VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2018 Karel Kuchař

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

BEZPEČNOST OPERAČNÍCH SYSTÉMŮ
SECURITY OF OPERATING SYSTEMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Karel Kuchař
AUTHOR

VEDOUCÍ PRÁCE — Ing. Vlastimil Člupek, Ph.D.
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**
Ústav telekomunikací

Student: Karel Kuchař ***ID:*** 185931
Ročník: 3 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Bezpečnost operačních systémů

POKYNY PRO VYPRACOVÁNÍ:

V bakalářské práci proveďte bezpečnostní analýzu operačních systémů (OS) pro různé platformy a popište jejich nejběžnější zranitelnosti. Navrhněte scénář zabezpečení vybraného OS pro osobní počítače a pro servery s využitím aplikačních programů. Scénáře aplikujte na vybrané OS a implementujte nástroje pro sledování událostí v OS (skenování systému, změny systémových stavů, přístup do OS, upozornění na nestandartní chování, tvorba dokumentace nestandartních stavů apod.). Na serverový OS nainstalujte vhodný webový server a navrhněte a proveďte jeho zabezpečení. Zabezpečené OS vystavte útokům cílených na analýzu jejich zranitelností či na jejich kompromitaci (penetračním testům). Proveďte vyhodnocení aplikovaných zabezpečení a doporučte další postup.

DOPORUČENÁ LITERATURA:

[1] PETROWSKI, Thorsten. Bezpečí na internetu: pro všechny. Dialog, 2014.

[2] HOFMANN, Owen S., et al. Inktag: Secure applications on an untrusted operating system. In: ACM SIGARCH Computer Architecture News. ACM, 2013. p. 265-278.

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 29.5.2018

Vedoucí práce: Ing. Vlastimil Člupek, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

Abstrakt

Bakalářská práce se zabývá bezpečností operačních systémů, na kterou je kladen stále větší důraz. Nejprve je věnována pozornost na základní zásady bezpečnosti operačních systémů. Dále jsou čtenáři přiblíženy jednotlivé prvky, které napomáhají tvořit komunikaci na internetu bezpečnou. V třetí kapitole jsou rozebrány z pohledu bezpečnosti nejpoužívanější operační systémy a popsány jejich nejběžnější zranitelnosti. Další kapitola se věnuje jednotlivým scénářům, ve kterých je definováno, jak vhodně nastavit jednotlivé prvky systému, tak aby byl operační systém bezpečný a také jsou použity aplikace, které slouží ke zvýšení bezpečnosti systému. Bezpečnostní scénáře jsou vytvořeny pro osobní počítat, server a webový server. V páté kapitole jsou jednotlivé scénáře aplikovány na stroje, které jsou virtualizovány. K otestování zvoleného zabezpečení byly jednotlivé stroje vystaveny penetračnímu testování. Poté je rozebráno, jak jednotlivé prvky systémů reagovaly a zaznamenaly probíhající penetrační testování. Posléze je provedeno vyhodnocení zvoleného zabezpečení a je doporučen další postup. Tato bakalářská práce si klade za cíl seznámit čtenáře s problematikou bezpečnosti operačních systémů a důležitostí implementace bezpečnostních opatření.

Klíčová slova

Bezpečnost**,** operační systémy, antivirus, firewall, šifrování, IDS.

Abstract

This bachelor thesis is focused on security of operating systems, which is more and more important. At first, basics of security of operating systems will be covered, followed by describing elements, which help secure communication on internet. In the third chapter, there are discussed the most using operating systems in terms of security and mentioned their most common weaknesses. Following chapter talks about scenarios in which is defined how to properly set each part of the system, so it is secure; also, security applications that make operating system more secure are mentioned. In fifth chapter are above mentioned scenarios applied to virtual machines. To test chosen security settings were individual machines exposed to penetration test. Afterwards, the chosen security settings are evaluated and further steps are recommended. The goal of this thesis is to apprise with problems of security of operating systems and the importance of implementation of security measures.

Keywords

Security, operating systems, antivirus, firewall, encryption, IDS.

Bibliografická citace

KUCHAŘ, K. *Bezpečnost operačních systémů.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 97 s. Vedoucí bakalářské práce Ing. Vlastimil Člupek, Ph.D.

Prohlášení

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

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

V Brně dne …… Podpis autora ……..

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Vlastimilovi Člupkovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce. Také své přítelkyni Evě Holasové, která provedla penetrační testování na zabezpečené operační systémy.

V Brně dne …… Podpis autora ……..

OBSAH

Seznam obrázků

SEZNAM TABULEK

1 ÚVOD

Operační systémy se neustále zdokonalují, disponují vyšším výpočetním výkonem, také poskytují služby svým uživatelům. Právě na takových systémech a službách jsme mnohem více závislí než dříve, proto je nutné dbát nejen u těchto systému na bezpečnost, dostupnost a integritu.

Bezpečnost je stále častěji používaný pojem. Bohužel, nikdy nedosahuje sta procent, a proto je nutné si uvědomit, že operační systémy nejsou zcela odolné vůči všem typům hrozeb, které ohrožují bezpečnost operačních systému.

Techniky útočníků se neustále zdokonalují a využívají stále sofistikovanějších nástrojů, pomocí kterých provádějí získávání informací. Důležité informace jsou získávány nejen pomocí skenovaní portů systémů, nebo z veřejných databází, ale také je mnohdy využíváno sociálního inženýrství. Následně jsou získané informace zneužity za účelem proniknutí do systému a jeho kompromitace. Mnohdy útočníci směrují nadměrný, nebo nekorektní síťový provoz na zařízení s cílem odepřít službu dalším legitimním uživatelům.

Právě z těchto důvodů je nutné dbát na bezpečnost zařízení jako celku a minimalizovat možnost proniknutí do systému. Proto je vhodné se zaměřit na zranitelnosti, provádět jejich odhalování a následné napravení. Cílem je útočníkovi co nejvíce ztížit práci při pokusu o průnik do OS. V nejlepším případě proaktivně odhalovat jednotlivé pokusy o průnik do OS a provést patřičné kroky, které eliminují další takový pokus.

V této práci se zaměříme na nejčastější zranitelnosti pro různé platformy. Vytvoříme a aplikujeme scénář zabezpečení jak pro osobní počítač, tak pro serverovou variantu operačního systému. Abychom byli schopni detekovat možný pokus o narušení bezpečnosti, tak implementujeme patřičné nástroje. Také vytvoříme a implementujeme návrh zabezpečení pro webový server. K vyhodnocení aplikovaného zabezpečení vystavíme OS cíleným útokům.

2 BEZPEČNOST

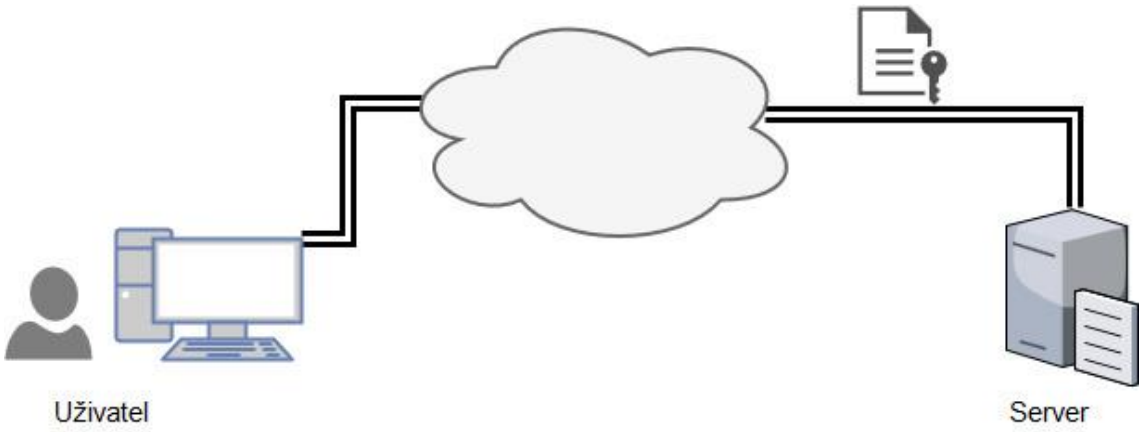
Bezpečnost je velice široký pojem. Lze ji definovat jako stav, kdy ztráta aktiv nepřekračuje stanovenou míru. Na základě, co je naším cílem chránit, lze provést dělení na fyzickou bezpečnost, bezpečnost sítě a služeb a na bezpečnost lidských zdrojů [1]. Samozřejmě abychom zajistili co nejvyšší bezpečnost, je třeba vynaložit úsilí, finanční prostředky a využívat různých prostředků k otestování funkčnosti našeho zabezpečení. Ovšem nikdy není nic stoprocentní a je třeba se v této oblasti stále zdokonalovat a udržovat naši bezpečnost ze všech ohledů na adekvátní úrovni.

Důležité body, na které je třeba brát zřetel v zaměření na bezpečnost sítě a služeb jsou [2]:

a) dostupnost,
b) autentičnost,
c) důvěrnost.

Dostupnost je stav, kdy uživatelům naší sítě garantujeme, že naše služby budou přístupné a v případě havárie se provedou adekvátní kroky k obnovení tohoto stavu. V oblasti bezpečnosti často skloňovaný pojem autentičnost je stav, kdy naše aktiva nebyla pozměněna a pomocí bezpečnostních mechanismů si lze tento stav ověřit. Důvěrnosti se rozumí stav, kdy je přístup k aktivům dovolen pouze autentizovaným a autorizovaným uživatelům naší sítě [2].

Na Obr. 2.1 je znázorněna komunikace kdy uživatel komunikuje skrze síť internet a připojuje se vzdáleně na server. Nejprve proběhne autentizace a autorizace uživatele do OS, poté se uživatel chce přihlásit na server, ten však musí být dostupný, v případě že tomu tak je a nebyl vyřazen z provozu ať útočníkem, nebo probíhající údržbou, havárií apod., se uživatel autentizuje a autorizuje i zde. Poté co jsou data stahovaná směrem k uživateli je pomocí symetrické/asymetrické kryptografie zajištěno, že nedošlo ke změnám a lze si to i ověřit (viz dále).



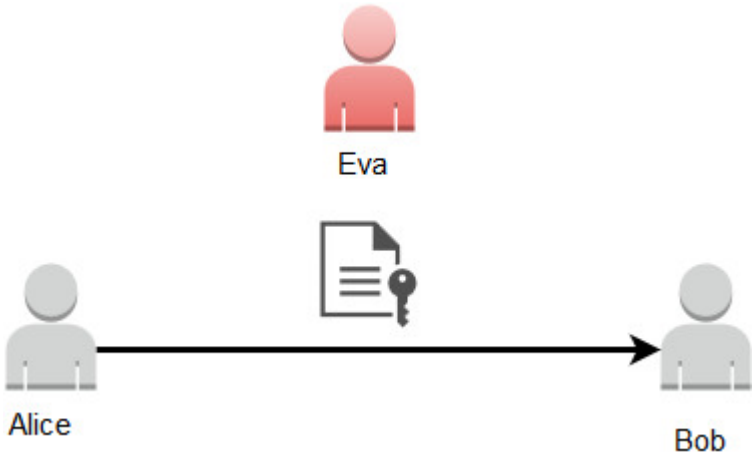
Obr. 2.1: Náhled na bezpečnost sítě a služeb.

Abychom zajistili co nejvyšší možnou bezpečnost dat, je třeba využívat určitých prvků, které napomáhají udržovat bezpečnost na takové úrovni která je dostačující pro běžnou komunikaci. Většina těchto prvků je použita, i když si to mnohdy ani neuvědomujeme. Cílem informační bezpečnosti je poskytnout sadu vlastností, které definují hlavní cíle bezpečnosti.

Jednotlivými vlastnostmi jsou:

a) Důvěrnost.
b) Autentičnost.
c) Nepopiratelnost.
d) Integrita.

Tyto vlastnosti si nejlépe objasníme na elektronickém dokumentu zasílaném přes síť internet. Představme si typický obrázek pro bezpečnost, viz Obr. 2.2. Záměr Alice je takový, aby si ona i Bob byli jisti, že je Eva (v roli útočníka) nemůže odposlouchávat. Je proto použito adekvátní šifrování, pomocí něhož je zajištěna důvěrnost. Bob si musí být jistý, že dokument odeslala Alice, a ne nikdo jiný. Toto je zajištěno Autentičností. Nepopiratelnost definuje to, že Alice není schopna popřít, že daný dokument pochází od ní. Bob ví, že data nebyla v průběhu přenosu žádným způsobem modifikována, protože spoléhá na integritu, tj. stálost dokumentu. V praxi realizováno elektronickým podpisem.



Obr. 2.2: Komunikace Alice a Boba.

V následujících podkapitolách si rozebereme jednotlivé prvky zajišťující bezpečnost.

2.1 Prostředí

Základním pravidlem bezpečnosti uživatele je samotné prostředí. Je samozřejmé, že nebude příliš bezpečné přihlašovat se na elektronické bankovnictví z internetové kavárny, a to stejné platí pro provádění takových operací pomocí nezabezpečeného bezdrátového připojení. Proto je důležité si rozmyslet, jaké kroky chcete provádět a nedůvěřovat všemu

co je zdarma a provádět důležité kroky jako například elektronické platby ze stroje, kterému důvěřujete a přes dostatečně zabezpečené připojení k síti internet.

2.2 Autentizace, autorizace

Autentizace je určena k tomu, aby se uživatel přistupující do systému jednoznačně identifikoval [3]. Běžně se tak provádí pomocí zadání uživatelského jména a uživatelského hesla k účtu, k němuž se chceme přihlásit. Typicky se k tomuto principu používá databáze uživatelů, vůči které je uživatel autentizován. Hesla nebývají uváděna v běžně čitelné podobně, ale využívají hashovacích mechanismů. Tyto mechanismy se vyznačují jednocestností, výstupem s konstantně definovanou délkou a výsledek je kompletně změněn v případě malé změny na vstupu. Navíc je velice rychlé a jednoduché porovnat dva řetězce, v případě shody hashů je uživatel autentizován. Autentizaci lze také provádět pomocí hardwarového zařízení, např. pomocí čipové karty.

Autorizace představuje proces, při kterém dojde k ověření uživatelských přístupových práv do systému [3]. Tento proces je typicky úzce spojen s autentizací a navazuje na ni. Jde tedy o to, aby bylo definováno, zda má uživatel práva k požadované akci.

2.2.1 Heslo, nutné zlo

Fráze pro přístup alias hesla jsou po nás vyžadována na každém elektronickém kroku. Potřebujeme je na přihlášení do e-mailu, Wi-Fi, operačního systému apod. [4]. Používání pouze jednoho hesla není správný směr, kterým bychom se měli vydat. V případě prolomení našeho hesla má útočník „otevřený“ přístup na všechny služby, které používáme a naše identita tak může být ukradena. Typicky je doporučováno používat rozdílná hesla, která s námi nejeví žádné spojení. Heslo by mělo obsahovat malá písmena, velká písmena, čísla a speciální znaky. Čím bude naše heslo delší a bude používat rozdílnější prvky, tím bude heslo více odolávat útokům ze strany útočníka [5].

Heslová fráze (Passphrase) [6], [7] se lehce odlišuje od hesla. Zatímco heslo je běžně spojeno s autentizací osob do systému, heslová fráze je typicky mnohem delší (více než 15 znaků) a je použita při šifrování soukromých klíčů, nebo ke kontrole přístupu uživatele do počítačového systému. Hlavní důraz je kladen na to, aby tato fráze nebyla náchylná slovníkovému útoku, tedy aby fráze nedávala žádný logický smysl.

Existuje mnoho způsobů, jak takové frázové heslo vytvořit. Nejbezpečnější je použití vygenerovaného řetězce, na něž jsou kladeny stejné nároky jako na heslo, jen s větší délkou [7]. Další metodou je použít tzv. vzorovou větu. Z této věty si lze vzít první písmena každého slova, dále podle určitého principu použít střídání malých a velkých písmen, použít speciální znak (například spojovník uvnitř věty), dále použít čísla. Vzniklý řetězec by měl být dostatečně silný a odolný vůči prolomení útočníkem.

2.2.2 Dvoufaktorová autentizace

Jedná se o způsob ověření identity na základě dvou nezávislých faktorů. Dochází tak ke snížení rizika při zneužití běžných přihlašovacích údajů.

Jednotlivé faktory lze rozdělit na [8]:

a) Faktor, který uživatel má → identifikační karta, mobilní telefon (příjem SMS).
b) Faktor, který uživatel zná → pin, jméno + heslo.
c) Faktor, který uživatel je → biometrický údaj.

Dvoufaktorová autentizace se dosahuje kombinací dvou faktorů. Nutnost zadávat dva faktory sice jemně protahuje proces autentizace, ale dochází k výraznému posílení bezpečnosti celého procesu.

2.3 Antivirus

Antivirus je počítačový program, jehož účelem je efektivně bránit a eliminovat výskyt škodlivého softwaru [9].

Antivirus může provádět detekci na několika vrstvách [1]:

a) Webový štít.
b) Statická ochrana.
c) Detekce na základě chování.
    a. Kódová emulace.
    b. Sandbox.
    c. Detekce v reálném čase.
    d. Cloudově založená detekce.

Webový štít testuje data, která se přenáší při navštěvování stránek a kontroluje jejich obsah. Jedná se o testování použitých skriptů, HTTPS a další.

Statická ochrana představuje základní část antivirového programu a je tvořena tzv. signatury, tj. popisy kódů nebezpečného softwaru, známými anomáliemi apod. Za účelem jednotné detekce virů byl vytvořen tzv. EICAR test, tedy bezpečný kód, který je antiviry označen za virus, lze tak jednoduše zkontrolovat aktivitu statické ochrany před nebezpečným softwarem.

Kódová emulace spadá do kategorie popisu viru na základě jeho chování. Kódová emulace provádí kontrolu aplikací bez potřeby je spouštět. Dalším zástupcem v kategorii detekce na základě chování je sandboxing, jak je z názvu patrné, umožníme spuštění kódu, ovšem na kontrolovaném a izolovaném území, na tzv. pískovišti. Detekce v reálném čase využívá monitoring spuštěných aplikací a jejich komunikaci s operačním systémem, vyhledávání určitých vzorů na základě umělé inteligence. Posledním zástupcem detekce na základě chování je cloudově založená detekce. Zde je vše prováděno na vzdáleném serveru, využívá detekci virů na základě označených URL a dále zde provádí kódovou emulaci, sandboxing a další postupy.

2.4 Firewall

Firewall představuje ucelený soubor bezpečnostních opatření bránící neoprávněnému elektronickému přístupu do počítačové sítě/stroje. Hlavním účelem je filtrace a monitorování probíhajícího provozu, identifikace komunikujících stran (IP adresy, porty), identifikace druhu služby, identifikace typu spojení a další. Po přesném definování spojení jsou na základě definovaných pravidel provedeny akce a paket je vpuštěn do sítě, nebo je odmítnut a zahozen.

Část sítě nacházející se na pomezí vnitřní a vnější sítě se nazývá demilitarizovaná zóna. Firewall tuto síť označuje vyšším stupněm důvěrnosti než vnější síť, ovšem nižším stupněm než síť interní.

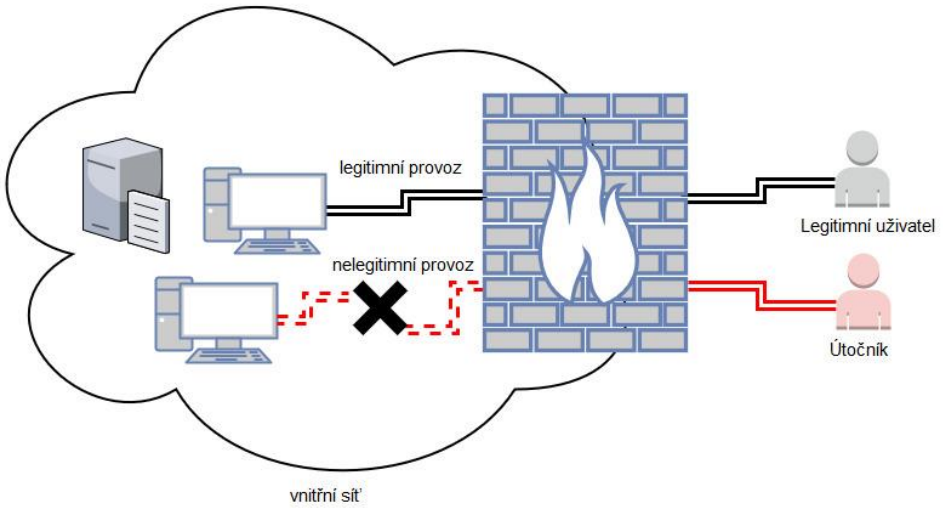
Dělení firewallů:

a) Paketový.
b) Stavový.
c) Aplikační.
d) Firewall nové generace (New Generation FireWall).

Nejstarším typem zařízení dohlížející na tok dat je tzv. paketový firewall, provádí pouze kontrolu paketů z pohledu IP adresy a portu. Dalším typem firewallu je stavový firewall, jedná se o kombinaci paketového firewallu s prováděním detekce stavu spojení. Dochází ke třídění spojovaného spojení a nespojovaného, dále je identifikováno, zda spojení bylo zahájeno z vnitřní či vnější sítě. Aplikační firewall kontroluje data komunikujících aplikací a dochází k vyhledávání vzorců možného útoku, nebo průniku nežádoucího softwarem. Firewally nové generace poskytují hloubkovou inspekci datového toku. Nejprve je provedeno základní zpracování paketu (zdrojová adresa, cílová adresa, zdrojový port, cílový port). Poté aplikována politika elementární bezpečnosti (zda jsou porty otevřené, typ spojení). Následně je detekována aplikace (šifrovaný datový tok, aplikovaná politika pro dešifrování, označení aplikace). Dalším krokem je uplatnění bezpečnostní politiky (kontrola bezpečnostními pravidly, kontrola bezpečnostními profily). Posledním krokem je uplatnění politiky po zpracování (opětovné šifrování provozu, aplikování NAT, přeposlání paketu). Mezi nejznámější NGFW patří například Palo Alto, Hillstone Firewall. Výhodou těchto firewallů je to, že je možné lépe zacházet s provozem. Díky značkování jednotlivých paketů lze třídit pakety dle jejich obsahu, což běžný firewall neumožňuje [1], [10].

Na Obr. 2.3 je znázorněn hlavní smysl, za jakým je firewall vytvořen, přesněji, aby byl přístup do sítě legitimnímu uživateli dovolen a zároveň nežádoucím službám a uživatelům znemožněn.

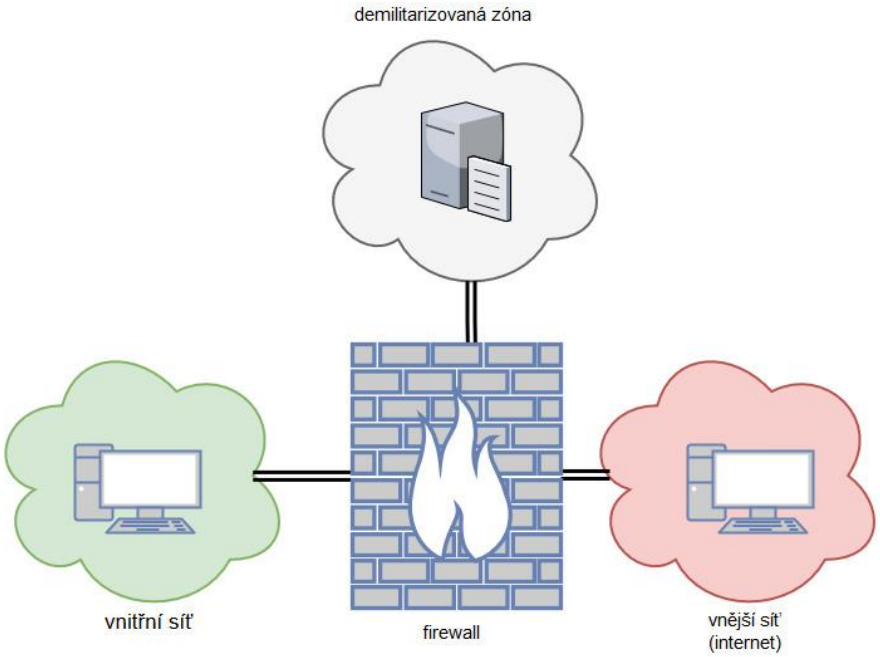
Další možnou variantou, kde je síťové zařízení firewall umístěno je uvnitř tzv. demilitarizované zóny. Jedná se oblast, kde je určitá míra důvěry mezi vnější sítí (internetem) a vnitřní sítí. Lze tak definovat, jaké jsou požadavky, resp. pravidla síťového provozu do určitých zón. Vizualizace zapojení je zobrazena na Obr. 2.4. Z obrázku je zjevné, že v tomto případě musí firewall disponovat třemi síťovými rozhraními.



Obr. 2.3: Základní myšlenka firewallu.

Doposud byl v této práci firewall považován za samostatně uložené zařízení chránící síť, mimo operační systém. Nyní se zaměříme na softwarové firewally.

Jedná se o osobní firewally, které běží přímo na koncovém operačním systému. Jejich správa je jednodušší než v případě hardwarové varianty. Společnosti vyvíjející operační systémy často poskytují vlastní firewally (např. Windows Defender v případě OS Windows 10), k dispozici jsou také komerční řešení od jiných společností. Pro vyšší míru uživatelské přívětivosti probíhá správa pomocí grafické nástavby. Účelem je zejména ochrana běžících aplikací.



Obr. 2.4: Firewall v zapojení s demilitarizovanou zónou.

2.5 PKI

Jedná se o infrastrukturu, která umožnuje existenci šifrované komunikace v oblasti asymetrické kryptografie. Tato struktura je založena na existenci dvou klíčů, soukromého a veřejného klíče a modula. Soukromý klíč si uživatel definuje a veřejný je z něj odvozen.

Pokud je našim cílem šifrovat používáme k tomu náš veřejný klíč. Veřejný klíč spolu s modulem je veřejně znám. Na druhou stranu, pokud chceme použít tyto klíče k podepsání dokumentu, použijeme k tomu náš soukromý klíč a náš veřejný klíč spolu s modulem může být veřejně znám [11].

2.6 Šifrování

Šifrování představuje převod dat do nečitelné podoby a zpětný převod je možný jen se znalostí tajné informace. Hlavním cílem je, aby převod, tedy dešifrování bylo umožněno pouze oprávněnému uživateli.

Kryptografii lze rozdělit do dvou hlavních proudů (dle počtu klíčů):

a) Symetrická kryptografie.
b) Asymetrická kryptografie.

Symetrická kryptografie používá k šifrování a dešifrování aktiva totožný klíč. Výhodou této kryptografie je její rychlost, na druhou stranu je zde problém distribuce klíče k příjemci. Předpokladem této kryptografie je výměna klíče pomocí zabezpečeného kanálu. Každý, kdo vlastní tento klíč může zasahovat do komunikace. Pomocí symetrické kryptografie nelze provádět elektronické podpisy. Nejznámějšími zástupci jsou algoritmy AES (Advanced Encryption Standard) a DES (Data Encryption Standart).

Asymetrická kryptografie, často označována jako kryptografie pomocí veřejného klíče, představuje skupinu kryptografických metod využívajících dva typy klíčů. Tyto klíče jsou k sobě inverzní, tudíž jeden slouží k šifrování a druhý k dešifrování zprávy. Jeden z klíčů se označuje jako „veřejný“ a druhý jako „soukromý“. Asymetrická kryptografie spoléhá na neřešitelnost diskrétního logaritmu či problému faktorizace velkých čísel.

Asymetrická kryptografie umožňuje provádět jak šifrování, tak provádění elektronických podpisů. V případě, že chceme využít asymetrické kryptografie k šifrování, používáme na straně odesílatele veřejný klíč příjemce (veřejně znám) a na straně příjemce soukromý klíč příjemce. Tedy kdokoli může poslat šifrovanou komunikaci směrem k příjemci a nikdo jiný, než on není schopen provést proces dešifrování. Toho je zajištěno z důvodu neznalosti soukromého klíče příjemce. Nejznámějším zástupcem asymetrické kryptografie je algoritmus RSA, pojmenovaný po svých autorech Ronald Lorin Rivest, Adi Šamir, Leonard Max Adleman [12], [13].

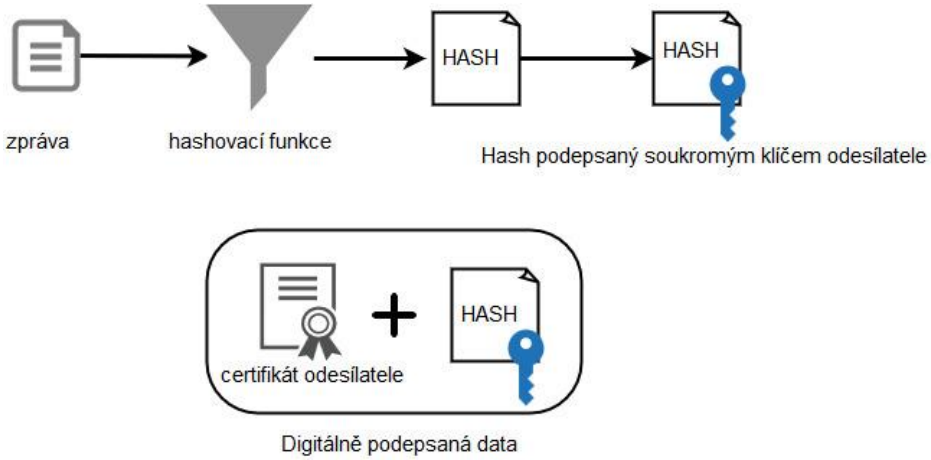
2.7 Elektronický podpis

Elektronický podpis ke své funkčnosti typicky využívá asymetrické kryptografie. K zaslané zprávě jsou připojena data, která příjemci umožňují ověření odesílatele. Běžně

dochází k ověření integrity dat. Hlavním důvodem používání elektronického podpisu je přesné definování odesílatele.

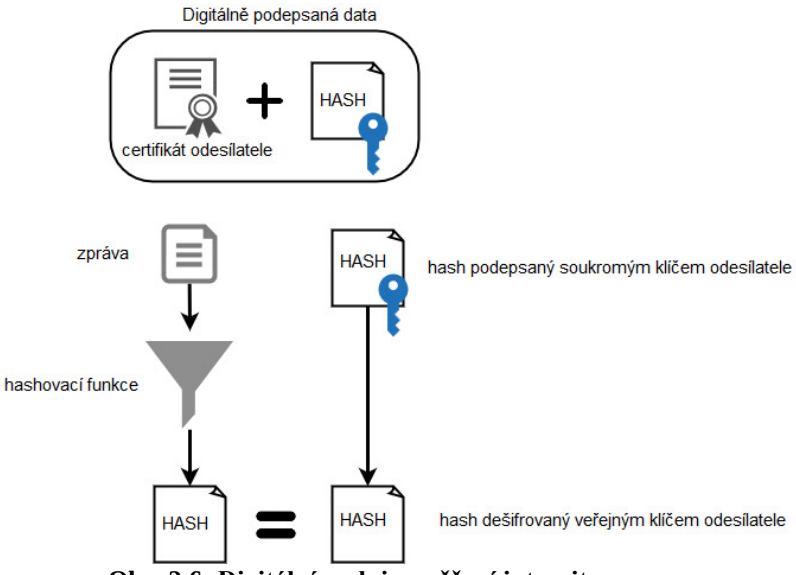
Odesílat zprávy vlastní svůj vlastní pár klíčů (soukromý a veřejný klíč). Identita odesílatele je ověřována dle certifikátu. Certifikát je vydán tzv. certifikační autoritou, nezávislým subjektem, který je všeobecně vnímán jako důvěryhodný (např. Česká pošta). Navíc i certifikát odesílatele je podepsán certifikační autoritou [14], [15].

Na Obr. 2.5 je znázorněn postup podepisování elektronického dokumentu. Nejdříve se vytvoří digitální otisk, tj. hash (jednocestná funkce s pevně definovanou délkou výstupu) elektronického dokumentu. Následně je tento otisk podepsán pomocí soukromého klíče odesílatele. Digitálně podepsaná data jsou tvořena certifikátem odesílatele a podepsaným otiskem odesílaných dat.



Obr. 2.5: Elektronický podpis, podepisování.

Na Obr. 2.6 je znázorněn opačný proces. Po přijetí digitálně podepsaných dat je vytvořen digitální otisk dat jako v předchozím případě (levá strana obrázku). Digitálně podepsaný otisk je ověřen (dešifrován) pomocí veřejného klíče a výstupem je otisk, který vznikl na straně odesílatele. Pokud se jednotlivé otisky shodují tak data, která jsme dostali, jsou původní a nebyly při přenosu nijak modifikovány.



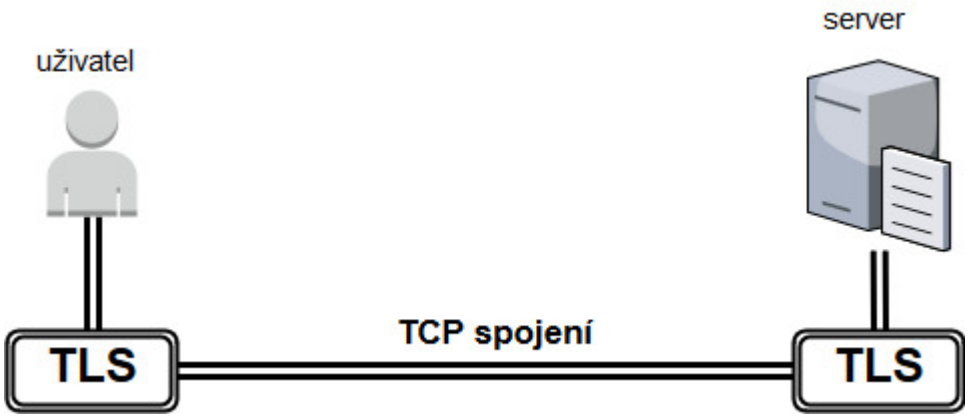
Obr. 2.6: Digitální podpis, ověření integrity.

2.8 SSL/TLS

Hlavním účelem protokolu TLS (Transport Layer Security) je zabezpečení přenášených dat. Předchůdcem protokolu TLS byl protokol SSL (Secure Socket Layer) a TLS z něj vychází, přesněji z verze 3. Protokol TLS může být využit jakýmkoliv vyšším aplikačním protokolem.

TLS slouží pro zabezpečení jak spojově orientovaného přenosu (TCP), tak nespojově orientovaného přednosu dat (typicky EAP-TLS).

Nyní si objasníme funkci protokolu TLS v případě použití TCP. TCP protokol je duplexní protokol určen pro komunikaci dvou stran. Pojem duplexní značí, že je umožněna komunikace oběma směry, jedním směrem je přenášena komunikace směrem ke klientovi a druhým směrem jsou data přenášena serveru. Je tak vytvářena mezi vrstva mezi aplikační protokolem a protokolem TCP, viz Obr. 2.7.



Obr. 2.7: TLS jako mezivrstva.

TLS není schopna zabezpečovat síťový provoz na úrovni aplikace, tedy nelze provádět funkce jako elektronický podpis apod. Zabezpečení probíhá na základě každého paketu. Důvěrnost paketů je zajištěna pomocí kryptografie a integrita pomocí kontrolního součtu.

Autentizace jednotlivých stran probíhá na základě certifikátů. Je umožněna autentizace obou komunikujících stran, ovšem běžně se využívá autentizace pouze serveru (anonymní TLS server). Přenos dat probíhá šifrovaně na základě symetrické kryptografie. Symetrický klíč se vymění na základě asymetrické kryptografie na začátku komunikace. Po definování těchto prvků vzniká TLS relace [16].

Protokol TLS se skládá z protokolů [17]:

a) Handshake protocol (HP).
b) Record Layer Protocol (RLP).

Handshake protokol se inicializuje ihned po navázání TCP spojení, slouží pro dohodnutí kryptografických algoritmů, klíčů.

RLP slouží pro zajištění bezpečnosti, přebírá pakety od aplikace a šifruje je podle dohodnutých algoritmů a vypočítává kontrolní součet.

Nyní si uveďme, co vše je třeba ke vzniku TLS relace:

a) Klient nabízí kryptografické algoritmy a server si z nich vybírá.
b) Server i klient vygenerují svá náhodná čísla.
c) Proběhne výměna certifikátů a kryptografické informace sloužící k autentizaci.
d) Autentizace.
e) Výměna kryptografických parametrů, vznik předběžného tajemství, z něj je následně vypočítáno hlavní sdílené tajemství. Protože strany znají certifikát protistrany, mohou používat veřejné klíče protistrany a zašlou zřetězené náhodné číslo, datum a čas. Takto vzniklo předběžné tajemství. Z tohoto předběžného tajemství je pomocí protokolu Diffie-Hellman vypočítáno hlavní sdílené tajemství.
f) Ověření, zda je hlavní sdílené tajemství totožné.
g) Poskytnutí protokolu RLP symetrický klíč, informace pro výpočet kontrolního součtu.

Samotný provoz se šifruje pomocí symetrické kryptografie za účelem urychlení komunikace. Typicky jsou využity protokoly AES, DES, 3DES a další. Typické použití SSL/TLS je pro HTTPS (Hyper Text Transfer Protocol Secure).

2.9 VPN

Pojem VPN (Virtual Private Network) [18], [19], [20] představuje sadu instrukcí k vyjednání bezpečného připojení do zabezpečené, avšak vzdálené sítě pomocí zabezpečeného tunelu.

Tato služba je typu klient-server. Tedy existují dvě části, jedna určena k instalaci na stanici uživatele, který se bude připojovat do vzdálené sítě. Druhá umožňující připojení těchto klientu na vstupu do zabezpečené sítě.

Při odeslání dat jsou data předána VPN klientovi, který tato data zašifruje a prostřednictvím tunelu přepošle na VPN server. V průběhu komunikace není možné dešifrovat přenášený obsah zpráv z důvodů aplikace kryptografických mechanismů. Na server VPN jsou kladeny vysoké bezpečnostní požadavky. Jsou zde dešifrována data a dále k cíli putují bez použití kryptografických mechanismů, viz Obr. 2.8.

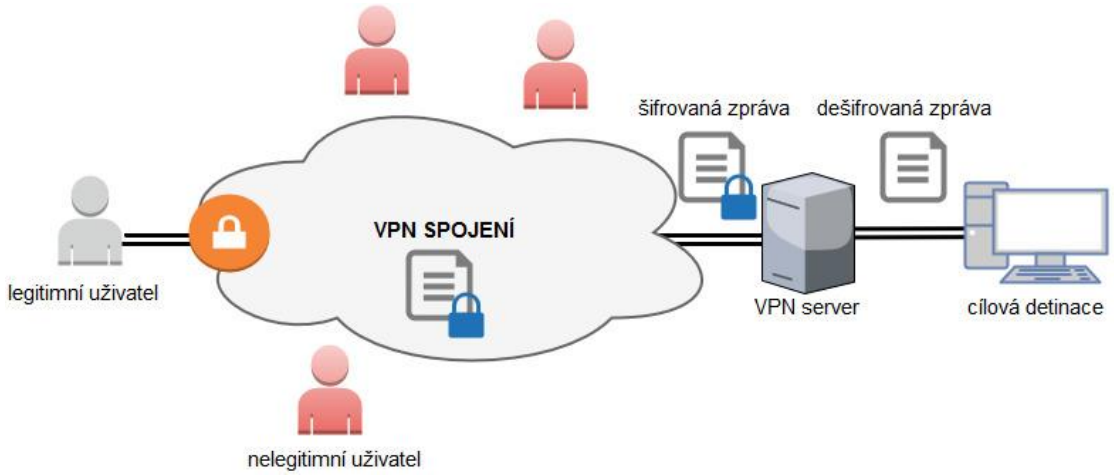
Existuje celá škála protokolů. Mezi nejznámější patří PPTP, L2TP/IPSec, OpenVPN a řada dalších [21]. Mezi nejběžněji používaný protokol komerčními službami patří OpenVPN, a proto si jej nyní rozebereme.

Jedná se o technologii s otevřeným zdrojovým kódem. Využívá knihovnu OpenSSL, která podporuje řadu kryptografických algoritmů a protokoly TLS. Technologie OpenVPN využívá transportní protokoly TCP a UDP, záleží na konfiguraci.

Samotné šifrování aplikované u OpenVPN se skládá ze dvou částí [21]:

a) Šifrování řídícího kanálu.
b) Šifrování datového kanálu.

Šifrování řídícího kanálu je také označováno, jako šifrování TLS. Jedná se o vyjednání bezpečného spojení mezi klientem a serverem. Skládá se ze samotné šifry, šifrovaného handshaku a HMAC. Šifrování datového kanálu představuje šifrování samotných dat. Skládá se ze samotné šifry a HMAC. V podstatě se jedná o šifru uvnitř šifry.



Obr. 2.8: VPN spojení.

K samotnému šifrování se používá symetrické kryptografie. Běžně šifra Blowfish a AES. Šifra Blowfish-128 je výchozí šifrování pro OpenVPN. Délka klíče se může pohybovat v rozmezí od 32 bitů do 448 bitů. Ovšem má své nedostatky a neměla by být používána pro šifrování na řídícím kanálu.

Další možností je algoritmus AES. Výhodou je, že délka bloku má 128 bitů, tudíž lze šifrovat bloky přesahující 4 GB, na rozdíl od Blowfish (pouze 64 bitové bloky). Délka klíče algoritmu AES může nabývat hodnot 128 bitů, 192 bitů, nebo 256 bitů.

Pro bezpečné navázání spojení se používá OpenVPN handshake TLS. Tato komunikace je založena na algoritmu RSA, nebo ECDH, popřípadě Diffie-Hellman.

Čím silnější šifrování bude aplikováno, tím bude celkové spojení pomalejší, proto se často volí silné šifrování řídících kanálu a méně silné šifrování datového kanálu.

VPN lze také využít pro anonymizaci uživatele. Cílová destinace totiž nemá přístup k IP adrese počítače, který je připojen do VPN tunelu, ale pouze adresu VPN serveru. Ovšem z důvodu zabezpečené komunikace dochází k mírnému zpomalení datového toku kanálem.

2.10 SSH

SSH (Secure Shell) je protokol, který umožňuje vytváření šifrované komunikace pomocí příkazové řádky mezi dvěma uzly. Nahrazuje nezabezpečený protokol Telnet [22]. Ke komunikaci využívá spojení typu klient-server. Komunikace typicky pracuje na protokolu TCP, na portu 22.

Při navazování spojení dojde k výměně klíčů. SSH verze 2 využívá pro ustanovení společného klíče algoritmus Diffie-Hellman. Po jeho výpočtu je komunikace šifrována pomocí symetrické šifry (typicky AES, 3DES) [22].

Operační systém Windows neumožňuje přímé připojení k vzdálenému serveru pomocí příkazové řádky. Je nutné použít aplikaci třetích stran. Na rozdíl OS Linux tuto službu umožňuje přímo v terminálu (označení příkazové řádky OS Linux).

2.10.1 Autentizace heslem

Aby server rozpoznal, s jakým klientem komunikuje, tak se využívá autentizace pomocí uživatelského jména a hesla. Příkaz k připojení na vzdálený server pomocí SSH: „`ssh <jméno uživatele>@<adresa serveru>`“.

Poté je uživatel vyzván k autentizaci heslem. Taková implementace je velice rychlá a jednoduchá a umožňuje ověřit uživatele odkudkoli. Ovšem útočník se může pokoušet o totéž a k hádání hesla a uživatelského jména může použít hrubou sílu. Jedná se tedy o možnou slabinu zabezpečení, přes kterou může být následně prováděn útok [23].

2.10.2 Autentizace certifikátem

Jedná se o metodu, kdy nedochází k autentizaci za pomocí hesla, ale pouze k ověření elektronického podpisu. Pomocí veřejného klíče přihlašovaného klienta, který je uložen na serveru, se ověří podpis, který byl vytvořen klientovým soukromým klíčem.

Od SSH verze 5.7 je umožněno použití algoritmů RSA, DSA, ECDSA. Na klientovy se vytvoří dva klíče (soukromý a veřejný). Soukromý klíč je možné chránit uživatelským heslem. Veřejný klíč uživatele je následně přenesen na serverovou část.

Samotné přihlašování na server probíhá stejným příkazem. Zadáme heslo k dešifrování soukromého klíče. Poté se spustí automatický proces, který není zobrazován uživateli. Server vygeneruje náhodná data a zašle je klientovi. Tyto data jsou podepsána za pomocí soukromého klíče a zaslána zpět na serverovou část. Zde se opět využívá toho, že tyto klíče nelze vzájemně odvodit. Server ověří podpis a v případě shody je klientovi umožněno se serverem dále komunikovat [24].

Tímto se stala komunikace bezpečnější a v případě, že je na serverové straně zakázána autentizace pomocí uživatelského jména a hesla, je chráněn i samotný server před útočníky, kteří se snaží prolomit bezpečnostní opatření a přístupové údaje.

Aby se situace stala pro uživatele jednodušší, lze využít tzv. „ssh agenta“. Tento program je schopen předávat soukromý klíč lokálnímu SSH klientovi.

2.11 IDS, IPS

V případě IDS (Intrusion Detection System) [1] se jedná o systém, jehož hlavním cílem je automatizování procesu detekce průniku. V případě, že je naším zájmem nejen detekovat průnik, ale provést protiopatření za účelem mitigace dopadu nežádoucích incidentů, tak hovoříme o IPS (Intrusion Prevention System). Tento systém může být realizován hardwarově, nebo softwarově.

Jedním z problémů těchto systémů je vhodné nastavení. Je třeba nastavit tyto systémy tak, aby falešné poplachy nevznikaly příliš často, ale zároveň, aby byla zajištěna obrana a probíhala účinná detekce průniku.

Aby byla tato funkce zajištěna, je třeba vytvářet tzv. logy, tedy textové, nebo binární záznamy o tom, jaká událost byla systémem zaznamenána. Ukládání logů je možné na logovací server, v případě rozsáhlejší sítě lze využít sběrná místa a následně logy ukládat na logovací server. Poslední variantou je využití sběrných míst, logovacího serveru propojeného s externím uložištěm.

Důvodem použití složitějších zapojení využívající externí uložiště je za účelem nemožnosti odstranění těchto údajů, i když naše obranné mechanismy selhaly a celá síť byla kompromitována. Je tedy vhodné propojit použití kryptografie s redundancí.

Tyto systémy pracují s dvěma typy událostí:

a) Známé nepříznivé události.
b) Neznámé nepříznivé události.

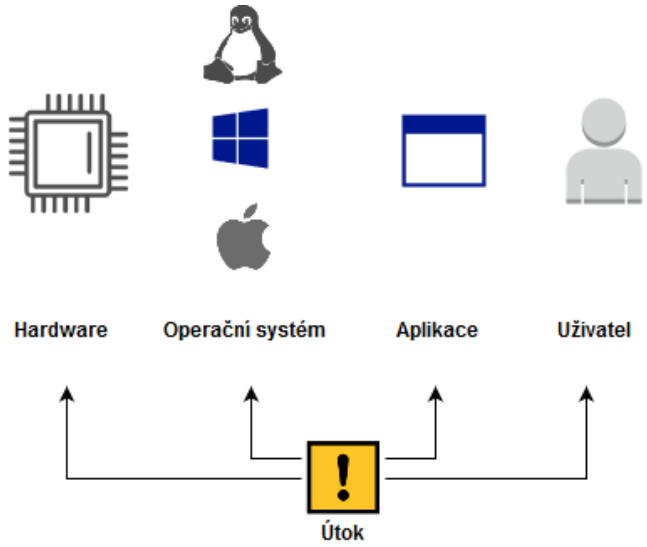
Známé nepříznivé události se detekují velice snadno na základě signatur, které jsou v síti vyhledávány.

Mnohem problematičtější je oblast neznámých nepříznivých událostí. Je třeba odhalovat i ty pokusy o průnik, které nejsou doposud známé. Toho je dosahováno tak, že se detekují anomálie, počet výskytu jevu za definovaný časový úsek, nebo se používá strojové učení k detekci anomálií. Na základě vyhodnocení jsou vykonány určité kroky za účelem ochrany sítě/systému.

3 BEZPEČNOSTNÍ ANALÝZA

Operační systém tvoří velmi zásadní část zabezpečení, proto je velmi důležité používat bezpečný operační systém. Na trhu je k dispozici několik druhů operačních systémů. Mezi nejznámější patří operační systém Windows, Linux a mac OS. Tyto operační systémy jsou učeny především k používání běžnými uživateli. Ovšem jsou k dispozici i serverové varianty operačních systémů. Tyto varianty nejsou určeny k běžnému užití, ale k nasazení na servery, které poskytují služby svým uživatelům. Například webový server, který poskytuje přístup k webovým stránkám jednotlivých uživatelů.

Útočník může provádět útok na veškeré úrovně. Proto je nutné zajistit fyzickou bezpečnost a chránit hardware jako takový, dále je nutné zajistit bezpečnost sítě a služeb, používat bezpečný operační systém s bezpečnými aplikacemi a v neposlední řadě bezpečnost lidských zdrojů, viz Obr. 3.1. Bezpečnost je nutné zajistit na všech úrovních.



Obr. 3.1: Jednotlivé úrovně, které je nutné zabezpečit.

V následujících podkapitolách si popíšeme jednotlivé platformy a nejběžnější zranitelnosti, na které je nutné brát ohled při provádění zabezpečení operačního systému. Popis OS Windows, Linux a mac OS v podkapitole 3.1, 3.2 a 3.3 byl čerpán z [25].

3.1 OS Windows

Mezi uživateli velice oblíbený operační systém. Operační systém Windows 1.0 byl uvolněn v roce 1985, jednalo se o 16 bitový operační systém s podporou minimálně 256 kB operační paměti. Toto byl základ pro tvorbu dalších operačních systému firmy Microsoft. Oproti tomu Windows 10 je více uživatelský a víceúlohový (umožňující multitasing) operační systém, jedná se 32 nebo 64 bitový OS s podporou 128 GB operační paměti a maximální kapacitou disku nad 2 TB [26], [27].

3.1.1 Uživatelská hesla

Ukládání uživatelských hesel probíhá ve formě hashe. Dříve k ukládání hesel sloužil LM hash o délce 128 b využívající algoritmus DES k torbě klíčů. Hlavní slabinou je rozdělení hashe na 2 části a z toho každá o velikosti 7 znaků (převedeno do znaků malé abecedy). Tento hash se již podařilo prolomit za 5,3 sekundy [28]. Následníkem je NT hash, jehož cílem je napravit slabiny LM hashe. Výstupem je 128 b řetězec založený na MD4. Poprvé aplikovaný do verze Windows NT, uvolněn v roce 1993.

Uživatelské účty se lze pokoušet prolomit několika metodami. Mezi tři nejběžnější patří:

a) Útok hrubou silou,
b) Použití slovníkového útoku,
c) Extrakce hashů a aplikace Rainbow tabulek.

V případě použití hrubé síly lze prolomit každé heslo, ovšem hlavním kritériem je čas. Tento čas totiž může být v hodnotě pár sekund až po roky, desítky let, v nejlepším případě bude čas nepolynomiální. Doba lámání závisí na struktuře a délce hesla a také na výkonnosti lámacího stroje. V případě slovníkového útoku se jedná o statisticky nejvíce frekventovaná hesla, která jsou uložena v souboru, tzv. slovníku a lámací stroj se systematicky pokouší zadávat hesla a pokouší se nalézt shodu. V tomto případě závisí na frekventovanosti hesla a kvalitě slovníku. V posledním případě získáme hashe uživatelských hesel a porovnáváme je s hashi v rainbow tabulce. Tato tabulka je založena na principu několika zřetězených hashovacích a redukčních funkcí. Jinými slovy se vytváří posloupnost určitého počtu hashů a tabulka zaznamenává pouze vstup a poté až i-tý hash jako výstup. Hash z uživatelského hesla porovnáváme s pravým sloupcem Rainbow tabulky a v případě jeho výskytu známe vstup, pomocí kterého se k němu lze dopracovat potupným hashováním známého vstupu. V případě, že tento uživatelský hash v tabulce nenalezneme, tak uživatelský hash zhashujeme a hledání zopakujeme. Jedná se o alternativu mezi výpočetní a prostorovou náročností. Další velkou výhodou je to, že tyto tabulky mohou být předpočítány předem [29].

3.1.2 Souborový systém a šifrování

Operační systém Windows používá dva typy souborových systémů, souborový systém FAT a NTFS (New Technology File System). Mezi největší výhodu souborového systému FAT patří to, že je podporovaný téměř všemi OS. Souborový systém NTFS byl poprvé použit u verze Windows NT a stal se primárním souborovým systémem od verze Windows XP. Jeho výhodnou je využívání řízení přístupu k jednotlivým souborům a podpora šifrování.

Pro šifrování disku je využita aplikace BitLocker, která umožňuje šifrovat celé oddíly na disku. Pro šifrování je zvolena symetrická kryptografie konkrétně šifrovací standart AES, v defaultním nastavení jsou zvoleny klíče s délkou 256 b. Tyto klíče jsou později zašifrovány veřejným klíčem uživatele a obnovovacího agenta. Je zde tedy kombinována symetrická a asymetrická kryptografie.

Další možností, jak šifrovat v OS Windows je použití šifrovacího systému EFS (Encrypting File System). Tento systém umožňuje šifrovat soubory a složky. Samotná data jsou šifrována symetrickou šifrou AES, DES, nebo 3DES. K šifrování je použit náhodný klíč FEK (File Encryption Key), tento klíč je následně zašifrován pomocí veřejného klíče uživatele. K dešifrování je použit soukromý klíč uživatele, který je chráněn uživatelským heslem. Následně je pomocí něj dešifrován FEK, pomocí kterého je dešifrován obsah. Tato funkce pracuje pouze na souborovém systému NTFS. V případě že se takto šifrovaný obsah přesouvá do jiného souborového systému, je obsah uložen v dešifrované podobě.

3.1.3 Windows a síť

OS Windows dělí sítě do tří kategorií, možnost volby se zobrazí při prvním připojení do sítě.

a) Domácí síť
   Ostatním zařízením je důvěřováno.
   Sdílení PC a zjištění PC jsou povoleny.
   Domácí sdílení je povoleno.
   Pracovní skupina je povolena.
b) Pracovní síť
   Sdílení PC a zjištění PC jsou povoleny.
   Domácí sdílení není povoleno.
   Pracovní skupina je povolena.
c) Veřejná síť
   Sdílení PC a zjištění PC jsou nepovoleny.
   Domácí sdílení nepovoleno.
   Pracovní skupina nepovolena.

Z rozdělení vyplývá, že pokud síti nedůvěřujeme, je vhodné zvolit „Veřejná síť“. Na tomto rozhodnutí závisí následná aplikace pravidel Brány Windows Firewall.

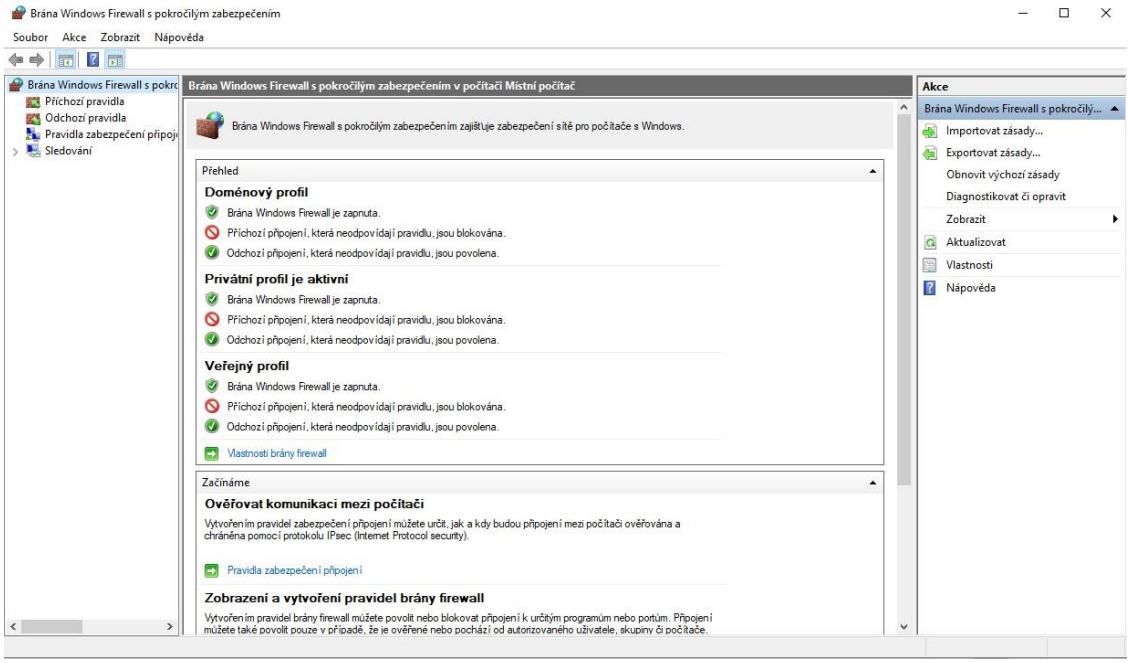
Opět se zde vyskytují dělení do tří profilů. Ukázka na Obr. 3.2

a) Doménový profil v případě aplikování Firemní sítě.
b) Privátní profil v případě aplikování Domácí sítě.
c) Veřejný profil v případě aplikování Veřejné sítě.

Brána Firewall se skládá z příchozích a odchozích pravidel. Tyto pravidla jsou přiřazována k jednotlivým profilům. V seznamu pravidel se vyhledá shoda pro momentální případ (např. se pokoušíme spustit vzdálenou plochu).

Postupuje se podle pravidla:

a) Přepsat blokování (Authenticated Bypass).
b) Blokovat.
c) Povolit.
d) Provést výchozí akci profilu.



Obr. 3.2: Nastavení výchozí politiky jednotlivých profilů brány Firewall, OS Windows.

Výchozí politiky jednotlivých profilů jsou zobrazeny v Tab. 3.1:

Tab. 3.1: Nastavení brány Firewall, OS Windows.

Doménový profil		Privátní profil		Veřejný profil	
Příchozí provoz	BLOKOVAT	Příchozí provoz	BLOKOVAT	Příchozí provoz	BLOKOVAT
Odchozí provoz	POVOLIT	Odchozí provoz	POVOLIT	Odchozí provoz	POVOLIT

3.2 OS Linux

Operační systém Linux se ve velké míře vyznačuje myšlenkou OS pod licencí GNU/GPL, tzn. volného šíření, možnost úprav jádra a jeho nástaveb, ovšem vždy je nutné uvést autory předchozí verze. Tento systém se dělí do mnoha variant systému, tzv. distribucí, mezi nejznámější patří Fedora, Ubuntu, Debian apod. OS Linux se vyznačuje velkou možností nastavení a systémovou nenáročností, proto je také často instalován na servery. Nejen v případě serverového uplatnění je často grafická nástavba vynechána a pro interakci s operačním systémem je využíváno příkazové řádky, tzv. terminálu.

OS Linux využívá přístupová práva k jednotlivým souborům. Je využíván formát rwx rwx rwx, první trojice označuje práva, kterými disponuje vlastník souboru, druhá trojice náleží skupinovým právům k souboru a poslední trojice opravňuje ostatní. Je používáno označení UGO (User, Group, Others). Trojice r-w-x označuje práva ke čtení, zápisu a spuštění souboru. Pokud se jedná o adresář, začíná těchto devět znaků písmenem „d“ ve významu adresář (directory). V případě že není právo přiděleno, je místo jednotlivého znaku uvedena pomlčka.

3.2.1 Uživatelská hesla

V OS Linux jsou informace o uživatelských účtech ukládány do /etc/passwd ve struktuře:

uživatelské jméno : x : uživatelský identifikátor : uživatelské informace : cesta k domovskému adresáři : uživatelský shell

Hesla jednotlivých uživatelů jsou ukládána do etc/shadow, ukládání je provedeno ve formě hashů. Jednotlivé hashe se liší (SHA-512, SHA-256, Blowfish, MD5). OS Linux je posílen proti použití rainbow tabulek pomocí přidání soli. Výsledný hash je tedy složen jako hash z uživatelského hesla a soli, která je pro každého uživatele odlišná.

Uživatelské účty se lze pokoušet prolomit několika metodami. Tři nejběžnější jsou:

a) Útok hrubou silou.
b) Použití slovníkového útoku.
c) Snaha o získání administrátorského účtu.

3.2.2 Souborový systém a šifrování

OS Linux není pevně spojen s určitým souborovým systémem, ovšem nejčastěji je využíváno ext3, ext4, swap. Momentálně používaným souborovým systémem je ext4. Ext3 svého předchůdce ext2 obohatil o žurnálování.

Žurnálování je určeno k ochraně dat v případě, že došlo k neočekávanému odpojení napájení PC, pádu operačního systému atp. Žurnálování typicky využívá buffer, do kterého jsou popsány změny, které se na disku mají provést a teprve poté, až jsou popsány veškeré změny, započne jejich skutečná realizace na disku a po provedení jsou tyto záznamy označovány za splněné a po dokončení je záznam odstraněn. V případě, že došlo k přerušení nebo pádu systému a akce nebyla dokončena, je možné operaci dokončit, nebo se vrátit do stavu před prováděním změn.

Dalším souborovým systémem je swap. Ovšem tento souborový systém není přímo určený k trvalému ukládání dat, ale jen jako odkládací místo. Kde jsou data dočasně odložena.

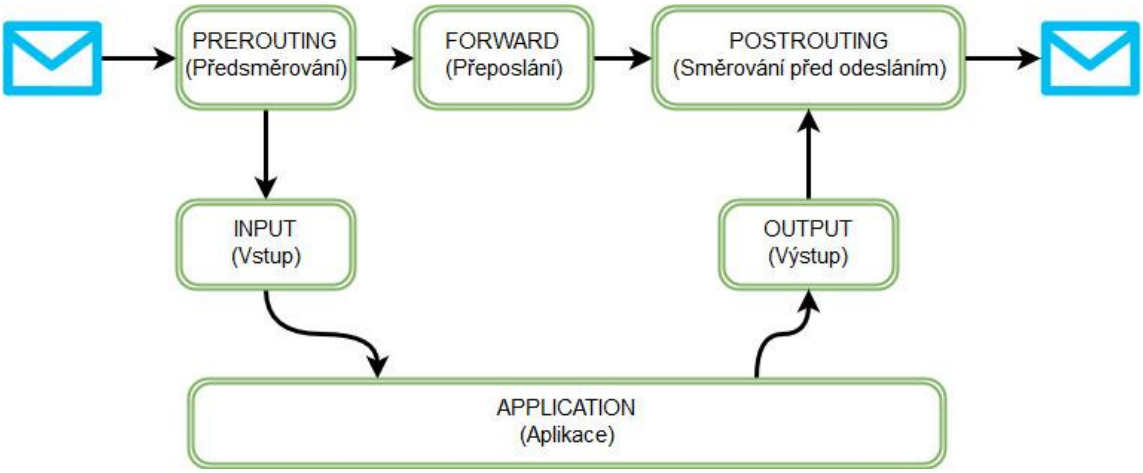
Šifrování je prováděno subsystémem dm-crypt. Jedná se o transparentní šifrování celého disku kromě zavaděče systému a samotného jádra systému nebo pouze oddílu. Výběr šifrování lze typicky zvolit již při instalaci OS. Tento subsystém podporuje módy XTS (XEX-based tweaked-codebook mode with ciphertext stealing), LRW a další, ovšem pro operační systém je doporučeno používat mód XTS (LRW).

3.2.3 OS Linux a síť

Nastavení jednotlivých parametrů je uloženo do etc/network/interfaces. Je zde označení síťového rozhraní, parametr, zda je IP adresa a ostatní parametry získávány pomocí DHCP, nebo zda je záznam statický.

Dále lze využívat skriptů, které se mají spustit před/po aktivaci a před/po deaktivaci rozhraní.

Pro filtraci paketů, je využívána součást jádra, netfilter, viz Obr. 3.3. Tento nástroj umožňuje velmi podrobné zacházení s pakety a jejich řazení do kategorií a lze je definovat pomocí příkazu iptables.



Obr. 3.3: Netfilter, OS Linux.

Řetězec pravidel PREROUTING zahrnuje postupy, které se mají provést před samotným vstupem paketu do systému, tedy například destination NAT, poté se rozhoduje, zda má paket vstoupit do tohoto zařízení, pokud ne, je předán do řetězce FORWARD, zde mohou proběhnout úpravy paketu a následně paket přechází do řetězce pravidel POSTROUTING, kde jsou aplikovaná poslední pravidla před tím, než paket opustí systém.

Pokud má paket vstoupit do tohoto zařízení, dojde k filtraci paketu v tabulce FILTER v řetězci INPUT. Pokud filtrace nalezne odpovídající pravidlo je okamžitě provedeno, tzn. postupuje se podle jednotlivých záznamů od shora dolů, dřívější shoda má tedy přednost a je provedena před jinými shodami. Pokud se žádná shoda nenalezla, provede se výchozí politika daného řetězce. Jednotlivé politiky mohou být definovány jako ACCEPT, tedy paket je přijat, popřípadě LOG, paket je zalogován do systému, REJECT, kdy je paket zahozen, ale původci zprávy je odeslána odpověď, že cíl je nedosažitelný a posledním pravidlem je DROP, kdy je paket smazán, aniž by kontaktoval odesílajícího zprávy.

Jednotlivé záznamy v tabulce mohou být definovány nejen pomocí paketového filtru, tedy pomocí IP adres, ale také pomocí stavu spojení, tedy zda byla komunikace zahájena zvenčí, nebo komunikaci zahájilo přímo toto zařízení. Dále je možné provádět kontrolu pomocí jednotlivých údajů v hlavičkách používaných protokolů a další.

Velmi vhodným rozšířením jádra je použití SELinuxu. Hlavním účelem je zamezit zneužití systému i tehdy, kdy došlo k prolomení bezpečnosti superuživatele. Bezpečnost je prováděna tak, že jsou používány tzv. domény a žádný uživatel, ani program nemůže vykonávat funkci mimo tuto doménu.

3.3 Mac OS

Operační systém mac OS je uživatelsky velice oblíbený pro svou podporu napříč celým spektrem zařízení, ať se jedná o osobní počítač, tablet, nebo o nositelné zařízení. Tento OS je vyvíjen společností Apple. První počítač s názvem Apple I byl uvolněn v roce 1976 a celá společnost je známá svým logem nakousnutého jablka.

Společnost volí politiku prodeje OS společně se svým hardwarem, tudíž nelze zakoupit OS odděleně a nainstalovat na běžný, nijak neupravený PC. Samotný systém je založený na Unixu, jsou si tudíž funkcionálně velice podobné. Momentálně uvolňované verze jsou jedině 64 bitové.

3.3.1 Uživatelská hesla

Uživatelská hesla jsou vždy ukládána ve formě hashe, metody provedení se však s každou verzí odlišují. Možnou zranitelností je však v případě kompatibility s OS Windows, a to přesně v případě povoleného sdílení pomocí protokolu SMB. Ve verzi 10.12 je aplikován mechanismus ukládání uživatelských hesel pro každého uživatele odděleně v tzv. plistu. Hashe jednotlivých hesel jsou vytvářeny pomocí SHA-512, která probíhá v přesně definovaném počtu iterací. Pro zesílení a odolnosti vůči rainbow tabulkám jsou hashe vytvářeny z uživatelského hesla a přidané soli.

Jak již bylo zmíněno, každý uživatel má vlastní plist, ke kterému má přístup povolen pouze superuživatel. Tento soubor nese informace o použitém typu hashe, počtu iterací a požité soli. Použité zabezpečení je robustní a bez znalosti plistu je možný pouze útok hrubou silou.

3.3.2 Souborový systém a šifrování

Nativním souborovým systémem mac OS je HFS+ (Hierarchical File System), volitelnými jsou však i další souborové systémy jako FAT32, popřípadě exFAT (FAT64). HFS+ se vyznačuje maximální velikostí souboru 8 EB, využívá žurnálování, podporuje řízení přístupu k souborům a transparentní šifrování. Řízení přístupových práv je totožné s OS Linux.

K zajištění samotného šifrování dat na disku byla vyvinuta služba FileVault. Tato služba slouží k zašifrování celého oddílu s OS s použitím šifry AES v XTS módu s dvěma klíči derivovanými z uživatelských hesel s délkou 128 b spolu s recovery klíčem pro případ zapomenutí hesla.

Mód XTS byl vyvinut za účelem šifrování disků, slouží pouze k zajištění bezpečnosti, integritu a autentičnost dat již nezajišťuje.

3.3.3 Mac OS a síť

Stejně jako většina OS i mac OS umožňuje nastavení jednotlivých síťových parametrů manuálně, nebo pomocí DHCP.

Nastavení nativního firewallu pomocí grafického rozhraní je velice omezené a neumožňuje vytvářet svá vlastní pravidla k filtraci síťového provozu. Ovšem firewall je převzat z FreeBSD, tudíž je umožněna konfigurace pomocí příkazové řádky s použitím příkazu pfctl a ta již umožňuje tvorbu vlastních pravidel.

3.4 Srovnání operačních systémů Windows, Linux, mac OS

Nyní si porovnáme operační systémy z pohledu uživatele a jejich bezpečnosti. Každý systém má svoje přednosti a své nedostatky.

3.4.1 Uživatelská hesla

Heslo každého uživatele systému musí být chráněno a prolomení autentizace by mělo být znesnadňováno různými mechanismy. Jak znázorňuje Tab. 3.2. Z hlediska bezpečnosti je nebezpečnějším operačním systémem mac OS. Důvodem je použití robustního zabezpečení. Hash s přidáním tzv. soli je odolný vůči použití Rainbow tabulek. Důvodem je neznalost této soli, útočník si není schopen předem vypočítat tyto tabulky. Navíc se tento celý postup provádí ve více iteracích. To zaručuje ještě vyšší bezpečnost. Na druhém místě s bezpečností uživatelských hesel je OS Linux, opět z důvodu přidání soli do hashe.

Tab. 3.2: Porovnání ukládání uživatelských hesel.

Uživatelská hesla			
OS	Windows	Linux	Mac OS
princip uložení už. hesel	NT HASH (používá MD4)	SHA 256/512 + sůl	SHA 512 s iteracemi + sůl
uložení	registry, dostupné jen při inicializaci OS (Winlogon)	etc passwd, etc shaddow dostupné jen administrátorovi	plist pro každého uživatele, dostupné jen administrátorovi
útok	Hrubou silou (Brute Force) Slovníkový útok Rainbow tables	Hrubou silou (Brute Force) Slovníkový útok	Hrubou silou (Brute Force) Slovníkový útok

3.4.2 Souborový systém

Velmi důležitou částí OS je souborový systém. Výhoda OS Linux je v používání souborového systému ext3, popř. ext4, který umí spolupracovat se všemi ostatními souborovými systémy na rozdíl od souborového systému NTFS. Nevýhodou OS Linux je nepodpora Plug & Play, tedy automatické nalezení ovladače pro připojené zařízení.

Z hlediska kryptografie, OS Windows s nástrojem Bitlocker neumožňuje aplikovat kryptografické mechanismy na složky, šifrování je dovoleno pouze pro logické oddíly disku. Pro šifrování disku je zvoleno šifrování AES-256. Souborový systém Linux požívá nástroj dm-crypt, pro tvorbu kryptografických kontejnerů, šifrování celého disku, včetně SWAP oddílu jedinou výjimkou je oddíl boot. Pro šifrování disku je zvoleno šifrování AES-256. Mac OS disponuje nástrojem FileVault, pro šifrování celého diskového oddílu je použito šifrování AES-128. Klíče pro toto šifrování jsou získávány derivací z uživatelských hesel.

Z hlediska bezpečnosti je velice obtížné zvolit OS, který má vůči jiným výhodu v souborovém systému. Porovnání viz Tab. 3.3.

Tab. 3.3: Porovnání souborových systémů.

Souborový systém			
OS	Windows	Linux	mac OS
používané souborové systémy	FAT, NTFS (nativní)	ext3, ext4 (nativní), swap (pro odkládací prostor)	HFS+, FAT 32, FAT 64
kryptografie	EFS Bitlocker	dm-crypt	FileVault
Plug & Play	ANO	NE	ANO

3.4.3 Operační systém a síť

Pro operační systém je velmi důležitá ochrana proti průniku z vnějšího prostředí. Proto operační systém Windows disponuje nástrojem Brána Windows Firewall. Nástroj přistupuje odlišně k síťovému provozu vzhledem definovanému profilu sítě. Je umožněno vytvářet vlastní pravidla. Operační systém Linux umožňuje velmi citlivě definovat síťový provoz, který vstupuje/prochází/vystupuje ze systému. Díky tomuto nástroji tento OS vyniká nad ostatními. Mac OS v nativním stavu disponuje pouze omezenými možnostmi, jak filtrovat síťový provoz. Tohoto lze dosáhnout jen pomocí dalších nástrojů. Porovnání viz Tab. 3.4.

Tab. 3.4: Porovnání z hlediska přístupu k síťovému provozu.

Operační systém a síť			
OS	Windows	Linux	mac OS
Firewall	3 kategorie profilů sítě (DOMÉNOVÝ, PRIVÁTNÍ, VEŘEJNÝ)	3 řetězce (INPUT, FORWARD, OUTPUT)	definování akcí pro instalované aplikace
		lze definovat odpověď (DROP, REJECT)	lze rozšířit pomocí pfctl
definování vlastních pravidel	ANO	ANO	NE (v grafickém režimu)

Zvolení nejbezpečnějšího operačního systému není jednoduché. Záleží na zvoleném pohledu a kritériích. V tomto srovnání se jeví operační systémy podobné. Nyní se zaměříme na „nedostatky“. Mac OS je znám svým důrazem na bezpečnost, ovšem nedostatkem je absence dostatečného provedení firewallu (v běžném GUI). Operační systém Windows je velice rozšířený a používaný, ovšem nedostatkem je méně efektivní zabezpečení uživatelských účtů. Operační systém Linux je také rozšířený, ovšem není pro běžného uživatele tolik uživatelsky přívětivý jak ostatní OS. Je nutné ovšem zmínit, že nabízí velké množství efektivních nástrojů.

4 BEZPEČNOSTNÍ SCÉNÁŘ

K zajištění bezpečnosti je nutné využít velké množství nástrojů a dodržovat určité zásady. V této části se zaměříme na počítač, a ne na jeho uživatele, který se svým chováním vysokou měrou podílí na výsledné úrovni bezpečnosti.

Další postup bude rozdělen na dva celky, a to na osobní počítač a server. Rozdělení je provedeno pro demonstraci různých přístupů k zabezpečení.

4.1 Bezpečnostní scénář osobního počítače s OS Windows

Tento operační systém byl vybrán z důvodu, že je mezi uživateli velmi hojně využíván, a proto je útočníky velmi často vybírán, jako možná cesta, jak dosáhnout svého nekalého cíle.

Aby byl OS opravdu bezpečný, je vhodné použít k přihlašování a k veškerým aplikacím silné a odlišné heslo. Dále je nutné udržovat OS a aplikace neustále aktualizované, protože právě toto je možné riziko průniku. Aktualizace slouží nejen k vylepšení funkcí aplikace, ale také k opravě zjištěných bezpečnostních rizik. Také je vhodné využít možnost šifrování (souborů a složek, popřípadě celých oddílů). Dalším prvkem je zapnutý a správně nakonfigurovaný firewall. Pravidelně aktualizovaný a aktivní antivirus. A posledním prvkem jsou další aplikace, na které bude tato kapitola také zaměřena, viz Obr. 4.1.

Obr. 4.1: Základní prvky zajišťující bezpečnost OS.

4.1.1 Bezpečné heslo

Základem bezpečného systému jsou bezpečná hesla. Proto je vhodné volit své hesla taková, aby dávala smysl pouze svému vlastníkovi. Například lze vytvořit velmi snadno zapamatovatelné heslo. Například heslo „20#FeVUTkT18*“ není na první pohled zapamatovatelné. Je složeno z roku „2018“, který je rozdělen na půl, dále je zde „VUT“ ve středu, které obklopuje „FekT“. Pro zesílení je přidán speciální znak mřížky a hvězdičky. Ovšem daleko silnějším heslem je „Červ*v180#b45!“. Toto heslo nedává snad žádný smysl. Ovšem když jako majitel tohoto hesla víme, že máme rádi červenou barvu (Červ), jsme vysocí (v) 180 cm (180) a velikost bot (b) je 45 (45) a tyto znaky oddělíme znakem hvězdičky, mřížky a na konec dodáme vykřičník. Poté je toto heslo velmi těžce prolomitelné. Navíc si můžeme být jisti, že toto heslo nespadá do žádného slovníku nejpoužívanějších hesel.

4.1.2 Aktualizace

Pravidelná aktualizace nejen operačního systému je jeden ze základních kamenů bezpečnosti systému. Je nutné si uvědomit, že právě nezabezpečené aplikace umožňují útočníkům použít exploit pro narušení bezpečnosti počítače. Tomuto se dá předejít pravidelným aktualizováním operačního systému a instalovaných aplikací.
Pokud se zaměříme na aktualizaci operačního systému jako takového, tak OS Windows nabízí možnost vyhledání dostupných aktualizací systému a jeho součástí pomocí „Windows update“. Ten lze nalézt v nastavení systému, jako první položku v levém menu. Samotné vyhledávání je plně automatizované, popřípadě lze vynutit kontrolu možných aktualizací stisknutím tlačítka „vyhledat aktualizace“.

V případě, že byly aktualizace nalezeny, provede se jejich stažení do systému a pokud je nutný restart, tak počítač automaticky počká na dobu mimo „dobu aktivního užívání“, kterou lze v systému nastavit. Cíl je takový, aby aktualizace nebyly notifikovány během běžného používání a nerušily uživatele při práci na počítači, ale bylo na ně upozorněno během doby, kdy počítač neslouží k práci jako takové. Po upozornění lze následný restart odložit, popřípadě souhlasit s restartováním počítače a provedení nutných změn.

U ostatních aplikací je také vhodné pravidelně kontrolovat dostupnost možných aktualizací. Ve většině případů se o toto stará aplikace sama, pokud jste při instalaci zatrhli možnost „automaticky vyhledávat aktualizace“, popřípadě udělili přímo svolení k jejich instalaci, bez nutnosti svolení jednotlivých aktualizací jednotlivě.

4.1.3 Šifrování

Šifrování souborů a složek v OS Windows je velmi jednoduché díky existenci EFS (Encrypting File System), který zajišťuje proces šifrování. Nejprve je nutné vybrat složku, která má být šifrována. Po kliknutí pravým tlačítkem zvolíme možnost „upřesnit“ v záložce „obecné“. Poté zvolíme možnost „Šifrovat obsah a zabezpečit tak data“.

Dále je třeba exportovat soukromý klíč uživatele, aby mohlo dojít k dešifrování složky i na jiném počítači. Je možné, že operační systém sám přes vyskakovací okno, nabídne zálohu soukromého klíče. Popřípadě, lze zvolit „ruční“ variantu a přes možnost „spustit“ (klávesa Windows + R) a zadání řetězce „certmgr.msc“ dojde k otevření okna s certifikáty. V kartě „Osobní“ a v objektu „certifikáty“ lze kliknout na certifikát uživatele, který lze přes pravé tlačítko exportovat výběrem „možnosti“, poté „Všechny úkoly“. V průvodci je nutné zvolit „exportovat i soukromý klíč“.

V případě zvolení exportu klíče vyskakovacího okna se zobrazí dotaz, zda chceme zálohovat certifikát a soukromý klíč nyní. Dále je možné upravit formáty, ve kterých bude certifikát exportován. Posledním krokem je zadání hesla pro ochranu soukromého klíče. Dále zvolíme místo, kde bude certifikát uložen a následně dojde k exportu certifikátu. Poté jen přeneseme certifikát bezpečnou cestou na místo, kde má být možné takovouto složku dešifrovat a certifikát zde importujeme.

Nyní se zaměříme na šifrování nesystémových disků pomocí nástroje BitLocker. Postup je opět velice jednoduchý. Nejprve klikneme pravým tlačítkem myši na disk, který chceme šifrovat a vybereme možnost „Zapnout nástroj BitLocker“. Následně vybereme,

zda chceme jednotku odemykat pomocí čipové karty, nebo pomocí hesla. Dále zvolíme místo, kam zálohovat obnovovací klíč. Poté je nutné zvolit, zda požadujeme šifrovat pouze využité místo na disku, nebo zda chceme šifrovat celou jednotku. Poslední volbou je výběr režimu šifrování. Dále jen zahájíme šifrování a tím spustíme proces šifrování.

4.1.4 Firewall

Brána Windows Firewall dovoluje podrobné zacházení se síťovým provozem. Existují dva přístupy, pomocí kterých lze přistupovat k nastavování brány firewall.

Přístupy brány Windows Firewall:

a) Negativní model.
b) Pozitivní model.

Negativní model přistupuje k nastavení brány firewall způsobem, že doporučuje veškerý provoz povolit a vytvořit tzv. „black list“. Na black listu jsou uvedeny známé hrozby (např. známé IP adresy útočníků). Tato možnost ovšem dokáže čelit pouze známým a již definovaným hrozbám. Opačným modelem je pozitivní model. Tento model využívá tzv. „white list“, na tomto listu jsou uvedeny pouze požadované služby (např. aplikace s odpovídajícím portem). Tento model je vhodnější z důvodu, že máme naprostou kontrolu nad probíhajícím síťovým provozem. Ovšem je zde nutná investice času na vytvoření potřebných pravidel.

V případě, že používáme pozitivní model, je třeba změnit pravidlo odchozího pravidla na stav s názvem „blokovat“. Je ovšem nutné opět zmínit, že v případě změny je firewallem propouštěn pouze výslovně povolený síťový provoz. Zbytek síťového provozu není firewallem propuštěn. Následně je tedy třeba začít vytvářet nová pravidla a umožnit tak OS komunikovat jen definovaným způsobem.

Brána Windows Firewall s pokročilým nastavením nabízí možnost lze nastavit, zda chceme zobrazit příchozí, nebo odchozí pravidla (umístěno v levém panelu). Poté na v pravém panelu lze vybrat akci „vytvořit nové pravidlo“. Pomocí těchto pravidel definujeme, který síťový provoz bude propuštěn. Dále lze vybrat, zda pravidlo, které vytváříme, bude pro program, port, nebo použijeme před definovaných pravidel, popřípadě je umožněno vytvářet pravidla vlastní.

Mezi pravidla, která je vhodná přidat do povoleného provozu patří například služby HTTP (HyperText Transfer Protocol), HTTPS (HTTP Secure), DHCP (Dynamic Host Configuration Protocol), DNS (Domain Name System), TCP (Transmission Control Protocol) typicky pouze navázané ze strany klienta, ICMP (Internet Control Message Protocol), FTP (File Transfer Protocol) atd. Dále je dobré povolit provoz e-mailovým klientům, tedy POP3 (Post Office Protocol), IMAP (Internet Message Access Protocol), SMPT (Simple Mail Transfer Protocol). Dále je třeba umožnit komunikaci používaným aplikacím, popřípadě povolit další služby.

V případě, že bychom používali pozitivní model, tak bychom museli zakázat vše mimo služeb zmíněných v předchozím odstavci. Je tedy jasné, že by tato konfigurace byla velmi náročná.

4.1.5 Antivirus

OS Windows nativně disponuje antivirovým programem s názvem „Windows Defender“. Ten ovšem poskytuje pouze určitou možnost kontroly nad tím, co a jakým způsobem je chráněno. Je možné tuto ochranu vypnout a nainstalovat na cílový operační systém jiný antivirový program od jiné společnosti.

Obecně je doporučeno:

a) Jednotlivé komponenty zabezpečení mít ve stavu aktivní.
b) Pravidelně, nejlépe automatizovaně, provádět aktualizaci signatur virů.

Některé komerční antivirové prostředky umožňují naleznout programy, které mohou být zranitelné, tedy je na ně znám exploit. Také lze detekovat nežádoucí doplňky internetových prohlížečů, naleznout slabá hesla apod. Dále je možné aktivovat funkce, které pracují jako štít webové kamery. Tedy integrovaná kamera je aktivována pouze se svolením uživatele. Programy také mohou „hlídat“ aktualizace programů nainstalovaných na PC a automaticky provádět jejich aktualizace a mnoho dalších funkcí [30].

Windows Defender umožňuje provedení kontroly ve třech stupních kontroly. Záložka „Aktualizovat“ umožňuje provedení aktualizace signatur. Vypnutí „Windows Defenderu“ je trochu komplikovanější. Je nutné kliknout na nastavení, poté jsme přesměrováni do nastavení, kde zvolíme možnost „Otevřít Centrum zabezpečení v programu Windows Defender“. Následně zvolíme možnost „Ochrana před viry a hrozbami“. Poté zvolíme možnost nastavení ochrany před viry a hrozbami. Zde lze zapnout/vypnout ochranu v reálném čase a cloudovou ochranu.

4.1.6 Aplikace zvyšující bezpečnost

Pomocí přídavných aplikací lze zvýšit bezpečnost systému. Těchto aplikací existuje celá řada. Pro monitorování datového toku lze využít placenou aplikaci „NetWorx“, která umožňuje třiceti denní zkušební verzi. Dále je také možné využít aplikaci „PRTG network monitor“.

NetWorx

Aplikace NetWorx [31] je kompletní nástroj pro správu sítě umožňující velkou škálu možných nastavení a přizpůsobení potřebám každého jednotlivého uživatele, umožňuje instalaci v českém jazyce. Po instalaci se ikona zobrazí na hlavním panelu a umožňuje uživateli podrobné nastavení. Lze nastavit uložiště, do kterého budou ukládány jednotlivé reporty, lze nastavit upozornění v případě, že nastala určitá událost, např. překročen běžný limit v datovém přenosu po síti. Na tuto událost lze nastavit i akci prostřednictvím upozornění uživateli, spuštění určitého programu, popřípadě celý počítač vypnout, restartovat apod. Poté lze zobrazovat grafy ve formě denního, týdenního, nebo měsíčního srovnání. Celé nastavení lze uzamknout pomocí hesla.

Tato aplikace může napomoci při vytváření představy, kolik datových jednotek uživatel běžně využije za daný časový úsek, následně je vhodné vytvořit si určitý datový

limit, který po přesáhnutí upozorní uživatele na anomálii. Tato anomálie může znamenat, že cílový operační systém byl napaden nežádoucím softwarem, který komunikuje s útočníkem, odesílá data apod.

PRTG network monitor

Webová aplikace PRTG network monitor [32] je aplikace, kterou lze instalovat ve verzi freeware, popřípadě trial verzi, která v prvních třiceti dnech nabízí využít neomezeného množství senzorů, pomocí kterých je cílený systém monitorován.

Tato aplikace umožňuje definovat jednotlivé senzory, které „dohlíží“ na daný operační systém. V případě, že senzor detekoval anomálii, např. že na operačním systému nejsou instalovány nejnovější aktualizace OS, jsme notifikování pomocí „alarmu“. Tento systém umožňuje provádění monitorování velkého spektra aspektů. Lze monitorovat mimo jiné kontrolu provádění pravidelných záloh systému, monitorování sítě, ve které jsou „objeveny“ jednotlivé komponenty, monitoring datového toku. Lze monitorovat dostupnost služeb/zařízení pomocí provádění pravidelného příkazu „Ping“, nebo využití procesoru.

Při přidávání jednotlivých senzorů jsme vyzvání k výběru zařízení, na které má být senzor aplikován. Například, zda se má senzor zaobírat připojením k internetu, nebo lokálním zařízením.

Instalace je velice jednoduchá. Po do končení standartní instalace jsme uvítání průvodcem, který nás provede aplikací PRTG. V tomto seznámení s aplikací jsme vyzvání k zadání přístupových údajů k operačnímu systému. Poté v záložce „setup“ zvolíme možnost „Account settings“, dále „My Account“, kde nastavíme přístupové heslo k této webové aplikaci. Následně dojde k restartu aplikace a je navázáno SSL spojení, u něhož je nutné souhlasit s certifikátem.

Další postup už závisí na samotném uživateli. Běžné senzory jsou aktivní a systém je postupně monitorován a zobrazují se senzory s několika statusy. Nejzávažnějším je status „Down“, mezi které se řadí například nedostatek místa na disku. Dalším typem je statusu je „Warning“, mezi které patří například „Windows update status“, který nás upozorňuje na nenainstalované systémové aktualizace. Posledním typem je status „Up“, který zobrazuje senzory, které jsou aktivní a odpovídají definovaným kritériím.

4.2 Bezpečnostní scénář pro server

Server je v oblasti poskytování služeb nezbytnou součástí. Jedná se o výkonný stroj, který poskytuje různé služby svým klientům. Server může být zaměřen na určitou oblast, například webový server, aplikační, souborový, doménový server a mnoho dalších.

Z důvodu poskytování služeb je zjevné, že je mnohdy nezbytné, aby takový stroj disponoval vyšším výpočetním výkonem. Tento výkon je dělen mezi jednotlivé klienty, které žádají přístup ke službám. Server často poskytuje více služeb v jednom časovém okamžiku, a proto je zde velký důraz na stabilitu systému.

Mezi nejznámější serverové varianty systému se řadí „Windows server“ a server využívající distribuce „Linuxu“. Ovšem častěji nasazovaným systémem je určitá distribuce „Linuxu“. Hlavní výhodou tohoto systému je jeho nízká systémová náročnost

na výpočetní výkon, jinými slovy výkon je více zaměřen pro potenciální klienty a pro samotný běh systému je potřeba méně systémových zdrojů než u jiných systémů. Tohoto je zejména dosáhnuto tím, že k uspokojování požadavků jednotlivých klientů není třeba grafická nástavba systému, tedy celý systém je řízen prostřednictvím příkazového řádku, resp. terminálu.

4.2.1 Operační systém

Server k obsluhování jednotlivých požadavků svých klientů využívá operační systém. Jak již bylo zmíněno, výběr tohoto operačního systému může být velice zásadní z důvodu, že každý operační systém nakládá odlišně s výpočetním výkonem. Následující scénář bude orientovaný operační systém „Ubuntu“, což je jedna z distribucí Linuxu, z důvodů jeho nízké náročnosti na výpočetní výkon a jeho častého serverového nasazení.

4.2.2 Bezpečné heslo

Server zpracovává velké množství informací a jím poskytované služby jsou mnohdy pro uživatele velmi důležité. Abychom se vyhnuli pro útočníka snadnému převzetí kontroly nad tímto strojem, je nutné jej dobře zabezpečit.

Bezpečnost v tomto případě začíná u vhodně zvoleného bezpečného hesla. Je jasné, že heslo musí splňovat veškeré bezpečnostní aspekty pro bezpečné heslo. Například heslo „(Q3<p*tzmN#l“ se jeví jako dostatečně silné heslo a jistě není v žádném existujícím slovníku nejpoužívanějších hesel. Ovšem jako správce si nesmíme ulehčovat práci tím, že bychom si heslo napsali někam, kde k němu má přístup kdokoli. Musíme si uvědomit, že znalost tohoto hesla umožňuje disponovat s veškerými daty, se kterými přichází server do styku.

Pokud si bezpečné heslo nechceme vymýšlet sami, existuje řada nástrojů, které takové bezpečné heslo vygenerují. Například pomocí nástroje „pwgen“ [33] si lze bezpečné heslo vygenerovat, lze vybrat i vhodnou délku.
Použití je velmi jednoduché. Stačí si pomocí příkazu „`apt-get install -y pwgen`“ nainstalovat nástroj a poté pomocí příkazu „`pwgen -s 20 2`“ nechat heslo vygenerovat. Parametr „s“ značí, že se mají generovat bezpečná hesla (je těžší si je zapamatovat), první číslo značí, že žádáme, aby délka ve znacích byla 20 a druhé číslo značí, že žádáme vygenerovat dvě hesla. Výsledek vygenerovaného heslo je například „W7bIGJDqZMLVvhlPOEWX“. Ovšem jak si můžeme všimnout, nebyly vygenerovány žádné speciální znaky, je vhodné je do textu zařadit. Proto lze zařadit do příkazu parametr „-y“, který vynutí použití speciálních symbolů. Po zadání příkazu i s tímto parametrem dostaneme heslo například „\DZvr1__L)&6D[N.XW]C“, které je naprosto bezpečné a odolá slovníkovým útokům.

4.2.3 Šifrování

Zda šifrovat, či ne je snad zbytečná otázka. Je třeba naše data udržet v bezpečí. Operační systém Ubuntu umožňuje aplikovat šifrování domovského adresáře přímo z instalačního procesu. Samotné šifrování využívá systém ecryptfs [34], což je kryptografický souborový systém Linuxu. Jednotlivá metadata jsou uložena u každého souboru,

jednotlivé šifrované soubory tak mohou být přesunuty mezi jednotlivými uživateli. Samotné šifrování je založeno na šifře AES, s délkou klíče 256 b.

Je zde ovšem pár věcí, na které je nutné myslet. Aby bylo možné aplikovat toto šifrování, tak se ze systémových důvodů přesunulo umístění uživatelských dat a to na „/home/.ecryptfs/UŽIVATEL/.Private“. Dalším problémem je to, že pokud je šifrování aplikováno, tak vzniká problém při přihlašování přes SSH prostřednictvím certifikátu. Jejich uložení je totiž v domovském adresáři, který není při nepřihlášeném uživateli nijak dostupný. Abychom toto umožnili, je možné přesunout veřejné klíče do nezašifrované části souborového systému a do typického umístění „~/.ssh/authorized_keys“ na něj vytvořit symbolický odkaz, popřípadě pozměnit jejich umístění a pozměnit konfigurační soubor.

Pokud takto učiníme, musíme také zadat přístupovou frázi, kterou budeme používat k dešifrování domovského adresáře při každém startu OS.

4.2.4 Aktualizace

Instalace aktualizací jsou v případě operačního systému Ubuntu velmi jednoduché. Lze použít příkaz „`apt-get update`“, který aktualizuje veškeré verze systémových balíčků. Pokud chceme provést upgrade, tak použijeme příkaz „`apt-get upgrade`“. V případě serveru je však provádění příkazu upgrade lehce nebezpečné. Příkaz „`apt-get upgrade`“ provede stažení dostupných aktualizací a zobrazí je. Poté je nutné odsouhlasit navrhnuté změny a to zadáním „Y“ popřípadě „N“ pro odmítnutí instalací. Bezpečnější variantou v případě provádění tohoto příkazu na serveru je instalace nástroje „aptitude“ [35], který umožňuje práci s balíčky.

Instalace tohoto nástroje je velmi jednoduchá. Příkazem „`sudo apt install aptitude`“ se provede instalace. Dále příkazem „`sudo aptitude safe-upgrade`“ provede bezpečný upgrade. To znamená, že dojde k aktualizaci instalovaných balíčků, může proběhnout i instalace nových, ale nedojde k odinstalaci balíčků. Tudíž veškeré služby zůstávají podporovány a nedojde k odepření služby odinstalací balíčku, který je nutný pro její běh.

4.2.5 SSH

V případě, že chceme se serverem bezpečně komunikovat, tak použijeme SSH [36], [37], [38]. SSH umožňuje také bezpečný přenos souborů pomocí příkazu „scp“.

Abychom mohli tohoto šifrovaného spojení využívat, je třeba jej nejprve nainstalovat. To provedeme příkazem „`sudo apt-get install openssh-server`“. Dále musíme vygenerovat soukromý a veřejný klíč uživatele. Je nutné zmínit, že v serveru, kam se budou přihlašovat jednotliví uživatele, budou v adresáři „~/.ssh/authorized_keys“ uloženy jejich veřejné klíče.

Pro změnu jednotlivých parametrů v nastavení SSH, vstoupíme s oprávněním superuživatele do konfiguračního souboru (za pomocí příkazu nano) „`sudo nano /etc/ssh/sshd_config`“. Zde je možné změnit naslouchající port apod.

K docílení toho, že bude možné se přihlásit pouze za pomocí certifikátu, je nutné v konfiguračním souboru změnit jeden parametr. Změníme tedy hodnotu

„PasswordAuthentication“ z možnosti „YES“ na možnost „NO“. Dále je vhodné zkontrolovat, zda je v konfiguračním souboru umožněno přihlášení pomocí certifikátu „PubkeyAuthentication YES“. Také je vhodné změnit oprávnění ke složce s veřejnými klíči pomocí příkazu „chmod 600 .ssh/authorized_keys“ a „chmod 700 ~/.ssh“. Aby se změny projevily, je nutné provést restart SSH serveru za pomocí příkazu „service ssh restart“.

Samotné klíče lze vygenerovat několika způsoby. V případě, že provádíme generování klíčů z operačního systému Linuxového typu, je možné využít příkazu „ssh-keygen -t rsa“. Po vygenerování provedeme přenos veřejného klíče pomocí příkazu „ssh-copy-id username@remotehost“.

V případě, že chceme generování klíčů provést z operačního systému Windows [39], lze využít nástroje „puttygen.exe“, který je část projektu „Putty“. Při vygenerování zadáme heslo, kterým bude chráněn vygenerovaný soukromý klíč. Dále je nutné veřejný klíč přenést na server. Pro připojování je nutná změna v nastavení „Putty“. V nastavení rozklikneme možnost „Connection“, dále „SSH“ a položku „Auth“. V otevřeném oknu vložíme soukromý klíč a následně je možné se přihlašovat běžným způsobem, jen při přihlašování zadáme heslo k soukromému klíči a přihlášení se provede. Přihlášení se provede přes příkaz „ssh <uživatelské_jméno>@< IP adresa_nebo_název_PC>“. Kopírování souborů na server lze provést přes příkaz „scp <soubor_k_přenosu> <uživatelské_jméno>@<IP_adresa_nebo_název_PC>: <cílový_adresář>“. Přenos souboru ze serveru k uživateli lze provést „scp <uživatelské_jméno>@<IP_adresa_nebo_název_PC>:<cesta_k_souboru>“.

Pokud bylo použito šifrování domovského adresáře, tak je nutné změnit uložiště souboru s veřejnými klíči, například jej lze uložit v „/etc“. Ovšem nesmíme zapomenout pozměnit cestu také v konfiguračním souboru, kde je nutné změnit hodnotu direktivy „AuthorizedKeysFile“.

Při přihlašování prostřednictvím certifikátu závisí veškerá bezpečnost na uživateli, který se k serveru smí přihlásit. Je tedy nutné, aby soukromé klíče, pomocí kterých se na server lze přihlásit zůstali v bezpečí a byly dostatečně zabezpečeny.

4.2.6 IDS, IPS

OS Linux neposkytuje nativní IDS ani IPS. Je tedy nutné využít aplikace třetích stran. Vhodné je například použít IPS s názvem Snort [40]. Jedná se open source systém, který podporuje OS Fedora, FreeBSD, CentOS, Debian, Windows a další. Po provedení instalace je třeba vstoupit do konfiguračního souboru a nastavit parametry pro lokální síť.

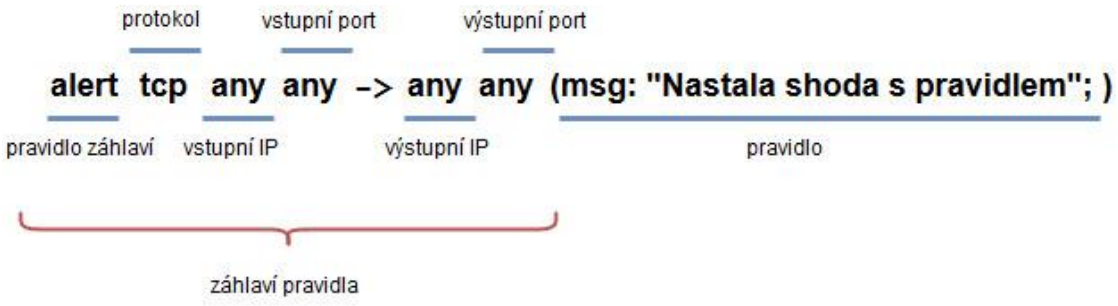
Tento systém umožňuje logovat provoz na základě nastavených parametrů, analyzovat síťový provoz v reálném čase, detekovat pokus o tichý sken portů, odhalení velkého množství útoků mířených na chráněná aktiva, provádění detekce na základě obsahu paketu a další.

Zacházení s pakety, pravidla záhlaví [40]:

a) Pravidlo alert

b) Pravidlo log.
c) Pravidlo pass.
d) Pravidlo activate.
e) Pravidlo dynamic.
f) Pravidlo drop.
g) Pravidlo reject.
h) Pravidlo sdrop.

Jednotlivá pravidla se ukládají do složky rules. V případě, že pravidlo začíná řetězcem alert, tak je pouze prováděna notifikace o akci, kterou pravidlo definuje. V případě shody s pravidlem log jsou jednotlivé záznamy zaznamenávány. Další možností je pravidlo pass, které síťový provoz v případě shody odstraní. Pravidlo activate provede alert a aktivuje jiné pravidlo dynamic. Pravidlo dynamic není prováděno, dokud není aktivováno pravidlem activate a poté pracuje jako pravidlo log. Pravidlo drop provede zalogování a blokaci paketu. Pravidlo reject odmítne paket a poskytne odpověď na požadavek. Pravidlo sdrop provede pouze blokaci paketu bez provedení logu. Základní princip vytváření pravidel viz Obr. 4.2.



Obr. 4.2: Základní pravidlo Snortu.

Obecný postup nastavení systémů IDS a IPS je prohledávání obsahu síťového provozu a vyhledávání škodlivého obsahu na základě signatur. Také je nezbytné nastavit limity pro anomálie, ovšem zde nelze přesně říci, kde má být hranice. Každý systém je odlišný a je třeba udržovat rovnováhu mezi falešně pozitivně a falešně negativně hodnoceným síťovým provozem.
Snort poskytuje doporučenou konfiguraci ve formě předdefinovaných pravidel, kterou lze doplnit o svá další pravidla.

IPS Snort umožňuje spuštění nastavené konfigurace ve třech módech. Prvním módem je tzv. Sniffer mód (packet dump mód), který umožňuje zobrazovat hlavičky paketů, které procházely sítí. Parametr „i“ definuje, který síťový adaptér bude kontrolován a podroben pravidlům. Další parametr „v“ znamená, že bude poskytnuto větší množství informací.

Dalším dostupným módem, ve kterém může být Snort spuštěn je mód Packet Logging Mode. Tento mód umožňuje zápis jednotlivých paketů, které hierarchicky třídí v souboru (logu). Je ovšem nutné uvést cestu k složce, ve které budou jednotlivé logy uloženy.

Posledním módem, ve kterém lze Snort spustit je IDS Mode. Nejedná se ovšem pouze o funkci IDS, ale i o IPS funkci, která může být definována v pravidlech. Tyto pravidla jsou načtena pomocí konfiguračního souboru, ke kterému je nutné uvést cestu (pomocí parametru „c“). V příkazu následuje vydefinování logovacího souboru, parametr „h“ značí, že bude prováděno ukládání včetně TCP/IP hlaviček. Poslední parametr uvádí adresu sítě, podle které budou třízeny do podadresářů.

Pokud se budeme chtít zaměřit na vlastní použití v OS Ubuntu, tak je velice jednoduché.

Nejprve je třeba provést update a upgrade, to provedeme příkazy „`apt-get update`“ a „`apt-get upgrade`“. Dále jen zadáme příkaz k instalaci Snortu „`sudo apt-get install snort`“.

Následně se zobrazí dotaz na zadání názvu rozhraní, na kterém má být IDS/IPS Snort spuštěn (k identifikaci lze použít příkaz „`ifconfig`“). Zadáme tedy například „enp0s3“ (jedná se o port v případě virtualizace). Do dalšího kroku se dostaneme tak, že stiskneme Tab, označíme možnost „Ok“ a stiskneme klávesu Enter. V dalším kroku zadáme vnitřní síť ve tvaru „<IP_adresa_sítě>/<maska_sítě>, opět vybereme a následně zvolíme možnost „Ok“.

Verzi Snortu si lze zobrazit příkazem „`snort -v`“. Nyní se podívejme, na strukturu. Veškeré soubory nalezneme v adresáři „/etc/snort“. Jednotlivé soubory, které obsahují skutečná pravidla, jsou uložena v adresáři „/etc/snort/rules“. Dále je třeba zmínit konfigurační soubor, do kterého lze vstoupit příkazem „`sudo nano /etc/snort/snort.conf`“. Zde lze upravit, zda budou použity jednotlivé soubory s pravidly, je zde i odkaz na soubor, který je určen pro ukládání vlastních pravidel. Tento soubor se jmenuje „local.rules“, opět uložen v „/etc/snort/rules“. Pokud by byl tento řádek zakomentován, lze jej odkomentovat (smazáním znaku #).

Konfiguraci můžeme ověřit tím, že zadáme příkaz „`snort -T -i <název_rozhraní> -c /etc/snort/snort.conf`“. Samotné spuštění lze provést příkazem „`snort -A console -q -v -c /etc/snort/snort.conf -i <název_rozhraní>`“.

Pokud bychom se chtěli zaměřit na logy, které jsou generovány, tak jsou uloženy v souboru „/var/log/snort“. Zde jsou uloženy podle časových razítek. Jednotlivý soubor lze přečíst zadáním příkazu „`sudo snort -dv -r /var/log/snort/<název_logu>`“.

Doposud byl Snort nastaven na IDS, pokud bychom chtěli aplikovat IPS, tedy použít i reaktivní opatření, je k tomu zapotřebí lehce pozměnit konfigurační soubor. V konfiguračním souboru, kolem řádku 160 (závisí na verzi) je třeba upravit (odkomentovat a doplnit výrazy) direktivy, a to do stavu:

```
config policy_mode: inline
config daq: afpacket
config daq_mode: inline
```

Příkaz pro spuštění je poté ve tvaru „`snort -A console -Q -v -c /etc/snort/snort.conf -i <vstupní_rozhraní>:<výstupní_rozhraní> -N`“. Jak vidíme, abychom mohli spustit Snort v režimu IPS je třeba použít dvě rozhraní.

Použití je totiž běžně na samostatných strojích, které fungují jako tzv. „pračky provozu“, propustí pouze legitimní a žádaný síťový provoz. Z toho vyplývá, že pokud chceme použít pravidlo „drop“, tak je nutné použít režim IPS, jinak toto pravidlo nebude účinné.

4.2.7 Firewall

Operační systém Ubuntu, ostatně jako každý operační systém založen na Linuxu, disponuje nástrojem pro filtraci síťového provozu „iptables“. Iptables [41] jsou složeny ze čtyř tabulek. Z tabulky „filter“, „nat“, „raw“ a „mangle“.

Pro provedení základního filtrování je vhodná tabulka „filter“. Tato tabulka je složena z řetězců „INPUT“, „OUTPUT“, „FORWARD“. Výpis nastavených pravidel lze vypsat příkazem „`iptables -L`“, kde vedle řetězce je viditelná výchozí politika. Pokud chceme tuto politiku změnit, použijeme příkaz „`iptables -P <řetězec> <ACCEPT/DROP>`“.

V případě serveru je velice vhodné povolit spojení, která jsou ve stavu „related“ nebo ve stavu „established“. Proto použijeme příkaz „`iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT`“. Proto, aby byl příkaz vykonán, je nutné, aby mezi stavy „established“ a „related“ byla čárka, ale výrazy jsou bez mezery.

V postupu, že zakážeme veškerý přístup a poté budeme povolat jednotlivé možnosti je jeden menší problém. Je nutné si uvědomit, že pokud konfigurujeme server vzdáleně přes „ssh“ a změníme výchozí politiku na stav „DROP“ tak spadá do tohoto pravidla i naše spojení a následně je mnohdy nutná lokální oprava. Abychom předešli tomuto problému, tak je vhodné povolit přístup pomocí portu. Připojení přes SSH využívá port 22, proto použijeme příkaz „`iptables -A INPUT -p tcp --dport ssh -j ACCEPT`“. Takto budeme nadále povolovat další služby, které jsou pro náš server důležité. Po ukončení je vhodné vložit poslední zápis ve tvaru „`iptables -A INPUT -j DROP`“. Tento zápis způsobí, že veškerá další spojení, která neodpovídají předchozím povolením, jsou zakázána, ovšem za předpokladu, že výchozí politika řetězce je „ACCEPT“, v případě, že je nastavena na „DROP“, tak je tento zápis redundantní.

Jistou nevýhodou je to, že při případném restartu dochází k odebrání všech pravidel. Abychom tomuto předešli, tak je nutné provedení zálohy. Pokud preferujeme ruční provedení, tak je možné využít příkaz „`iptables-save > /etc/iptables.rules`“ pro uložení definovaných pravidel, které je nutné po restartu obnovit příkazem „`iptables-save > /etc/iptables.rules`“. Automatické načtení pravidel lze provést více způsoby.

Nejjednodušším způsobem je úprava konfiguračního souboru „/etc/network/interfaces“. Je ovšem nutná znalost, přes které síťové rozhraní probíhá komunikace. Samotná úprava je možná přes editor textu, například pomocí příkazu „`nano /etc/network/interfaces`“. Zde nalezneme příslušný interface a na konec záznamu přidáme příkaz „`pre-up iptables-restore < /etc/iptables.rules`“, který před startem načte uložený soubor.

Ovšem OS Ubuntu disponuje nástrojem „UFW“ (Uncomplicated Firewall) [42], který slouží k nastavení firewallu. Tento nástroj je součástí základní distribuce, ovšem není aktivní. Abychom tento nástroj aktivovali, použijeme příkaz „`sudo ufw enable`“.

K definování provozu lze využít politiku „allow“, tedy povoleno, „deny“, tedy zakázat a neodpovídat na požadavky. Dále „reject“, který je totožný, jako „deny“, ovšem odpovídá zdroji, že cíl spojení odmítá. Poslední možností je „limit“, který definuje počet, po který je provoz akceptován a při jeho překročení, je provoz zakázán. Výchozí politika je nastavena tak, že veškerý příchozí provoz je zakázán a odchozí povolen. Lze zobrazit příkazem „`sudo ufw status verbose`“. K výpisu pravidel je vhodné použít příkaz „`sudo ufw status`“, který pravidla zobrazí, pokud nejsou, zobrazí pouze stav firewallu. UFW také umožňuje logování, to lze zapnout, či vypnout příkazem „`sudo ufw logging <on, off>`“.

Obecný příkaz pro nastavení výchozí politiky má tvar „`sudo ufw [--dry-run] default <allow, deny, reject> <incoming, outgoing>`“. Parametr „dry-run“ značí zkušební provoz.

Tento nástroj umožňuj práci jak s porty, tak se jmény služeb. V případě práce s porty může příkaz vypadat následovně „`sudo ufw <allow, deny> <číslo_portu>/<protokol>`“. V případě práce se službami, může příkaz vypadat „`sudo ufw <allow, deny> <služba>`“. Pokud potřebujeme konkrétní pravidlo lze definovat zdroj a cílový port. Příkaz poté může vypadat „`sudo ufw <allow, deny> <IP_adresa> to any port <číslo_portu>`“.

Pokud je třeba pravidlo smazat, použijeme příkaz „`sudo ufw delete <pravidlo>`“. Jednotlivá pravidla lze také upravovat přímo v souborech umístěných v „/etc/ufw“.

4.2.8 Zálohování

Zálohování dat je velice důležitá věc a není radno ji podceňovat. V případě, že se cokoli pokazí a nejsme plně schopni uvést systém do plného provozu, tak je možné nahrát zálohu, a tak opět poskytovat svým klientům služby, které garantujeme. Pokud chceme vytvořit zálohu celého systému, můžeme vyžít nástroj „tar“ [43], který je dostupný v každém Linuxu. Pokud bychom chtěli pravidelně zálohovat uživatelská data, tak lze využít odlišných nástrojů, jako například nástroje SBackup, nebo nástroje „Déjà Dup“.

Záloha prováděna nástrojem „tar“ využívá toho, že superuživatel má přístup ke všem souborům a není nijak omezen. Vše provedeme příkazem „`tar czf /backup.tgz --one-file-system --ignore-failed-read --sparse --exclude=/backup.tgz /`“. Když se podíváme blíže, co se provádí, tak „tar“ je název příkazu, volby „czf“ udávají, že se má provést archivace, použít komprese a výsledné umístění bude soubor. Řetězec „/backup.tgz“ udává výsledný název a umístění (/ značí kořenový adresář). Dále volba „one-file-system“ značí, že se provede pouze záloha operačního systému, a ne přídavných OS, jako je například připojené USB. Poté „ignore-failed-read“ zajistí, že příkaz nebude ukončen v případě, že nebude možné přečíst veškeré soubory. Možnost „sparse“ zajistí, aby se provedla záloha i sparse souborů. Abychom do zálohy nezahrnovali námi vytvářenou zálohu, vložíme volbu „exclude=/backup.tgz“. Posledním je znak „/“ jež udává cíl zálohy, tedy co se bude zálohovat, pokud použijeme tento znak, tak se provede záloha celého systému.

Obnova je lehce komplikovanější tím, že musíme být opatrní, abychom vše provedli korektně. Nejprve je nutné nabootovat jiný operační systém, kde si získáme práva superuživatele. Zde si vytvoříme adresář pomocí příkazu „`mkdir /mnt/ubuntu`“ a připojíme oddíl, na němž je umístěn OS „`mount cesta_k_oddilu /mnt/ubuntu`“ (cesta k oddílu může vypadat například takto „/dev/sda2“. Nyní se přepneme do námi vytvořeného adresáře za pomocí příkazu „`cd /mnt/ubuntu`“. V posledním kroku spustíme obnovení systému ze zálohy příkazem „`tar xzf backup.tgz`“. Je nutné si uvědomit, že tento příkaz je možný pouze v případě, že námi spouštěná záloha je umístěna v kořenovém adresáři, jinak bychom museli uvést cestu k tomuto souboru.

4.2.9 Logování

Operační systém Linux vytváří při změně stavů tzv. logy. Logy [44] jsou tvořeny různými komponenty systému, komponenty jádra, jednotlivými démony a dalšími prvky. Logy jsou typicky ukládány do „/var/log“, kde jsou jednotlivé logy uloženy ve formě souborů. Soubory mohu být také uloženy v podadresáři. Mezi nejznámější logy patří například syslog, kern.log, faillog a další. Jednotlivé logy uchovávají důležité informace a je tedy vhodné tyto logy zálohovat, či přenášet na jiný počítač, za účelem uchování dat, která mohou být získána při prolomení zabezpečení útočníkem, který tyto logy může smazat a zahladit tak po sobě stopy.

Ruční procházení logů může být příliš náročné, a proto je lepší využít nástroje jako „logcheck“, nebo „logwatch“. Logcheck posílá e-mail správci systému o jednotlivých lozích, které prošli přes definovaná pravidla. Logwatch slouží spíše k sumarizaci, posílá tedy souhrny a grafy než jednotlivé logy.

Logcheck ke své činnosti využívá nástroje „cron“, ten automatizovaně, jednou za definovaný úsek odešle e-mail s hláškami z logů. Interval, ve kterém jsou jednotlivé e-maily zasílány lze upravit v „/etc/cron.d/logcheck“ (defaultně jednou za hodinu). Tyto e-maily jsou zasílány na „root@localhost“, v případě, že chceme pozměnit příjemce, lze tak provést v konfiguračním souboru „/etc/logcheck/logcheck.conf“ kde pozměníme položku „sendmailto“.

4.3 Webový server

Pojem webový server [45] má dva významy. Prvním z nich je webový server jako program, například webový server Apache. Tento program poskytuje jednotlivé webové stránky směrem ke svým klientům a pro samotný přenos využívá protokol HTTP, v případě zabezpečené varianty protokol HTTPS. Druhý význam označuje samotný stroj, který poskytuje webové služby svým klientům, ovšem i na tomto stroji je instalován webový server ve významu programu.

4.3.1 Instalace

Na server budeme instalovat vybraný webový server Apache [46]. Jedná se o velmi rozšířený webový server a jeho použití je velmi jednoduché. Instalaci provedeme tak, že do terminálu zadáme příkaz „`sudo apt-get install apache2`“. Po instalaci je tento

webový server ihned aktivní, lze si to ověřit pomocí příkazu „`systemctl status apache2`“.

V případě, že budeme chtít zacházet s Apache serverem, můžeme využít těchto příkazů „`systemctl <parametr> apache2`“. Tento parametr může být například „stop“, „start“, „restart“, „reload“ (provede načtení provedených změn v konfiguračních souborech, aniž by ukončil sezení, které má navázané s klienty), „enable“, „disable“ (defaultní hodnota je „enable“, tzn. že webový server je spuštěn po každém načtení systému serveru, pokud tomuto chceme předejít, změníme hodnotu na „disable“ a bude nutné manuální spuštění serveru).

Pokud budeme chtít měnit obsah webových stránek, jež se zobrazují klientům, tak je nutné změny provést v adresáři „/var/www/html“. Pozn. první webová stránka, která se klientům zobrazí, je s názvem index. Veškeré konfigurační soubory jsou umístěny v adresáři „/etc/apache2“. Základní konfigurační soubor nese název „apache2.conf“. Jak již bylo zmiňováno, systém zaznamenává logy, stejně tak jako Apache. Každý požadavek na zobrazení webové stránky lze nalézt v umístění „/var/log/apache2/access.log“ a veškeré chybové hlášky v souboru „/var/log/apache2/error.log“. V defaultním nastavení jsou zaznamenávány veškeré chybové hlášky, pokud bychom nechtěli zaznamenávat veškeré hlášky, lze provést změnu v parametru „LogLevel“ v konfiguračním souboru.

4.3.2 Zabezpečení

Zabezpečení webového serveru je velice důležitá věc [47], [48]. První věcí, na kterou bychom měli myslet je pravidelné aktualizování. V Dalším kroku vstoupíme do konfiguračního souboru umístěného v „/etc/apache2/conf-available/security.conf“, nesmíme zapomenout, že je nutné mít oprávnění administrátora. Zde upravíme pár hodnot.

První věcí, na kterou se zaměříme je „zamaskování“, nebo lépe řečeno nezobrazení použité verze webového serveru. To je důležité, protože v případě, že útočník chce napadnout náš webový server, tak se mu právě takové informace hodí k tomu, aby lépe cílil útok. Krizovým místem je webová stránka, která zobrazuje, že stránka není dostupná, právě zde na sebe webový server Apache v defaultním nastavení prozradí nejen použitý webový server, jeho verzi, ale dokonce i použitý operační systém. Tomu zabráníme změnou hodnoty direktivy „`ServerSignature`“ na „`Off`“. K tomuto postačuje, pouze odkomentování a zakomentování řádku.

V dalším kroku je vhodné pozměnění HTTP response hlaviček. V konfiguračním souboru nalezneme direktivu „`ServerTokens`“ a nastavíme hodnotu „`Prod`“, tím se omezíme pouze na zobrazení názvu webového serveru, bez zobrazení verze, nebo operačního systému.

Nástrojem, jak zobrazit hlavičku, která je zaslána je nástroj „curl“. Do terminálu stačí zadat „`curl -I <IP_adresa_nebo_doména>`“.

Dále, abychom omezili možný pohyb uživatele [49], je nutné pozměnit nastavení v hlavním konfiguračním souboru, tedy v „/etc/apache2/apache2.conf“. Pozměníme tedy direktivu `<Directory /var/www/>` a hodnotu „`Options Indexes FollowSymLinks`“

pozměníme na „`Options FollowSymLinks`“. Ovšem stále je možné si po zadání přímé cesty k logu apache serveru toto logo zobrazit (na logu je uvedena i jeho verze) přes cestu „<doména>/icons/apache _bp2.png“. Abychom i tomuto předešli, tak musíme upravit „`<Directory />`“. Uvedeme parametry `Require all denied; Options None; AllowOverride None`. Pozn. každý parametr bude na novém řádku bez znaku středníku.

Také musíme vytvořit alias v konfiguračním souboru pomocí příkazu „`Alias /icons/ "usr/share/apache2/icons/"`“. Dále vytvoříme vlastní direktivu k adresáři „icons“, kde zakážeme přístup. Do konfiguračního souboru připíšeme:

```
<Directory "/usr/share/apache2/icons">
       Require all denied
</Directory>
```

Poté nesmíme zapomenout server restartovat příkazem „`sudo systemctl restart apache2`“.

4.3.3 SSL certifikát

Doposud bylo zabezpečení zaměřeno na webový server jako takový, nyní se zaměřme na zabezpečení spojení [50]. Aby bylo spojení šifrováno, využijeme SSL certifikát (viz kap. 2.8). V běžném použití obdržíme certifikát podepsaný certifikační autoritou. V této práci se zaměříme na „self-signed certificate“ to lze přeložit jako certifikát, který je podepsán samotným serverem. Neproběhlo tedy podepsání certifikační autoritou a certifikát se tak klientům jeví jako neověřený. Ovšem zabezpečení je totožné.

Jako první věc je nutné vygenerovat certifikační pár, tedy soukromý a veřejný klíč. Tento veřejný klíč je podepsaný vydavatelem, tedy samotným strojem, na kterém se tento příkaz zadává. Provedeme tedy příkaz „`sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt`“. Nyní se podívejme blíže na části příkazu. Samotný příkaz „openssl“ je nástroj pro vytváření a správu certifikátů, klíčů apod. Podpříkaz „req“ definuje, že použijeme nástroj pro žádost o podpis certifikátu (nový certifikát bude X.509), následující parametr „x509“ značí, že žádáme o vygenerování certifikátu, který je podepsán sám sebou (self-signed certificate). Následuje parametr „nodes“, ten přeskočí možnost zabezpečit certifikát heslem z důvodu, aby webový server měl k tomuto souboru přístup. Parametr „days“ definuje dobu, po kterou bude vydaný certifikát platný. Parametr „newkey rsa:2048“ znamená, že žádáme o vygenerování nového klíče a certifikátu, klíč bude mít délku 2048 bitů. Parametr „keyout“ značí cestu, kde bude uložen vygenerovaný soukromý klíč a parametr „out“ definuje místo, kde bude uložen certifikát. Následně budeme vyzvání k vyplnění údajů jako „Country name“, „State or Province Name“, „Locality Name“, „Organization Unit Name“, „Common Name“ (velmi důležitý parametr, zde se uvádí IP adresa serveru, popřípadě doménové jméno a posledním parametrem, na který jsme dotázání je „Email Address“. Jednotlivé certifikáty jsou uloženy v „/etc/ssl“

Nyní je třeba aplikovat protokol Diffie-Hellman, vygenerujeme tedy Diffie-Hellman group pomocí příkazu: „`sudo openssl dhparam -out`

`/etc/ssl/certs/dhparam.pem 2048`“. Výsledek bude následně uložen v „/etc/ssl/certs/dhparam.pem“.

Dalším krokem je nakonfigurovat Apache server k aplikování SSL. Pomocí příkazu: „`sudo nano /etc/apache2/conf-available/ssl-params.conf`“ vytvoříme konfigurační soubor pro Apache server, kde definujeme jednotlivá nastavení SSL. Do tohoto souboru zadáme:

```
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3
SSLHonorCipherOrder On
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
SSLCompression off
SSLSessionTickets Off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"
```

Dále upravíme defaultní nastavení SSL v konfiguračním souboru „/etc/apache2/sites-available/default-ssl.conf“. Nejprve si však provedeme zálohu, a to pomocí příkazu: „`sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/default-ssl.conf.bak`“. Nyní vstoupíme do konfiguračního souboru pomocí příkazu „`sudo nano /etc/apache2/sites-available/default-ssl.conf`“. V tomto souboru provedeme několik změn. Do direktivy „`ServerAdmin`“ zadáme e-mailový kontakt, vložíme novou direktivu „`ServerName`“ jako hodnotu uvedeme IP adresu webového serveru. Dále pozměníme direktivu „`SSLCertificateFile`“ a nastavíme hodnotu na „/etc/ssl/certs/apache-selfsigned.crt“, dále v direktivě „`SSLCertificateKeyFile`“ změníme hodnotu na „/etc/ssl/private/apache-selfsigned.key“. Posledním krokem bude odkomentování direktivy „`BrowserMatch`“. Její hodnoty jsou:

```
BrowserMatch "MSIE [2-6]" \
       nokeepalive ssl-unclean-shutdown \
       downgrade-1.0 force-response-1.0
```

Také je vhodné provést přesměrování z http na https. Vstoupíme do konfiguračního souboru pomocí příkazu „`sudo nano /etc/apache2/sites-available/000-default.conf`“ a vložíme direktivu „`Redirect "/" "<doména_nebo_IP_adresa>"`“.

Nyní je třeba povolit SSLa hlavičkové moduly. To provedeme příkazem „`sudo a2enmod ssl`“ a příkazem „`sudo a2enmod headers`“. Dále povolíme SSL Virtual Host pomocí příkazu „`sudo a2ensite default-ssl`“. Aby mohl „ssl-params.conf“ používat hodnoty, které jsme nastavili, je třeba to explicitně povolit a to příkazem „`sudo a2enconf ssl-params`“.

Je nutné ověřit konfiguraci, to provedeme příkazem „`sudo apache2ctl configtest`“. Pokud bude výstup obsahovat hlášku „Syntax OK“ je vše v pořádku,

i když bude v prvním řádku hlásit, že je nutno nastavit „ServerName“. Provedeme tedy restart serveru „`sudo systemctl restart apache2`“.

V tomto okamžiku by mělo být vše funkční. Ovšem pokud používáme starší konfiguraci je nutné pár věcí upravit. Vstoupíme do konfiguračního souboru Apache serveru příkazem „`sudo nano /etc/apache2/apache2.conf`“. Upravíme direktivu `<Directory \>` a místo hodnot „`Order deny,allow`“ a „`Deny from all`“ zadáme jedinou hodnotu „`Require all denied`“. Nyní by mělo být při správném postupu vše funkční. Nevýhodou však je, že náš server je opět zranitelný na zobrazení obrázku apache serveru pomocí zadání přímého odkazu. Abychom tomuto zamezili, vytvoříme si alias v konfiguračním souboru pomocí příkazu „`Alias /icons/ "usr/share/apache2/icons/"`“. Dále vytvoříme vlastní direktivu k adresáři icons, kde zakážeme přístup. Do konfiguračního souboru připíšeme:

```
<Directory "/usr/share/apache2/icons">
       Require all denied
</Directory>
```

Po provedení restartu služby příkazem „`sudo systemctl restart apache2`“ je webový server nakonfigurován tak, aby využíval SSL a adresář icons je nedostupný i při zadání přímého odkazu na obrázek apache.
Na uživatelské straně je nyní nutné certifikát zařadit mezi důvěryhodné a navštívit webovou stránku.

5 APLIKACE SCÉNÁŘE ZABEZPEČENÍ

Scénáře, které zde budou představeny, budou prováděny ve virtuálním prostředí. K virtualizaci bude použit nástroj „Oracle VMVirtualbox“. Jednotlivé stroje budou propojeny pomocí vnitřní sítě. Veškeré operace tak neopustí virtuální prostředí.

5.1 Scénář zabezpečení osobního počítače

Nejprve se zaměříme na bezpečnostní scénář osobního počítače. Tento scénář je zaměřen zejména na běžné uživatele, kteří preferují funkčnost, jednoduchost a bezpečnost na standartní úrovni. Scénář bude proveden pro operační systém Windows 10.

5.1.1 Bezpečné heslo

Jedním ze základních prvků bezpečného OS je bezpečné heslo. To se definuje při samotné instalaci OS, vytváření nového uživatele, popřípadě lze provést změnu přímo v systému.

Změna hesla jednotlivého účtu je velmi jednoduchá. V nabídce Start zadáme „Změnit heslo“ a možnost vybereme, viz Obr. 5.1.

Obr. 5.1: Změna hesla, OS Windows.

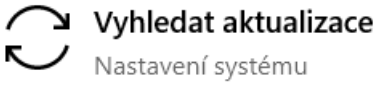
Poté jsme přesměrování do nastavení, kde můžeme provést změnu hesla, PIN kódu, popřípadě obrázkového hesla, viz Obr. 5.2.

Obr. 5.2: Změna přihlašování, OS Windows.

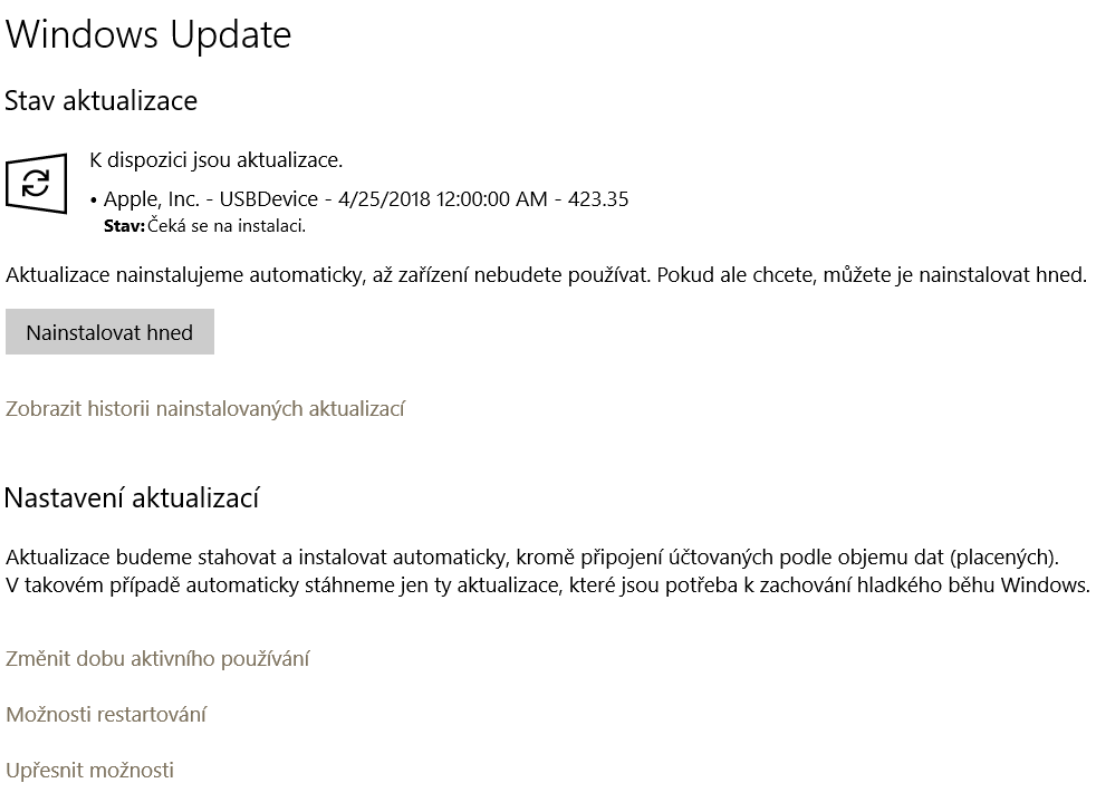
Při výběru hesla musíme nejprve zadat stávající heslo a poté heslo změnit. Nesmíme opomenout, že takové heslo musí být bezpečné, ve scénáři použijeme heslo „Červ*v180#b45!“. Další možností, jak provést přihlášení je pomocí PIN kódu. Tato možnost není z bezpečnostního hlediska doporučena (snižuje možnost znaků na čtyři číslice). Také se lze přihlásit pomocí „Obrázkového hesla“. Po vložení obrázku lze definovat jednotlivé akce, které budou provedeny myší (utvoření kruhu, klik, tah kurzoru apod.). V tomto scénáři ovšem použijeme pouze heslo.

5.1.2 Aktualizace

Aktualizace se v OS Windows provádějí zcela automatizovaně, jejich nastavení lze provést v nastavení počítače. Lze jej také vyvolat tak, že do nabídky start zadáme text „Vyhledat aktualizace“, viz Obr. 5.3. Následně se zobrazí nabídka k vyhledání aktualizací, viz Obr. 5.4.



Obr. 5.3: Vyhledat aktualizace, OS Windows.



Obr. 5.4: Aktualizace, OS Windows.

Pokud jsou aktualizace k dispozici, tak lze zvolit okamžitou instalaci. Pokud tomu tak není, lze manuálně provést vyhledání aktualizací. V defaultním stavu se aktualizace

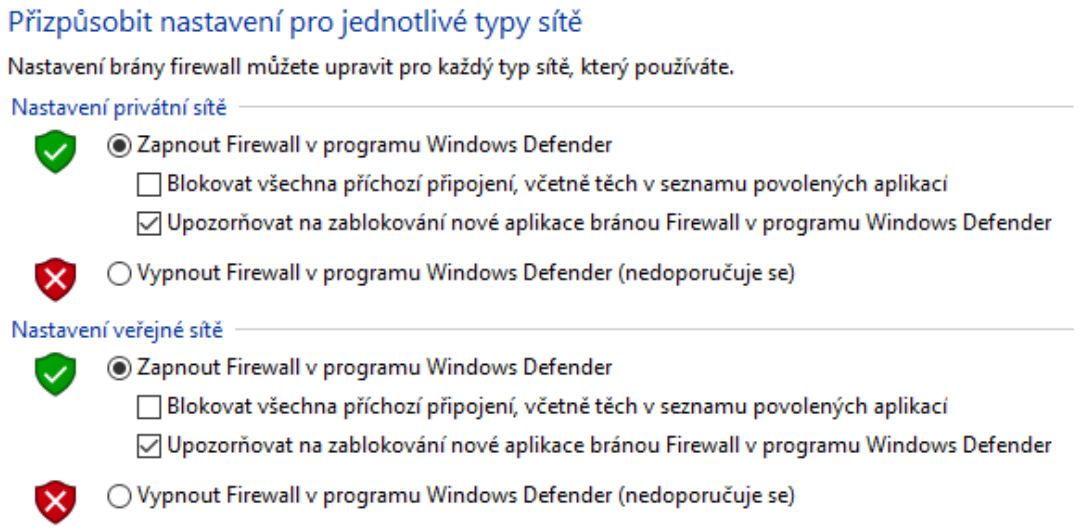
provádějí v době, kdy počítač není „aktivně používán“. Dobu „aktivního používání“ lze pozměnit v nastavení aktualizací.

5.1.3 Firewall

Jak již bylo řečeno v bezpečnostním scénáři osobního počítače, existují dva modely přístupu k Firewallu. V tomto scénáři využijeme v podstatě oba modely. Aby byl tento scénář blíže „běžnému uživateli“, tak použijeme výchozí politiku Windows Firewall. Veškerý příchozí provoz bude blokován a povolen bude pouze ten žádoucí. Ovšem veškerý odchozí provoz bude povolen a případný nežádoucí provoz je třeba blokovat. Z důvodu, že Windows Firewall je vhodně nastaven není třeba přidávat, popřípadě modifikovat existující pravidla, je to ovšem velmi jednoduché.

Pro spuštění do nabídky Start zadáme text „Firewall v programu Windows Defender“. Spuštění lze provést i přes ovládací panely, kde zvolíme „Systém a zabezpečení“, dále „Firewall v programu Windows Defender“.

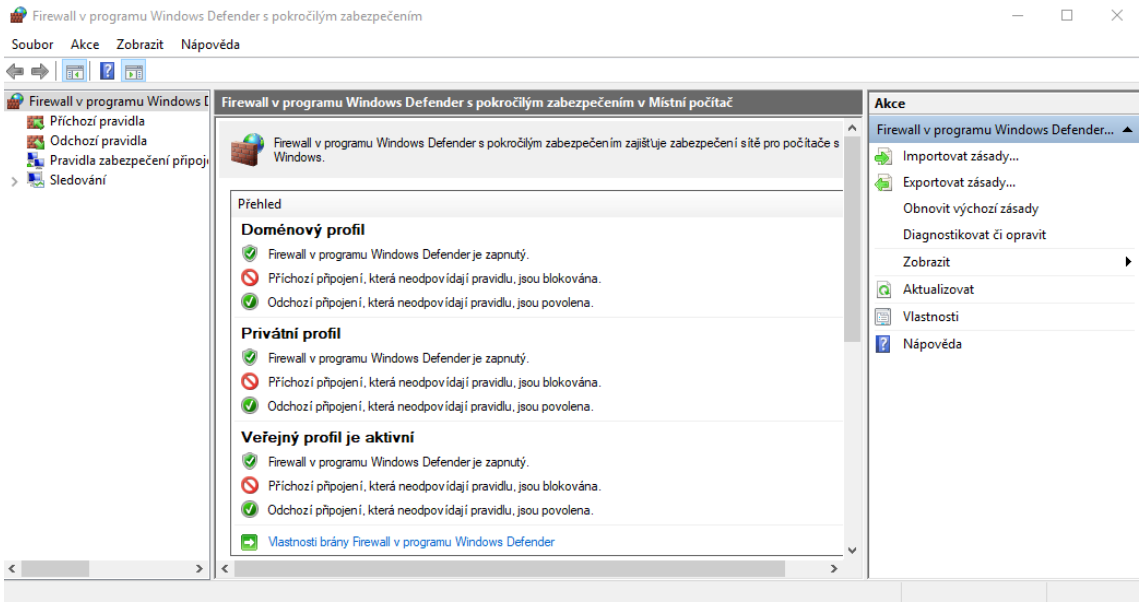
Zapnutí brány, popřípadě vypnutí lze provést v otevřeném oknu, když klikneme na možnost „Zapnout nebo vypnout Firewall v programu Windows Defender“ v levém panelu. Zde vidíme možnost, že v případě že nainstalujeme novou aplikaci a bude možné ji „kontaktovat“ z vnější strany budeme na to upozorněni a lze poté automaticky vydefinovat nové pravidlo pro její povolení, viz Obr. 5.5.



Obr. 5.5: Brána Firewall, OS Windows.

Pokud bychom chtěli vytvořit nové pravidlo, vybereme v levém panelu možnost „upřesnit nastavení“. Otevřené okno je rozděleno na tři části. Ve středu lze vidět jednotlivé profily a aktivní profil. Výchozí politiku by bylo možné změnit po kliknutí na „Vlastnosti Brány Firewall v programu Windows Defender“, což v tomto scénáři provádět nebudeme.

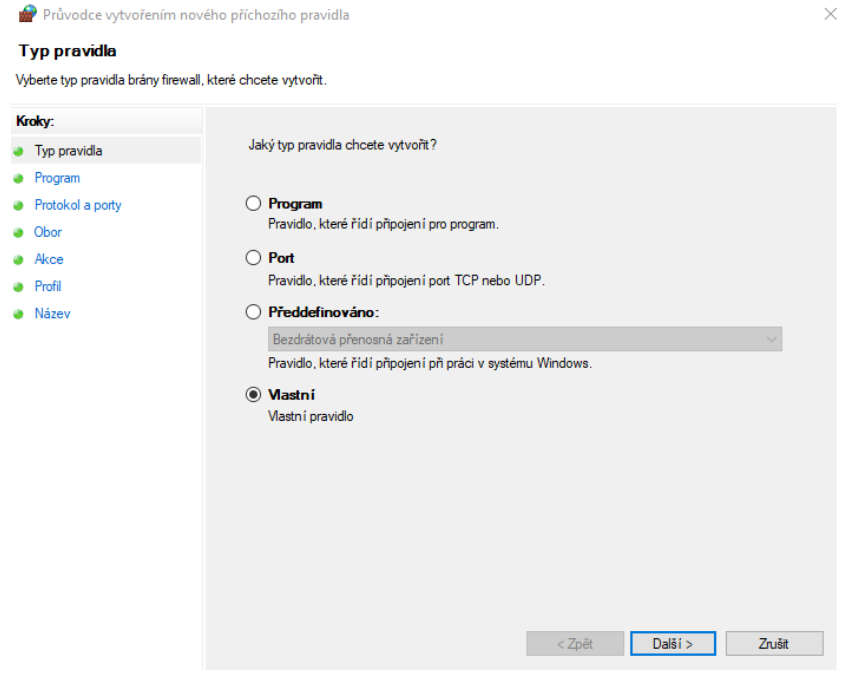
V levém panelu se lze pohybovat v příchozích a odchozích pravidlech. V pravém panelu lze vybrat jednotlivé akce, viz Obr. 5.6.



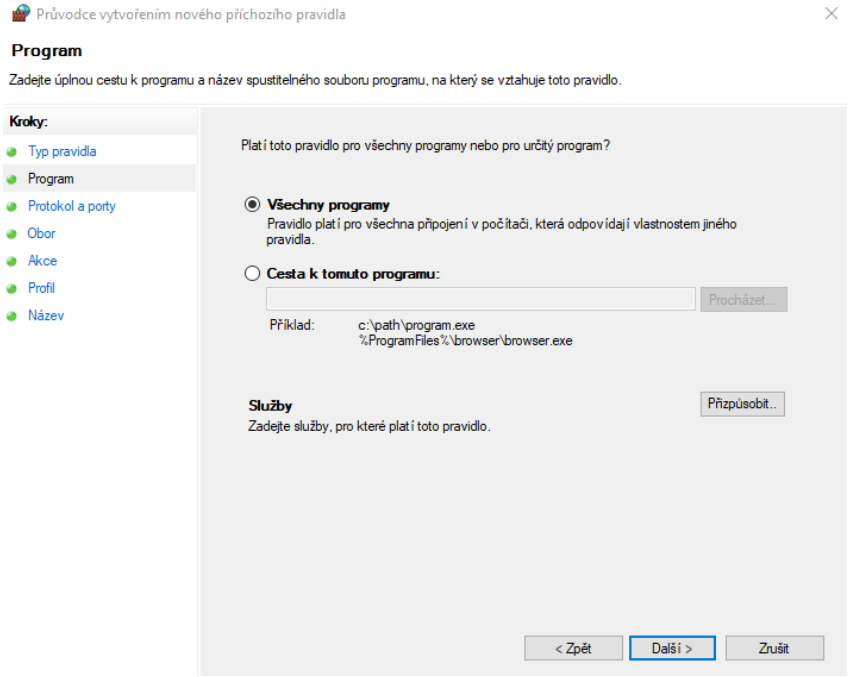
Obr. 5.6: Firewall v programu Windows Defender s pokročilým zabezpečením.

Nyní si uvedeme demonstrativní případ pro přidání nového pravidla pro odeslání ICMP reply, aby bylo možné počítač identifikovat v síti pomocí příkazu Ping. Jedná se o příchozí pravidlo, vybereme tedy v levém panelu možnost „příchozí pravidla“. V zobrazeném seznamu lze toto pravidlo také nalézt pod názvem „Sdílení souborů a tiskáren (požadavek na odezvu – ICMPv4-In)“, ovšem není aktivní, to lze provést kliknutím pravého tlačítka na pravidlo a zvolit možnost „povolit pravidlo“, je třeba si dávat pozor na jednotlivé profily, abychom aktivovali to správné pravidlo v profilu, který je momentálně aktivní. Pokud bychom si takové pravidlo chtěli vytvořit sami, vybereme možnost „Nové pravidlo…“, kterou lze nalézt v pravém panelu.

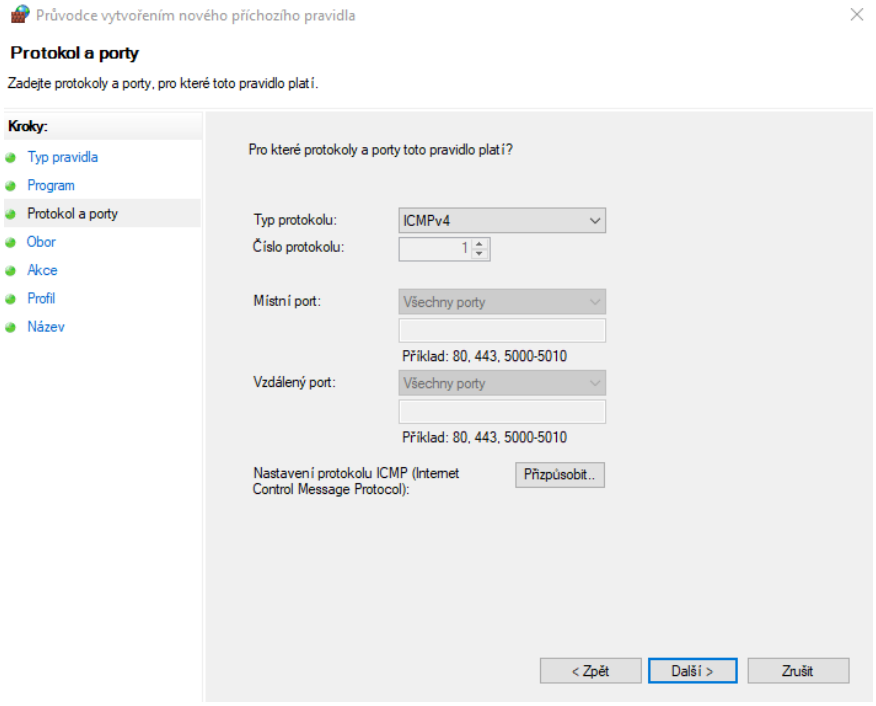
V otevřeném oknu vybereme možnost „Vlastní“, viz Obr. 5.7. V dalším kroku vybereme možnost „Všechny programy“, viz Obr. 5.8. V dalším kroku vybereme typ protokolu, vybereme „ICMPv4“ (jako další pravidlo je vhodné vytvořit pravidlo pro ICMPv6), viz Obr. 5.9. Dále lze upravit IP adresu lokální/vzdálené pro které toto pravidlo platí v případě ICMP lze ponechat možnost „libovolná IP adresa“, viz Obr. 5.10. V dalším kroku lze definované spojení povolit, povolit pouze zabezpečené, popřípadě blokovat. Zvolíme možnost „Povolit připojení“, viz Obr. 5.11. Předposledním krokem je vybrat profil, pro který bude toto pravidlo platit, viz Obr. 5.12. V posledním kroku toto pravidlo pojmenujeme, např. „ICMPv4“, viz Obr. 5.13.



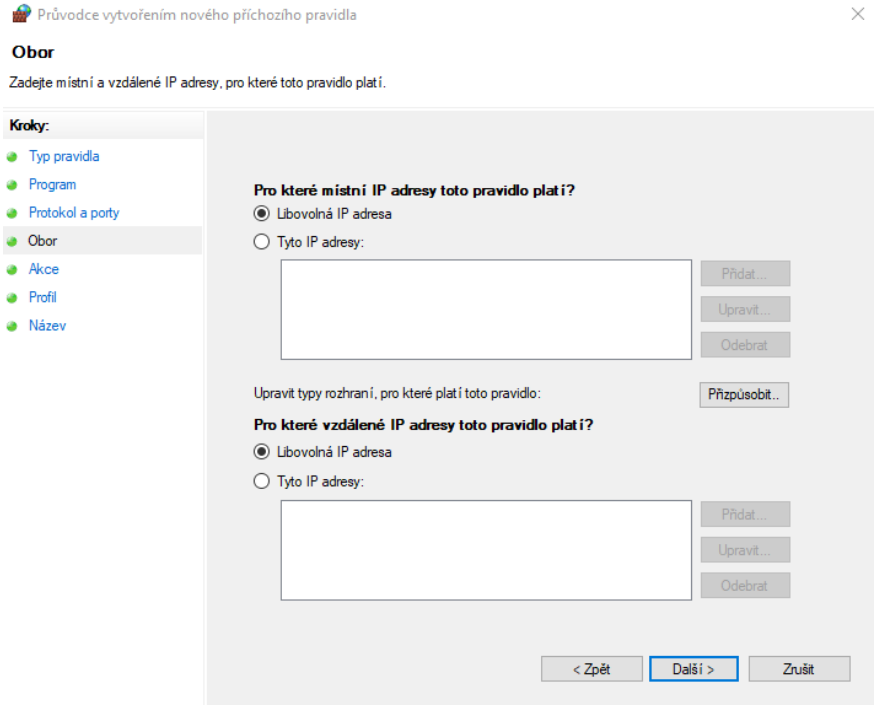
Obr. 5.7: Firewall, tvorba pravidla I.



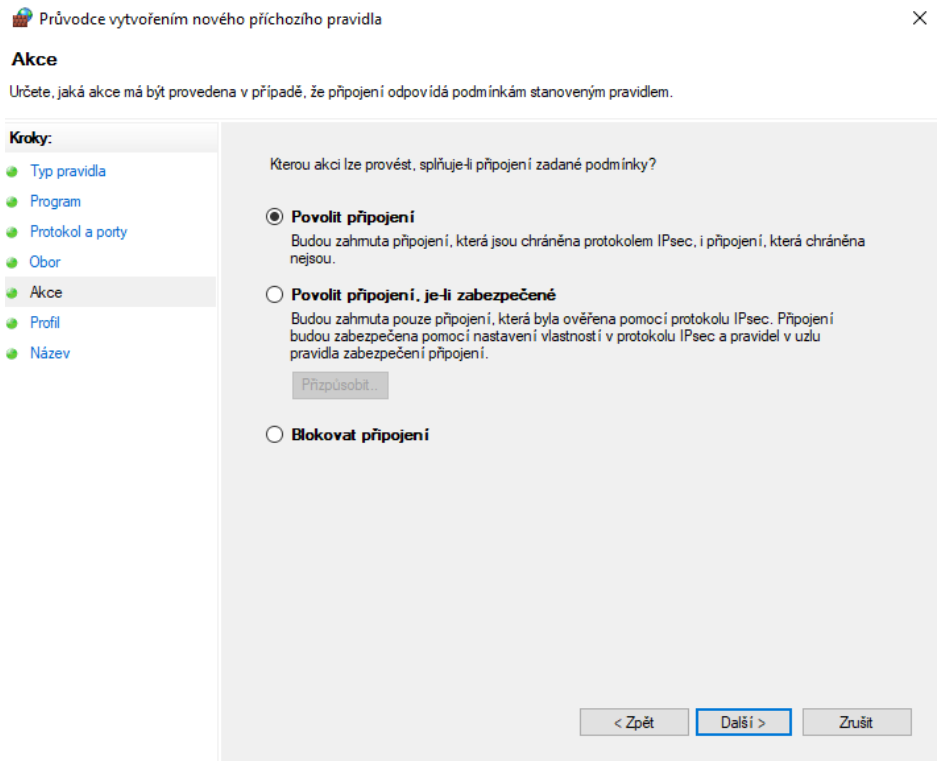
Obr. 5.8: Firewall, tvorba pravidla II.



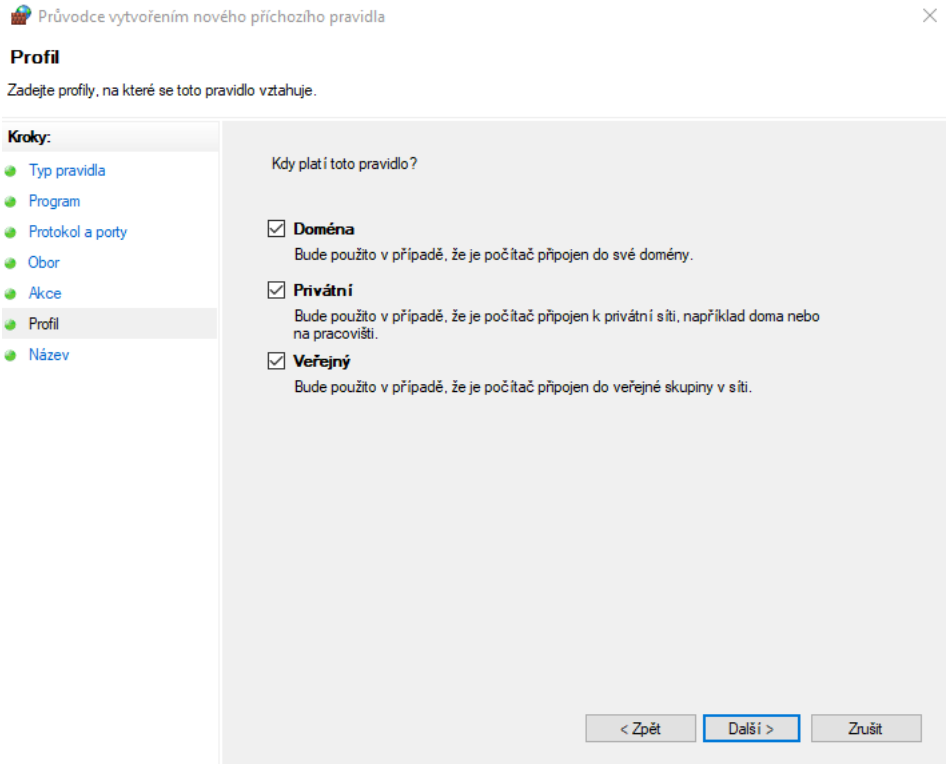
Obr. 5.9: Firewall, tvorba pravidla III.



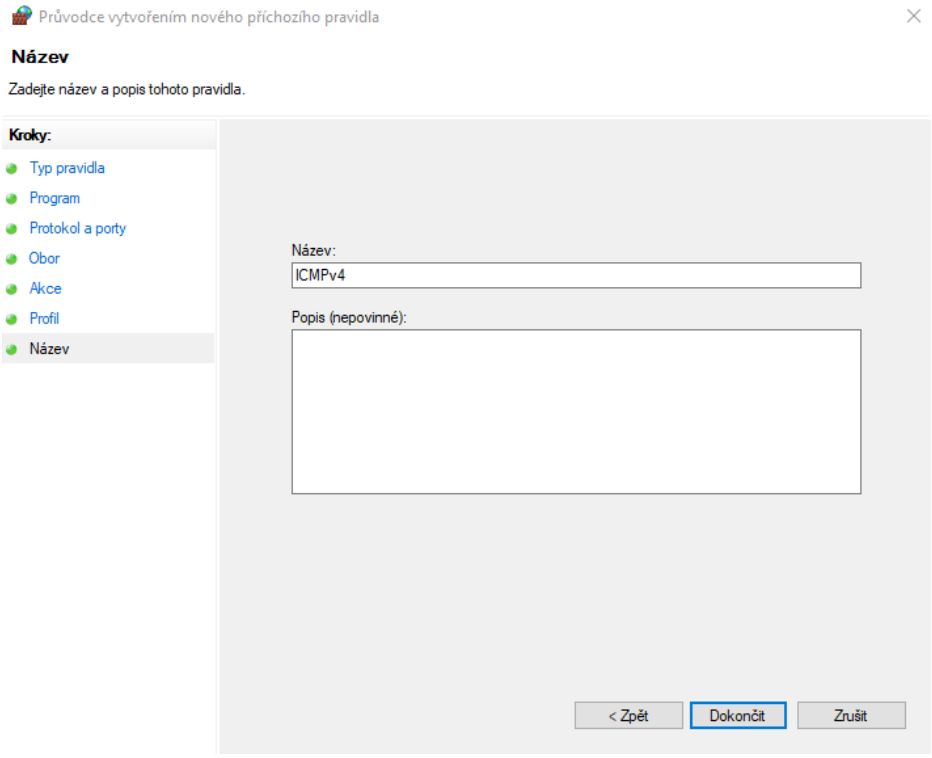
Obr. 5.10: Firewall, tvorba pravidla IV.



Obr. 5.11: Firewall, tvorba pravidla V.



Obr. 5.12: Firewall, tvorba pravidla VI.

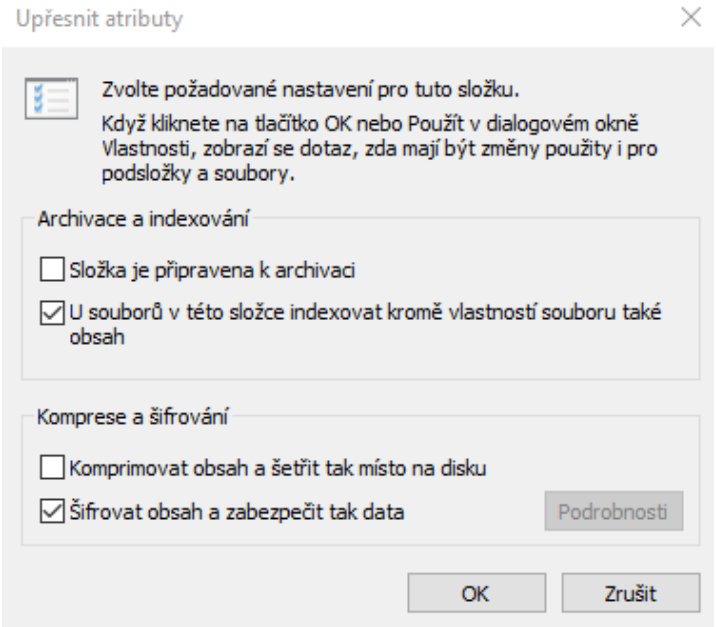


Obr. 5.13: Firewall, tvorba pravidla VII.

5.1.4 Šifrování

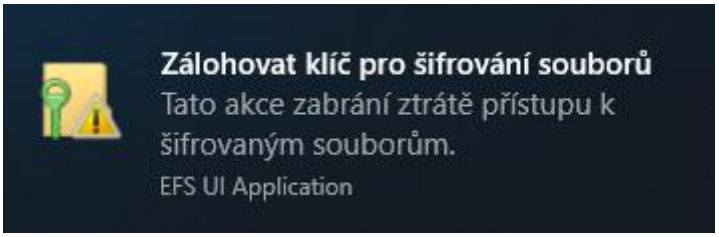
V OS Windows lze šifrovat jak jednotlivé složky, tak celé oddíly.

Šifrování složek se skládá z několika kroků. Nejprve je třeba vybrat složku, která má být šifrována. Klikneme na ni pravým tlačítkem myši a vybereme možnost „Vlastnosti“. Zde v kartě „Obecné“ zvolíme možnost „Upřesnit“. V otevřeném okně vybereme množnost „Šifrovat obsah a zabezpečit tak data“, viz Obr. 5.14.



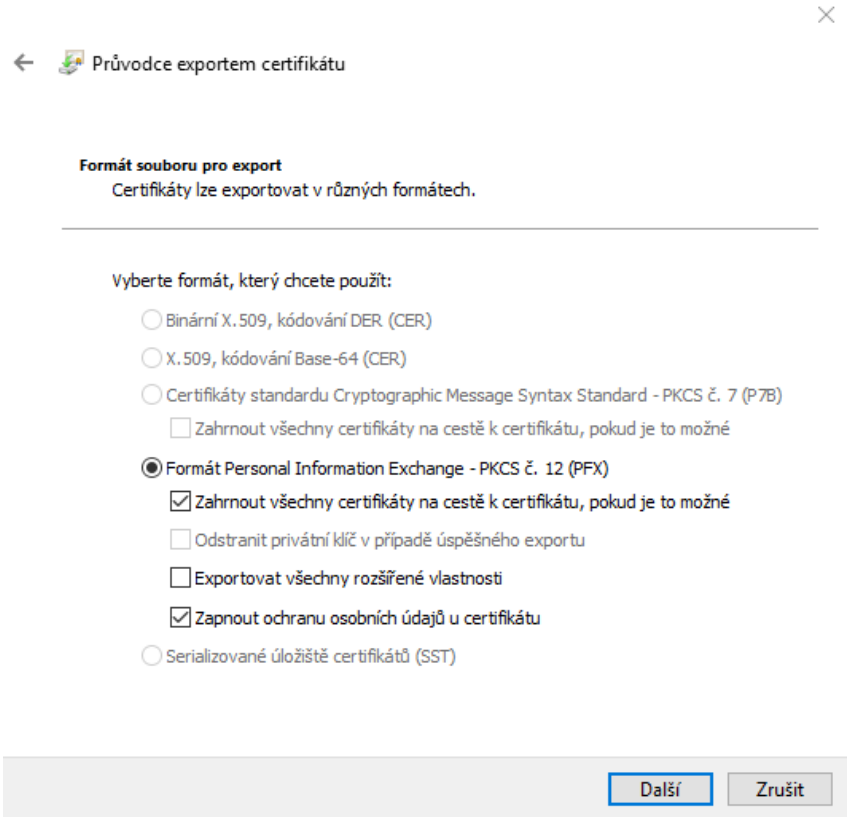
Obr. 5.14: Upřesnění atributů složky, OS Windows.

Poté nás OS Windows automaticky vyzve k záloze klíče, viz Obr 5.15.



Obr. 5.15: Záloha soukromého klíče uživatele.

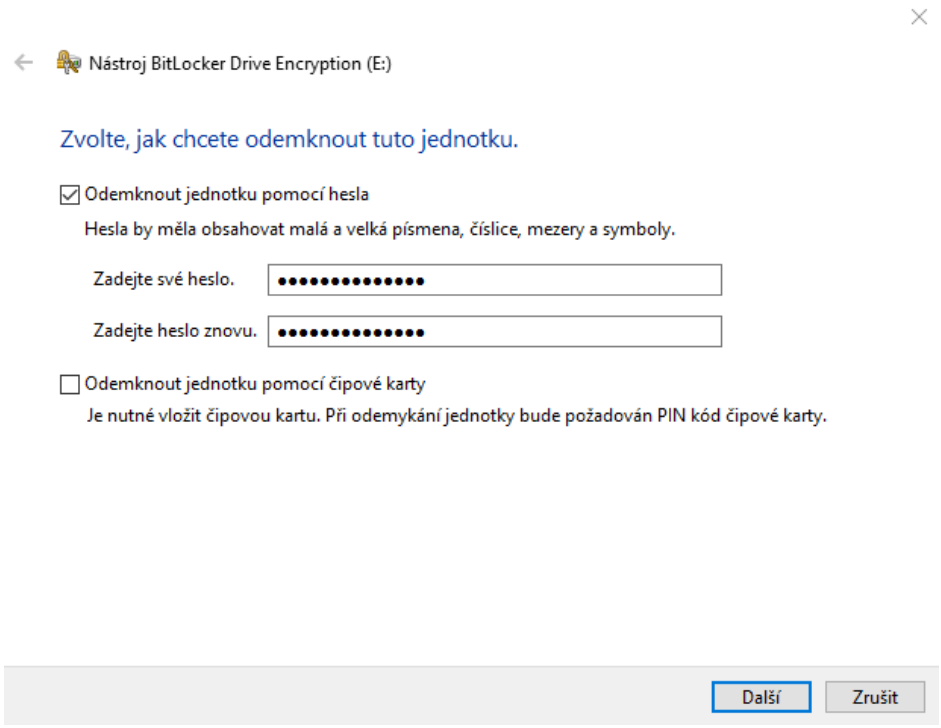
Na tuto možnost klikneme a zvolíme možnost „Zálohovat nyní“, poté se zobrazí průvodce exportem certifikátu. V dalším kroku vybereme formát exportu, viz Obr. 5.16.



Obr. 5.16: Průvodce exportu certifikátu, volba formátu.

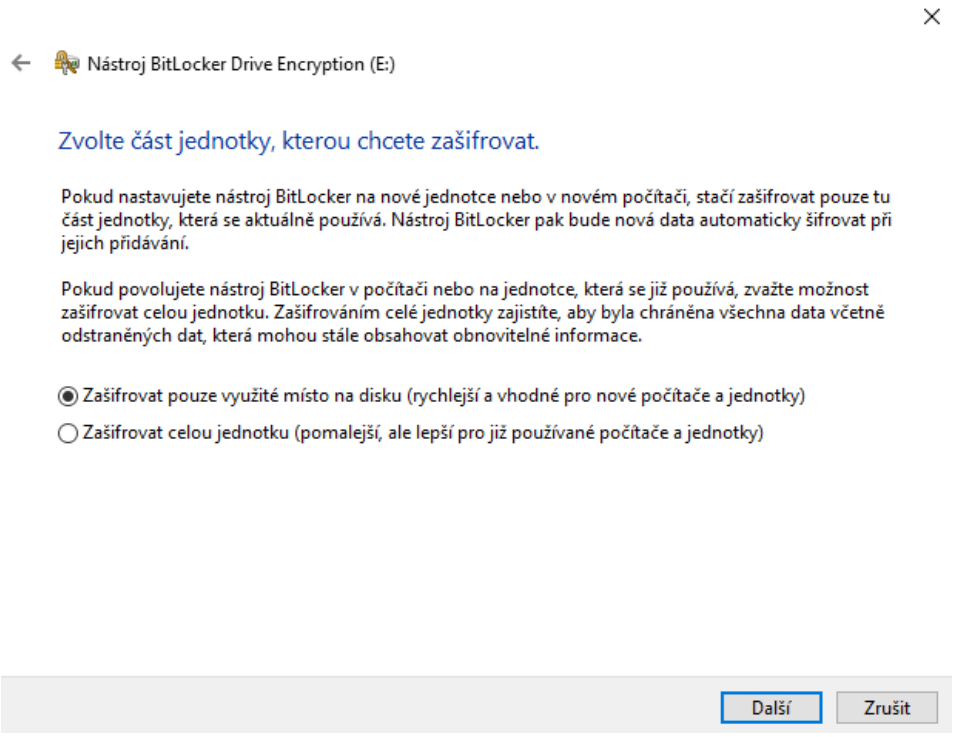
V dalším kroku zadáme heslo, kterým bude chráněn soukromý klíč. V posledním kroku uvedeme název souboru a vybereme jeho umístění. Poté lze složku přenést na jiný PC a po nainstalování soukromého klíče tuto složku lze otevřít.

Šifrování oddílů lze provést pomocí nástroje BitLocker. Vstoupíme do „Tento počítač“ a vybereme diskový oddíl, který bychom chtěli šifrovat. Klikneme na něj pravým tlačítkem myši a zvolíme možnost „Zapnout nástroj BitLocker“. Po kliknutí se zobrazí průvodce. Zvolíme první možnost „Odemknout jednotku pomocí hesla“ a toto heslo vyplníme, viz Obr. 5.17.



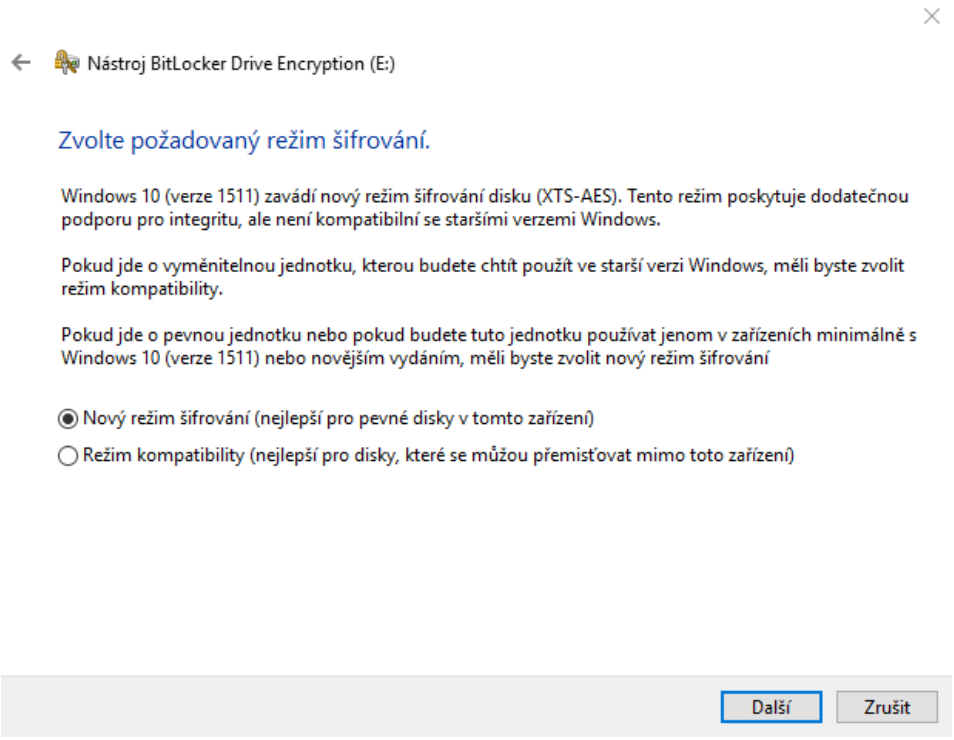
Obr. 5.17: Šifrování diskových oddílů pomocí nástroje BitLocker I.

Následně je zobrazena nabídka, zda chceme zálohovat obnovovací klíč. Zvolíme možnost uložit do souboru, vybereme umístění a potvrdíme. V dalším kroku zvolíme, zda chceme šifrovat pouze využité místo na disku, nebo zda chceme šifrovat celou jednotku. Zvolíme první možnost, viz Obr. 5.18.



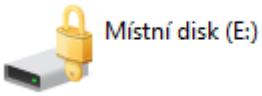
Obr. 5.18: Šifrování diskových oddílů pomocí nástroje BitLocker II.

Následně zvolíme možnost použít nový režim šifrování XTS-AES. Tento režim poskytuje dodatečnou podporu integrity a je vhodný pro pevné disky, viz Obr. 5.19.

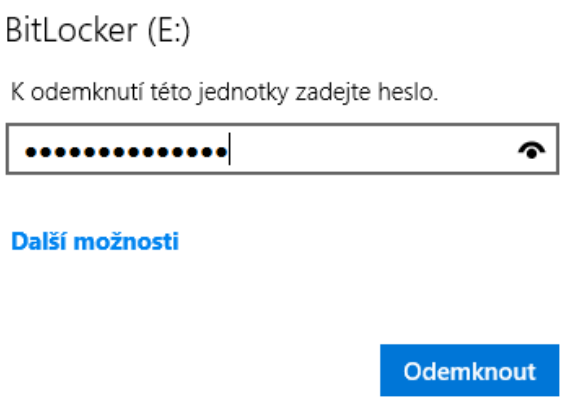


Obr. 5.19: Šifrování diskových oddílů pomocí nástroje BitLocker III.

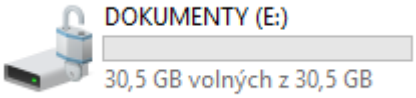
V dalším kroku zahájíme šifrování a poté lze jednotku začít používat. Po zapnutí PC bude třeba zadat přístupové heslo pro dešifrování jednotky. To provedeme tak, že na jednotku klikneme dvakrát levým tlačítkem myši a po zadání hesla je jednotka dešifrována, viz Obr. 5.20, Obr. 5.21 a Obr. 5.22.



Obr. 5.20: Diskový oddíl před dešifrováním.



Obr. 5.21: Zadání přístupového heslo pro dešifrování diskového oddílu.

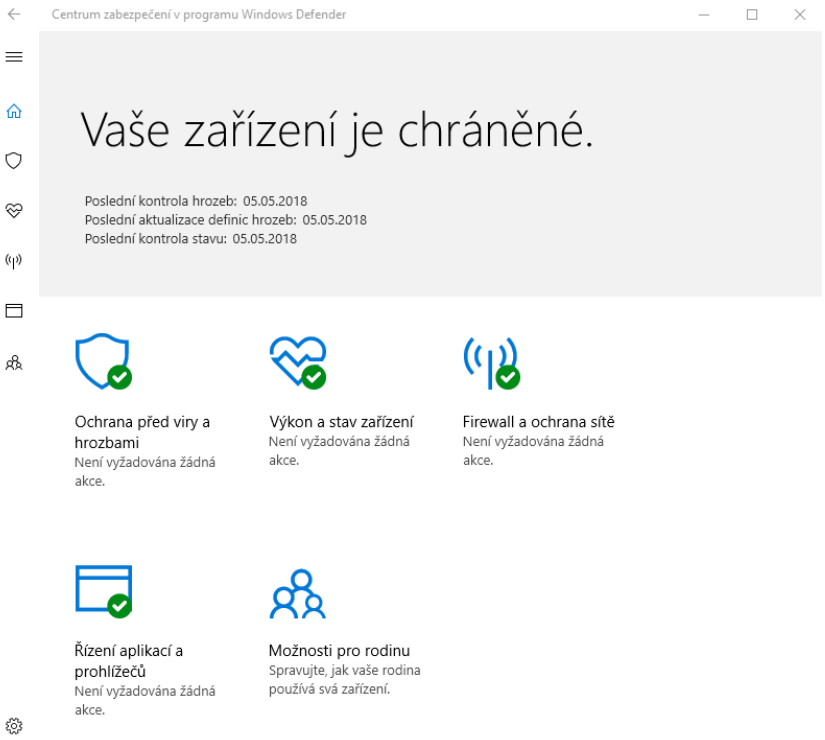


Obr. 5.22: Diskový oddíl po dešifrování.

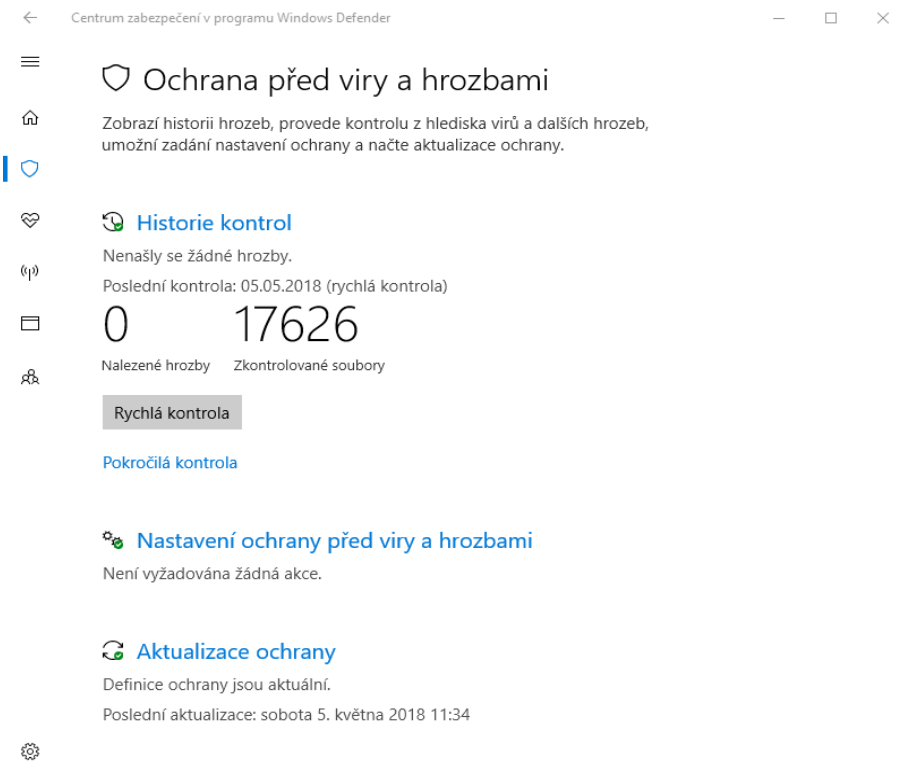
5.1.5 Antivirus

OS Windows v nativním stavu disponuje nástrojem Windows Defender. Spravovat jej lze v „centru zabezpečení v programu Windows Defender“.

V kartě „Domů“, viz Obr. 5.23 je zobrazen globální přehled. V kartě „Ochrana před viry a hrozbami“, viz Obr. 5.24, lze provést „rychlou kontrolu“, popřípadě „pokročilou kontrolu“, tyto kontroly se ovšem provádějí automatizovaně. Dále lze po kliknutí na odkaz „Nastavení ochrany před viry a hrozbami“ ochranu deaktivovat, což z bezpečnostního hlediska není doporučeno. Dále v kartě „Řízení aplikací a prohlížečů“ je vhodné ponechat upozornění na podezřelé soubory a aplikace.



Obr. 5.23: Windows Defender, karta Domů.



Obr. 5.24: Windows Defender, karta ochrana před viry a hrozbami.

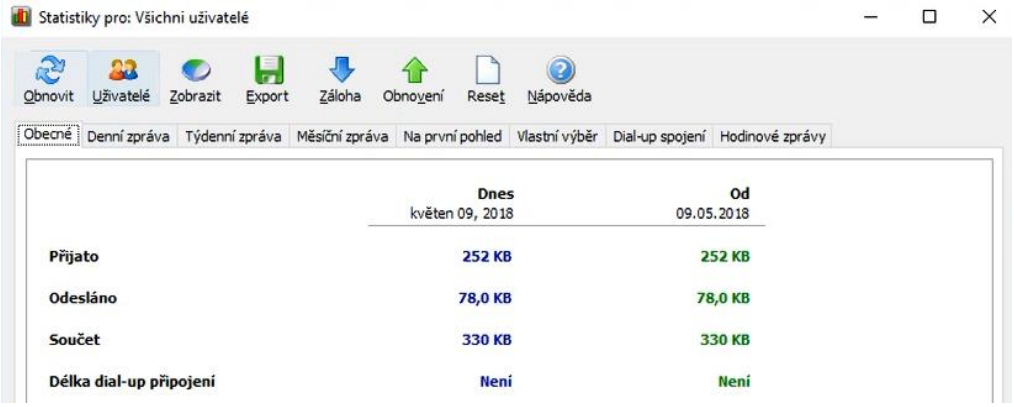
5.1.6 Aplikace zvyšující bezpečnost

Pomocí dodatečných aplikací lze zvýšit bezpečnost systému. Lze monitorovat síťové připojení a další parametry. Ve scénáři provedeme instalaci aplikace NetWorx a aplikace PRTG Network Monitor.

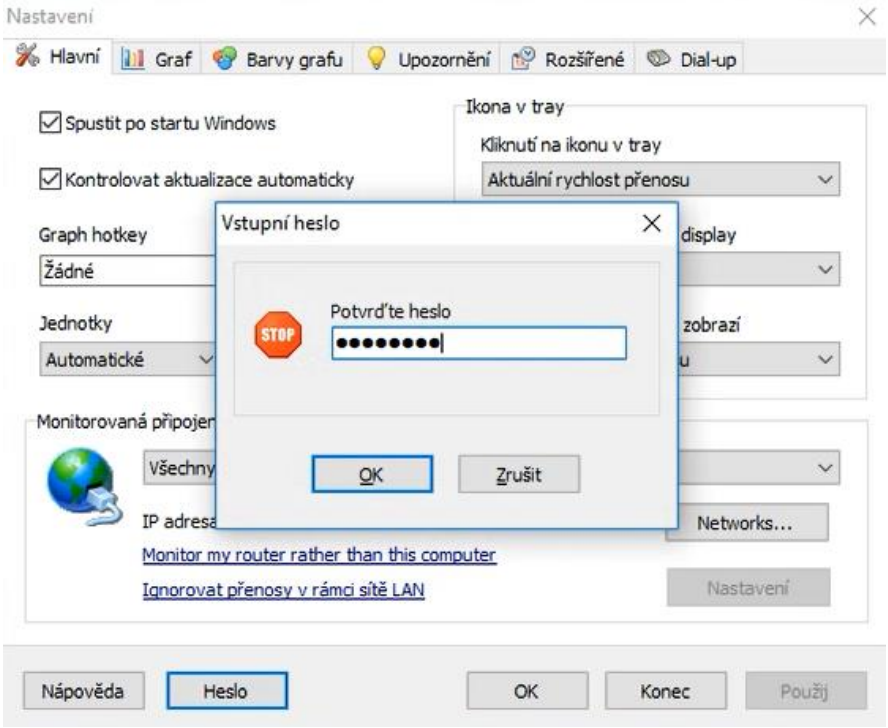
Aplikace NetWorx

Po standartní instalaci lze nalézt ikonu aplikace ve statusové liště. Po kliknutí na obrázek aplikace se zobrazí statistika připojení, viz Obr.5.25. Abychom mohli tuto aplikaci využít k zvýšení bezpečnosti našeho osobního počítače, tak je třeba nashromáždit data. Z bezpečnostního hlediska je tedy vhodné tuto aplikaci nechat v běhu delší dobu a poté vstoupit do nastavení a v kartě „Upozornění“ nastavit limit síťového provozu (dle získaných hodnot průměrně přijatých/odeslaných KB). V případě, že bude limit překročen, tak lze využít „Vizuální a zvukový alarm“. Překročení může značit anomálii, která může být způsobena třetí aplikací. Poté je vhodné zjistit příčinu nárůstu objemu datových jednotek odeslaných přes síť.

Po nastavení limitů je vhodné nastavení uzamknout bezpečným heslem, aby případný útočník toto nastavení nemohl pozměnit. To provedeme v nastavení, dále v kartě „Hlavní“ ve spodní části klikneme na „Heslo“. Poté vložíme heslo a uložíme, viz Obr. 5.26.



Obr. 5.25: Aplikace NetWorx, statistika.



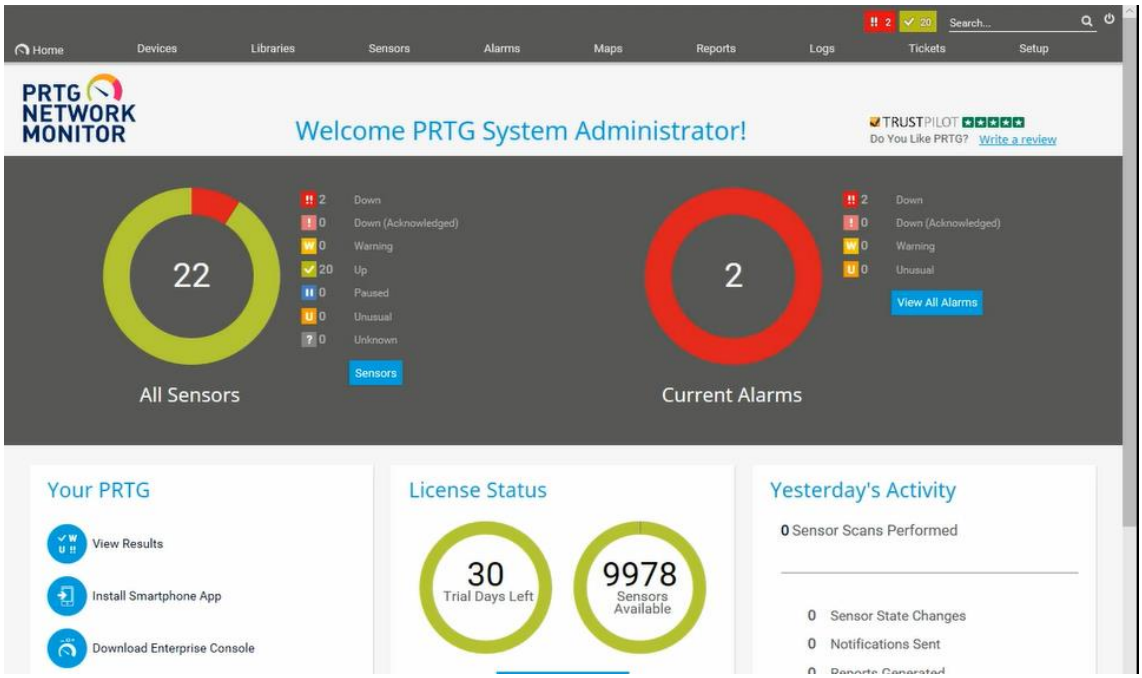
Obr. 5.26: Aplikace NetWorx, zabezpečení heslem.

Aplikace PRTG Network Monitor

K detekci anomálií použijeme aplikaci PRTG Network Monitor. Po standartní instalaci vložíme licenční klíč. Pokud chceme vyzkoušet 30denní zkušební verzi lze kliknout na odkaz a je nám vygenerován licenční klíč k trial verzi.

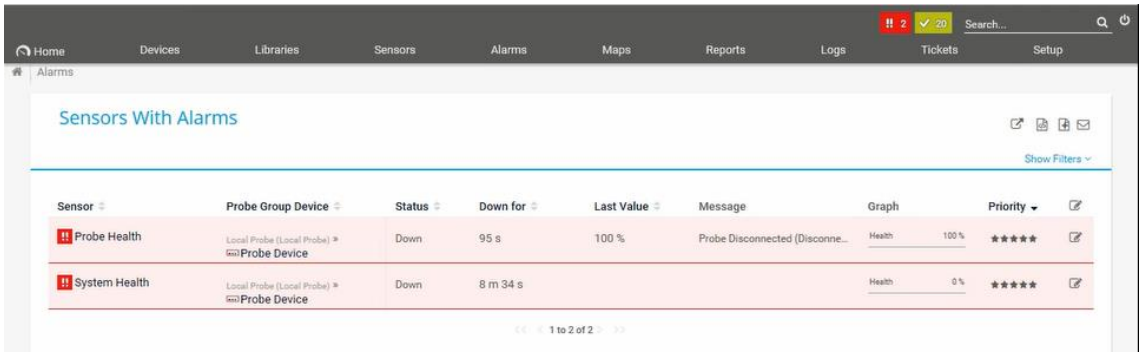
Po spuštění aplikace ve webovém prohlížeči se zobrazí průvodce, který provede uživatele po nově instalované aplikaci a nastaví některé parametry, jako jméno počítače, heslo k systému a další, aby mohla být provedena kontrola jednotlivých prvků, a tak mohly být zapnuty jednotlivé senzory. Dále musíme nastavit nové heslo určené k přihlašování do aplikace. Posledním důležitým krokem je využití protokolu SSL, tzn. že budeme přesměrování z HTTP verze webové stránky na HTTPS verzi webové stránky. Poté jsme přesměrování na úvodní stránku, kde lze vidět aktivní senzory a jednotlivé

alarmy, viz Obr. 5.27. Alarmy značí chybu, kterou hlásí některý ze senzorů (pravá horní strana okna).



Obr. 5.27: PRTG Network Monitor, výchozí obrazovka.

Po kliknutí na „View all alarms“ (popřípadě červený vykřičník v pravém horním rohu) se zobrazí jednotlivé alarmy, které lze následně vyřešit, viz Obr. 5.28.



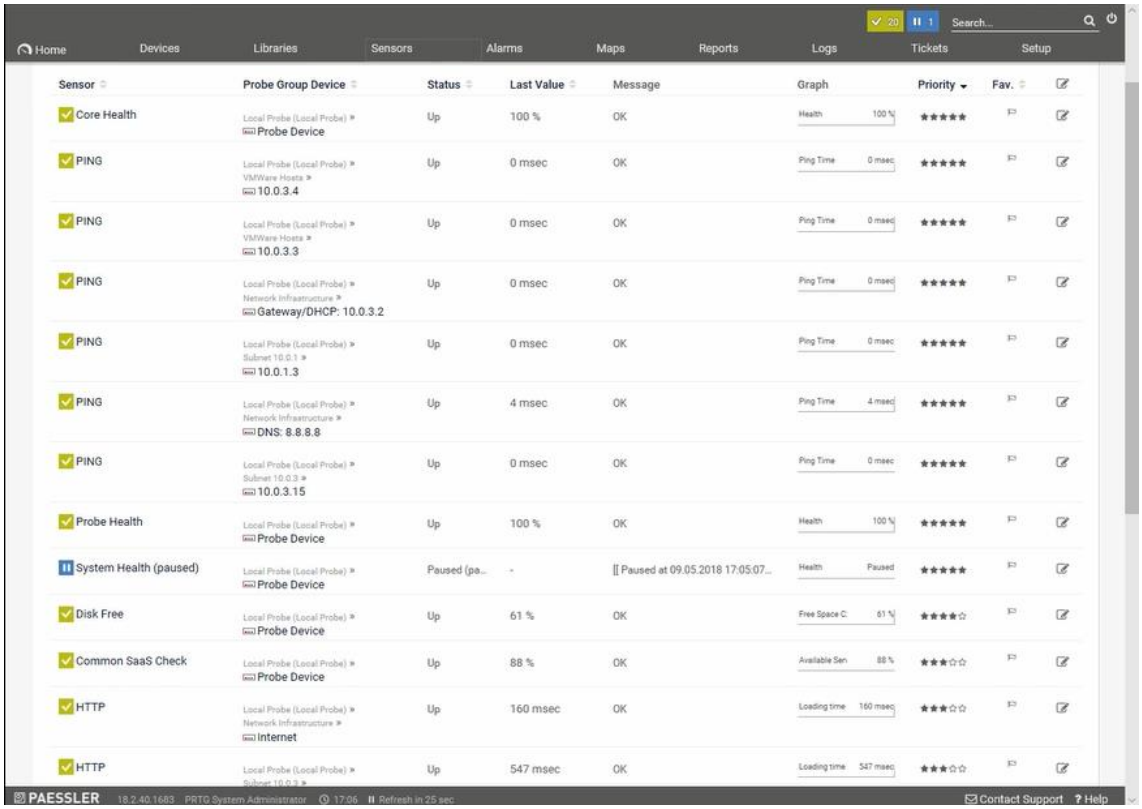
Obr. 5.28: PRTG Network Monitor, alarmy.

Jednotlivé alarmy lze dále rozkliknout a zobrazit si podrobný důvod vzniku, jako například v našem případě problém se „zdravím“ našeho systému. Náš systém hlásí přílišné zatížení, to je ovšem způsobeno virtualizací více systémů na jednom stroji.

Po rozkliknutí „Watch“ (žlutý znak vykřičníku v pravém horním rohu) lze zobrazit alarmy, které nevyhlásili poplach, ale je nutná jejich kontrola.

Pokud bychom chtěli zobrazit jednotlivé logy, lze je rozkliknout v pravém horním rohu, kde se zobrazí text „New Log Entries“.

Jednotlivé senzory lze rozkliknout tak, že klikneme na záložku „Sensors“, viz Obr. 5.29.



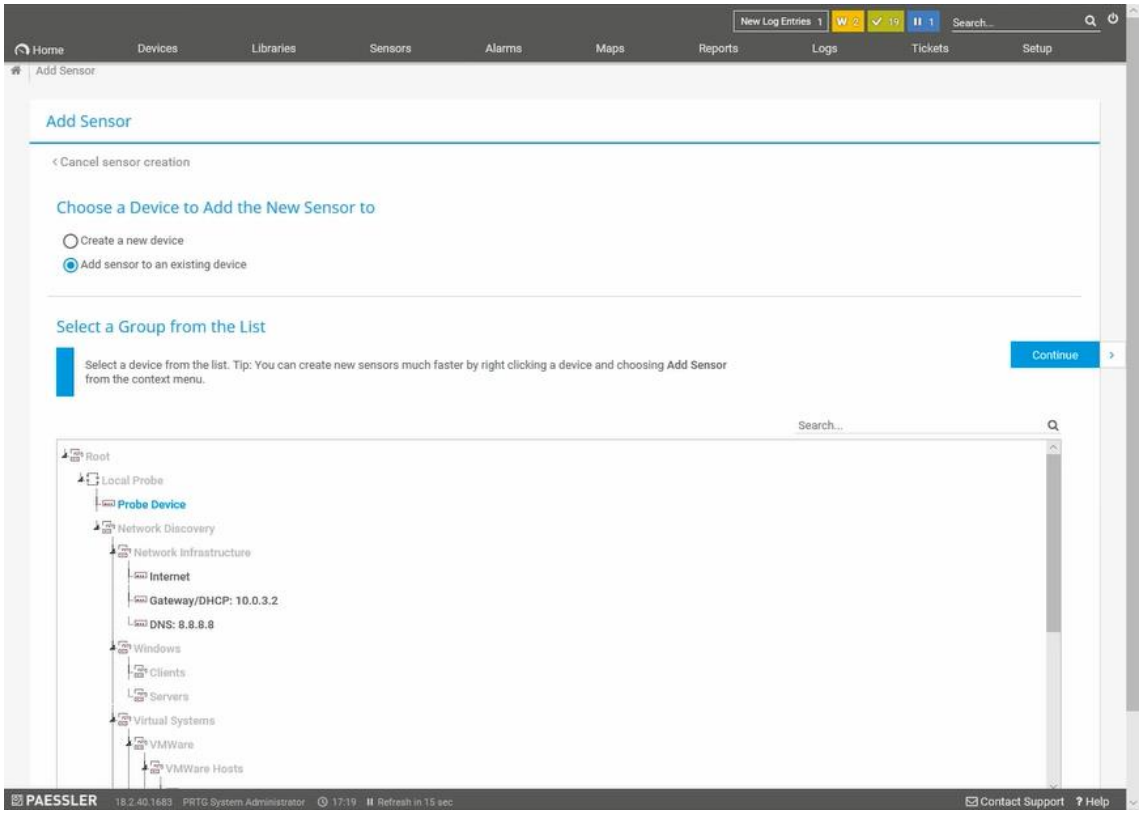
Obr. 5.29: PRTG Network Monitor, instalované senzory.

Když bychom chtěli nastavit nový senzor, klikneme na záložku „Sensors“ a v nabídce vybereme „Add Sensor“, ve spodní nabídce vybereme, pro co by měl být senzor určen, vybereme například „Probe Device“, viz Obr. 5.30. V nově rozbalené nabídce vybereme jednotlivý senzor a klikneme na znaménko plusu, viz Obr. 5.31. Poté lze jednotlivý senzor rozkliknout a zobrazit jeho stav, je ovšem nutné chvíli počkat, než si senzor shromáždí dostatečné množství informací pro vyhodnocení dat, viz Obr. 5.32.

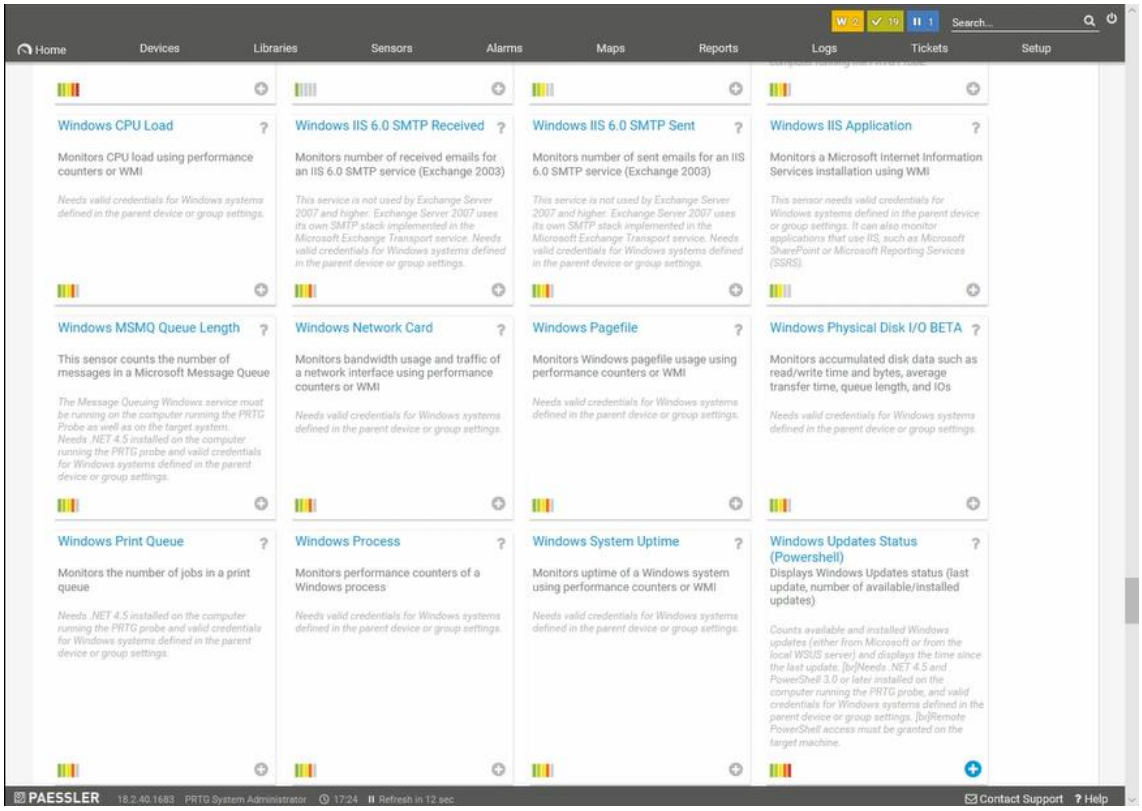
V záložce „Home“ lze zvolit „Dashboard“, kde se zobrazí mapa detekované sítě, jednotlivé alarmy atd., viz Obr. 5.33.

V našem nastavení jsme nechali aktivní předinstalované senzory, které slouží jako detekce anomálií, převážně HW typu (stav procesoru apod.), které indikují nárůst potřebného výkonu. Také je prováděna kontrola sítě. Pokud bychom chtěli, je možné přidávat nová pravidla. Je to ovšem velmi individuální. Lze přidat nový senzor, který bude pomocí příkazu „Ping“ detekovat dostupnost určitého zařízení.

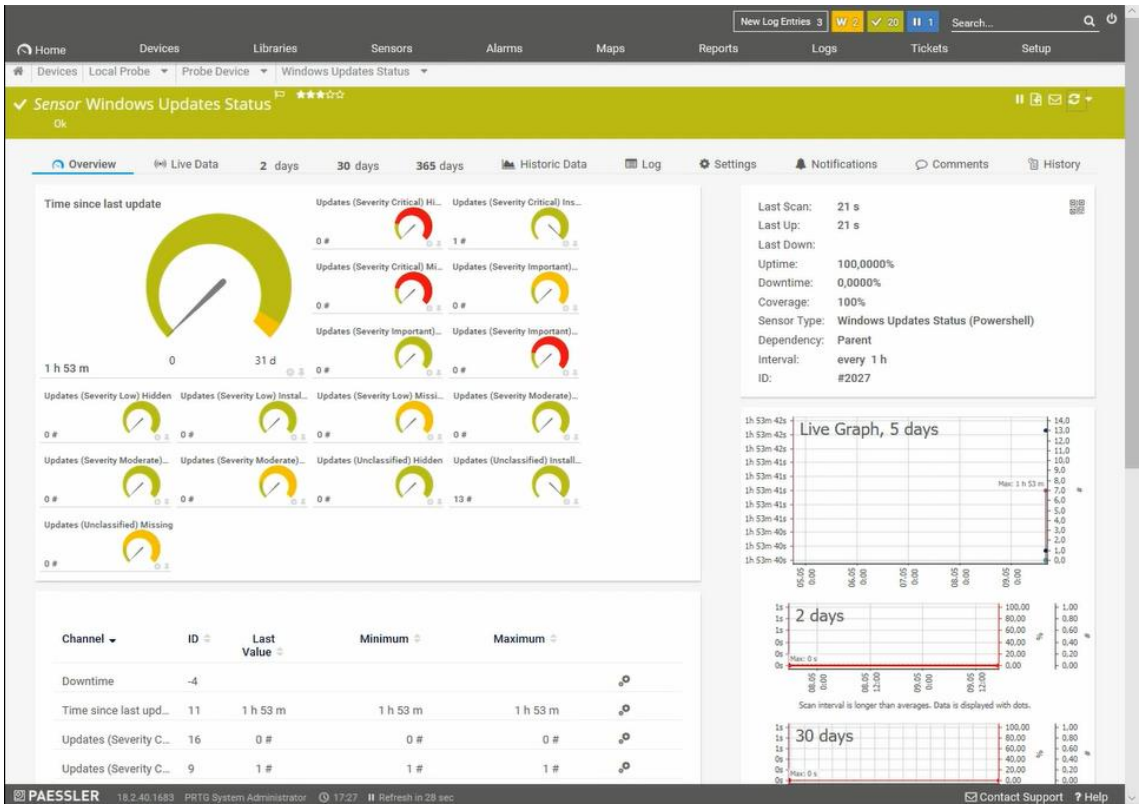
Je tedy doporučeno projít nabídku senzorů a ty, které jsou pro daný uzel relevantní instalovat do aplikace.



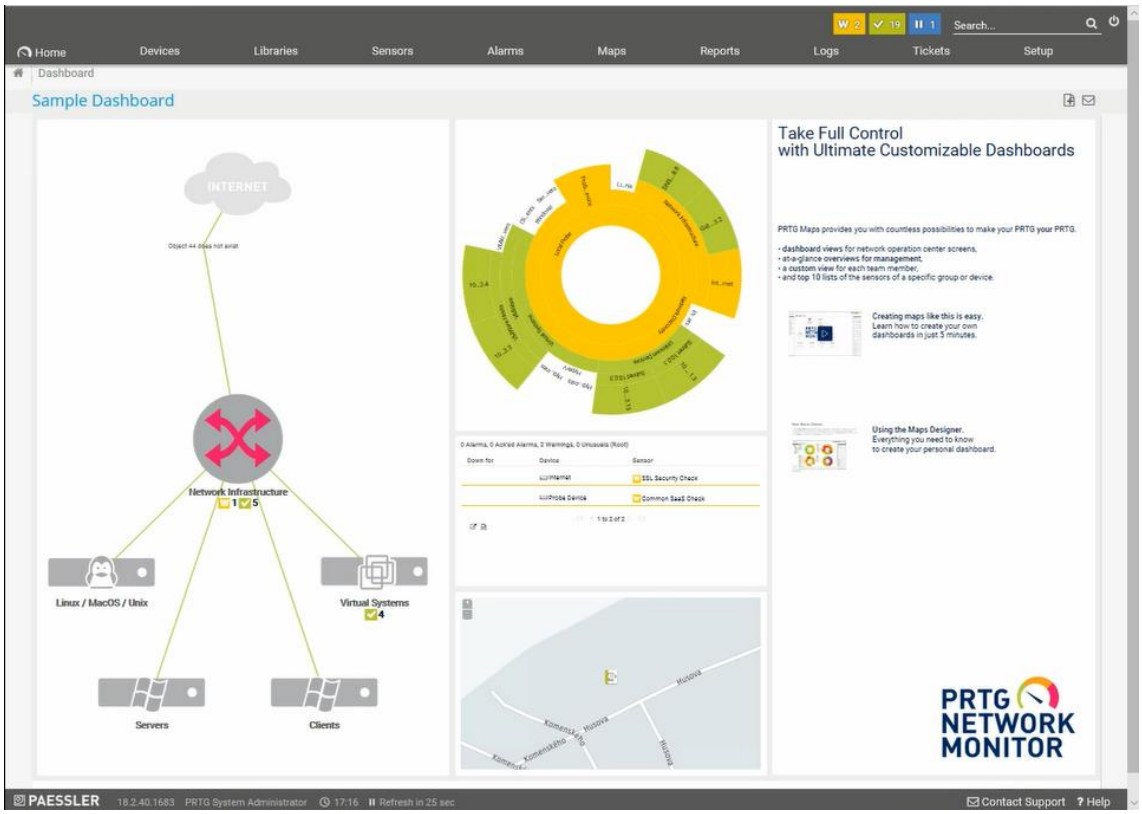
Obr. 5.30: PRTG Network Monitor, přidání nového senzoru I.



Obr. 5.31: PRTG Network Monitor, přidání nového senzoru II.



Obr. 5.32: PRTG Network Monitor, nový senzor.



Obr. 5.33: PRTG Network Monitor, mapa sítě.

5.2 Scénář zabezpečení serveru

Nyní se zaměříme na zabezpečení serverové varianty OS Ubuntu. Z důvodu, že samotná bezpečnost použitého operačního systému je definována během samotné instalace OS, tak bude popsána i samotná instalace OS.

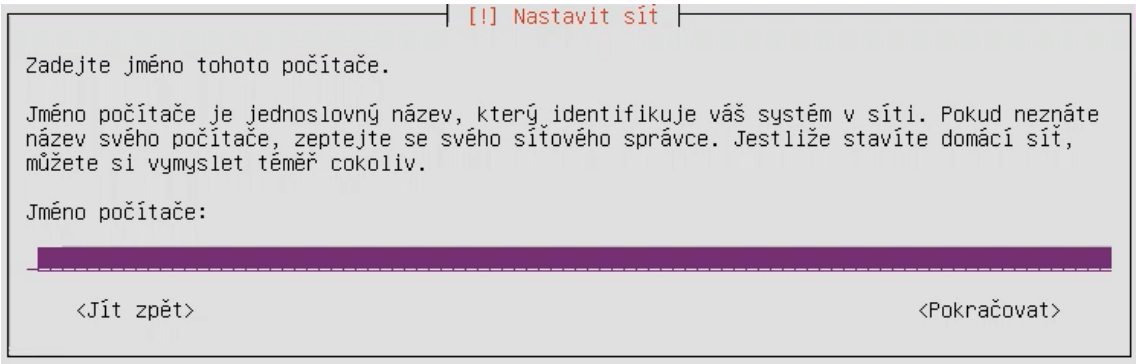
5.2.1 OS, bezpečné heslo, šifrování, přístupová fráze

Jako operační systém pro instalaci na server byl vybrán OS Ubuntu v serverové variantě. Přesněji se jedná o verzi Ubuntu 16.0.4. Jednotlivé kroky scénáře jsou tak upraveny tak, aby byly použitelné pro tuto verzi OS.

Samotná instalace je velmi jednoduchá. Po spuštění instalačního média je nejprve zobrazen seznam možných jazykových balíků, ze kterého vybereme, například „Čeština“, poté se zobrazí možnost instalovat OS, viz Obr.5.34.

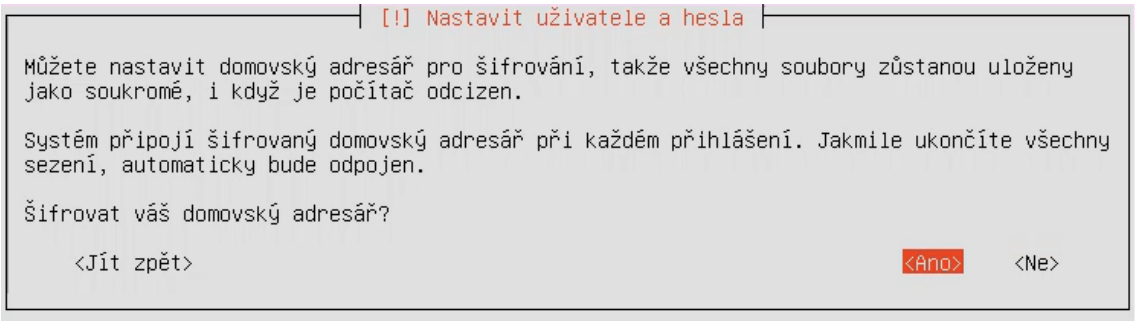
Obr. 5.34: Instalace OS Ubuntu 16.04.

Dále vybereme umístění, ve kterém je umístěn OS, aby bylo časové pásmo nastaveno korektně. Poté identifikujeme nastavení klávesnice (lze provést automaticky, je zadán text, abychom napsali znak závorky a při stisku klávesy „Shift“ je identifikována česká klávesnice). Následně uvedeme název počítače, viz Obr. 5.35



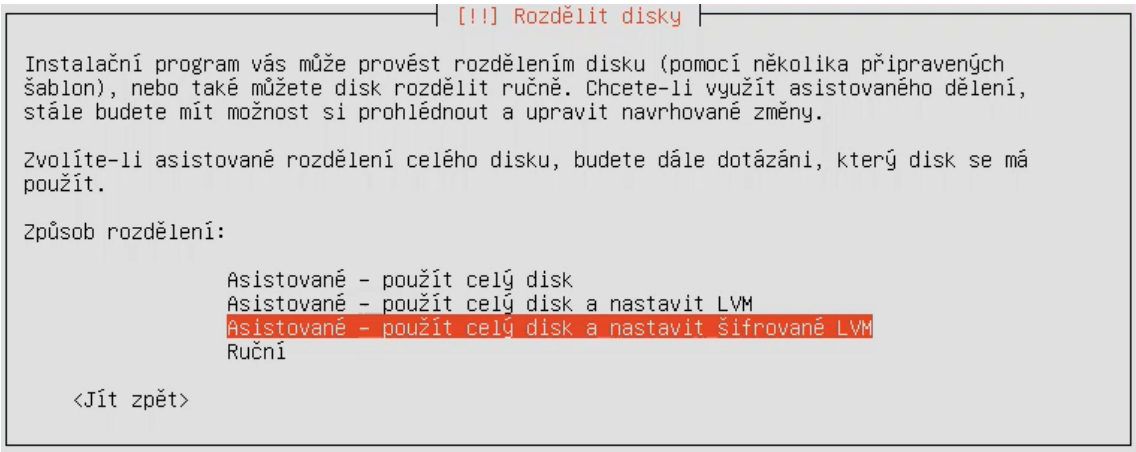
Obr. 5.35: Nastavení názvu počítače.

Dále uvedeme jméno uživatele a zadáme heslo, pokud je toto heslo kratší než osm znaků, tak jsme upozorněni a máme možnost tento „nedostatek“ napravit. Použijeme bezpečné heslo „\DZvr1__L)&6D[N.XW]C“. V dalším kroku máme možnost nastavit, že domovský adresář bude šifrován, viz Obr. 5.36. Je na zvážení každého správce, jakou možnost zvolit. V tomto scénáři bude zvoleno možnost „ANO“.



Obr. 5.36: Šifrování domovského adresáře.

V dalším kroku můžeme zvolit, zda chceme šifrovat logické svazky. Pokud zvolíme poslední možnost, viz Obr. 5.37, tak je nutné při každém startu OS zadat přístupovou frázi (doporučená délka je 20 znaků), bez které není možné korektně nabootovat systém. V dalších krocích schválíme požadované změny.



Obr. 5.37: Rozdělení disku.

Mezi posledními kroky máme na výběr, jak chceme zacházet s aktualizacemi, zda je instalovat manuálně, nebo instalovat pouze bezpečnostní aktualizace, nebo spravovat systém pomocí „Landscape“. V tomto scénáři byla vybrána možnost „Žádné automatické aktualizace“, abychom měli absolutní kontrolu nad instalovanými aktualizacemi.

V posledním kroku jsme dotázáni, zda chceme instalovat „GRUB“ tedy zavaděč systému. Pokud se jedná o jediný OS instalovaný na počítači, tak je doporučeno zvolit možnost instalovat, a tak je provedena instalace do MBR (master boot record).

5.2.2 Aktualizace

Aktualizace jsou prováděny velice jednoduše. Abychom provedli aktualizaci, tak nám postačuje příkaz „`sudo apt-get update`“ a pro upgrade „`sudo apt-get upgrade`“, viz Obr. 5.38.

```
student@ubuntu:~$ sudo apt-get update
Mám:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Mám:2 http://cz.archive.ubuntu.com/ubuntu xenial InRelease
Mám:3 http://cz.archive.ubuntu.com/ubuntu xenial-updates InRelease
Mám:4 http://cz.archive.ubuntu.com/ubuntu xenial-backports InRelease
Načítají se seznamy balíků… Hotovo
student@ubuntu:~$ sudo apt-get upgrade_
```

Obr. 5.38: Update, upgrade.

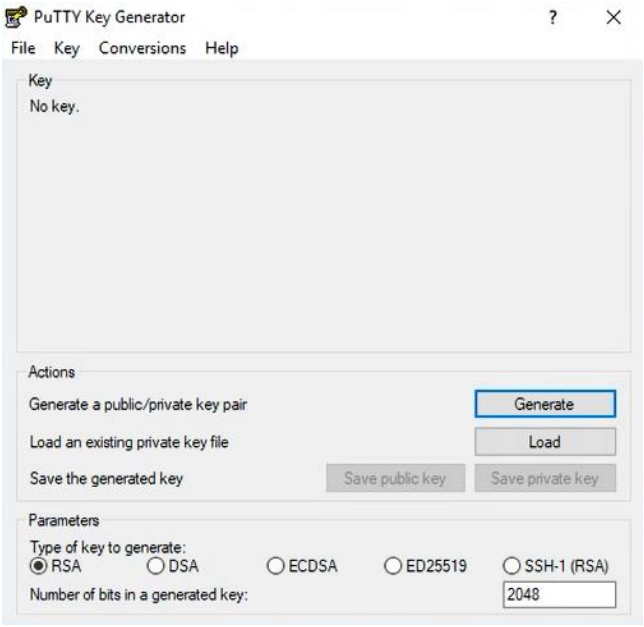
Ovšem je zde riziko, i když mnohdy nízké, že aktualizace mohou učinit služby, které server poskytuje neschopné provozu z důvodu provedení aktualizace. Proto je možné využít nástroje „aptitude“, který provádí aktualizaci tak, že instaluje jen nové balíčky a neprovede odinstalaci žádného z instalovaných balíčků, které by mohli učinit některou ze služeb neschopnou provozu.

Instalace nástroje „aptutide“ se provede zadáním příkazu „`sudo apt install aptitude`“. Samotné provedení bezpečného upgradu se provede pomocí příkazu „`sudo aptitude safe-upgrade`“.

5.2.3 SSH

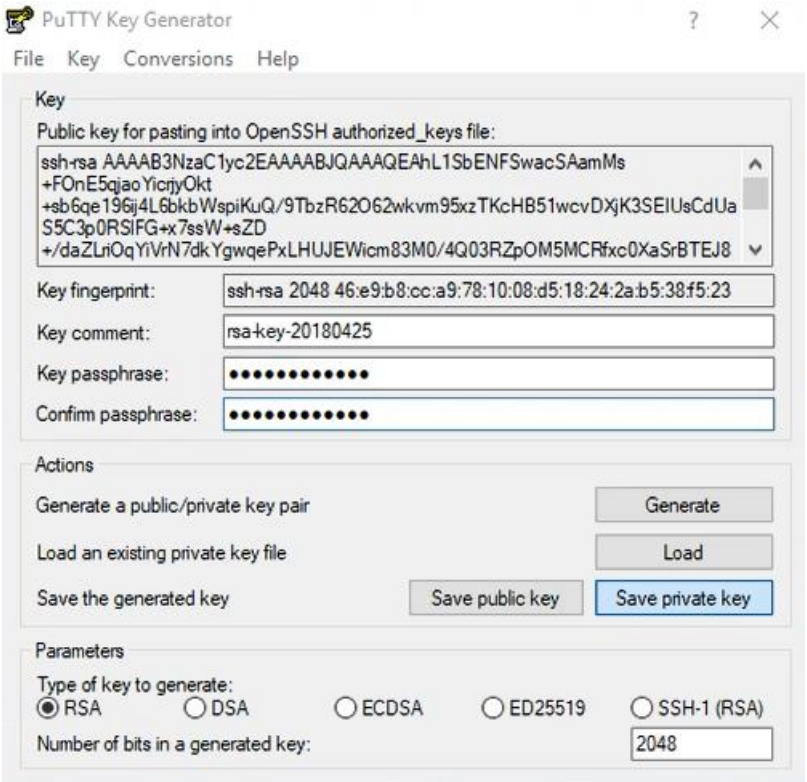
Instalaci SSH na serveru provedeme příkazem „`sudo apt-get install openssh-server`“. Dále pokud budeme chtít přenést veřejný klíč uživatele, který se bude přihlašovat z OS Windows (aby bylo možné se přihlašovat pouze certifikátem), použijeme aplikaci Putty.

V OS Windows spustíme aplikaci puttygen.exe a klikneme na tlačítko „Generate“, viz Obr. 5.39, poté je nutný pohyb kurzoru po šedé ploše, abychom získali dostatek náhodných dat ke generování klíčů.



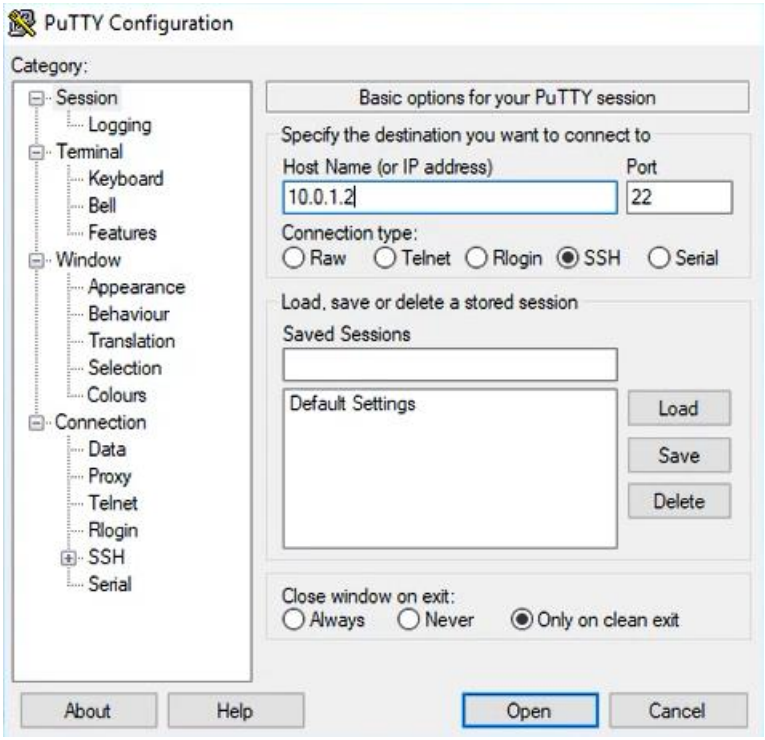
Obr. 5.39: Putty key generator.

Po vygenerování klíčů zadáme heslo pro ochranu soukromého klíče uživatele a klíč si uložíme, viz Obr. 5.40. Dále zkopírujeme veřejný klíč uživatele, který je umístěn v horní části.



Obr. 5.40: Putty key generator, uložení klíče.

V dalším kroku se přihlásíme pomocí aplikace putty, vyplníme IP adresu a klikneme na tlačítko „Open“, viz Obr. 5.41.



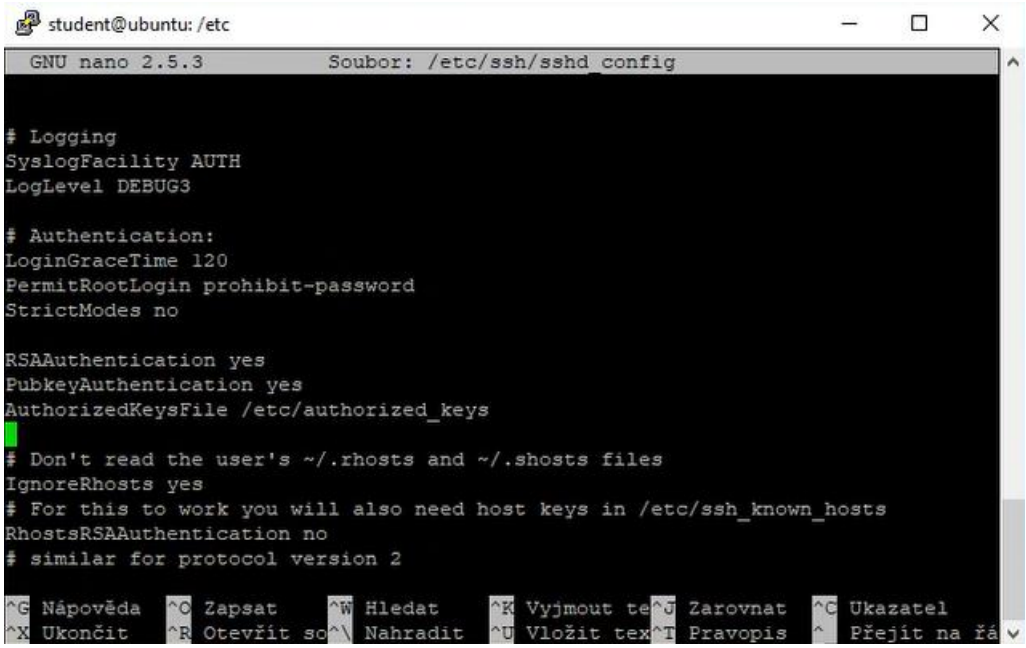
Obr. 5.41: Připojení přes SSH pomocí putty.

Po přihlášení musí veřejný klíč uživatele vložit do souboru „authorized_keys“, ovšem z důvodu, že jsme aplikovali šifrování domovského adresáře uživatele, tak by přihlášení nebylo možné ověřit. Musíme tedy změnit umístění.

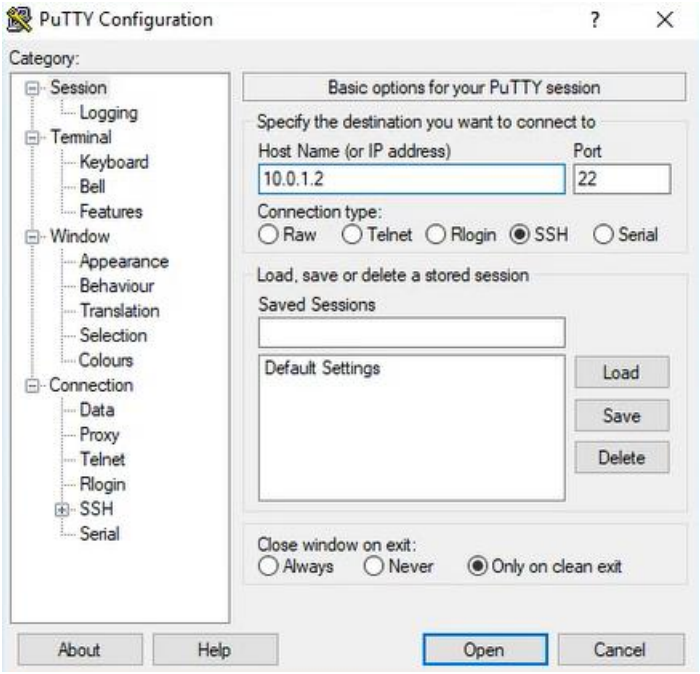
Klíče můžeme uložit například do adresáře „/etc/“, pro přechod do tohoto adresáře zadáme příkaz „`cd /etc/`“ a zde vytvoříme soubor „authorized_keys“ příkazem „`sudo nano authorized_keys`“, po otevření klikneme pravým tlačítkem myši a vložíme veřejný klíč uživatele.

Nyní je třeba změnit konfigurační soubor. Vstoupíme do něj pomocí příkazu „`sudo nano /etc/ssh/sshd_config`“. Odkomentujeme direktivu „`AuthorizedKeysFile`“ a změníme hodnotu na cestu k souboru s veřejnými klíči povolených uživatelů na hodnotu „/etc/authorized_keys“, viz Obr. 5.42. V tomto konfiguračním souboru zakážeme přihlašování heslem. Nalezneme tedy direktivu „`PasswordAuthentication`“ a změníme hodnotu na „`no`“.

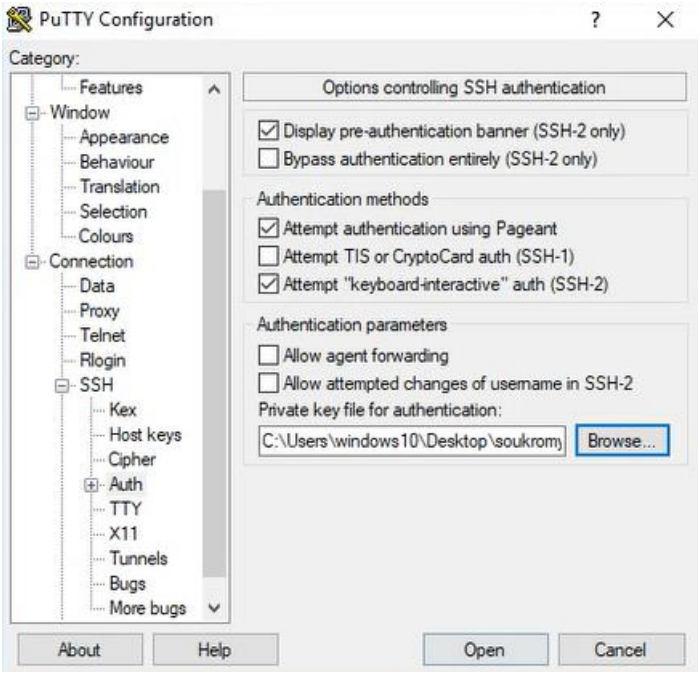
Poté provedeme restart služby pomocí příkazu „`sudo service ssh restart`“. Nyní je možné se přihlásit pouze za pomocí certifikátu. Otevřeme putty, do pole „Host Name“ zadáme IP adresu cíle (v našem případě 10.0.1.2), viz Obr.5.43. Dále v levém menu „Category“ klikneme na „Connection“, dále „SSH“ a poté „Auth“. Do pole „Private key file for authentication“ vložíme cestu k soukromému klíči a klikneme na „Open“, viz Obr. 5.44. V otevřeném terminálu zadáme uživatelské jméno, poté heslo k soukromému klíči a jsme přihlášeni, viz Obr. 5.45.



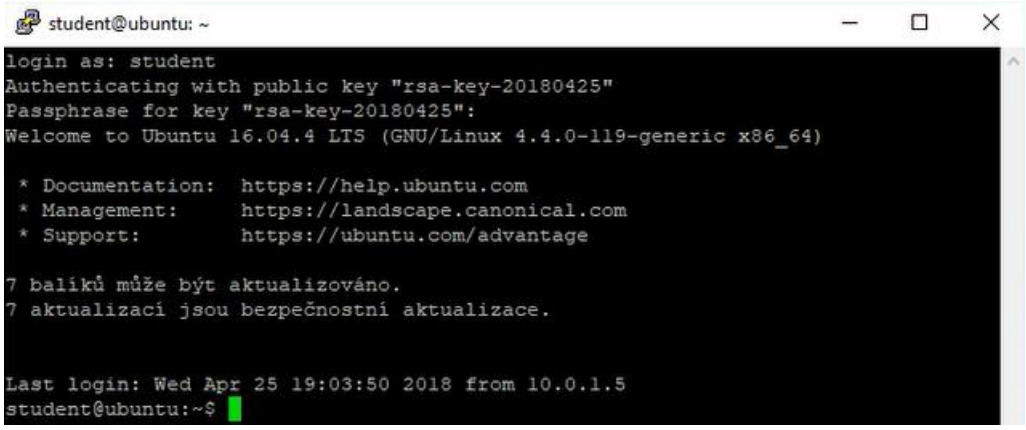
Obr. 5.42: Konfigurační soubor SSH.



Obr. 5.43: Připojení přes SSH pomocí certifikátu I.



Obr. 5.44: Připojení přes SSH pomocí certifikátu II.



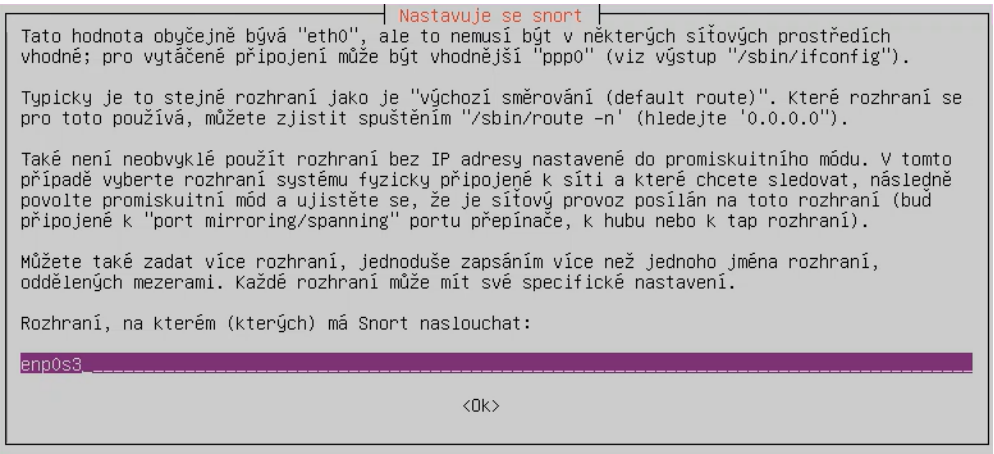
Obr. 5.45: Připojení přes SSH pomocí certifikátu III.

5.2.4 IDS

Pro provedení IDS byl vybrán nástroj Snort. Jeho instalace je obdobná jako u většiny balíčků, zadáme tedy příkaz „`sudo apt-get install snort`“. Po vykonání se zobrazí okno, které nás vyzve k zadání názvu rozhraní, přes který má být síťový provoz detekován. V našem případě se rozhraní jmenuje „enp0s3“, tento název je nastaven Virtualboxem, viz Obr.5.46.

V dalším kroku zadáme adresní rozsah sítě v našem případě „10.0.1.0/24“. Opět potvrdíme název naslouchajícího rozhraní, vybereme možnost „Ok“.

Verzi si lze zobrazit zadáním příkazu „`sudo snort -v`“, viz Obr. 5.47. Test konfigurace provedeme příkazem „`sudo snort -T -i enp0s3 -c /etc/snort/snort.conf`“. Výstupem je načtení všech souborů a pokus o start, v případě úspěchu je na posledním řádku napsáno „Snort successfully validated the Configuration!“, viz Obr. 5.48.



Obr. 5.46: Definování naslouchajícího rozhraní snortu.

```
student@ubuntu:~$ sudo snort -v
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "enp0s3".
Decoding Ethernet

        --== Initialization Complete ==--

           -*> Snort! <*-
  ,,_      Version 2.9.7.0 GRE (Build 149)
 o"  )~    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
   ''''    Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.38 2015-11-23
           Using ZLIB version: 1.2.8

Commencing packet processing (pid=5340)
```

Obr. 5.47: Zobrazení verze snortu.

```
        --== Initialization Complete ==--

           -*> Snort! <*-
  ,,_      Version 2.9.7.0 GRE (Build 149)
 o"  )~    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
   ''''    Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.38 2015-11-23
           Using ZLIB version: 1.2.8

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.4  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>

Snort successfully validated the configuration!
Snort exiting
```

Obr. 5.48: Test konfigurace snortu.

Jednotlivá pravidla (soubory s pravidly) snortu nalezneme v „/etc/snort/rules“, viz Obr. 5.49, která lze po otevření modifikovat, popřípadě vytvářet pravidla vlastní. Ihned po instalaci je k dispozici velké množství pravidel. V konfiguračním souboru „etc/snort/snort.conf“ lze určit, které soubory s pravidly budou použity.

```
student@ubuntu:~$ cd /etc/snort/rules/
student@ubuntu:/etc/snort/rules$ ls
attack-responses.rules          community-web-iis.rules   pop2.rules
backdoor.rules                  community-web-misc.rules  pop3.rules
bad-traffic.rules               community-web-php.rules   porn.rules
community-bot.rules             ddos.rules                p2p.rules
community-deleted.rules         deleted.rules             rpc.rules
community-dos.rules             dns.rules                 rservices.rules
community-exploit.rules         dos.rules                 scan.rules
community-ftp.rules             experimental.rules        shellcode.rules
community-game.rules            exploit.rules             smtp.rules
community-icmp.rules            finger.rules              snmp.rules
community-imap.rules            ftp.rules                 sql.rules
community-inappropriate.rules   chat.rules                telnet.rules
community-mail-client.rules     icmp-info.rules           tftp.rules
community-misc.rules            icmp.rules                virus.rules
community-nntp.rules            imap.rules                web-attacks.rules
community-oracle.rules          info.rules                web-cgi.rules
community-policy.rules          local.rules               web-client.rules
community-sip.rules             misc.rules                web-coldfusion.rules
community-smtp.rules            multimedia.rules          web-frontpage.rules
community-sql-injection.rules   mysql.rules               web-iis.rules
community-virus.rules           netbios.rules             web-misc.rules
community-web-attacks.rules     nntp.rules                web-php.rules
community-web-cgi.rules         oracle.rules              x11.rules
community-web-client.rules      other-ids.rules
community-web-dos.rules         policy.rules
student@ubuntu:/etc/snort/rules$
```

Obr. 5.49: Soubory s pravidly snortu.

Nástroj Snort ovšem nedefinuje veškeré způsoby, jak detekovat skenování portů, například pomocí příkazu nmap. Vytvoříme si tedy několik vlastních pravidel. Nejprve vstoupíme do souboru s vlastními pravidly „`sudo nano /etc/snort/rules/local.rules`“. Do tohoto souboru vložíme příkazy, viz Obr. 5.50.

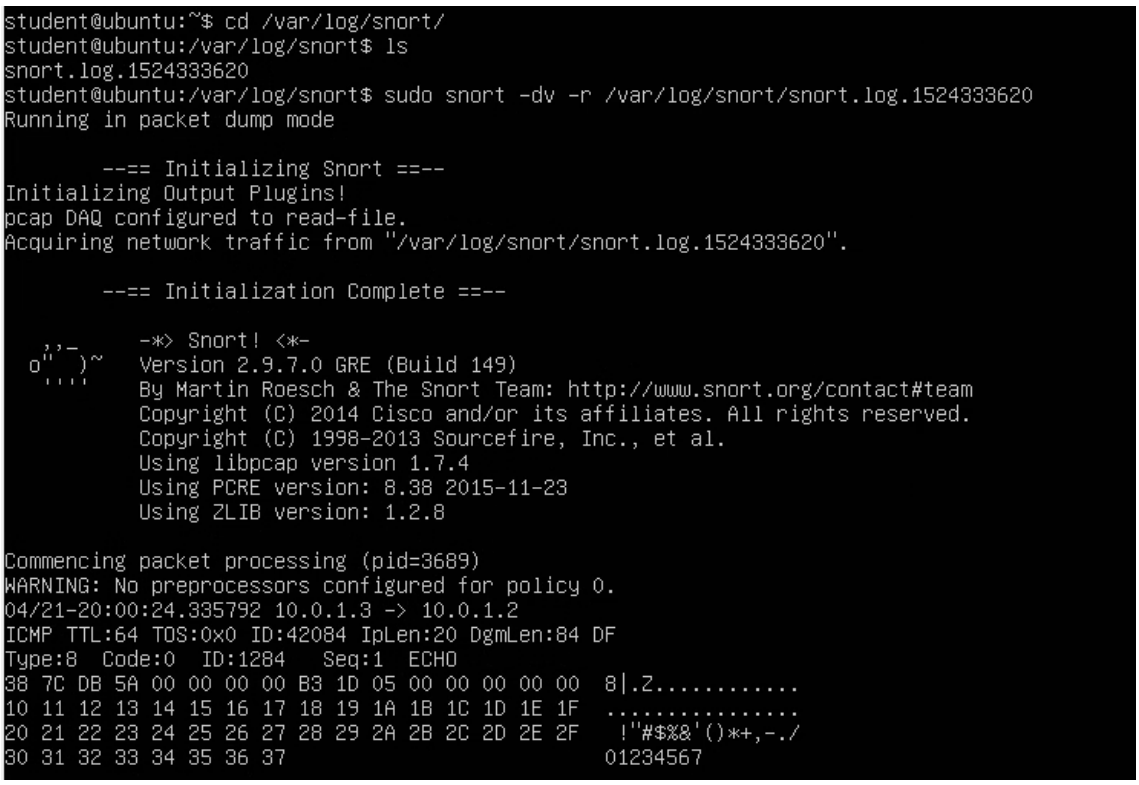
```
# LOCAL RULES
# ---------------
# This file intentionally does not come with signatures.  Put your local
# additions here.

log icmp any any -> 10.0.1.2 any (msg:NMAP ping sweep scan"; dsize:0;sid:10000004; rev:1;)
log tcp any any -> 10.0.1.2 22 (msg:"NMAP tcp scan";sid:10000005; rev:2;)
log tcp any any -> 10.0.1.2 22 (msg:"NMAP XMAS tree scan"; flags:FPU; sid:10000006; rev:1;)
log tcp any any -> 10.0.1.2 22 (msg:"NMAP Fin scan"; flags:F; sid:1000008; rev:1;)
log tcp any any -> 10.0.1.2 22 (msg:"NMAP NULL scan"; flags:0; sid:1000009; rev:1;)
log udp any any -> 10.0.1.2 any (msg:"NMAP udp scan"; sid:1000010; rev:1;)
```

Obr. 5.50: Vlastní pravidla pro detekci skenu portů.

Snort lze spustit několika způsoby. Nejjednodušším z nich je použít příkaz „`sudo snort -A console -q -v -c /etc/snort/snort.conf -i enp0s3`“. Pozn. „enp0s3“ je název použitého rozhraní. Na Obr. 5.51 je vidět spuštění snortu a detekce příkazu Ping.

Obr. 5.51: Snort, detekce přkazu Ping.

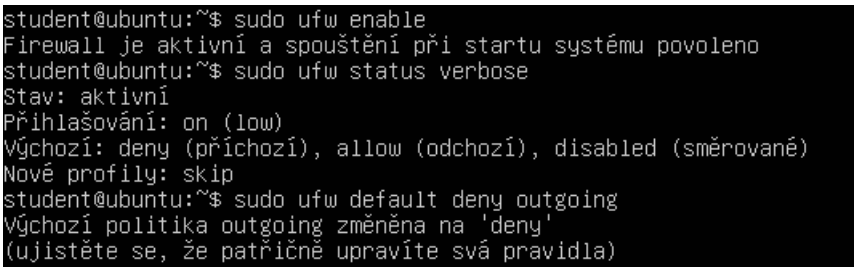
Jednotlivé logy jsou uloženy v „/var/log/snort“, jejich přečtení lze provést příkazem „`sudo snort -dv -r /var/log/snort/<název_logu>`“, viz Obr. 5.52.



Obr. 5.52: Snort log.

5.2.5 Firewall

Ke správě firewallu lze použít nástroj UFW. Nejprve je nutné nástroj povolit, to uděláme příkazem „`sudo ufw enable`“. Poté lze změnit výchozí politiky, to uděláme pomocí příkazu „`sudo ufw default deny outgoing`“, viz Obr. 5.53.



Obr. 5.53: UFW, výchozí politiky.

K nastavení jednotlivých pravidel lze využít jednoduchého příkazu „`sudo ufw allow <název_služby>`“, popřípadě `sudo ufw allow out <název_služby>`“. Pokud přidáme parametr out, definujeme povolení pro provoz ze stroje. Jak si lze na Obr. 5.54 povšimnout byl povolen i port 53. Toto je provedeno, aby bylo možné úspěšně provést příkazy „`apt-get update`“ a „`apt-get upgrade`“. Také je doporučeno povolit logování pomocí příkazu „`sudo ufw logging on`“. Ovšem je na Firewall nutné myslet při každé instalaci nové služby, jedná se o častou příčinu nefunkčnosti nové služby.

```
student@ubuntu:~$ sudo ufw allow http
Pravidla aktualizována
Pravidla aktualizována (v6)
student@ubuntu:~$ sudo ufw allow https
Pravidla aktualizována
Pravidla aktualizována (v6)
student@ubuntu:~$ sudo ufw allow ssh
Pravidla aktualizována
Pravidla aktualizována (v6)
student@ubuntu:~$ sudo ufw allow out http
Pravidla aktualizována
Pravidla aktualizována (v6)
student@ubuntu:~$ sudo ufw allow out https
Pravidla aktualizována
Pravidla aktualizována (v6)
student@ubuntu:~$ sudo ufw allow out ssh
Pravidla aktualizována
Pravidla aktualizována (v6)
student@ubuntu:~$ sudo ufw allow out 53
Pravidla aktualizována
Pravidla aktualizována (v6)
student@ubuntu:~$ sudo ufw logging on
Záznam zapnutý
student@ubuntu:~$
```

Obr. 5.54: UFW, jednotlivá pravidla.

Pokud je třeba jednotlivá pravidla resetovat, můžeme použít příkaz „`sudo ufw reset`“. Ovšem tento příkaz se nijak netýká výchozích politik, ty je nutné změnit „ručně“.

5.2.6 Zálohování

K zálohování lze využít velké množství nástrojů k záloze celého systému nám postačuje nástroj „tar“. Do terminálu zadáme příkaz „`tar czf /backup.tgz --one-file-system --ignore-failed-read --sparse --exclude=/backup.tgz /`“, viz Obr. 5.55. Záloha je dle příkazu uložena přímo v kořenovém adresáři.

```
student@ubuntu:/var$ sudo tar czf /backup.tgz --one-file-system --ignore-failed-read --sparse --excl
ude=/backup.tgz /
tar: Odstraňuje se úvodní „/„ z názvů prvků
tar: Odstraňuje se úvodní „/„ z cílů pevného odkazu
tar: /var/lib/lxcfs: soubor byl během čtení změněn
tar: /var/lib/lxd/unix.socket: soket ignorován
tar: /: soubor byl během čtení změněn
student@ubuntu:/var$
```

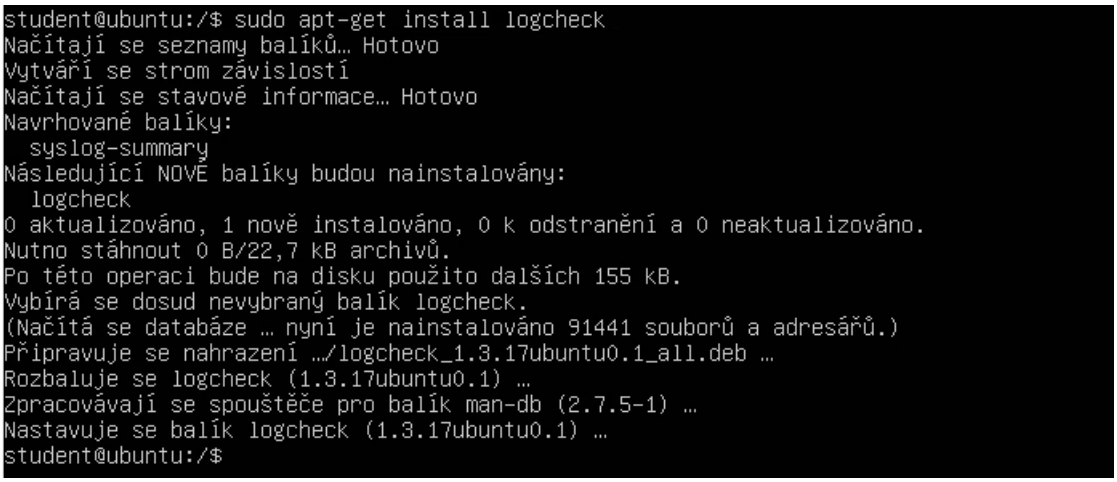
Obr. 5.55: Záloha pomocí nástroje tar.

5.2.7 Logování

OS Ubuntu provádí automatické logování do souboru ve „/var/log“, viz Obr. 5.56.

Obr. 5.56: Adresář pro logy.

Pro automatizované zacházení s logy využijeme nástroj „logcheck“. Instalaci provedeme příkazem „`sudo apt-get install logcheck`“, viz Obr. 5.57. Po vykonání je zobrazena možnost nastavení emailového serveru. Jednotlivé logy prochází kritériemi a pokud se jedná o závažné logy jsou zaslány na email. Ve výchozím stavu se jedná o email roota, který je umístěn v „/var/mail“. Pokud bychom chtěli zasílat logy mimo server je nutné provést změnu v konfiguračním souboru, který je umístěn v „/etc/logcheck/logcheck.conf“, kde pozměníme parametr „senmailto“. Ovšem nesmíme poté opomenout provést změnu ve Firewallu.



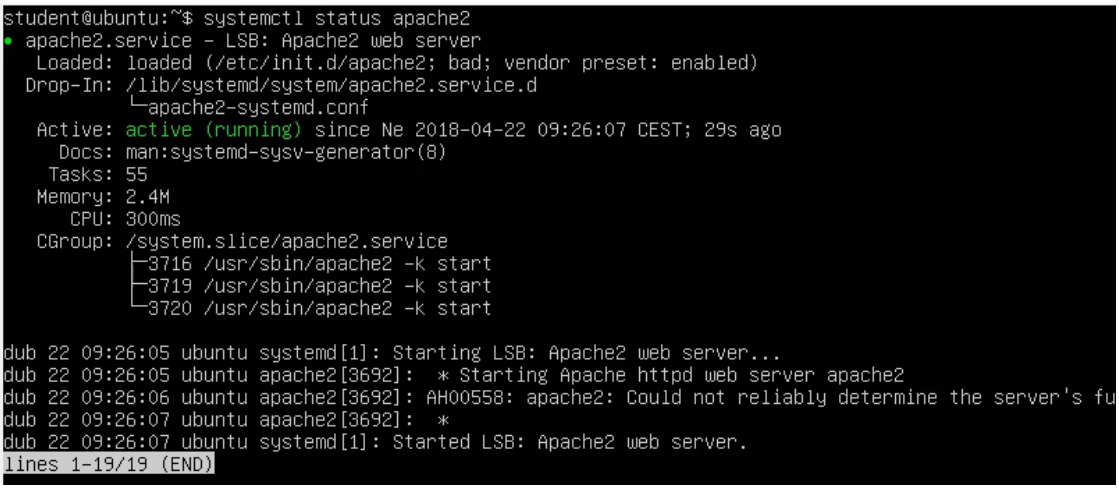
Obr. 5.57: Instalace Logcheck.

5.3 Scénář zabezpečení webového serveru

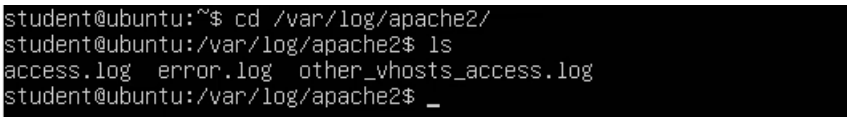
Nejprve provedeme instalaci webového serveru apache. To provedeme příkazem „`sudo apt-get install apache2`“. Status webového serveru lze zobrazit příkazem „`sudo systemctl status apache2`“, viz 5.58.

Webová stránka, který bude zobrazena po navštívení domény je umístěna v „/var/www/html“, dokument se jmenuje „index.html“. Jednotlivé logy jsou uloženy v „/var/log/apache2“, viz Obr.5.59.

Základní zabezpečení provedeme v „/etc/apache2/conf-available/security.conf“. Zde změníme direktivu „`ServerSignature`“ na „`Off`“ a „`ServerTokens`“ na „`Prod`“ (je nutno ji doplnit), viz Obr. 5.60.



Obr. 5.58: Apache status.



Obr. 5.59: Adresář pro logy webového serveru Apache.

Obr. 5.60: Pozměnění konfiguračního souboru Apache.

Nyní se zaměříme na další konfigurační soubor a to „/etc/apache2/apache2.conf“. Zde upravíme direktivu „`<Directory />`“, vytvořit „Alias“ a vytvořit direktivu „`<Directory "/usr/share/apache2/icons">`“, viz Obr. 5.61.

Obr. 5.61: Upravení direktiv v konfiguračním souboru Apache.

Posledním krokem je webový server restartovat příkazem „`sudo systemctl restart apache2`“.

5.3.1 SSL certifikát

Certifikát, který bude použit, bude typu „self-signed“, začneme vygenerováním certifikačního páru příkazem „`sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt`“. Poté je nutné vyplnit potřebné údaje, viz Obr.5.62.

```
student@ubuntu:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/
apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
Generating a 2048 bit RSA private key
.....................+++
.............+++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CZ
State or Province Name (full name) [Some-State]:Czech republic
Locality Name (eg, city) []:Brno
Organization Name (eg, company) [Internet Widgits Pty Ltd]:-
Organizational Unit Name (eg, section) []:-
Common Name (e.g. server FQDN or YOUR name) []:server
Email Address []:root@localhost_
```

Obr. 5.62: SSL, generování certifikačního páru.

Dále vygenerujeme Diffie-Hellman group příkazem „`sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048`“, viz Obr. 5.63.

```
student@ubuntu:~$ sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
```

Obr. 5.63: Generování Diffie-Hellman group.

V dalším kroku nakonfigurujeme webový server k použití SSL, vytvoříme tedy konfigurační soubor příkazem „`sudo nano /etc/apache2/conf-available/ssl-params.conf`“. Do tohoto souboru vložíme text, viz Obr. 5.64.

```
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3
SSLHonorCipherOrder On
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
SSLCompression off
SSLSessionTickets Off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"
```

Obr. 5.64: Nastavení SSL parametrů.

Poté vytvoříme zálohu konfiguračního souboru příkazem „`sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-`

`available/default-ssl.conf.bak`“. Vstoupíme do konfiguračního souboru příkazem „`sudo nano /etc/apache2/sites-available/default-ssl.conf`“. Zde pozměníme hodnotu direktivy „`ServerAdmin`“, kde vložíme e-mailovou adresu (v tomto případě zadána e-mailová superuživatele) a vytvoříme novou direktivu „`ServerName`“, kde v hodnotě bude uvedena IP adresa webového serveru, v tomto případě „10.0.1.2“, viz Obr. 5.65.

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin root@localhost
                ServerName 10.0.1.2
```

Obr. 5.65: Nastavení direktiv v konfiguračním souboru default-ssl.conf.

V tomto dokumentu musíme také upravit cestu k certifikátům. Pozměníme direktivu „`SSLCertificateFile`“ na hodnotu „/etc/ssl/certs/apache-selfsigned.crt“ a direktivu „`SSLCertificateKeyFile`“ změníme hodnotu na „/etc/ssl/private/apache-selfsigned.key“, viz Obr. 5.66.

```
#   SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
```

Obr. 5.66: Úprava souborů s certifikáty.

V tomto konfiguračním souboru musíme také odkomentovat direktivu „`BrowserMatch`“ a všechny její hodnoty do podoby viz Obr. 5.67. Pozn. direktiva je umístěna na konci dokumentu.

```
BrowserMatch "MSIE [2-6]" \
                nokeepalive ssl-unclean-shutdown \
                downgrade-1.0 force-response-1.0
```

Obr. 5.67: BrowserMatch direktiva.

V konfiguračním souboru „/etc/apache2/sites-available/000-default.conf“ vložíme direktivu a hodnotu „`Redirect "/" "https:\\10.0.1.2"`“, což provede automatické přesměrování na HTTPS.

Nyní zadáme příkazy „`sudo a2enmod ssl`“, „`sudo a2enmod headers`“,. „`sudo a2ensite default-ssl`“, „`sudo a2enconf ssl-params`“.

Poté ověříme konfiguraci příkazem „`sudo apache2ctl configtest`“, pokud bude výstupem „Syntax OK“, je vše nakonfigurováno korektně.

Posledním krokem je provést restart Apache serveru, viz Obr. 5.68.

```
student@ubuntu:~$ systemctl restart apache2.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to restart 'apache2.service'.
Authenticating as: student,,, (student)
Password:
==== AUTHENTICATION COMPLETE ===
student@ubuntu:~$ _
```

Obr. 5.68: Restart webového serveru Apache.

6 PENETRAČNÍ TESTOVÁNÍ

Za účelem ověření zabezpečení bylo použito penetrační testování. Pojem penetrační testování označuje snahu proniknout do zabezpečeného systému, a tak ověřit úroveň zabezpečení. Tato snaha je prováděna prostředky, které mohou být využity útočníkem při snaze o proniknutí do systému. Je nutné zmínit, že pokud je prováděno penetrační testování, tak vždy musí být opatřeno souhlasem (v případě prováděného testu na jiný subjekt). Penetrační testování může být kvalifikováno jako trestný čin, a tudíž důrazně není jeho provádění bez souhlasu majitele systému doporučeno. Penetrační testování osobního PC a serveru.

Penetrační testování zabezpečených OS a webového serveru bylo provedeno v bakalářské práci s názvem „Kybernetické útoky na operační systémy“ [51]. Získané údaje jsou tedy čerpány z této bakalářské práce.

Penetrační testování zabezpečeného osobního počítače (IP adresa 10.0.1.3) odhalilo otevřené síťové porty pro služby HTTP (80/TCP), HTTPS (443/TCP), WSDAPI (5357/TCP). Byla odhalena aplikace PRTG Network Monitor, také byl odhalen operační systém Windows.

Dále bylo provedeno testování zabezpečeného webového serveru (IP 10.0.1.2). Testování odhalilo otevřené síťové porty pro služby SSH (22/TCP), HTTP (80/TCP), HTTPS (443/TCP). I u tohoto stroje byl odhalen operační systém, tedy OS Ubuntu. Na tomto stroji je také instalován webový server. Penetrační testování odhalilo, že se jedná o webový server Apache a podalo již známou informaci, že je použit operační systém Ubuntu.

U testovaných strojů nebyla narušena jejich bezpečnost. Nebylo tak možné proniknout do systému a získat citlivé údaje.

6.1 Detekce penetračního testování

Osobní počítač nebyl schopen přímo detekovat provedené penetrační testování. Aplikace PRTG Network Monitor nezachytila žádný pokus o narušení bezpečnosti systému. Účel této aplikace je detekovat vnitřní chování počítače (možné systémové aktualizace, vytíženost systému apod.). Aplikace NetWorx detekovala nepatrný nárůst síťové komunikace, ovšem nebyl natolik velký, aby vyvolal poplach.

Naopak webový server zcela evidentně zachytil probíhající skenování sítě. Pokud se zaměříme na IDS Snort, tak tento systém detekoval skenování (přímo nástroj NMap), také detekoval zdroj skenování (10.0.1.4), viz Obr. 6.1

```
05/22-20:32:10.363077  [**] [1:409:7] ICMP Echo Reply undefined code [**] [Classification: Misc acti
vity] [Priority: 3] {ICMP} 10.0.1.2 -> 10.0.1.4
05/22-20:32:10.388412  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {
ICMP} 10.0.1.4 -> 10.0.1.2
05/22-20:32:10.388452  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority
: 3] {ICMP} 10.0.1.2 -> 10.0.1.4
05/22-20:32:10.544410  [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Le
ak] [Priority: 2] {TCP} 10.0.1.4:39862 -> 10.0.1.2:37601
```

Obr. 6.1: Snort, detekce skenování sítě.

Firewall UFW detekoval velké množství pokusů o vniknutí narušení bezpečnosti systému. Tyto snahy firewall úspěšně zablokoval, viz Obr. 6.2, na kterém můžeme vidět „ufw.log“ umístěný ve „/var/log“ (MAC adresa zkrácena). Totožný výsledek dostaneme, pokud nahlédneme do logovacího souboru „syslog“, nebo „kern.log“.

```
May 22 14:17:26 ubuntu kernel: [  357.404942] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:08:00 SRC=10.0.1.4 DST=10.0.1.2
May 22 14:17:38 ubuntu kernel: [  369.500943] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:08:00 SRC=10.0.1.4 DST=10.0.1.2
May 22 14:17:58 ubuntu kernel: [  389.312087] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:08:00 SRC=10.0.1.4 DST=10.0.1.2
May 22 14:18:18 ubuntu kernel: [  409.276533] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:08:00 SRC=10.0.1.4 DST=10.0.1.2
May 22 14:18:38 ubuntu kernel: [  429.437342] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:08:00 SRC=10.0.1.4 DST=10.0.1.2
May 22 14:18:59 ubuntu kernel: [  449.949261] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:08:00 SRC=10.0.1.4 DST=10.0.1.2
May 22 14:19:18 ubuntu kernel: [  469.501214] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:08:00 SRC=10.0.1.4 DST=10.0.1.2
```

Obr. 6.2: UFW, blokování síťového provozu.

Pokud si zobrazíme soubor „auth.log“, který je také umístěn ve „/var/log“, viz Obr. 6.3, tak vidíme neúspěšný pokus o přihlášení ke službě SSH.

```
May 22 14:17:26 ubuntu sshd[1544]: Connection closed by 10.0.1.4 port 43091 [preauth]
May 22 14:17:38 ubuntu sshd[1547]: Connection closed by 10.0.1.4 port 38221 [preauth]
May 22 14:17:58 ubuntu sshd[1549]: Connection closed by 10.0.1.4 port 42659 [preauth]
May 22 14:18:18 ubuntu sshd[1551]: Connection closed by 10.0.1.4 port 39487 [preauth]
```

Obr. 6.3: Auth.log, záznam pokusu přihlášení k SSH.

Při výpisu logovacího souboru „access.log“ pro Apache server umístěného ve „/var/log/Apache2“, vidíme, že také detekoval skenování síťových portů pomocí nástroje NMap, viz Obr. 6.3.

```
10.0.1.4 - - [22/May/2018:12:15:57 +0200] "POST /sdk HTTP/1.1" 302 538 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine;
10.0.1.4 - - [22/May/2018:12:15:57 +0200] "GET / HTTP/1.0" 200 13193 "-" "-"
10.0.1.4 - - [22/May/2018:12:15:58 +0200] "GET / HTTP/1.1" 200 13174 "-" "-"
10.0.1.4 - - [22/May/2018:12:15:58 +0200] "GET /HNAP1 HTTP/1.1" 404 2056 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine
10.0.1.4 - - [22/May/2018:12:15:58 +0200] "GET / HTTP/1.1" 302 513 "-" "-"
```

Obr. 6.4: Apache, detekování skenování síťových portů.

Informace, které jsou obsaženy v ostatních systémových lozích (mail.log, bootstrap.log, wtmp) nebyly pro detekci penetračního testování relevatní.

7 VYHODNOCENÍ

Zabezpečený osobní počítač byl zabezpečen proti narušení bezpečnosti pomocí bezpečného hesla, byly provedeny aktualizace, dále bylo aplikování šifrování vybrané složky a diskového oddílu. O síťovou bezpečnost se starala aktivní brána Windows Firewall a jako antivirus byl vybrán antivirus Windows Defender. Aplikace, jejichž účel byl zvýšit bezpečnost operačního systému, probíhající skenování systému nedetekovali.

Aplikace NetWorx skenování systému odhalit nemohla. Síťový provoz byl lehce navýšen, ale je naprosto závislý na zvolené míře, kterou uživatel nastaví, po které je vyvolán poplach. Aby k tomuto došlo, bylo by nutné, aby byl přenesen velký objem dat.

Aplikace PRTG Network Monitor také nedetekovala probíhající skenování, je spíše zaměřena na operace probíhající v počítači a na jeho stav. Navíc tato aplikace v penetračním testování napomohla k identifikování operačního systému z důvodu otevřených portů, které tato aplikace využívá.

Ovšem bezpečnost zabezpečeného osobního počítače v penetračním testování nebyla narušena. Lze tak považovat operační systém za bezpečný.

Jako další postup je doporučeno otestovat odlišné aplikace, které budou přímo detekovat probíhající skenování sítě a také dále instalovat aktualizace jak operačního systému, tak všech instalovaných aplikací.

Zabezpečený serverový OS byl zabezpečen šifrovaným domovským adresářem, dále bylo použito šifrování logického oddílu a byly instalovány dostupné aktualizace. Pro vzdálené připojení k serveru bylo použito SSH, které vyžaduje přihlášení pomocí certifikátů. IDS systém Snort detekoval probíhající skenování portů. Ubuntu Firewall UFW detekoval pokusy o přístup, které neumožnil a připojení blokoval. Také jsme vytvořili zálohu celého systému. Díky jednotlivým logům vidíme, že pokus o skenování portů, resp. narušení bezpečnosti proběhl. Identifikovali jsme zdroj, jako další postup by bylo vhodné tuto adresy uvést do firewallu a plně ji blokovat a dále instalovat aktualizace systému a jednotlivých aplikací. Také by bylo vhodné nástroj IDS Snort umístit na samostatné zařízení, které na kterém by byl nástroj Snort v režimu IPS a sloužil jako „pračka provozu“. Bezpečnost serverového OS se nepodařilo narušit a lze tak považovat operační systém za bezpečný.

Webový server Apache byl zabezpečen pomocí SSL certifikátu, také byl omezen přístup, který je umožněn návštěvníkovi, a také byly omezeny informace, které jsou návštěvníkovi poskytovány prostřednictvím chybových hlášek. V průběhu penetračního testování se podařilo získat název instalovaného webového serveru, tedy Apache a také instalovaný operační systém, tedy Ubuntu. Ovšem získání instalovaných verzí již nebylo úspěšné. Webový server také ve svém logu identifikoval skenování portů. Jako další postup je vhodné na tento server instalovat programovací jazyk např. PHP, aby poskytované webové stránky byly dynamické, a také jej zabezpečit. Popřípadě umístit na webové stránky aplikaci, která bude také adekvátně zabezpečena. Dále je doporučeno pravidelně instalovat aktualizace. V průběhu penetračního testování se nezdařilo získat velké množství informací, které by přesněji identifikovali operační systém, nebo instalovaný webový server, lze tak považovat webový server za bezpečný.

8 ZÁVĚR

V této bakalářské práci zaměřené na bezpečnost operačních systémů jsme nejprve věnovali pozornost jednotlivým prvkům bezpečnosti. Dále jsme provedli bezpečnostní analýzu nejpoužívanějších operačních systémů a dále vytvořili bezpečnostní scénáře. První scénář byl zaměřen na zabezpečení osobního počítače. V tomto scénáři jsme si uvedli, co je vhodné používat, aby byl počítač bezpečný. Další bezpečnostní scénář se zaměřil na vytvoření bezpečného serverového operačního systému, kde jsme uvedli vhodné možnosti, které napomáhají ke zvýšení bezpečnosti systému. Poslední scénář se zaměřoval na webový server, jako takový. Uvedli jsme, jak server nastavit, tak aby poskytoval minimum informací o instalovaných verzích, ale přesto byl funkční.

V další části jsme jednotlivé scénáře aplikovali na virtualizované stroje a k otestování bezpečnosti využili penetračního testování. Penetrační testování zabezpečených strojů bylo provedeno v bakalářské práci „Kybernetické útoky na operační systémy“. Po skončení penetračního testování jsme prohlédli jednotlivé logy a zjistili jsme, že zabezpečený osobní počítač penetrační testování nedetekoval, ale naopak vybraná aplikace prozradila použitý operační systém. V případě webového serveru jsme penetrační testování zachytili, instalovaný webový server (Apache) průběh penetračního testování detekoval také.

Aby byl operační systém bezpečný, je nutné na něj pohlížet, jako na jeden celek. Ponechávat v počítači jen aktualizované a používané aplikace. Každá nepoužívaná nebo neaktualizovaná aplikace může představovat slabinu v zabezpečení a může způsobit kompromitaci dat.

LITERATURA

[1] Přednášky z předmětu Bezpečnost ICT2, TIC2, 2017/2018 VUT FEKT UTKO. Garant předmětu prof. Ing. Jiří Mišurec, CSc.

[2] BURDA, Karel. Bezpečnost informačních systémů [elektronicky]. VUT FEKT UTKO, 2013 [cit. 2017-11-27]. ISBN 978-80-214-4890-2.

[3] Autentizace a autorizace. Www.trisul.cz [online]. [cit. 2017-11-27]. Dostupné z: http://www.trisul.cz/bezpecnost-autentizace-autorizace/

[4] Autentizace a autorizace. Www.jaknainternet.cz [online]. [cit. 2017-11-27]. Dostupné z: https://www.jaknainternet.cz/page/1178/pocitacova-hesla/

[5] PETROWSKI, Thorsten. Bezpečí na internetu: pro všechny. Liberec: Dialog, 2014. Tajemství (Dialog). ISBN 978-80-7424-066-9.

[6] Forget about passwords: You need a passphrase!. Www.welivesecurity.com [online]. 2016 [cit. 2017-11-28]. Dostupné z: https://www.welivesecurity.com/2016/05/05/forget-about-passwords-you-need-a-passphrase/

[7] What Is a Passphrase in Computer Networking? Www.lifewire.com [online]. 2017 [cit. 2017-11-28]. Dostupné z: https://www.lifewire.com/what-is-passphrase-818353

[8] Autentizace a autorizace. Www.cleverity.cz [online]. [cit. 2017-11-27]. Dostupné z: https://www.cleverity.cz/hesla-nejsou-bezpecna-dvoufaktorova-autentizace/

[9] Antivir [online]. 2012 [cit. 2017-11-27]. Dostupné z: http://www.guard-dynamics.cz/antivir

[10] What is a Computer Firewall? And It's Functions [online]. 2016 [cit. 2017-11-27]. Dostupné z: https://antivirusinsider.com/computer-firewall-functions/

[11] DOSTÁLEK, Libor a Marta VOHNOUTOVÁ. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. Brno: Computer Press, 2006. ISBN 80-251-0828-7.

[12] HUSEBY, Sverre H. Zranitelný kód. Brno: Computer Press, 2006. ISBN 80-251-1180-6.

[13] Bezpečnostní funkce v počítačových sítích [online]. 2011 [cit. 2017-11-27]. Dostupné z: http://ics.muni.cz/bulletin/articles/171.html

[14] Elektronický podpis pro začátečníky. Www.digipodpis.cz [online]. 2014 [cit. 2017-11-27]. Dostupné z: http://www.digipodpis.cz/zaciname.php

[15] Elektronický podpis. Www.ica.cz [online]. [cit. 2017-11-27]. Dostupné z: http://www.ica.cz/Elektronicky-podpis

[16] PUŽMANOVÁ, Rita. Bezpečnost bezdrátové komunikace: jak zabezpečit Wi-Fi, Bluetooth, GPRS či 3G. Brno: CP Books, 2005. ISBN 80-251-0791-4.

[17] Transport Layer Security (TLS). Hpbn.co [online]. 2013 [cit. 2017-11-27]. Dostupné z: https://hpbn.co/transport-layer-security-tls/

[18] VPN pro začátečníky: princip fungování, výhody a nevýhody. Www.root.cz [online]. [cit. 2017-11-27]. Dostupné z:

https://www.root.cz/clanky/vpn-pro-zacatecniky-princip-fungovani-vyhody-a-nevyhody/
[19] HOW DOES A VPN WORK? Www.ign.com [online]. 2017 [cit. 2017-11-27]. Dostupné z: http://www.ign.com/articles/2017/08/11/how-does-a-vpn-work
[20] Encryption and Security Protocols in a VPN. Computer.howstuffworks.com [online]. [cit. 2017-11-27]. Dostupné z: https://computer.howstuffworks.com/vpn7.htm
[21] VPN Encryption: The Complete Guide. Www.bestvpn.com [online]. 2017 [cit. 2017-11-27]. Dostupné z: https://www.bestvpn.com/vpn-encryption-the-complete-guide/
[22] SSH – bezpečné používání vzdáleného počítače a kopírování dat. Www.dsl.cz [online]. [cit. 2017-11-27]. Dostupné z: http://www.dsl.cz/jak-na-to/jak-na-ssh
[23] Používání klíčů v OpenSSH. Www.root.cz [online]. [cit. 2017-11-27]. Dostupné z: https://www.root.cz/clanky/pouzivani-klicu-v-openssh/
[24] Jak se přihlašovat na SSH bez zadávání hesla. Www.root.cz [online]. [cit. 2017-11-27]. Dostupné z: https://www.root.cz/clanky/jak-se-prihlasovat-na-ssh-bez-zadavani-hesla/
[25] Přednášky z předmětu Bezpečnost ICT1, TIC1, 2016/2017 VUT FEKT UTKO. Garant předmětu doc. Ing. Jan Hajný, Ph.D.
[26] Windows support for hard disks that are larger than 2 TB. Www.microsoft.com [online]. 2017 [cit. 2017-11-27]. Dostupné z: https://support.microsoft.com/en-us/help/2581408/windows-support-for-hard-disks-that-are-larger-than-2-tb
[27] Hesla z Windows XP lze prý prolomit za 5,3 sekundy. Www.cnews.cz [online]. 2010 [cit. 2017-11-27]. Dostupné z: https://www.cnews.cz/hesla-z-windows-xp-lze-pry-prolomit-za-53-sekundy/
[28] Procházka historií Microsoft Windows. Http://pcworld.cz [online]. 2010 [cit. 2017-11-27]. Dostupné z: http://pcworld.cz/software/prochazka-historii-microsoft-windows-1-dil-16395
[29] Hash. Www.pocet-znaku.cz [online]. [cit. 2017-11-27]. Dostupné z: http://www.pocet-znaku.cz/hash
[30] Avast Premier. Www.avast.com [online]. [cit. 2017-12-06]. Dostupné z: https://www.avast.com/cs-cz/premier
[31] NetWorx: Bandwidth monitoring and data usage reports for Windows, macOS and Linux [online]. 2018 [cit. 2018-03-06]. Dostupné z: https://www.softperfect.com/products/networx/
[32] Paessler: PRTG monitors every aspect of your IT infrastructure [online]. 2018 [cit. 2018-03-06]. Dostupné z: https://www.paessler.com/prtg
[33] How to use a command line random password generator PWGEN on Linux. Linuxconfig.org [online]. 2015 [cit. 2018-03-09]. Dostupné z:

https://linuxconfig.org/how-to-use-a-command-line-random-password-generator-pwgen-on-linux
[34] Wiki.ubuntu.cz [online]. 2012 [cit. 2018-03-25]. Dostupné z: http://wiki.ubuntu.cz/bezpe%C4%8Dnost/%C5%A1ifrovan%C3%BD_domovsk%C3%BD_adres%C3%A1%C5%99
[35] Aptitude safe-upgrade equivalence with apt-get. Askubuntu.com [online]. 2012 [cit. 2018-03-09]. Dostupné z: https://askubuntu.com/questions/117088/aptitude-safe-upgrade-equivalence-with-apt-get
[36] SSH. Wiki.ubuntu.cz [online]. 2012 [cit. 2018-03-25]. Dostupné z: http://wiki.ubuntu.cz/ssh
[37] Bezpečné přihlašování na server pomocí SSH klíče. Coolhousing.net [online]. 2016 [cit. 2018-03-25]. Dostupné z: https://www.coolhousing.net/cz/bezpecne-prihlasovani-na-server-pomoci-ssh-klice
[38] OpenSSH Server. Help.ubuntu.com [online]. [cit. 2018-03-25]. Dostupné z: https://help.ubuntu.com/lts/serverguide/openssh-server.html
[39] SSH klíče ve Windows - PuTTY. Blog.frantovo.cz [online]. 2014 [cit. 2018-03-25]. Dostupné z: https://blog.frantovo.cz/c/10/SSH%20kl%C3%AD%C4%8De%20ve%20Windows%20-%20PuTTY
[40] Snort [online]. 2017 [cit. 2018-03-21]. Dostupné z: https://www.snort.org/
[41] Iptables. Wiki.ubuntu.cz [online]. 2015 [cit. 2018-03-21]. Dostupné z: http://wiki.ubuntu.cz/bezpe%C4%8Dnost/firewall/iptables
[42] How To Setup a Firewall with UFW on an Ubuntu and Debian Cloud Server. Digitalocean.com [online]. 2013 [cit. 2018-05-05]. Dostupné z: https://www.digitalocean.com/community/tutorials/how-to-setup-a-firewall-with-ufw-on-an-ubuntu-and-debian-cloud-server
[43] Záloha a obnova systému. Wiki.ubuntu.cz [online]. 2012 [cit. 2018-03-26]. Dostupné z: http://wiki.ubuntu.cz/z%C3%A1loha_a_obnova_syst%C3%A9mu
[44] Linuxexpres.cz [online]. 2011 [cit. 2018-04-02]. Dostupné z: https://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-sledovani-logu-pomoci-logcheck
[45] Webový server. Htmlguru.cz [online]. [cit. 2018-04-02]. Dostupné z: http://htmlguru.cz/vystaveni-webovy-server.html
[46] How To Install the Apache Web Server on Ubuntu. Digitalocean.com [online]. 2017 [cit. 2018-04-02]. Dostupné z: https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16-04
[47] Zabezpečení serveru Apache a PHP. Security-portal.cz [online]. 2004 [cit. 2018-04-02]. Dostupné z: http://www.security-portal.cz/clanky/zabezpe%C4%8Den%C3%AD-serveru-apache-php

[48] Hide Apache ServerSignature / ServerTokens / PHP X-Powered-By. If-not-true-then-false.com [online]. 2016 [cit. 2018-04-02]. Dostupné z: https://www.if-not-true-then-false.com/2009/howto-hide-and-modify-apache-server-information-serversignature-and-servertokens-and-hide-php-version-x-powered-by/

[49] 10 things you should do to secure Apache. Techrepublic.com [online]. 2009 [cit. 2018-04-02]. Dostupné z: https://www.techrepublic.com/blog/10-things/10-things-you-should-do-to-secure-apache/

[50] How To Create a Self-Signed SSL Certificate for Apache in Ubuntu 16.04. Digitalocean.com [online]. 2016 [cit. 2018-04-09]. Dostupné z: https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-16-04

[51] HOLASOVÁ, E. Kybernetické útoky na operační systémy. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 87 s. Vedoucí bakalářské práce Ing. Vlastimil Člupek, Ph.D

Veškeré diagramy v této bakalářské práci byly vytvořeny pomocí webového nástroje Draw. Dostupné z: https://www.draw.io/

SEZNAM ZKRATEK

AES	Advanced Encryption Standard
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EAP-TLS	Extensible Authentication Protocol-Transport Layer Security
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EFS	Encrypting File System
EICAR	European Institute for Computer Antivirus Research
exFAT	Extended FAT (File Allocation Table)
FAT	File Allocation Table
FAT32	32 bitový FAT (File Allocation Table)
FEK	File Encryption Key
FTP	File Transfer Protocol
GNU/GPL	GNU General Public License
GRUB	Grand Unified Bootloader
GUI	Graphical User Interface
HFS+	Hierarchical File System
HMAC	Hash-based Message Authentication Code
HP	Handshake protocol
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP (Hypertext Transfer Protocol) Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IPS	Intrusion Prevention System
L2TP/IPSec	Layer 2 Tunneling Protocol/ Internet Protocol Security
LM Hash	LanMan hash
LRW	Liskov, Rivest, Wagner
MAC	Media Access Control
MBR	Master Boot Record
MD4	Message Digest 4 algoritmus
MD5	Message Digest 5 algoritmus
NAT	Network Address Translation
NGFW	Next Generation Firewall
NMap	Network Mapper
NTFS	New Technology File System
OpenSource	Otevřený software, zdrojový kód je volně přístupný
OpenSSL	OpenSource SSL (Secure Sockets Layer)
OpenVPN	OpenSource virtual private network

OS	Operační systém
PHP	Hypertext Preprocessor
POP3	Post Office Protocol,verze 3
PPTP	Point to Point Tunneling Protocol
RLP	Record Layer Protocol
RSA	Rivest, Shamir, & Adleman
SELinux	Security Enhanced Linux
SHA	Secure Hash Algorithm
SSH	Secure Shell
SSL	Secure Sockets Layer
SWAP	Odkládací prostor na disku, virtuální paměť
TCP	Transmission Control Protocol
Telnet	Network Virtual Terminal Protocol
TLS	Transport Layer Security
UFW	Uncomplicated Firewall
UGO	User, Group, Others
URL	Uniform Resource Locator
VPN	Virtual Private Network
Wi-Fi	Označení bezdrátové technologie, vychází z IEEE 802.11
XTS	XEX-based tweaked-codebook mode with ciphertext stealing
3DES	Triple DES (Data Encryption Standard)

SEZNAM PŘÍLOH

A. OBSAH PŘILOŽENÉHO CD

Na přiloženém CD je uložena elektronická verze této bakalářské práce ve formátu PDF.