přístupu (každý vidí jen to, co nutně potřebuje) a je viditelné workflow celého incidentu [24].

V minulosti se ke komunikaci používala například sdílená e-mailová schránka – v porovnání se systémem pro správu požadavků se ovšem jedná o chaotické a nepříliš efektivní řešení bez možnosti vytvářet mezi incidenty vazby [24].

2.4.1 Traffic Light Protocol

Při vzájemné komunikaci mezi bezpečnostními týmy je možné se setkat s protokolem TLP (*Traffic Light Protocol*), který pomocí čtyř příznaků definuje, jakým způsobem může příjemce nakládat s informacemi, jež mu byly zaslány jiným bezpečnostním týmem nebo institucí [3, 21].

Typicky se jeden z příznaků (např. `TLP:AMBER`) vyskytuje už v předmětu zasílaného e-mailu (resp. bezpečnostního incidentu). Podobně se lze s tímto označením setkat i v záhlaví, popř. zápatí souvisejících dokumentů (viz obr. 2.5) [3].



Obrázek 2.5: Hlavička dokumentu vydaného NÚKIB s uvedeným příznakem `TLP:WHITE`. Dokument obsahoval analýzu hrozby napadení veřejných institucí (jako např. nemocnic) ransomwarem v *České republice*.

Protokol obsahuje následující čtyři příznaky [3], přičemž každý z nich definuje jiné podmínky použití související s důvěrností poskytované informace:

`TLP:RED` Nejpřísnější příznak, který zakazuje sdílení informace s dalšími subjekty (tzn. informace je určena jen původnímu příjemci) [24].

`TLP:AMBER` Limituje poskytnutí informace omezené komunitě uživatelů (uvnitř organizace) za předpokladu, že bezpečnostní tým uzná za vhodné, aby o situaci věděli i další členové [18, 24].

`TLP:GREEN` Neomezuje konkrétní skupinu uživatelů, kterým může být informace dále předána, nicméně vybraní uživatelé by měli být v nějakém kontextu s obdrženou informací (např. bezpečnostní komunita, partnerská organizace). Informace není určena pro obecnou veřejnost [12].

TLP:WHITE Sdílení informací žádným způsobem neomezuje a je na bezpečnostním týmu, jak s obdrženými informacemi naloží [21].

Při nedodržení podmínek použití definovaných TLP protokolem si tým snižuje reputaci a hrozí, že mu další bezpečnostní týmy přestanou poskytovat důležité a důvěrné informace [24]. Při zasílání důvěrných informací je tak nutné se ujistit, zdali jsou obě protistrany s příznaky protokolu TLP obeznámeny a dodržují je [12].

2.5 Pojmenování bezpečnostních týmů

S bezpečnostními týmy se pojí i jejich identifikace formou jedinečné zkratky (viz příklady uvedené v tabulce 2.1).

Zkratka	**Popis bezpečnostního týmu**
CSIRT.CZ	Národní bezpečnostní tým provozovaný sdružením CZ.NIC
GOVCERT.CZ	Vládní bezpečnostní tým provozovaný NÚKIB
CESNET-CERTS	Bezpečnostní tým sdružení CESNET
WIRT	Univerzitní bezpečnostní tým *Západočeské univerzity v Plzni* (ZČU)
CSIRT-MU	Univerzitní bezpečnostní tým *Masarykovy univerzity v Brně* (MUNI)
CSIRT-CUNI	Univerzitní bezpečnostní tým *Univerzity Karlovy* (CUNI)

Tabulka 2.1: Příklady názvů několika bezpečnostních týmů působících v *České republice*.

V některých názvech bezpečnostních týmů se vyskytuje zkratka CERT (*Computer Emergency Response Team*), která je zároveň registrovanou ochrannou známkou [24]. Zkratka CERT vznikla za cílem vytvoření bezpečnostního týmu na půdě americké *Carnegie Mellon University*, a to už v roce 1988 jako reakce na tehdejší narušení *Internetu* virusem *Morris* [19].

Alternativou ke zkratce CERT je volně dostupná zkratka CSIRT (*Computer Security Incident Response Team*), která se často vyskytuje u nově vzniklých bezpečnostních týmů [24]. Za zkratkou CSIRT poté typicky následuje zkratka organizace, kterou má bezpečnostní tým ve správě (např. CSIRT-MU v případě *Masarykovy univerzity v Brně*). Jak ale ukazuje tabulka 2.1, před zavedením a ustálením zkratky CSIRT bylo možné se setkat i s jinými názvy bezpečnostních týmů, jako například CESNET-CERTS nebo WIRT, které zkratku CSIRT neobsahují.

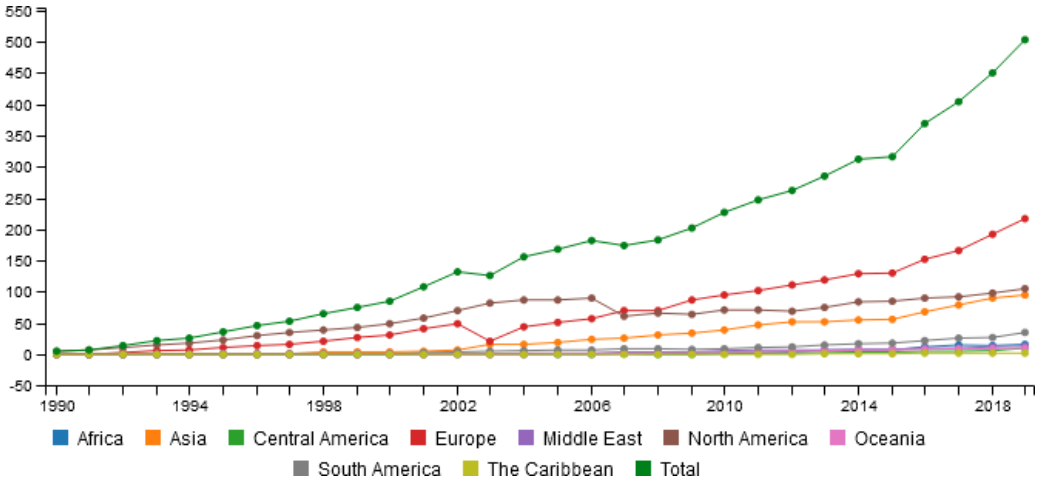
2.6 Certifikace bezpečnostních týmů

Protože je důvěryhodnost bezpečnostních týmů založena především na osobních vazbách mezi jednotlivými týmy, je nutné, aby měl bezpečnostní tým u dosud neznámých týmů garantováno, že komunikuje s kvalifikovaným a ověřeným protějškem.

A právě zvýšení důvěryhodnosti bezpečnostních týmů, potažmo nezávislé prověření postupů CSIRT týmů a standardizaci provádí dvě certifikační autority, a sice mezinárodní organizace FIRST a evropská organizace TI.

2.6.1 Organizace FIRST

FIRST (*Forum of Incident Response and Security Teams*) je mezinárodní organizace sdružující bezpečnostní týmy, které prošly ověřovacím procesem stejnojmenné organizace. Organizace FIRST zveřejňuje seznam prověřených bezpečnostních týmů na svých oficiálních webových stránkách `first.org`. K březnu roku 2021 bylo mezi členy organizace FIRST zařazeno 562 bezpečnostních týmů z 97 zemí světa (v případě *České republiky* se jednalo o 4 bezpečnostní týmy – národní CSIRT.CZ, vládní GOVCERT.CZ a dva komerční) [4]. Zastoupení a nárůst členů napříč kontinenty po jednotlivých letech (1990–2019) ukazuje graf na obr 2.6.



Obrázek 2.6: Graf převzatý z [7], který znázorňuje vzrůstající počet členů (bezpečnostních týmů) organizace FIRST v letech 1990–2019 v různých částech světa.

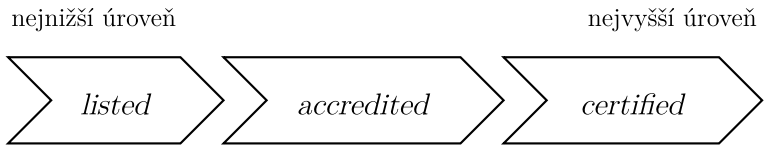
Aby mohl být na seznam prověřených bezpečnostních týmů zapsán nový CSIRT tým, je nutné, aby jiné dva, již dříve prověřené bezpečnostní týmy,

provedly u nového CSIRT týmu fyzickou kontrolu a audit práce [24]. Hodnotí se například proces zpracování bezpečnostních incidentů, ověření postupů, vzdělání členů bezpečnostního týmu, fyzické zabezpečení pracoviště apod. [19]. S členstvím v organizaci FIRST je spojen každoroční poplatek [18].

2.6.2 Organizace Trusted Introducer

Evropská organizace *Trusted Introducer* (dále TI) je dalším zástupcem sdružující převážně evropské bezpečnostní týmy. Působí pod organizací *GÉANT*, která poskytuje e-infrastrukturu pro vzdělávací a výzkumné instituce [19]. K březnu roku 2021 bylo organizací TI akreditováno na 419 bezpečnostních týmů z 66 zemí světa, z toho v 51 případech se jednalo o bezpečnostní týmy z ČR [2].

Organizace TI udržuje tři různé úrovně členství, přičemž pro získání konkrétní úrovně je třeba splnit několik podmínek. Pro získání vyšší úrovně členství je nutné nejprve získat úroveň nižší (např. úroveň *accredited* lze získat jen tehdy, pokud je bezpečnostní tým již zapsán v úrovni *listed* – viz obr. 2.7).



Obrázek 2.7: Úrovně členství v organizaci TI. Pro získání vyšší úrovně je nejprve nutné získat úroveň nižší.

Seznam úrovní členství a hrubý výčet požadavků, které je nutné pro danou úroveň členství splnit, je následující:

Listed je nejnižší, bezplatný status. Pro zařazení je nutná podpora od dvou jiných, ale již akreditovaných (tj. úroveň *accredited*) bezpečnostních týmů [18], vyplnění formuláře s informacemi o bezpečnostním týmu [19] a každoroční účast na testech [18].

Accredited je zpoplatněný status (roční poplatky) vyžadující ověření postupů bezpečnostního týmu, dále vyplnění formuláře dle standardu RFC 2350[4] (*Expectations for Computer Security Incident Response*) [19], který obsahuje základní informace o bezpečnostním týmu jako

[4]RFC 2350 – `https://www.ietf.org/rfc/rfc2350.txt`

například e-mailovou *abuse* adresu, na kterou je možné hlásit bezpečnostní incidenty, dále pak veřejné klíče, způsob šifrování komunikace apod. – viz příklad vyplněného dokumentu u vládního bezpečnostního týmu GOVCERT.CZ ve zdroji [14].

Certified je zpoplatněný (roční poplatky), nejvyšší stupeň certifikace spočívající v externím auditu bezpečnostního týmu, který trvá až jeden rok [18, 19].

Úplný výpis požadavků každé z úrovní včetně například maximální doby řízení je uveden na oficiálních stránkách organizace TI – `trusted-introducer.org`.

Počty členů organizace TI k březnu roku 2021 v jednotlivých úrovních členství včetně dalších možných stavů (kandidáti na zařazení do některé z úrovní a další) ukazuje tabulka 2.2, která vychází z oficiálního seznamu [2] všech evidovaných týmů.

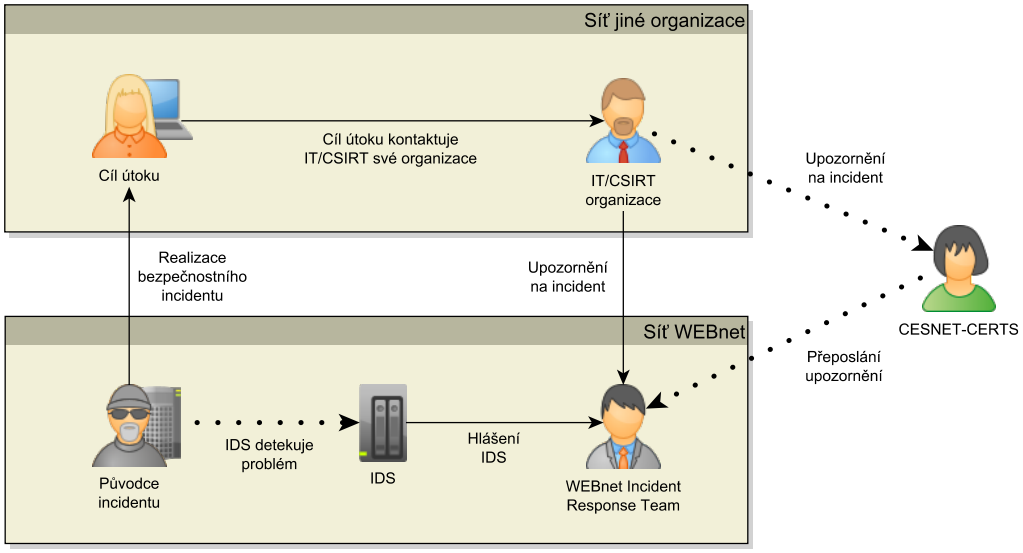
Úroveň členství/aktuální status	**Počet členů**
listed	178
accredited	181
certified	30
candidate	20
pending	8
suspended	2

Tabulka 2.2: Počty bezpečnostních týmů dle úrovně členství, resp. aktuálního stavu v organizaci TI k březnu roku 2021.

3 Řešení bezpečnostních incidentů

Jak už bylo zmíněno v kapitole 2.1, základní službou, kterou každý bezpečnostní tým poskytuje, je řešení bezpečnostních incidentů (*incident handling*).

Každý bezpečnostní incident (například útok na zařízení v síti) má svého původce a ten má zase konkrétní cíl útoku (viz obr. 3.1) [23]. Prováděný útok je detekován buď automatickými sondami, nebo abnormální chování zaznamená oběť samotná. Oběť si na podezřelé chování stěžuje svému IT oddělení (nebo bezpečnostnímu týmu), které hlášení oběti analyzuje. Upozornění na incident a žádost o nápravu je poté předáno k řešení buď přímo výkonnému bezpečnostnímu týmu organizace, ze které útok pochází (např. týmu WIRT), nebo koordinačnímu bezpečnostnímu týmu (na obr. 3.1 je uveden CESNET-CERTS), který hlášení poté předá konkrétnímu výkonnému týmu.



Obrázek 3.1: Schéma vytvořené na základě původního schématu ze zdroje [23], které zachycuje průběh řešení bezpečnostního incidentu.

Standardní postup při řešení bezpečnostního incidentu se skládá z pěti dílčích částí [25], a sice z:

1. přípravy (*Preparation*),
2. zjištění (*Detection and Reporting*),

3. vyhodnocení a analýzy (*Triage and Analysis*),

4. izolace a neutralizace (*Containment and Neutralization*),

5. závěrečné činnosti (*Post-Incident Activity*).

Postup byl vytvořen na základě poznatků z řešení bezpečnostních incidentů a platí pro jakýkoliv typ incidentu [24]. Detailní popis každé z pěti částí bude uveden v následujících podkapitolách.

3.1 Příprava – *Preparation*

Příprava se zabývá možným výskytem bezpečnostního incidentu a způsobem, jakým incidentu předejít nebo jak incident pomocí předem definovaných postupů efektivně vyřešit. Cílem je především následné ulehčení vyřešení bezpečnostního incidentu. Bezpečnostní tým má tak předem připravené prostředí, postupy a definovaná práva a povinnosti pro řešení konkrétních typů incidentů. Konkrétně je cílem [25]:

- *minimalizovat výskyt incidentu* – místo 1000 napadených počítačů bude bezpečnostní tým řešit jen jejich zlomek (např. díky zapnutému firewallu),

- *minimalizovat dopad incidentu* – problém přijde, škoda ale není tak zásadní (např. díky pravidelnému zálohování případný výskyt ransomwaru nezpůsobí takové škody),

- *připravit se na výskyt incidentu* – je připraven jasný postup na řešení incidentu.

3.1.1 Doporučení a *best practices*

Příprava na vyřešení incidentů může vycházet například z doporučení a *best practices* prezentovaných na konferencích a seminářích od členů jiných bezpečnostních týmů, správců a dalších autorit, jejichž zkušenosti mohou napovědět, jakým způsobem mají být správně nastaveny služby, jaké typy událostí je vhodné zaznamenávat do logů apod. [24].

3.1.2 Logování

Cílem logování a monitoringu je mít k dispozici data o událostech, ke kterým v počítačové síti nebo v systému dochází nebo došlo a získané informace použít jako podklady pro další analýzu nebo vyšetřování [24].

Základem je zaznamenávání vazby *IP adresa – časová značka – uživatel*, která definuje, jakému uživateli a v jakém čase byla přidělena konkrétní IP adresa.

V případě logu služby DHCP (*Dynamic Host Configuration Protocol*) je navíc součástí každého záznamu MAC (*Media Access Control*) adresa zařízení (viz obr. 3.2).

```
1. Feb 18 20:41:57 147.228.54.25 dhcpd[10348]: DHCPDISCOVER
   from 94:65:9c:**:**:** via 147.228.128.3
2. Feb 18 20:41:58 147.228.54.25 dhcpd[10348]: DHCPOFFER
   on 147.228.136.200 to 94:65:9c:**:**:** (example-ntb) via 147.228.128.3
3. Feb 18 20:41:58 147.228.54.25 dhcpd[10348]: DHCPREQUEST
   for 147.228.136.200 (147.228.54.25) from 94:65:9c:**:**:** (example-ntb) via 147.228.128.3
4. Feb 18 20:41:58 147.228.54.25 dhcpd[10348]: DHCPACK
   on 147.228.136.200 to 94:65:9c:**:**:** (example-ntb) via 147.228.128.3
```

Obrázek 3.2: Část výstupu DHCP logu při snaze zjistit, jakému uživateli byla ve specifikovaný čas přidělena IP adresa `147.228.136.200`. Na výstupu je vidět i výše zmíněná MAC adresa zařízení včetně například jeho `hostname`. Zjištění vazby na uživatelské jméno je poté provedeno přes RADIUS (*Remote Authentication Dial In User Service*) server.

Společně s logováním systémových událostí je důležité, aby byly zaznamenávány i události a akce provedené v jednotlivých službách – například akce při změně nastavení firewallu, při provedení změn v centrální správě antimalware řešení apod.

3.1.3 Sběr toků v počítačových sítích – *NetFlow*

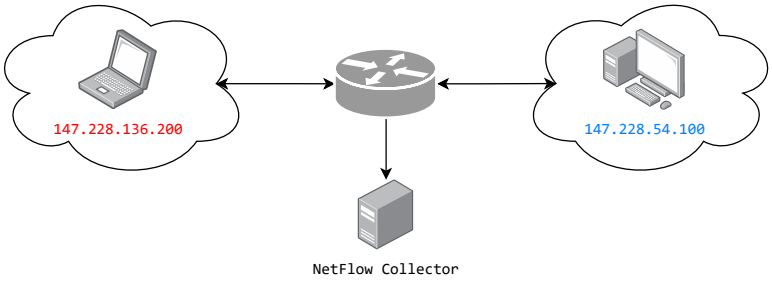
S logováním souvisí i zaznamenávání dalších informací jako je sběr toků v počítačových sítích. Aktivní prvky sítě (routery, switche) umožňují zaznamenávat data o tocích v počítačových sítích, a zaznamenaná data následně ukládat na určené místo [24]. Typickým představitelem je protokol *Cisco NetFlow*. Administrátor a automatické sondy mají možnost zachycená data v reálném čase monitorovat, analyzovat a detekovat potenciální problémy.

U každého síťového toku jsou sbírány vlastnosti jako:

- zdrojová IP adresa (`Src-IP`) a cílová IP adresa (`Dst-IP`),
- zdrojový port (`Src-Port`) a cílový port (`Dst-Port`),
- použitý protokol (`Protocol`),
- časová značka zahájení přenosu (`Flow-Start`) a časová značka ukončení přenosu (`Flow-End`), popř. doba komunikace,

- počet packetů (`Pkts-estimated`),
- počet přenesených bajtů (`Bytes-measured`).

Zjednodušené schéma komunikace mezi dvěma uzly v síti včetně zachycených informací ukazuje obr. 3.3.



NetFlow	1.	2.
Src-IP	147.228.136.200	147.228.54.100
Dst-IP	147.228.54.100	147.228.136.200
Src-Port	56323	443
Dst-Port	443	56323
Protocol	TCP	TCP
Flow-Start	2021/02/18 21:28:45.157	2021/02/18 21:28:45.329
Flow-End	2021/02/18 21:28:45.366	2021/02/18 21:28:45.685
Bytes-measured	818 B	9 kB
Pkts-measured	6 p	11 p

Obrázek 3.3: Ukázka schématu a výstupu *NetFlow* při snaze zjistit, s jakou IP adresou a na jakém portu komunikovala IP adresa 147.228.136.200 (`Src-IP`), a to dne 18. 2. 2021 ve 21:28 (`Flow-Start`).

3.1.4 Aktivní zásahy (blokování)

Bezpečnostní tým by měl mít možnost provádět aktivní zásahy s cílem minimalizovat škody (typicky odpojit napadené zařízení od sítě nebo zablokovat kompromitované konto), a to bez nutnosti žádat o autorizaci samostatně pro každý bezpečnostní incident [24]. Mezi aktivní zásahy patří blokování

- uživatele,
- zařízení,
- domény – zablokování celé domény za účelem ochránění uživatelů organizace z důvodu napadení webu útočníkem napříč podstránkami,
- konkrétní URL adresy – zablokování jedné konkrétní, útočníkem napadené stránky (ostatní stránky jsou nedotčeny).

3.1.5 Šablony odpovědí

Bezpečnostní tým by měl mít předem připravené šablony (vzory) odpovědí, resp. odpovědi na hlášení konkrétních incidentů. Díky šablonám bude zaručeno, že v obsahu odpovědi jsou srozumitelně zmíněny všechny potřebné a zároveň jen nutné informace bez dalších podrobností [24]. V ideálním případě je připravené sdělení před odesláním pouze parametricky doplněno například o incidentem dotčené IP adresy, časové značky a další parametry.

Parametrizované šablony jsou zároveň jedním z požadavků na systém implementovaný v rámci diplomové práce.

3.2 Zjištění – *Detection and Reporting*

K detekování bezpečnostního incidentu dochází buď vlastním zjištěním (*detection*) přímo uvnitř organizace, kterou bezpečnostní tým spravuje, nebo zvenčí, tj. nahlášením třetí stranou (*reporting*) [24].

Je důležité, aby bylo na každé obdržené hlášení bezpečnostního incidentu vždy a včas reagováno. Výjimkou je, pokud je v hlášení uvedeno, že odpověď není vyžadována [24]. V případě, že se jedná o rozsáhlejší bezpečnostní incident a je třeba nejprve provést související analýzu, stačí, aby byla druhá strana informována o právě probíhajícím zpracovávání incidentu s tím, že závěrečné stanovisko bude zasláno po dokončení analýzy [24]. Bezpečnostní tým (nebo jiný stěžovatel), kterým byl incident detekován a nahlášen, tak bude mít zpětnou vazbu o tom, zdali se řešením incidentu někdo zabývá. Nereagování na hlášení může způsobit znovu zasílání totožných hlášení nebo může vést k eskalaci celého incidentu a předání hlášení koordinačnímu CSIRT týmu (viz podkapitola 2.2.2), poskytovateli připojení, právnímu oddělení nebo orgánům činným v trestním řízení [24].

3.2.1 Vlastní detekce – *Detection*

Vlastním zjištěním (*detection*) uvnitř organizace může být bezpečnostní incident identifikován například na základě analýzy *NetFlow*, pomocí IDS (*Intrusion Detection System*) nebo na základě sond umístěných v síti – např. pomocí nástrojů poskytujících monitoring anomálií v síti [24].

Dalším možným zdrojem mohou být hlášení od uživatelů nebo od zabezpečení koncových stanic s centrální správou, v rámci níž jsou shromažďovány události detekované na uživatelských stanicích (tzv. *endpoint security*, viz obr. 3.4). Hlášení o incidentech mohou být také získána od členů uživatelské podpory organizace.

Blokovane aktivity na stanicich

Threat Target IP Address->Action Taken->Threat Target File Path	Number of Threat Events
147.228.[redacted]	1
Delete	1
D:\TEMP\[redacted]\7zOCC74280F\Piložená_faktura.exe	1

Obrázek 3.4: Příklad automatického hlášení z centrální správy *McAfee Endpoint Security* (zabezpečení koncových stanic), které na jedné z koncových stanic na ZČU detekovalo a zabránilo spuštění potenciálně nebezpečného souboru typu EXE, vydávajícího se za fakturu.

3.2.2 Externí nahlášení – *Reporting*

Druhou zmíněnou možností jsou externí hlášení (*reporting*), tj. hlášení pocházející mimo organizaci.

Typicky se jedná o hlášení od organizací, které identifikovaly, že je jejich síť využívána nezvyklým, abnormálním způsobem zařízeními nebo uživateli z jiné organizace. U poškozené organizace může být například zaznamenáno neodůvodněné skenování počítačové sítě, cílené útoky typu (D)DoS – *(Distributed) Denial of Service* nebo může být organizace zahlcena nadměrným množstvím nevyžádaných e-mailů z cizí sítě.

```
Vážení kolegové.

Naše detekční systémy zaznamenaly následující možné problém(y) související s Vaším
rozsahem IP adres nebo Vaší doménou (uvedené časy jsou v časové zóně Europe/Prague):

[1] Stroj je zkompromitován a je součástí botnetu.

Zdroj            První událost     Poslední událost  Detekt  Zpráv
─────────────────────────────────────────────────────────────────────
147.228.**.**    08.03. 10:57:21   08.03. 10:57:21        1      1
─────────────────────────────────────────────────────────────────────

Tento report obsahuje událostí se STŘEDNÍ závažností. Prosím prohledejte
dotčené systémy a napravte zjištěné problémy.
```

Obrázek 3.5: Příklad úryvku automatického hlášení systému *Mentat*, které dorazilo na e-mail `abuse@zcu.cz` jako reakce na možný bezpečnostní incident v síti ZČU.

Další hlášení mohou pocházet od organizací provádějících službu „*sdílení dat o bezpečnosti a incidentech*“ (viz seznam činností v kapitole 2.1). Příkladem může být systém *Mentat* založený na sondách v počítačových sítích, který je provozován bezpečnostním týmem CESNET-CERTS [24]. V případě detekce možného problému je systémem *Mentat* automaticky zasláno hlášení konkrétnímu bezpečnostnímu týmu na adresu `abuse@domain.tld` (viz obr. 3.5).

Dalším zdrojem externích hlášení jsou také vlastníci autorských práv, z jejichž strany jsou doručovány stížnosti na uživatele, kteří sdílejí autorsky chráněný obsah, a připravují tak vlastníky o potenciální zisk z prodeje děl.

3.3 Vyhodnocení a analýza – *Triage and Analysis*

Po každém hlášení incidentu dochází nejprve k ověření, zdali k nahlášené události skutečně došlo, a to na základě logů. Jsou tak vyloučena možná falešně pozitivní hlášení. Zároveň dochází k zajištění důkazů, že konkrétní zařízení nebo uživatel skutečně prováděl neoprávněné akce. Na základě důkazů je následně možné omezit nebo zablokovat zařízení, případně uživatele. V případě stížností ze strany blokovaného lze oprávněnost a nutnost akce doložit zajištěnými důkazy. [24]

Po ověření legitimnosti bezpečnostního incidentu dochází k posouzení závažnosti (*triage*). Závažnost definuje, jaký dopad má incident na organizaci – tj. kolik zařízení, uživatelů nebo služeb je nebo může být v organizaci daným problémem zasaženo, úrovní zasažení a jaké důsledky má problém na data v souvislosti s triádou kybernetické bezpečnosti CIA (viz kapitola 2) [24].

Cílem analýzy (*analysis*) je získat podrobnější informace o rozsáhlosti a závažnosti incidentu (tj. o typu a počtu zařízení, jež jsou incidentem zasažena) a na základě získaných informací zvolit vhodný postup řešení incidentu [24]. Zároveň dochází k identifikaci konkrétního problémového uživatele nebo napadeného zařízení.

Na základě závažnosti a paralelně běžící analýzy je zvolen vhodný postup řešení [24]:

- informování problémového uživatele nebo správce napadeného zařízení,
- vyzvání ke zjednání nápravy,
- omezení služeb nebo zablokování přístupu,
- případně ignorování problému (u zanedbatelné šance na šíření).

Součástí je zabránění dalšímu šíření problému a provedení adekvátního opatření – např. omezení konkrétních portů, služeb, IP rozsahů, upgrade operačního systému, aplikací a dalších.

3.4 Izolace a neutralizace – *Containment and Neutralization*

Cílem je využití výsledků analýzy (viz podkapitola 3.3) a izolování problémových zařízeních a uživatelů tak, aby nemohli pokračovat v dosud prováděné činnosti a problém se dál nešířil (*containment*) [24]. Typicky se jedná o činnosti jako je odpojení napadených zařízení nebo blokace napadených účtů [25]. Zároveň musí být uvažována i podmnožina dalších zařízení nebo účtů, které by mohly být daným bezpečnostním incidentem také ovlivněny (napadeny) a případně u nich aplikovat patřičné kroky (například instalace *hotfix* aktualizace, izolace zařízení od *Internetu* apod.).

Neutralizace spočívá v odstranění vzniklých škod (*neutralization*) [24]. Jedná se o dosažení normálního, provozuschopného stavu [20]. Technicky se neutralizace týká odstranění činnosti útočníka – odstranění malware, reinstalace operačního systému, provedení změny hesel apod. [25].

3.5 Závěrečná činnost – *Post-Incident Activity*

Po každém netriviálním bezpečnostním incidentu je vhodné provést zhodnocení přípravy, zhodnocení průběhu řešení incidentu nebo lokalizovat slabá místa a z případných problémů se poučit (*lesson learned*) [24, 25].

Součástí závěrečné činnosti je například zjišťování původu problému a hledání způsobu, jak se podobnému problému v budoucnu vyhnout. Analýzou použitých postupů lze odhalit, že některé postupy nebyly při řešení incidentu dostatečně efektivní nebo byly příliš pomalé a bylo by vhodné, je prověřit a vylepšit (např. CSIRT týmu chybělo povolení k provedení konkrétní akce, a bylo by tak vhodnější, aby bylo povolení získáno v předstihu a reakce na incident byla rychlejší). Také je třeba zvážit negativní ohlasy od uživatelů k provedeným opatřením – např. přehnaná reakce bezpečnostního týmu na banální problém. [24]

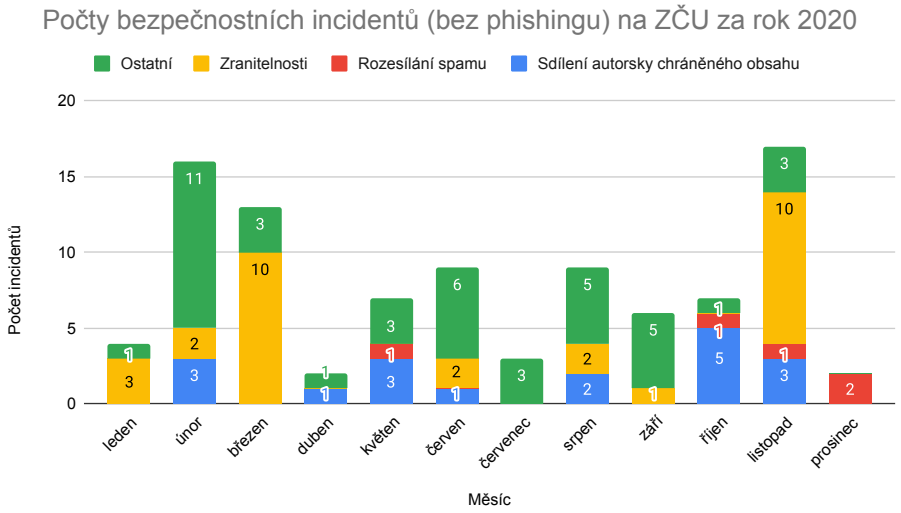
U rozsáhlejších bezpečnostních incidentů je doporučeno po určitou dobu (v řádu 1–2 týdnů) provádět kontrolu logů a monitoring sítě tak, aby bylo zaručeno, že se stejný problém nezačne projevovat znovu, ale na jiném místě (např. u dalších infikovaných pracovních stanic, které ale byly v době řešení incidentu vypnuté a k jejich zapnutí došlo až po vyřešení incidentu) [24].

4 Bezpečnostní incidenty na ZČU

V následujících podkapitolách budou popsány nejčastější a nejzávažnější typy bezpečnostních incidentů, které jsou nebo byly řešeny univerzitním bezpečnostním týmem WIRT na ZČU.

Protože počet řešených bezpečnostních incidentů na ZČU neustále narůstá, bylo by vhodné mít k dispozici software, který týmu WIRT řešení incidentů usnadní a zefektivní. Systém implementovaný v rámci diplomové práce bude pomáhat s řešením incidentů, které se typicky týkají uživatelů nebo zařízení přímo připojených do univerzitní sítě *WEBnet*. Systém bude dále označován zkratkou CAIH (*Computer-aided Incident Handling*).

Díky různorodosti univerzitního prostředí je třeba uvažovat, že do univerzitní počítačové sítě jsou zapojeny nejen tisíce uživatelů a univerzitních zařízení (včetně běžných klientských stanic, popř. různých výzkumných zařízení), ale i tisíce externích zařízení, a to jak ze strany studentů, tak ze strany zaměstnanců. Vlivem různorodosti prostředí dochází k výskytu všech možných bezpečnostních incidentů, nicméně jen několik z incidentů se opakuje ve větším měřítku (viz graf na obr. 4.1).



Obrázek 4.1: Počty ručně řešených (tzn. problém nebyl vyřešen automatickými nástroji) bezpečnostních incidentů na ZČU za rok 2020 (bez započítání hlášení podvodných e-mailů, viz samostatný graf v podkapitole 4.2.1).

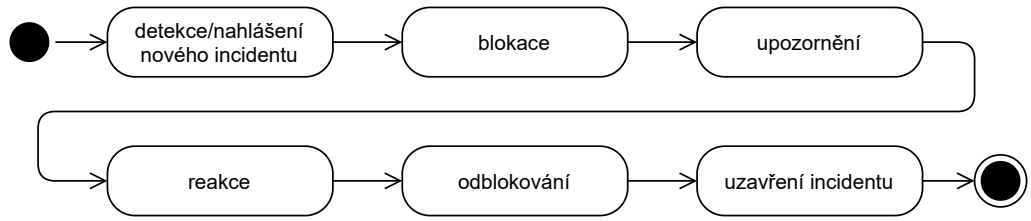
Vysoké školy (VŠ) obecně jsou navíc lákavým cílem útočníků hned z několika důvodů. V informačních systémech univerzit je uchováno velké množství citlivých a interních informací a dat včetně těch z výzkumu. Běžně lze na VŠ narazit i na zastaralý software a hardware. Navíc se v podobných institucích s ohledem na počet uživatelů obtížně zajišťuje intenzivní školení týkající se kybernetické bezpečnosti. [5]

Kromě toho se každoročně opakuje situace nástupu nových studentů univerzity do prvních ročníků, kteří se povětšinou před nástupem na vysokou školu s kyberbezpečností, resp. s pravidly počítačové sítě nikdy nesetkali.

Na *Západočeské univerzitě v Plzni* každý student nebo zaměstnanec při přebírání univerzitního, přihlašovacího *Orion konta* a průkazu (tzv. *JIS karta*) podepisuje dodržování směrnice rektora „*Pravidla používání sítě WEBnet*“ [15]. Dokument stanovuje zásady a pravidla při používání univerzitní sítě *WEBnet* včetně umožnění aktivního zásahu ze strany pracoviště CIV (*Centrum informatizace a výpočetní techniky*), pod které spadá bezpečnostní tým WIRT. Při porušení pravidel se student, popř. zaměstnanec vystavuje riziku omezení služeb nebo blokaci. V případě opakovaného porušování pravidel je podle uvedené směrnice rektora možnost zahájit i disciplinární řízení na fakultě studenta, u zaměstnance pak může být chápáno jako porušení pracovní kázně [15].

4.1 Incidenty řešené CAIH

V následujících podkapitolách budou popsány bezpečnostní incidenty, s jejichž řešením bude členům týmu WIRT pomáhat systém typu CAIH vzniklý z diplomové práce. Konkrétně se jedná o incidenty, u nichž je shodný postup u analytických a komunikačních kroků. Systém usnadní řešení popisovaných incidentů právě automatizací konkrétních kroků. Jednotný postup při zpracování následujících incidentů ukazuje zjednodušený stavový diagram na obr. 4.2.



Obrázek 4.2: Zjednodušený stavový diagram řešení incidentu.

Popis jednotlivých stavů diagramu uvedeného na obr. 4.2 je následující:

Detekce/nahlášení problémového uživatele nebo napadeného zařízení dle konkrétní IP adresy a časové značky, které jsou předmětem stížnosti. Systém v návaznosti na zadané údaje vyhledá napříč dostupnými logy jako je log pro *eduroam* připojení (*Wi-Fi*), log pro pevné (drátové) připojení, log pro VPN (*Virtual Private Network*) připojení apod. záznamy spojené se zadanou IP adresou v inkriminovaném čase. U problémových uživatelů bude k IP adrese zároveň automaticky zjištěno i jméno uživatele, čímž dojde k identifikaci konkrétního původce incidentu, který může být v případě *eduroam* připojení i z jiné organizace (např. z jiné vysoké školy).

Blokace zařízení nebo uživatele v konkrétní oblasti počítačové sítě. Díky vytvořenému systému nebude muset bezpečnostní tým manuálně upravovat konfigurační soubory nebo data v dalších systémech, v nichž jsou uvedeni zablokovaní uživatelé.

Upozornění zablokovanému uživateli, nebo správci zablokovaného zařízení o provedené blokaci. E-maily budou založeny na parametrizovaném sdělení, jak uvádí jedno z doporučení v kapitole 3.1.5. S tím souvisí i zaslání upozornění stěžovali o vyřešení incidentu a evidence bezpečnostního incidentu v interním systému pro správu požadavků (jak je popsáno v kapitole 2.4).

Reakce zablokovaného uživatele nebo správce zablokovaného zařízení k incidentu a provedené blokaci.

Odblokování uživatele nebo zařízení je inverzní úkon k blokaci, který je doprovázený rozesláním příslušných e-mailových zpráv o odblokování.

Uzavření bezpečnostního incidentu v CAIH systému a v interním systému pro správu požadavků.

Stejný postup nyní v manuálním režimu bezpečnostní tým WIRT používá a přechod k automatizované verzi umožní zefektivnit průběh řešení celého incidentu. Kroky prováděné manuálně členy bezpečnostního týmu navíc vyžadují oprávnění k jednotlivým serverům, profesní zkušenosti a postupy, což bude díky vytvořenému systému usnadněno.

Konkrétní typy incidentů, které budou v systému obvykle zpracovávány a se kterými bezpečnostnímu týmu vzniklý systém usnadní práci, jsou uvedeny v následujících podkapitolách.

4.1.1 Sdílení autorsky chráněného obsahu

Typicky se jedná o incidenty spojené se sdílením autorsky chráněného obsahu pomocí P2P (*Peer-to-peer*) sítí. Stejně tak se ale může jednat o případ s vystavením díla na některé z webových stránek, jež jsou pod správou univerzity.

Pokud je sdílení autorsky chráněného obsahu detekováno vlastníkem práv, pak následuje zaslání stížnosti bezpečnostnímu týmu instituce (tedy např. univerzitě), z níž pochází IP adresa uživatele, který dílo sdílel nebo URL adresa, kde je dílo k dohledání. Stížnost obsahuje název sdíleného díla (např. název filmu, knihy), IP adresu, která se incidentu účastnila, časovou značku a další údaje.

Bezpečnostní tým na základě údajů uvedených ve stížnosti identifikuje konkrétního uživatele (pomocí logů), který autorsky chráněné dílo v uvedený čas sdílel. V případě pozitivního nálezu následuje omezení služeb uživatele, a to zablokováním přístupu do sítě *WEBnet* tak, aby uživatel v činnosti nemohl nadále pokračovat a nebylo poškozeno dobré jméno univerzity. Pokud bylo dílo uloženo na serverech univerzity, dojde k jeho odstranění.

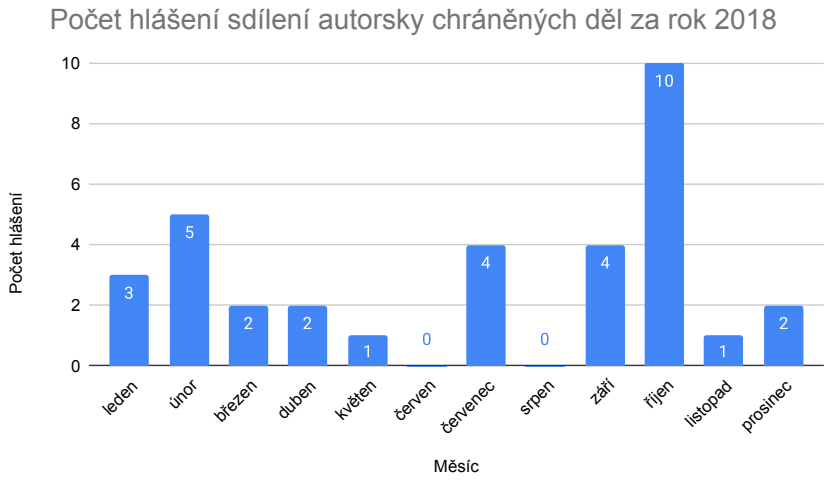
Celý incident vyústí minimálně v osobní pohovor člena bezpečnostního týmu WIRT s uživatelem, v rámci nějž je uživatel poučen a seznámen s porušením pravidel, resp. se směrnicí rektora [15]. V případě studentů následuje dvouměsíční omezení možnosti využívat nadstandardní služby univerzitní sítě *WEBnet* (např. kolejní síť, bezdrátové *eduroam* připojení).

Jednotlivé kroky při řešení incidentu ukazuje tabulka 4.1.

Detekce nebo **nahlášení**	Sondy v sítích detekující P2P provoz
	Hlášení cizích organizací o sdílení autorsky chráněného obsahu
Ověření	Analýza *NetFlow* a logů
Izolace	Blokace uživatele pro přístup do sítě a jeho upozornění
Neutralizace	Odstranění autorsky chráněného díla, pohovor s uživatelem
Odblokování	Odblokování uživatele a uzavření incidentu

Tabulka 4.1: Kroky při řešení sdílení autorsky chráněného obsahu.

Se sdílením autorsky chráněného obsahu se bezpečnostní tým WIRT setkává především při nástupu studentů prvních ročníků, popř. u zahraničních studentů, kteří jsou na univerzitě na výměnném studijním pobytu. Situaci reflektuje i graf na obr. 4.3 uvádějící počet zaznamenaných incidentů sdílení v roce 2018, kde je vidět abnormální vzestup incidentů na počátku zimního semestru nového akademického roku.



Obrázek 4.3: Počet hlášení sdílení autorsky chráněného obsahu za rok 2018 bezpečnostnímu týmu WIRT na ZČU. V grafu si lze všimnout výrazné změny v měsíci říjnu, kdy začíná zimní semestr nového akademického roku a spolu s ním probíhá nástup prvních ročníků.

4.1.2 Napadená zařízení

Zařízení infikovaná škodlivým programem (tzv. *malware*) mohou být například zdrojem nevyžádaných e-mailů (podkapitola 4.1.3), mohou neoprávněně skenovat síť organizace (podkapitola 4.1.4), mohou být zdrojem DoS nebo DDoS útoků (podkapitola 4.1.5) nebo na nich může být umístěna například podvodná, phishingová stránka (podkapitola 4.1.7).

Velmi závažným incidentem je především výskyt ransomwaru, jehož činností je obvykle zasažená organizace významně paralyzována. Jedná se o komerčně motivovaný útok, kde je zdrojem nákazy nechtěný program, jehož cílem je zašifrovat data uživatele a následně po uživateli požadovat zaplacení výkupného výměnou za dešifrování dat (druh ransomware označovaný jako *filecoders*) [11]. Zdrojem ransomwaru jsou například zavirované přílohy ve phishingových e-mailech [11]. Minimalizaci škod je možné zabránit důsledným zálohováním dat, vzděláváním uživatelů v oblasti phishingu, dostatečnou segmentací sítě apod. [11, 24].

Infikované počítače, napadené ať už ransomwarem nebo jiným druhem malwaru, je nutné ve většině případů přeinstalovat. Rovněž je nutné zjistit, zdali se malware nerozšířil i na další zařízení v počítačové síti organizace.

Postup bezpečnostního týmu u napadených zařízení je uveden v tabulce 4.2.

Detekce nebo **nahlášení**	Sondy v sítích a logy (podezřelá aktivita zařízení)
	Vlastník nebo správce zařízení, jiné CSIRT týmy
Izolace	Blokace IP adresy zařízení, uživatele nebo konkrétního portu na switchi a upozornění vlastníka zařízení
Neutralizace	Reinstalace operačního systému napadeného zařízení
Odblokování	Odblokování IP adresy a uzavření incidentu
Závěrečná činnost	Kontrola, zdali nejsou incidentem dotčeny i další zařízení, u kterých se problém zatím nemusel projevit

Tabulka 4.2: Kroky při řešení napadených zařízení.

4.1.3 Rozesílání nevyžádané pošty

Rozesílání nevyžádaných e-mailů má souvislost s napadenými zařízeními uživatelů, které vykazují na některých portech podezřelé chování nebo abnormálně vysokou aktivitu, jenž může být důsledkem právě rozesílání nevyžádané pošty. Nezvyklou aktivitu zaznamenají i v síti umístěné sondy (viz obr. 4.4) nebo záznamy v logu poštovního serveru.

	zdrojová IP	jméno	počet spojení	počet cílů
1.	147.228.**.***	ek***n**-kte.fel.zcu.cz	1584	1
2.	147.228.244.82	zeus-db.zcu.cz	1112	1
3.	147.228.***.**	eduroam-***-***.zcu.cz	786	343
4.	147.228.58.48	dspace5.zcu.cz	665	1

Obrázek 4.4: Část výstupu sondy detekující vysoký počet spojení na protokolu SMTP. Jedno ze zařízení (ve výstupu označeno řádkem č. 3) za jediný den provedlo 786 spojení na 343 cílů. Analýzou bylo zjištěno, že se jednalo o notebook napadený malwarem, který rozesílal podvodné e-maily.

Zařízení, u něhož byla aktivita zaznamenána, je po prověření zablokován přístup do sítě tak, aby kompromitované zařízení nemohlo v činnosti nadále pokračovat. Jednotlivé kroky při řešení incidentu ukazuje tabulka 4.3.

Detekce nebo **nahlášení**	Sondy v sítích (vysoká aktivita na určitých portech) a logy
	Hlášení o nevyžádaných e-mailech, ve kterých je jako odesílatel e-mailů uvedeno napadené zařízení
Ověření	Analýza *NetFlow* a logů
Izolace	Blokace IP adresy zařízení a upozornění vlastníka
Neutralizace	Odstranění malwaru
Odblokování	Odblokování IP adresy a uzavření incidentu

Tabulka 4.3: Kroky při řešení rozesílaní nevyžádané pošty.

4.1.4 Skenování sítě

Skenování sítě je činnost, jejímž cílem je hrubou silou prozkoumat další IP adresy v síti (např. v celém rozsahu) a zjistit, jaké služby (a na jakých portech) na jednotlivých zařízeních běží [24]. Případně může být skenování omezeno jen na konkrétní port nebo skupinu portů (například na služby, které v daném čase trpí konkrétní zranitelností) s cílem zjistit, kolik zařízení v síti je potenciálně napadnutelných [24].

Může se jednat o činnost způsobenou cíleně uživatelem při využívání nástroje `nmap` a jiných (viz obr. 4.5), popř. činností malwaru bez vědomí uživatele. Rovněž může ale jít o falešně pozitivní hlášení, kdy skenování sítě způsobí například antivirový program nebo protokol SMB (*Server Message Block*) při vyhledávání sdílených disků.

```
root@ntb:~$ nmap phishingator.zcu.cz
Starting Nmap 7.70 ( https://nmap.org ) at 2021-03-09 10:36 CET
Nmap scan report for phishingator.zcu.cz (147.228.54.100)
Host is up (0.011s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 11.54 seconds
```

Obrázek 4.5: Výstup po spuštění nástroje `nmap` nad subdoménou `phishingator.zcu.cz` (`147.228.54.100`). Nástroj na dané IP adrese detekoval tři otevřené porty – SSH, HTTP a HTTPS.

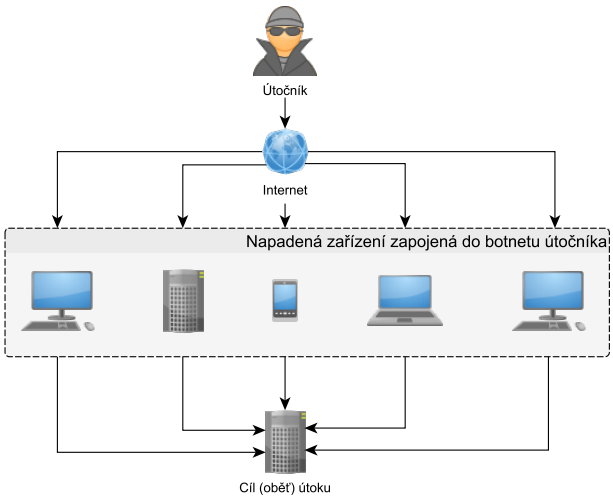
Každý incident detekovaný sondami jako skenování sítě tak musí být detailně prozkoumán bezpečnostním týmem. V případě pozitivního hlášení dochází k zablokování zařízení (viz postup v tabulce 4.4).

Detekce nebo **nahlášení**	Sondy v sítích (síťové toky v *NetFlow* na různé IP adresy a porty v rozsahu organizace)
	Hlášení cizích organizací na neoprávněné skenování jejich sítě
Ověření	Analýza *NetFlow* a logů
Izolace	Blokace IP adresy zařízení a upozornění vlastníka
Neutralizace	Odstranění malwaru, při cíleném skenování poučení uživatele o používání skenovacích nástrojů
Odblokování	Odblokování IP adresy a uzavření incidentu

Tabulka 4.4: Kroky při řešení neoprávněného skenování sítě.

4.1.5 DoS a DDoS útoky

Předmětem několika incidentů jsou útoky typu DoS (*Denial of Service*), případně jejich obdoba, distribuované DoS útoky (DDoS), kdy je útok prováděn z několika uzlů naráz (viz obr. 4.6). Cílem (D)DoS útoku je omezit nebo vyřadit služby počítačových systémů [6]. Může se jednat buď o generování nadměrného počtu podvržených požadavků s cílem zahltit systém nebo přenosovou cestu [6]. Případně se může jednat o sofistikovaný útok na předem zvolená, slabá místa v cílovém systému nebo přenosové cestě [6]. Zároveň se ovšem nemusí vždy jednat o cílený útok, ale například o využívání běžného nástroje nevhodným způsobem, který svou činností zahltí síť nebo některý z uzlů, a způsobí tak útok typu DoS [24].



Obrázek 4.6: Příklad útoku typu DDoS – napadená zařízení (též označovaná jako *zombies*) na podnět útočníka (pachatele) naráz útočí na konkrétní cíl, aby jej omezila nebo vyřadila z provozu.

Útoky tohoto typu mohou pocházet jak ze zařízení, která jsou napadena malwarem (typicky jsou zařízení součástí botnetu a čekají na pokyn útočníka – tzv. *zombies*), tak ze zařízení, která například chybným nastavením produkují nadměrný provoz, a mohou tak významně zatěžovat některý z uzlů sítě (např. abnormálně zatěžovat DHCP server).

Dalším příkladem útoku může být na první pohled nevinná snaha uživatele vytvořit si tzv. *crawler* [24]. Crawler je nástroj určený k automatickému stahování webových stránek z konkrétní domény. Chování podobného nástroje může být ovšem vyhodnoceno jako útok typu DoS, zvlášť, pokud není mezi jednotlivými požadavky na server dostatečný časový odstup.

Jednotlivé kroky při řešení incidentu ukazuje tabulka 4.5.

Detekce nebo **nahlášení**	Sondy v síti (vysoká aktivita z konkrétních IP adres), logy
	Hlášení cizích organizací o nezvykle vysoké aktivitě, která abnormálně vytěžuje jejich síť nebo některé z uzlů
Ověření	Analýza *NetFlow* a logů
Izolace	Blokace IP adresy zařízení, nebo IP rozsahu sítě, ze které pochází útok DDoS a upozornění původce incidentu
Neutralizace	Odstranění malwaru, případně poučení uživatele při nesprávném používání nástrojů (např. *crawler*) nebo při nevhodné konfiguraci zařízení, které způsobuje abnormální provoz
Odblokování	Odblokování IP adresy a uzavření incidentu

Tabulka 4.5: Kroky při řešení DoS nebo DDoS útoků.

4.1.6 Zranitelná zařízení a otevřené porty

Častým úkolem bezpečnostních týmů je průzkum do *Internetu* otevřených (přístupných) zařízení organizace a jejich portů – např. nezabezpečených tiskáren, zranitelných NAS (*Network Attached Storage*) serverů apod.

Zjišťování většinou podléhá předchozímu varování týkajícího se zranitelnosti konkrétních služeb, přičemž jednotlivé kroky bezpečnostního týmu ukazuje tabulka 4.6.

Detekce nebo **nahlášení**	Analýza *NetFlow*, logů, webových rozhraní zařízení
	Hlášení od jiných CSIRT týmů nebo od autora zranitelné služby
Ověření	Nalezení zranitelných zařízení skenováním sítě
Izolace	Blokace IP adresy/portu zneužitelného zařízení, které je přístupné z *Internetu* (v opodstatněných případech)
Upozornění	Upozornění vlastníků zranitelných zařízení, aby zjednali nápravu a zařízení zabezpečili nebo aktualizovali
Odblokování	Případné odblokování IP adresy/portu a uzavření incidentu

Tabulka 4.6: Kroky při řešení zranitelných zařízení.

4.1.7 Napadené webové projekty

Mezi závažné bezpečnostní incidenty spadá napadení webových stránek, které běží na doméně nebo subdoménách organizace.

Ve většině případů se jedná o webové projekty, které jsou spravovány pomocí globálně známých redakčních systémů označovaných zkratkou CMS

(*Content Management System*), mezi něž se řadí například *WordPress*, *Joomla* nebo *Drupal*.

Vzhledem k tomu, že se často jedná o jednorázové webové projekty, které následně nejsou udržovány, a jádro CMS tak není nikým aktualizováno, je pro útočníky snazší takové projekty napadnout. V průběhu času bývá navíc v některé ze starších verzí redakčního systému odhalena bezpečnostní zranitelnost a je zveřejněn postup jejího zneužití (tzv. CVE – *Common Vulnerabilities and Exposures*). Útočníci na zveřejněné zranitelnosti reagují skenováním potenciálně napadnutelných webových projektů (resp. zranitelných verzí) a v případě pozitivního nálezu jejich napadením. Podobně reagují i bezpečnostní týmy, jejichž cílem je uvnitř jimi spravované domény potenciálně napadnutelné systémy lokalizovat a napadení předejít.

Útočníkům také vyhovuje zneužití webu na dosud důvěryhodné doméně, která není zanesena v žádných *blacklistech* (seznam nebezpečných stránek) a uživatel na ní neočekává případnou hrozbu. Napadené webové stránky na důvěryhodné doméně se tak mohou stát nebezpečným prostředníkem mezi útočníkem a uživatelem. Postup při řešení incidentu ukazuje tabulka 4.7.

Detekce nebo **nahlášení**	Skenování zranitelností CMS systémů, analýza logů, nalezení podvodné stránky u webového projektu
	Hlášení od jiných CSIRT týmů, hlášení od vlastníka webu
Ověření	Analýza zdrojového kódu a logů
Izolace	Blokace IP adresy, kde běží napadený web a upozornění vlastníka webu
Neutralizace	Odstranění malwaru, přeinstalace CMS
Odblokování	Odblokování IP adresy a uzavření incidentu

Tabulka 4.7: Kroky při řešení napadených webových projektů.

4.2 Incidenty mimo oblast CAIH

Do této kategorie spadá velké množství typů bezpečnostních incidentů, které se ovšem mezi sebou výrazně liší. Díky různorodosti je nutné každý z následujících incidentů řešit unikátním způsobem, a člen CSIRT týmu tak musí většinu kroků vykonat manuálně. Incidentů stejných typů (tj. shodný postup řešení) je navíc poměrně málo a zařazení do CAIH systému by nebylo ekonomicky výhodné. Evidence incidentů je navíc provedena už v systému pro správu požadavků a nemá smysl záznamy duplikovat. Dále popisované incidenty tak nebudou určeny ke zpracování ve vyvíjeném CAIH systému.

4.2.1 Podvodné e-maily – *Phishing*

Mezi nejčastěji detekované bezpečnostní události na ZČU patří *phishing*. Jedná se o podvodné e-maily běžně svázané s konkrétní podvodnou stránkou, jejichž cílem je zneužít uživatelem zadané údaje.

Cílem bezpečnostního týmu je, aby nedošlo k navštívení podvodné stránky a vyplnění údajů nepozornými uživateli a následnému zneužití zadaných údajů. Postup při řešení podvodných e-mailů obsahujících odkaz na podvodnou stránku ukazuje tabulka 4.8.

Detekce nebo **nahlášení**	Obdržení phishingu
	Hlášení od uživatelů organizace o obdržení phishingu
Ověření	Analýza phishingu včetně dostupnosti podvodné stránky
Izolace	Blokování konkrétní URL adresy (nebo domény) s podvodnou stránkou v síti organizace a nahlášení URL podvodné stránky do *blacklistů*, které jsou pod správou mezinárodních organizací (např. `phishtank.com` a `safebrowsing.google.com`)
Závěrečná činnost	Upozornění uživatelů organizace (v případě nebezpečného phishingu), školení uživatelů o problematice phishingu

Tabulka 4.8: Kroky při řešení phishingu s odkazem na podvodnou stránku.

V případě, že je některý z phishingových útoků cílen přímo na uživatele organizace (tzv. *spear phishing* – např. u ZČU by se mohlo jednat o podvodnou stránku nápadně připomínající univerzitní přihlašovací stránku), pak musí CSIRT tým provést ověření, zdali některý z uživatelů do podvodné stránky nezadal své přihlašovací údaje. Existuje riziko, že útočník začne zneužívat získané identity uživatelů k rozesílání phishingu z poštovního serveru organizace [18].

Pokud dojde ke zjištění, že je podvodná stránka hostována na zařízení (resp. webovém serveru) v organizaci, je k němu přistupováno jako k napadenému zařízení (webu), viz podkapitola 4.1.7.

Další problém představují phishingové e-maily obsahující zavirovanou přílohu, která se vydává za falešnou fakturu, fiktivní předvolání k soudu nebo jiný, důležitý dokument od známé autority. O podobném problému pojednává podkapitola 4.1.2.

Jen v roce 2020 bezpečnostní tým WIRT řešil 321 hlášení phishingu ze strany uživatelů (viz graf na obr. 4.7).

Obrázek 4.7: Počet hlášení phishingu za rok 2020 ze strany uživatelů bezpečnostnímu týmu WIRT na ZČU.

4.2.2 Úniky přihlašovacích údajů

Další událostí, se kterou se bezpečnostní tým WIRT běžně setkává, jsou úniky přihlašovacích údajů (tzv. *credential leaks*). Jedná se o úniky přihlašovacích údajů (identit uživatelů) ze služeb třetích stran, do nichž se uživatel sítě *WEBnet* registroval univerzitním e-mailem (ve službě byl e-mail uveden například jako kontaktní e-mail, nebo jako přihlašovací jméno).

Postup bezpečnostního týmu při řešení incidentu ukazuje tabulka 4.9.

Detekce nebo **nahlášení**	Cílené vyhledávání veřejně dostupných, uniklých databází s identitami uživatelů
	Nahlášení databáze jiným bezpečnostním týmem
Ověření	Analýza, zdali jsou uživatelé organizace únikem identit ohroženi (např. v případě ZČU na základě výskytu vzoru `*@*.zcu.cz` v některém ze záznamů) a kontrola, zdali uživatelé v organizaci stále pracují nebo studují (může se jednat o databázi staršího data)
Upozornění	Upozornění dotčených uživatelů o odcizené identitě s doporučením změny hesla minimálně v univerzitních informačních systémech

Tabulka 4.9: Kroky při řešení úniku přihlašovacích údajů.

Uniklé informace bývají zveřejňovány například ve formátu `e-mail:value`, kde `value` může být heslo uvedené v *cleartextu* (tj. nebylo

před únikem zašifrováno). Název služby, ze které k úniku identit došlo, nemusí být specifikován. Navíc se může jednat o agregovaná data z několika služeb.

4.2.3 Bezpečnostní chyby v informačních systémech

Mezi velmi závažné incidenty se řadí bezpečnostní chyby v informačních systémech. V případě nalezení bezpečnostní chyby útočníkem by mohlo dojít k úniku osobních dat studentů a zaměstnanců a poškození dobrého jména univerzity nebo jiné organizace. Kroky při řešení incidentu ukazuje tabulka 4.10.

Detekce nebo **nahlášení**	Cílené vyhledávání chyb (testování vstupních parametrů apod.), automatické nástroje testující bezpečnostní chyby
	Hlášení cizích organizací o nalezené bezpečnostní chybě, hlášení od autora aplikace
Ověření	Simulace chyby na základě jejího detailního zdokumentování z předchozího kroku
Neutralizace	Oprava nalezené chyby a nasazení opravené verze informačního systému do produkčního prostředí
Závěrečná činnost	Průběžná kontrola logů, zdali k chybě již nedochází, ale například na jiném místě

Tabulka 4.10: Kroky při řešení chyby v informačních systémech.

Podobné informační systémy, jako například univerzitní IS/STAG (*Informační systém studijní agendy*), jsou tak na základě zákona o kybernetické bezpečnosti zařazeny mezi *významné informační systémy* (VIS).

Důkladný audit například univerzitního systému IS/STAG vyvíjeného pracovištěm CIV na ZČU, který je poskytován i několika dalším vysokým školám v ČR, je v pravidelných intervalech realizován *Forenzní laboratoří* (FLAB) CESNET.

4.2.4 Zveřejňování interních dokumentů

Zveřejňování interních dokumentů organizace souvisí s případy, v rámci nichž jsou dokumenty vystaveny na veřejně přístupných místech (například výpisy adresářů webového serveru), kde mohou být posléze zaindexovány internetovými vyhledávači, které zajistí nechtěný přístup k dokumentům i veřejnosti.

Zaměstnanec organizace může například za cílem výměny dokumentů s jiným kolegou umístit soubory obsahující citlivé informace na server orga-

nizace, resp. webovou stránku, která ale není zabezpečena proti neoprávněnému přístupu. Zaměstnanec ovšem může zapomenout dokumenty smazat, a soubory tak zůstanou přístupné veřejnosti.

Jednotlivé kroky při řešení incidentu jsou znázorněny v tabulce 4.11.

Detekce nebo **nahlášení**	Skenování a procházení veřejně přístupných adresářů a souborů
	Hlášení od jiných CSIRT týmů nebo jednotlivců
Ověření	Analýza logů, ověření existence dokumentu
Neutralizace	Odstranění dokumentu, poučení uživatele a případné administrativní kroky související se zveřejněním dokumentu

Tabulka 4.11: Kroky CSIRT týmu při řešení veřejně dostupných, interních dokumentů.

4.2.5 Žádosti o součinnost s orgány činnými v trestním řízení

Bezpečnostní tým (resp. oslovená organizace) také na základě žádosti spolupracuje s orgány činnými v trestním řízení (dále OČTŘ). Žádost o součinnost na vyšetřování je předána a formálně schvalována ředitelem pracoviště, který ji následně odevzdává k řešení bezpečnostnímu týmu [24].

Žádost o součinnost je spojena především s velmi závažnými bezpečnostními incidenty přesahující oblast univerzitní sítě, popř. oblast státu [20]. Z pohledu bezpečnostního týmu je úkolem především analýza logů činnosti podezřelých uživatelů (resp. konkrétních IP adres, které jsou předmětem vyšetřování), případně průběžné zajišťování důkazů o chování podezřelých uživatelů pro potřeby OČTŘ. Běžné, počáteční kroky při řešení incidentu jsou uvedeny v tabulce 4.12, přičemž další kroky závisí na pokynech OČTŘ.

Nahlášení	Žádost o součinnost od orgánů činných v trestním řízení
Ověření	Analýza *NetFlow* a logů

Tabulka 4.12: Běžné, počáteční kroky při řešení žádostí o součinnost s OČTŘ.

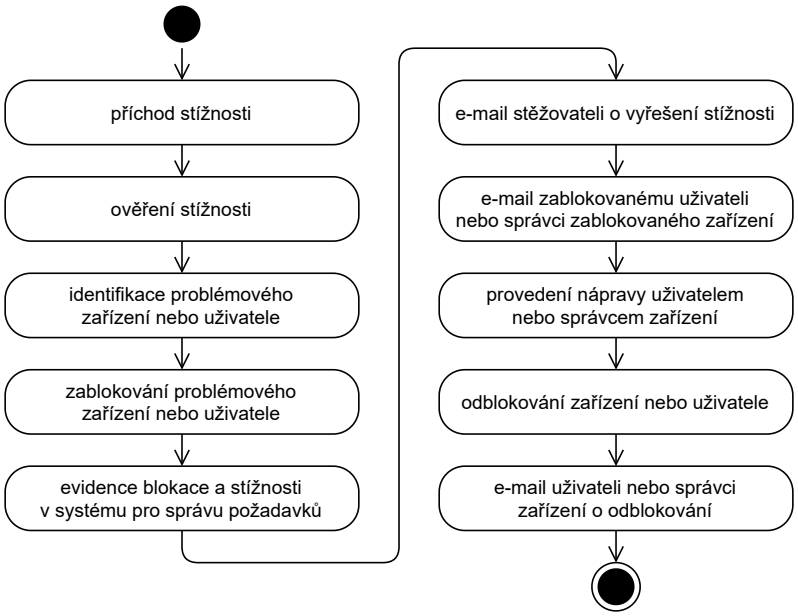
Vzhledem k tomu, že jde o činnost související s probíhajícím vyšetřováním, je do podobné činnosti zapojeno minimální množství osob. O výsledku a dalším průběhu vyšetřování není bezpečnostní tým povětšinou informován.

5 Implementace

V současné době je na ZČU využíván vlastní software *MySphere2*, který dokáže některé z kroků používaných při řešení incidentu automatizovat [17].

Systém *MySphere2* je již ovšem k dnešku zastaralý (implementace proběhla v roce 2011 [17]), řadu věcí neumožňuje provádět a neposkytuje jednotné rozhraní pro řešení incidentů z různých typů sítě (*eduroam*, VPN, pevná síť). Systém navíc není dostatečně intuitivní a pro využívání vyžaduje praxi uživatele. Některé z klíčových vlastností systému jsou navíc i pevně svázány (tzv. *hard coding*) se zdrojovým kódem aplikace (například text odesílaných e-mailů). Systém *MySphere2* umožňoval v době psaní diplomové práce automaticky vyhledat, zablokovat a odblokovat konkrétního uživatele ZČU v síti *eduroam*. Pevná síť a síť VPN není systémem *MySphere2* plně podporována. Systém dosud pouze umožňoval odesílat přednastavené texty e-mailů, ale blokace musela být provedena mimo tento systém.

Na základě současného stavu je cílem implementovat nový systém typu CAIH, který členům bezpečnostního týmu WIRT usnadní řešit incidenty, u nichž je shodný postup u analytických a komunikačních kroků. Jednotlivé kroky řešení u incidentů zmíněných v kapitole 4.1 totiž ukazují, že postup řešení každého incidentu je téměř stejný a odpovídá diagramu na obr. 5.1.



Obrázek 5.1: Diagram řešení bezpečnostního incidentu v systému typu CAIH, který bude výstupem diplomové práce.

Mezi hlavní požadavky na nově vytvářený systém patří:

- *intuitivnost a přehlednost celého systému* (vypovídající grafické rozhraní umožňující systém používat bez nutnosti nahlížet do manuálu),
- *zjednodušení procesu řešení incidentu* bez nutnosti přepínání do jiných, již existujících systémů (tj. dostatečná integrace) a s tím spojené snížení nutné kvalifikace pro ovládání systému,
- *konzistence* (zavedení jednotných šablon pro komunikaci s vnějším světem a jednotného způsobu řešení incidentu pro různé typy incidentů, ke kterým dochází v různých typech sítí),
- *modulárnost* (možnost přidání nového modulu bez nutnosti významného zásahu do jádra systému).

Z pohledu funkčnosti umožní nově vytvořený systém identifikovat konkrétního, problémového uživatele nebo zařízení na základě logů, zablokovat nebo odblokovat jeho činnost, a to v následujících částech univerzitní počítačové sítě *WEBnet*:

- bezdrátová síť *eduroam*:
  - uživatelé z organizace ZČU,
  - uživatelé mimo ZČU, kteří se do sítě *eduroam* mohou připojit pomocí jiných RADIUS serverů (např. z jiné vysoké školy nebo jiné zapojené organizace[1]),
- síť VPN,
- zařízení připojená do pevné sítě.

Součástí nového systému bude i správa šablon odpovědí pro e-maily zasílané problémovému uživateli nebo stěžovateli (viz doporučení v podkapitole 3.1.5), a to pro každý typ incidentu.

V systému bude zároveň vedena databáze všech univerzitních lokálních správců, kteří jsou schopni reagovat na problémy v síti na jednotlivých fakultách, katedrách či jiných odděleních.

[1]Seznam připojených organizací do sítě *eduroam* – `https://www.eduroam.cz/cs/pripojene_organizace`

5.1 Použité technologie

Pro vývoj aplikace byly vybrány a použity technologie, u nichž se očekává, že bude jejich funkčnost zachována i v následujících letech a nebudou podléhat rychlému zániku. Z pohledu serverových technologií (*back end*) se jedná o:

- *Apache* – webový server s nainstalovanými moduly:
  - `libapache2-mod-shib` a `shibboleth-sp-utils` umožňující provádět autentizaci uživatelů pomocí služby jednotného přihlášení *Shibboleth*,
- *PHP* – interpret jazyka PHP společně s následujícími moduly:
  - `php-curl` umožňující využívat nástroj `curl`,
  - `php-ldap` pro potřeby práce s LDAP,
  - `php-mbstring` pro potřeby *multibyte* funkcí (z důvodu diakritiky v českém jazyce),
  - `php-mysqli` pro potřeby práce s databází,
- *MySQL (MariaDB)* – databázový systém,
- *syslog-ng* – k zajištění přístupu k logům z jiných serverů (viz podkapitola 5.4),
- *Certbot* – k automatickému vydání a obnově důvěryhodného HTTPS certifikátu od certifikační autority *Let's Encrypt*.

Z pohledu *front end* technologií a knihoven zajišťující vykreslení stránky se jedná o následující výčet:

- *HTML5* a *CSS3* s využitím frameworku *Bootstrap* (viz dále),
- *Bootstrap 5.0.0* – *front end* framework (viz kapitola 7),
- *Feather v4.28.0* – sada piktogramů do frameworku *Bootstrap*,
- *Chart.js v3.1.1* – knihovna pro vykreslování grafů společně s pluginem:
  - `chartjs-plugin-datalabels` (v2.0) umožňující provést úpravu zobrazení popisků a legend u grafů.

5.2 Architektura

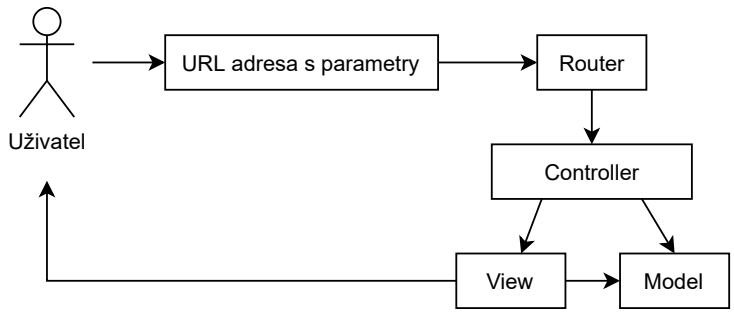
Architektura vytvořené aplikace splňuje vlastnosti třívrstvé architektury *Model–view–controller* (MVC), která se běžně využívá u webových aplikací [16]. Cílem MVC architektury je oddělit logiku aplikace od výstupu rozdělením činnosti do následujících tří vrstev [16]:

- *Model* – logika aplikace, práce s daty a databází,
- *Controller* – obsluha požadavků od uživatele,
- *View* – prezentační vrstva ve formě šablon s výstupem pro uživatele.

Architektura byla implementována ve skriptovacím jazyce PHP (*Hypertext Preprocessor*). Modulárnost a budoucí rozšíření celého systému zajišťují především jednotlivé moduly, které jsou popsány v podkapitole 5.3.

5.2.1 Komunikace mezi vrstvami

Požadavky ze strany uživatele jsou zachycovány třídou `RouterController`, která je následně podle parametrů v URL adrese předává konkrétnímu *Controlleru* (sekci v systému). Vybraný *Controller* na základě dalších parametrů v URL adrese aktivuje konkrétní šablonu, která je poté zobrazena uživateli (vrstva *View* – např. formulář pro přidání nového záznamu či výpis všech záznamů). Data pro šablonu jsou získána ze souvisejícího *Modelu*. Provázání mezi vrstvami zachycuje i obr. 5.2.



Obrázek 5.2: Proces zpracování požadavku od uživatele v použité MVC architektuře.

5.3 Moduly

Hlavními součástmi celého systému jsou moduly obsluhující konkrétní rozsah počítačové univerzitní sítě. Cílem každého z modulů je umožňovat nezávisle

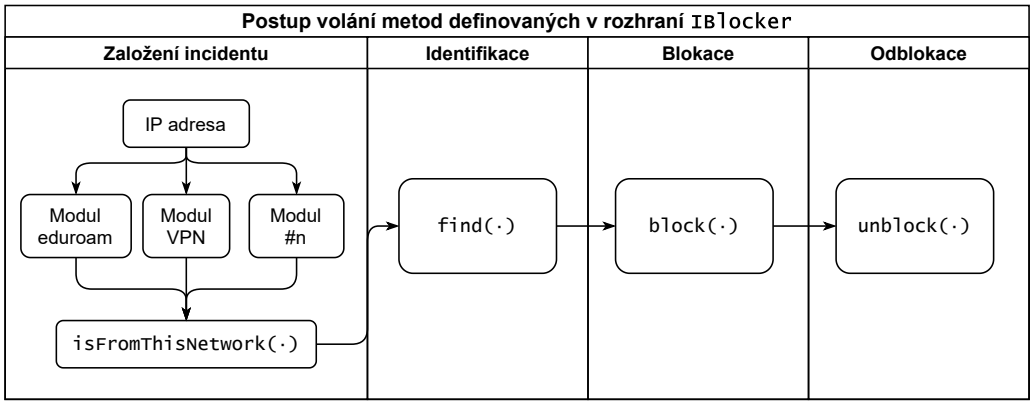
provádět činnosti v konkrétním rozsahu univerzitní sítě (tzn. jeden modul umožnuje provádět operace v síti VPN, další modul v síti *eduroam* apod.). Každý z modulů musí umožňovat:

- ověřit, zdali konkrétní IP adresa spadá do kompetence modulu, aby s IP adresou mohly být prováděny další operace (viz následující body),
- vyhledávat související záznamy v logu dané sítě,
- zablokovat konkrétního uživatele (popř. zařízení) v dané síti,
- odblokovat konkrétního uživatele (popř. zařízení) v dané síti.

5.3.1 Rozhraní

Implementovaný CAIH systém obsahuje rozhraní `IBlocker`, které definuje jednotný způsob, kterým bude systém s moduly komunikovat.

Díky použití jednotného rozhraní pro všechny moduly je zároveň zajištěno, že počet modulů může být v budoucnu rozšiřován, aniž by to vyžadovalo zásahy do jádra samotného systému (viz obr. 5.3). Systém totiž pouze očekává instance modulů, které splňují rozhraní `IBlocker`.



Obrázek 5.3: Schéma znázorňující volání metod rozhraní `IBlocker` v závislosti na aktuálním kroku řešení incidentu. Během prvního kroku se na základě IP adresy vybere vhodný modul a ten je následně využíván v následujících krocích.

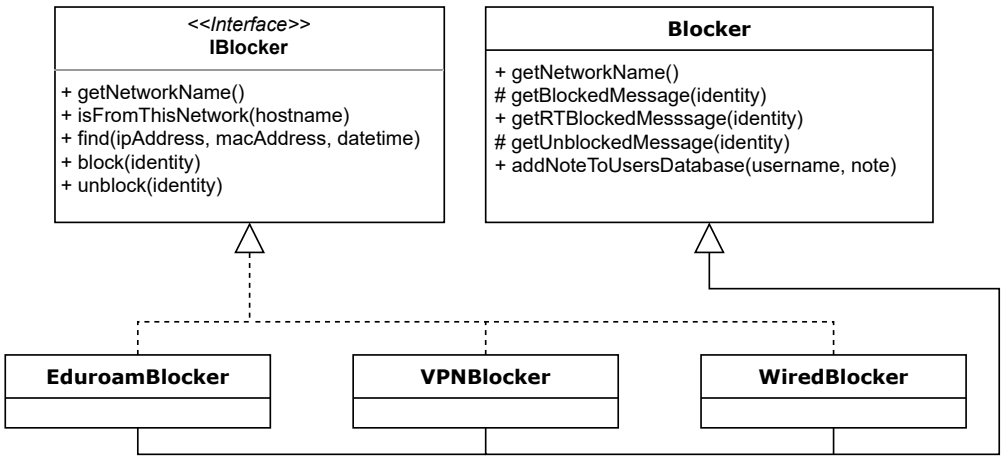
Rozhraní `IBlocker` obsahuje hlavičky metod, které musí každý z modulů implementovat. Jedná se o metody:

- `getNetworkName()` – vrátí označení části sítě, kterou modul umožňuje spravovat,

- `isFromThisNetwork(·)` – ověří, zdali IP adresa (resp. hostname) spadá do kompetence daného modulu a zdali modul umožňuje pro danou IP adresu vyhledat uživatele (zařízení), který ji měl v konkrétní čas přidělenou a případně ji zablokovat nebo odblokovat (viz další metody),
- `find(·)` – pokusí se vyhledat konkrétní identitu v logu,
- `block(·)` – zablokuje konkrétní identitu,
- `unblock(·)` – odblokuje konkrétní identitu.

Každý z modulů zároveň dědí od třídy `Blocker`, která všem modulům poskytuje obecné metody, jež mohou být programátorovi během implementace užitečné (například sestavení krátké, jednotné zprávy o provedení blokace).

Vazby mezi moduly (třídami) a rozhraním ukazuje UML (*Unified Modeling Language*) diagram tříd na obr. 5.4.



Obrázek 5.4: Diagram tříd znázorňující provázání modulů. Metody konkrétních modulů (*eduroam*, VPN a pevná síť) jsou pro zjednodušení skryty.

5.3.2 Předání identity

Protože každý z modulů může pro blokaci vyžadovat jinou identitu (např. buď uživatelské jméno, nebo IP adresu) a hlavičky metod napříč moduly musí být dle rozhraní `IBlocker` jednotné, očekávají jednotlivé metody jako parametr instanci třídy `BlockedIdentity`.

Třída `BlockedIdentity` slouží pouze k uchování informací o identitě, která má být zablokována nebo odblokována. Identita je ve třídě reprezentována atributy, jako je:

- použitá IP adresa,
- fyzická MAC adresa spárovaná s IP adresou,
- uživatelské jméno bez názvu organizace (např. `msebela`),
- uživatelské jméno s názvem organizace (např. `msebela@ZCU.CZ`)

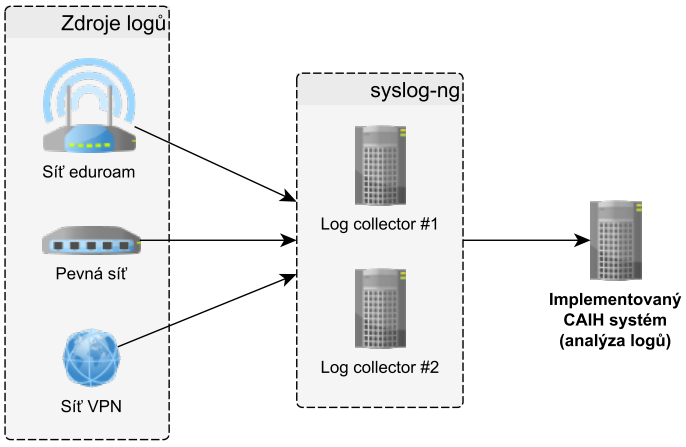
a dalšími informacemi.

K atributům třídy se programátor při implementaci modulů dostane pomocí tzv. *getter* metod, které vrátí hodnotu konkrétního atributu. Instanci jako takovou vytváří systém a v závislosti na tom, do kterého modulu spadá daná IP adresa, je identita předána konkrétnímu modulu.

5.4 Síťové logy

Protože v každém z modulů musí být implementována metoda `find(·)`, jejímž cílem je vyhledat záznamy o přidělení IP adresy v inkriminovaném datu a čase v některém ze souvisejících logů, bylo nutné zajistit přístup k uvedeným souborům.

Na ZČU jsou logy o aktivitě v univerzitní síti shromažďovány na dva konkrétní servery, přičemž jeden ze serverů je záložní. Aby nebylo nutné kvůli analýze logů přímo přistupovat na daný server, bylo využito řešení *syslog-ng*, které umožňuje konkrétní logy automaticky distribuovat i na další uvedené servery, kde mohou být následně prováděny nad logy další analýzy (viz obr. 5.5) [10].



Obrázek 5.5: Princip řešení *syslog-ng*, které umožňuje shromažďovat logy z různých zdrojů na vybraných serverech a dále je distribuovat dalším klientům (např. serveru, kde je nasazena diplomová práce).

Zvolené řešení je navíc v souladu s bezpečností, neboť v případě kompromitace serveru s aplikací z diplomové práce nedojde k ovlivnění serverů, kde jsou standardně ukládány logy.

Moduly ve vytvořené aplikaci (viz podkapitola 5.3) vyžadují k analýze dva soubory, a sice DHCP log a log se záznamy o přidělení IP adresy v síti VPN. Logy jsou na serveru s diplomovou prací ukládány pomocí *syslog-ng* do adresáře `/mnt/data`, a dále pak do podadresářů podle jednotlivých částí data, a to podle vzoru `YYYY/MM/log.YYYYMMDD`, kde `YYYY` je označení pro rok, `MM` pro měsíc a `DD` pro den, jehož se záznamy v logu týkají.

Protože jsou záznamy z logů čteny PHP funkcemi, bylo nutné změnit vlastníka a oprávnění u souborů tak, aby měl k logům přístup i uživatel `www-data`, který je určen k činnostem webového serveru. Přístup k souborům je umožněn díky změně vlastníka souborů s logy, a to úpravou výchozí konfigurace *syslog-ng* (viz obr. 5.6).

```
owner(www-data) group(www-data)
perm(0750) dir_perm(0750)
```

Obrázek 5.6: Příkazy *syslog-ng* zabezpečující změnu vlastníka a skupiny (`chown`, `chgrp`) a dále pak změnu oprávnění (`chmod`) u logů ukládaných do adresáře `/mnt/data`.

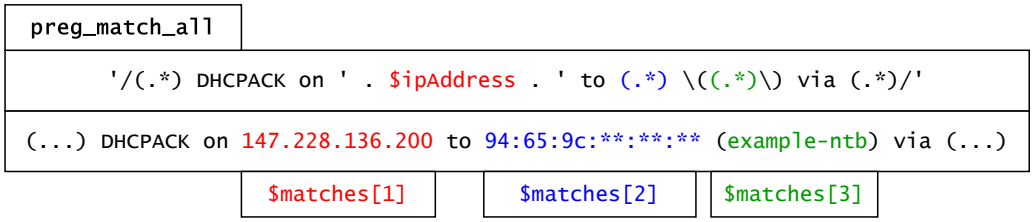
5.4.1 Vyhledávání

Samotné vyhledávání v logu je cílem metody `find(·)`, která je definovaná rozhraním `IBlocker`. U implementovaných modulů probíhá vyhledávání v následujících logách:

- *eduroam* síť – DHCP log s vazbou na RADIUS log,
- VPN síť – VPN log,
- pevná síť – DHCP log.

V každém z logů jsou vyhledávány záznamy, které potvrzují přidělení IP adresy konkrétnímu uživateli nebo zařízení v inkriminovaném datu a čase (např. v případě DHCP logu je to záznam `DHCPACK`).

Pro vyhledávání v DHCP logu a v logu pro VPN síť se používá vestavěná funkce `preg_match_all(·)` v PHP, která jako jeden z parametrů očekává regulární výraz, na základě něhož je ověřováno, zdali testovaný řetězec odpovídá výrazu. Regulární výraz lze navíc sestavit tak, aby funkce



Obrázek 5.7: Využitý způsob vyhledávání pomocí funkce `preg_match_all(·)`, která do pole `$matches` ukládá konkrétní části (IP, MAC adresu, hostname) dle regulárního výrazu uvedeného výše.

`preg_match_all(·)` vracela konkrétní části, které výrazu vyhovují – viz příklad na obr. 5.7.

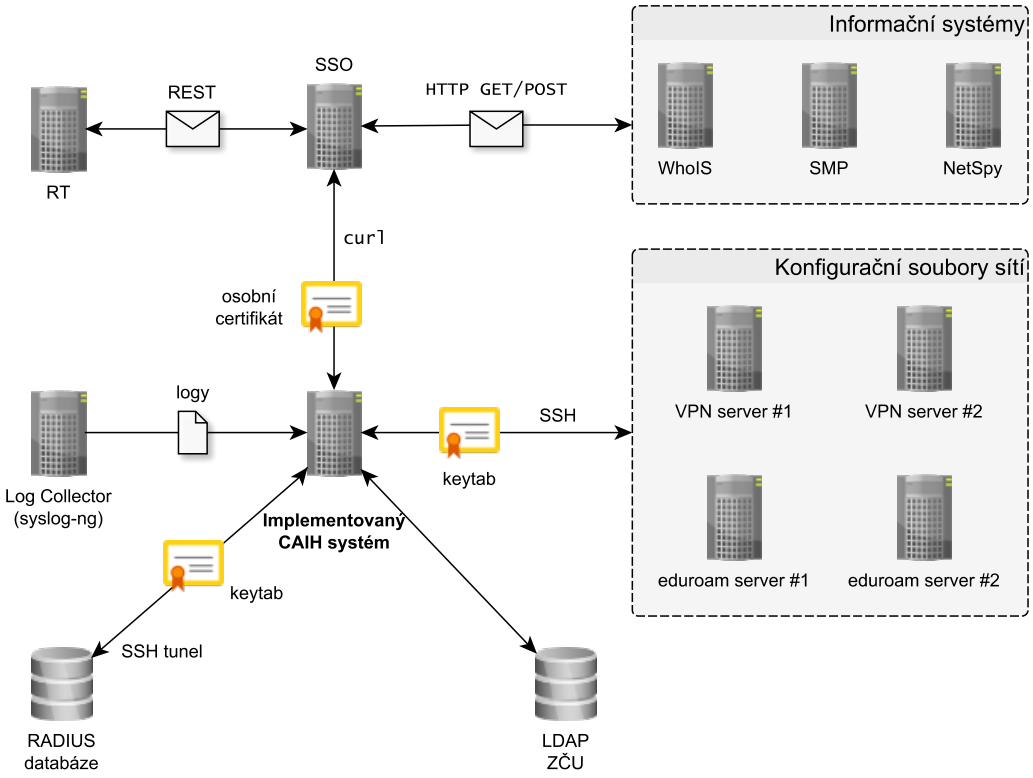
V případě sítě *eduroam* dochází po zjištění MAC adresy v DHCP logu (na základě IP adresy) k vyhledávání uživatelského jména v RADIUS logu, které v inkriminovaný čas zjištěnou MAC adresu používalo. RADIUS log má na ZČU formu databáze, vyhledávání tedy probíhá pomocí SQL (*Structured Query Language*) dotazu.

5.5 Integrace s univerzitními systémy

Vytvořená aplikace je založena především na integraci s již existujícími univerzitními systémy, přičemž každý ze systémů slouží k jinému účelu a je třeba s ním komunikovat odlišným, specifickým způsobem (viz schéma na obr. 5.8).

Pro proces řešení incidentu je vyžadována komunikace se systémem pro správu požadavků (viz podkapitola 5.5.2) a dále pak se systémem, pomocí něhož je možné zablokovat nebo odblokovat původce incidentu (problémový uživatel nebo zařízení). Všechny integrované systémy budou popsány v následujících podkapitolách.

Aby mohlo být přistupováno do dále popisovaných systémů, bylo nutné vytvořit speciální, servisní konto, které bude mít oprávnění k přístupu do daných systémů. V souvislosti s přístupy byly zároveň upraveny konfigurační soubory dotčených systémů, do nichž má být přistupováno, popř. přidány záznamy o novém, povoleném uživateli do databáze. Pro servisní konto bylo zároveň nutné vydat osobní certifikát, pomocí něhož se bude uživatel autentizovat ve službě jednotného přihlášení (viz podkapitola 5.5.1). Protože je pomocí servisního konta strojově přistupováno i na další ze serverů pomocí protokolu SSH, byl vytvořen tzv. *keytab* soubor, v němž je uchován klíč pro autentizaci. V žádném ze skriptů tak není viditelné *hard coded* heslo konta.



Obrázek 5.8: Schéma zachycující integraci s již existujícími univerzitními systémy, servery a databázemi. Schéma zobrazuje i způsob autentizace.

5.5.1 Autentizace do univerzitních systémů

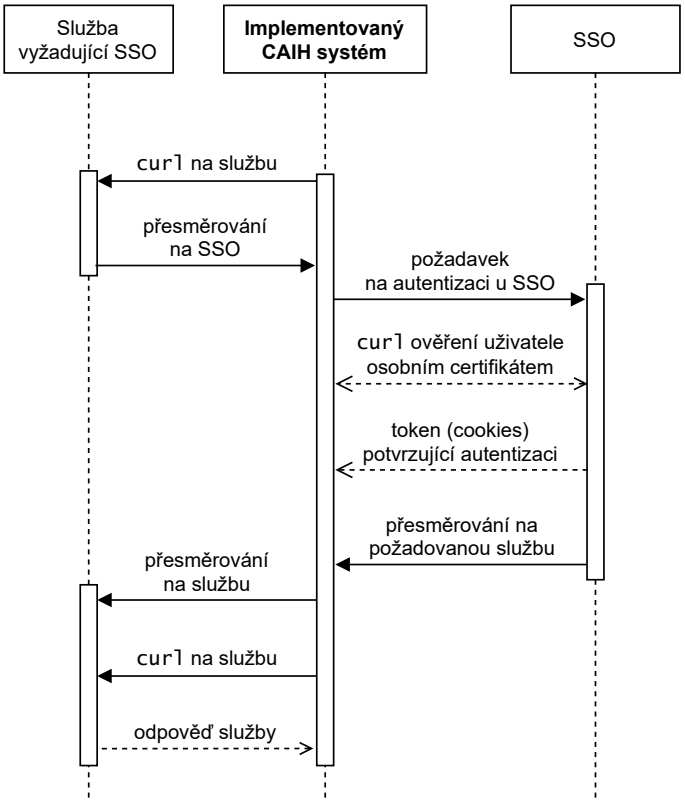
Pro přístup ke každému z informačních univerzitních systémů je nutné se autentizovat u služby jednotného přihlášení označované zkratkou SSO (*Single Sign-On*). Jedná se o způsob přihlášení (ověření identity uživatele), který uživateli zajistí přístup i do dalších systémů v organizaci bez nutnosti znovu zadávat přihlašovací údaje.

Protože jsou služby ostatních systémů strojově volány pomocí konkrétních metod ve zdrojovém kódu vytvořené aplikace, bylo nutné zvolit automatický a bezpečný způsob autentizace u služby SSO. K autentizaci byl na základě tohoto požadavku využit nástroj `curl`, pomocí něhož je možné vzdálenému serveru předat osobní certifikát uživatele a autentizovat se u služby jednotného přihlášení právě tímto způsobem.

Aby byl vybraný způsob autentizace funkční, musí SSO používané v organizaci umožňovat autentizaci pomocí osobního certifikátu. Osobní certifikát musí být zároveň vydán tak, aby bylo možné se pomocí něho prokazovat u služby SSO.

Protože služba SSO při úspěšném ověření uživatele vytváří *cookies* (do-

časné soubory v paměti webového prohlížeče) obsahující informace o úspěšné autentizaci (*token*) a zároveň provádí přesměrování na původně požadovanou službu, je nutné, aby byl nástroj `curl` nakonfigurován tak, aby povoloval přesměrování a zároveň umožňoval ukládání *cookies* z externích serverů. V opačném případě by sice došlo k úspěšné autentizaci, z důvodu neuložení *cookies* by ovšem nedošlo k přesměrování na původně požadovanou službu.



Obrázek 5.9: Diagram zobrazující průběh komunikace mezi implementovaným CAIH systémem a mezi dalšími univerzitními systémy pomocí nástroje `curl`, které předchází autentizace u služby SSO.

Kroky umožňující vstup do dalších systémů zabezpečených službou SSO jsou tedy následující:

1. Volání služby pomocí nástroje `curl` uvedením konkrétní URL adresy, kam je nutné přistoupit. Pokud jsou součástí URL adresy i další data (například GET nebo POST požadavek s konkrétními argumenty), je nutné data zakódovat např. pomocí funkce `http_build_query(·)`.

2. Služba SSO přístup na URL adresu nepovolí a požaduje provedení autentizace uživatele.

3. Pomocí nástroje `curl` je SSO předán osobní certifikát uživatele, který má do systému chráněného pomocí SSO přístup.

4. Služba SSO osobní certifikát uživatele ověří a v případě úspěchu vrátí autentizační token, uloží do *cookies* informaci o úspěšném ověření uživatele a provede přesměrování na původně požadovanou službu.

Návaznost jednotlivých kroků ukazuje i diagram na obr. 5.9.

5.5.2 Systém pro správu požadavků

Systém pro správu požadavků slouží v případě řešení bezpečnostního incidentu k evidenci komunikace se stěžovatelem a s původcem incidentu, a to formou jednotlivých lístků (viz podkapitola 2.4). Bezpečnostní tým WIRT na ZČU pro evidenci komunikace využívá systém RT (*Request Tracker*).

Cílem implementovaného CAIH systému je integrací se systémem RT automaticky provádět následující operace:

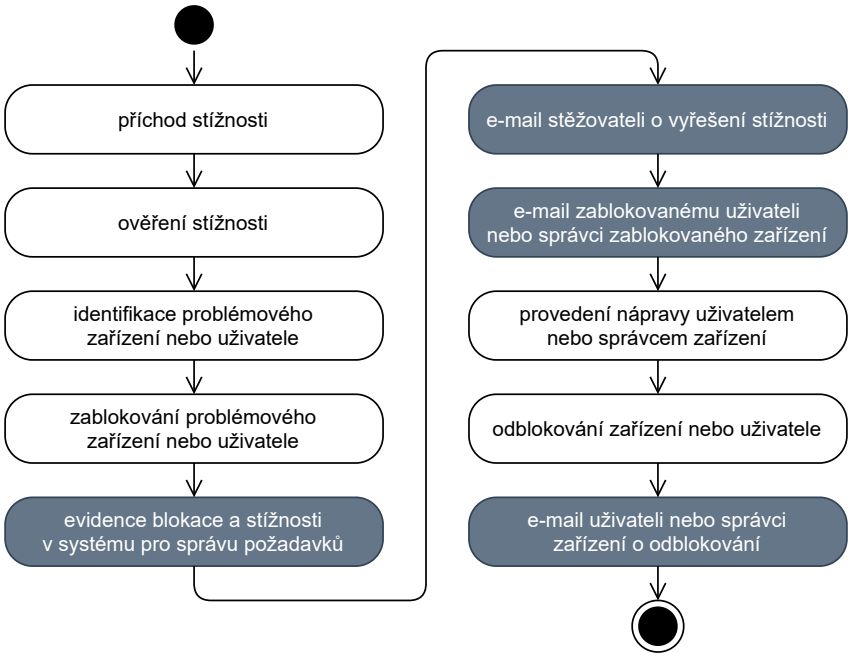
- založit lístek pro problémového uživatele (popř. pro lokálního správce) s upozorněním o provedené blokaci,
- odpovědět stěžovateli ohledně vyřešení incidentu,
- upozornit uživatele (lokálního správce) na odblokování,
- získat metadata o konkrétním lístku,
- přidat vazbu k existujícímu lístku (vytvoření vazby mezi lístkem stěžovatele a lístkem původce incidentu),
- uzavřít existující lístek (v souvislosti s vyřešením incidentu).

Na obr. 5.10 s diagramem zobrazujícím kroky řešení bezpečnostního incidentu se typicky jedná o čtyři zvýrazněné kroky. V případě nahlášení incidentu stěžovatelem a zasláním stížnosti do RT systému lze uvažovat i první krok, a sice *příchod stížnosti*.

Komunikace

Systém RT je možné ovládat pomocí rozhraní REST (*Representational State Transfer*) [9]. Rozhraní REST umožňuje komunikovat s databází RT systému pomocí zpráv zapouzdřených v protokolu HTTP [9].

V implementovaném CAIH systému byla vytvořena třída `Ticketing System`, která poskytuje metody pro volání konkrétních REST endpointů systému RT.



Obrázek 5.10: Zvýrazněné kroky jsou řešeny díky integraci v systému pro správu požadavků.

5.5.3 Systém evidence uživatelů

Na ZČU existuje vlastní software evidence identifikačních údajů o uživatelích označovaný názvem *WhoIS* [17]. Systém agreguje data z několika dalších univerzitních systémů a umožňuje dostatečně oprávněnému uživateli získat informace o konkrétní osobě na univerzitě.

Systém je využíván například pracovníky uživatelské podpory, kteří jsou schopni při dotazu ze strany uživatele provést jeho identifikaci a případně zjistit další informace (např. při ztrátě univerzitního průkazu či při žádosti související se zapomenutím nebo vypršením hesla do univerzitních systémů).

U každého konta systém udržuje tzv. *deník*, v němž jsou evidovány informace o nahlédnutí na profil osoby, o změně hesla ze strany uživatelské podpory nebo o dalších podobných operacích. Každá činnost včetně náhledu je tak v systému zaznamenávána.

Cílem implementovaného CAIH systému je do aplikace *WhoIS* zapisovat informace o provedené blokaci (případně o odblokování) u dotčené osoby. Pokud by tak zablokovaný uživatel kontaktoval uživatelskou podporu s dotazem ohledně nefunkčnosti svého připojení do univerzitní sítě (resp. do sítě *eduroam* či VPN), pracovník mu může sdělit informace o blokaci a její konkrétní důvod.

Komunikace

Systém *WhoIS* neobsahuje žádné standardizované rozhraní pro komunikaci s vnějším světem (resp. jinými systémy). Zároveň je v systému ovšem zavedena možnost vložení nového záznamu do *deníku* při volání konkrétní URL adresy. Předpokladem je, aby uživatel, který URL adresu s konkrétními parametry zavolá, měl do systému skutečně přístup, a to jak ze strany SSO, tak ze strany webového serveru, kde je uveden konečný počet oprávněných uživatelů.

5.5.4 Bezdrátová síť *eduroam*

Pro správu bezdrátové sítě *eduroam* existuje na ZČU nástroj SMP (*Správa mobilního připojení*), který umožňuje uživatelům ZČU aktivovat a dále spravovat přístup do uvedené sítě [1]. Dostatečně oprávněným uživatelům je umožněno provádět i blokaci a odblokování konkrétních uživatelů sítě, a to pomocí připraveného grafického rozhraní [1].

Cílem implementovaného CAIH systému je využívat aplikaci SMP právě k zablokování a případnému odblokování konkrétních uživatelů v síti *eduroam*.

Komunikace

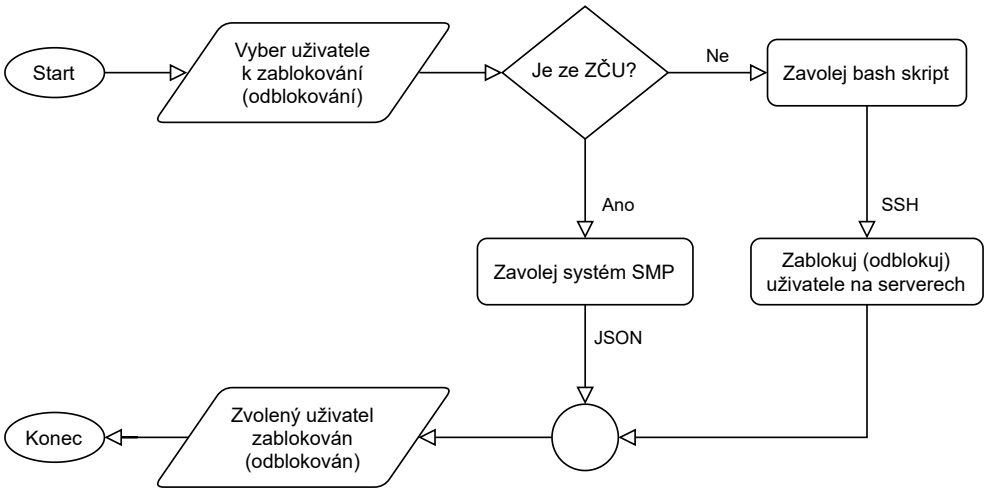
Systém SMP umožňuje na konkrétní URL adrese odesílat pomocí HTTP POST předem připravený formulář, který způsobí zablokování či odblokování konkrétního uživatele. Předpokladem je, aby uživatel, který URL adresu s konkrétními parametry zavolá, měl do systému SMP přístup, a to jak ze strany SSO, tak ze strany webového serveru, kde je uveden konečný počet oprávněných uživatelů. Odpověď systému SMP je ve formě JSON (*JavaScript Object Notation*).

Uživatelé z jiných organizací

Protože k přístupovým bodům sítě *eduroam* mohou přistupovat i uživatelé z jiných, zapojených organizací a systém SMP umožňuje blokovat aktivitu pouze u uživatelů z organizace ZČU, bylo zapotřebí u uživatelů z jiných organizací zajistit alternativní způsob blokace.

Na ZČU existují dva servery, na nichž jsou uloženy konfigurační soubory sítě *eduroam*. Konfigurační soubory umožňují specifikovat konkrétní identitu, které nebude umožněno se do sítě *eduroam* přihlásit. Soubory jsou automaticky upravovány pomocí *bash* skriptu vytvořeného v rámci diplomové práce, který se vzdáleně připojuje na dané servery přes SSH (viz podkapitola 5.5.7).

V případě blokace uživatelů ze ZČU je tedy volán již zmíněný systém SMP, v případě uživatelů z jiných organizací dochází k blokaci editací konfiguračního souboru na konkrétních serverech (viz diagram na obr. 5.11).



Obrázek 5.11: Vývojový diagram zobrazující část klíčového procesu blokace (nebo odblokování) u uživatelů sítě *eduroam*.

5.5.5 Síť VPN

Na ZČU nebyl v době psaní diplomové práce nasazen žádný webový nástroj, který by umožňoval provést zablokování nebo odblokování konkrétního uživatele v síti VPN. K provedení operace je tak nutné zasáhnout přímo do konfiguračních souborů sítě VPN, kde je možné specifikovat blokované uživatele.

Blokaci nebo případné odblokování řeší vlastní *bash* skript, který je dále popsán v podkapitole 5.5.7.

5.5.6 Pevná síť

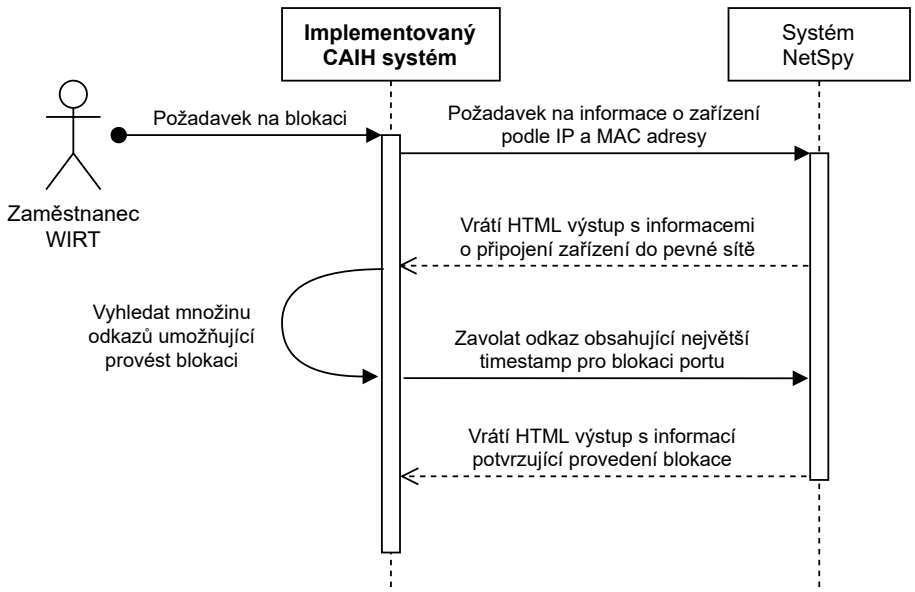
Zařízení připojená do pevné sítě lze na ZČU spravovat pomocí vlastního systému s názvem *NetSpy*. Systém umožňuje deaktivovat port na aktivním prvku sítě, a znemožnit tak problémovému zařízení se připojit do pevné sítě [17].

Cílem vytvářeného CAIH systému je využívat systém *NetSpy* k blokování a případnému odblokování problémových zařízení, které jsou připojeny do pevné sítě.

Komunikace

Systém *NetSpy* umožňuje voláním konkrétní URL adresy umístit nebo odebrat zařízení z tzv. karanténní podsítě [17]. Ověřování probíhá přes SSO, dále pak ověřením dle výčtu povolených uživatelů v konfiguraci webového serveru a podle použité IP adresy. Odpověď systému *NetSpy* je zasílána ve formě XML (*Extensible Markup Language*) [17]. Karanténní síť již ovšem není na ZČU aktivní a systém *NetSpy* zároveň neumožňuje vzdáleně provádět blokaci portu, která je vytvářeným CAIH systémem požadována.

Protože systém *NetSpy* již není nadále vyvíjen a zásahy do systému by byly ekonomicky nevýhodné a ani nebyly předmětem diplomové práce, bylo nutné najít alternativní, dočasný způsob řešení. Systém *NetSpy* bude navíc v nejbližších měsících nahrazen novým systémem umožňující provádět správu zařízení přes autentizované API (*Application Programming Interface*).



Obrázek 5.12: Diagram popisující dočasný způsob řešení, kterým je prováděno blokování zařízení připojeného do pevné sítě. Pro zjednodušení diagram neobsahuje počáteční krok autentizace ve službě SSO.

Dočasně implementované řešení spočívá ve strojovém ovládání systému *NetSpy*, resp. otevření URL adresy s přehledem připojení konkrétního zařízení do pevné sítě. Pro otevření přehledu je nicméně nutné znát kromě IP adresy i MAC adresu zařízení, která je získána v kroku předcházející *blokaci*, a to během *identifikace* (z DHCP logu). K otevření a získání obsahu webové stránky se využívá nástroj `curl` (viz podkapitola 5.5.1). V získaném

přehledu připojení (resp. z HTML výstupu) je následně nutné nalézt odkaz obsahující největší časové razítko (tj. poslední zaznamenané datum připojení do pevné sítě), kterým je možné činnost zařízení zablokovat (případně odblokovat). Odkaz pro blokaci, resp. pro odblokování je získán pomocí regulárního výrazu. Postup je znázorněn i na diagramu na obr. 5.12.

5.5.7 Blokace *bash* skriptem

Pro některé případy neexistuje na ZČU webový nástroj, který by umožnil provádět zablokování nebo odblokování voláním konkrétní webové služby či URL adresy. Tímto způsobem není možné (od)blokovat:

- uživatele sítě *eduroam* připojené kontem jiným než ze ZČU (např. jiná VŠ),
- uživatele sítě VPN.

Pro uvedené případy byl pro potřeby diplomové práce vytvořen *bash* skript, který umožňuje automaticky provést úpravu patřičných konfiguračních souborů. Konfigurační soubory jsou na serverech pro správu sítí *eduroam* a VPN, přičemž úprava spočívá v přidání (případně odebrání) řádku s uvedenou, zablokovanou (odblokovanou) identitou.

Před připojením k serveru je nutné provést autentizaci pomocí klíče uloženého v *keytab* souboru (viz kapitola 5.5), následně je možné se ke vzdálenému serveru připojit přes SSH. Předpokladem pro úspěšné připojení pomocí SSH je vytvoření pravidla ve firewallu cílového serveru.

Vytvořený *bash* skript jako vstup očekává tři povinné argumenty, a to v následujícím pořadí:

1. *typ sítě* – `eduroam` nebo `vpn`,
2. *typ akce* – `block` (blokace) nebo `unblock` (odblokování),
3. *identita* – uživatel, u něhož má být provedena daná akce.

Například zablokování uživatele `msebela` v síti VPN lze provést následujícím způsobem: `./remoteServers.sh vpn block msebela` (za předpokladu, že jsou skriptu nastavena práva pro spuštění).

V implementovaném CAIH systému je *bash* skript volán pomocí vestavěné funkce `shell_exec(·)` v PHP. Skriptu se jako vstup předává jméno uživatele identifikovaného CAIH systémem na základě logů, které se má zablokovat (nebo odblokovat). Tento vstup je validován na straně PHP:

1. obalením vstupu a ošetřením speciálních znaků, které by mohly mít v operačním systému *Linux* i jiný význam,

2. ověřením, zdali předaná identita vyhovuje množině povolených znaků, které se očekávají v uživatelském jméně.

Další ošetření provádí i samotný *bash* skript, který předanou identitu opět ošetří od potenciálně nebezpečných znaků.

Po validaci, ošetření vstupu a po úspěšném připojení se ke vzdálenému serveru, dochází k vyhledání konkrétního řetězce (viz obr. 5.13) v konfiguračním souboru. V případě blokace dojde v uvedeném řetězci k nahrazení `#orionlogin` za zablokovanou identitu, v případě odblokování pak ke smazání řádku, kde je místo výrazu `#orionlogin` vyplněna identita uživatele.

```
#orionlogin Auth-Type = Reject # Blockator, neupravovat
```

Obrázek 5.13: Řetězec, který se vyhledává v konfiguračních souborech.

Jednotlivé kroky prováděné *bash* skriptem při blokaci uživatele jsou tedy následující:

1. ošetření vstupu (předané identity),

2. autentizace u služby *Kerberos*,

3. připojení se přes SSH ke vzdálenému serveru,

4. vytvoření záložní kopie konfiguračního souboru,

5. vyhledání řetězce z obr. 5.13 v konfiguračním souboru,

6. nahrazení `#orionlogin` za skutečnou identitu ve vyhledávaném řetězci,

7. vložení nezměněného řetězce z obr. 5.13 do konfiguračního souboru (pro budoucí blokace),

8. uložení změn v konfiguračním souboru,

9. aplikace změn restartováním služby.

5.6 Proces řešení incidentu

V implementovaném CAIH systému proces řešení incidentu sestává z pěti, resp. čtyř následujících kroků:

1. *založení incidentu*,
2. *identifikace* původce incidentu,
3. *blokace*,
4. *upozornění stěžovatele* (uvažováno pouze tehdy, pokud se incident řeší jako někým nahlášená stížnost v systému pro správu požadavků),
5. *odblokování*.

Činnost v krocích odpovídá již prezentovanému diagramu na obr. 5.1.

Během prvního kroku dojde na základě IP adresy k výběru vhodného modulu (resp. na základě výstupu volání metody `isFromThisNetwork(·)`), který umožňuje provádět i následující kroky – tj. nalézt záznamy o IP adrese v relevantním logu (*identifikace*) a provést *blokaci* a *odblokování* (viz podkapitola 5.3). Obsluha modulů je předmětem činnosti třídy `IncidentModel`.

Každý incident je třeba řešit krok po kroku bez možnosti některé z kroků úmyslně přeskočit. Při úspěšném dokončení některého z kroků je uživatel automaticky přesměrován na následující krok. Datum a čas úspěšného dokončení kroku je evidován v databázi. Po potvrzení *blokace* (třetí krok) již nelze upravovat údaje v předchozích krocích.

5.6.1 Založení incidentu

Založení incidentu spočívá v získání základních údajů o incidentu, a to konkrétně o IP adrese z rozsahu ZČU (`147.228.0.0/16`), která se incidentu účastnila a dále pak o datu a čase, během něhož k incidentu došlo.

U zadané IP adresy dojde pomocí PHP funkce `gethostbyaddr(·)` ke zjištění hostname, které na ZČU obsahuje informaci o tom, z jakého typu sítě IP adresa je (např. IP adresa `147.228.137.253` má hostname `eduroam-137-253.zcu.cz`, jde tedy o IP adresu ze sítě *eduroam*). Případně je možné IP adresu převést pomocí funkce `ip2long(·)` do datového typu `long integer` a následně pomocí porovnávacích operátorů zjistit, zdali se nachází ve stanoveném IP rozsahu. Díky tomuto postupu je možné vybrat vhodný modul, který dokáže s IP adresou dále pracovat.

V případě, že uživatelem vyplněný datum a čas vzniku incidentu obsahuje záznam o časové zóně (například u stížností ze zahraničí), bude uvedený

čas automaticky převeden do středoevropského času (SEČ), ve kterém jsou zároveň vedeny i záznamy v logách. Časovou zónu, do které má být čas převeden, je případně možné změnit v konfiguračním souboru `config.php`.

Třetím povinným vstupem je typ incidentu. Uživatel vybírá z množiny typů incidentů (systém umožňuje přidat i další typy), přičemž zvolený typ slouží ke kategorizaci incidentu v souhrnném přehledu, v grafech a především jsou na základě zvoleného typu incidentu importovány předpřipravené šablony odpovědí používané v dalších krocích.

Nepovinně může být vyplněna fyzická MAC adresa zařízení, přičemž při jejím vyplnění budou vyhledány záznamy, které obsahují jak uživatelem vyplněnou IP adresu, tak zároveň vyplněnou MAC adresu.

Pokud uživatel vyplní číslo lístku se stížností ze systému pro správu požadavků, bude automaticky do procesu řešení incidentu zahrnut čtvrtý krok (celkem jich tedy bude pět), a sice *upozornění stěžovatele* (viz podkapitola 5.6.4).

Potvrzením kroku dojde na základě IP adresy (resp. hostname) k vybrání vhodného modulu a přechodu k *identifikaci*.

5.6.2 Identifikace původce incidentu

Cílem *identifikace* je poskytnout uživateli detailní informace o tom, komu byla zadaná IP adresa v inkriminovaný čas přidělena.

Modul, vybraný v předchozím kroku, pomocí metody `find(·)` prohledá určené logy (typicky DHCP log) a vrátí informace o:

- *uživateli, který měl přidělenou zadanou IP adresu* (popř. o uživatelích, pokud jich bylo v okolí zadaného času více),
- případně *použitou MAC adresu*,
- případně *hostname zařízení*.

Zároveň jsou členovi bezpečnostního týmu vráceny i konkrétní záznamy z logu, které jsou k danému incidentu relevantní a na základě nichž systém uživatele identifikoval. Je tak možné manuálně zkontrolovat, zdali systém identifikoval uživatele správně. Všechny podstatné identity (uživatelské jméno, IP, MAC adresa, hostname) vyskytující se v logu, jsou pro větší přehlednost barevně odlišeny od dalších údajů.

V případě, že byla IP adresa přidělena zařízení, které je připojeno do pevné sítě, je automaticky dohledán lokální správce, který může zařízení na dané fakultě, katedře nebo oddělení odpojit a zabezpečit. Seznam lokálních správců je evidován v databázi vytvořeného CAIH systému, přičemž

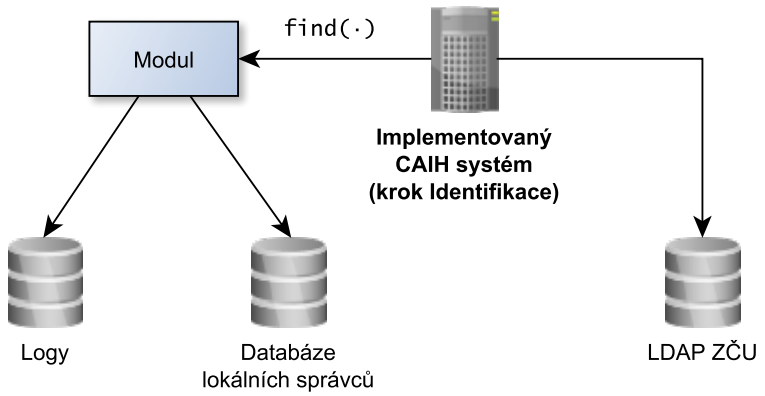
uvedený seznam je možné přímo v systému i spravovat. Vazba mezi kompetentním lokálním správcem a zařízením připojeným do pevné sítě je zjištěna na základě hostname zařízení, které typicky obsahuje zkratku názvu pracoviště. Protože historicky nejsou názvy hostname zařízení na ZČU jednotné, je dohledání lokálního správce řešeno několika regulárními výrazy. Systém zároveň umožňuje k jednomu zařízení dohledat více lokálních správců.

U každého identifikovaného uživatele (případně lokálního správce) je z LDAP (*Lightweight Directory Access Protocol*) dodatečně získáno jméno a příjmení a zároveň zobrazen přímý odkaz na detaily o uživateli do interního systému *WhoIS*.

Pokud se systému nepodaří identifikovat žádnou osobu, která by mohla mít zařízení ve správě, vrací systém jako výchozí hodnotu `abuse@zcu.cz`. Tím je zaručeno, že bude možné vytvořit lístek o incidentu v systému pro správu požadavků.

Na základě vyhledaných informací zaměstnanec bezpečnostního týmu potvrzuje blokaci uživatele identifikovaného CAIH systémem a volí jazyk, pomocí něhož bude s identifikovaným uživatelem komunikováno (a případně i se stěžovatelem, pokud bylo v prvním kroku vyplněno číslo lístku se stížností). Podle zvoleného jazyka je z databáze importována konkrétní jazyková mutace šablony s odpovědí pro stěžovatele a původce incidentu. Systém umožňuje vybrat i více jazykových mutací najednou, a to například pro incidenty, které se týkají zahraničních studentů.

Provázání se soubory a databázemi požadovanými během *identifikace* ukazuje diagram na obr. 5.14.



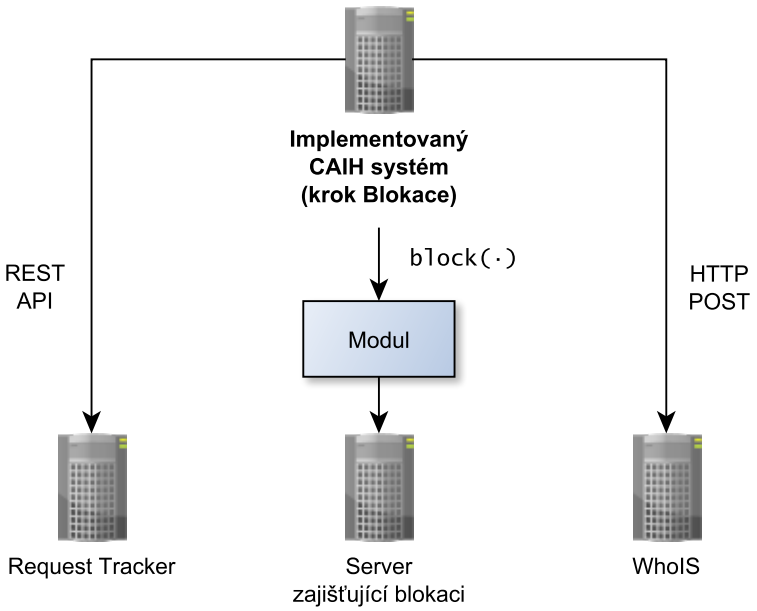
Obrázek 5.14: Soubory a databáze, které jsou využívány během identifikace.

5.6.3 Blokace

Během *blokace* dojde u vybraného modulu k volání metody `block(·)`, která v systému, jenž je pro proces blokování určen, provede blokaci konkrétního uživatele nebo zařízení (viz obr. 5.15).

Zablokováním zároveň dochází k vytvoření nového lístku v systému pro správu požadavků (systém *Request Tracker*). Lístek obsahuje informace o důvodu blokace a je určen pro zablokovaného uživatele (tj. původce incidentu), popř. pro lokálního správce. Obsah zprávy je založen na šablonách odpovědí pro e-maily načtených z databáze, které jsou rovněž spravovány ve vytvořeném CAIH systému. Protože šablony mohou být personalizovány, dochází před jejich načtením k nahrazení parametrů za skutečné údaje (např. `%incident_user%` za jméno uživatele, `%incident_network%` za název sítě, v níž k incidentu došlo apod.).

Při úspěšné blokaci také dochází ke vložení informativního záznamu do interního systému *WhoIS* určeného k evidenci uživatelů, aby měli případně pracovníci uživatelské podpory informaci o provedené blokaci (viz podkapitola 5.5.3).



Obrázek 5.15: Systémy, se kterými komunikuje implementovaný CAIH systém během kroku blokace.

5.6.4 Upozornění stěžovatele

Jedná se o krok, který je uživateli zobrazen pouze tehdy, pokud bylo při založení incidentu (tj. první krok) vyplněno číslo lístku se stížností.

Smyslem kroku *upozornění stěžovatele* je automaticky odpovědět na původní stížnost, a to opět díky předpřipraveným šablonám s odpověďmi. Členovi bezpečnostního týmu je tak zobrazen připravený text zprávy pro stěžovatele, a to podle zvolené jazykové mutace. Jediným úkolem řešitele incidentu je odeslání zprávy potvrdit (případně dodatečně upravit text zprávy).

Potvrzením dojde k odeslání připravené zprávy do systému pro správu požadavků, k automatickému uzavření (vyřešení) lístku se stížností a k přechodu na poslední krok – k *odblokování*.

5.6.5 Odblokování

Během *odblokování* je nad vybraným modulem volána metoda `unblock(·)`, jejímž cílem je odblokovat konkrétní identitu (problémového uživatele či zařízení) v systému, který je pro odblokování v dané části sítě určen.

Součástí kroku je i odeslání zprávy pomocí systému pro správu požadavků. Zpráva obsahuje informaci o odblokování a je určena původně zablokovanému uživateli (tj. původci incidentu), případně lokálnímu správci. Zpráva je personalizována a odeslána ve vybraných jazykových mutacích.

Podobně jako v kroku *blokace* dochází i během *odblokování* ke vložení záznamu o odblokování do interního systému *WhoIS* pro pracovníky uživatelské podpory.

Úspěšným odblokováním, odesláním zprávy původně zablokovanému uživateli a vložením záznamu do systému *WhoIS* je celý incident uzavřen a ve vytvořeném CAIH systému označen stavem *vyřešeno*.

6 Struktura databáze

Implementovaný CAIH systém využívá k ukládání dat databázi navrženou v databázovém systému *MySQL* (*MariaDB*), přičemž všechny tabulky používají úložiště *InnoDB*. Kódování u každé z tabulek odpovídá `utf8_czech_ci`.

Systém se do databáze připojuje pomocí rozhraní PDO (*PHP Data Objects*), SQL dotazy jsou skládány pomocí *prepared statements*. Parametry pro připojení k databázi lze nastavit v konfiguračním souboru `config.php`.

6.1 Databázové tabulky

V databázi je vytvořeno celkem devět databázových tabulek. Všechny tabulky mají ve svém názvu prefix `blck_` (několik písmen z názvu implementovaného CAIH systému pojmenovaného *Blockator*) pro případ, že by se tabulky nalézaly ve stejné databázi s dalšími tabulkami, ovšem z jiných aplikací.

V každé z databázových tabulek je vytvořen primární klíč (tzv. *primary key* označovaný též zkratkou PK), který umožňuje vytvořit vazbu (pomocí cizího klíče, tzv. *foreign key*, zkratka FK) se záznamy v ostatních databázových tabulkách (viz jejich použití na diagramu na obr. 6.1).

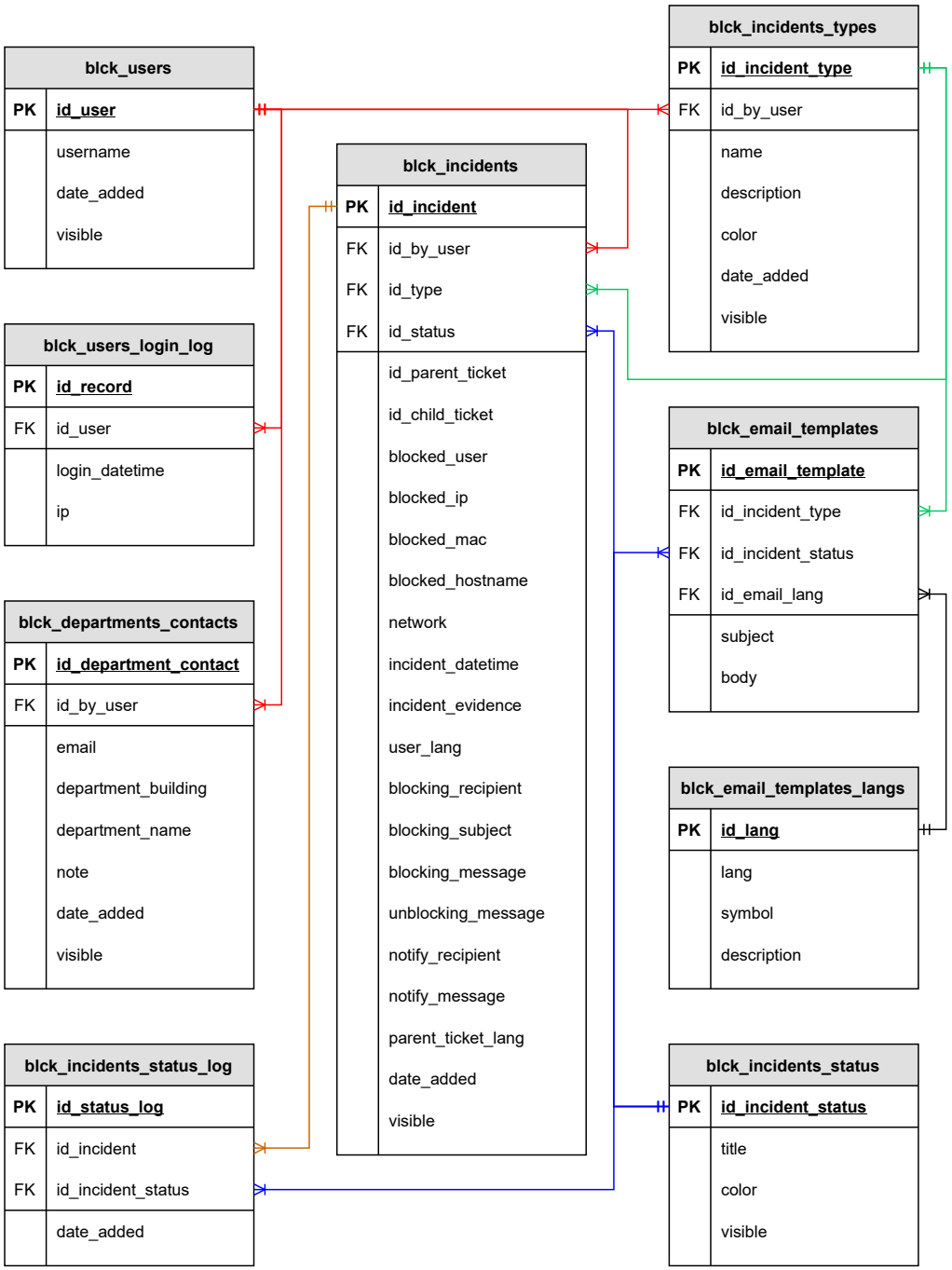
Použité datové typy u jednotlivých atributů byly voleny tak, aby byl zvolený rozsah datového typu efektivně využit, ale aby zároveň poskytoval dostatečnou datovou rezervu (u atributů, které to vyžadují). Například atributy, které nabývají jen číselných hodnot (`TINYINT` u atributu `visible`, `SMALLINT` u primárních klíčů apod.) a u nichž se neočekávají záporné hodnoty, jsou tak vedeny jako `UNSIGNED` (neznaménkové datové typy).

Některé z tabulek obsahují atribut s názvem `visible`, který slouží k deaktivaci konkrétního záznamu ve vytvořeném CAIH systému. Atribut `visible` nabývá buď hodnoty `1` (tj. záznam je viditelný, výchozí hodnota), nebo `0` (záznam je deaktivován a není v systému viditelný). Na základě těchto dvou hodnot, kterých atribut `visible` nabývá, je definován jako `UNSIGNED TINYINT(1)`.

V několika databázových tabulkách je také mezi atributy obsažen cizí klíč `id_by_user`, který slouží k uchování informace o tom, jaký z uživatelů (resp. členů bezpečnostního týmu) daný záznam do databáze vložil.

6.2 ERA diagram

Vazby mezi devíti databázovými tabulkami ukazuje ERA diagram na obr. 6.1. V diagramu jsou použity zkratky označující použití databázového klíče – PK (*primary key*) a FK (*foreign key*).



Obrázek 6.1: ERA diagram navržené databáze.

7 Grafické rozhraní

Grafické rozhraní označované zkratkou GUI (*Graphical User Interface*) vytvořené webové aplikace je založeno na volně dostupném *front end* frameworku *Bootstrap*, a to konkrétně na poslední verzi 5.0 (nejnovější verze v době psaní diplomové práce).

GUI využívá značkovacího jazyka HTML5 (*HyperText Markup Language*) a CSS3 (*Cascading Style Sheets*). Zdrojové kódy jsou z pohledu online validátoru[1] konsorcia W3C (*World Wide Web Consortium*) validní.

Použitý framework *Bootstrap* obsahuje předpřipravené CSS třídy společně se základními a hotovými komponentami (např. hlavička, menu, elementy formuláře), které jsou často vývojáři při vývoji každé webové aplikace požadovány. *Bootstrap* zároveň poskytuje responzivní CSS třídy pro základní responzivnost (správné zobrazení obsahu stránky na zařízeních s různým rozlišením) vytvořené aplikace.

Při vývoji aplikace bylo cílem navrhnout a sestavit komponenty tím způsobem, aby každá stránka byla sama o sobě vypovídající a nebylo nutné při běžném používání aplikace nahlížet do uživatelské příručky. Komponenty, u nichž by si nemusel být uživatel jistý účelem (např. vstupní pole formuláře), je vždy doplněna stručná nápověda.

7.1 Vzhled

Vzhled aplikace je založen na šabloně *Dashboard*, která je k dispozici na oficiálních stránkách[2] frameworku *Bootstrap*. Šablona *Dashboard* byla dále upravována pro potřeby aplikace, a to především po stránce responzivnosti postranního menu. Stejnou šablonu (resp. verzi z roku 2019) používá i již dříve vyvinutá aplikace *Phishingator*, která je bezpečnostním týmem rovněž využívána. Použitím stejné šablony je tak zaručen stejný způsob ovládání i jednotnost grafického rozhraní u používaných systémů.

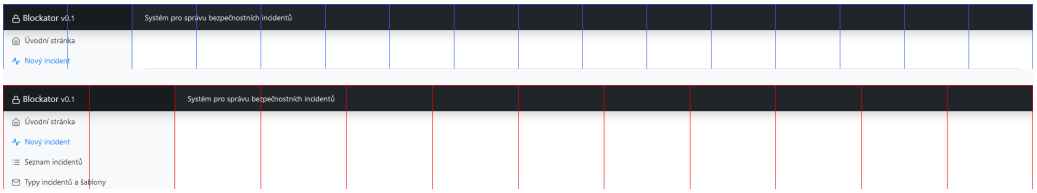
Framework *Bootstrap* byl zároveň zkompilován v odlišném nastavení, než ve kterém je standardně k dispozici. Úprava spočívá ve změně počtu sloupců, na něž je stránka logicky rozdělena (tzv. *grid system* [13]). Zatímco původní nastavení frameworku *Bootstrap* definuje rozdělení na 12 sloupců [13] (tzn. šířka webové stránky je rozdělena na 12 stejně širokých sloupců, jejichž

[1] Validátor HTML – `https://validator.w3.org`

[2] Šablony využívající framework *Bootstrap* – `https://getbootstrap.com/docs/5.0/examples/`

šířka se automaticky přizpůsobuje šířce okna webového prohlížeče), v případě vytvořené aplikace byl framework *Bootstrap* zkompilován s počtem sloupců nastaveným na hodnotu 16.

Změna počtu sloupců na vyšší číslo umožnila lépe pracovat se šířkou stránky tak, aby levé, postranní menu nezasahovalo na webové stránce do prostoru, který by mohl být určen pro obsah formuláře (viz obr. 7.1 s různým počtem sloupců vyobrazeným na GUI, resp. šabloně *Dashboard*, vytvořené aplikace).



Obrázek 7.1: Rozdíl při použití 16 sloupců (horní část obrázku) a 12 sloupců (dolní část obrázku) rozdělující stránku při šířce 1920 pixelů. Použití 16 sloupců vyhovuje velikosti položek v postranním menu, a zároveň tak menu zbytečně nezasahuje do prostoru, který může být určen pro obsah stránky.

Vytvořená aplikace je také díky použití frameworku *Bootstrap* a předpřipraveným responzivním CSS třídám plně responzivní, tj. přizpůsobena zařízením s různým rozlišením a orientací obrazovky.

Díky použití frameworku *Bootstrap* bylo také docíleno a otestováno, že je vytvořená aplikace funkční a správně zobrazena ve všech moderních webových prohlížečích. U vlastních, nově definových CSS tříd byly případně doplněny tzv. *CSS prefixy* (např. prefix `-webkit-` pro starší webové prohlížeče s vykreslovacím jádrem *WebKit*), které zabezpečující správné aplikování CSS vlastnosti i u starších webových prohlížečů.

8 Zabezpečení aplikace

Protože implementovaný systém umožňuje získávat citlivé informace a provádět operace, ke kterým by bylo nutné v síti ZČU získat nejvyšší oprávnění, bylo požadováno, aby byla aplikace řádně zabezpečena.

Samozřejmostí je nasazení důvěryhodného HTTPS certifikátu a přesměrování všech požadavků právě na zabezpečený protokol HTTPS. Přesměrování požadavků bylo docíleno nastavením pravidel v souboru `.htaccess` a díky HTTPS hlavičce HSTS (*HTTP Strict Transport Security*).

V adresáři `/.well-known` je zároveň dle podkapitoly 2.3 připraven soubor `security.txt` s informacemi ke kontaktování bezpečnostního týmu.

Vstup do systému je zabezpečen díky implementaci modulu služby jednotného přihlášení (SSO) *Shibboleth* používané na ZČU. Služba umožňuje v konfiguraci webového serveru definovat seznam uživatelů, kteří budou mít do aplikace přístup (například pouze zaměstnanci bezpečnostního týmu).

Pro vstup do systému je rovněž nutné přistupovat z administrátorského IP rozsahu (resp. z určité podsítě na ZČU). Pokud tak uživatel nemá přidělenou IP adresu z uvedeného rozsahu, jsou jeho požadavky serverem automaticky zahazovány.

8.1 Validace uživatelského vstupu

Veškerý vstup je validován jak na straně uživatele, tak v serverové části aplikace.

Pro validaci vstupu na straně uživatele poskytuje značkovací jazyk HTML u vstupních polí možnost definovat, jaká data jsou ve vstupním poli očekávána (např. `type="email"` pro kontrolu zadané e-mailové adresy). Zároveň je možné pomocí atributu `required` určit, zdali musí být vstupní pole před odesláním formuláře vyplněno. Vytvořený systém navíc ke vstupním polím přidává parametr `maxlength` určující maximální počet znaků, který je založen na maximálním možném počtu znaků daného atributu v databázi. Po odeslání formuláře jsou všechna zadaná data znovu validována stejným způsobem v serverové části aplikace, a to voláním metody `validateData()`.

Na straně serveru jsou kromě již uvedených kontrol prováděna i další ověření. Například je ověřováno, zdali e-mail zadaný uživatelem skutečně na ZČU existuje (na zakladě dat v LDAP) nebo zdali není cílem podstrčit aplikaci záznam, který neexistuje nebo byl již smazán, a to například změnou

parametrů v URL adrese nebo podstrčením vlastní hodnoty v HTML tagu `<option>` ve výběrovém seznamu.

Dále je na straně serveru ověřováno, zdali byl součástí odeslaného formuláře validní CSRF (*Cross-site request forgery*) token, který potvrzuje, že formulář nebyl odeslán neoprávněně (viz podkapitola 8.3).

Před vložením nebo úpravou dat v databázi dochází k ošetření uživatelského vstupu pomocí *prepared statements* pro zabránění útoku *SQL injection*.

8.2 Ošetření výstupu

Výstupy v aplikaci pocházející od uživatele nebo z jiného zdroje (včetně dat z databáze) jsou ošetřeny proti útoku XSS (*Cross-site scripting*).

Pokud by výstup nebyl ošetřen, bylo by možné v uživatelské části aplikace vykonat kód dopravený útočníkem (například by se mohlo jednat o zdrojový kód v jazyce *Javascript*, který by útočníkovi přeposlal autentizační token).

8.3 Obrana proti CSRF

Během přidávání, úpravy nebo mazání konkrétního záznamu je v serverové části aplikace ověřováno, zdali byl součástí formuláře tzv. *CSRF token*. Jedná se jedinečný otisk pro každého uživatele, který je generován při přihlášení do systému a který je platný právě po dobu přihlášení. Bez předání validního otisku není možné v aplikaci provádět klíčové operace.

Pokud by došlo k odeslání formuláře bez validního otisku (např. vzdáleným voláním a odesláním dat ve formuláři), došlo by k zamezení provedení takové operace. Vzhledem k tomu, že je *CSRF token* přenášen pomocí metody HTTP POST a po protokolu HTTPS, není možné standardními postupy daný otisk ani odposlechnout.

8.4 Zabezpečení cookies

V HTTP *cookies* je uložen otisk `PHPSESSID`, který identifikuje konkrétní *session* soubor uložený na serveru. V souboru jsou uloženy informace o přihlášeném uživateli, a pokud by tak došlo k získání nebo odposlechnutí `PHPSESSID`, mohl by útočník získat přístup do systému.

Implementovaný systém pro zabezpečení a používání *cookies* definuje následující bezpečnostní opatření:

- `session.cookie_secure: true` – *cookie* bude přenášeno pouze po protokolu HTTPS,
- `session.cookie_httponly: true` – *cookie* nebude možné přečíst nebo odcizit na straně klienta pomocí jazyka *JavaScript*.

8.5 Skrytí adresářové struktury

Další zvýšení bezpečnosti bylo dosaženo vytvořením vhodné adresářové struktury projektu a odstíněním jádra systému.

K odstínění jádra došlo změnou volby `DocumentRoot` v konfiguraci webového serveru, pomocí níž lze specifikovat, který adresář je viditelný do *Internetu*. Díky provedenému nastavení jsou uživateli viditelné jen nezbytné HTML, CSS a *JavaScript* soubory a dále pak PHP soubor `index.php`, který zachycuje požadavky ze strany uživatele. Klíčové a konfigurační soubory jsou tak uživateli skryty a není možné se k nim dostat zadáním URL adresy. URL adresa je navíc ovlivněna přepisovacími pravidly definovanými v `.htaccess`.

Dále je rekurzivně na všechny veřejně přístupné adresáře aplikováno pravidlo `Options -Indexes` v souboru `.htaccess`, které zablokuje zobrazení výpisu souborů v daném adresáři (pokud se v něm nenalézá soubor `index.*`).

8.6 HTTPS hlavičky

Rovněž došlo k implementaci bezpečnostních HTTPS hlaviček (tzv. *secure headers* [8]), které celkově zvyšují zabezpečení webové aplikace.

Jedná se o hlavičky, které webový prohlížeč informují, jak nakládat s potenciálně nebezpečným obsahem a jak předcházet bezpečnostním chybám (např. útočníkem neoprávněně vložený zdrojový kód v jazyce *JavaScript*) [8].

Na základě doporučení [8] organizace OWASP (*Open Web Application Security Project*) došlo k implementaci následujících HTTPS hlaviček:

- politika HSTS – vynucuje komunikaci mezi serverem a klientem pouze pomocí zabezpečeného protokolu HTTPS [8],
- `X-Frame-Options` s nastavením `deny` – zablokuje vykreslení stránky uvnitř rámu (HTML tag `<iframe>`) [8],
- `Content-Security-Policy` – zablokuje externí obsah (např. CSS a *Javascript* soubory) z jiných než autorem aplikace povolených zdrojů [8],
- `Referrer-Policy` s nastavením `strict-origin-when-cross-origin` zabezpečující přenos požadavků v `Referer` omezeným na doménu [8].

9 Pilotní provoz

V rámci diplomové práce došlo již v počátcích vývoje k nasazení celého řešení na virtuální server na pracovišti CIV ZČU. V souvislosti s nasazením bylo nutné provést i patřičnou konfiguraci serveru a zřídit potřebná oprávnění a testovací účty, na nichž byly testovány procesy identifikace, blokace a odblokování. Bez možnosti testování a pilotního provozu by nebylo možné vytvořený systém správně integrovat s existujícími univerzitními systémy a patřičně otestovat.

Vytvořený systém tak bylo možné průběžně testovat pracovníky bezpečnostního týmu WIRT přímo v ostrém provozu na ZČU. Na základě testování bylo možné reagovat na případné nepřesnosti a systém upravovat či doplňovat o nové funkce.

9.1 Testování

Aby bylo možné porovnat situaci před a po nasazení implementovaného CAIH systému, byly u dvou zaměstnanců bezpečnostního týmu testovány tři scénáře, během nichž bylo cílem vyřešit tři různé bezpečnostní incidenty.

Každý z incidentů pocházel z jiného typu sítě (*eduroam*, VPN a pevná síť), přičemž úkolem zaměstnanců bylo na základě IP adresy, data a času uvedeného ve stížnosti, vyhledat v logách aktivitu konkrétního testovacího uživatele ze ZČU (popř. zařízení v případě pevné sítě) a jeho činnost zablokovat. Po zablokování bylo nutné upozornit stěžovatele na vyřešení incidentu a poté problémového uživatele (zařízení) odblokovat.

Testovány byly následující tři způsoby (postupy) řešení:

1. manuálním vyřešením incidentu,
2. vyřešením incidentu pomocí systému *MySphere2* (viz kapitola 5),
3. vyřešením incidentu pomocí systému CAIH implementovaného v rámci diplomové práce.

U každé ze situací byly zaznamenávány následující časy úspěšného dokončení každého z kroků:

1. *čas zahájení* řešení incidentu,

2. *čas identifikace* původce incidentu (včetně přihlášení do systému, vyplnění potřebných údajů, jako je IP adresa, inkriminované datum a čas apod.),

3. *čas zablokování* (spočívá ve vytvoření lístku v RT systému, vložení komentáře k lístku o provedení blokace v RT systému, vložení poznámky o provedené blokaci do systému *WhoIS*, zablokování uživatele v externím systému),

4. *čas upozornění stěžovatele* (spočívá v odeslání odpovědi z RT systému stěžovateli a vytvoření vazby na lístek s problémovým uživatelem nebo zařízením),

5. *čas odblokování* (spočívá v odeslání zprávy o provedeném odblokování původně zablokovanému uživateli, vložení poznámky o odblokování do systému *WhoIS* a samotné odblokování uživatele v externím systému),

6. *čas vyřešení* incidentu.

Cílem bylo zjistit, kolik času potřebuje zaměstnanec bezpečnostního týmu k vyřešení incidentu v různých typech sítě v závislosti na zvoleném postupu.

9.1.1 Startovní podmínky

Pro každý scénář jsou uvažovány následující startovní podmínky:

- otevřené anonymní okno webového prohlížeče bez dosud provedené autentizace v SSO,

- otevřený terminál bez provedené autentizace do služby *Kerberos* (tj. výstup příkazu `klist` v terminálu neukazuje žádné existující lístky),

- po kroku *upozornění stěžovatele* dojde k vypnutí internetového připojení, k otevření nového anonymního okna webového prohlížeče a ke znovu přihlášení (tímto bodem je simulován čas, během něhož se čeká na reakci zablokovaného uživatele, protože ve skutečnosti také není uživatel okamžitě po zablokování odblokován).

9.1.2 Výsledky

Během měření bylo zaznamenáno 108 různých hodnot (3 různé incidenty $\times$ 3 postupy $\times$ 6 časových okamžiků u každého měření $\times$ 2 pracovníci WIRT). Naměřené časy pracovníků byly následně zprůměrovány. Průměrný čas nutný k vyřešení incidentu v závislosti na použitém postupu ukazuje tabulka 9.1.

	Síť *eduroam*	Síť VPN	Pevná síť
Manuální postup	15:30	17:00	9:20
Systém *MySphere2*	6:30	10:00	6:45
Implementovaný CAIH systém	2:45	2:50	2:30

Tabulka 9.1: Průměrný čas nutný k vyřešení incidentu v různých typech sítě a v závislosti na použitém postupu.

Naměřené hodnoty v tabulce 9.1 dokládají, že řešení incidentu se při použití systému CAIH z diplomové práce výrazně urychlilo. Urychlení bylo zaznamenáno jak vůči manuálnímu postupu, tak vůči existujícímu softwaru *MySphere2*. Konkrétní hodnoty urychlení jsou uvedeny v tabulce 9.2.

Zároveň si lze v tabulce 9.1 všimnout podobných časů, kterých bylo dosaženo při řešení incidentu pomocí implementovaného CAIH systému. Podobných časů je docíleno především díky jednotnému rozhraní systému, které je nezávislé na tom, v jakém typu sítě se incident řeší.

Výsledky v tabulce 9.1 také ukazují, že výrazné snížení času nutného k vyřešení incidentu umožnil i již dříve nasazený software *MySphere2*. Díky nasazení systému CAIH z diplomové práce ovšem došlo k ještě dalšímu urychlení, a to i vůči nástroji *MySphere2* (viz poslední řádek v tabulky 9.2).

Reálně by bylo možné dosáhnout ještě lepších časů (a většího urychlení), ale jak již bylo uvedeno v podkapitole 9.1.1, po *upozornění stěžovatele* dochází k úmyslnému vypnutí internetového připojení a k odhlášení se ze systémů. V každém z časů je tak započítáno i nutné přihlašování do systému SSO a další režijní činnosti, které by ovšem byly prováděny i při standardním postupu při řešení incidentu.

	Síť *eduroam*	Síť VPN	Pevná síť
Urychlení vůči manuálnímu postupu	5,64×	6,00×	3,73×
Urychlení vůči systému *MySphere2*	2,35×	3,53×	2,70×

Tabulka 9.2: Urychlení při řešení incidentu pomocí implementovaného CAIH systému vůči ostatním způsobům řešení v různých typech sítě.

Průměrně tak vytvořený CAIH systém urychlil řešení incidentu:

- 5,1× oproti manuálnímu postupu,
- 2,9× oproti původně používanému nástroji *MySphere2*.

10 Závěr

Cílem diplomové práce bylo seznámit se s fungováním bezpečnostních týmů typu CERT/CSIRT, s jejich postupy a reakcemi při řešení bezpečnostních incidentů a dále provést analýzu řešených bezpečnostních incidentů na ZČU. Na základě zjištěných postupů a analýzy řešených incidentů na ZČU byl navržen a implementován software typu CAIH, který bezpečnostnímu týmu WIRT na ZČU usnadní řešit běžně se vyskytující bezpečnostní incidenty, a to automatizací celého procesu. Automatizace je docíleno především úspěšnou integrací s několika již existujícími univerzitními systémy a jejím převedení do jednotného, konzistentního a intuitivního procesu. Zároveň tak bylo dosaženo snížení nutné kvalifikace a doby školení nového zaměstnance, neboť nutné činnosti pro řešení incidentu nyní dokáže zajistit implementovaný CAIH systém.

Vytvořený systém, nazvaný *Blockator*, byl již v rámci diplomové práce nasazen do ostrého provozu na serveru ZČU. Software tak bylo možné testovat přímo zaměstnanci bezpečnostního týmu WIRT, a to na skutečných bezpečnostních incidentech a na skutečných datech.

Přínos vytvořeného softwaru dokládá i kapitola 9 o pilotním provozu, v níž dochází k porovnání doby nutné pro vyřešení incidentu před a po zavedení systému *Blockator*. Výsledky ukazují, že díky systému implementovaného v rámci diplomové práce se řešení incidentu průměrně urychlilo až téměř pětinásobně oproti manuálnímu postupu a trojnásobně oproti aktuálně používanému systému *MySphere2*. Dle měření je díky systému *Blockator* možné projít procesem řešení incidentu maximálně do tří minut. Rychlost reakce bezpečnostního týmu na incidenty se tak může významně zrychlit.

Blockator bude nadále testován v ostrém provozu a postupně doplňován o nové funkce (například generování exportovatelné statistiky řešených incidentů), případně upravován v závislosti na řešených incidentech. Zároveň se nabízí možnost implementovat další funkce, jako například automatické odchytávání hlášení od důvěryhodných bezpečnostních týmů a automaticky na hlášení reagovat. Systém by tak v pravidelných intervalech ověřoval, zdali dorazila elektronicky podepsaná stížnost od jiného, důvěryhodného bezpečnostního týmu a v případě jejího výskytu by v systému všechny informace o incidentu předvyplnil a počkal na potvrzení pracovníka WIRT. Případně by bylo možné v nočních a brzkých ranních hodinách uvažovat režim „autopilot“, který by v uvedený čas na základě hlášení prováděl automatickou blokaci a vyřešení incidentu. Reakce na incident by tak mohla

být v řádu desítek sekund.

Vytvořený systém by bylo možné snadno nasadit i na dalších vysokých školách a obecně i v jiných sítích dalších institucí, neboť bezpečnostní týmy v nich řeší stejné typy bezpečnostních incidentů. Systém je navíc modulární a je možné jej v případě potřeby dále rozšiřovat.

Všechny cíle diplomové práce byly splněny.

Přehled zkratek

API	Application Programming Interface
CERT	Computer Emergency Response Team
CAIH	Computer-aided Incident Handling
CIA	Confidentiality Integrity Availability
CIV	Centrum informatizace a výpočetní techniky
CSRF	Cross-site request forgery
CSIRT	Computer Security Incident Response Team
CSS	Cascading Style Sheets
CVE	Common Vulnerabilities and Exposures
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
EXE	Windows Executable File
FLAB	Forenzní laboratoř CESNET
FIRST	Forum of Incident Response and Security Teams
GUI	Graphical User Interface
HSTS	HTTP Strict Transport Security
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IP	Internet Protocol address
IS/STAG	Informační systém studijní agendy
IT	Informační technologie
JIS	Jednotný identifikační systém
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
MAC	Media Access Control
MVC	Model–view–controller
NAS	Network Attached Storage
NÚKIB	Národní úřad pro kybernetickou a informační bezpečnost
OČTŘ	Orgány činné v trestním řízení
OTRS	Open-Source Ticket Request System

OWASP	Open Web Application Security Project
P2P	Peer-to-peer
PDO	PHP Data Objects
PGP	Pretty Good Privacy
PHP	Hypertext Preprocessor
RADIUS	Remote Authentication Dial In User Service
REST	Representational State Transfer
RFC	Request for Comments
RT	Request Tracker
SEČ	Středoevropský čas
SMB	Server Message Block
SMP	Správa mobilního připojení ZČU
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
SSO	Single Sign-On
SQL	Structured Query Language
TI	Trusted Introducer
TLP	Traffic Light Protocol
URL	Uniform Resource Locator
VIS	Významný informační systém
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WEBnet	West Bohemia network
WIRT	WEBnet Incident Response Team
XML	Extensible Markup Language
XSS	Cross-site scripting
ZČU	Západočeská univerzita v Plzni

Literatura

[1] *Správa mobilního připojení* [online]. Support ZČU, 2013. [cit. 2021-05-13]. Dostupné z: https://support.zcu.cz/index.php/Správa_mobilního_připojení.

[2] *Team Database* [online]. Trusted Introducer, 2021. [cit. 2021-03-06]. Dostupné z: https://www.trusted-introducer.org/directory/teams.html.

[3] *Traffic Light Protocol (TLP)* [online]. FIRST — Forum of Incident Response and Security Teams. [cit. 2021-02-25]. Dostupné z: https://www.first.org/tlp/.

[4] *FIRST Members around the world* [online]. FIRST — Forum of Incident Response and Security Teams. [cit. 2021-03-04]. Dostupné z: https://www.first.org/members/map.

[5] Analýza hrozby: Kybernetické útoky na vysoké školy jsou stále častější (dokument v režimu TLP:AMBER). 2021.

[6] *Doporučení pro případ napadení DDoS útokem – jak se zachovat a jak postupovat* [online]. Národní úřad pro kybernetickou a informační bezpečnost (NÚKIB), 2013. [cit. 2021-03-04]. Dostupné z: https://www.nukib.cz/cs/infoservis/doporuceni/1452-doporuceni-pro-pripad-napadeni-ddos-utokem-jak-se-zachovat-a-jak-postupovat/.

[7] *FIRST History* [online]. FIRST — Forum of Incident Response and Security Teams. [cit. 2021-03-04]. Dostupné z: https://www.first.org/about/history.

[8] *OWASP Secure Headers Project* [online]. OWASP, 2021. [cit. 2021-05-15]. Dostupné z: https://owasp.org/www-project-secure-headers/.

[9] *REST – Request Tracker Wiki* [online]. 2021. [cit. 2021-05-11]. Dostupné z: https://rt-wiki.bestpractical.com/wiki/REST.

[10] *Administration Guide* [online]. Syslog-ng, 2021. [cit. 2021-05-14]. Dostupné z: https://www.syslog-ng.com/technical-documents/doc/syslog-ng-open-source-edition/3.22/administration-guide/.

[11] Analýza hrozby: Vyděračské útoky ransomwarem jsou cílenější. *Národní úřad pro kybernetickou a informační bezpečnost (NÚKIB)*. 2020, s. 1–13. Dostupné z: https://www.nukib.cz/download/publikace/analyzy/Analyza_hrozby_ransomware.pdf.

[12] *Doporučení k používání protokolu TLP ke sdílení chráněných informací* [online]. Národní úřad pro kybernetickou a informační bezpečnost (NÚKIB), 2020. [cit. 2021-03-04]. Dostupné z: https://nukib.cz/cs/infoservis/doporuceni/1593-doporuceni-k-pouzivani-protokolu-tlp-ke-sdileni-chranenych-informaci/.

[13] *Grid system* [online]. Bootstrap, 2021. [cit. 2021-05-16]. Dostupné z: https://getbootstrap.com/docs/5.0/layout/grid/.

[14] *RFC 2350 standard* [online]. Národní úřad pro kybernetickou a informační bezpečnost (NÚKIB), 2020. [cit. 2021-02-25]. Dostupné z: https://nukib.cz/cs/kyberneticka-bezpecnost/vladni-cert/rfc-2350-standard/.

[15] *Směrnice rektora č. 10R/2008 – Pravidla používání sítě WEBnet* [online]. Support ZČU, 2008. [cit. 2021-03-04]. Dostupné z: https://support.zcu.cz/index.php/Pravidla_používání_sítě_WEBnet.

[16] *Úvod do architektury MVC* [online]. 2009. [cit. 2021-05-11]. Dostupné z: https://zdrojak.cz/clanky/uvod-do-architektury-mvc/.

[17] Bodó, R. – Kostěnec, M. Zkvalitnění procesu řešení bezpečnostních incidentů v síti WEBnet. 2011.

[18] Duračinská, Z. Základy fungování CSIRT týmu. Praha, 2017. CZ.NIC.

[19] Duračinská, Z. Bezpečnostní týmy v Evropě i ve světě. *SecurityWorld*. 2017, 2017, 1, s. 40–41.

[20] Javorník, M. Incident Handling. Brno, 2021. MUNI CSIRT-MU.

[21] Kolouch, J. – Bašta, P. *CyberSecurity*. CZ.NIC, z.s.p.o., 2019. ISBN 978-80-88168-31-7.

[22] Kropáčová, A. *CERT/CSIRT týmy a jejich role* [online]. Root.cz, 2013. [cit. 2021-03-04]. Dostupné z: https://www.root.cz/clanky/cert-csirt-tymy-a-jejich-role/.

[23] Padrta, A. *Řešení bezpečnostních incidentů* [online]. Support ZČU, 2003. [cit. 2021-03-07]. Dostupné z: https://support.zcu.cz/index.php/Řešení_bezpečnostních_incidentů.

[24] Padrta, A. Problematika bezpečnostních týmů a řešení bezpečnostních incidentů. Plzeň, 2021. FLAB CESNET.

[25] Padrta, A. – Orkáč, R. Phishing – Postup při řešení incidentu. Praha, 2019. Seminář o bezpečnosti, FLAB CESNET.

Přílohy

A Uživatelská příručka

V následujících podkapitolách jsou uvedeny klíčové informace a postupy k používání aplikace *Blockator* (Systém pro správu bezpečnostních incidentů).

A.1 O aplikaci

Aplikace *Blockator* (Systém pro správu bezpečnostních incidentů) slouží ke správě bezpečnostních incidentů formou intuitivního webového rozhraní. Do aplikace je možné se přihlásit na URL adrese `https://blockator.zcu.cz`.

Blockator je určen oprávněným uživatelům, resp. bezpečnostnímu týmu WIRT na ZČU, přičemž cílem systému je usnadnit a zefektivnit proces řešení bezpečnostních incidentů, ke kterým dochází v různých typech univerzitní sítě. Systém automatizuje proces řešení incidentu a integruje již existující systémy, které zabezpečují některé z klíčových činností (např. blokaci původce incidentu – problémového uživatele, nebo zařízení). Při výskytu incidentu je díky tomu možné využít jednotné rozhraní systému *Blockator* nezávisle na tom, v jaké části univerzitní sítě k incidentu došlo.

Aby bylo možné v systému bezpečnostní incidenty kategorizovat, lze v sekci *Typy incidentů a šablony* definovat konkrétní typy incidentů včetně šablon s odpověďmi pro původce incidentu a pro stěžovatele. Šablony s odpověďmi lze personalizovat díky možnosti používat proměnné, které jsou před odesláním nahrazeny skutečnými údaji.

Systém obsahuje i správu lokálních správců, a to především ve spojitosti s incidenty, které se vyskytují v pevné síti (např. zařízení na jednotlivých katedrách či odděleních). U každého lokálního správce je možné specifikovat konkrétní oblast, kterou má ve správě.

Vznik aplikace proběhl v ak. roce 2020/2021 na FAV ZČU formou diplomové práce *Systém pro správu bezpečnostních incidentů (Martin Šebela)* pod vedením *Ing. Jiřího Čepáka*. Text diplomové práce obsahuje konkrétní implementační detaily.

A.2 Návod pro uživatele

Pro vstup do systému *Blockator* je nutné přejít na URL adresu `https://blockator.zcu.cz` a do systému se přihlásit autentizací ve službě jednotného přihlášení (SSO).

Registrace do systému probíhá automaticky na základě údajů předaných službou SSO. Po přihlášení je uživatel automaticky přesměrován na sekci požadovanou v URL adrese, případně na úvodní stránku systému.

Následující podkapitoly popisují jednotlivé sekce v systému *Blockator* a možnosti, které lze v sekcích vykonávat.

Úvodní stránka

Cílem úvodní stránky je poskytovat zaměstnanci bezpečnostního týmu aktuální a rychlý přehled o řešených bezpečnostních incidentech, a to jak formou stručné statistiky, tak formou tří grafů. Z pohledu grafů se jedná o:

1. koláčový graf zobrazující aktuální rozložení stavů u všech řešených incidentů (tj. počet incidentů ve stavu *vyřešeno*, počet incidentů ve stavu *blokace* apod.),
2. koláčový graf zobrazující počet všech řešených incidentů dle typu (např. incidenty způsobené *sdílením v P2P sítích*, incidenty způsobující *rozesílání spamu* apod.),
3. sloupcový graf zobrazující počty incidentů řešených v aktuálním roce, a to rozložením do jednotlivých měsíců a barevně odlišených dle jejich typu.

Názvy typů incidentů, které jsou zobrazeny v grafech, stejně jako barvy použité v grafech, lze upravit v sekci *Typy incidentů a šablony*.

Nový incident

Jedná se o hlavní sekci celého systému, v níž dochází k prvotním krokům při řešení bezpečnostního incidentu. Celý proces je rozdělen buď do čtyř, nebo do pěti kroků. Do pěti kroků je řešení incidentu rozděleno tehdy, pokud je v prvním kroku vyplněno číslo lístku se stížností ze systému pro správu požadavků. V takovém případě je přidán krok *Upozornění stěžovatele*, jehož cílem je odpovědět na stížnost uvedenou v prvním kroku.

Po úspěšném dokončení každého z kroků je možné se k datům ve formuláři zpětně vrátit (i u incidentu, kde zatím proběhla pouze identifikace původce). Nedokončený incident lze případně smazat tlačítkem *Smazat formulář* v pravém horním rohu.

Založení incidentu

Cílem prvního kroku je předat systému základní informace o řešeném incidentu, a to především o IP adrese a datu a času vzniku incidentu. Vstupní pole formuláře jsou tedy následující:

1. *IP adresa*, která je původcem incidentu,
2. *MAC adresa zařízení*, se kterou byla IP adresa spjata (slouží k zacílení vyhledávání; vyplnění je nepovinné),
3. *typ incidentu* (pro následnou kategorizaci a pro nahrání šablon odpovědí v kroku *blokace*, *upozornění stěžovatele* a v kroku *odblokování*).
4. *číslo lístku se stížností* ze systému pro správu požadavků (při vyplnění bude součástí procesu řešení incidentu i krok *upozornění stěžovatele*; vyplnění je nepovinné).

Pokud se nepodaří IP adresa v inkriminovaném čase v žádném z logů nalézt, je uživateli zobrazena chybová zpráva. V opačném případě dojde k přesměrování na následující krok.

Identifikace

Během identifikace jsou uživateli zobrazeny všechny nalezené záznamy z logu, v nichž se vyskytla hledaná IP adresa, a to v datu a čase, který byl vyplněn v předchozím kroku. Pokud jsou v záznamech logu k dispozici i další informace (MAC adresa a hostname zařízení), jsou doplněny do vypisované tabulky. Zároveň dojde k identifikaci původce incidentu (uživatelské jméno).

Všechny nalezené informace, které mohou sloužit k identifikaci, jsou v jednotlivých záznamech z logů pro přehlednost barevně odlišeny.

Pokud se jedná o pevnou síť, není identifikován původce incidentu (protože se jedná o zařízení), ale lokální správce, v jehož kompetenci dané zařízení je. Lokální správce je určen na základě hostname zařízení, které dle konvencí na ZČU typicky obsahuje buď název katedry, oddělení nebo fakulty.

V případě, že je nalezeno více možných původců incidentu či více lokálních správců, dá systém uživateli na výběr, který z původců má být zablokován (zablokován bude pouze jeden) či který lokální správce má být kontaktován (lokálních správců lze naopak zvolit několik).

Blokace

Cílem blokace je odeslat upozornění původci bezpečnostního incidentu a problémového uživatele či zařízení zablokovat.

Do vstupních polí s předmětem a obsahem zprávy jsou automaticky nahrány šablony odpovědí podle typu incidentu zvoleného v prvním kroku (*Založení incidentu*). Šablony jsou nahrány v jazykových mutacích, které byly vybrány v kroku *Identifikace*. Vstupní pole jsou tedy následující:

1. *e-mail příjemce zprávy o blokaci* – e-mail uživatele, který byl vybrán v předchozím kroku *Identifikace*), lze případně manuálně dopsat další příjemce (oddělovačem e-mailů je čárka, tj. symbol **,**)

2. *předmět zprávy* – personalizovaný předmět e-mailu a zároveň název lístku v systému pro správu požadavků (doplněno na základě šablony, lze případně upravit),

3. *obsah zprávy* – personalizovaný obsah šablony (viz sekce *Typy incidentů a šablony*), lze případně upravit.

Stiskem tlačítka *Odeslat upozornění a zablokovat* dojde k vytvoření lístku v systému pro správu incidentu, kde jako žadatel bude uveden problémový uživatel či lokální správce vyplněný v poli *e-mail příjemce zprávy o blokaci*. Zároveň tím dojde k zablokování činnosti daného uživatele či zařízení.

Upozornění stěžovatele

Krok *upozornění stěžovatele* je zobrazen pouze tehdy, pokud bylo v první kroku vyplněno číslo lístku se stížností ze systému pro správu požadavků.

Do vstupních polí formuláře je automaticky načten obsah, a sice:

1. *e-mail stěžovatele* – e-mail žadatele (odesílatele stížnosti) ze systému pro správu požadavků (již nelze změnit),

2. *obsah zprávy* – personalizovaný obsah šablony s odpovědí pro stěžovatele (viz sekce *Typy incidentů a šablony*), lze případně upravit (obsah zprávy je nepovinný, viz dále).

Pokud nebude obsah zprávy vyplněn, bude stěžovateli odesláno pouze automatické, informativní sdělení o vyřešení stížnosti (resp. o uzavření požadavku v systému pro správu požadavků).

Odblokování

Posledním krokem je odblokování původce incidentu – problémového uživatele nebo zařízení.

Do vstupních polí formuláře je automaticky načten obsah, a sice:

1. *e-mail příjemce zprávy o blokaci* – e-mail problémového uživatele nebo lokálního správce ze systému pro správu požadavků (již nelze změnit, vychází z e-mailu uvedeného v kroku *blokace*),

2. *obsah zprávy* – personalizovaný obsah šablony s odpovědí (viz sekce *Typy incidentů a šablony*), lze případně upravit.

Odesláním formuláře dojde k odblokování daného uživatele nebo zařízení a k vyřešení incidentu. Uživatel je po vyřešení incidentu automaticky přesměrován do sekce *Seznam incidentů*.

Seznam incidentů

V sekci lze nalézt seznam všech řešených bezpečnostních incidentů řazených od nejnovějšího po nejstarší (tzn. navrchu je nejnověji řešený incident).

Po otevření sekce je k dispozici jen stručný přehled o každém z incidentů, na základě něhož je zřejmé, v jakém stavu každý z incidentů je, kdo byl původcem incidentu (a další údaje) a dále pak především odkazy do systému pro správu požadavků, kde jsou evidovány lístky s původcem incidentu a se stěžovatelem (pokud byla stížnost součástí incidentu).

Pro zjištění více informací o každém z incidentů nebo k dořešení incidentu (tj. incident dosud není ve stavu *vyřešeno*) slouží tlačítko *Upravit*. V sekci *Seznam incidentů* se každý incident objevuje od stavu *blokace*.

V souhrnném přehledu je rovněž ve sloupci *Zablokováno* informace o rychlosti reakce – ta je definována jako rozdíl dvou časů, a sice kdy došlo k zablokování problémového uživatele nebo zařízení (tj. stav *blokace*), od něhož je odečteno datum vzniku incidentu. Hodnota tedy udává, po kolika hodinách a minutách od vzniku incidentu došlo k zablokování původce incidentu.

Typy incidentů a šablony

Sekce slouží k definování vlastních typů incidentů a šablon odpovědí, které s daným typem incidentu souvisí.

Během procesu řešení incidentu je totiž požadováno, aby zaměstnanec bezpečnostního týmu uvedl, o jaký typ incidentu se jedná. Na základě zvoleného typu incidentu je následně incident kategorizován v souhrnném přehledu a v grafech na *Úvodní stránce* systému. Zvolený typ incidentu zároveň určuje, jaké šablony odpovědí budou používány.

V sekci *Typy incidentů a šablony* lze tedy provádět následující čtyři operace:

- přidat nový typ incidentu,

- upravit záznam o již přidaném typu incidentu,
- smazat záznam o existujícím typu incidentu,
- upravit šablony odpovědí pro daný typ incidentu.

Formulář pro přidání nebo úpravu incidentu obsahuje následující vstupní pole:

1. *název* typu incidentu, který bude viditelný v přehledu a v grafech,
2. *popis* charakterizující typ incidentu (nepovinné),
3. *barva* v HEX formátu včetně znaku # pro barevné odlišení typu incidentu v přehledu a v grafech.

U každého z typů incidentů je následně možné doplnit šablony odpovědí, které jsou určeny pro původce incidentu a pro stěžovatele. U každé šablony systém umožňuje definovat dvě jazykové mutace, a sice českou a anglickou. Obsah šablony bude nahrán a uživateli zobrazen během procesu řešení incidentu, a to při dosažení stavu, pro který je šablona určena. Šablony lze upravovat u každého z incidentů a pro každý z následujících stavů:

- šablona pro stav *blokace*,
- šablona pro stav *upozornění stěžovatele*,
- šablona pro stav *odblokování*.

V šablonách odpovědí lze používat proměnné, které budou při procesu řešení incidentu nahrazeny skutečným obsahem. Systém umožňuje používat následující proměnné, a to jak v předmětu, tak v těle šablony:

- `%incident_user%` – jméno a příjmení uživatele spojeného s incidentem,
- `%incident_network%` – název sítě, ve které k bezpečnostnímu incidentu došlo,
- `%csirt_member_name%` – jméno a příjmení člena bezpečnostního týmu,
- `%csirt_name%` – název bezpečnostního týmu včetně zkratky.

Lokální správci

Sekce umožňuje spravovat seznam lokálních správců, přičemž informace o lokálních správcích jsou využívány při hledání kompetentní osoby, která má práva a možnosti vyřešit bezpečnostní incident pocházející z pevné sítě v konkrétním oddělení (např. na určité katedře fakulty).

Konkrétně lze v sekci *Lokální správci* provádět následující akce:

- přidat nového lokálního správce,
- upravit záznam o již přidaném lokálním správci,
- smazat záznam o existujícím lokálním správci.

Formulář pro při přidání nebo úpravu záznamu o lokálním správci obsahuje následující vstupní pole:

1. *univerzitní e-mail* (na základě e-mailu bude automaticky dohledáno jméno a příjmení lokálního správce z informací v LDAP),
2. *zkratka fakulty* (popř. budovy), kterou má lokální správce ve správě,
3. *zkratka katedry* (oddělení), kterou má lokální správce ve správě (nepovinné),
4. interní *poznámka* (nepovinné).

Vhodný lokální správce je zaměstnanci bezpečnostního týmu zobrazen v kroku *Identifikace* při řešení bezpečnostního incidentu pocházejícího z pevné sítě.

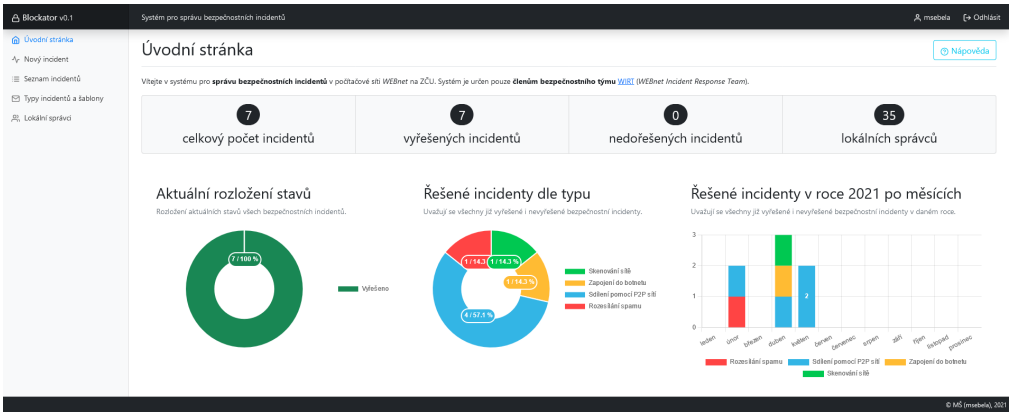
Systém vhodného lokálního správce vybírá na základě *zkratky fakulty* a *zkratky katedry* vyplněné u lokálních správců. Zařízení připojená do pevné sítě totiž ve svém hostname obsahují právě zkratku fakulty, katedry nebo oddělení.

Pokud je na katedře (oddělení) či fakultě definováno více lokálních správců (tj. *zkratka fakulty/katedry* obsahuje u více osob stejnou hodnotu), systém uživateli poskytne možnost si vybrat konkrétního lokálního správce, případně skupinu správců. S vybranými lokálními správci bude následně komunikováno v dalších krocích při řešení bezpečnostního incidentu (tj. budou uvedeni jako žadatelé lístku v systému pro správu požadavků).

B Obrazová příloha

Příloha zahrnuje několik screenshotů z implementovaného CAIH systému.

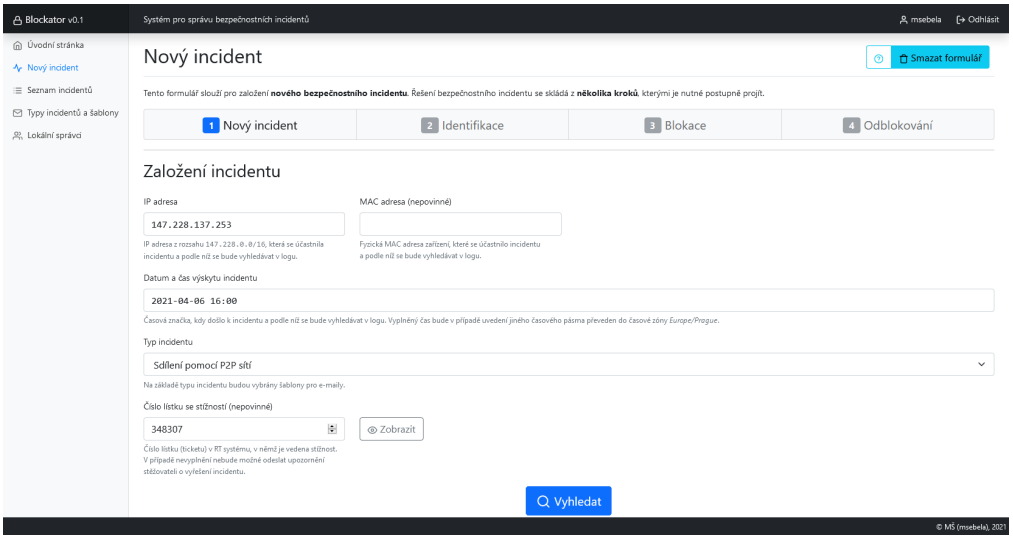
B.1 Úvodní stránka



Obrázek 10.1: Úvodní stránka systému se souhrnnou statistikou a s přehledem řešených bezpečnostních incidentů zobrazených formou grafů.

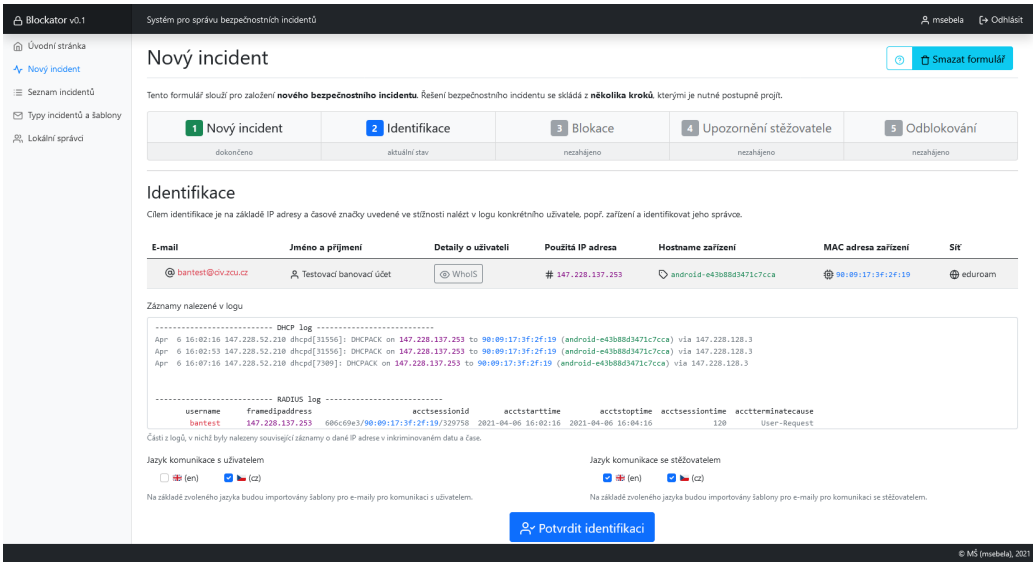
B.2 Proces řešení bezpečnostního incidentu

Založení incidentu



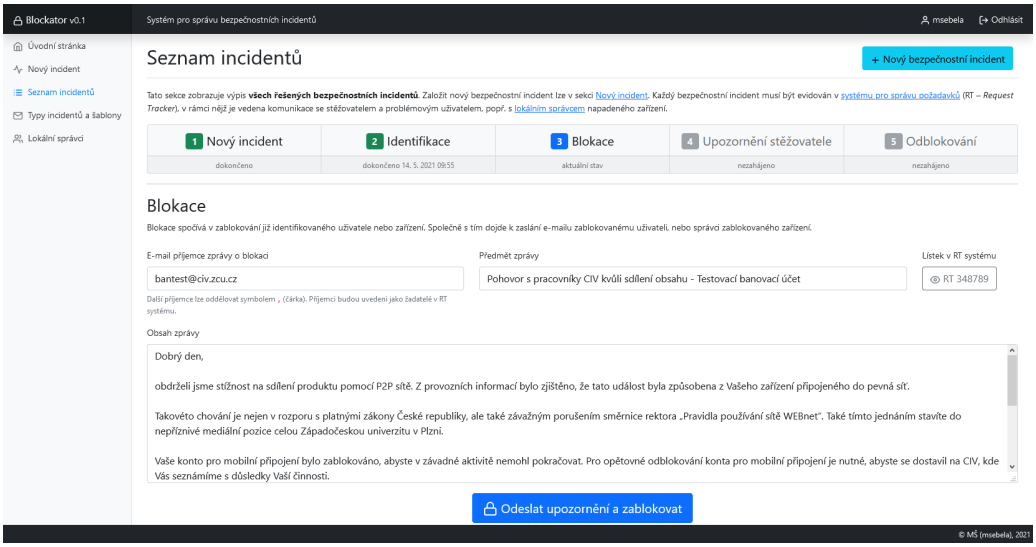
Obrázek 10.2: První krok slouží k vyplnění informací, které jsou předmětem incidentu – tedy především IP adresa a datum a čas výskytu incidentu.

Identifikace



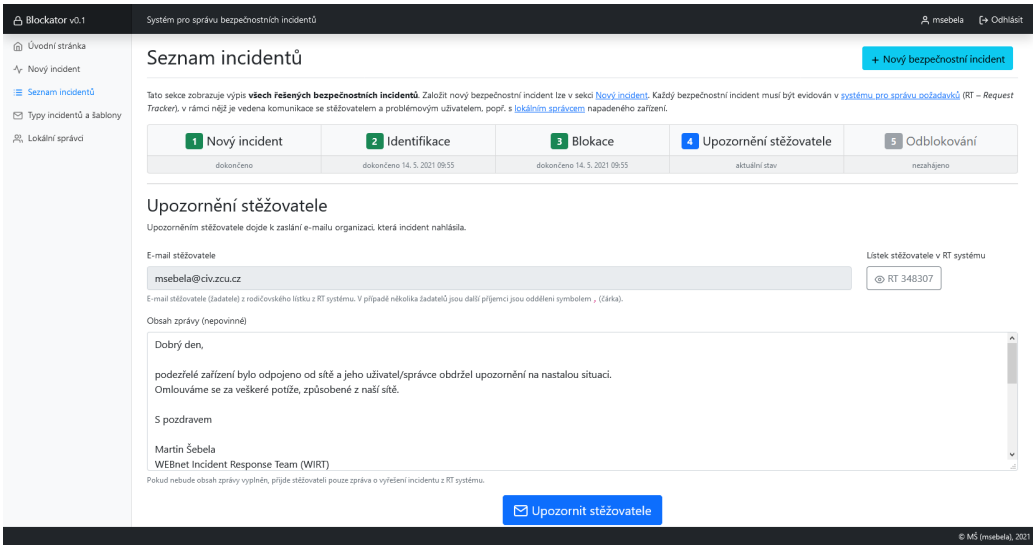
Obrázek 10.3: Identifikací dochází k vyhledání IP adresy v daném datu a čase v logu, tedy k identifikaci původce incidentu. Všechny podstatné informace jsou barevně zvýrazněny. V dolní části screenshotu si lze všimnout vybraných jazykových mutací šablon odpovědí, které jsou využity v dalších krocích.

Blokace



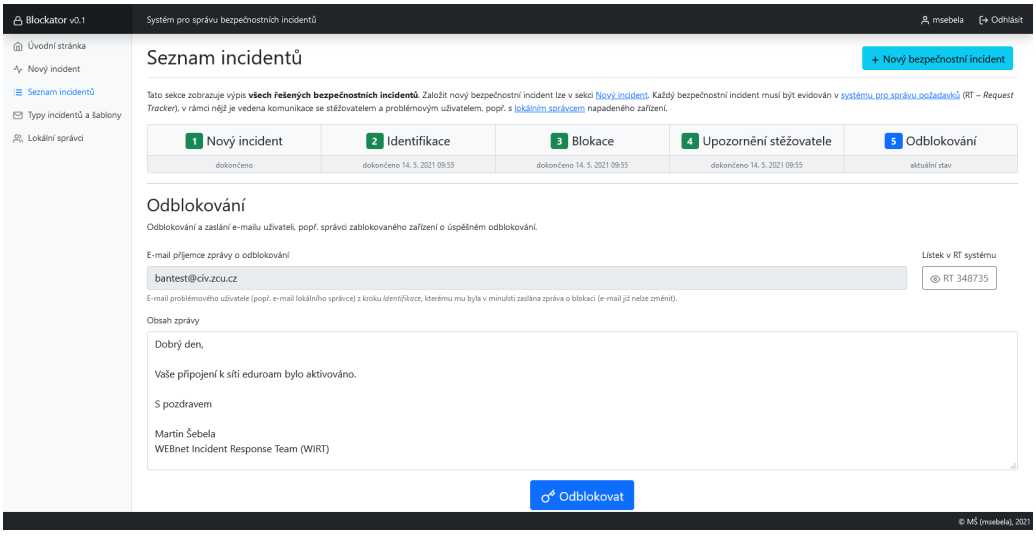
Obrázek 10.4: Blokací dojde k zablokování původce incidentu v daném typu sítě. Zároveň je zde vidět šablona odpovědi pro stav *blokace* (viz obr. 10.7).

Upozornění stěžovatele



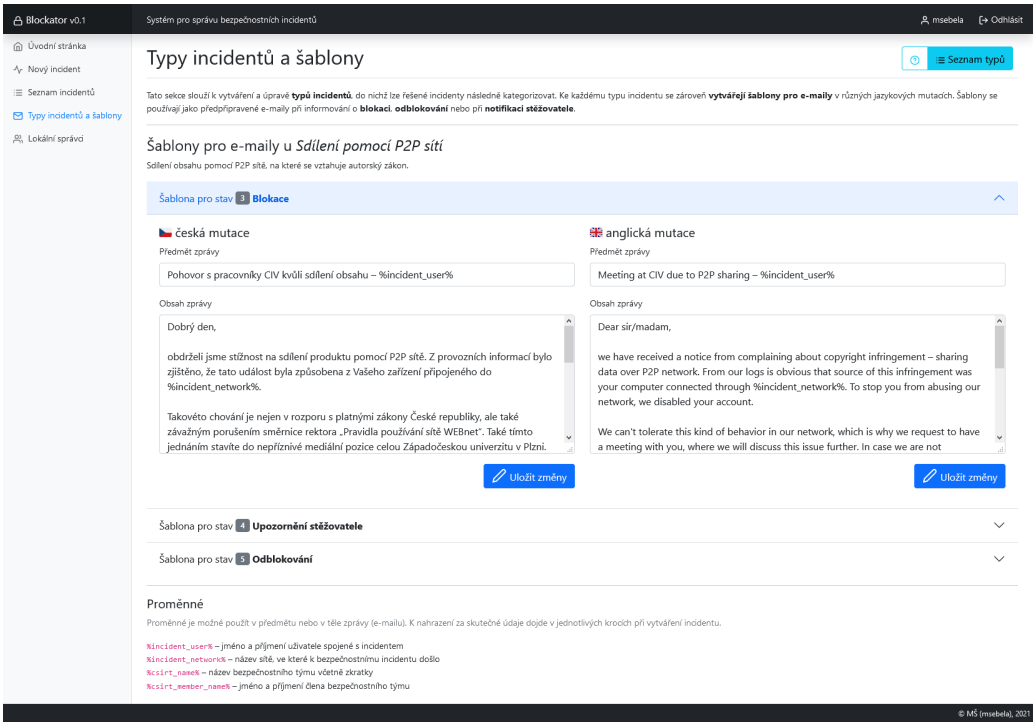
Obrázek 10.5: Krok *upozornění stěžovatele* je zobrazen pouze tehdy, pokud bylo v rámci prvního kroku (viz obr. 10.2) uvedeno číslo lístku se stížností ze systému pro správu požadavků. Zároveň je zde vidět načtená šablona odpovědi v jazykových mutacích vybraných v kroku na obr. 10.3.

Odblokování



Obrázek 10.6: Odblokování je posledním krokem procesu řešení bezpečnostního incidentu. V dolní části screenshotu je vidět načtená šablona odpovědi.

B.3 Šablony odpovědí



Obrázek 10.7: Příklad úpravy šablony v české a anglické jazykové mutaci, a to pro stav *blokace*, která bude řešiteli incidentu zobrazena tehdy, pokud bylo jako typ incidentu zvoleno *sdílení pomocí P2P sítí*. V šabloně jsou použity proměnné, které se před odesláním původci incidentu (zablokovanému uživateli) nahradí za skutečné údaje (např. za jméno uživatele nebo za název sítě, v níž k incidentu došlo apod.). V dolní části screenshotu jsou vidět odkazy na další šablony odpovědí, a to pro stav *upozornění stěžovatele* a pro stav *odblokování*.

C Struktura odevzdávaného archivu

- `/`
  - `Text_prace`
    - `src` ... Zdrojové soubory textu diplomové práce v LaTeX, zdrojové soubory schémat a diagramů a použité obrázky.
  - `Poster`
    - `src` ... Zdrojové soubory posteru včetně použitých diagramů.
  - `Aplikace_a_knihovny`
    - `src` ... Zdrojové soubory implementovaného CAIH systému.
    - `sql` ... SQL soubor pro vytvoření databáze a její naplnění základními daty.
  - `Vysledky` ... Zadání testovacích incidentů a průměrné časy jednotlivých kroků z testování.
  - `Readme.txt`