VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Lukáš Karabina

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNOSTNÍ HRY "O VLAJKU"
SECURITY GAMES "CAPTURE THE FLAG"

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Lukáš Karabina

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Zdeněk Martinásek, Ph.D.

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Lukáš Karabina ***ID:*** 203458

Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Bezpečnostní hry "O vlajku"

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce bude zprovoznění Capture the Flag (CTF) platformy a vytvoření několika úkolů na podporu výuky předmětů zabývajících se kryptografií a kybernetickou bezpečností. V rámci teoretické části se seznamte s bezpečnostními hrami, zaměřte se zejména na CTF. Analyzujte dostupné řešení a platformy pro realizaci her. Na základě rešerše dostupných bezpečnostních her, navrhněte hry vlastní. Důraz bude kladen na zpřístupnění hry pro hráče s různými dovednostmi a znalostmi (například využívání nápověd nebo vzorových řešení). V rámci práce zprovozněte platformu CTF a navrhněte nejméně dvě komplexní hry obsahující několik tématických úkolů. Jednotlivé úkoly budou cíleny na kryptografické problémy. Dále navrhněte a implementujte pět pokročilých her CTF zabývající se kybernetickou bezpečností (př. analýza ransomware, analýza incidentu, zátěžové testování atd.).

DOPORUČENÁ LITERATURA:

[1] CHUNG, Kevin. Live Lesson: Lowering the Barriers to Capture The Flag Administration and Participation. In: 2017 USENIX Workshop on Advances in Security Education (ASE 17). 2017.

[2] SCHREUDERS, Z. Cliffe, et al. Security Scenario Generator (SecGen): A Framework for Generating Randomly Vulnerable Rich-scenario VMs for Learning Computer Security and Hosting CTF Events. In: 2017 USENIX Workshop on Advances in Security Education (ASE 17). 2017.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Bakalářská práce je zaměřená na poskytnutí obecných informací o bezpečnostních hrách "o vlajku". Věnuje se základní kategorizaci tohoto druhu bezpečnostních her, dále popisuje vybrané kategorie pro úkoly, které lze do těchto her zahrnout. V práci jsou analyzovány vybrané dostupné platformy pro pořádání bezpečnostních her "o vlajku".
Práce se nejprve věnuje bezpečnostním hrám obecně, poté se věnuje obecným kategoriím úkolů, které jsou často v těchto hrách zahrnuty, dále pokračuje problematikou výběru vhodné platformy pro praktickou část práce, která má sloužit k podpoře výuky předmětu zabývajících se kryptografií a kybernetickou bezpečností a na závěr popisuje jednotlivé úkoly, které byly implementovány v praktické části.

KLÍČOVÁ SLOVA

Bezpečnostní hra, digitální forenzní analýza, informační bezpečnost, kryptografie, platforma pro bezpečnostní hry, soutěž, úkol

ABSTRACT

Bachelor's thesis is focused on providing general information about "capture the flag" security games. It deals with the basic categorization of this type of games, it also describes various categories for tasks that can be included within these games. The thesis analyzes selected available platforms for organizing "capture the flag" security games.
At first the thesis deals with security games in general, then deals with basic categorization of tasks, that are often included in these games, then continues with the issue of selecting a suitable platform for the practical part of the thesis to support the courses dealing with cryptography and cyber sercurity. Finally, thesis discribes the individual tasks that were implemented in its practical part.

KEYWORDS

Security game, digital forensic analysis, information security, crypthography, platform for security games, competition, task

KARABINA, Lukáš. *Bezpečnostní hry "O vlajku"*. Brno, 2020, 64 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Bezpečnostní hry "O vlajku"“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Zdeňku Martináskovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Tato práce vznikla jako součást klíčové aktivity KA6 - Individuální výuka a zapojení studentů bakalářských a magisterských studijních programů do výzkumu v rámci projektu OP VVV Vytvoření double-degree doktorského studijního programu Elektronika a informační technologie a vytvoření doktorského studijního programu Informační bezpečnost, reg. č. CZ.02.2.69/0.0/0.0/16_018/0002575.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

Obsah

Seznam obrázků

Úvod

Tuto práci jsem si vybral z důvodu zájmu o bezpečnostní hry. Věřím, že bezpečnostní hry dokážou hráče zábavnou formou naučit mnoho cenných zkušeností a pomoci mu rozvinou své dovednosti potřebné k získání, nebo naopak zabezpečení libovolné informace. Práce popisuje základní myšlenku bezpečnostních her, dostupné platformy, na který lze hry organizovat a jednotlivé úkoly, které je možné do bezpečnostních her zařadit. Bezpečnostní hry CTF poskytují hráči možnost vyzkoušet si principy reálných útoků a metod forenzní analýzy, které může využít v praxi. Dále hráče motivují ke zdokonalení se v používání programů a nástrojů, které jsou v tomto odvětví hojně užívané. Čtenáři tato práce poskytuje přehled o tom, co jsou bezpečnostní hry, jaké jsou volně dostupné platformy a může jej také inspirovat k vytvoření her vlastních, nebo vytvoření vlastních úkolů, podobných těm, které jsou v této práci zahrnuty.

1 Bezpečnostní hry

Bezpečnostní hry je skupina her, které jsou navrženy za účelem poskytnout zábavný i soutěživý způsob učení se a zdokonalování svých schopností a znalostí v oboru informační bezpečnosti. V dnešní době je informační bezpečnost velké téma naší společnosti a dotýká se každého uživatele sítě internet. Mnoho uživatelů však stále informační bezpečnost podceňuje a neuvědomuje si rizika s ní spojené. Právě proto je žádané rozšířit povědomí o informační bezpečnosti a přiblížit tuto problematiku i běžným uživatelům. A toho mohou docílit bezpečnostní hry. Bezpečnostní hry mohou být navrženy pro širokou skupinu lidí od laiků přes nadšence až po odborníky a profesionály. Záleží pouze na obtížnosti úkolů a zkušenostech hráčů. Zároveň se během hraní s každým splněným úkolem hráčovy schopnosti zdokonalují, a tak je možné obtížnost úkolů do jisté míry stupňovat.

1.1 Capture the flag

Původem Capture the flag (CTF, česky vlajková, nadále bude v práci využíván jen anglický pojem, jelikož je v oboru užívanější) je tradiční venkovní týmová aktivita, kdy hráči mají za cíl získat jednu společnou vlajku, nebo získat vlajku soupeře a ochránit vlajku vlastní. Adaptace této hry do kategorie her informační bezpečnosti je v principu taková, že vlajka se nahradí informací a venkovní prostředí se nahradí tím digitálním. CTF je velmi známým a rozšířeným druhem bezpečnostních her pro týmy i pro jednotlivce. CTF může mít různé podoby, přesto existují jakési dva základní přístupy. Prvním z nich je jeopardy (česky ohrožující, nadále bude v práci využíván jen anglický pojem, jelikož je v oboru užívanější)[1] a druhým z nich je attack-defense (česky útoč a braň, nadále bude v práci využíván jen anglický pojem, jelikož je v oboru užívanější). Případně tyto dva přístupy je možno libovolně kombinovat a záleží na organizátorovi CTF jaký přístup zvolí.

1.1.1 Jeopardy

Jeopardy CTF spočívá v tom, že hráč hledá tajné odpovědi řešením různých úkolů problematiky informační bezpečnosti, např. kryptografie, analýza přenosu, analýza dat, reverzní inženýrství, lámání hesel, lámání hash apod. Každý úkol po hráči vyžaduje skrytou odpověď, kterou hráč nalezne vyřešením problému, který je v úkolu popsán. Za získání odpovědi a splnění úkolu jsou zpravidla udělovány body. Hráč,

[1] Název jeopardy není pro tento případ CTF doslovný, ale jedná se o jakousi referenci na populární televizní vědomostní soutěž ze Spojených států, která byla vysílána v 70. a 80. letech.

resp. tým s nejvíce body na konci hry vyhrává. Při jeopardy CTF se zpravidla neútočí na server, který zajišťuje službu CTF jako takovou. Řešení daných úkolů obecně nebývá omezeno, a tak umožňuje hráči přistoupit k řešení daných problémů vlastním způsobem a nemusí se držet standardního přístupu nebo přístupu, který by očekával organizátor. Hráči po konci hry mohou sdílet tato zajímavá řešení s ostatními. Při řešení úkolů mají hráči obvykle možnost využít nápovědy k danému úkolu, která bývá bodově penalizována, a tak záleží na hráči, zda své body využije k získání nápovědy. Tento druh CTF bývá většinou pro jednotlivce a nevyžaduje fyzickou přítomnost hráče na akci, kde je CTF pořádáno. Z toho důvodu bývá časová omezenost hry v řádech dnů až několika měsíců, takže hráči mají dostatek času na řešení úkolů.

1.1.2 Attack-defense

Attack-defense CTF se zaměřuje spíše na týmy než na jednotlivce. Týmy mají vlastní síť, nebo stroj se zranitelností, které lze využít ke kompromitaci. Týmy se pak snaží zabránit soupeři zranitelnost využít a kompromitovat síť, nebo stroj a zároveň se snaží využít zranitelnosti v síti, resp. stroji soupeře a kompromitovat ji, resp. jej. Za obranu i útok tým dostává body a zároveň pokud se soupeři podaří jeho síť, nebo stroj kompromitovat může body ztratit. Jedná se tak o formu souboje mezi týmy. Při tomto druhu CTF jsou všem týmům poskytnuty stejné zdroje, aby rozhodovaly pouze schopnosti bránit a útočit. Na členech týmu pak záleží, kolik zdrojů alokují do útoku na cizí zdroje a kolik do obrany zdrojů vlastních. Zpravidla se útočí pouze na síť, nebo stroj, kde se nachází zmíněná zranitelnost a neútočí se na hodnotící server, pracovní stanice členů týmu, jiné služby sítě, resp. stroje nebo postranní kanály. Toto však není pevným pravidlem a záleží na organizátorovi, jaká pravidla hráčům nastaví. Attack-defense CTF většinou vyžaduje fyzickou přítomnost členů týmu na pořádané akci. Z těchto organizačních důvodů trvá tento druh CTF několik hodin, případně dnů. Zároveň tento druh CTF je často pořádán pro zkušenější hráče.

1.2 Obory úkolů pro CTF

1.2.1 Všeobecné dovednosti

Úkoly cílí na všeobecné dovednosti a znalosti při práci s různými operačními systémy a práci s příkazovou řádkou. Rovněž tyto úkoly zahrnují schopnosti převádět mezi sebou jednotlivé číselné soustavy, nebo schopnosti pracovat a orientovat se v počítačové síti. Tyto úkoly zpravidla nevyžadují téměř žádné odborné znalosti a jsou tak vhodné pro začínající hráče.

1.2.2 Kryptografie

Tyto úkoly se zabývají kryptografickými problémy, aplikováním kryptografických algoritmů, lámáním šifer, šifrováním a dešifrováním, frekvenční analýzou šifrového textu, hledáním ustanoveného klíče některých kryptografických protokolů, lámáním hash, lámáním hesel apod. Úkoly vyžadují znalosti kryptografických algoritmů a protokolů, tudíž nemusí být vhodné pro laiky a hráče bez předešlých zkušeností. Často je třeba k řešení těchto úkolů použít různé programové nástroje a jiné zdroje jako například slovníky nebo rainbow tables (česky duhové tabulky, nadále bude v práci využíván jen anglický pojem, jelikož je v oboru užívanější)[2]. Tyto zdroje nemusí být vždy poskytnuty organizátorem, a tak je nucen hráč pracovat s vlastními či jemu dostupnými zdroji.

1.2.3 Digitální forenzní věda

Odvětví forenzní vědy, které se snaží o získání informace z dat v digitální podobě. Cílem je nalézt užitečnou informaci v přiložených datech, které informaci obsahují. K jejímu získání je však nutné provést analýzu dat. Do této kategorie lze zahrnout analýzy zachycených přenosů, analýzy logů událostí, hledání informace v metadatech souborů, analýza obrazové nebo zvukové stopy, obnova informace z poškozeného souboru atd. Tyto úkoly zpravidla nevyžadují rozsáhlé znalosti, ale spíše mají za cíl prověřit analytické myšlení hráče a jeho schopnost přistupovat k problému analytickým a systematickým přístupem. Analýza dat často vyžaduje použití různých programových nástrojů, které umožňují práci s daty a umožňují provedení analýzy, jelikož tato data se často vyskytují v nečitelné podobě.

1.2.4 Reverse engineering

Tento typ úkolů je zaměřen na prověření znalostí a schopností orientace ve zdrojových kódech, knihovnách nebo binárních souborech různým aplikací. Výsledkem tohoto procesu je nalezení tajemství, které bylo ve zdrojových kódech ukryto. Tyto úkoly vyžadují znalosti programovacích jazyků, programových struktur a mohou vyžadovat použití některých nástrojů např. nástroje pro dekompilaci programů.

1.2.5 Síťové služby

Úkoly se zaměřením na síťové služby mají prověřit hráčovy schopnosti a orientaci v síťovém prostředí. Hráč při řešení využívá především protokolů zaměřených na

[2]Rainbow tables je soubor předpočítaných hodnot hash funkce, který lze využít k rychlému a snadnému prolomení hash funkce a získání např. hesla nebo klíče

síťovou komunikaci a výměnu dat mezi zařízeními. Jmenovitě se tak nejčastěji jedná o protokoly HTTP, HTTPS, FTP, SSH, Telnet apod. Rovněž sem spadá také využívání zranitelností webových aplikací a síťových služeb, skrze které lze získat data ze serveru, která nejsou běžným způsobem přístupná. Takové úkoly zpravidla vyžadují spuštění a zpřístupnění pomocných serverů, které hostí zranitelnou aplikaci nebo službu. K vyřešení úkolu je často třeba přistoupit na živý server. Proto by měl organizátor zajistit identické podmínky pro všechny hráče a zabránit nedovolené manipulaci s pomocnými servery.

1.3 Platformy pro CTF

Platformou pro CTF se rozumí aplikace, nebo framework[3] pro vytváření a správu CTF. Slouží organizátorovi ke zjednodušení práce s vytvářením úkolů, správou uživatelů a jejich aktuálního pokroku v soutěži atd. Platformy lze rozdělit do dvou kategorií. Volně dostupné ke stažení, a platformy, které jsou poskytovány na internetu a organizátor si na této platformě vytvoří vlastní hru, kterou naplní úkoly a pořádá CTF na cizích zdrojích. Pro účel této práce byly porovnány tři platformy z hlediska jejich intuitivnosti při zprovoznění, jejich možnostech správy úkolů a uživatelů a jejich schopnosti obsloužit testované množství uživatelů jako webovou zátěž. Platforma pro tuto práci byly vybrány pouze platformy, které umožňují pořádání jeopardy druhu CTF.

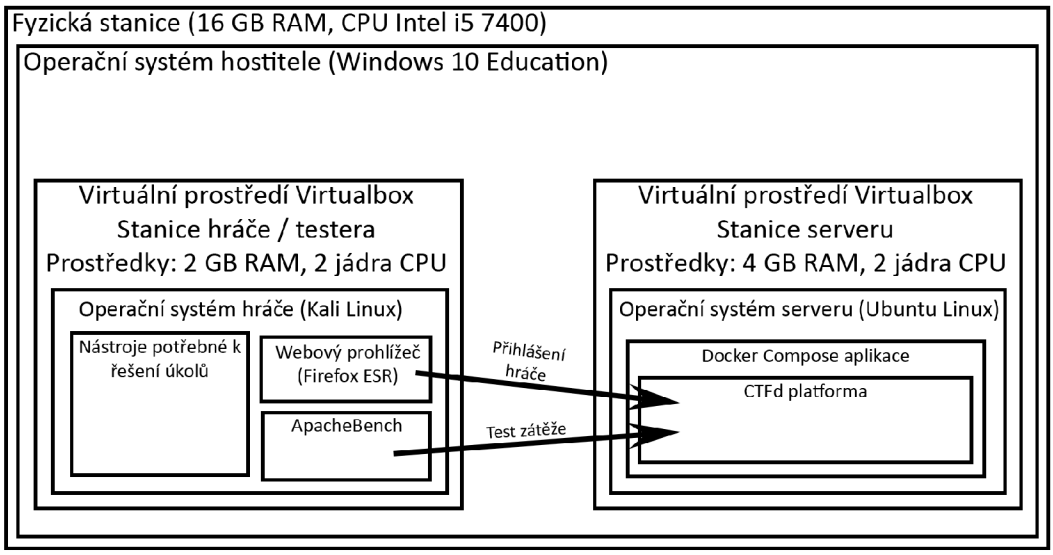
[3]Framework je část software, která poskytuje funkce, pro vývoj a správu jiného software projektu

2 Výběr platformy pro CTF

Pro praktickou část této práce bylo nutné vybrat vhodnou platformu, na které budou jednotlivé úkoly implementovány. Tato kapitola pojednává o třech vybraných platformách a jejich vhodnosti pro praktickou část této práce.

2.1 Server pro hostování CTF

Všechny platformy byly zprovozněny ve virtualizovaném[1] operačním systému Linux Ubuntu (verze 18.04). Platformy byly instalovány na čisté instalaci operačního systému s grafickým prostředím GNOME. Všem virtuálním strojům byly přiděleny stejné zdroje z hostitelského systému[2].



Obr. 2.1: **Schéma virtuálního prostředí** vytvořeného na fyzické stanici. Schéma znázorňuje jakým způsobem byl v práci proveden test zátěže a jak byly testovány úkoly. Pro ostatní platformy test probíhal analogicky s rozdílem aplikace a platformy v operačním systému serveru.

[1] Hostitelský systém pro virtualizaci Windows 10 Education. Virtualizační software Oracle VM VirtualBox 6.0

[2] Přidělené zdroje: 2 jádra CPU Intel i5-7400, 4 GB RAM

2.2 Vybrané platformy pro CTF

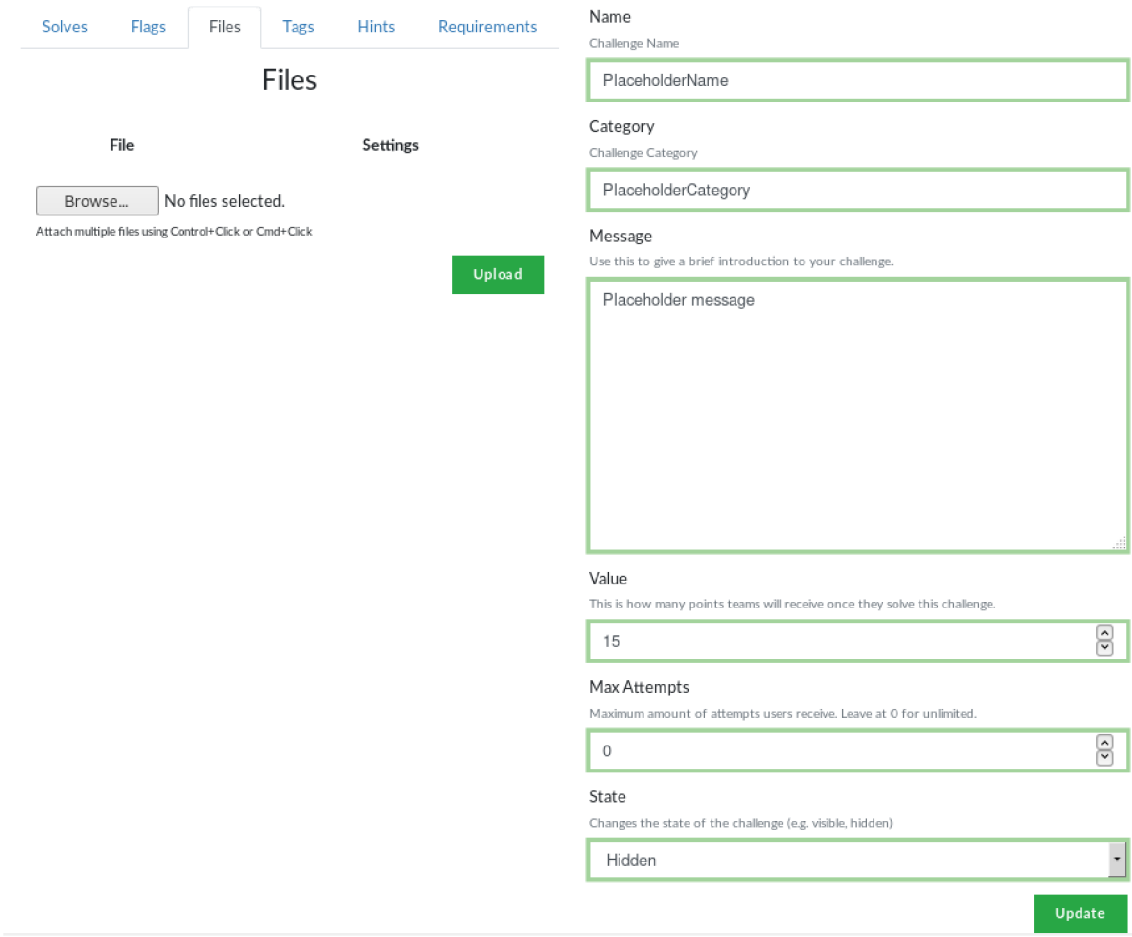
Pro účel této práce byly vybrány tři volně dostupné platformy, které umožňují organizovat CTF na vlastních zdrojích a jsou primárně zaměřeny na jeopardy druh CTF.

2.2.1 CTFd

CTFd je volně dostupný open source framework[3], pro vytvoření a organizování CTF her především druhu jeopardy. Jedná se o framework vývojáře Kevin Chung. Další vývoj a správu framework zastřešuje právnická osoba Major League Cyber[4]. Tento framework je poskytován volně ke stažení z: `https://github.com/CTFd`. Tento framework poskytuje možnost vytvářet úkoly, na které se odpovídá prostřednictvím kontextové nabídky, která je jedinečná, pro každý úkol. Na této kontextové nabídce je možné hráči zobrazit popisný text pro danou úlohu a přiložené soubory ke stažení a poskytnout mu nápovědy, které mohou být bodově penalizovány. Po odeslaní správné odpovědi je úkol označen jako vyřešený a hráči jsou uděleny body. Body jsou udělovány buď konstantní, nebo se mohou snižovat dle počtu řešení daného úkolu jinými hráči. Platforma rovněž umožňuje omezit počet pokusů každé individuální úlohy a úkoly hráči zpřístupňovat v závislosti na jeho aktuálním pokroku ve hře. Do této platformy je možné zakoupit různé rozšíření (plugins) a grafická témata (themes). Rozšířit platformu lze o úkoly s více správnými odpověďmi formou testu nebo o úkoly, které je možné manuálně kontrolovat. Manuálně kontrolované úlohy mohou zahrnovat delší textové odpovědi, jako například části skriptů apod. Placená grafické témata pak umožňují měnit vzhled uživatelského rozhraní a přizpůsobovat jej dle dostupných řešení.

[3]Webová stránka CTFd: `https://ctfd.io/`

[4]Webová stránka CTFd LLC Major League Cyber: `https://www.majorleaguecyber.org`



Obr. 2.2: **CTFd rozhraní vytváření úkolu**. V záložce „Flag“ je možné přidat odpovědi na úlohu a nastavit citlivost na velikost písma. Záložka „Files“ umožňuje k úkolu připnout soubory, které budou hráči zobrazeny spolu s popiskem úkolu. A záložka „Requirements“ nastavuje, které předešlé úkoly musí mít hráč splněné, aby se mu tento úkol zobrazil. Další obrázky rozhraní CTFd jsou v příloze C.

2.2.2 Facebook CTF (FBCTF)

FBCTF je platforma pro organizování a hostování her CTF. Platforma je licencována pod Creative Commons Attribution-NonCommercial 4.0 International licencí a je spravována a vyvíjena společností Facebook, Inc. Pod licencí dostupná ke stažení z: `https://github.com/facebook/fbctf`. Platforma kromě jeopardy druhu CTF umožňuje hostovat i „King of the Hill“ druh CTF[5]. Tato platforma je zaměřena primárně na soutěživost hráčů, jelikož každý úkol lze řešit jen jednou. Platforma

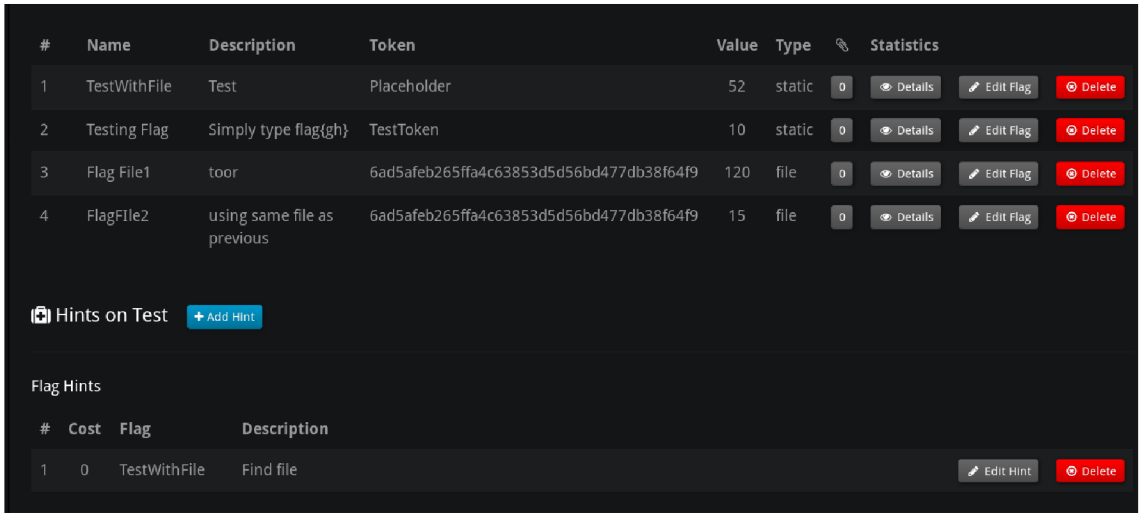
[5]King of the Hill (česky král kopce) je zvláštní druh CTF her, kdy hráči vzájemně soutěží o vybudování silné pozice na daném stroji. První hráč stroj kompromituje a snaží se jej ochránit před ostatními hráči. Po pravidelnou dobu, kdy hráč stroj drží ve svém vlastnictví, dostává body.

v rozhraní používá mapu světa a jednotlivým státům lze přidělit úkoly. Jakmile hráč úkol vyřeší získá body a obsadí území, které již další hráči obsadit nemohou. Pouze území, které jsou vybrané do části „King of the Hill“ je možné vzít jinému hráči. Platforma umožňuje hráčům využívat nápovědy, které mohou být bodově penalizovány. Dále platforma umožňuje k úkolům připnout soubory a pro část „King of the Hill“ i specifikovat IP adresu stroje, který mají hráči napadat. Platforma umožňuje vytvářet týmy několika hráčů a společně se podílet na řešení úkolů.

Obr. 2.3: **Rozhraní hráče FBCTF**. Hráč vidí mapu světa s hranicemi jednotlivých států. Státy představují území, které hráč získá za splnění daného úkolu. Vyřešení úkolu je odměněno body a území se pro ostatní hráče stane nedostupné. Další obrázky rozhraní FBCTF jsou v příloze D.

2.2.3 Root the Box (RTB)

Root the Box je volně dostupná platforma pro vytvoření a hostování her CTF. Platforma byla vytvořena v roce 2012 a je stále aktualizována komunitou na GitHub[6]. Platforma umožňuje vytvářet několik druhů úkolů pro jeopardy druh CTF. Dále umožňuje vytvořené úkoly řadit do úrovní a vytvářet tak velmi intuitivní prostředí pro hráče, kteří nejsou příliš zkušení a v průběhu hraní se zlepšují a tím si otevírají nové úrovně úkolů. Kromě této možnosti má platforma zabudovanou podporu pro vytváření botnets[7]. Platforma má rovněž zabudovány mechanizmy pro vzájemné napadání a sabotáže jednotlivých týmů, takže lze vytvořit velice soutěživé prostředí.



Obr. 2.4: **Rozhraní administrátora RTB**. V tomto rozhraní může administrátor vytvářet jednotlivé úkoly a nápovědy k úkolům. Další obrázky rozhraní RTB jsou v příloze E.

[6]GitHub RTB: `https://github.com/moloch--/RootTheBox`

[7]Botnet je skupina zotročených zařízení, které jsou připojeny k síti a řízeny centrální stanicí (CnC – Command and Control). Tyto zařízení lze použít například k provedení DDoS (Distributed Denial–of–Service) útoků. Jedná se tak o síť pro distribuované útoky, které neprobíhá pouze z jedné stanice.

2.3 Porovnání platforem

Jednotlivé platformy byly porovnány na základě variability, kterou je možno využít pro vytvoření úloh, které jsou v této práci popsány. Dále na základě intuitivnosti při vytváření a řešení úloh, jednoduchosti a přehlednosti konfigurace CTF her, hardware nároků a schopnosti obsloužit dostatečné množství uživatelů v přijatelném čase. K testování byl použit program *ApacheBench (ab)*, který je určen pro měření a testování výkonu webových serverů. Pro testování pomocí *ApacheBench* byl použit příkaz pro otestování dostupnosti celkem 5000 uživatelů při zatížení 200 uživatelů najednou.

```
root@kali:~# ab -c 200 -n 5000 -r "adresaServeru:port"
   >> "nazevSouboruProUlozeni"
```

Výstupy testů byly uloženy do souborů jedinečných pro každou testovanou platformu. Výpis těchto souborů je dostupný v příloze F. Hardware nároky byly zhodnoceny na základě informací, které byly poskytnuty buď vývojářem, nebo na základě hardware nároků služby, pod kterou platforma v systému běží.

Tab. 2.1: Porovnání platforem na základě požadavků a zátěžovém testu programu *ApacheBench*

Platforma	**Hostující služba (Daemon)**	**Minimální HW a SW požadavky**	**Čas obsloužení 50% uživatelů**	**Čas obsloužení 90% uživatelů**	**Slovní popis**
CTFd	Docker	2 GB RAM, single-thread, multiplatformní s podporou Docker	1390 ms	1801 ms	Platforma má velmi intuitivní a snadné vytváření a správu úkolů. Úkoly lze jednoduše kategorizovat a zpřístupňovat na základě předešle vyřešených úkolů. Zviditelnění úkolů je nutné provést u každého úkolu zvlášť.
Facebook CTF	HHVM	3 GB RAM, multi-thread, Ubuntu 16.04 x64 a vyšší	1605 ms	1740 ms	Platforma má animované uživatelské rozhraní a jsou znatelná zpoždění, mezi uživatelskými akcemi. Nastavení úkolů a jejich zviditelnění hráčům není příliš intuitivní. Vytváření úkolů není příliš variabilní.
Root the Box	Tornado	2 GB RAM, single-thread, Ubuntu nebo Debian, Python 2.7. nebo Python 3.6, PyPy	324 ms	388 ms	Platforma má komplexní a rozsáhlé rozdělení pro úkoly. Platforma obsahuje plnou podporu pro hraní jednotlivců i týmů. Úkolům nelze přímo připnout soubory, ale je možné soubory prostřednictvím webu zveřejnit všem.

2.3.1 Vybraná platforma pro praktickou část

Pro tuto práci byla vybrána platforma CTFd. CTFd lze velice jednoduše spustit a nakonfigurovat. Lze vytvářet úkoly a řadit je do kategorií. Jednotlivým úkolům lze přiřazovat soubory, které má hráč k dispozici ke stažení z rozhraní řešeného úkolu. Úkoly na sebe lze vázat a vytvářet závislosti mezi nimi. CTFd lze spustit na jakémkoliv zařízení, které podporuje provoz prostředí Docker[8]. Na základě povahy úkolů byly, vytvořeny celkem dvě identické platformy CTFd. Liší se pouze úkoly, které jsou v platformách implementovány. Rozdělení tak umožní hostovat hry s úkoly pro

[8]Docker je open source software, který poskytuje rozhraní pro izolaci aplikací do kontejnerů. Jedná se tak o virtualizaci na úrovni operačního systému.

podporu výuky předmětů **Základy kryptografie** a **Bezpečnost ICT 1 a 2** odděleně. FBCTF nebyla vybrána z důvodu malé variability úkolů a hlavně z důvodu záseků a nepříjemné práce s rozhraním kvůli animacím. RTB platforma má velmi podobné vlastnosti jako vybrané CTFd. Pro CTF většího rozsahu by bylo lepší vybrat platformu RTB, jelikož lze úkoly lépe rozřadit do kategorií a úrovní obtížnosti. Zároveň lze lépe úkoly zviditelňovat hráči. Chybí však možnost jednotlivým úkolům přiřazovat soubory, které mohou být potřebné k vyřešení daného úkolu.

2.4 Stanice hráče CTF

Vlastní řešení úkolů CTF her zpravidla neprobíhá přímo ve webovém prostředí a je nutné použít různé nástroje a programy na stanici hráče. Z toho důvodu je hráči doporučenou použít platformu s operačním systémem Kali Linux. Kali Linux má mnoho nástrojů a programů, které se využívají k řešení těchto úkolů již předinstalované. Zároveň lze s využitím repositářů potřebný software doinstalovat. Pro testování řešení úkolů v této práci byl použit operační systém Kali Linux ve verzi 2019.3.

2.5 Pomocné stanice pro síťové úkoly

Povaha některých úkolů vyžaduje použití pomocných virtuálních strojů, které jsou živě spuštěny a zařazeny do síťové topologie, aby k nim hráči měli přístup. Tyto pomocné virtuální stroje hrají roli serverů, na kterých jsou spuštěny služby, se kterými hráč při řešení úkolu pracuje. Zpravidla se tak jedná o webové servery, jejichž prostřednictvím hráč pracuje se servery a hledá odpověď, nebo operační systémy, jejichž zranitelností má hráč využít a prolomit se přímo do systému, ve kterém hledá odpovědi. Alternativou spuštění jednoho společného serveru, je distribuce virtuálních disků všem hráčům, kteří si následně virtuální stroj spustí na své pracovní stanici. Toto řešení však může být problematické. Virtuální disky mají velikosti v řádech GB a jejich distribuce všem hráčům může být obtížná. Výhodou této alternativy je fakt, že každý hráč má potřebné zdroje virtualizované lokálně na své stanici, a tudíž nedochází k vzájemnému ovlivňování hráčů při práci na společném stroji.

3 Implementované úkoly BZKR

Tato kapitola popisuje, které úlohy z jednotlivých kategorií byly implementovány do praktické části platformy CTF pro předmět BZKR. Jednotlivé úlohy jsou popsány teoreticky a je stručně popsána jejich problematika. Rovněž je částečně vysvětleno řešení, kterým lze jednotlivé úkoly splnit. Konkrétní úlohy, hodnoty a jejich řešení v této kapitole ani textové části této práce uvedeny nejsou.

Obr. 3.1: **Záložky úkolů pro předmět BZKR**, některé úkoly jsou skryté, kvůli jejich návaznosti na předešlé úkoly. Tyto skryté úkoly se hráči zobrazí po dokončení příslušných úkolů. Seznam všech úloh v rozhraní organizátora CTF v příloze G.1

3.1 Všeobecné znalosti

3.1.1 Převody soustav

Jedná se o úkoly všeobecných znalostí, které pokrývají problematiku převodu hodnot mezi jednotlivými číselnými soustavami o různých základech. Vyjádření do ASCII

dle obrázku A.1

Tyto úkoly jsou zahrnuty v platformě pro BZKR, i v platformě pro TIC. Jelikož tento druh úkolů má hráče seznámit s fungováním CTF, jsou úkoly s různými hodnotami společné pro obě platformy.

Dekadické vyjádření ASCII

Libovolný text s odpovědí byl vyjádřen v dekadické podobě na základě ASCII[1] znaků, ze kterých byl složen. K vyřešení úlohy je nutné dekadické hodnoty vyjádřit jako ASCII znaky, text přečíst a odeslat odpověď, která je zmíněná v textu.

Binární vyjádření ASCII

Libovolný text s odpovědí byl vyjádřen v binární podobě na základě ASCII znaků, ze kterých byl složen. K vyřešení úlohy je nutné binární hodnoty vyjádřit jako ASCII znaky, text přečíst a odeslat odpověď, která je zmíněná v textu.

Hexadecimální vyjádření ASCII

Libovolný text s odpovědí byl vyjádřen v hexadecimální podobě na základě ASCII znaků, ze kterých byl složen. K vyřešení úlohy je nutné hexadecimální hodnoty vyjádřit jako ASCII znaky, text přečíst a odeslat odpověď, která je zmíněná v textu.

Oktalové vyjádření ASCII

Libovolný text s odpovědí byl vyjádřen v oktalové podobě na základě ASCII znaků, ze kterých byl složen. K vyřešení úlohy je nutné oktalové hodnoty vyjádřit jako ASCII znaky, text přečíst a odeslat odpověď, která je zmíněná v textu.

Base64 vyjádření ASCII

Libovolný text s odpovědí byl vyjádřen v číselné soustavě base64 na základě ASCII znaků, ze kterých byl složen. K vyřešení úlohy je nutné hodnoty číselné soustavy base64 přepočítat a vyjádřit text jako ASCII znaky. Text přečíst a odeslat odpověď, která je zmíněná v textu[2].

[1] ASCII - American Standard Code for Information Interchange

[2] Base64 je binárně textové kódovací schéma, které pro kódování používá soustavu o 64 tisknutelných znacích ASCII tabulky. Toto kódování je popsáno v RFC 4648[1]

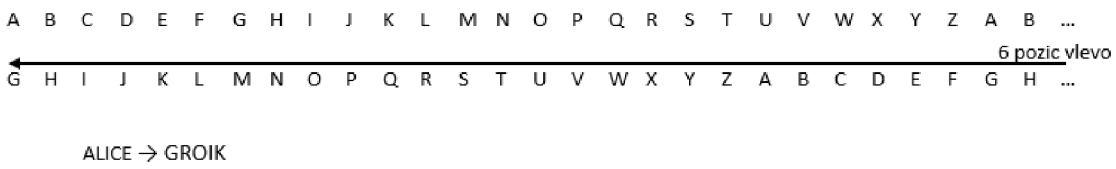
3.2 Kryptografie

3.2.1 Substituční šifry

Jedná se o druh kryptografické šifry, při kterém se zaměňují znaky jedné sady za znaky sady jiné. Typicky se jedná o abecedy posunuté o daný počet pozic. Písmena otevřeného textu se následně zamění za písmena jiné abecedy a vznikne šifrový text. V moderní kryptografii již nelze samotné substituční šifry považovat za bezpečné s výjimkou Vernamovy šifry. Substituce se přesto používá i v moderních šifrách k zajištění konfúze šifry a to nejčastěji v podobě S-box.

Caesarova šifra

Caesarova šifra je mono-alfabetická substituční šifra. Původně se jednalo o rotaci šifrové abecedy o tři pozice, poté byla zobecněna na libovolnou rotaci šifrové abecedy. Šifruje se po znacích a každý znak šifrového textu má dle posunu dán znak šifrového textu. Všechny znaky se šifrují stejnou šifrovou abecedou. Dešifrování je analogické k šifrování, pouze se provádí posun o stejný počet znaků na opačnou stranu.



Obr. 3.2: Příklad **Caesarovy šifry** s posunem o šest znaků.

.

Nejjednodušší způsob útoku na Caesarovu šifru je hrubou silou[3] provést všechny možné rotace abeced nad šifrovým textem a jediný smysluplný výsledek je otevřený text. V běžném případě to znamená provést 26 rotací podle znaků anglické abecedy. Jelikož stejné znaky otevřeného textu jsou šifrovány jako stejné znaky šifrového textu, šifrový text si tak ponechává všechny vlastnosti otevřeného textu. Zůstává zachována délka zprávy, často délka slov, frekvenční výskyt znaků v jazyce zůstává zachován, takže lze útočit pomocí frekvenční analýzy textu.

V praktické části této práce je použita Caesarova šifra českého textu, který byl vytvořen pro tuto úlohu. V textu je uvedena odpověď na danou otázku.

[3]Útok hrubou silou - anglicky brute-force attack

Vigenérova šifra

Vigenérova šifra je poly-alfabetická substituční šifra. Princip šifrování je použití 26 šifrových abeced, pokud se jedná o šifrování v anglické abecedě. Podobně jako u Caesarovy šifry se šifrují jednotlivé znaky otevřeného textu. Ale na rozdíl od Caesarovy šifry se nepoužívá pouze jediná posunutá abeceda, ale všech 26 rotovaných abeced v závislosti na klíči, který již není pouhé číslo rotace, ale zvolený řetězec. Znak klíče určí, kterou abecedu použijeme pro šifrování znaku otevřeného textu. Klíč zpravidla bývá kratší než zpráva a tak se jednoduše opakuje do doby, než jsou znaky otevřeného textu zašifrovány. Pro lepší představu je v příloze uveden obrázek tabulky B.1.

Na Vigenérovu šifru se nejčastěji útočí pomocí frekvenční analýzy textu. Jelikož klíč nebývá stejně dlouhý jako zpráva a dochází k jeho opakování, pak dochází k tomu, že stejné znaky otevřeného textu jsou šifrovány stejným znakem klíče. Tyto anomálie se v textu opakují po násobcích délky klíče. Analýzou těchto anomálií lze odhadnout délku klíče. Poté lze šifrový text rozdělit po znacích do k-skupin, kde k je délka klíče. V těchto skupinách pak provádíme frekvenční analýzy znaků, které odpovídají frekvenci znaků v jazyce, ve kterém byl text šifrován. Tuto metodu lze vylepšit frekvenční analýzou n-tic znaků. A analýzou znaků, kterými běžně v jazyce slova začínají nebo končí, nebo které se vyskytují po dvojicích vedle sebe. Na základě těchto znalostí můžeme odhadnout pravděpodobné kandidáty klíče a postupně jimi dešifrovat šifrový text, dokud nenalezneme otevřený text.

Pro praktickou část této práce byl vybrán originální text z díla Jáma a kyvadlo amerického spisovatele Edgara Allana Poea. Tento text je dostatečně dlouhý a výskyt znaků přibližně odpovídá frekvenční analýze znaků anglického jazyka.

3.2.2 Teorie čísel

Teorie čísel je odvětví matematiky, které se zaměřuje na pozorování vlastností celých čísel a celočíselných funkcí. Toto odvětví co prolíná téměř do každého odvětví matematiky a pokrývá problematiku běžné i modulární aritmetiky, prvočísel a mnoho dalšího. Spolu s geometrií, algebrou a matematickou analýzou tvoří největší pilíře moderní matematiky. Vzhledem k faktu, že moderní kryptografické systémy jsou založeny na matematických problémech, je teorie čísel obor, který je v moderní kryptografii hojně využíván.

Příkladem může být Eulerova funkce celého čísla:

$$\begin{gathered}\Phi(n) = (p_1 - 1)p_1^{(m_1-1)}(p_2 - 1)p_2^{(m_2-1)} \dots p_k^{(m_k-1)}, \\ \text{kde } p_i^{m_i} \text{ jsou prvky kanonického rozkladu } n\end{gathered} \tag{3.1}$$

Pro tuto práci bylo implementováno několik úloh, které svou podstatou spadají do teorie čísel. Jmenovitě se jedná o úlohy počítání Eulerovy funkce, hledání největšího společného dělitele, prvočíselné testy, rozklady složených čísel na prvočíselné dělitele, nebo použití Rozšířeného Euklidova algoritmu. Celkem praktická část obsahuje 7 úkolů, které se těmito problémy zabývají.

3.2.3 Modulární aritmetika

Modulární aritmetika je odvětví matematiky zabývající se především operací modulo (tj. zbytek po celočíselném dělení). Tento druh operací se používá v celé řadě kryptografických algoritmů, kde tyto operace zajišťují matematickou bezpečnost.

V praktické části této práce je zahrnuto několik úloh, které se zabývají problémy modulární aritmetiky. Příklad takové úlohy je výpočet modulárně multiplikativního inverzního prvku:

$$a \cdot a^{-1} \equiv 1 \pmod{n} \tag{3.2}$$

Tato úloha je přípravou na řešení úlohy RSA problému. Rovněž je implementována úloha pro nalezení modulárně aditivního inverzního prvku. Dále lze jmenovat úlohy, které jsou zaměřeny na řešení kongruencí (modulárních rovnic), nebo třeba modulární mocnění. Lze sem tedy zařadit i 2 implementované úkoly, které vyžadují řešení soustavy kongruencí pomocí Čínské věty o zbytcích. Celkem je implementováno 7 úkolů, které se v různých ohledech dotýkají problémů modulární aritmetiky.

3.2.4 Teorie grup

Teorie grup je součástí algebry. Tato část abstraktní algebry je zaměřená na algebraické struktury. V kryptografii se v některých případech, zejména při práci s eliptickými křivkami, asymetrickými šiframi a protokolem Diffie-Hellman pracuje s grupami. Často jsou grupy využívány jako množiny, ze kterých jsou vybírány parametry pro daný protokol.

V práci se zahrnuto několik úkolů, které vyžadují práci s grupami. Jedním z úkolů je nalezení generátoru cyklické grupy. Generátor je prvek cyklické grupy, který pokud budeme postupně mocnit celými čísly, výsledky nám vygenerují všechny prvky v této cyklické grupě. Další úkol, který se dotýká problematiky teorie grup je Diffie-Hellman.

3.2.5 RSA problém

RSA (Rivest-Shamir-Adleman) je asymetrický kryptosystém používaný k šifrování symetrických klíčů a k podepisování. RSA problém je kryptografickým problém o+získání soukromého klíče RSA ze znalosti veřejného klíče RSA. Pokud lze pro

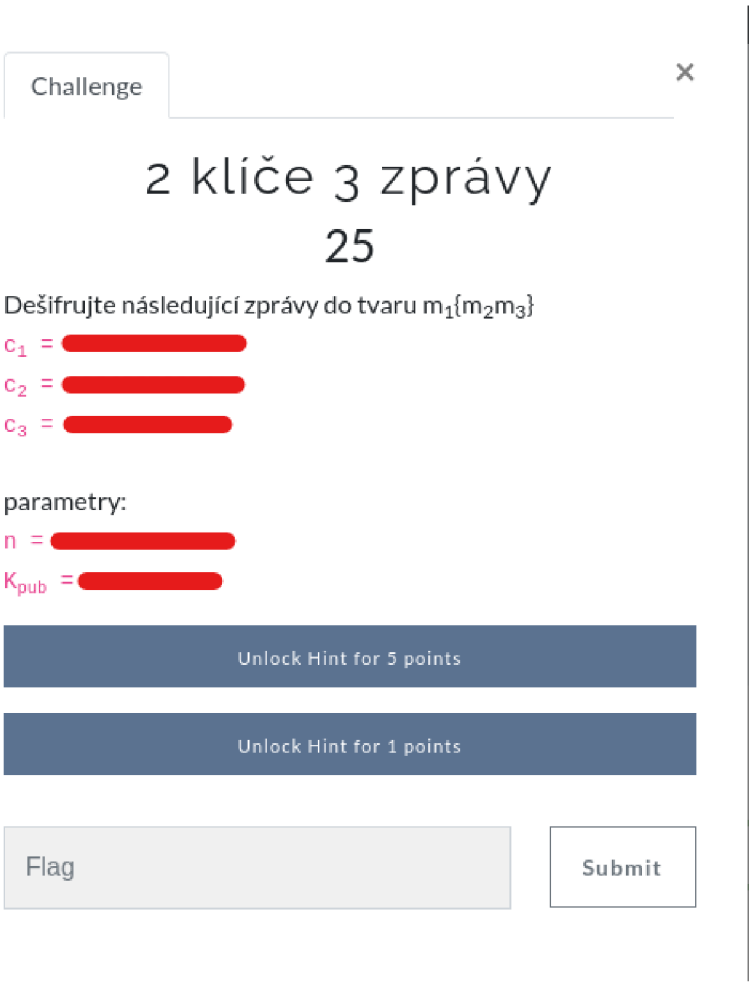
daný kryptosystém používající RSA vyřešit RSA problém, je možné systém kompromitovat. Klíčové pro vyřešení RSA problému je schopnost faktorizace (rozložení na prvočíselné dělitele) čísla n, které v RSA vzniká vynásobením dvou dostatečně velkých a bezpečných prvočísel $p, q > 2^{1024}$.

Vztah klíčů v RSA je následující:

$$K_{priv} = K_{pub}^{-1} \pmod{\Phi(n)}, \text{ kde } \Phi(n) \text{ je Eulerova funkce čísla } n \tag{3.3}$$

V dnešní době není znám algoritmus, který umožňuje spočítat v polynomiálním čase Eulerovu funkci čísla n, bez znalosti jeho prvočíselných dělitelů p, q. Rovněž není znám algoritmus jak číslo n v polynomiálním čase faktorizovat a nalézt jeho prvočíselné dělitele p, q. Pokud jsou tak prvočísla p, q dostatečně velká, může trvat řešení RSA problému stovky až tisíce let. Více o RSA v doporučení RFC 8017[2].

V praktické části této práce jsou hráčům dány pouze veřejné parametry (K_{pub}, n) kryptosystému RSA a je vyžadováno řešení problému RSA. Parametry jsou uměle a záměrně oslabeny, aby bylo možné problém RSA vyřešit v přijatelném čase. Další varianta tohoto úkolu navíc obsahuje zprávu, která je pomocí RSA zašifrovaná. Úkolem hráče je nalézt K_{priv} a dešifrovat zprávu.



Obr. 3.3: **Rozhraní úkolu RSA obsahující zprávu**. V tomto rozhraní hráč vidí všechny potřebné parametry ke splnění úkolu. Hráč má možnost využít bodově penalizované nápovědy k usnadnění řešení.

3.2.6 Diffie-Hellman protokol

Diffie-Hellman protokol v kryptografii slouží k vzájemnému ustanovování klíčů symetrické kryptografie přes potenciálně nebezpečné přenosové médium. DH protokol využívá obtížnosti řešení diskrétního logaritmu. Přes veřejné médium je sdíleno velké číslo n a číslo g, pro které platí:

$$\mathrm{GCD}(n,\ g) = 1 \tag{3.4}$$

Obě strany si vyberou dostatečně velké číslo a resp. b z intervalu $< 0,\ n >$. Poté obě strany provedou operaci:

$$g^a (\mathrm{mod}\ n) = A \text{ resp. } g^b (\mathrm{mod}\ n) = B \tag{3.5}$$

Výsledky si strany vzájemně vymění. Poté provedou další operaci pro výpočet společného symetrického klíče:

$$K = A^b (\text{mod } n) \text{ resp. } K = B^a (\text{mod } n) \tag{3.6}$$

Tento vztah lze dle vzorce 3.5 rozepsat jako:

$$K = g^{a^b} (\text{mod } n) \text{ resp. } K = g^{b^a} (\text{mod } n) \tag{3.7}$$

Tímto je ustanoven společný symetrický klíč, který není možné pouze z veřejných parametrů g, n, A, B získat. K získání symetrického klíče tímto způsobem je třeba vyřešit problém diskrétního logaritmu:

$$a = \log_g A \ (\text{mod } n) \text{ resp. } b = \log_g B \ (\text{mod } n) \tag{3.8}$$

Na řešení tohoto problému není znám algoritmus, který by poskytl řešení v polynomiálním čase pro velká čísla n ($n > 2^{2048}$). Více o DH protokolu v doporučeních RFC 2631[3] a RFC 3526[4].

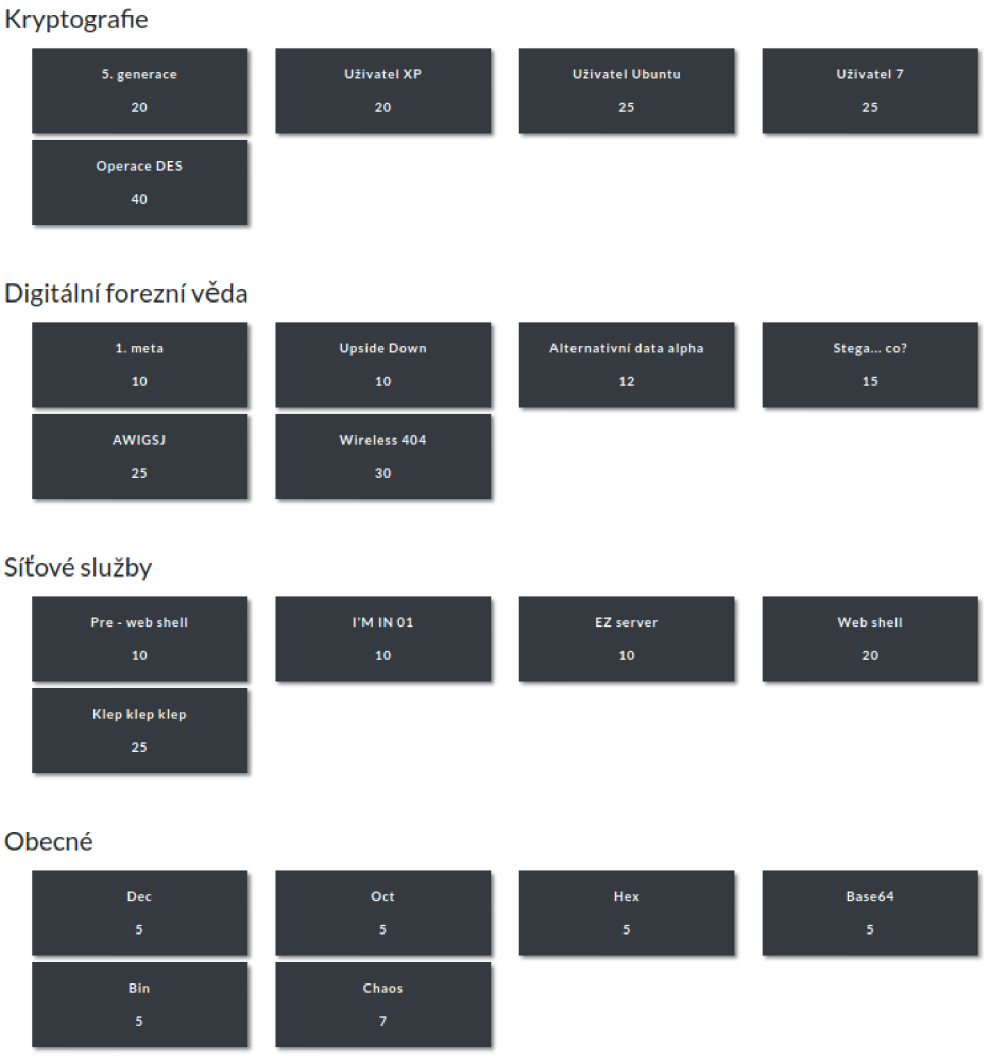
V praktické části této práce jsou hráčům poskytnuty veřejné parametry g, n, A, B. Velikost čísla n je uměle a záměrně omezena, aby bylo možné vyřešit diskrétní logaritmus a nalézt ustanovený symetrický klíč. Druhý úkol této kategorie navíc obsahuje zprávu, která byla zašifrována bitovou operací XOR s ustanoveným klíčem K.

Obr. 3.4: **Rozhraní úkolu DH**. V tomto rozhraní hráč vidí všechny potřebné parametry ke splnění úkolu. Hráč má možnost využít bodově penalizované nápovědy k usnadnění řešení.

4 Implementované úkoly TIC 1 a 2

Tato kapitola popisuje, které úlohy z jednotlivých kategorií byly implementovány do praktické části platformy CTF pro předmět TIC 1 a 2. Jednotlivé úlohy jsou popsány teoreticky a je stručně popsána jejich problematika. Rovněž je částečně vysvětleno řešení, kterým lze jednotlivé úkoly splnit. Konkrétní úlohy, hodnoty a jejich řešení v této kapitole ani textové části této práce uvedeny nejsou.

Nejsou zde znovu popsány úkoly s převáděním soustav, které byly popsány v kapitole 3.1.1. Úkoly jsou však v platformě implementovány také.



Obr. 4.1: **Záložky úkolů pro předmět TIC**, některé úkoly jsou skryté, kvůli jejich návaznosti na předešlé úkoly. Tyto skryté úkoly se hráči zobrazí po dokončení příslušných úkolů. Seznam všech úloh v rozhraní organizátora CTF v příloze G.2

4.1 Všeobecné znalosti

4.1.1 Práce s příkazovou řádkou

Do souboru „chaos.file“, který obsahuje náhodné znaky a řetězce byla ukryta odpověď začínající slovem „FLAG“. Cílem je v souboru odpověď nalézt. K tomu lze použít následující příkaz terminálu:

```
root@kali:~# strings chaos.file | grep FLAG
```

4.2 Kryptografie

4.2.1 Lámání hash funkcí

Hash funkce je jakákoliv funkce, která může být použita k převedení dat libovolné délky na data konstantní délky. Z principu se jedná o jednocestný proces, který je deterministický. Jedná se tak o jakýsi otisk (anglicky fingerprint) originálních dat, kterým data identifikujeme, ale nezískáme z něj užitečné informace, které data nesou. Pro hash funkci je klíčové také poskytovat bezkolizní výsledky. Je nežádoucí, aby různá data měla stejné otisky a tím byla znemožněna identifikace. Úplnou bezkoliznost je však nemožné teoreticky poskytnout ve všech případech, jelikož výstup hash funkce je vždy konstantní délky nezávisle na velikosti vstupu. Proto pro každá vstupní data existuje nekonečný počet jiných vstupních dat různé délky, které poskytnou stejný výsledek hash funkce. Počet kombinací výstupu omezeného délkou nemůže vždy poskytnout různé výsledky pro libovolný vstup různé délky, proto vždy bude existovat nekonečné množství kolizí pro vstupní data. Hash funkce jsou navrženy tak, aby bylo velmi obtížné tyto kolize nalézt, čímž je zajištěna presumpce bezkoliznosti.

Výsledkem procesu lámání hash funkce je buď nalezení originálních vstupních dat, nebo nalezení takových dat, které vytvoří kolizi s daty původními. Při nalezení kolize však kolidující data již nemusí nést stejnou informaci, jako data původní. Proto je žádanější nalezení dat původních a tím i získání informace, kterou data nesla. Hash funkce jsou z principu jednocestné a tak není možné proces hash funkce zvrátit. Je však možné zkoušet různé vstupy hash funkce a hledat shodu výsledku. Takto nalezená shoda je pak s vysokou pravděpodobností původní vstup do hash funkce. Další možností je využít již soubor, který obsahuje předpočítané výsledky hash funkce pro jednotlivé vstupy. V tomto souboru pak velice rychle nalezneme vstupní hodnotu hash funkce. Takto předpočítaný soubor se nazývá rainbow table. Jako ochrana proti útoku použitím rainbow tables se používá tzv. sůl. Sůl je v tomto případě generovaná hodnota, která je přidána k původním datům na vstupu hash

funkce. Výsledek hash funkce je pak otisk dat a přidané soli. Sůl je uváděna spolu s výsledkem hash funkce, aby bylo možné znovu opakovat hash funkci se stejným výsledkem. Přidáním soli k původním datům znemožníme provedení útoku pomocí rainbow table, jelikož není možné předpočítat všechny možné výsledky pro data, ke kterým je přidána generovaná sůl.

Prolomení MD5 hash

MD5 (Message-digest algorithm 5) je široce používaná hash funkce, která nahradila předchozí MD4 hash funkci. Výstupem z MD5 funkce je 128-bit hodnota. Bližší popis o algoritmu MD5 lze nalézt v doporučení RFC 1321[5].

V praktické části této práce je hráči dán MD5 hash řetězce např. hesla. Cílem hráče je nalézt vstupní řetězec MD5 hash funkce. Vstupní řetězec byl vybrán ze slovníku *rockyou.txt*, který je dostupný v běžné instalaci Kali Linux. Tím, že je řetězec obsažen v tomto slovníku je možné provést slovníkový útok s pomocí nástroje *hashcat*, který se používán pro lámání hash funkcí.

Prolomení LM hash

LM (LAN Manager) je autentizační protokol, který používá stejnojmenou hash funkci. LM hash je zastaralá a prolomená funkce hash, která byla používána v operačních systémech Microsoft Windows. Do operačního systému Windows Vista byla v základu tato hash funkce používána k vytváření hash hesel uživatelů. Nejedná se o plnohodnotnou funkci hash, jelikož je založena na šifrování pomocí algoritmu DES[1]. Délka hesla byla omezena na 14 znaků. Písmena v hesle byly všechny převedeny na písmena velké abecedy a rozděleno na dvě poloviny. Délka hesla byla případně doplněna nulami do délky 14 znaků. Tyto dvě poloviny po sedmi znacích měly délku 56-bit každá. Obě poloviny byly použity jako klíče algoritmu DES k zašifrování konstantní hodnoty. Výsledek DES byl zpětně spojen dohromady a vytvořil 128-bit hodnotu LM hash. LM hash zásadně porušuje bezpečnostní zásady pro vytváření hash hesel. Porušením zásad bezpečného uložení a hash funkce hesla je rozdělení na dvě poloviny, používání pouze znaků velké abecedy, omezení délky, ověřování polovin odděleně a použití nevhodného algoritmu jako hash funkce. Takto vytvořené hash hesel lze s pomocí rainbow tables prolomit v řádech maximálně minut.

V praktické části této práce je hráči předán soubor, který obsahuje přihlašovací údaje uživatele v operačním systému Windows XP. Za běžných okolností je třeba tento soubor nejprve získat z operačního systému. Tuto činnost hráč provádět nemusí

[1]DES – Data Encryption Standard je algoritmus symetrické kryptografie používající 56-bit klíč a výsledkem jsou bloky šifrového textu po 64-bit

a je mu předán rovnou soubor s údaji. Jeho úkolem je prolomit LM hash funkci a nalézt tak uživatelovo heslo.

Prolomení NTLM autentizačního protokolu

NTLM (New Technology LAN Manager) je novější protokol v operačních systémech Microsoft Windows. V tomto protokolu byla nahrazena zastaralá a nebezpečná funkce hash LM. V tomto protokolu jsou používány MD4 a MD5 hash v závislosti na verzi a konfiguraci. Více o tomto autentizačním protokolu v dokumentaci Microsoft–NLMP[6].

V praktické části této práce je hráči předán soubor, který obsahuje přihlašovací údaje uživatele v operačním systému Windows 7. Za běžných podmínek je třeba tento soubor nejprve získat z operačního systému. Tuto činnost hráč provádět nemusí a je mu předán rovnou soubor s údaji. Jeho úkolem je prolomit protokol NTLM a získat heslo uživatele.

Prolomení SHA512 s 8 byte solí

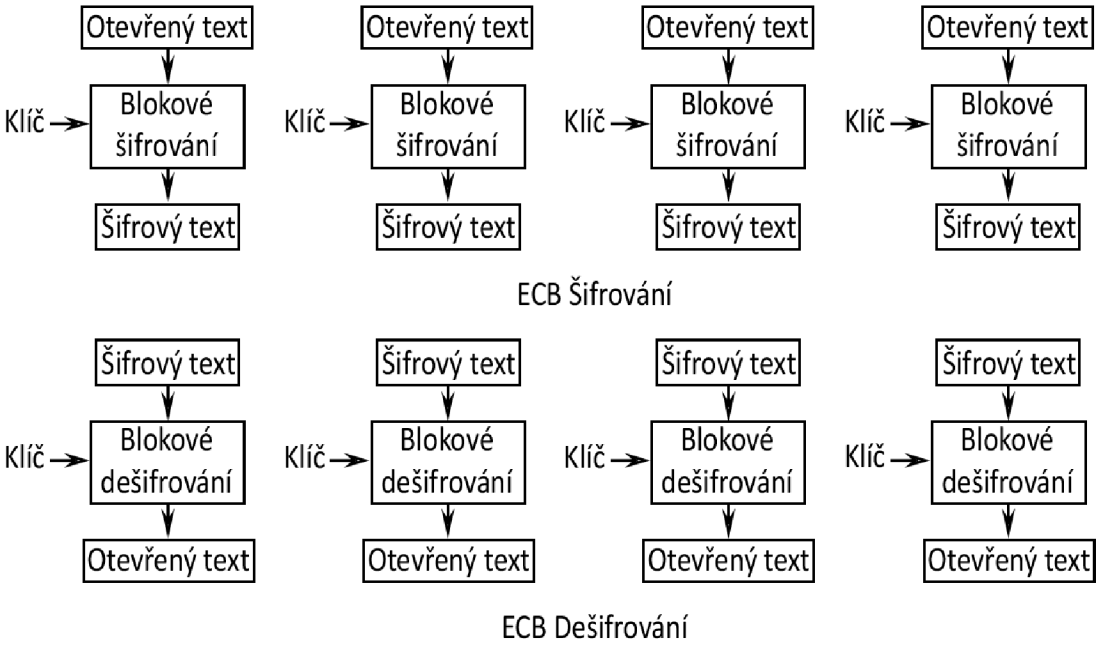
SHA (Secure Hash Algorithms) je skupina hash funkcí publikována americkým National Institute of Strandards and Technology (NIST) ve spolupráci s americkou National Security Agency (NSA) a dalšími subjekty. Tato skupina se dále dělí na podskupiny SHA–0, SHA–1, SHA–2, SHA–3. Tyto podskupiny se dále dělí na konkrétní varianty hash funkcí. Hash funkce skupiny SHA jsou pravděpodobně celosvětově nejpoužívanější hash funkce současnosti. Operační systémy Linux používají hash funkce ze skupiny SHA s přidáním soli pro ukládání hesel. Konkrétní nastavení lze specifikovat přímo v systému.

V praktické části této práce byl použit operační systém Ubuntu (18.04) ze kterého byly extrahovány soubory obsahující hash hesla uživatele operačního systému. Použitý hash je podskupiny SHA–2 SHA–512 s přidáním 8 byte soli. V souboru je obsaženo uživatelské jméno, 8 byte sůl a hash hesla. Dále jsou zde údaje o uživateli, jeho zařazení do skupin apod. Po hráči je vyžadováno hash prolomit a získat heslo uživatele. K vyřešení tohoto úkolu je třeba vytvořit vlastní slovník pomocí nástroje *Crunch* obsahující všechny číslice a malá i velká písmena jména uživatele. Aby bylo možné heslo prolomit v přijatelném čase obsahuje pouze 5 znaků. O parametrech pro správné vytvoření slovníku je hráč informován. S pomocí nástroje *John the Ripper* je poté možno hash funkci prolomit a získat heslo.

4.2.2 Operační módy blokových šifer

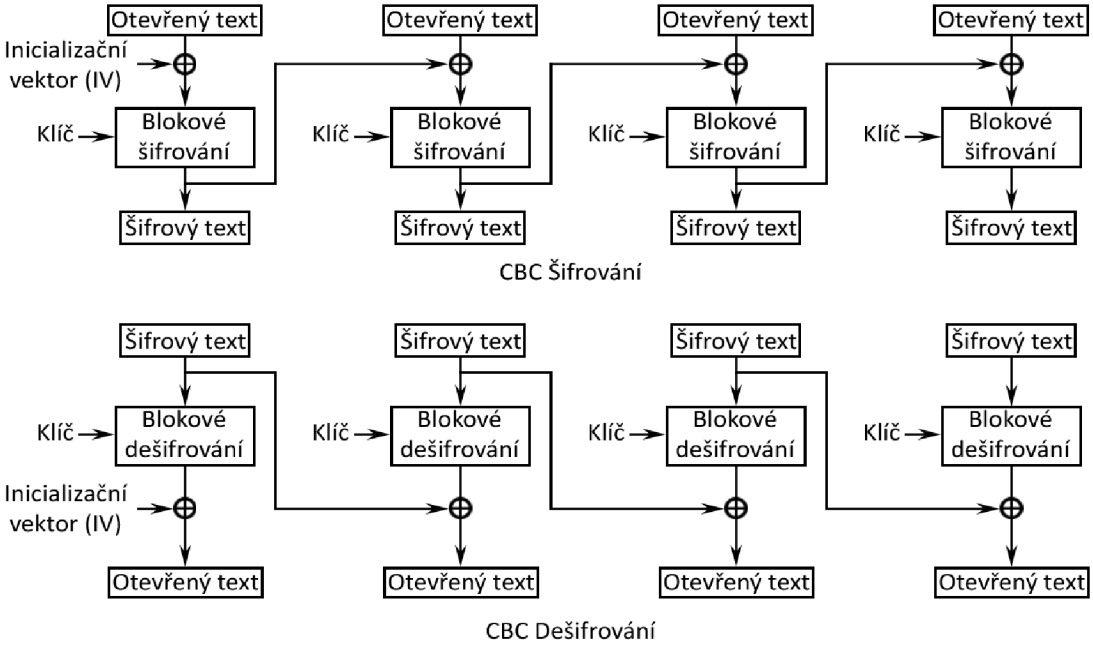
Operačními módy blokových šifer je udáván způsob, jakým je bloková šifra použita, resp. jakým způsobem je prováděno zřetězení více bloků za sebe. Pokud má otevřený text větší délku, než je délka výstupního bloku dané šifry, je třeba šifrování zřetězovat a šifrovat opakovaně, dokud není otevřený text zcela zašifrován. Pokud délka otevřeného textu není dělitelná velikostí bloku často se otevřený text doplňuje o padding (česky vycpávku). Způsob jakým budeme nakládat při zřetězování bloků nám udává operační mód. Kromě klíče, který je použit vždy, se používá také inicializační vektor (IV), který je použit pouze jednou během relace šifrování a dešifrování a pomáhá tak zabraňovat pokusům o zlomení šifry pomocí slovníků. Běžnými operačními módy blokových šifer jsou:

- ECB – Electronic Codebook
  - Žádné zřetězení neprobíhá. Otevřený text je rozdělen na bloky a ty jsou jednotlivě zašifrovány. Tímto zůstávají v šifrovém textu obrysy původní zprávy. Tento způsob tak neposkytuje žádnou difúzi, což je nežádoucí a může vést ke snadnějšímu prolomení šifry.



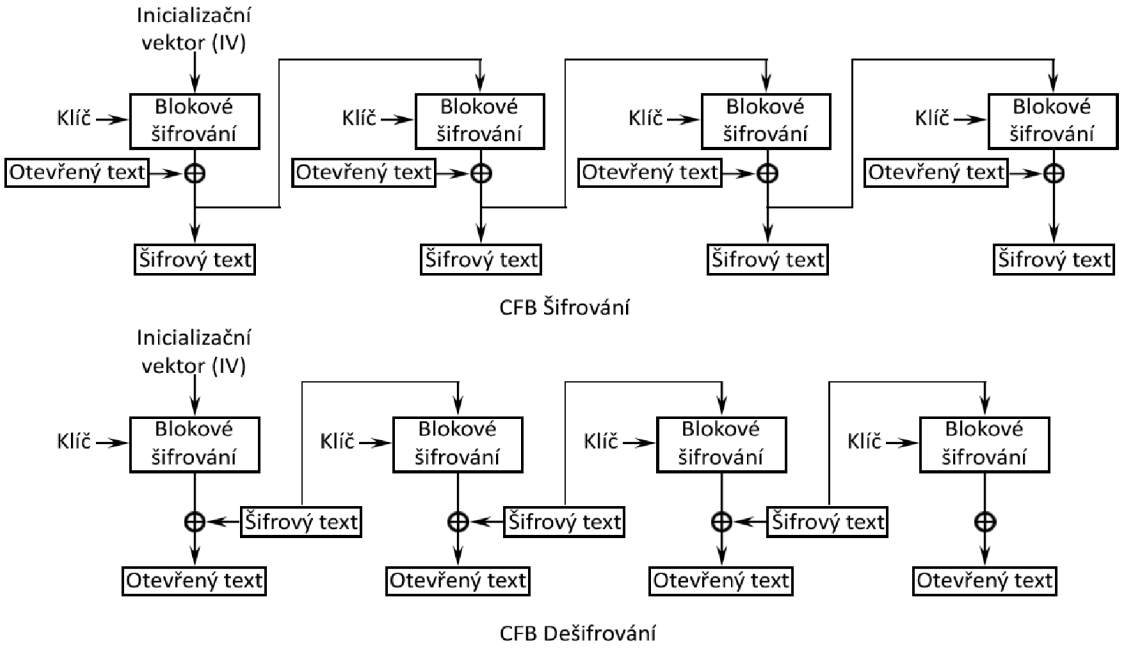
Obr. 4.2: Šifrování a dešifrování při použití operačního módu ECB.

- CBC – Cipher Block Chaining
  - První blok otevřeného textu je před zašifrováním XORován s IV. Každý zašifrovaný blok je XORován s následujícím blokem otevřeného textu před jeho zašifrováním.



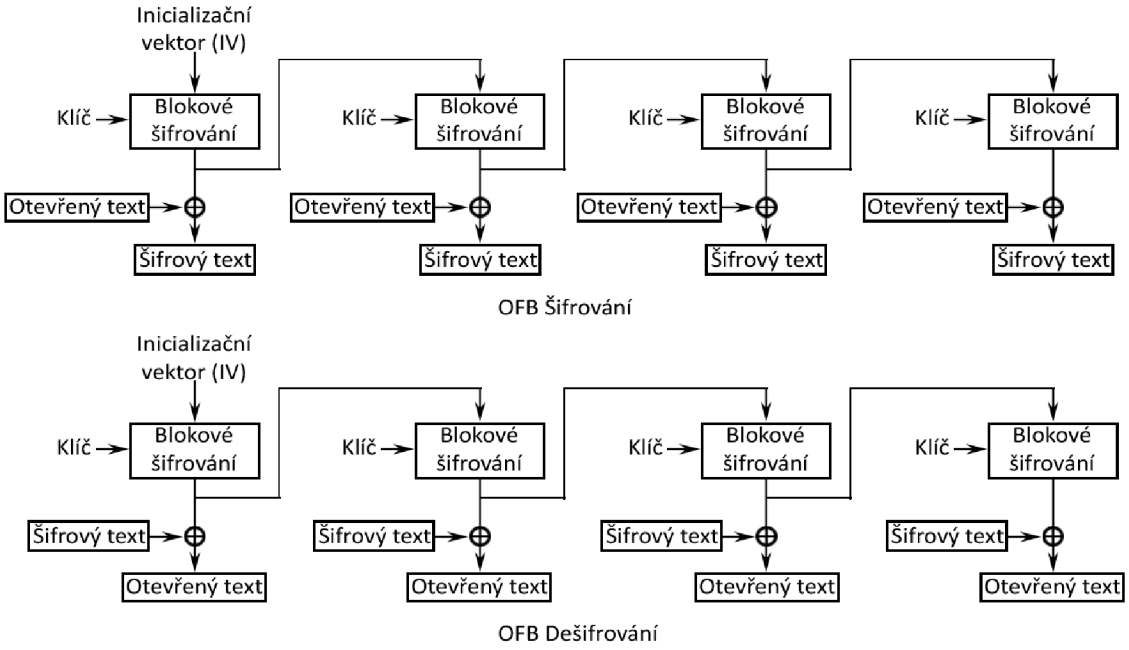
Obr. 4.3: Šifrování a dešifrování při použití operačního módu CBC.

- CFB – Cipher Feedback
  - Prvním vstupem do šifrování je IV. Blok otevřeného textu je XORován s výstupem šifrování. Po této operaci máme první blok šifrového textu, který se zřetězí na vstup dalšího šifrování, po kterém je výstup XORován s dalším blokem otevřeného textu.



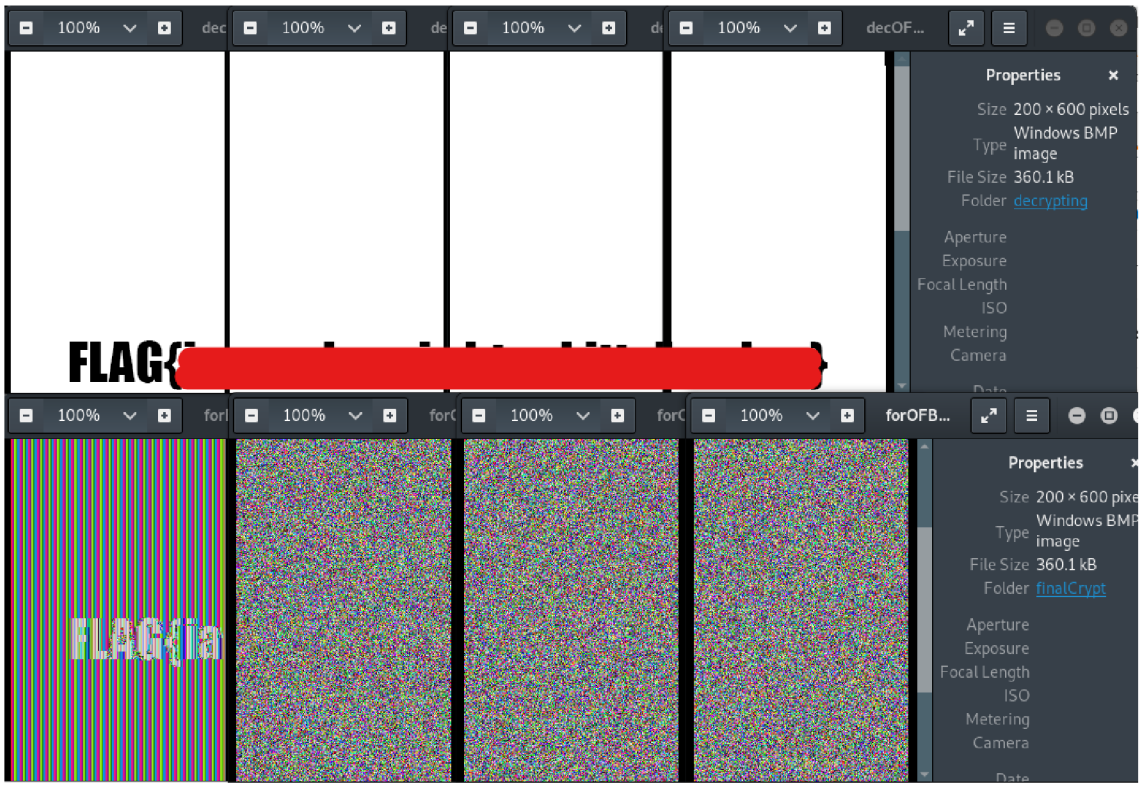
Obr. 4.4: Šifrování a dešifrování při použití operačního módu CFB.

- OFB – Output Feedback
  - Prvním vstupem do šifrování je IV. Blok otevřeného textu je XORován s výstupem šifrování. Samotný výstup blokové operace před XOR je veden na vstup další blokové operace.



Obr. 4.5: Šifrování a dešifrování při použití operačního módu OFB.

V praktické části této práce jsou hráči poskytnuty 4 soubory obrázků. Každý z těchto souborů byl zašifrován jiným operačním módem blokové šifry DES. Všechny tyto obrázky dají dohromady odpověď, kterou musí hráč nalézt. Hráči jsou poskytnuty údaje potřebné k dešifrování těchto obrázků a je mu poskytnuta čistá hlavička souboru, bez které není možné obrázky zobrazit. Více o operačních módech DES ve standardu FIPS 81[7].



Obr. 4.6: Šifrované a dešifrované obrázky, které jsou použity v úloze. Každý z nich byl zašifrován jinou blokovou šifrou. Výsledný text je úmyslně zakryt.

4.3 Digitální forenzní věda

4.3.1 Analýza provozu Wi-Fi a zlomení hesla

Bezdrátové technologie Wi-Fi, nebo také WLAN mohou používat různá šifrování komunikace. Příkladem jsou varianty WEP, WPA a WPA2. Nejsilnější varianta je WPA2. WPA2 používá k šifrování symetrickou blokovou šifru AES[2]. Jelikož se jedná o symetrickou šifru, musí dojít mezi AP[3] a zařízením klienta k výměně klíče a dalších informací včetně hesla, kterým se klient autentizuje. Tato výměna proběhne způsobem tzv. 4–way handshake (čtyřcestný handshake). Heslo pro přihlášení klienta k AP lze získat pouze z tohoto procesu, jelikož dále se pro šifrování komunikace používá klíč pro AES, který vznikl během 4–way handshake. Pokud chce útočník nalézt heslo k přihlášení klienta k AP a dešifrovat klientovu komunikace, musí mít zaznamenán klientův 4–way handshake s AP a musí být schopen heslo prolomit. Tento 4–way

[2]AES (Advanced Encryption Standard) – symetrická bloková šifra. Nahradila starší a zranitelnou šifru DES. Kromě útoku postranními kanály není znám žádný útok na plnohodnotné použití šifry AES

[3]AP (Access Point) – přístupový bod WLAN sítě.

handshake je prováděn pouze při autentizaci mezi klientem a AP. Útočník však může klienta proti jeho vůli deautentizovat a donutit jej provést 4–way handshake znovu. Útočník tak zaznamená 4–way handshake klienta a může provést buď slovníkový útok nebo útok hrubou silou na zašifrované heslo. Jelikož je operace výpočetně náročná, útok hrubou silou může trvat značnou dobu. Použití slovníkového útoku se správným slovníkem může trvat v řádu minut. Jakmile útočník heslo získá může dešifrovat komunikaci mezi klientem a AP.

V praktické části této práce je hráči poskytnut soubor odchytu síťové komunikace mezi klientem a AP. Tento odchyt byl proveden pomocí bezdrátového USB – Wi-Fi adaptéru TP-Link. V tomto odchytu je zaznamenán 4–way handshake. S pomocí programu *aircrack-ng* a použitím slovníku může hráč prolomit heslo a dešifrovat tak komunikaci klienta přes AP. V této komunikaci se pak skrývá odpověď, kterou musí hráč odeslat.

Obr. 4.7: Bezdrátový USB–Wi-Fi adaptér TP-Link TL–WN722N

4.3.2 Analýza metadat

Metadata jsou data, která poskytují informace o jiných datech. Metadata jsou zpravidla strojově zpracována, aby stroj mohl uživateli prezentovat data s užitečnou informací v čitelné formě. Může se tak jednat o hlavičky souborů, stránek, nebo metadata souborů a složek pro souborový systém. V těchto metadatech se pak zpravidla nachází informace o typu souboru, struktuře souboru, umístění souboru na disku, verze protokolu webové stránky, časy vytvoření a změny souboru atd. Metadata většinou nejsou uživateli prezentovány spolu s užitečnou informací.

V praktické části této práce jsou zpracovány dva úkoly na toto téma. V prvním je hráči dán soubor, v jehož metadatech se ukrývá odpověď, kterou musí hráč nalézt a odeslat. V druhé úloze je hráči dán obrázek s QR kódem, který skrývá odpověď. V hlavičce souboru byly změněna metadata o rozlišení obrázku a bez jejich obnovy není možné obrázek správně zobrazit.

4.3.3 Analýza obrázku – steganografie

Staganografie je metoda skrytí informace mezi jiné informace. Na první pohled se tak zdá, že data žádnou dodatečnou informaci nenesou. Příkladem může být skrytí textu do obrázku. Obrázek se nijak znatelně nezmění a není nijak zřetelné, že ukrývá nějaký text. Zpráva se do obrázku skryje tak, že se změní bit nebo 2 bity s nejmenší váhou, které nesou původně informaci o barevném kanále pixelu. Tato změna je nepatrná a na obrázku se projeví malou změnou barvy. Pokud používáme 8 bitovou informaci o barevném kanále a změníme poslední 2 bity jedné barvy, získáme barvu rozdílnou o 4 odstíny od barvy původní. Jelikož s 8 bitovou informací o barvě máme odstínů celkem 16777216. Je tato změna vskutku nepatrná. Kdybychom tuto změnu provedli ve všech kanálech jednoho pixelu získáme barvu rozdílnou o 64 odstínů. Jeden pixel by pak měl o 64 odstínů jinou barvu než na původním obrázku, který žádnou informaci neskrýval. Do každého takového pixelu pak můžeme skrýt 6 bitů informace, pokud vyčleníme 2 bity s nejmenší vahou každého kanálu.

V praktické části této práce je hráči dán obrázek. V tomto obrázku je pomocí steganografie ukrytá odpověď, kterou musí hráč odeslat. Jelikož způsobů skrytí textu do obrázku existuje mnoho, je doporučené použít online řešení uživatele GitHub stylesuxx dostupné z: `https://stylesuxx.github.io/steganography/`, na kterém byla úloha otestována.

4.3.4 Alternativní datové proudy

Alternativní datový proud je vlastnost NTFS (New Technology File System). Jedná se o možnost k původnímu datovému proudu v souborovém systému přiřadit další

datové proudy. Tyto připojené datové proudy se nazývají alternativní a jsou s původním datovým proudem spojeny. Alternativní datové proudy se spolu s původními daty kopírují, přesouvají a mažou. Alternativní datové proudy tak dovolují uložit soubory do jiných souborů. Vložené soubory se pak nezobrazují v graficky uživatelském rozhraní a bez zadání patřičných parametrů pro výpis adresářové struktury v příkazové řádce se nezobrazí ani ve výpisu. Do těchto alternativních datových proudů pak lze ukládat libovolné množství souborů o libovolné velikosti. Zdánlivě prázdná složka nebo malý textový soubor tak mohou obsahovat velké množství souborů, které nejsou běžně viditelné.

V praktické části této práce není přímo zahrnut úkol s touto problematikou. Důvodem je nemožnost použití alternativních datových proudů na jiném souborovém systému než NTFS. Jelikož platforma je spuštěna v operačním systému Linux Ubuntu (18.04), který souborový systém NTFS s datovými proudy nepodporuje, není možné efektivně distribuovat a jakkoliv použít soubory s alternativními datovými proudy. Alternativní možnost je distribuovat hráčům virtuální disky s operačním systémem, který NTFS podporuje a v tomto operačním systému úkol vyřešit. Kvůli těmto komplikacím není v práci úkol tohoto typu zahrnut a je zde pouze úkol, který naznačuje potenciální použití alternativního datového proudu.

4.3.5 Analýza ransomwaru

Ransomware je druh malware, který zpravidla napadne operační systém oběti a proti její vůli zašifruje soubory na disku. Za dešifrování souboru poté požaduje výkupné, většinou v bitcoinech skrze prohlížeč Tor, aby transakce nebyla vystopovatelná. Každý ransomware napadá systém unikátně, a proto bývá velmi těžké obnovit soubory jiným způsobem, než spolupracovat s útočníky, výkupné zaplatit a doufat, že poskytnou klíč k dešifrování souborů. Tvůrci ransomware bývají mnohdy velmi pečliví a používají takové metody, které není možné prolomit. Mnoho ransomwarů má však zranitelné místo, které lze pečlivou analýzou objevit a získat data zpět i bez zaplacení výkupného. Často se této činnosti věnují týmy, které vyvíjejí antivirové programy a jiné softwarové ochrany. Pokud takový tým nalezne způsob, jak ransomware prolomit, často zveřejní volně dostupný nástroj pro dešifrování souborů pro daný ransomware. Analýzy jsou však velmi zdlouhavé a vyžadují mnoho zkušeností.

V praktické části této práce je implementován jeden úkol, kde byl ransomware použit. Ransomwarem byly zašifrovány obrázky, které obsahují odpověď na daný úkol. Úkolem hráčů je analýzou souborů zjistit, který ransomware byl pro jejich zašifrování použit. Na tento ransomware existuje volně stažitelný nástroj pro dešifrování, proto úkolem hráčů je tento nástroj nalézt a dešifrovat všechny soubory.

Obr. 4.8: **Rozhraní úkolu Analýza ransomwaru**. V tomto rozhraní má hráč přístup k souborům, které jsou potřebné ke splnění úkolu. Hráč má možnost využít bodově penalizované nápovědy k usnadnění řešení.

4.4 Síťové služby

4.4.1 Web shell

Web shell je kategorie hrozeb pro webové služby a servery. Skrze web shell hrozbu lze zneužít nahráním skriptu na webový server a jeho spuštěním. Spuštěný skript poskytne útočníkovi vzdálený přístup k serveru s administrátorskými právy. Tyto hrozby má na svědomí špatná implementace a konfigurace webového serveru nebo služby, která je na serveru spuštěna. Pokud dané webové prostředí dovoluje útočníkovi nahrát a spustit jeho vlastní skript je potencionálně zranitelné na web shell a útočníkovi nic nebrání převzít kontrolu nad systémem. Skrze web shell lze dále útočit pomocí SQL Injection, Remote code Execution, Cross–site scripting atd.

V praktické části této práce je hráči poskytnuta adresa webového serveru, který je zranitelný na web shell. Konkrétně je pro tento úkol vybrán webový server Apache, který umožňuje útočníkovi spustit skripty v jazyce PHP. Úkolem hráče je nalézt odpověď v jednom ze souborů na webovém serveru, který není bez využití web shell zranitelnosti přístupný. Hráč tak musí využít libovolný skript, který mu poskytne

přístup k tomuto souboru.

4.4.2 Analýza webového prostředí

Analýza webového prostředí je proces, kterým jsou získávány informace o webovém serveru. Jedná se o metodické postupy, které útočníkovi / pentesterovi poskytnou informace, kterých poté může zneužít k získání kontroly nad serverem, nebo nedovolenému přístupu. Tento proces je nedílnou součástí penetračního testování webových služeb. Pokud útočník / tester neví, jaký webový server je spuštěn, na jakém operačním systému, jaký jazyk byl použit pro vytvoření webu atd., nemůže efektivně využít zranitelností, které tyto prvku mají a proniknout do systému. Každá informace, kterou při analýze útočník / tester získá může vést ke snadnějšímu průniku.

V praktické části této práce jsou implementovány 2 úkoly, které vyžadují analýzu webového prostředí. Použitím běžných nástrojů, lze získat potřebné informace pro splnění úkolů. Jeden úkol požaduje, aby hráč zjistil pouze, na kterých webových serverech běží dané webové služby. Druhý úkol vyžaduje získání více informací o jednom konkrétním serveru. Jmenovitě se jedná o název webového serveru, verze webového serveru a název operačního systému, na kterým je server spuštěn.

4.4.3 Metasploit Project

Metasploit Project je projekt bezpečnostní společnosti Rapid7. Jedná se o projekt shromažďování informací o zranitelnostech a jejich využívání. Projekt má za cíl poskytovat informace, které pomáhají získávat přehled a zkušenosti na poli penetračního testování. Nejznámější produkt je Metasploit Framework[4]. Jedná se o nástroj, který umožňuje využívat zranitelností z databáze proti vzdáleným stanicím. Nástroj je často používán k výukovým a testovacím účelům, není však vyloučeno, že nástroj bude použit ofenzivně proti legitimnímu serveru.

V praktické části této práce jsou zahrnuty 3 vzájemně propojené úkoly, které využívají Metasploit Framework a pomocnou virtuální stanici, kde běží upravená verze Linuxu, která obsahuje množství zranitelných služeb. Jeden z úkolů hráče donutí zjistit mezinárodní označení konkrétní zranitelnosti z databáze zranitelností. V dalším úkolu musí hráč využít dalšího nástroje jménem *nMap*, který umožní oskenovat síť a nalézt v ní potencionálně zranitelné stanice. V posledním úkolu hráč musí využít zranitelnosti proti pomocnému serveru a získat přístup na stanici, kde se nachází odpověď na poslední úkol.

[4]Webová stránka Metasploit: `https://www.metasploit.com/`

4.4.4 Útok skrze SSH

SSH (Secure Shell) je síťový klient–server protokol používaný k vzdálenému přístupu na zařízení v síti. Zpravidla se jedná o vzdálené a zabezpečené zpřístupnění příkazové řádky (shell) přes nezabezpečenou síť. SSH vznikl jako náhrada za nezabezpečený protokol Telnet. Běžně se SSH používá na TCP (Transmission Control Protocol) portu 22. Protokol používá kombinaci symetrické a asymetrické kryptografie. Při použití správných druhů šifer je tento protokol teoreticky zcela bezpečný. Dosud není známý žádný útok, který by dokázal kompromitovat samotný SSH protokol. Více o protokolu SSH v doporučení RFC 4250[8] a přidružených.

V praktické části této práce je implementovaný úkol, při kterém budou hráči pomocí ofenzivního nástoje *hydra* provádět slovníkový útok na OpenSSH implementaci serveru, která je spuštěná v jedné z pomocných virtuálních stanic. Nejdříve však musí provést analýzu logů, které byly staženy ze serveru po úspěšně provedeném útoku. Tyto logy obsahují informaci o existenci útoku a jeho úspěšnosti. Takto hráči zjistí, jakým způsobem na server zaútočit. Jakmile útok dokončí, mohou se přihlásit na stanici a nalézt odpověď.

Závěr

Tato práce poskytuje obecné informace o tom, co jsou to bezpečnostní hry a především tedy CTF hry. Dále jsou v práci popsány základní druhy CTF a některé dostupné platformy, na kterých lze CTF hry pořádat. Práce také pojednává o jednotlivých kategoriích úkolů pro tyto hry. Práci tak lze využít jako inspiraci pro vytvoření her vlastních.

V rámci této práce byly vybrány 3 různé dostupné platformy, které by případný organizátor mohl využít k pořádání CTF her. Praktická část této práce zahrnuje dvě oddělené platformy CTFd, na kterých jsou zpracovány jednotlivé úkoly. Jedna platforma slouží k podpoře výuky předmětu Základy kryptografie a druhá k předmětům Bezpečnost ICT 1 a 2. Z jednotlivých kategorií jsou vybrány některé úkoly, které byly pro tuto práci zpracovány. Celkově pro tuto práci bylo implementováno 51 úkolů. Všechny úkoly byly vytvořeny přímo pro tuto práci a nejsou odnikud převzaty, koncepty některých úkolů jsou výsledkem inspirace několika různých a dostupných CTF her, které již byly nebo stále jsou pořádány v síti internet a inspirace některými laboratorními úkoly, které již danou problematiku řešily. Jmenovitě se jedná o bezpečnostní hry Hack the Box[5], picoCTF 2019[6] a The Catch 2019 od CESNETu[7]. Každý úkol zahrnutý v této práci je popsán v kapitolách 3 a 4, tento popis zahrnuje teorii úkolu a základní koncept řešení. Ke každému úkolu byly přiřazeny nápovědy, které jsou patřičně bodově penalizovány. Nápovědy obsahují náznaky, které hráče mohou dovést k postupu řešení. A některé nápovědy obsahují popis a vzorové řešení úkolu tohoto typu. Úkoly jsou patřičně bodově ohodnoceny a rozřazeny do patřičných kategorií pro přehlednost.

Praktickou část této práce lze použít jako laboratorní úlohu. Studenti jsou tak posazeni do role hráčů, kteří využívají své dosavadní znalosti k řešení úloh. Během řešení se zdokonalují v používání nástrojů a postupů, které jsou využitelné v praxi penetračního testování, analýzy atd.

Teoretická i praktická část práce splnily všechny stanovené cíle bakalářské práce. Další rozvoj práce by vedl k přidávání dalších úkolů do obou platforem a rozšiřování tak oblasti, kterou pokrývají úkoly již implementované.

[5]Webová stránka Hack the Box: `https://www.hackthebox.eu/`

[6]Webová stránka picoCTF: `https://picoctf.com/`

[7]Webová stránka The Catch: `https://www.thecatch.cz/`

Literatura

[1] JOSEFSSON, S.: *The Base16, Base32, and Base64 Data Encodings* [online]. RFC 4648, RFC Editor, 2006 [cit. 25. 11. 2019]. Dostupné z URL: <http://www.rfc-editor.org/rfc/rfc4648.txt>.

[2] MORIARTY, K., KALISKI, B., JONSSON, J., RUSCH, A.: *PKCS #1: RSA Cryptography Specifications Version 2.2* [online]. RFC 8017, RFC Editor, 2016 [cit. 26. 11. 2019]. Dostupné z URL: <http://www.rfc-editor.org/rfc/rfc8017.txt>.

[3] RESCORLA, E.: *Diffie-Hellman Key Agreement Method* [online]. RFC 2631, RFC Editor, 1999 [cit. 28. 11. 2019]. Dostupné z URL: <http://www.rfc-editor.org/rfc/rfc2631.txt>.

[4] KIVINEN, T., KOJO, M.: *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)* [online]. RFC 3526, RFC Editor, 2003 [cit. 28. 11. 2019]. Dostupné z URL: <http://www.rfc-editor.org/rfc/rfc3526.txt>.

[5] RIVEST, L. R.: *The MD5 Message-Digest Algorithm* [online]. RFC 1321, RFC Editor, 1992 [cit. 29. 11. 2019]. Dostupné z URL: <http://www.rfc-editor.org/rfc/rfc1321.txt>.

[6] Microsoft Corporation: *NT LAN Manager (NTLM) Authentication Protocol Revision 31.0* [online]. MS-NLMP, Microsoft Corp., 2019 [cit. 29. 11. 2019]. Dostupné z URL: <https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-nlmp/b38c36ed-2804-4868-a9ff-8dd3182128e4>.

[7] National Institute of Standards and Technology: *FIPS 81 – Des Modes of Operation* [online]. FIPS 81, Publication NIST, 1980 [cit. 29. 11. 2019]. Dostupné z URL: <https://csrc.nist.gov/CSRC/media/Publications/fips/81/archive/1980-12-02/documents/fips81.pdf>.

[8] LEHTINEN, S., LONVICK, C.: *The Secure Shell (SSH) Protocol Assigned Numbers* [online]. RFC 4250, RFC Editor, 2006 [cit. 01. 06. 2020]. Dostupné z URL: <http://www.rfc-editor.org/rfc/rfc4250.txt>.

Seznam symbolů, veličin a zkratek

CTF	Capture the flag
FBCTF	Facebook Capture the Flag
RTB	Root the Box
RSA	Rivest-Shamir-Adleman
DH	Diffie-Hellman
MD	Message-digest algorithm
LM	LAN Manager
NTLM	New Technology LAN Manager
SHA	Secure Hash Algorithms
AP	Access Point
NTFS	New Technology File System
SSH	Secure Shell
BZKR	Základy kryptografie
TIC	Bezpečnost ICT

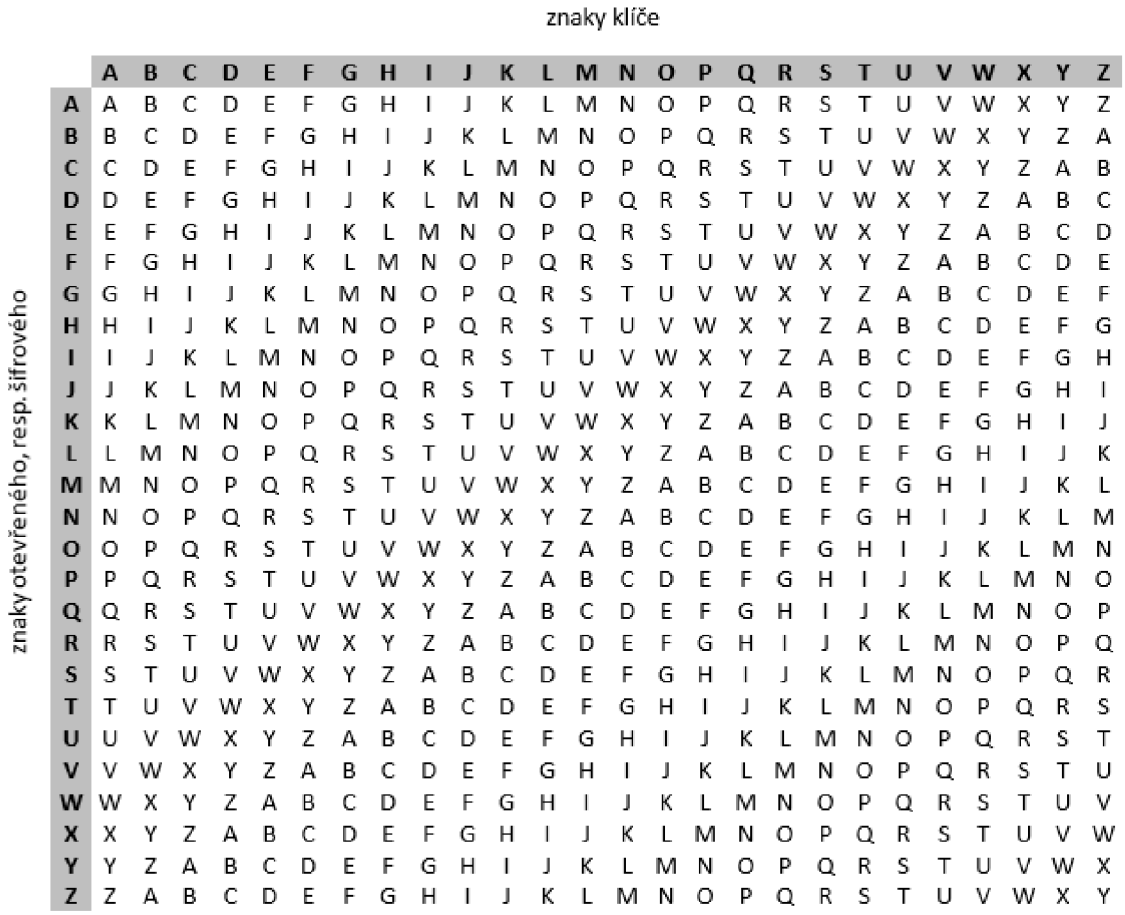
Seznam příloh

A ASCII tabulka

DEC	HEX	OCT	BIN	CHAR	DEC	HEX	OCT	BIN	CHAR	DEC	HEX	OCT	BIN	CHAR
32	20	040	00100000		64	40	100	01000000	@	96	60	140	01100000	`
33	21	041	00100001	!	65	41	101	01000001	A	97	61	141	01100001	a
34	22	042	00100010	"	66	42	102	01000010	B	98	62	142	01100010	b
35	23	043	00100011	#	67	43	103	01000011	C	99	63	143	01100011	c
36	24	044	00100100	$	68	44	104	01000100	D	100	64	144	01100100	d
37	25	045	00100101	%	69	45	105	01000101	E	101	65	145	01100101	e
38	26	046	00100110	&	70	46	106	01000110	F	102	66	146	01100110	f
39	27	047	00100111	'	71	47	107	01000111	G	103	67	147	01100111	g
40	28	050	00101000	(	72	48	110	01001000	H	104	68	150	01101000	h
41	29	051	00101001	)	73	49	111	01001001	I	105	69	151	01101001	i
42	2A	052	00101010	*	74	4A	112	01001010	J	106	6A	152	01101010	j
43	2B	053	00101011	+	75	4B	113	01001011	K	107	6B	153	01101011	k
44	2C	054	00101100	,	76	4C	114	01001100	L	108	6C	154	01101100	l
45	2D	055	00101101	-	77	4D	115	01001101	M	109	6D	155	01101101	m
46	2E	056	00101110	.	78	4E	116	01001110	N	110	6E	156	01101110	n
47	2F	057	00101111	/	79	4F	117	01001111	O	111	6F	157	01101111	o
48	30	060	00110000	0	80	50	120	01010000	P	112	70	160	01110000	p
49	31	061	00110001	1	81	51	121	01010001	Q	113	71	161	01110001	q
50	32	062	00110010	2	82	52	122	01010010	R	114	72	162	01110010	r
51	33	063	00110011	3	83	53	123	01010011	S	115	73	163	01110011	s
52	34	064	00110100	4	84	54	124	01010100	T	116	74	164	01110100	t
53	35	065	00110101	5	85	55	125	01010101	U	117	75	165	01110101	u
54	36	066	00110110	6	86	56	126	01010110	V	118	76	166	01110110	v
55	37	067	00110111	7	87	57	127	01010111	W	119	77	167	01110111	w
56	38	070	00111000	8	88	58	130	01011000	X	120	78	170	01111000	x
57	39	071	00111001	9	89	59	131	01011001	Y	121	79	171	01111001	y
58	3A	072	00111010	:	90	5A	132	01011010	Z	122	7A	172	01111010	z
59	3B	073	00111011	;	91	5B	133	01011011	[	123	7B	173	01111011	{
60	3C	074	00111100	<	92	5C	134	01011100	\	124	7C	174	01111100	|
61	3D	075	00111101	=	93	5D	135	01011101	]	125	7D	175	01111101	}
62	3E	076	00111110	>	94	5E	136	01011110	^	126	7E	176	01111110	~
63	3F	077	00111111	?	95	5F	137	01011111	_					

Obr. A.1: **Výtah tisknutelných znaků ASCII tabulky** s dekadickými, hexadecimálními, oktalovými a binárními hodnotami.

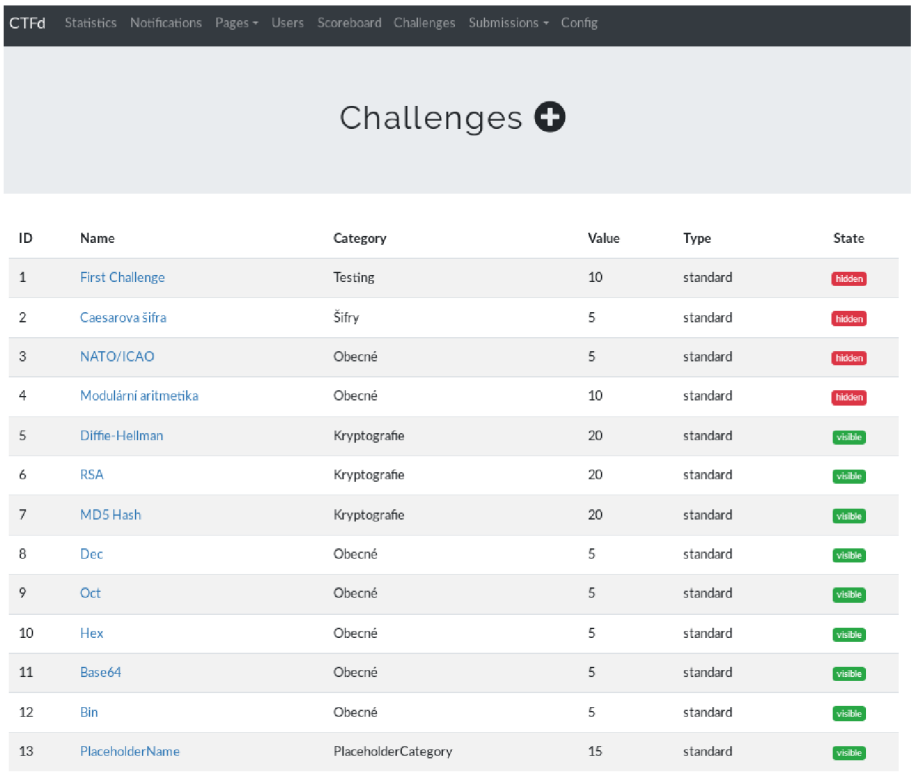
B Vigenérova šifrovací tabulka



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

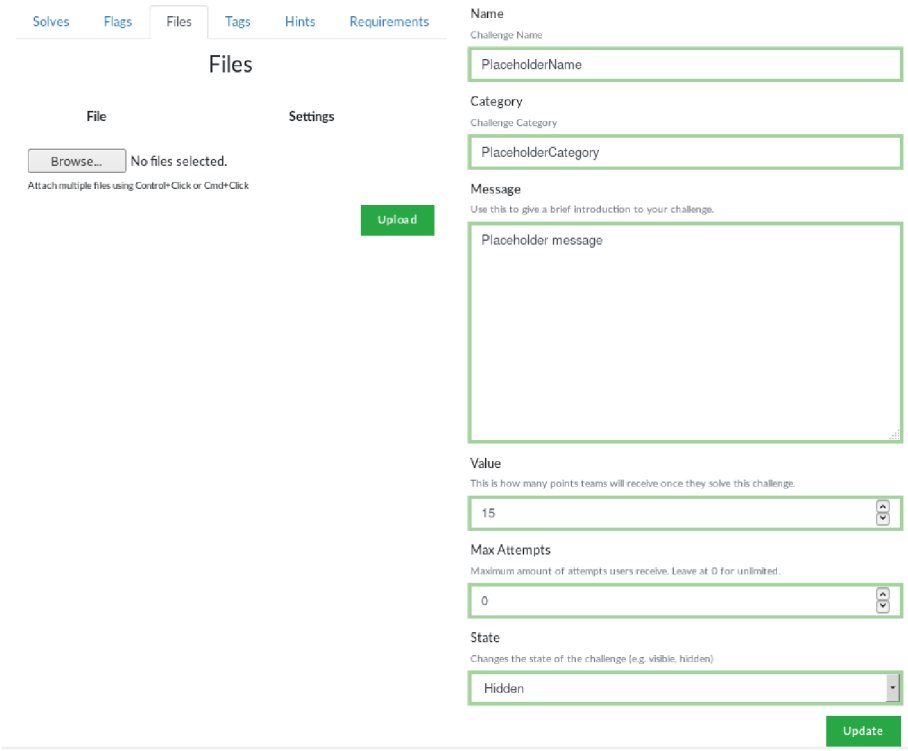
Obr. B.1: **Šifrovací tabulka** používaná, při šifrování a dešifrování **Vigenérovy šifry**. Při šifrování se dle znaku otevřeného textu a znaku klíče najde znak šifrového textu. Při dešifrování se dle znaku šifrového textu a znaku klíče najde znak otevřeného textu.

C CTFd rozhraní

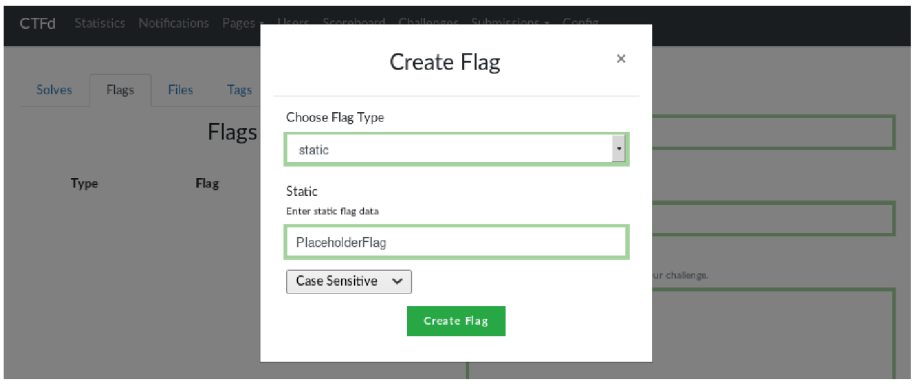


ID	Name	Category	Value	Type	State
1	First Challenge	Testing	10	standard	hidden
2	Caesarova šifra	Šifry	5	standard	hidden
3	NATO/ICAO	Obecné	5	standard	hidden
4	Modulární aritmetika	Obecné	10	standard	hidden
5	Diffie-Hellman	Kryptografie	20	standard	visible
6	RSA	Kryptografie	20	standard	visible
7	MD5 Hash	Kryptografie	20	standard	visible
8	Dec	Obecné	5	standard	visible
9	Oct	Obecné	5	standard	visible
10	Hex	Obecné	5	standard	visible
11	Base64	Obecné	5	standard	visible
12	Bin	Obecné	5	standard	visible
13	PlaceholderName	PlaceholderCategory	15	standard	visible

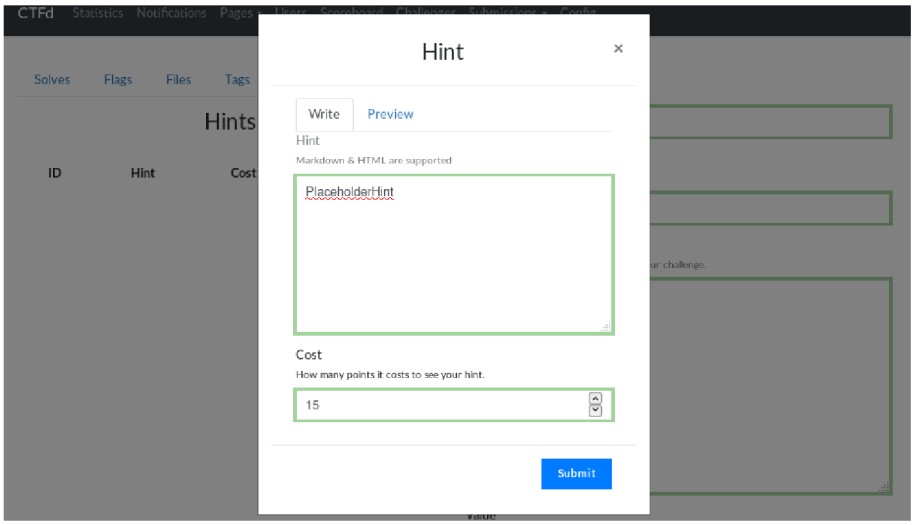
Obr. C.1: **Administrativní rozhraní úkolů CTFd.** V tomto rozhraní je možné úkoly přidávat a upravovat úkoly existující.



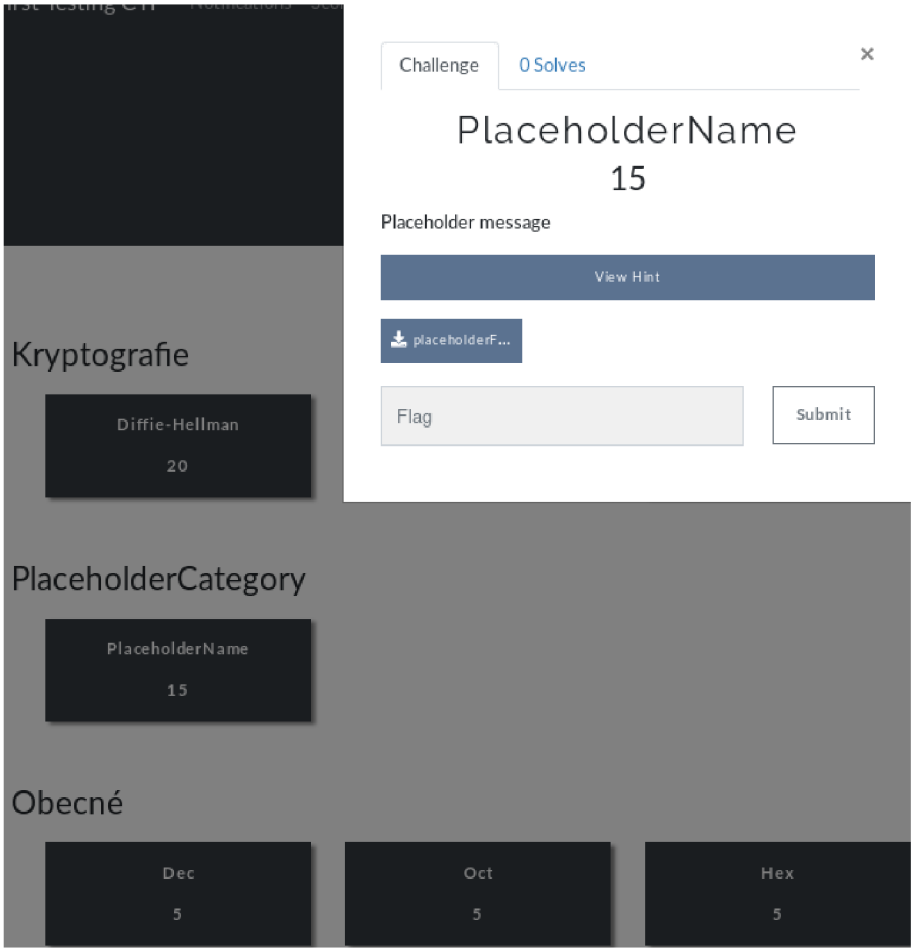
Obr. C.2: **Administrativní rozhraní pro vytvoření a úpravu úkolů CTFd.** V tomto rozhraní se nastavují odpovědi, nápovědy, připínají soubory a upravují závislosti na ostatních úkolech pro aktuálně vybraný úkol. Také se zde upravuje počet bodů za vyřešení, maximální počet pokusů a viditelnost úkolu pro hráče.



Obr. C.3: **Kontextová nabídka přidání odpovědi úkolu.** V této nabídce je možné přidat odpovědi pro aktuálně upravovaný úkol. Odpovědi je možné formátovat jako regulérní výraz nebo konstantně a je možné nastavit citlivost na velikost písmen.

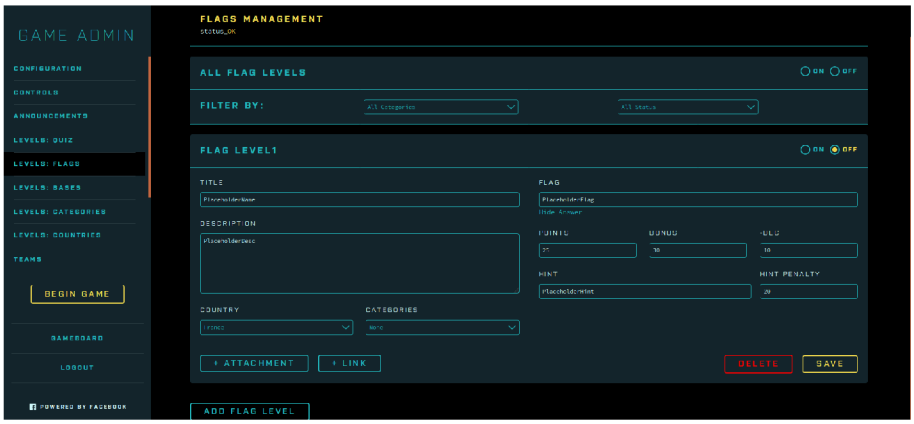


Obr. C.4: **Kontextová nabídka přidání nápověd úkolu** V této nabídce je možné přidat nápovědy pro aktuálně upravovaný úkol a udávat bodovou penalizaci.



Obr. C.5: **Hráčovo rozhraní úkolů.** V tomto rozhraní hráč plní zadané úkoly. V pozadí má do kategorií rozdělené ostatní úkoly a v kontextové nabídce volitelnou nápovědu a připnutý soubor pro aktuálně řešený úkol.

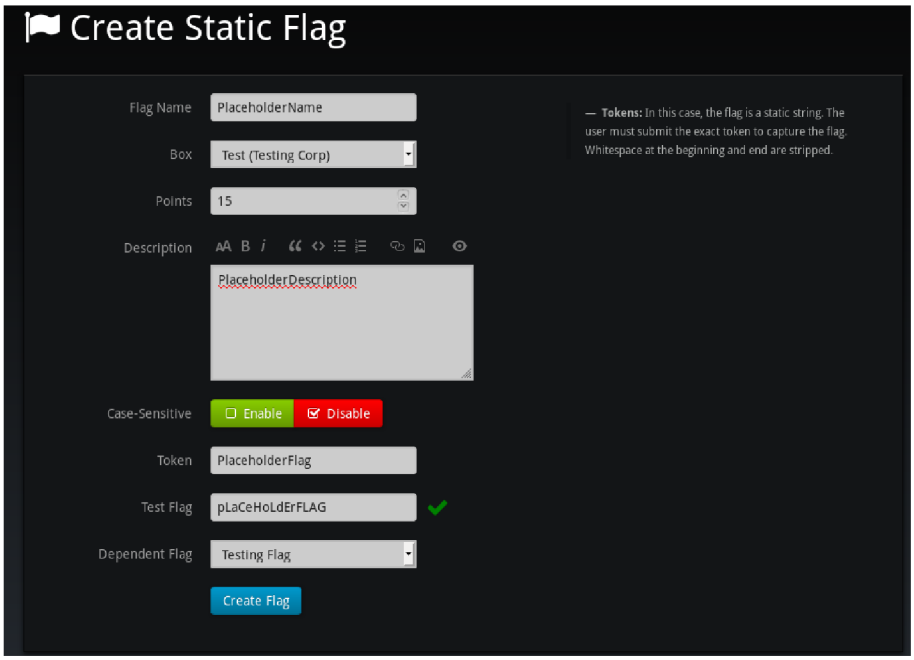
D FBCTF rozhraní



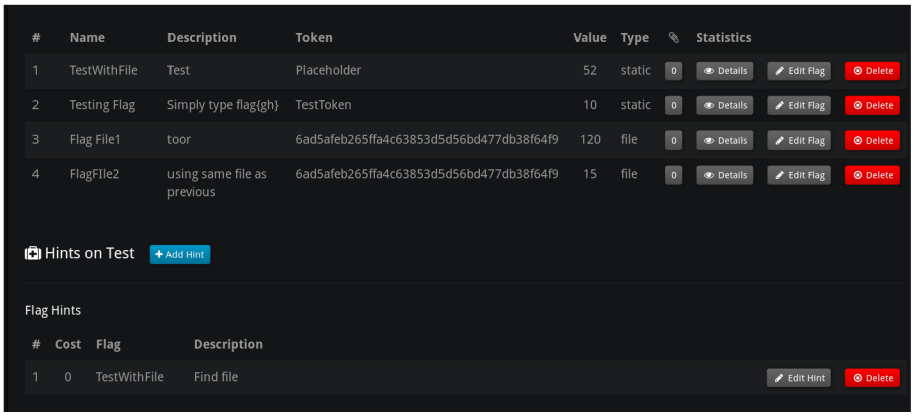
Obr. D.1: **Vytváření úloh v FBCTF** Organizátor vytváří úlohy a propojuje je se státy, které poté hráč vidí ve svém rozhraní obr.: D.2. Organizátor může k úkolu připnout soubor nebo odkaz a může k úkolu přidat bodově penalizovanou nápovědu pro hráče.

Obr. D.2: **Rozhraní hráče FBCTF**. Hráč vidí mapu světa s hranicemi jednotlivých států. Státy představují území, které hráč získá za splnění daného úkolu. Vyřešení úkolu je odměněno body a území se pro ostatní hráče stane nedostupné.

E RTB rozhraní



Obr. E.1: **Administrativní rozhraní pro vytvoření a úpravu úkolů RTB.** V tomto rozhraní lze úkol vytvořit a přidat mu popisek. Rovněž se zde úkolu přiřazuje odpověď, kterou je možno rovnou otestovat. Úkol se pak řadí do tzv. boxu, což jsou v tomto případě jakési kategorie. Také zde lze nastavit závislost na jiném úkolu.



Obr. E.2: **Rozhraní administrátora RTB**. V tomto rozhraní může administrátor vytvářet jednotlivé úkoly a nápovědy k úkolům.

F Výsledky testování výkonu platforem

Výpis F.1: Výstup testování CTFd pomocí programu *ApacheBench*

```
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.0.2.9 (be patient)


Server Software:
Server Hostname:      10.0.2.9
Server Port:          8000

Document Path:        /
Document Length:      263 bytes

Concurrency Level:    200
Time taken for tests: 35.643 seconds
Complete requests:    5000
Failed requests:      0
Non-2xx responses:    5000
Total transferred:    2885000 bytes
HTML transferred:     1315000 bytes
Requests per second:  140.28 [#/sec] (mean)
Time per request:     1425.740 [ms] (mean)
Time per request:     7.129 [ms] (mean, across all concurrent requests)
Transfer rate:        79.04 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median max
Connect:        0   20 137.5     0    1038
Processing:    13 1391 450.6  1385    4289
Waiting:        7 1390 450.6  1384    4289
Total:         28 1411 472.9  1390    4290

Percentage of the requests served within a certain time (ms)
  50%   1390
  66%   1456
  75%   1507
  80%   1547
  90%   1801
  95%   2067
  98%   3088
  99%   3290
 100%   4290 (longest request)
```

Výpis F.2: Výstup testování FBCTF pomocí programu *ApacheBench*

```
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.0.2.9 (be patient)


Server Software:      nginx
Server Hostname:      10.0.2.9
Server Port:          80

Document Path:        /
Document Length:      178 bytes

Concurrency Level:    200
Time taken for tests: 1.112 seconds
Complete requests:    5000
Failed requests:      0
Non-2xx responses:    5000
Total transferred:    2925000 bytes
HTML transferred:     890000 bytes
Requests per second:  4495.91 [#/sec] (mean)
Time per request:     44.485 [ms] (mean)
Time per request:     0.222 [ms] (mean, across all concurrent requests)
Transfer rate:        2568.47 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median max
Connect:        2   20  66.8     5     465
Processing:    15 1562 240.4  1597    1884
Waiting:       11 1561 240.5  1596    1884
Total:         30 1582 204.2  1605    1888

Percentage of the requests served within a certain time (ms)
  50%   1605
  66%   1654
  75%   1676
  80%   1693
  90%   1740
  95%   1767
  98%   1794
  99%   1823
 100%   1888 (longest request)
```

Výpis F.3: Výstup testování RTB pomocí programu *ApacheBench*

```
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.0.2.9 (be patient)


Server Software:
Server Hostname:      10.0.2.9
Server Port:          8888

Document Path:        /
Document Length:      3039 bytes

Concurrency Level:    200
Time taken for tests: 15.593 seconds
Complete requests:    5000
Failed requests:      0
Total transferred:    16730000 bytes
HTML transferred:     15195000 bytes
Requests per second:  320.67 [#/sec] (mean)
Time per request:     623.700 [ms] (mean)
Time per request:     3.119 [ms] (mean, across all concurrent requests)
Transfer rate:        1047.80 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median max
Connect:        0   60 406.0     0    7297
Processing:     6  438 1112.1   323   13591
Waiting:        2  438 1112.1   323   13591
Total:         10  498 1201.4   324   14364

Percentage of the requests served within a certain time (ms)
  50%    324
  66%    330
  75%    340
  80%    374
  90%    388
  95%   1197
  98%   1756
  99%   3812
 100%  14364 (longest request)
```

G Implementované úlohy

Name	Category	Value	Type	State
Klasická šifra 1	Kryptografie	10	standard	visible
Modulární aritmetika	Kryptografie	6	standard	visible
Ustanovení klíče	Kryptografie	20	standard	visible
Asymetrický problém	Kryptografie	20	standard	visible
Dec	Obecné	5	standard	visible
Oct	Obecné	5	standard	visible
Hex	Obecné	5	standard	visible
Base64	Obecné	5	standard	visible
Bin	Obecné	5	standard	visible
Klasická šifra 2	Kryptografie	15	standard	visible
Zase modulo?	Kryptografie	6	standard	visible
Scandinavian Defense	Obecné	10	standard	visible
Fiiiiii	Kryptografie	6	standard	visible
Greatest Corn Duck	Kryptografie	6	standard	visible
CRiTical	Kryptografie	10	standard	visible
Gen	Kryptografie	12	standard	visible
Inception ascii	Kryptografie	15	standard	visible
Exorcism	Kryptografie	25	standard	visible
2 klíče 3 zprávy	Kryptografie	25	standard	visible
PrvoČíslo MR	Kryptografie	10	standard	visible
PrvoČíslo LL	Kryptografie	10	standard	visible
Náměstí a násobení	Kryptografie	12	standard	visible
Na frontě klid	Kryptografie	15	standard	visible
Fůůůůůůů	Kryptografie	10	standard	visible
Decay	Kryptografie	8	standard	visible
widePeepoHappy extended	Kryptografie	8	standard	visible

Obr. G.1: **Úlohy implementované pro BZKR** Seznam úkolů implementované pro předmět Základy kryptografie.

5. generace	Kryptografie	20	standard	visible
Dec	Obecné	5	standard	visible
Oct	Obecné	5	standard	visible
Hex	Obecné	5	standard	visible
Base64	Obecné	5	standard	visible
Bin	Obecné	5	standard	visible
Uživatel Ubuntu	Kryptografie	25	standard	visible
Uživatel XP	Kryptografie	20	standard	visible
Uživatel 7	Kryptografie	25	standard	visible
Chaos	Obecné	7	standard	visible
Operace DES	Kryptografie	40	standard	visible
Wireless 404	Digitální forezní věda	30	standard	visible
1. meta	Digitální forezní věda	10	standard	visible
2. meta	Digitální forezní věda	12	standard	visible
Stega... co?	Digitální forezní věda	15	standard	visible
Alternativní data alpha	Digitální forezní věda	12	standard	visible
Web shell	Síťové služby	20	standard	visible
Upside Down	Digitální forezní věda	10	standard	visible
Klep klep klep	Síťové služby	25	standard	visible
AWIGSJ	Digitální forezní věda	25	standard	visible
Pre - web shell	Síťové služby	10	standard	visible
I'M IN 01	Síťové služby	10	standard	visible
I'M IN 02	Síťové služby	10	standard	visible
I'M IN 03	Síťové služby	15	standard	visible
EZ server	Síťové služby	10	standard	visible

Obr. G.2: **Úlohy implementované pro TIC** Seznam úkolů implementované pro předměty Bezpečnost ICT 1 a 2.