Sem vložte zadání Vaší práce.

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Diplomová práce

Zabezpečený lock-down systém

Bc. Ondřej Hájek

Vedoucí práce: Ing. Ivan Šimeček, Ph.D.

1. května 2015

Poděkování

Chtěl by poděkovat svému vedoucímu práce Ing. Ivanu Šimečkovi, Ph.D. za podnětné připomínky a pomoc při tvorbě této práci. Dále také děkuji své rodině a přátelům nejen za podporu při psaní této práce, ale i za podporu po celou dobu mého studia.

Prohlášení

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

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

V Praze dne 1. května 2015 .....

Odkaz na tuto práci

Hájek, Ondřej. *Zabezpečený lock-down systém.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato práce se zabývá implementací zabezpečeného lock-down systému umožňující zobrazení pouze prohlížeče s definovanými povolenými doménami využívající běžnou Linuxovou distribuci. První část obsahuje analýzu existujících řešení. Druhá část se zabývá analýzou vhodné Linuxové distribuce a výběrem nástrojů pro zabezpečení systému. Třetí část popisuje realizaci systému včetně automatických skriptů pro nastavení prostředí. Čtvrtá část obsahuje testování prostředí a skriptů. Poslední část srovnává vlastní řešení s již existujícími.

Klíčová slova

kiosk, lock-down, Ubuntu, LTSP, Chromium, AuFS, zabezpečený systém

Abstract

This thesis deals with implementation of secure lock-down system allowing user to display only browser with defined authorized domains using a regular Linux distribution. The first part contains an analysis of existing solutions. The second part analyzes suitable Linux distributions and selection of tools for system security. The third part describes the implementation of the system, including automated scripts for setting up the environment. The fourth

part contains testing of the environment and automated scripts. The last part compares this solution with existing ones.

Keywords

kiosk, lock-down, Ubuntu, LTSP, Chromium, AuFS, secured system

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Motivace

Velké množství firem a institucí potřebuje pro své potřeby nějaký terminál, který bude veřejně dostupný pro každého, kdo do dané společnosti zavítá. Jedním takovým příkladem je veřejná knihovna, která má ze zákona[1] povinnost umožnit každému přístup k informacím na Internetu. Také některé firmy potřebují podobný systém, ale s omezením na prohlížení pouze určité skupiny stránek (určených například pro prezentaci instituce veřejnosti). Z důvodu přístupu široké veřejnosti se mohou takové stroje stát potencionálním nebezpečím umožňující zneužití, ať už se jedná o úmyslný útok nebo neznalého či zvědavého uživatele. V případě úspěšného zneužití systému je pak velmi těžké, až zcela nemožné, zpětně dohledat případného útočníka. Takový systém je tedy nutno co nejlépe zabezpečit, aby byl odolný proti jakýmkoliv pokusům narušit jeho jedinou funkci, a to zobrazovat (určené) webové stránky.

V posledních letech se ukázalo, že na trhu je po takových řešeních poptávka a vzniklo několik upravených operačních systémů připravených přímo pro tyto situace. Možností je využití právě takových systémů, ale problémem je většinou velmi složitá rozšířitelnost o funkce, které nejsou implicitní součástí systému. Druhým problémem může být finanční otázka takového řešení, kdy se cena za licenci pro jeden počítač může vyšplhat až k několika tisícům korun. Pokud je systém postaven na některém komerčním operačním systému (například Microsoft Windows), je nutné navíc vlastnit licenci k provozování takového systému pro každý počítač.

Druhou možností je vytvoření takového systému svépomocí s využitím některého zdarma dostupného operačního systémem. Nejjednodušší variantou je postavení takového systému nad Linuxovou distribucí, díky čemuž bude mít administrátor plnou kontrolu nad instalovaným softwarem a může libovolně distribuci upravovat. Nevýhodou je nutnost znalostí funkcí takového systému a způsob upravení pro vytvoření zabezpečeného lock-down systému.

Právě tuto nutnou znalost fungování systému bych chtěl nahradit automatickým skriptem, který vybranou Linuxovou distribuci přenastaví pro použití jako lock-down systém. Pokud administrátor bude chtít provést některé změny v systému, nebude problém upravit samotný systém po dokončení instalace nebo přímo upravit skript před instalací. Pro jednoduší orientaci v tomto skriptu budou sloužit komentáře přímo v kódu pro pochopení funkce některých kroků.

Pokud v nějaké instituci bude třeba takových veřejných počítačů s prohlížečem větší množství, je velmi neefektivní vytváření a nastavování separátních operačních systému. V takové rozsáhlejší síti by byla i jednoduchá aktualizace částí systému administrativně velmi náročná. Jako součást této práce bych chtěl tedy také vytvořit skript pro vytvoření serveru, ke kterému by se všechny veřejné počítače vzdáleně připojovaly, čímž by fungovaly v režimu tenkých klientů a odpadla by nutnost individuální administrace každého stroje.

Struktura práce

Před samotným návrhem vlastní realizace zabezpečeného prostředí nejdříve představím existující Linuxové implementace takových systémů a jejich nalezené zranitelnosti s možným postupem využití těchto zranitelností.

V další části se budu zabývat výběrem vhodné Linuxové distribuce, která bude sloužit jako základ systému. Zde také provedu analýzu možných nástrojů a programů pro vytvoření zabezpečeného prostředí jak pro samostatného klienta, tak pro server s tenkými klienty.

Stěžejní částí bude sekce popisující implementaci systému. V této sekci bude popsáno nastavení systému a programů použitých k vytvoření zabezpečeného prostředí. Zároveň bude z nastavení vytvořen automatický skript, který všechna popisovaná nastavení systému a programů provede bez nutnosti ručních úprav souborů.

V následující sekci proběhne testování funkčnosti a zabezpečení jak samotného zabezpečeného systému, tak i skriptu pro automatické nastavení prostředí.

Poslední sekce bude obsahovat krátké srovnání mého řešení s existujícími implementacemi orientované zejména na nalezené problémy existujících řešení.

Základní pojmy

AuFS

Advanced multi layered unification filesystem (AuFS) je filesystém umožňující sjednotit několik různých adresářů do jediného umístění. Tyto adresáře mohou

používat různé filesystémy a lze k nim definovat i různá oprávnění (například používat určité adresáře pouze ke čtení a zapisovat data do jiného).

API

Application Programming Interface (API) je sbírka procedur, funkcí a tříd knihovny nebo programu, které může programátor při programování využívat. API definuje funkcionalitu, která je nezávislá na vlastní implementací, což umožňuje změnu definice a implementaci, aniž by byla ohrožena daná funkcionalita.

CLI

Command Line Interface (CLI) je uživatelské rozhraní realizované pomocí příkazové řádky, ve kterém uživatel s programy nebo operačním systémem komunikuje zapisováním příkazů do příkazového řádku. Na rozdíl od textového a grafického uživatelského rozhraní není využívána myš ani menu.

DHCP

Dynamic Host Configuration Protocol (DHCP) server se používá pro automatickou konfiguraci počítačů připojených do počítačové sítě. Server dočasně přiděluje počítačům pomocí DHCP protokolu IP adresu, masku sítě, implicitní bránu a adresu DNS serveru.

Display manager

Display manager se používá k přihlášení uživatele z lokálního počítače nebo přes počítačovou síť a slouží jako bezpečnostní prvek systému k zamezení neoprávněnému přístupu do systému. V závislosti na použitém softwaru se mohou jednotlivé vlastnosti lišit, může se jednat o schopnost přizpůsobení grafického motivu, automatické přihlášení, výběr z více grafických prostředí atp.

DNS

Domain Name System (DNS) je hierarchický systém doménových jmen, který pomocí DNS serverů a DNS protokolu vzájemně převádí IP adresy uzlů v síti a doménová jména. Jeho další funkcí může být i přenos ostatních informací jako jsou záznamy elektronické pošty.

GRUB

GRand Unified Bootloader (GRUB) je zavaděč systému umožňující uživateli mít několik různých operačních systémů na jednom počítači současně s možností výběru, který systém spustit při startu počítače. Kromě různých operač-

ních systémů lze načíst i jeden systém s různými jádry nebo předat zaváděcí parametry příslušnému jádru.

ICMP

Internet Control Message Protocol (ICMP) je protokol používající se v síti pro odesílání chybových zpráv, například pro oznámení, že požadovaná služba nebo počítač není dostupný nebo dosažitelný. Nejznámějším použitím protokolu je odesílání zprávy *echo request* pomocí nástroje *ping* pro zjištění dostupnosti a doby odezvy počítače v síti.

TUI

Text User Interface (TUI) je textové uživatelské rozhraní na pomezí příkazové řádky (CLI) a grafického uživatelského rozhraní (GUI). Pracuje v textovém režimu, kdy je obrazovka rozdělena na rastr a pomocí speciálních znaků jsou sestaveny ovládací prvky, jako jsou okna, tlačítka a další klasické prvky grafického rozhraní.

NAT

Network Address Translation (NAT) je způsob úpravy síťového provozu přes router přepisem výchozí nebo cílové IP adresy, který se většinou používá pro přístup více počítačů z lokální sítě na Internet pod jedinou veřejnou adresou.

NBD

Network block device (NBD) je speciální soubor[1], jehož obsah je poskytován vzdáleným serverem. Většinou je NBD využíváno k přístupu k paměťovému zařízení, které není fyzicky umístěné v místním počítači.

NFS

Network File System (NFS) je internetový protokol pro vzdálený přístup k souborům přes počítačovou síť. Používá se hlavně k připojení disku pomocí NFS klienta ze vzdáleného serveru a umožnění lokální práce.

Phishing

Phishing je podvodná technika používaná k získávání citlivých údajů v elektronické komunikaci. Ve většině případů se jedná o pokus vyvolání dojmu oficiální žádosti různých institucí s výzvou adresáta k zadání jeho údajů na odkazovanou stránku. Odkazovaná stránka vzhledem připomíná nebo kopíruje

[1] Speciální soubor je rozhraní k ovladači zařízení, který vypadá jako normální soubor ve filesystému.

oficiální podobu stránky, ale samotná stránka je uložena na serveru útočníka. Uživatel při zadání svých údajů na této stránce prozradí dané informace útočníkovi a ten s jejich pomocí získává přístup k účtu uživatele na oficiální stránce instituce.

Politiky

Politika je systém zásad určených k pomoci při rozhodování a dosažení stanovených cílů. V rámci této práce bude na politiky nahlíženo jako na soubor definovaných neporušitelných zásad, kterými je cílový subjekt vázán.

RPM

RPM Package Manager (RPM) je balíčkovací systém vyvinutý firmou Red Hat ovládaný z příkazové řádky schopný instalovat, odinstalovat, kontrolovat, zjišťovat stav a aktualizovat softwarové balíky.

TFTP

Trivial File Transfer Protocol (TFTP) je jednoduchý protokol pro přenos souborů určený pro přenos souborů v případech, kdy je běžný protokol FTP nevhodný pro svou komplikovanost. Nejčastěji je využíván pro bootování bezdiskových počítačů ze sítě, kdy se celý přenosový protokol musí vejít do omezeného množství paměti.

Kapitola 1

Existující Linuxové implementace

1.1 Porteus Kiosk

Porteus Kiosk[2] je úzce zaměřený systém vycházející z přenositelného systému Porteus (který je založen na Linuxové distribuci Slackware). Jedná se o minimalistický systém momentálně v aktivním vývoji využívající modifikované Linux Live Scripts běžící z RAM paměti počítače.

Aktuální verze 3.2.0, vydaná v prosinci 2014, jako hlavní součást systému využívá prohlížeč Mozilla Firefox 31.0. Chování prohlížeče je ovlivněno vlastními skripty, jedná se zejména o odstranění některých ovládacích prvků prohlížeče a blokování určitých domén či přímo jednotlivých stránek. Dále jsou zakázané některé klávesové zkratky pro znemožnění otevření dialogů s nastavením.

Při prvním spuštění systému je zobrazen průvodce s možností ovlivnění některých nastavení (více informací o průvodci viz. [3]). Mezi tyto funkce patří URL filter, Wake-on-lan, firewall, nastavení obrazovky a rozložení klávesnice, automatické vypnutí, SSH nebo VNC server. Po provedení nastavení je zde možnost uložení upraveného systému na disk či přímo spuštění systému.

1.1.1 Zabezpečení

Systém je celý načítán při spuštění do paměti a disk dále není používán, jakékoliv změny v systému nejsou tedy mezi restarty počítače zachovány a nelze provézt trvalé změny. Dále jsou v systému k dispozici pouze nutné nástroje pro běh systému, čímž jsou omezeny možnosti útoků.

1.1.2 Rozšířitelnost

Systém je dle [4] rozšiřitelný před samotnou instalací pomocí xzm doplňků. Jedná se o zkomprimovaný soubor pomocí nástroje Squashfs, ve kterém se nachází požadované soubory ve stejné struktuře, jako mají být ve finálním systému. Při požadavku na instalaci některého nástroje je kromě zjištění, jaké adresáře a kde jsou vytvářeny při instalace, nutné také ruční vyřešení všech případných závislostí na dalších balících.

Druhou možností rozšiřitelnosti je využití placené služby vývojářů pro uzpůsobení instalace podle požadavků.

1.1.3 Problémy implementace

Systém je zbaven všech nástrojů kromě naprosto nutných nástrojů pro běh systému. Toto řešení má velkou výhodu ve ztížení napadnutí systému, na druhou stranu ale také výrazně snižuje možnosti rozšiřitelnosti nebo úprav nad rámec základního nastavení určeného vývojáři. Systém sice podporuje rozšíření pomocí xzm doplňků, ale zvolené řešení není zcela intuitivní. Pokud by systém v instalaci zahrnoval některý balíčkovací systém nebo přímo možnost pracovat s výslednou strukturou, takový systém by se stal z pohledu administrátora mnohem robustnější.

Jako implicitní prohlížeč využívá systém Mozillu Firefox a jeho zabezpečení je realizováno kombinací explicitním zakázáním určitých klávesových zkratek, změnou vzhledu prohlížeče pomocí souboru `userChrome.css` a blokováním domén pomocí souboru `userContent.css`. Způsob fungování těchto dvou souborů je popsán v sekci 2.2.4.

1.1.3.1 Zabezpečení prohlížeče

Systémovým zakázáním pravého tlačítka myši a všech známých klávesových zkratek pro ovládání prohlížeče společně s upraveným rozhraním je docíleno nemožnosti otevření interních dialogů, ale zároveň je i snížen komfort uživatele. Pokud je například uživatel navyklý používat pravé tlačítko a dialog pro kopírování textu, zde tato metoda nebude fungovat.

Dalším problémem může být, zda jsou v dané implementace pokryty opravdu všechny klávesové zkratky, tímto řešením je vždy možné na některou zapomenout. Toto ale není problém dané implementace, ale spíše problém prohlížeče, kde se daný problém nedá řešit jiným způsobem (kromě vytvoření vlastního JavaScript rozšíření).

1.1.3.2 Zabezpečení filesystému

Porteus dále neobsahuje žádné ochrany proti změnám systému v rámci jednoho spuštění systému, pouze při ukončení jedné relace (vypnutí prohlížeče) je promazána domovská složka. Ostatní adresáře nejsou nijak chráněné.

Zneužití zranitelnosti

Po úniku z prohlížeče do příkazové řádky byl pomocí dostupných nástrojů v systému vytvořen jednoduchý skript, který hledá zapisovatelné složky mimo domovský adresář a dočasný adresář `/tmp`. Tento skript nalezl celkem 2 další zapisovatelné složky:

- `/usr/lib/firefox/browser`
- `/usr/lib/firefox/webapprt`

V těchto složkách může uživatel, pod kterým je spuštěn webový prohlížeč, vytvářet jakékoliv soubory a složky. Pomocí příkazu `cat /dev/zero > /usr/lib/firefox/browser/huge.file` lze následné zcela zaplnit dostupný prostor a do dalšího restartu počítače systém nereaguje na žádné podněty v grafickém prostředí.

Druhou možností je smazat dané složky, čímž je porušen kiosk systém – webový prohlížeč se vůbec nespustí a systém se stává z pohledu uživatele nefunkční až do dalšího restartu počítače.

1.1.3.3 Zabezpečení URL filtru

Kvůli způsobu fungování URL filtrů není bezpečnost nikdy zaručena. URL filtr neblokuje samotný obsah, ale pouze místo daného obsahu zobrazuje přednastavený obrázek. Uživatel tedy sice nevidí žádný obsah, ale na pozadí je tento obsah zpracováván včetně provádění různých funkcí.

Zneužití zranitelnosti URL filtru

Jednoduchým příkladem, jak narušit funkčnost i při blokování všech domén, je přístup na stránku *http://www.newdream.net/crash/js2.html*. Obsahem této stránky je jednoduchý JavaScript kód, který rozdělí okno na polovinu a v každé polovině spustí instanci stejného kódu. Tímto systémem vznikají do nekonečna uvnitř webové stránky kopie té samé stránky až do chvíle, kdy počítači dojde RAM paměť a prohlížeč musí být nějakým způsobem vynuceně restartován, jinak nereaguje na žádné příkazy z grafického prostředí.

1.2 Webconverger

Systém Webconverger[5] je částečně open-source Linuxová distribuce založená na distribuci Debian GNU/Linux využívající systém Git pro vývoj a automatickou aktualizaci systému. Nenakonfigurovaná edice je k používání uvolněna zdarma, při jakémkoliv uzpůsobení je nutné uhradit jednorázový nebo pravidelný příspěvek (*subscription*) podle platného ceníku vývojářů. V současné

době je cena podle ceníku [6] stanovena na $100 ročně nebo $200 jednorázově za jeden počítač používající tento systém.

Aktuální verze 27.1 vydaná v prosinci 2014 obsahuje jak možnost běžet z vyměnitelného média, tak i instalaci na pevný disk. Při prvním spuštění je k možnosti uzpůsobení systému nutná registrace na stránkách vývojářů systému, která je identifikována unikátním klíčem vypočteným z informací o hardwaru daného počítače. Po registraci je možné uzpůsobení systému pomocí interní API zadávané jako příkazy při bootování systému. Pomocí této API je možné nastavit domovskou složku, spořič obrazovky, nastavení tisku, blokování adres nebo automatické vypnutí (viz. dokumentace [7]).

1.2.1 Zabezpečení

Použitý prohlížeč Mozilla Firefox je pomocí vlastního rozšíření Webconverger zbaven všech funkcí spojených s možností nastavení prohlížeče včetně klávesových zkratek. Některé nabídky byly také nahrazeny vlastními (například nabídka při stisknutí pravého tlačítka myši).

Systém je načítán z média pouze pro čtení se zapisovatelnou částí do RAM paměti počítače. Veškeré změny provedené v systému jsou při restartu ztraceny. Samotný filesystém je stahován ze vzdáleného Git repozitáře umístěného na adrese `https://github.com/Webconverger/webc`, tímto způsobem je řešena tedy i automatická aktualizace systému.

1.2.2 Rozšiřitelnost

Kromě využití funkcí API systému nelze systém dále rozšířit. Jedinou možností zůstává úprava samotného kódu systému zduplikováním Git repozitáře do jiného umístění, provedení změn a následně upravení cesty bootovacího procesu do nového umístění.

1.2.3 Problémy implementace

Webconverger obsahuje propracovanou sadu skriptů ovládající celý systém a starající se o jeho zabezpečení. Skripty přijímají příkazy pomocí načtení parametrů při bootování a stažení uloženého nastavení ze stránek vývojářů, které lze po zaplacení libovolně editovat a na dálku ovlivňovat chování systému. Největší nevýhodou systému zůstává jeho cena, $100 ročně za jeden počítač už může být pro některé organizace velký problém.

1.2.3.1 Zabezpečení filesystému

Filesystém je pomocí AuFS připojen pouze pro čtení se zapisovatelnou částí v RAM paměti. Při ukončení relace prohlížeče jsou smazány některé složky v domovském adresáři, ostatní obsah zůstává neměněn a nekontrolován.

Zneužití zranitelnosti

Stejně jako u podobné zranitelnosti Porteus kiosku lze v domovské složce vytvořit obrovský soubor pomocí příkazu `cat /dev/zero > $HOME/huge.file` a zaplnit celý dostupný prostor. Protože zapisovatelná část je efektivně uložena v RAM paměti, dojde k téměř plnému obsazení paměti. Systém je chráněn proti úplnému zaplnění paměti pomocí implicitního limitu filesystému dle [8] na maximálně 50% kapacity RAM. Toto omezení se dá obejít zaplněním dočasné složky */tmp*, která má sice nastavený stejný limit, ale jedná se o samostatný limit; dohromady je tedy možné zaplnit 100% RAM paměti.

Toho je docíleno pomocí vytvoření jednoduchého skriptu spuštěného na pozadí:

```
while true; do
        cat /dev/zero > $HOME/huge.file
        cat /dev/zero > /tmp/huge.file2
done
```

Pomocí tohoto skriptu je neustále zaplňováno volné místo na disku i pokud se systém pokusí nějaké místo uvolnit. Po spuštění skriptu přestává systém spouštět prohlížeč a na všechny příkazy poslané z příkazové řádky pouze vypisuje chybovou hlášku *Cannot allocate memory*. Systém se stává až do restartu počítače zcela nefunkčním.

1.2.3.2 Zabezpečení prohlížeče

Použitý prohlížeč Mozilla Firefox 34.0.5 z prosince 2014 je zabezpečen pomocí doplňku Webconverger, který je ve verzi 47 z března 2014. Tento doplněk nahrazuje implicitní chování prohlížeče, kromě odstranění všech nabídek také nahrazuje některé reakce na uživatelovi příkazy. Zde se jedná hlavně o zakázání všech klávesových zkratek ovládající prohlížeč, nemožnost zobrazení filesystému a několik interních stránek `about:`. Doplněk se stará také o implementaci filtru povolených domén, pokud je tak v nastavení specifikováno.

Jediným možným problémem je automatické zahrnutí doplňku Adobe Flash Player 11.2.202.425, který obsahuje dle [9] velké množství zranitelností, díky kterým může útočník spustit jakýkoliv kód a může tak tento software být nebezpečný.

1.3 Instant WebKiosk/UB

Instant WebKiosk/UB[10] je operační systém založený na Debian GNU/Linux a dále vyvíjen pod licencí GPL. Jedná se o systém bez počáteční instalace pouze s překopírováním obrazu systému na disk nebo nějaké jiné přepisovatelné médium (například USB flash paměť).

Aktuální verze systému ve verzi 13.0 vydaná v srpnu 2014 používá jako prohlížeč Chromium verze 35.0.1916.153. Veškerá nastavení systému probíhají pomocí webové stránky běžící na lokálním webovém serveru. Pro změnu všech nastavení je nutné využití placené verze, která dovoluje změnu domovské stránky, spořiče obrazovky, zabezpečení heslem proti změnám a automatické restartování prohlížeče při neaktivitě. Dále je možnost zakoupení *Extreme* edice, která navíc podporuje nastavení proxy serveru a přístupového URL filteru.

1.3.1 Zabezpečení

Samotné prostředí prohlížeče není nijak zabezpečeno, uživatel může libovolně měnit nastavení prohlížeče v rámci jedné relace. Po ukončení relace (zavření prohlížeče) je nastavení vráceno na původní hodnoty.

Filesystém je zabezpečen podobně jako u systému Webconverger. Celý operační systém je uložen na disku jako Squashfs soubor, který je při startu systému načten pouze pro čtení s přidanou zapisovatelnou vrstvou v RAM paměti. Po vypnutí počítače jsou všechny případné provedené změny ztraceny.

1.3.2 Rozšiřitelnost

Systém lze rozšířit po rozbalení Squashfs souboru pomocí dalšího operačního systému. Zde vzniká kompletní struktura načítaného systému, kde je možné provézt jakékoliv změny standardními nástroji systému Debian GNU/Linux. Po následném zabalení zpět do Squashfs souboru bude tento upravený systém načítán při startu systému místo původního.

1.3.3 Problémy implementace

Systém Instant WebKiosk/UB obsahuje hned několik zranitelností, které mohou být zneužity. Nejviditelnějším a nejsnadněji zneužitelným problémem je ponechání prohlížeče v nezabezpečeném stavu, tedy povolení jakýchkoliv úprav nastavení prohlížeče, včetně stahování souborů nebo doplňků prohlížeče a prohlížení filesystému. Přestože po restartu relace je nastavení obnoveno, v rámci jedné relace může uživatel s prohlížečem provádět téměř cokoliv.

1.3.3.1 Zabezpečení filesystému

Filesystém je pomocí AuFS připojen pouze pro čtení se zapisovatelnou částí v RAM paměti stejně jako u ostatních distribucí. Při ukončení relace je smazána složka s nastavením prohlížeče a všechny soubory v téže složce, ovšem nejsou zde mazány žádné další adresáře ani jejich obsah. Následně je zde zkopírován výchozí profil prohlížeče s nastavením, který je umístěn v domovském adresáři uživatele a je tedy uživateli zcela přístupný. Ostatní adresáře nejsou nijak chráněné proti změnám systému v rámci jednoho spuštění systému.

Zneužití zranitelnosti

V kombinace s povoleným stahováním souborů do libovolného umístění může uživatel jednoduše přepsat soubor s osobním nastavením prohlížeče `/chromium-webkiosk-defaults/Default/Preferencies` za nějaký náhodný (například binární) soubor. Při restartování prohlížeče bude vždy Chromium hlásit chybu s poškozeným profilem.

Druhou možností je stažení jakéhokoliv velkého souboru do složky s profilem, který zaplní celý disk a systém následně přestane reagovat na vstup až do restartu počítače.

1.3.3.2 Zabezpečení webového serveru

Nastavení systému je ovládáno skrze webové rozhraní, které je připojené na lokální webový server Apache. Tento server běží pod uživatelem *www-data*, který má pomocí souboru `/etc/sudoers` nastavená práva ke spouštění jakéhokoliv příkazu se zvýšeným oprávněním bez zadání hesla.

Zneužití zranitelnosti

Ponechání administrátorských práv webovému serveru je obrovskou zranitelností. Pokud se uživateli podaří pomocí tohoto server vyvolat externí příkaz (například využitím chyby, jako byla bezpečnostní chyba Shellshock[2]), může uživatel v systému dělat cokoliv. Nejdestruktivnější možnou změnou je přepsání celého disku pomocí příkazu `dd if=/dev/urandom of=/dev/sda1 bs=4096`, po tomto příkazu bude celý systém permanentně smazán bez možnosti obnovy a s nutností stažením nového obrazu systému.

Další nebezpečnou možností je nainstalování škodlivého kódu na disk, ať už se jedná o spam-bot nebo o jiný druh malwaru. Tento kód by byl následně persistentní i po restartech systému.

[2]Shellshock byla bezpečnostní chyba obsažená v Bash shellu zveřejněná v září 2014. Pomocí této chyby bylo možné přimět Bash shell provézt jakýkoliv příkaz v případě, že je tento příkaz připojeny na konci definice funkce, která je uložena v hodnotách proměnných prostředí. Více v [11].

Kapitola 2

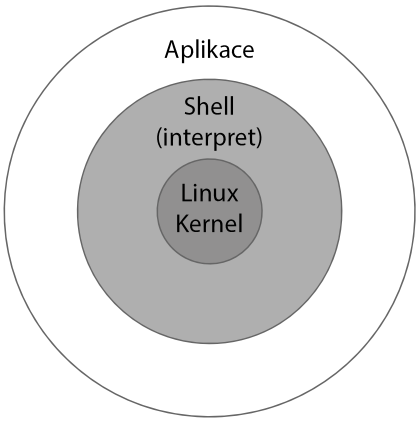
Analýza a návrh

2.1 Operační systém GNU/Linux

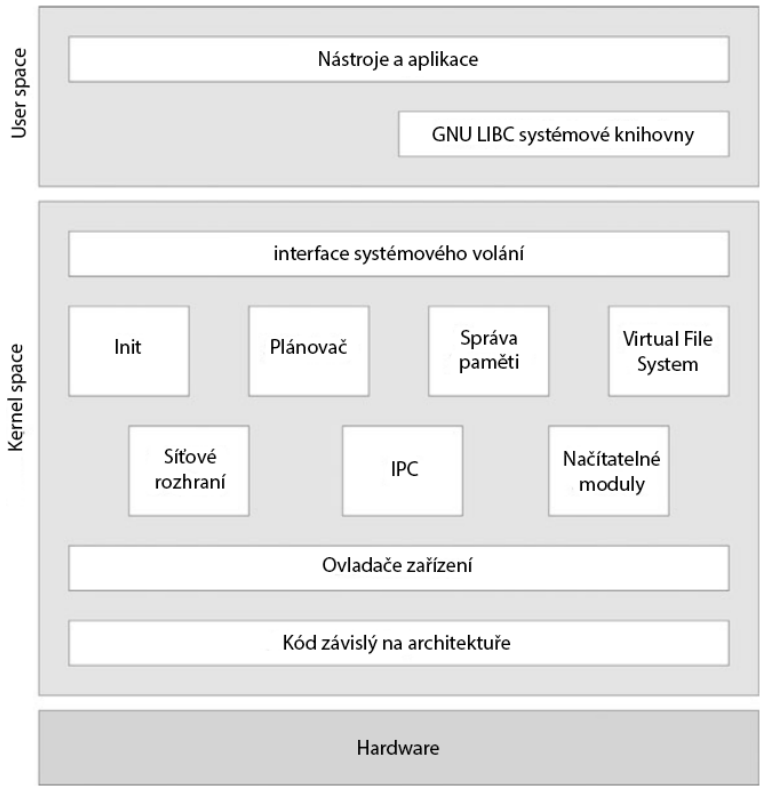
GNU/Linux je rodina UN*X operačních systémů založených na modelu svobodného a open-source vývoje a distribuci, který je organizován do několika vrstev. V zjednodušeném pohledu na obrázku 2.1 se jedná o 3 hlavní vrstvy.

Architektura systému

Jádrem systému je monolitický Linux kernel, který zprostředkovává přístup k hardwarovým prostředkům jako jsou paměť, CPU (prostřednictvím plánovače) a periferie. Kernel obsahuje také velké množství hardwarových ovladačů, které zjednodušují přístup k těmto periferiím. O stupeň vyšší vrstvou je *shell*, který poskytuje přístup uživateli ke kernelu. *Shell* poskytuje interpretaci příkazů a prostředky k načtení aplikací a jejich spuštění. Nejvyšší vrstva systému



Obrázek 2.1: Zjednodušená architektura GNU/Linux systému



Obrázek 2.2: Architektura GNU/Linux zaměřená na kernel

jsou pak samotné aplikace, jako je například grafické prostředí, webový prohlížeč, mailový server atp.

Obrázek 2.2 ukazuje architekturu zaměřenou na kernel, kde je systém rozdělen na 2 části. Horní část ukazuje *user space*, který obsahuje aplikace, nástroje a GNU C knihovnu a spodní *kernel space* - tedy část, která obsahuje různé komponenty kernelu. Tyto 2 části také značí rozdíly v přístupu k paměťovému prostoru, kde každý proces v *user space* části má svou vlastní nesdílenou nezávislou paměť, zatímco kernel operuje s vlastním adresním prostorem, ale všechny části kernelu sdílejí stejný prostor. Pokud jakákoliv část kernelu například odkazuje na špatný paměťový prostor, celý kernel selže a nastává tzv. *kernel panic*.

Konkrétní funkce jednotlivých částí kernelu nejsou pro tuto práci stěžejní, proto je zde podrobněji nebudu popisovat. Více o architektuře systému v [12].

Distribuce GNU/Linux

GNU/Linux distribuce je sada Linuxového jádra a doprovodného softwaru (viz sekce 2.1), které je sestavené do funkčního celku a šetří práci uživateli tím, že

nemusí sám celý systém kompilovat a sestavovat. Různých distribucí existuje velké množství a většinou se liší podle svého zaměření na cílového uživatele nebo svojí filozofií. Každá distribuce dle [13] obsahuje Linuxové jádro, systémové knihovny, běžné unixové nástroje, prostředí a grafické rozhraní (pokud se nejedná o čistě CLI distribuci bez grafického rozhraní). Naopak se mohou lišit ve skladbě programů, ve způsobu distribuce dodatečných programů, ve frekvenci a způsobu vydávání aktualizací, v instalačním programu a konfiguračních nástrojích, v řešení startovacích skriptů a jejich obsahu, v dodatečné úpravě některých programů a zejména jádra, v ceně a poskytovaných službách přidané hodnoty – dokumentace, technická podpora atd. (více v [14]).

Většina Linuxových distribucí obsahuje stejné nebo velice podobné jádro systému a rozdíly mezi jednotlivými distribucemi jsou až na vyšší softwarové vrstvě, proto v tomto ohledu není vůbec důležité, nad kterou distribucí bude systém postaven. Při výběru distribuce se zaměřím hlavně na dobu podpory. Podle [15] mezi nejpopulárnější distribuce patří Debian GNU/Linux, Ubuntu, Linux Mint, Fedora a openSUSE, budu tedy používat jednu z těchto distribucí.

2.1.1 Debian GNU/Linux

Debian GNU/Linux (zkráceně Debian) je Linuxová distribuce složena téměř výhradně ze svobodného open-source softwaru udržovaného a vyvíjeného skupinou nazývanou Debian project. Debian project je demokratická organizace s pevně danou strukturou a pravidly stanovenými zakládajícími dokumenty (podrobněji viz [16]).

Vývojové větve

Debian project udržuje vždy tři větve - *Stable*, *Testing* a *Unstable*. *Unstable* větev, nazývána také kódovým názvem *Sid*, obsahuje balíky které nejsou důkladně testovány na kompatibilitu s celým systémem. Jedná se o větev používanou většinou vývojáři pro vývoj samotného systému, protože obsahuje nejnovější knihovny. Tato větev může být nainstalována pouze pomocí systémového upgradu z větve *Testing*.

Testing větev slouží jako přechodné stádium mezi *Unstable* a *Stable*. Balíky se automaticky dostávají do *Testing* větve pokud splní určitá pevně daná kritéria. Celá větev se tedy průběžně aktualizuje, dokud nedojde k jejímu zmražení.

Stable větev obsahuje aktuální vydávanou verzi, která obsahuje stabilní a otestovaný software. *Stable* vzniká z *Testing* větve, kde dojde na několik měsíců ke zmražení, zatímco jsou opravovány chyby balíků, popřípadě jsou odstraněny některé balíky úplně, pokud obsahují mnoho chyb.

Více o vývojových větví v [17].

Cyklus vydání

Celý proces k vydání stabilní verze je velmi zdlouhavý a díky kvalitnímu testování je systém velmi stabilní. Kvůli dlouhému vývoji *Stable* větev neobsahuje nejnovější software a velmi rychle stárne, hlavně z toho důvodu, že nové verze Debianu jsou vydávány v průměru každé 2 roky. Aktuální verzí je Debian 7 s kódovým označením *Weezy* vydaný v květnu 2013. Podpora je stanovena přibližně na 1 rok po vydání další verze, podpora jedné verze tedy dle [18] trvá přibližně 3 roky. Verze Debian 6 jako první obsahuje tzv. *Long Term Support* (LTS) verzi, u které podpora trvá celkem 5 let, nyní konkrétně do února 2016. Pro Debian 7 a další verze není LTS podpora v době psaní práce potvrzena, ale ani také zcela vyloučena. Více o LTS v [19].

2.1.2 Ubuntu

Ubuntu je založena na distribuci Debian *Sid* (*Unstable* větev), ale s některými významnými balíky aktualizovanými na nejnovější verzi (např. GNOME, Firefox nebo LibreOffice).

Rozdíly oproti Debianu

Oproti Debianu využívá Ubuntu vlastní repozitáře balíků a vývoj systému je veden společností Canonical Ltd. Vydávání nových verzí je podle [20] pravidelné každých 6 měsíců s podporou pro bezpečnostní aktualizace po dobu 18 měsíců. Každé 2 roky také vychází LTS verze, pro kterou je podpora rozšířena na 5 let. Aktuální verzí je verze 14.10 s podporou do července 2015 a aktuální LTS verze je 14.04.1 s podporou do dubna 2019.

Dalšími rozdílnými vlastnostmi jsou vývoj vlastních součástí systému (grafické prostředí Unity, grafický server Mir), jednoduchá instalace proprietárních ovladačů (například pro grafické jednotky AMD nebo NVIDIA), podpora pro nesvobodné nebo patentově zatížené mediální kodeky a v neposlední řadě velkou komunitu uživatelů starající se o rozsáhlou dokumentaci a podporu způsobem wiki systému.

Ubuntu se dále dělí na 2 hlavní oficiální edice pro PC: Ubuntu Desktop a Ubuntu Server. Ubuntu Server se od Desktop verze podle [21] liší absencí X Window System (který umožňuje vytvoření GUI) a instalačním procesem, který místo GUI obsahuje pouze textové uživatelské rozhraní (TUI). V ostatních ohledech jsou obě varianty stejné a v žádných aspektech se neliší (chybějící X Window System lze navíc v případě potřeby do Server edice dodatečně doinstalovat).

2.1.3 Linux Mint

Linux Mint je distribuce založená podle zvolené edice buď na Ubuntu nebo Debianu. Jedná se hlavně o desktopový systém obsahující oproti Ubuntu nebo

Debianu vyšší počet implicitně nainstalovaných aplikací, vlastní uživatelské rozhraní Cinnamon[22] a sadu grafických nástrojů MintTools pro konfiguraci a správu systému (více informací o distribuci v [23]).

Cyklus vydání

Linux Mint vydává dvě edice – edici založenou na aktuálních distribucích Ubuntu a edici založenou na Debianu.

Edice založená na Ubuntu následuje stejný cyklus vydání i dobu podpory jako Ubuntu, kdy nové verze vycházejí přibližně s měsíčním zpožděním oproti Ubuntu.

Edice založená na Debianu využívá větev Debian *Testing* (viz 2.1.1) a místo pevných vydávání následuje tzv. *semi-rolling release* model. Oproti plnému *rolling release* modelu, kterým je větev *Testing*, Linux Mint pravidelně vytváří obraz této větve, kterou po určité době testování vydává jako balík aktualizací. Více o této edici v [24].

2.1.4 Fedora

Fedora je distribuce vyvíjena komunitou Fedora Project[25] a vlastněná společností Red Hat, Inc. Vznikla po zániku systému Red Hat Linux, ze kterého podle [26] vznikl komerční Red Hat Enterprise Linux a komunitou spravovaná distribuce Fedora (přesto stále financována společností Red Hat, Inc). Hlavní myšlenkou Fedory je inovace a striktní dodržování filosofie svobodného softwaru. Jako správce balíků využívá RPM Package Manager vyvinutý vlastní komunitou, který ve svých oficiálních repozitářích obsahuje pouze svobodný software (s možností přidání dalších repozitářů třetích stran, například právě s nesvobodným softwarem).

Cyklus vydání

Protože jeden z hlavních cílů systému dle [27] je soustředění se na nové a vývojové technologie, je vývojový cyklus celého systému velmi krátký, typicky nová verze vychází každých 6 měsíců. Fedora také nepodporuje žádnou LTS verzi, všechny verze mají podle [28] podporu přibližně 13 měsíců, konkrétně podpora končí vždy jeden měsíc po vydání distribuce vyšší o 2 verze. Aktuální verze systému je Fedora 21 vydaná v prosinci 2014 s předpokládaným koncem podpory v lednu 2016 po vydání Fedory 23.

2.1.5 openSUSE

OpenSUSE je distribuce vyvíjená komunitou openSUSE Project a financována z největší části společností SUSE (celý seznam sponzorů v [29]). OpenSUSE, stejně jako Fedora, využívá RPM Package Manager, ale pro své potřeby vyvinuli konfigurační nástroj YaST, který se stará například o nastavení systému,

konfiguraci sítě a firewallu nebo o správu RPM balíků. Pomocí modulů se dá tento systém rozšířit o mnohem větší počet funkcí, celý seznam je dostupný v [30].

Další vlastností jsou 2 hlavní verze systému – klasické vydávání nové stabilní verze po určitém vývojovém cyklu a Tumbleweed[31] edice, která je zástupcem tzv. *rolling release*, tedy vydávání nejnovějších verzí veškerého softwaru ihned po testování bez čekání na periodické cykly vydání. *Rolling release* je vhodný pokud potřebujeme vždy nejnovější verze s tím rizikem, že může být nutné například zkompilovat vlastní modul pro kernel systému pro některou funkčnost (zejména grafické ovladače).

Cyklus vydání

OpenSUSE podle [32] vychází každých 8 měsíců s oficiální podporou vždy do 2 měsíců po vydání dvou dalších verzí, typicky se jedná tedy o 18 měsíců. Nejaktuálnější verzí je openSUSE 13.2 vydaný v listopad 2014 s podporou přibližně do května 2016. Komunita ale vždy vybírá některé verze, pro které poskytuje prodlouženou podporu s názvem Evergreen[33]. Aktuální takovou verzí je openSUSE 13.1, pro kterou je komunitní podpora naplánována do listopadu 2016, tedy prodloužení na 3 roky.

2.1.6 Srovnání systémů a výběr

Všechny zde analyzované distribuce mají stejný základ systému a pouze se většinou liší v dodávané nadstavbě, ať už v grafických prostředích nebo i samotných nástrojích určených pro správu systému. Pro účely lock-down systému tyto rozdíly ale nejsou vůbec důležité, z pohledu uživatele systém bude pouze zobrazovat webový prohlížeč, tudíž všechny ostatní prvky systému zde nehrají žádnou roli.

V analýze jednotlivých systémů jsem se proto zaměřil hlavně na dobu jejich podpory a způsob vývoje. Pro systém určený na jednoúčelové zařízení není nutný rapidní vývoj aplikací, na druhou stranu ale bezpečnostní aktualizace jsou i tak stále nutné. Je tedy třeba zvolit takovou distribuci, která poskytuje přijatelnou délku podpory, aby nebylo nutné systém tak často aktualizovat po vypršení podpory starší verze. Všechny zde popisované distribuce podporují systémový upgrade ze starší verze na novější bez nutnosti instalovat systém znovu, tento postup ale může být problémový a někdy způsobovat chyby, které by bylo nutné při upgradech řešit (viz. [34]).

Z těchto důvodů jsem se zaměřil pouze na LTS distribuce, které jsou v tomto případě reprezentovány distribucemi Ubuntu a Debian GNU/Linux s podporou 5 let. Ubuntu je založené na Debianu (viz. sekce 2.1.2) a pouze ho rozšiřuje o některé nástavby určené spíše domácím uživatelům, proto se jako vhodnější volba zdá Debian GNU/Linux. Protože ale Debian GNU/Linux v době psaní práce stále nemá jasně definovanou podporu pro další LTS

verze a podpora aktuální LTS verze končí již v únoru 2016, nakonec jsem se rozhodl pro využití distribuce Ubuntu.

Konkrétně pro tuto práci budu používat distribuci Ubuntu 14.04.1 LTS instalovanou bez grafického rozhraní z Minimal CD[35], která obsahuje pouze nejnutnější balíky pro instalaci systému a samotné balíky pro systém stahuje z repozitářů až během instalace, je tedy zaručena aktuálnost všech balíků.

2.2 Webový prohlížeč

Webový prohlížeč představuje klíčovou aplikaci celého systému, jeho vlastnosti a schopnosti určují, jaké další bezpečnostní prvky by bylo třeba implementovat v operačním systému. Jedná se zejména o schopnosti blokovat určené stránky (nebo naopak pouze určené stránky povolit) a zabezpečení prostředí proti změnám konfigurace prohlížeče. Dále není žádoucí, aby bylo pomocí některé vestavěné funkčnosti prohlížeče možné prohlížení nebo změna filesystému – přestože filesystém bude chráněn zvlášť, tato funkčnost by jen dávala prostor k pokusům o prolomení bezpečnosti.

2.2.1 Chromium

Chromium je open-source prohlížeč původně sloužící pouze jako open-source projekt, na kterém měl být založen Google Chrome. Nakonec ale vývojáři vydali i Chromium jako samostatný prohlížeč (více v [36]).

Blokování stránek

Blokování stránek je zajištěno pomocí souboru, který obsahuje administrátorem uplatňované politiky. Politiky jsou dle [37] rozděleny na 2 typy – politiky vynucené administrátorem a pouze doporučené politiky. Vynucené politiky se nacházejí ve složce `/etc/chromium/policies/managed/` ve formátu JSON[3] souboru. Pro blokování stránek se zde konkrétně jedná o direktivy *URLBlacklist* a *URLWhitelist*. Direktivy podporují použití zástupného znaku `*` , který odpovídá libovolnému znaku a jeho počtu. *URLWhitelist* se používá při použití *URLBlacklist* pro specifikování, které stránky budou přístupné. Je například možné zablokovat jednu doménu a pomocí *URLWhitelist* následně povolit jedinou subdoménu zakázané domény.

Příklad obsahu JSON souboru pro blokování všech domén kromě domén *google.com* a *google.cz* je v 2.1.

[3] JavaScript Object Notation je způsob zápisu dat v podobě dvojice *atribut—data*, určený pro přenos dat v lidsky čitelné podobě.

Listing 2.1: Příklad povolení pouze domén google.cz a google.com v prohlížeči Chromium

```
{
"URLBlacklist": ["*"],
"URLWhitelist": ["google.cz","google.com"]
}
```

Zabezpečení proti změnám konfigurace

Chromium obsahuje režim Kiosk, který spouští prohlížeč v celoobrazovkovém módu bez adresního řádku a jakýchkoliv ovládacích prvků. Při absenci navigačního panelu může nastat problém s návratem na předchozí nebo domovskou stránku. Pokud se uživatel chce vrátit na domovskou stránku (dostal se na stránku, kam nechtěl atp.), bez znalosti klávesové zkratky se na tuto stránku nijak nedostane.

Druhou možností je znovu využití politik. Politiky obsahují velké množství direktiv pro vynucené chování prohlížeče (kompletní seznam je na stránkách Chromium Project[4]). Pomocí těchto direktiv se dá vynutit chování téměř všech funkcí, včetně vynucení instalovaných doplňků. Společně s blokováním adres lze následně zakázat přístup do dialogu nastavení a to díky tomu, že Chromium využívá pouze interní adresu `chrome://config` pro všechna nastavení. I kdyby ale uživatel měl k tomuto nastavení přístup, díky nastaveným politikám budou téměř všechny možnosti nastavení nezměnitelné.

2.2.2 Google Chrome

Google Chrome je svobodný prohlížeč vyvíjený společností Google Inc. Je zcela založen na projektu Chromium Project pouze s několika malými rozdíly. Vše, co podporuje Chromium, lze najít i v Google Chrome prohlížeči.

Rozdíly oproti prohlížeči Chromium jsou v přítomnosti hlášení havárií prohlížeče, sběru dat o využívání, vestavěná podpora licenčně zatížených kodeků a nesvobodný PPAPI[5] plugin pro Adobe Flash. Další odlišnosti jsou závislé na linuxové distribuci, seznam možných změn je popsán v dokumentaci[6].

2.2.3 Opera

Opera je vyvíjena společností Opera Software ASA jako proprietární software s uzavřeným zdrojovým kódem. Licenční ujednání [38] mimo jiné zakazuje jakékoliv úpravy programu, jediná možnost jak rozšířit (nebo naopak omezit)

[4] `http://www.chromium.org/administrators/policy-list-3`

[5] PPAPI je multiplatformní API pro pluginy pro webové prohlížeče.

[6] `https://code.google.com/p/chromium/wiki/ChromiumBrowserVsGoogleChrome`

funkčnost je pomocí doplňků a rozšíření. Od verze 15.0 Opera využívá vykreslovací jádro Chromium a velkou část Chrome API pro tvorbu rozšíření, z toho také vyplývá možnost použití některých doplňků určených pro Chrome i v Opeře bez úprav. Kompletní seznam převzatých API funkcí je k nalezení v [39].

Opera v poslední době vyvíjí pro Linuxové distribuce jen 64-bitovou verzi (momentálně ve verzi 27), pro 32-bitovou architekturu je dle [40] jako nejnovější verze dostupná verze 12.16 z roku 2013.

Blokování stránek

Blokování stránek lze provést pomoci funkce blokování obsahu, které je implicitně nastavené využíváním blacklistu, tedy v režimu povolení všech stránek až na určité výjimky. Z grafického prostředí se toto chování nedá změnit, ale podle [41] dá se změnit pomocí direktivy *prioritize excludelist* v konfiguračním souboru `urlfilter.ini`, díky které se dá docílit blokování všech stránek kromě specifikovaných. Ve specifikovaných pravidlech lze použít zástupné znaky `*` (jeden nebo více libovolných znaků) a `?` (přesně jeden libovolný znak).

Blokování stránek ale nefunguje na vestavěný editor pokročilých funkcí prohlížeče, ke kterému se přistupuje pomocí zadání adresy `opera:<funkce>`, například pro zobrazení kompletního nastavení se jedná o příkaz `opera:config`. Tyto stránky zůstávají vždy přístupné i při blokování všech stránek.

Zabezpečení proti změnám konfigurace

Opera obsahuje režim Kiosk, který slouží pro zakázání nebo vynucení některých funkcí. Kiosk režim je ovlivňován pomocí přepínačů (jejich seznam a popis v tabulce B.1), které se mohou libovolně kombinovat. Pomocí dostupných přepínačů lze prohlížeč téměř zcela zabezpečit proti změnám, ovšem za cenu snížené uživatelské přívětivosti (více v [42]).

Například pro zakázání přístupu k dialogu nastavení prohlížeče je nutné zakázat použití klávesových zkratek a zároveň zakázat kontextové menu (při zakázaném menu nelze kliknout na hlavní nabídku, v opačném případě toto je možné a lze se dostat do nastavení). Vedlejším efektem tohoto nastavení zároveň bude nemožnost kopírovat a vkládat text, což by mohl být problém například v systémech sloužících jako vyhledávací katalog pro veřejnost, kde by tato možnost mohla chybět.[41]

2.2.4 Mozilla Firefox

Mozilla Firefox (zkráceně Firefox) je svobodný prohlížeč vyvíjen společností Mozilla Foundation pod MPL 2.0 (celý obsah v [43]) licencí. Tato licence je

Listing 2.2: Příklad blokování domény google.com v prohlížeči Mozilla Firefox

```
@–moz–document regexp(".*google.com.*"){
html, page, window { display: none !important; }
}
```

variantou copyleft[7] licence, která dovoluje kombinaci vlastního kódu s kódem pod jinou licencí (open-source i proprietární) s minimálními omezeními. Zdrojový kód Firefoxu je tedy možné libovolně doplňovat vlastním kódem, pokud kód stále zůstává pod MPL 2.0 licencí.

Blokování stránek

Firefox neobsahuje žádné možnosti pro blokování stránek či obsahu. Tuto funkčnost lze doplnit pomocí instalovatelných doplňků či pomocí změny obsahu stránky pro vykreslování.

Blokování pomocí změny obsahu stránky lze dosáhnout pomocí editací souboru `userContent.css`, který je určen pro změnu vzhledu webových stránek (více v [44]). Vhodnou editací tohoto souboru lze dosáhnout například nahrazení obsahu všech blokovaných stránek za prázdnou stránku (Kód 2.2 ukazuje kód pro blokování domény google.com). Velká nevýhoda tohoto řešení ale je, že blokovaná stránka se stále bude načítat, uživateli se pouze místo obsahu bude zobrazovat prázdná stránka.

Zabezpečení proti změnám konfigurace

Prohlížeč neobsahuje žádný Kiosk režim nebo podobné zabezpečení, znovu se dá toto chování emulovat pomocí instalovatelných doplňků nebo pomocí změny konfiguračního souboru ovlivňující vzhled UI.

Vzhled a složení UI lze ovlivnit pomocí souboru `userChrome.css`, příklad kódu pro odstranění hlavní nabídky je zobrazen na 2.3. Pomocí tohoto souboru lze ovlivnit všechny části, ať už skrýt, přemístit nebo přidat libovolné prvky UI.

I přes odstranění hlavní nabídky se ale lze do nastavení dostat pomocí klávesové zkratky. Tyto zkratky se daly zakázat změnou kódu aplikace, bohužel ale kvůli nevyřešené chybě[8] z roku 2003, kdy toto nastavení nebylo bráno v potaz, možnost změny klávesových zkratek z kódu zanikla. Stále je ale možné klávesové zkratky zakázat pomocí doplňku nebo přímo zakázáním v operačním systému.

[7] Copyleft je forma licencování, která dovoluje každému distribuce či úpravu díla, pokud toto dílo bude stále pod stejnou licencí.

[8] `https://bugzilla.mozilla.org/show_bug.cgi?id=201011`

Listing 2.3: Příklad odstranění hlavní nabídky prohlížeče Mozilla Firefox

```
@namespace url("http://www.mozilla.org/keymaster/
    gatekeeper/there.is.only.xul");
#PanelUI-button { display: none !important; }
```

2.2.5 Výběr prohlížeče

Z dostupných prohlížečů se zdá jako nejlepší volba Chromium, hlavně díky velmi jednoduchému systému blokování stránek a vynucení nastavení pomocí pravidel bez možnosti jejich změny uživatelem. Oproti ostatním prohlížečům není nutné tyto funkce řešit doinstalováním doplňků nebo využitím funkcí, které k tomu nebyly původně určeny. Jako vhodnou alternativou může být prohlížeč Opera, zde je ale problém, že vývoj pro 32-bitový Linuxový systém skončil v roce 2013 a jedná se o velmi neaktuální prohlížeč nepodporující většinu funkcí 64-bitové verze včetně Chrome API. Pro svou práci budu tedy využívat pouze prohlížeč Chromium.

2.3 Nástroje pro zabezpečení systému

Zabezpečení prohlížeče společně se zabráněním vypnutí prohlížeče zabezpečuje základní ochranu proti zneužití a takové nastavení je dostatečné pro obyčejného uživatele. Pokud se ale k systému dostane útočník, který se bude snažit z prostředí prohlížeče dostat, může se mu to například pomocí nějaké chyby v prohlížeči povést. Zde je vždy na místě provézt i nějaké další dodatečné zabezpečení systému a předpokládat, že došlo k prolomení ochrany prohlížeče.

2.3.1 Zabezpečení kořenového adresáře

Zabezpečení kořenového adresáře znamená zabezpečení přístupu a hlavně zabránění změně obsahu celého filesystému. Po dosažení takového zabezpečení nebude mít útočník i po prolomení prohlížeče žádnou možnost změnit chování samotného systému a bude se stále jednat o konzistentní celek.

Kompletnímu zakázání zápisu se dá docílit používáním kořenového adresáře jako pouze pro čtení, další možným způsobem je popřípadě změna umístění kořenového adresáře pomocí volání *chroot*.

Filesystém pouze pro čtení

Nejjednodušším řešením, jak zabránit změnám na celém filesystému, je přepnutí celého systému do režimu pouze pro čtení. Útočník pak nemá možnost žádný soubor změnit, a to ani kdyby k danému souboru měl mít práva zápisu.

Pro korektní funkci systému je ale nutné některé složky zpřístupnit i pro zápis, jinak části systému nemusí správně fungovat.

Mezi složky a soubory, který potřebují zapisovatelný prostor, podle [45] patří soubory `/etc/mtab`, `/etc/resolv.conf` a složka `/var`.

Soubor `mtab` obsahuje seznam všech připojených filesystémů společně s parametry, které byly použity při připojení. Soubor `resolv.conf` se používá pro ukládání informací od DHCP serveru, konkrétně informace o DNS resolverech (pokud systém nepoužívá statickou konfiguraci, pak tento soubor může zůstat pouze pro čtení). Složka `/var` obsahuje často měněné soubory jako jsou zálohy klíčových systémových souborů, cache paměť některých aplikací, logy atp.

Systémové volání chroot

Chroot[46] je privilegované systémové volání, které slouží ke změně kořenového adresáře pro daný proces a všechny jeho potomky. Vzniká tím samostatný oddělený prostor, často nazývaný *chroot jail*. Neprivilegovaný uživatel nemůže takové prostředí opustit a může ovlivňovat soubory pouze v daném prostředí bez možnosti zásahu do opravdového kořenového adresáře.

Chroot volání může provézt pouze privilegovaný účet a zároveň také pouze on může prostředí opustit, jinak by běžný uživatel mohl prostředí jednoduše opustit. Právě ale tato nutnost použití privilegovaného účtu činí systém zranitelným. Přestože se program může vzdát oprávnění správce, je možné tyto práva získat zpět pomocí nějaké chyby v programu či přímo v jádru systému (příklad opuštění prostředí v [47]). Z tohoto důvodu nelze chroot považovat za bezpečnostní prvek a používá se spíše pro testování aplikací mimo produkční prostředí nebo pro obnovení poškozeného systému (načtení systému z jiného média a následovným přepnutí pomocí chroot do poškozeného filesystému).

2.3.2 Zabezpečení domovské složky

Do domovské složky má uživatel implicitní výhradní přístup – kromě jeho osobních souborů se zde nachází vyrovnávací paměť či osobní nastavení různých aplikací. Můžou se zde vyskytovat i další osobní nastavení, viz. [48], kompletní zakázání přístupu uživatelovi do této složky by tedy mohlo znamenat nekorektní běh některých aplikací, které do této složky potřebují zapisovat.

Pro lock-down systém je tento přístup nevhodný, složka by měla zůstávat mezi jednotlivými relacemi persistentní, na druhou stranu kvůli vyrovnávací paměti aplikací je povolený zápis nutný. Jako řešení se nabízí povolení zápisu do domovské složky, záloha implicitního profilu v jiném umístění a překopírování jeho obsahu po skončení každé relace. Vylepšením takového způsobu může ještě být využití UnionFS filesystému.

UnionFS

UnionFS dovoluje spojení souborů a složek z různých filesystému do jediného uceleného filesystému. Obsah těchto adresářů je spojen a zobrazován v jedné sloučené složce v novém virtuálním filesystému. Spojované složky mohou být spojeny buď jako pouze pro čtení nebo zapisovatelné, čehož lze pro zabezpečení domovské složky využít. Implicitní profil bude sloužit jako část pouze pro čtení, zatímco nějaké dočasné umístění, které bude po konci relace smazáno, bude sloužit jako zapisovatelná část.

2.3.3 Správce oken

Správce oken není bezpečnostní prvek, přesto jeho výběrem lze bezpečnost jistým způsobem ovlivnit. V případě použití některého obyčejného správce oken bude možno velikosti a umístění okna libovolně měnit. Útočník pak může okno zmenšit a přesunout téměř mimo obrazovku a do ukončení relace bude vidět pouze tapeta na ploše. Samotný systém nelze tímto způsobem nijak ohrozit, normální uživatel může ale systém pokládat za nefunkční a zbytečně volat správce systému k opravě.

Nodm

Ideální display manager by neměl dovolit nic jiného, než zobrazit okno jediné aplikace a přesně toto dovoluje nástroj nodm. Jedná se o minimálního správce oken, který se automaticky přihlásí jako daný uživatel a spustí grafické uživatelské rozhraní. Spuštěné programy nelze přesouvat ani měnit rozměry jejich okna, pouze je lze vypnout.

Nodm implicitně používá konfigurační soubor `/etc/default/nodm`, který obsahuje 5 direktiv[9], ze kterých nejdůležitější jsou číslo virtuálního terminálu, kde má nodm běžet, jméno uživatele, pod kterým se má systém přihlásit, a zda je automatické přihlášení aktivní. Při startu prostředí používá soubor `.xsession` jako skript, který obsahuje, jaký program má být v grafickém prostředí spuštěn. Tento soubor může obsahovat jakýkoliv shellový skript (jak spuštění jediné grafické aplikace, tak i složitý skript obsahující volání dalšího skriptu a programů na pozadí), po jehož dokončení se ukončí i celé grafické prostředí.

2.3.4 Další zabezpečení

SysRq

SysRq funkce jsou klávesové zkratky využívající klávesu SysRq, které přijímá kernel a dávají uživateli přístup k nízkoúrovňovým příkazům. Většinou se

[9]Kompletní seznam direktiv v dokumentaci na `http://apt-browse.org/browse/debian/wheezy/main/amd64/nodm/0.11-1.3/file//usr/share/doc/nodm/README/`

používá k restartování nebo obnovení systému v nouzových stavech při ztrátě odezvy bez poškození filesystému. Více informací o funkci SysRq v [49].

U zabezpečeného klienta nejsou tyto funkce zcela žádoucí, zabezpečení systému sice nijak nepoškodí, mohly by ale případně ovlivnit funkčnost samotného stroje (například funkce připojení všech disků jako pouze pro čtení by generovalo velké množství chyb na zdánlivě funkčním počítači). Jako ideální řešení se nabízí buď zakázání všech SysRq funkcí nebo povolení pouze restartování a vypínaní počítače.

USB periferie

Existuje reálná možnost použití obyčejné USB flash paměti s upraveným firmwarem pro útok (konkrétní detaily o útoku v [50]), ať už na samotný stroj, tak i na ostatní uživatele (phishing). Pro eliminaci tohoto a i možných dalších problému je ideální zakázat použití všech paměťových USB zařízení.

Virtuální terminály

GNU/Linux obsahuje několik nezávislých virtuálních terminálů, typicky se jedná o textové terminály na prvních šesti terminálech a grafické rozhraní na sedmém (viz. [51]). Stejný systém používá i Ubuntu a dovoluje mezi virtuálními terminály libovolně přepínat pomocí klávesových zkratek.

2.4 Technologie pro tenké klienty a server

Oproti samostatnému klientovi se architektura tenkého klienta a serveru v mnohém neliší. Na straně server stále musí být nějakým způsobem funkční kopie systému, který chceme, aby běžel na tenkém klientovi. Tedy zde se bude jednat téměř o stejný systém, jako na samostatném klientovi. Existuje několik variant tenkých klientů, které se liší hlavně ve způsobu, jakým klient získává data ze serveru a jakým hardwarem musí být klient vybaven. Klient může obsahovat svůj vlastní disk s minimálním operačním systémem nebo také může být zcela bez disku s bootováním po síti.

2.4.1 Tenký klient s lokálním diskem

Zástupcem tenkých klientů s vlastním diskem a systémem je klient využívající připojení ke vzdálené ploše pomocí protokolů k tomu určených (například RFB protokol používaný u VNC nebo proprietární Microsoft protokol RDP).

Tenký klient v tomto případě bootuje z lokálního disku a načítá lokální operační systém (ať už se jedná o plnohodnotný systém nebo pouze o minimální operační systém). Následnou jedinou funkcí systému je připojení ke vzdálenému serveru, ze kterého čerpá veškerá data.

Výhodou je jednoduchost konfigurace, nevýhod je ale hned několik. Mezi největší patří nutnost správy také lokálních systémů klientů místo správy pouze samotného serveru. Při nasazení na různý hardware mohou někdy vznikat chyby, které by bylo nutné samostatně řešit. Dalším problémem může být náročnost systému, v závislosti na použitém systému mohou být hardwarové nároky neúměrně vysoké, od tenkých klientů se ale očekávají minimální požadavky na hardware a to hlavně z ekonomických důvodů (ať už z hlediska spotřebované energie nebo umožnění použití staršího, jinak nepoužívaného, hardwaru).

Právě z důvodů těchto nevýhod nebudu ve své implementaci používat klienty s vlastním diskem (a tedy systémem), ale budu uvažovat pouze o bezdiskovém systému.

2.4.2 Bezdiskový tenký klient

Bezdiskový klient využívá k zavedení systému bootování po síti pomocí vzdáleného serveru na kterém je umístěn systém, který klient bude využívat jako svůj lokální. Díky tomuto řešení není nutný žádný lokální disk, pouze RAM paměť pro dočasné soubory.

Jediným požadavkem takového systému je podpora prostředí PXE u síťové karty. Jedná se o prostředí umožňující vyvolání bootování ze vzdáleného serveru pomocí síťového rozhraní (podrobněji v [52]). V případě, že síťová karta tento protokol nepodporuje, existují softwarové náhrady, které tuto funkci emulují (například projekt Etherboot/gPXE nebo iPXE). V tomto případě se pomocí média s takovým softwarem načte systém, který následně vyvolá bootování ze sítě. Kromě podpory PXE nejsou žádné další požadavky na hardwarovou výbavu klientů, pro běh dostačuje několik desítek MB RAM paměti a procesor s rychlostí v řádech stovek MHz. Není také nutná žádná administrace jednotlivých klientů, veškeré servisní zásahy jsou prováděny na serveru na jediném místě. Změna načítaného systému se projeví na všech klientech bez dalších nutných úkonů.

Nevýhodou oproti tenkým klientům s vlastním diskem zůstává pouze složitější konfigurace.

2.4.2.1 Linux Terminal Server Project

LTSP je Linuxový open-source projekt aktuálně ve verzi 5.5.1, implementující logiku serveru pro tenké klienty. Podporuje jak bezdiskové tenké klienty tak i klasické klienty s vlastním systémem a výpočetním výkonem (aplikace se spouštějí lokálně, ale některé soubory jsou vyvolávány a ukládány pouze na server, například domovská složka). LTSP je balík předkonfigurovaných aplikací společně se skripty řídícími chování serveru, nejedná se tedy o celý operační systém a je možné ho doinstalovat již do existujícího prostředí. Tento balík je podle [53] k dispozici v repozitářích Linuxových distribucích Ubuntu

GNU/Linux, Gentoo, OpenSUSE a Ubuntu – pro ostatní distribuce je balík k dispozici jako zdrojový kód.

Pro chod serveru jsou určené balíky v sadě ltsp-server-standalone, nejdůležitější balíky zde jsou[10]:

- `debootstrap` – Vytvoří strukturu nového systému v určeném umístění a stáhne a nainstaluje zde nezbytné balíky pro základní funkčnost[54]
- `nbd-server` – Poskytuje soubor jako blokové zařízení pro vzdálené GNU-/Linux počítače[55]
- `squashfs-tools` – Nástroj pro vytvoření vysoce komprimovaného read-only souborového systém pro GNU/Linux[56]
- `tftpd-hpa/atftpd/dnsmasq` – Různé varienty TFTP serveru
- `isc-dhcp-server/dhcp3-server/dnsmasq` – Různé varianty DHCP serveru

Tyto programy zajišťují chod a komunikaci serveru s klienty, je ale třeba také vytvořit samotné prostředí, ve kterém budou klienti pracovat – k tomu slouží další sada balíků *ltsp-client*.

Balíky obsažené v *ltsp-client* obsahují nástroje pro chod klientů, jedná se například o NBD, TFTP, SSH a DHCP klienta, nástroj pro počáteční zavedení systému z RAM paměti (initramfs) nebo nástroje pro běh grafického prostředí (xorg, x11-xserver-utils)[11].

Princip LTSP serveru

Serverová část kromě nutných programů pro podporu PXE bootování a spojení s klienty (DHCP, TFTP, SSH server) obsahuje také druhý Linuxový systém, který slouží jako systém pro klienty. Tento druhý systém je vytvořen v chrootovém prostředí na serveru a je zde nainstalován minimální GNU/Linux systém, který je upraven tak, aby byl co nejefektivnější při bootování po síti (podrobněji v [57]). Součástí prostředí jsou také upravené init skripty, které ovládají zavádění systému na klientech. Tyto skripty jsou libovolně upravitelné a lze díky nim měnit chování klientů nejen při zavádění systému, ale i po něm. Chrootové prostředí reprezentuje kompletní systém, jaký bude klient používat – jaké programy zde budou nainstalovány, takové programy bude mít klient k dispozici.

Chování klientů lze podle [58] kromě přímou změnou init skriptů také ovlivnit nepřímo samotnými konfiguračními soubory LTSP. Hlavní konfigurační soubor klientů `lts.conf` může kromě globálních nastaveních obsahovat

[10] Kompletní seznam balíků na `http://packages.ubuntu.com/precise/ltsp-server-standalone`

[11] Kompletní seznam balíků na `http://packages.ubuntu.com/trusty/ltsp-client`

i nastavení konkrétní pro jednotlivé klienty identifikované pomocí MAC adresy. Může se zde jednat například o automatické přihlášení vybraného uživatele, počet povolených virtuálních terminálů a jaké relace zde mají běžet nebo implicitní display manager. Konfiguračních parametrů klientů je velké množství[12], tyto hodnoty jsou ale z hlediska vytvoření lock-down systému nejdůležitější.

Pro vzdálený přístup k tomuto systému podporuje LTSP jak NBD, tak NFS. Pokud je vybráno NFS, klient přistupuje přímo do chrootového prostředí, v případě NBD je nutné nejdříve vytvořit z tohoto prostředí blokové zařízení, které následně bude klient používat. Z tohoto plyne i výhoda NFS, kdy změna v prostředí se projeví okamžitě, zatímco v případě NBD je vždy třeba blokové zařízení vytvořit znovu. Na druhou stranu je dle [59] NBD oproti NFS rychlejší při vzdáleném přístupu.

Bootování klientů

Celé načtení systému ze serveru až do funkčního stavu u klienta je podle [60] a [61] rozděleno do několika fází:

1. **Bootování** – Načítání systému začíná bootováním ze sítě pomocí prostředí PXE.

2. **Získání IP adresy** – V další fázi PXE posílá zprávu typu *DHCPDISCOVER* s PXE příznakem. DHCP server přijímající PXE požadavky odpovídá zprávou *DHCPOFFER* se základní konfigurací síťového rozhraní společně s adresou TFTP serveru.

3. **Stažení kernelu** – PXE posílá TFTP serveru zprávu D*HCPREQUEST*, který odpovídá cestou ke kernelu. Klient následně pomocí TFTP stáhne daný kernel do RAM paměti a spustí ho.

4. **Připojení dočasného filesystém** – Po spuštění předává PXE řízení kernelu, který dočasně připojí malý filesystém do RAM paměti (filesystém je součástí staženého kernelu) a spouští speciální *linuxrc* skript.

5. **Načítání kernel modulů** – skript *linuxrc* skenuje PCI sběrnici a hledá všechna síťová rozhraní. Při nalezení a identifikování síťového adaptéru je načten kernel modul podporující dané zařízení.

6. **Konfigurace rozhraní** – v této fázi je znovu poslán DHCP požadavek a po odpovědi je nakonfigurováno síťové rozhraní *eth0* podle došlých informací.

[12] Kompletní seznam možných parametrů a jejich vysvětlení na `http://bazaar.launchpad.net/~ltsp-docwriters/ltsp/ltsp-docs-trunk/view/head:/lts.conf.xml`

7. **Připojení filesystému** – Po konfigurace síťového rozhraní je připojen filesystém ze serveru pouze pro čtení; může se jednat o připojení pomocí NFS protokolu nebo získání NBD zařízení z NBD serveru.

   a) NBD – Network Block Device je blokové zařízení, které se používá k přístupu k paměťovému zařízení, které není fyzicky umístěné v lokálním počítači, ale jehož obsah je umístěn na vzdáleném serveru. Lokální počítač s tímto zařízením pracuje jako s klasickým lokálním blokovým zařízením, pouze všechny operace jsou prováděny vzdáleně.[62]

   b) NFS – Network File System je distribuovaný filesystém protokol dovolující přístup k souborům na vzdáleném počítači stejně jako kdyby byly připojeny lokálně.[63]

   Následně je nad daným filesystémem vytvořen i zapisovatelný prostor v RAM paměti pomocí OverlayFS.

8. **Předání řízení** – V poslední fázi je řízení předáno init programu, který podle souboru `/etc/inittab` načte celý systém.

Pro klienta bude vše od této chvíle fungovat stejně, jako kdyby bootování bylo prováděno z lokálního disku.

Kiosk režim

LTSP obsahuje na Debian GNU/Linux a Ubuntu distribucích jednoduchý kiosk režim. Pokud je při vytváření prostředí pro klienty použit parametr *kiosk*, na konci se pomocí jednoduchého skriptu nainstaluje prohlížeč, správce oken a barevné schéma oken. Následně lze v hlavním konfiguračním souboru specifikovat spuštění kiosk režimu na určeném virtuálním terminálu. Tato kiosk relace je implicitně ovládána skriptem `/usr/share/ltsp/screen.d/kiosk` v chrootovém prostředí, který nastaví proměnné systému, vytvoří dočasného kiosk uživatele s dočasnou domovskou složkou a pomocí tohoto uživatele pouští další skript `/usr/share/ltsp/kioskSession`, který se stará přímo o samotnou relaci. V původní implementaci skript obsahuje kromě několika kontrol nastavení pouze spuštění prohlížeče, funkčností se ale jedná o stejný skript jako je soubor `.xsession` popsaný v sekci 2.3.3. Po vykonání skriptu (tedy po ukončení relace) jsou všechny běžící procesy uživatele ukončeny a dočasný uživatel je i s domovským adresářem smazán.

Tyto skripty ale vytvářejí relaci na lokálním počítači a zcela nevyužívají výkon serveru. Server v tomto případě slouží pouze jako vzdálený přístup k souborům, samotný prohlížeč je ale spuštěn v RAM paměti klienta pro vyšší rychlost odezvy.

2.4.3 Výběr systému

Z těchto dvou přístupů k tenkému klientovi je jednoznačně výhodnější bezdiskový klient, kromě nutnosti podpory PXE bootování (která lze v případě nutnosti nahradit externím softwarovým řešením) má nižší hardwarové nároky na klienty a zároveň jednodušší správu těchto stanic.

Pro vlastní implementaci využiji projekt LTSP a to hlavně z důvodů jednoduché úpravy jakéhokoliv chování serveru. LTSP se stará o korektní konfigurace všech nutných služeb a zároveň dodává vlastní skripty pro vytvoření prostředí pro klienty. Oproti vlastnímu sestavení celé serverové části LTSP nabízí funkční základ s možností soustředění se pouze na funkci klientů, a to právě díky možnosti úpravy skriptů, kterými lze libovolně měnit předkonfigurované chování klientů, ať už se jedná o samotné zavedení systému nebo o jeho následný běh.

2.4.4 Operační systém

Operační systém bude stejný jako u samostatného klienta, tedy Ubuntu 14.04 LTS, v tomto případě ale v 64-bitové verzi.

32-bitová verze u samostatného klienta byla použita z možnosti využití staršího hardwaru, který nemusí podporovat 64-bitové instrukce. V případě serveru ale takováto situace je velice nepravděpodobná a použití 32-bitového kernelu by omezovalo výkonnost celého serveru. Největším omezením 32-bitové verze je adresace maximálně 2^{32} bitů paměti, tedy 4 GiB adresovatelné paměti. Toto omezení lze obejít použitím procesoru a kernelu s podporou pro *Physical Address Extension* (PAE), díky kterému lze adresovat až 64 GiB s určitým velmi malým výkonnostním úbytkem oproti klasickému 32-bitovému kernelu.

PAE virtuálně rozšiřuje možnosti adresace na 2^{36} bitů, a to pomocí mapování 32-bitové virtuální adresy na 36-bitovou fyzickou adresu. Jeden proces tedy stále nemůže adresovat více jak 4 GiB paměti, ale 2 různé procesy mohou mít namapovány 4 GiB zcela různé paměti (více v [64]).

I přesto, že Ubuntu poskytuje všechny kernely s PAE podporou, je 64-bitový kernel výhodnější, a to hlavně proto, že při stejné konfiguraci je podle testu [65] výkonnost 64-bitového kernelu znatelně vyšší, než u 32-bitového kernelu.

2.4.5 Dohledový systém

Pro dohledový systém by byl ideální některý zdarma dostupný a jednoduchý dohledový systém, který by hlídal pouze odezvu samotných počítačů a v případě nedostupnosti informoval správce. U takovýchto počítačů by sledování jiných vlastností nemělo vysoký význam a pouze by zvyšovalo síťovou zátěž, na druhou stranu v případě potřeby či specifických požadavků (například sledování i jiných stanic nebo serverů), by systém měl také umožňovat rozšíření

na komplexní dohledový systém pro dohled nad libovolnými službami a vlastnostmi daných počítačů.

Výše požadované vlastnosti splňuje open-source systém Nagios[66], který pomocí konfiguračních souborů určuje, jaké parametry cílových strojů budou sledovány. Základní sledování dostupnosti samotného stroje lze rozšířit na kontrolu jakýchkoliv parametrů a to pomocí více jak 3500 stahovatelných pluginů ze stránky [67], popřípadě vytvořením vlastního pluginu.

Nagios

Nagios je dohledový systém, který kontroluje stav hostitelů a různých služeb, které na těchto strojích běží. Hlavním účelem takového systému je nalezení a oznámení chyby administrátorovy dříve, než na tuto chybu narazí běžný uživatel. Nagios jako takový ale žádné kontroly služeb neprovádí. Tyto kontroly provádí pluginy, které hlásí stav kontroly a Nagios tato hlášení pouze zpracuje, popřípadě nahlásí problém, pokud plugin narazil na chybu u hostitele nebo služby.

Pluginy do takového systému mohou být napsány v kterémkoliv programovacím jazyce, stačí aby jejich výstupem byl příznak stavu služby (OK, WARNING, CRITICAL nebo UNKNOWN) a doprovodné informace o stavu, které se ale nijak nezpracovávají a slouží především jako informace pro administrátory. Výstupem také může být číselná hodnota (například procento volného disku nebo teplota procesoru), kde Nagios nastaví odpovídající stavový kód podle zadaného rozsahu hodnot u těchto stavů. Tímto řešením se stává Nagios neuvěřitelně škálovatelný a pomocí pluginů může kontrolovat prakticky jakoukoliv činnost serverů.

Když už se některý počítač nebo jeho služba stane nedostupná, Nagios rozlišuje tzv. Soft a Hard stav. Při první detekci chyby se přepne tato služba do Soft stavu a zvýší se frekvence kontrol. Pokud se služba do určeného počtu pokusů stane znovu dostupná, Nagios nenahlásí žádnou chybu. Pokud se ale služba nezpřístupní, přepne se do Hard stavu a upozorní e-mailem administrátora, že daná služba se stala nedostupná.

Tento systém dokáže pracovat ve dvou vzájemně kombinovatelných režimech, a to aktivní a pasivní kontrola. Při aktivní kontrole si Nagios sám odesílá požadavky na zkontrolování pluginem, zatímco při pasivním režimu pluginy samy odesílají informace bez vyžádání, například v nepravidelných intervalech, když zjistí, že některá služba neběží (v případě, že služba běží, neposílají se žádné informace).

2.4.6 Hromadná správa klientů

Hromadná správa klientů je v případě bezdiskových klientů již integrována implicitně v principu fungování. Všichni klienti načítají jediný obraz systému v podobě blokového zařízení a pokud je třeba udělat nějaké změny v sys-

tému, provedou se v daném blokovém zařízení. Po této změně pak automaticky všechny klientské stroje budou načítat aktualizovaný systém a není třeba provádět žádné jednotlivé správcovské úkony na jednotlivých klientech. Nějaký další program pro hromadnou správu klientů je zde tedy zbytečný a byl by vhodný pouze při použití s klienty obsahující vlastní lokální systém.

Kapitola 3

Realizace zabezpečeného systému

3.1 Nastavení společné pro samostatného klienta i server/klient architekturu

Většina aplikační logiky je stejná jak u samostatného klienta tak i u serveru pro tenkého klienta. Liší se hlavně v nastavení operačního systému, ale i zde je několik společných prvků.

3.1.1 Nastavení operačního systému

3.1.1.1 Práva složky pro stažené soubory prohlížečem

V prohlížeči Chromium je jako implicitní složka pro stahování nastavena cesta `/$HOME/Downloads`. Přestože uživatel nemá ke složce možnost přístupu skrze prohlížeč, samotný Chromium má stále práva do této složky stahovat soubory a to například díky internímu PDF prohlížeči, který obsahuje funkci na stažení souboru. Z toho důvodu jsem majitele složky `/$HOME/Downloads` změnil na uživatele root, díky čemu do této složky uživatel ztratí kompletní přístup. Při pokusu o stahování Chromium generuje uživatelsky přívětivou hlášku, že stahování je zakázáno. Na druhou stranu se ale v logu začne objevovat hlášení systému o nepřístupné složce.

3.1.1.2 Zakázání všech nepotřebných virtuálních terminálů

Zakázání přepínání mezi terminály lze buď pomocí editací konfigurace `/etc/X11/xorg.conf`, který zakáže klávesové zkratku pro přepínaní, nebo pomocí odstranění samotných virtuálních konzolí pomocí editace souboru `/etc/sysconfig/init` a povolení pouze jediného s grafickým prostředím. Ideálním řešením je provést obě editace, při povolení pouze přepínání by zde sice

žádný terminál neběžel, ale uživatel by stále měl možnost přepnout do prázdného prostředí.

3.1.1.3 Zabezpečení administrátorského účtu

Součástí systému je i administrátorský účet, který je implicitně zabezpečen heslem. Účet lze i dále zabezpečit pomocí zakázání přihlašování k tomuto účtu z neprivilegovaného účtu. Toho lze docílit využitím *Pluggable authentication module* (PAM), což je modulární framework určený pro definování pravidel pro autentizaci. Pro docílení požadovaného chování je třeba editovat soubor echo `/etc/pam.d/su` a přidat řádku *auth required pam_wheel.so use_uid*. Při následném pokusu uživatele o spuštění nástroje `su` pro přepnutí na jiný uživatelský účet je zkontrolováno, zda je daný uživatel součástí skupiny *wheel* (pokud tato skupina neexistuje, je kontrolována skupina root, viz [68]). Pokud uživatel není součástí skupiny, je mu přístup zamítnut s chybovým hlášením *su: Permission denied*.

Pro povolení využívání stejného nástroje administrátorem stačí uživatelský účet přidat do dané skupiny.

3.1.1.4 Zakázání USB flash pamětí

Zakázání všech USB flash pamětí lze provézt editací souboru `/etc/modprobe.d/blacklist.conf` a přidáním klíčového výrazu *blacklist usb-storage*.

3.1.1.5 Zakázání SysRq funkcí

Jádro operačního sytému se rozhoduje o přijímání různých SysRq příkazů podle masky definované v souboru `/etc/sysctl.d/10-magic-sysrq.conf`. Pro úplné zakázání všech SysRq příkazů je zde nastavení *kernel.sysrq = 0*, pro povolení například pouze nouzového vypnutí je nutné nastavit *kernel.sysrq = 128*.

3.1.2 Použité programy a jejich nastavení

Kromě standardních součástí Ubuntu distribuce byly pro realizaci zabezpečeného systému použity následující balíky (a jejich závislosti):

- `aufs-tools 3.2` – Implementace UnionFS souborového systému dovolující spojení více různých existujících souborových systémů do jednoho připojeného adresáře
- `chromium-browser 41.0.2272.76` – Webový prohlížeč Chromium
- `language-pack-cs 14.04` – Podpora pro český jazyk a klávesnici
- `nodm 0.11-1.3` – Jednoduchý display manager

- `quota 4.01-3` – Nástroj pro omezení obsazeného místa v souborovém systému
- `rsync 3.1.0-2` – Nástroj, který slouží k synchronizaci obsahu souborů a adresářů mezi dvěma různými umístěními
- `xautolock 2.2-4` – Nástroj pro spuštění libovolného skriptu v případě neaktivity uživatele
- `xorg 7.7` – Implementace X Window System, která umožňuje vytvořit v operačním systému grafické uživatelské rozhraní (GUI).

3.1.2.1 Skript pro ukončení relace při nečinnost

Ve skriptu pro nečinnost A.1 může být pouze příkaz pro ukončení všech procesů se jménem *chromium*. Tímto způsobem se ale bude každé 2 minuty zbytečně restartovat prohlížeč, a to i když nevznikla žádná aktivita ze strany uživatele. Z toho důvodu je ve skriptu hlídáno, jak dlouhá doba uplynula od posledního vyvolání nečinnosti. Pokud tato doba byla 2 minuty, od poslední kontroly nevznikla žádná aktivita a je tedy zbytečné prohlížeč restartovat.

3.1.2.2 Zabezpečení prostředí prohlížeče pomocí politik

Pomocí politik je vynuceno nastavení všech součástí prohlížeče, jedná se například o použití proxy, ukládání oblíbených položek, cookies, hesel atd. Všechny nastavené politiky jsou k nalezení v příloze A.2. Za zmínku zde stojí hlavně direktivy *URLBlacklist*, *URLWhitelist* a *EnabledPlugins*.

URLBlacklist

Pro případ zabezpečeného systému, je třeba blokovat všechny stránky (kromě několika výjimek), použije se zde tedy právě nastavení * , které bude odpovídat všem webům, včetně lokálního filesystému a nastavení prohlížeče.

URLWhitelist

Při zablokování všech stránek se zde specifikují povolené stránky, ať už se jedná o celé domény, subdomény, konkrétní adresy nebo například i protokoly. Zde je třeba také povolit všechny domén, ke kterým vedou odkazy z hlavní povolené domény. Pokud například webová stránka využívá *maps.google.com* pro zobrazení polohy pobočky, je třeba i tuto doménu explicitně povolit, jinak se tato informace v prohlížeči nebude zobrazovat.

Tabulka 3.1: Minimální systémové požadavky

CPU	Intel Pentium 4 nebo novější
RAM	512 MiB
GPU	podpora rozlišení alespoň 800x600
HDD	5 GB

EnabledPlugins

Pro tento systém je povolen pouze interní PDF prohlížeč, protože stránky mohou obsahovat PDF dokumenty, které by jinak uživatel nemohl otevřít. Java i Flash zůstanou zakázány, protože mohou představovat určité bezpečnostní riziko a navíc v drtivé většině stránek nejsou ke správné funkci potřeba.

3.2 Samostatný klient

3.2.1 Nastavení operačního systému

Systém je postaven na 32-bitové verzi Ubuntu 14.04 LTS, a to konkrétně na minimálním CD, které si všechny potřebné balíky stahuje z repositáře při instalaci. Tímto způsobem se nainstalují pouze potřebné balíky, které navíc budou vždy při instalaci aktuální. Nevýhodou je potřeba internetového připojení při instalaci.

3.2.1.1 Systémové požadavky

Nejnáročnější součástí systému je prohlížeč Chromium, minimální požadavky ale vývojáři nespecifikovali. Tyto požadavky se dají přibližně odhadnout z požadavků prohlížeče Google Chrome, jehož zdrojový kód vychází právě z prohlížeče Chromium a který minimální požadavky uvádí (viz. [69]).

Kombinací těchto aplikací vychází minimální požadavky přibližně tak, jak jsou uvedeny v tabulce 3.1.

3.2.1.2 Rozdělení disku

Disk je rozdělen na 3 oddíly – kořenový adresář (připojen do `/`), odkládací oddíl (připojen do `/swap`) a oddíl, který bude sloužit jako zapisovatelné dočasné umístění (připojen do `/tempFiles`). Od doporučeného rozdělení disku se tato strategie liší z důvodu vyšší bezpečnosti, kde uživatel bude mít přístup pouze do jediného zapisovatelného oddílu.

Kořenový adresář obsahuje kompletní nainstalovaný systém se všemi soubory. Odkládací oddíl slouží pouze na odkládání souborů z RAM paměti v případě nedostatku její kapacity. Poslední oddíl slouží pro běžného uživatele, kam se budou ukládat dočasné soubory, které uživatel svojí činností vytvoří.

Poměry velikostí jednotlivých oddílů jsou přibližně 70% pro kořenový adresář, 10% pro odkládací prostor (minimálně ve stejné velikosti, jako je paměť RAM) a zbytek (20%) pro zapisovatelný dočasný prostor.

3.2.1.3 Zabezpečení bootovacího procesu

Ubuntu používá jako zavaděč GNU GRUB, který podle zadaných parametrů bootuje systém. Tyto parametry jdou změnit před samotným procesem bootování a lze například vynutit načtení módu jediného privilegovaného uživatele a získat neomezený přístup k systému.

Mód jednoho uživatele je dokonce přímo přednastaven při výběru systému v bootovací tabulce pod položkou Recovery. Tento záznam lze odstranit pomocí editace souboru `/etc/default/grub` a to konkrétně direktivou *GRUB_DISABLE_RECOVERY="true"*.

Změna bootovacích parametrů lze zabezpečit pomocí libovolného uživatelského jména a hesla, které nemusejí korespondovat s uživateli v systému. Docílí se toho pomocí editace souboru `/etc/grub.d/00_header` přidáním:

```
set superusers="user"
password_pbkdf2 user password
```

kde *password* je pbkdf2 otisk použitého hesla. Následně při pokusu o změnu bootovacích parametrů bude uživatel vždy dotázán na uživatelské jméno a heslo, jinak změna bude zamítnuta.

3.2.1.4 Omezený uživatelský účet

Jako neprivilegovaný uživatel je vytvořen uživatel *guest*, který nemá heslo, jeho interpret je nastaven na `/usr/sbin/nologin` (tedy má zakázané přihlášení) a je členem pouze skupiny *video* pro přístup k grafickému prostředí.

Součástí startup skriptu (viz 3.2.2.1) je ale příkaz `rsync` s kopírováním souborů, která vyžadují zvýšená oprávnění. Uživatel byl tedy přidán do souboru `/etc/sudoers`, který mu dovoluje provádět pouze jediný konkrétní příkaz `rsync` bez zadání hesla, na všechny ostatní příkazy omezená práva zůstávají.

3.2.1.5 Automatické přihlášení

Nodm podporuje automatické přihlášení uživatele při startu systému, při testování se ale ukázalo toto přihlašování jako nespolehlivé. V některých situacích nedošlo k automatickému přihlášení i při nastaveném interpretu na standardní `/bin/bash` a proto je automatické přihlášení řešené pomocí skriptu `/etc/rc.local`, který je spouštěn vždy na konci zavádění systému u každého víceuživatelského módu (více o souboru `rc.local` v [70]).

Automatické přihlášení je v tomto skriptu realizováno pomocí spuštění grafického prostředí za uživatele *guest* s nastaveným správným interpretem pro

tuto jednu relaci. Při úspěšném pokusu o ukončení této relace je celý počítač z bezpečnostního důvodu restartován. Obsah celého skriptu je v příloze A.6.

3.2.1.6 Zabezpečení domovské složky

Pod uživatelem *guest* je spuštěn webový prohlížeč, díky čemuž se vytvoří implicitní profil, historie prohlížení a další soubory používané pro identifikaci uživatele prohlížeče a jeho nastavení.

Pomocí privilegovaného účtu je následně celý domovský adresář přesunut do složky `/bkupRO/`, kde bude sloužit jako výchozí domovský profil, který bude dále určen pouze pro čtení. Jako zapisovatelná část domovského adresáře bude sloužit složka `/tempFiles/home/` na oddílu pro dočasné soubory. Nakonec jsou tyto dva adresáře spojeny dohromady do původního umístění v `/home` pomocí nástroje `AuFS` s parametry *noexec,nosuid,nodev*, které zakazují v daném umístění spouštění programů, vytváření speciálních souborů a používání suid a sgid bitů.

V aktuální verzi operačního systému je ale připojení filesystémů prováděno paralelně a tedy v nedeterministickém pořadí. Z toho důvodu nelze připojení AuFS řešit automaticky při startu systému, protože adresář `/tempFiles`, na kterém tyto filesystémy závisí, nemusí být ještě připojen. Připojení těchto částí je řešeno až po samotném startu systému pomocí souboru `/etc/rc.local`. Posloupnost příkazu je v příloze A.4.

3.2.1.7 Nastavení kořenové složky pouze pro čtení

Nastavení kořenové složky pouze pro čtení při startu počítače pomocí souboru `/etc/fstab` není zcela doporučováno, proto zůstává při startu kořenový adresář i s možností zápisu, ale ihned po načtení systému je připojen pouze pro čtení pomocí příkazu ve skriptu `/etc/rc.local` před samotným spuštěním relace s prohlížečem.

Pro vytvoření celé kořenové složky jako pouze pro čtení, je třeba některé složky či soubory zduplikovat do zapisovatelného oddílu – pokud by zůstaly jako nezapisovatelné, některé části systému by mohly selhat.

`/etc/resolv.conf`

Soubor `/etc/resolv.conf` se používá pro ukládání informací od DHCP serveru, konkrétně informace o DNS resolverech (pokud systém nepoužívá statickou konfiguraci). Tento soubor je zkopírován do zapisovatelné částí systému a následně vytvořen symbolický odkaz do původního umístění. Tímto způsobem se bude při pokusu o zápis do `/etc/resolv.conf` fyzicky zapisovat do zapisovatelného prostoru.

`/var`

Adresář `/var` obsahuje často měněné soubory jako jsou zálohy klíčových systémových souborů, cache paměť aplikací, logy a tak podobně. Záloha tohoto umístění je provedena úplně stejně jako zabezpečení domovské složky v sekci 3.2.1.6, kdy se vytvoří záloha původního adresáře pouze pro čtení spojená s dočasnou zapisovatelnou částí, která se následně při ukončení prohlížeče promazává. Pokud chceme ale uchovávat logy z předchozích spuštěních, musí být ze synchronizace vynechána složka `/var/log`, jinak by všechny logy o činnosti systému a aplikací byly vždy smazány. Následné promazávání této složky může probíhat buď ručně administrátorem nebo automaticky za určitou časovou periodu pomocí plánovače `cron`.

3.2.1.8 Nastavení kvót na disku

Kvóty na obsazený prostor jsou nastavovány pouze na zapisovatelném disku, na ostatních oddílech by neměly žádný význam. V případě nějakého problému (například uživatel nějakým způsobem začne plnit svůj uživatelský prostor daty), je uživateli znemožněno dále vytvářet soubory, zatímco systémové soubory mohou stále dále vznikat. Soft i hard limit na velikost obsazeného prostoru je nastaven na 100 MiB a celkový počet vytvořených souborů na 500, které normálním používáním nejsou dosažitelná. Pokud tento limit je dosažen, je jasné, že došlo k nějakému incidentu a vytváření odlišného soft limitu tedy není nezbytné.

3.2.2 Nastavení programů

3.2.2.1 Úprava startup skriptu grafického prostředí

Startup skriptem grafického prostředí je soubor `.xsession`, který definuje a ovládá celou grafickou relaci.

Před samotným spuštěním grafického prostředí jsou ve skriptu příkazy k úpravě vzhledu prostředí, je zde příkaz pro změnu kurzoru z implicitního křížku na klasického ukazatele, načtení českého prostředí a české klávesnice nebo zakázání vypínání obrazovky a tlačítka F1 pro zobrazení nápovědy.

V hlavní ovládací smyčce je dále promazávání zapisovatelné části systému před startem prohlížeče. Promazávání pouze při startu prohlížeče nemusí být ale dostatečné (systém může běžet třeba celý den i více), proto je zde dále využit nástroj `xautolock`, který při detekování nečinnosti klienta spustí skript pro vynucené vypnutí prohlížeče, čímž se automaticky vyvolá i promazání profilu. Obsah skriptu pro vypnutí prohlížeče je dále diskutován v 3.1.2.1.

Vynucené vypnutí prohlížeče může vyvolat v prohlížeči obnovení poslední relace, které se aktivuje podle hodnoty *exited_cleanly* a *exit_type* v konfiguračním souboru. Přestože tato nastavení by se díky mazání dočasného profilu

neměla nikdy zachovat, pro větší jistotu tyto hodnoty pomocí nástroje sed vždy nahrazuji za hodnoty zapisované při korektním ukončení.

Obsah celého skriptu je v příloze A.5

3.2.2.2 Firewall

Firewall by měl povolovat pouze nutné porty, všechny ostatní porty by měly být zakázané. Z toho důvodu zůstávají směrem ven otevřené porty pro HTTP, HTTPS, SSH a DNS. Dále jsou povoleny pakety *ICMP reply* a *ICMP request*, všechny ostatní pakety od uživatele *guest* jsou odmítnuty se zprávou *ICMP Host Prohibited*. Ostatní pakety jsou zahazovány.

Příchozí provoz je filtrován tak, aby byly povoleny pakety *ICMP reply* a *ICMP request* a aby systém dále přijímal pouze SSH provoz a provoz, který byl již dříve navázán. Není tedy možné, aby přes firewall prošel paket inicializující nové spojení (pokud se nejedná o SSH připojení). Konkrétní příkazy jsou v příloze A.3

Aby se tyto pravidla aplikovaly při každém spuštění, jsou uloženy v souboru `/etc/iptables.rules`. Následně jsou pravidla načítána při inicializaci síťového rozhraní pomocí úpravy souboru `/etc/network/interfaces` přidáním příkazu *pre-up iptables-restore < /etc/iptables.rules*.

3.3 Bezdiskový tenký klient a server

3.3.1 Nastavení operačního systému

Operační systém vychází z 64-bitové verzi Ubuntu 14.04 LTS rozšířený o balík ltsp-server-standalone 5.5.1-1 starající se o funkčnost LTSP serveru pro bezdiskové tenké klienty.

3.3.1.1 Systémové požadavky

Hardwarové požadavky LTSP se liší podle předpokládaného využití. Zatím co pro klienta jsou stále stejné, server je nutné škálovat podle počtu připojených klientů.

Server

Požadavky na server se liší nejenom počtem připojených klientů, ale také i způsobem použití klientů. Při klasickém prohlížení statických webových stránek budou nižší nároky než při používání náročných webových Java appletů nebo prohlížení stránek s velkým počtem Flashových animací. Průměrné požadavky takového serveru podle [71] jsou:

- Procesor – V případě používání náročnějších operací, 2-3 GHz jednojádrový procesor dokáže obsloužit přibližně 20 klientů. U moderních dvou

a více jádrových procesorů lze bez potíží obsloužit 30 klientů, pokud by bylo ale třeba ještě více klientů, je možnost využití více LTSP serverů a rozdělit mezi ně zátěž.

- Paměť – Nutná paměť v MiB se vypočítá podle vzorce $1500 + 300 * N$, kde N je počet klientů; například pro 20 klientů by to bylo celkem 7500 MiB paměti na celý server.

- Síťová karta – Pro více než 20 uživatelů je doporučen Gigabit Ethernet. Přestože normální vytížením jedním klientem se pohybuje v rozmezí 0,5 — 2 Mbps, maxima při náročném obsahu mohou být poměrně vysoká (až 70 Mbps). Zároveň je doporučeno použití dvou NIC, jeden adaptér pro vnitřní síť a druhý pro připojení do Internetu.

- Pevný disk – Doporučeno použití RAID systému, ideálně RAID 1+0[13] pro zvýšení jak bezpečnosti dat, tak rychlosti přístupu.

Klient

Tenký klient lze provozovat na téměř jakémkoliv stroji s hardwarem až 15 let starým[14], hardwarové požadavky jsou minimální a splňují požadavek na znovu využití dosluhujícího hardwaru. Konkrétní požadavky podle [71] jsou:

- Procesor – Jakýkoliv CPU s frekvencí 533 MHz a vyšší.

- Paměť – 128 MiB, ideálně 256 MiB paměti.

- Síťová karta – Nutná podpora PXE nebo softwarová náhrada funkce například pomocí Etherboot/gPXE.

- Grafická karta – Jakákoliv s alespoň 16 MiB velkou pamětí.

- Pevný disk – žádný.

3.3.1.2 IP forwarding

Ubuntu implicitně nemá z bezpečnostních důvodů u kernelu povolený IP forwarding. V tomto případě je ale nutné IP forwarding na serveru povolit, jinak

[13] RAID 1+0 je kombinací diskových polí RAID 1 (zrcadlení) a RAID 0 (prokládání). Jedná se o vytvoření RAID 0 pole nad dvěma a více RAID 1 poli, kde počet disků musí být sudý a obsahovat alespoň 4 disky. Ve většině případů je takovéto pole rychlejší jak všechny ostatní RAID pole (s výjimkou RAID 0, které má vyšší propustnost). Jedná se tedy o ideální RAID pole pro využití v aplikacích s vysokým počtením I/O operací jako je web server nebo databáze. Pole dokáže být stále funkční bez ztráty dat i při výpadku více disků (pokud nedojde k výpadku všech disků v jedné zrcadlené sadě). Více v [72].

[14] Společnost Intel Corporation vydala v říjnu 1999 procesor Pentium III s kódovým označením Coppermine a frekvencemi v rozmezí 500 — 733 MHz

klienti nebudou mít přístup do internetu a mohli by přistupovat pouze k serveru. Pokud by například na stejném serveru přímo běžel webový server se stránkami, kam se klient chce připojit, není IP forwarding nutný. V ostatních přepadech je ale IP forwarding nutný a je třeba ho povolit.

Trvalé povolení pro protokol IPv4 se provede přidáním řádky `net.ipv4.ip_forward = 1` do souboru `/etc/sysctl.conf`.

3.3.2 LTSP prostředí

LTSP prostředí je vytvořeno pomocí programu ltsp-build-client, které vytváří chroot prostředí pro 32-bitové klienty v implicitní složce `/opt/ltsp/i386`. Je možné také nainstalovat další architektury a to pomocí přepínače *ltsp-build-client –arch=x*, kde *x* je kódové označení architektury. V mém příkladě budu předpokládat, že se zde vyskytují pouze stanice s 32-bitovou architekturou.

3.3.2.1 Nastavení klientského prostředí

Předpřipravený systém je upraven podle sekce 3.1. Úpravy zahrnují nastavení systému, vytvoření skriptů a doinstalování daných programů.

3.3.2.2 Ovládání klientské relace

Pro specifikaci spuštění kiosk relace slouží soubor `/var/lib/tftpboot/ltsp/i386/lts.conf`. V tomto konfiguračním souboru je určeno pro všechny klienty automatické přihlášení, nodm jako display manager, spuštění kiosk relace na sedmém virtuálním terminálu a zakázání ostatních virtuálních terminálů. Obsah souboru je k nalezení v příloze A.7.

Nastavení relace je následně ovládána skriptem umístěným v chrootovském prostředí `/usr/share/ltsp/screen.d/kiosk`. Tento skript byl editován a z části doplněn, jedná se hlavně o použití jiného webového prohlížeče a přenastavení práv složce Downloads pro dočasného uživatele (viz sekce 3.1.1.1). Další funkčnost jako vytvoření uživatele s domovskou složkou v dočasném adresáři `/tmp` a spuštění relace pod dočasným uživatelem zůstala zachována. Obsah celého skriptu je v příloze A.8.

Samotné spuštění relace je zařizováno skriptem `/usr/share/ltsp/kioskSession`. Jedná se o určitou náhradu souboru `.xsession`, obsah tohoto souboru je tedy velmi podobný jako u samostatného klienta v sekci 3.2.2.1. Hlavním rozdílem je zde mazání celé domovské složky (kromě složky `Downloads`) místo kopírování implicitního profilu. Původní skript obsahuje i možnost určení automaticky spuštěných programů na pozadí při startu relace. Tato funkčnost není v mém sestavení využitá, přesto jsem je ve skriptu zanecha,l a to z důvodu jednodušší úpravy chování bez nutnosti editovat přímo spouštěcí skript. Celý skript je v příloze A.9.

3.3.3 Nastavení programů

Pro funkčnost samotného serveru a jeho poskytovaných služeb jsou použity programy:

- `nagios3 3.5.1` – Dohledový systém nad klienty
- `isc-dhcp-server 4.2.4-7` – DHCP server
- `openssh-server 6.6p1-2` – SSH server
- `dnsmasq 2.68-1` – TFTP a DNS server

3.3.3.1 Dohledový systém Nagios

Hlavní konfigurační soubor Nagiosu `nagios.cfg` umístěný implicitně v `/etc/nagios` obsahuje cestu k dalším konfiguračním souborům, popřípadě celým složkám. Dále se zde nachází UID a GID uživatele, pod kterým běží Nagios, nastavení logování nebo povolení externích příkazů pro externí příkazový soubor. Tento konfigurační soubor obsahuje ještě mnoho dalších parametrů, tyto jsou ale ty nejdůležitější pro moji konfiguraci.

Nagios rozlišuje v konfiguračních souborech několik druhů základních definovatelných příkazů:

- **command** – Zde se definují libovolné příkazy, které lze poté použít v systému Nagios pro dohled nad systémem.
- **host** – Zde jsou definovány šablony, které pak jednotlivé konfigurační soubory používají pro zkrácení délky. Mohou zde být definovány jak šablony pro kontakty, tak i pro služby a servery. Kromě šablon je možné definovat i přímo jednotlivé stroje, šablony pouze šetří práce s definováním jednotlivých strojů.
- **contact** – Zde jsou definováni uživatelé, kterým chodí upozornění, pokud je některá služba nebo server nedostupný.
- **contactgroup** – Obsahuje definování skupin, do kterých patří příslušní uživatelé dohledového systému.
- **hostgroup** – Tato direktiva obsahuje definice skupin serverů pro seskupení a jednodušší správu.

Ukázka nastavení konfiguračních souborů Nagiosu pro dva LTSP kiosky s IP adresami `192.168.56.101` a `192.168.56.102` je v příloze A.10.

3.3.3.2 Firewall

Firewall je pro server nastaven stejně jako pro samostatného klienta v sekci 3.2.2.2. Rozdíl je ale v nastavení firewallu na rozhraní pro klienty. Na tomto rozhraní je povolen veškerý příchozí i odchozí provoz. Zároveň jsou povoleno všechny přeposlané pakety od klientů přes server, všechny navázané přeposlané pakety v opačném směru a přesměrovávání IP paketů v NAT tabulce pravidel z rozsahu adres klientů. Konkrétní příkazy jsou v příloze A.11.

3.3.3.3 DHCP server

Nastavení DHCP serveru záleží na stupni bezpečnosti, který požadujeme. V případě jednodušší a méně bezpečné konfigurace bude DHCP server přidělovat IP adresy všem strojům, které si zažádají.

Z bezpečnostního hlediska může tato konfigurace vytvářet určité riziko, útočník může počítat vypojit a připojit vlastní zařízení a ihned získat IP adresu a přístup do sítě. Tato konfigurace není vhodná i kvůli dohledovému systému, který jednotlivé počítače rozlišuje na základě IP adresy.

Z těchto důvodů je DHCP server nastaven na přidělování IP adresy pouze klientům definovaným v hlavním konfiguračním souboru `/etc/ltsp/dhcpd.conf` pomocí MAC adresy jejich síťového rozhraní. Ostatním klientům DHCP server na požadavky nebude odpovídat. Zároveň DHCP server naslouchá pouze na rozhraní interní sítě a případné požadavky z externí sítě jsou ignorovány.

Ukázková konfigurace s 2 definovanými klienty a rozsahem adres pro všechny klienty `192.168.56.0/24` je v příloze A.12.

3.4 Skript pro automatické nastavení

Pro jednodušší nastavení celého systému byl napsán Bash skript pro oba typy systému, který vše nastavuje sám. Tento skript předpokládá, že uživatel má znalosti administrátora a vyžaduje jeho interakci v několika bodech (například domény blokovaných webů, uživatelská jména atp.). Skript také obsahuje v určitých případech kontrolu vstupů a možné řešení nevalidního zadání, popřípadě jeho opravu uživatelem. Uživatel je také dotazován, zda chce nainstalovat všechny programy, či zda některé nekritické součásti nemají být nainstalovány.

Samostatný klient

Skript `standaloneKiosk.sh` vyžaduje pro správnou funkčnost rozdělení disku podle sekce 3.2.1.2, funkční síťové připojení do Internetu a operační systém Ubuntu. Kromě instalace programů z Ubuntu repozitáře byly ale pro vytvoření skriptu využity pouze GNU nástroje, skript by měl být tedy kompatibilní i

s ostatními GNU Linuxovými distribucemi využívající APT balíčkovací systém s příslušnými programy v repozitářích.

Při instalaci je uživatel dotazován na použití hesla pro GRUB, jméno omezeného účtu a účtu administrátora, rozlišení obrazovky, domovskou stránku prohlížeče, seznam povolených domén, interface pro SSH server, použití kvót na disku a vypnutí funkčnosti SysRq.

Server pro tenké klienty

Skript pro server využívá dva konfigurační soubory v jednom umístění – `serverLTSP.sh` pro nastavení samotného serveru a `chrootLTSP.sh` pro nastavení prostředí pro klienty. Oproti samostatnému klientovi jsou pro funkčnost skriptů nutné alespoň dvě síťová rozhraní. Skript také nekonfiguruje Nagios a povolené počítače pro DHCP, pouze založí ukázkový konfigurační soubor (podobný A.10), podle kterého si uživatel doplní opravdovou konfiguraci, pokud má o tyto funkce zájem.

Při provádění skriptu `serverLTSP.sh` je uživatel dotazován pouze na nastavení DHCP serveru a rozhraní připojení do Internetu. Součástí je i spuštění skriptu `chrootLTSP.sh`, který se uživatele ptá na domovskou stránku prohlížeče a seznam povolených domén. Pokud uživatel zvolí instalaci i dohledového systému, na konci instalace je uživateli sděleno, jak tento systém nakonfigurovat včetně umístění souboru, kde je ukázková konfigurace.

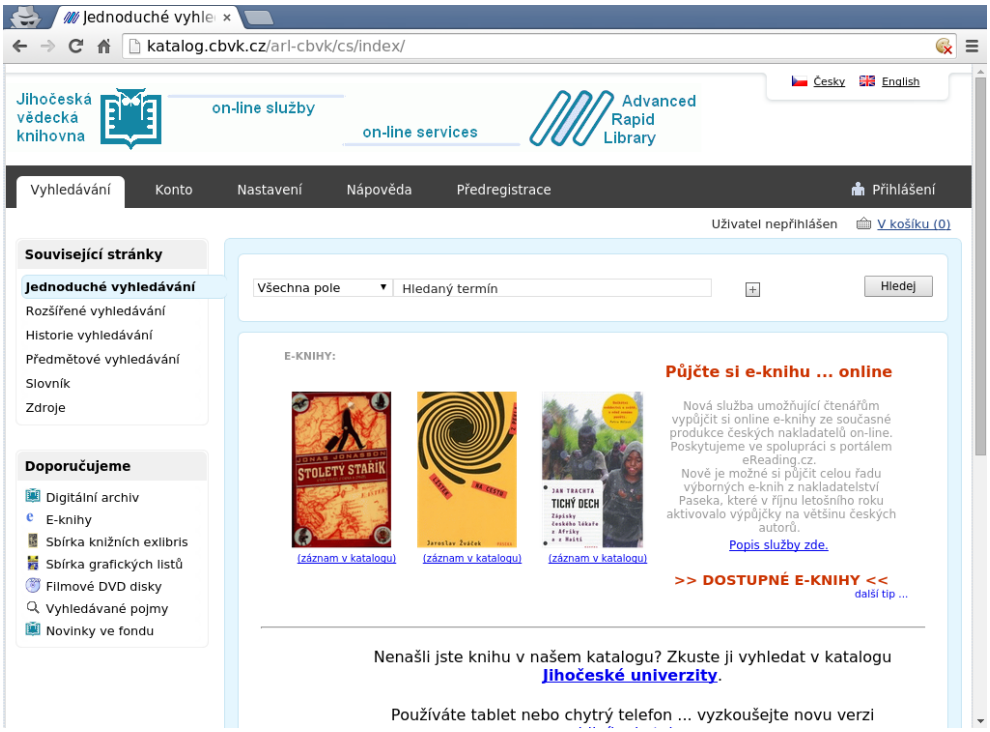
Všechny skripty jsou k nalezení na přiloženém CD.

3.5 Výsledný systém z pohledu uživatele

Po zapnutí počítače se uživateli zobrazí úvodní obrazovka operačního systému, kdy po načtení systému dojde k automatickému spuštění prohlížeče na celé obrazovce. Prostředí prohlížeče je zcela klasické (viz snímek prohlížeče 3.1), které používá Chrome i Chromium, práce s tímto prostředím se tedy nijak nemění oproti tomu, jak může být uživatel zvyklý z vlastního systému. Jediným rozdílem jsou některé položky v nabídce prohlížeče, které jsou buď zcela nepřístupné nebo jejich otevření vede na zablokovanou stránku (jedná se například o přístup k záložkám, historii, nastavení, nástrojům pro vývojáře atd.). Blokovaných funkcí prohlížeče je většina, neblokované funkce jsou zejména funkce pro práci s textem (vyhledávání, kopírování, vkládání, lupa) a pro usnadnění práce s prohlížečem.

Součástí prohlížeče je také vestavěná podpora pro prohlížení PDF souborů (viz. sekce 3.1.2.2), ale znovu s omezenou funkčností v podobě zakázání tisku a ukládání daných souborů.

Práce se samotným oknem prohlížeče je omezena na vypnutí prohlížeče pomocí ikony křížku nebo klávesové zkratky, ostatní funkce jako přesouvání nebo minimalizování okna nejsou podporovány. Pokud uživatel prohlížeč ukončí,



Obrázek 3.1: Rozhraní prohlížeče se zobrazenou stránkou Jihočeské vědecké knihovny v Českých Budějovicích

uživateli se zobrazí prázdná černá obrazovka a během chvíle je prohlížeč znovu nastartován na domovské stránce. Prohlížeč je také automaticky restartován, pokud uživatel nechal systém 2 minuty v nečinnosti.

Kromě výše zmíněné funkčnosti uživatel nemá přístup k žádným dalším externím programům ani nástrojům a může pracovat pouze v prostředí prohlížeče. Konkrétně které stránky může navštěvovat, je definováno administrátorem při instalaci.

Kapitola 4

Testování systému

Všechna testování probíhala ve virtualizovaném prostředí poskytované softwarem Oracle VM VirtualBox 4.3.20. Tento software mimo jiné dokáže omezit maximální povolené zatížení fyzického procesoru, což ovšem ne zcela přesně emuluje procesor s nižší frekvencí. Při povolení nízkého zatížení se stává systém mnohem méně responsivním a prodlužují se i všechny I/O operace, včetně síťových operací. Jedná se spíše o ochranu proti plnému zatížení hostitelského stroje, přesto ale tuto funkci v rámci testování použiji. Více v [73].

Jako testovací scénář bude použito vytvoření systému pro Jihočeskou vědeckou knihovnu v Českých Budějovicích, konkrétně vytvoření stanic pro prohlížení katalogu knihovny umístěném v doméně *katalog.cbvk.cz*.

Po analýze webových stránek *katalog.cbvk.cz*, které budou sloužit jako domovská stránka, je zjištěno, že kromě domény *cbvk.cz* je nutné povolit také *obalkyknih.cz* (zobrazení náhledů knih), *toc.nkp.cz* (zobrazení obsahů knih), *books.google.cz*, *books.google.com* (náhledy některých digitalizovaných dokumentů) a *mojeid.cz* (přihlášení přes systém MojeID). Součástí jsou i další domény (například náhled knih ve formátu pro čtečky knih), tento obsah ale zůstává schválně blokován, protože se jedná o rozšiřující služby pro domácího uživatele.

Tyto získané údaje jsou následně využity při spuštění automatických skriptů a jsou zadány v okamžiku, kdy je uživatel na tyto informace dotazován. V případě specifických požadavků na funkčnost prohlížeče (například využití proxy serveru), lze následně přímo editovat soubor`/etc/chromium/policies/managed/chrome.json`. V mém případě používám nastavení vytvořené skriptem, které je v příloze A.2.

4.1 Samostatný klient

Pomocí Oracle VM Virtualboxu je systému přiděleno:

- **Procesor** – Maximálně 20% zatížení jednoho jádra procesoru Intel Core i7-2600k taktovaném na 3400 MHz
- **Operační paměť** – 512 MiB
- **Grafická karta** – neakcelerovaná s 8 MiB paměti
- **Síťová karta** – emulovaná Intel PRO/1000 MT Desktop (82540EM) v režimu *NAT*
- **Pevný disk** – 5 GB

4.1.1 Instalace

Po počáteční instalaci operačního systému a rozdělení disku podle 3.2.1.2 je do systému doinstalován balík nazvaný *Přídavky pro hosta*, který je součástí Oracle VM Virtualbox. Součást tohoto balíku jsou ovladače a skripty starající se o kooperaci mezi Oracle VM Virtualboxem a virtualizovaným systémem.

Dále je do systému nakopírován skript `standaloneKiosk.sh`, který je spuštěn se zvýšeným oprávněním (v opačném případě je skript ukončen). V následném instalačním procesu jsou zvolena tato nastavení:

- Zabezpečení GRUB tabulky jménem a heslem při pokusu o změnu
- Uživatelský účet *guest* pro omezený účet
- Uživatelský účet *kiosk* pro administrátorský účet
- Rozlišení obrazovky 1280x1024
- Domovská stránka prohlížeče *katalog.cbvk.cz*
- Použití seznamu povolených domén se stránkami: *cbvk.cz*, *obalkyknih.cz*, *toc.nkp.cz*, *books.google.cz*, *books.google.com*, *mojeid.cz*
- Naslouchání SSH serveru na IP adrese síťové karty
- Nastavení kvót pro oddíl `/tempFiles`
- Omezení funkčnosti SysRq na povolení pouze nouzového restartu a vypnutí

4.1.2 Bootování systému

Před bootováním systému se na jednu vteřinu objeví GRUB tabulka s výběrem systému a následně pokračuje bootování systému. Pokud uživatel před vypršením limitu bootování stiskne některou klávesu, automatické bootování implicitního systému se přeruší a uživatel má na výběr z tabulky. Při pokusu o editaci příkazu či výběru jiné, než implicitní položky, je uživatel dotázán

na uživatelské jméno a heslo. V případě zadání špatných hodnot je uživatel navrácen na výběr systému.

Bootování pokračuje prováděním init skriptů systému na pozadí a po dokončení se ihned spouští grafické rozhraní s prohlížečem se zobrazenou domovskou stránkou běžící pod omezeným účtem *guest*.

4.1.3 Okno prohlížeče

Vzhledem prohlížeče se jedná o klasické rozhraní, které používá Chromium. Při prohlížení stránek na povolených doménách se chová prohlížeč standardně, ale při pokusu o přístup na ostatní domény je zobrazena informace, že stránka je blokována. Stejná informace je zobrazena i při pokusu o zobrazení filesystému pomocí protokolu *file://* a interních stránek prohlížeče (například adresa *chrome://flags* pro nastavení experimentálních funkcí prohlížeče).

Hlavní menu prohlížeče je sice přístupné, většina voleb je zde ale neaktivní či jejich výběr negeneruje žádnou zpětnou odezvu. Kromě funkcí pro práci se schránkou zde zůstávají funkční položky *Zobrazit zdrojový kód* a *Vytvořit zástupce...* v kategorii *Další nástroje*.

Varianta bez použití seznamu povolených domén

V případě povolení přístupu na všechny stránky zůstává stále blokován přístup do filesystému pomocí protokolu *file://* a do interních stránek prohlížeče, ve kterých by mohl eventuálně uživatel měnit nastavení (jedná se hlavně o stránky *chrome://flags* a *chrome://settings*). Všechny ostatní webové stránky jsou pro uživatele přístupné.

4.1.4 Simulace prolomení prostředí

Po zakázání všech možností opuštění prohlížeče, které byly zjištěny, se nepodařilo prohlížeč opustit. Proto bylo následně provedena simulace prolomení určitých ochran, ke kterým by mohlo dojít například využitím zatím neznámé chyby v prohlížeči.

Ukončení grafického prostředí

Ukončení automaticky načteného skriptu po startu systému bylo docíleno úpravou skriptu A.5, konkrétně odstraněním hlavní spouštěcí smyčky.

Po následném startu systému je standardně spuštěn prohlížeč, při jeho zavření dochází ale k ukončení grafického prostředí a okamžitému restartování celého počítače a znovu načtení systému a prohlížeče. Ukončením grafického prostředí uživatel nezískává žádnou další možnost prolomení.

Získání shellu

Shell byl získán pomocí spuštění prohlížeče na pozadí zároveň s grafickým interpretem příkazové řádky.

V tuto chvíli uživatel získává přístup do filesystému pod omezeným účtem s právem zápisu do vlastní domovské složky (kromě složek `Downloads` a skriptu pro startup grafického prostředí) a do adresáře `/tempFiles`.

Vytváření souborů

Ve složkách s právem zápisu může uživatel vytvářet až 500 souborů do celkové velikosti 100 MiB, další soubory nepůjdou díky kvótám na disk vytvářet. Po ukončení relace prohlížeče běžícím na pozadí nebo restartu systému dojde k smazání všech zde vytvořených souborů.

Dále má uživatel práva ke smazání své domovské složky – v takovém případě prohlížeč pouze vytváří novou adresářovou strukturu v daném umístění a systém funguje dál. Po ukončení relace je zpět nahrazena původní adresářová struktura i s obsahem před smazáním uživatelem.

Uživatel nemá práva zápisu do jiných umístění, pokud by ale byla chyba v některém programu, která by umožňovala zápis do jiného umístění i pro neprivilegovaného uživatele, zápis je stále odmítnut s informací, že se jedná o systém pouze pro čtení.

Získání privilegovaného účtu

V případě předpokladu, že uživatel získal shell, uživatelské jméno a heslo privilegovaného uživatele, může se pokusit o přepnutí do tohoto účtu. Jedná se ale o tak závažný bezpečnostní incident, že lze jen těžko mírnit následky. Přesto v některých případech lze zamítnutí přístupu docílit.

Při pokusu na omezeném účtu o spuštění programu `su` pro běh programů pod jiným uživatelským účtem je tento požadavek zamítnutý s chybovou zprávou *su: Permission denied*. Druhou možností je použití příkazu `sudo` pro vykonání jednoho příkazu jako privilegovaný účet. Uživatel ale nemá nastaveno heslo, které `sudo` vyžaduje a zároveň není členem skupiny umožňující použití tohoto příkazu. Vždy tedy dochází k zamítnutí požadavku s chybou o špatném hesle.

Další variantou je využití vzdálené připojení pomocí SSH. Pokud není nastavení SSH omezeno na přijímaní připojení z určité IP adresy, uživatel se může z lokálního počítače přihlásit pod privilegovaném účtu a počítat ovládnout.

Poslední možností je změna GRUB záznamu. Pokud získané uživatelské jméno a heslo souhlasí i pro zabezpečení GRUB tabulky, uživatel může pozměnit bootovací záznam systému pro načtení single módu, který automaticky načítá systém pod privilegovaným účtem nevyžadující heslo. V takovém případě získává uživatel plnou kontrolu nad strojem.

4.2 Tenký klient a server

Pomocí Oracle VM Virtualboxu je přiděleno počítačům:

Tenký klient

- **Procesor** – Maximálně 20% zatížení jednoho jádra procesoru Intel Core i7-2600k taktovaném na 3400 MHz
- **Operační paměť** – 256 MiB
- **Grafická karta** – neakcelerovaná s 8 MiB paměti
- **Síťová karta** – emulovaná Intel PRO/1000 MT Desktop (82540EM) v režimu *Síť pouze s hostem*
- **Pevný disk** – žádný

Server

- **Procesor** – Maximálně 80% zatížení jednoho jádra procesoru Intel Core i7-2600k taktovaném na 3400 MHz
- **Operační paměť** – 1 GiB
- **Grafická karta** – neakcelerovaná s 8 MiB paměti
- **Síťová karta 1** – emulovaná Intel PRO/1000 MT Desktop (82540EM) v režimu *NAT*
- **Síťová karta 2** – emulovaná Intel PRO/1000 MT Server (82545EM) v režimu *Síť pouze s hostem*
- **Pevný disk** – 8 GiB

4.2.1 Instalace

Po instalaci 64 bitového systému Ubuntu 14.04.1, je do systému stejně jako u samostatného klienta doinstalován balík nazvaný *Přídavky pro hosta*. Dále je nakonfigurováno síťové rozhraní se *Sítí pouze s hostem* na statickou adresu 192.168.56.2 s maskou 255.255.255.0, které bude sloužit jako vnitřní síť pro tenké klienty.

Do systému jsou také do jedné složky nakopírovány skripty `serverLTSP.sh` a `chrootLTSP.sh` a následně je skript `serverLTSP.sh` spuštěn se zvýšeným oprávněním (v opačném případě je skript ukončen).

V následném instalačním procesu jsou zvolena tato nastavení:

- Nastavení DHCP pro klienty:
  - podsíť 192.168.56.0
  - maska 255.255.255.0
  - broadcast 192.168.56.255
  - Rozsah přidělovaných adres 192.168.56.101 — 192.168.56.254
  - adresa DNS 192.168.0.1 (v tomto případě se jedná o adresu reálného DNS serveru mimo virtualizované prostředí)
- Určení prvního rozhraní (eth0) jako rozhraní připojené do Internetu
- Domovská stránka prohlížeče *katalog.cbvk.cz*
- Použití seznamu povolených domén se stránkami: *cbvk.cz*, *obalkyknih.cz*, *toc.nkp.cz*, *books.google.cz*, *books.google.com*, *mojeid.cz*
- Omezení funkčnosti SysRq na povolení pouze nouzového restartu a vypnutí
- Instalace dohledového systému Nagios pro dohled nad tenkými klienty

Po instalaci je uživateli sdělena informace jak systém Nagios nakonfigurovat s ukázkovým konfiguračním souborem pro klienty. Nakonec je zde také informace o možném dalším zabezpečení DHCP pro povolení pouze známých klientů.

4.2.2 Bootování systému

Nastavení DHCP serveru je doplněno o zakázání přidělování IP adres neznámým počítačům identifikovaným podle MAC adresy. Zároveň jsou definováni dva klienti a je jim přidělená stálá IP adresy podle přílohy A.12.

Při bootování systému přes síť získává klient ze serveru IP adresu. Pokud MAC adresa síťového rozhraní klienta je v tabulce DHCP serveru, klient získává IP adresu a načítá obraz systému. V opačném případě DHCP neodesílá žádnou odpovědi a bootování je ukončeno s chybovou hláškou o vypršení časového limitu.

Po získání obrazu se načte systém přímo do prostředí webového prohlížeče.

4.2.3 Okno prohlížeče

Pro vytvoření prostředí prohlížeče jsou použity stejné technologie jako u samostatného klienta, testování prostředí probíhá tedy zcela stejně jako v sekci 4.1.3.

4.2.4 Simulace prolomení prostředí

Testování prolomení ochran bylo testováno pomocí úprav hlavního ovládacího skriptu `kioskSession`, kterým se spouští celá relace.

Ukončení grafického prostředí

Ukončení je simulované odstraněním hlavní ovládací smyčky. Po vypnutí webového prohlížeče jsou explicitně smazány všechny soubory v domovské složce uživatele a zároveň jsou ztraceny i všechny ostatní změny, protože byly ukládány pouze v RAM paměti počítače. Následně je počítač restartován do výchozího stavu bez narušení bezpečnosti.

Získání shellu

Shell je získán pomocí spuštění grafického terminálu v hlavním ovládacím skriptu. Omezený uživatel má následně právo zápisu pouze do své domovské složky, po ukončení relace jsou všechny soubory a složky v domovské složce smazány.

Získání privilegovaného účtu

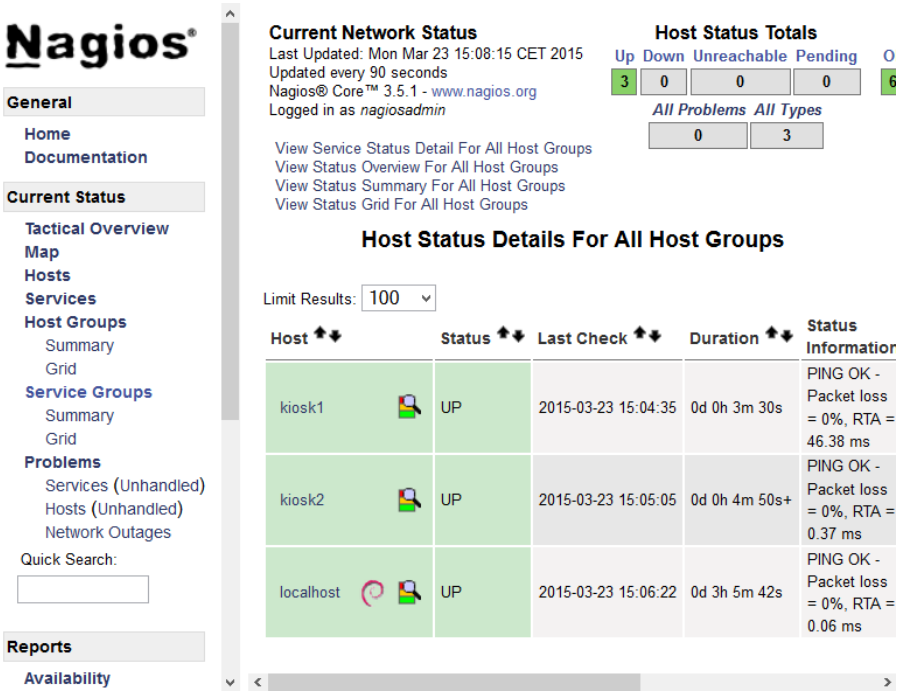
Získání privilegovaného účtu z běžícího systému není téměř možné. Kromě implicitního privilegovaného účtu *root* (ke kterému není ale definováno heslo, tudíž se nelze za něj přihlásit) zde není žádný správcovský účet, který by tyto práva měl.

Pro úplnost ale budu předpokládat, že byl z nějakého důvodu založen správcovský účet a uživatel se dozvěděl přihlašovací údaje. Při pokusu o přihlášení z omezeného účtu za privilegovaný účet pomocí příkazu *su* je přístup odmítnut s chybou *su: Permission denied*. Druhou možností je vzdálené přihlášení, ale na klientském počítači neběží žádná služba pro vzdálený přístup, připojení tedy není možné. V implicitním nastavení tedy není možné těmito způsoby získat kontrolu nad strojem.

4.2.5 Dohledový systém Nagios

Do konfigurační složky `/etc/nagios3/conf.d/` je přidán soubor `kiosk.cfg`, jehož obsah je v příloze A.10. Po restartu služby a přihlášení do webové administrace na adrese `127.0.0.1/nagios3` pomocí uživatelského jména *nagiosadmin* a hesla definovaném při instalačním procesu, lze sledovat definované klienty a lokální server.

V případě, že se tenký klient stane nedostupným, po pěti neúspěšných pokusech dostává administrátor zprávu o nedostupnosti počítače. V momentě, kdy je klient opět dostupný, administrátor je znovu informován pomocí zprávy.



Obrázek 4.1: Webové rozhraní systému Nagios po konfiguraci pro dva tenké klienty

Kapitola 5

Srovnání s existujícími řešeními

Všechny nalezené zranitelnosti existujících implementací popisovaných v kapitole 1 byly u mé implementace otestovány a žádná z nich se v systému neprojevila.

5.1 Rozšířitelnost

Největší problém většiny již existujících řešení je jejich velmi těžká rozšiřitelnost nad rámec možného nastavení. Rozšiřitelnost není ale zcela nemožná i díky open-source licencí u všech zde zmíněných distribucí, je tedy možné po důkladném prostudování způsobu fungování systém nějakým způsobem rozšířit.

V mé implementaci u samostatného klienta je rozšiřitelnost řešena pomocí konvenčního balíčkovacího systému Linuxové distribuce. Administrátor se přihlásí buď pomocí SSH nebo dočasnou úpravou bootovacích parametrů (přidáním klíčového slova *single*), doinstaluje a nakonfiguruje potřebný software a systém je po restartu zpět v módu pouze s prohlížečem, ale s nově doinstalovaným softwarem.

V případě tenkých klientů je rozšiřitelnost řešena stejným způsobem pouze navíc s nutností vytvoření blokového zařízení na serveru po provedení údržby. Tenký klient po restartu následně automaticky načítá upravený systém.

5.2 Licence

V případě distribucí Webconverger a Instant WebKiosk/UB je pro používání pokročilých funkcí vyžadováno zaplacení určitého licenčního poplatku. Podle aktuálního ceníku [6] a [74] je cena stanovena na $100 ročně nebo $200 jednorázově pro jedno PC u Webconvergeru a €12.90 pro Instant WebKiosk/UB.

Porteus kiosk je v základu zdarma bez nutnosti placení poplatků. Zpoplatněná část systému jsou automatické aktualizace na novější verze, za kterou je

cena stanovena na $30—$36 za jednu stanici ročně podle celkového počtu stanic (více v [75]). Další zpoplatněnou službou je vytváření uzpůsobených instalací podle [76], za tuto službu je účtován jednorázový poplatek podle náročnosti požadavků.

Mnou napsané skripty, které přetvářejí klasickou distribuci na zabezpečený kiosk systémy, jsou licencovány podle *GNU General Public License v3* bez nutnosti platit jakýkoliv poplatek za používání.

Závěr

Cílem mé práce bylo zmapovat existující Linuxové systémy implementující lock-down systém, provést analýzu nástrojů pro vytvoření vlastního řešení takového systému a provést jeho implementaci jak ve verzi pro samostatného klienta, tak i pro tenké klienty se serverem. Výsledek implementace měly být také skripty, které automaticky nastaví systém a potřebné nástroje. Dále mělo být provedeno testování funkčnosti a nakonec srovnání s existujícími implementacemi.

V první kapitole své práce jsem představil existující Linuxové implementace se zaměřením na jejich funkce a princip fungování. U každého systému jsem také zanalyzoval zabezpečení a vyhodnotil možné zranitelnosti i s příklady využití těchto zranitelností a dopady na funkčnost systému.

Ve druhé kapitole jsem provedl výběr Linuxové distribuce z několika kandidátů na základě způsobu distribuce nových verzí a délky podpory jednotlivých verzí. Podobný výběr jsem provedl i u webového prohlížeče, zde ale výběr probíhal na základě možností blokování určitých prvků prohlížeče. Ve stejné kapitole jsem také zanalyzoval různé nástroje pro zabezpečení systému a představil technologie pro vytvoření prostředí tenkých klientů včetně dohledového systému.

Ve třetí kapitole jsem využil poznatky z druhé kapitoly a popsal jsem konkrétní nastavení systému a všech nástrojů pro vytvoření zabezpečeného lock-down systému pro samostatného klienta i server s prostředím pro tenké klienty. Z těchto nastavení jsem také vytvořil skripty, které celý systém nastavují automaticky, pouze s nutným interaktivním vstupem od administrátora.

Další kapitola se věnovala testování systému. Zde jsem otestoval nejdříve funkčnost samotných skriptů na instalaci systému podle určeného scénáře. Při testování zabezpečení systému nebyla zjištěna metoda prolomení hlavního zabezpečovacího mechanismu, testování bylo tedy následně prováděno simulací prolomení jednotlivých ochran a jejich dopadů na systém.

V poslední kapitole bylo provedeno srovnání vlastního řešení s již existujícími implementacemi.

Těmito kroky jsem nejen splnil zadání své diplomové práce, ale také hlavní úlohu své práce – vytvořit jednoduše rozšiřitelný zabezpečený lock-down systém nezatížený licenčními (či jinými) poplatky. Díky automatické instalaci může být systém nastaven i neodborníkem na Linuxový systém, ale na druhou stranu expert může do automatického skriptu nahlédnout a upravit ho podle svých specifických potřeb.

Jako možnost pro další vývoj této práce vidím ve vytvoření grafického instalátoru místo využití textového rozhraní. Vytvoření přívětivějšího rozhraní instalátoru ale může na druhou stranu také znamenat zhoršení čitelnost samotného skriptu pro případné administrátory a jeho složitější následnou úpravu.

Literatura

[1] *Zákon č. 257/2001 Sb. Zákon o knihovnách a podmínkách provozování veřejných knihovnických a informačních služeb (knihovní zákon)*. [online]. [cit. 2015-01-28]. Dostupné z: http://aplikace.mvcr.cz/sbirka-zakonu/ViewFile.aspx?type=c&id=3676

[2] Porteus Kiosk: *Porteus Kiosk*. [online]. [cit. 2015-01-28]. Dostupné z: http://porteus-kiosk.org

[3] Porteus Kiosk: *Kiosk Wizard*. [online]. [cit. 2015-01-28]. Dostupné z: http://porteus-kiosk.org/wizard.html

[4] Porteus Kiosk: *Manual Kiosk Customizations*. [online]. [cit. 2015-01-28]. Dostupné z: http://porteus-kiosk.org/kiosk-customization.html

[5] Webconverger Limited: *Webconverger*. [online]. [cit. 2015-01-28]. Dostupné z: https://webconverger.com/

[6] Webconverger Limited: *Pricing*. [online]. [cit. 2015-01-28]. Dostupné z: https://webconverger.com/pricing/

[7] Webconverger Limited: *Webconverger API documentation*. [online]. [cit. 2015-01-28]. Dostupné z: http://webconverger.org/API/

[8] *Tmpfs filesystem*. [online]. [cit. 2015-01-28]. Dostupné z: https://www.kernel.org/doc/Documentation/filesystems/tmpfs.txt

[9] CVE Details: *Adobe Flash Player: Security Vulnerabilities*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.cvedetails.com/vulnerability-list/vendor_id-53/product_id-6761/Adobe-Flash-Player.html

[10] Binary Emotions: *Instant WebKiosk/UB, free full-screen browser-only OS*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.binaryemotions.com/webkiosk-os/index.html

[11] The Register: *Patch Bash NOW: 'Shellshock' bug blasts OS X, Linux systems wide open.* [online]. [cit. 2015-01-28]. Dostupné z: http://www.theregister.co.uk/2014/09/24/bash_shell_vuln/

[12] Jones, M. T.: *GNU/Linux Application Programming.* Cengage Learning, druhé vydání, 2008, ISBN 1-58450-568-0, 667 s.

[13] Wikipedia: *Linuxová distribuce — Wikipedia, otevřená encyklopedie.* [online]. [cit. 2015-01-28]. Dostupné z: http://cs.wikipedia.org/wiki/Linuxov%C3%A1_distribuce

[14] Root.cz: *Rozdíly mezi jednotlivými distribucemi.* [online]. [cit. 2015-01-28]. Dostupné z: http://www.root.cz/specialy/vyber-distribuce/rozdily-mezi-jednotlivymi-distribucemi/

[15] DistroWatch: *DistroWatch Page Hit Ranking.* [online]. [cit. 2015-01-28]. Dostupné z: http://distrowatch.com/dwres.php?resource=popularity

[16] Debian: *Debian Social Contract.* [online]. [cit. 2015-01-28]. Dostupné z: https://www.debian.org/social_contract.en.html

[17] Debian: *Resources for Debian Developers and Debian Maintainers.* [online]. [cit. 2015-01-28]. Dostupné z: https://www.debian.org/doc/manuals/developers-reference/resources.html#s4.6.4

[18] Debian Wiki: *DebianReleases.* [online]. [cit. 2015-01-28]. Dostupné z: https://wiki.debian.org/DebianReleases

[19] Debian Wiki: *LTS: Long Term Support.* [online]. [cit. 2015-01-28]. Dostupné z: https://wiki.debian.org/LTS/

[20] Ubuntu Wiki: *Ubuntu Releases.* [online]. [cit. 2015-01-28]. Dostupné z: https://wiki.ubuntu.com/Releases

[21] Ubuntu Documentation Team: *Preparing to Install — Server and Desktop Differences.* [online]. [cit. 2015-01-28]. Dostupné z: https://help.ubuntu.com/14.04/serverguide/preparing-to-install.html#intro-server-differences

[22] Mint Team: *Introducing Cinnamon.* [online]. [cit. 2015-01-28]. Dostupné z: http://blog.linuxmint.com/?p=1910

[23] Mint Team: *About Us.* [online]. [cit. 2015-01-28]. Dostupné z: http://www.linuxmint.com/about.php

[24] Mint Team: *Linux Mint Debian 201204 RC (MATE/Cinnamon & Xfce) released!* [online]. [cit. 2015-01-28]. Dostupné z: http://blog.linuxmint.com/?p=1967

[25] The Fedora Project: *What is the Fedora Project?* [online]. [cit. 2015-01-28]. Dostupné z: http://fedoraproject.org/wiki/Fedora_Project_Wiki

[26] *Fedora Project: Announcing New Direction*. [online]. [cit. 2015-01-28]. Dostupné z: www.redhat.com/archives/rhl-list/2003-September/msg00064.html

[27] The Fedora Project: *Objectives*. [online]. [cit. 2015-01-28]. Dostupné z: http://fedoraproject.org/wiki/Objectives

[28] The Fedora Project: *Fedora Release Life Cycle*. [online]. [cit. 2015-01-28]. Dostupné z: http://fedoraproject.org/wiki/Fedora_Release_Life_Cycle

[29] openSUSE Project: *Sponsors*. [online]. [cit. 2015-01-28]. Dostupné z: https://en.opensuse.org/Sponsors

[30] openSUSE Project: *YaST Modules*. [online]. [cit. 2015-01-28]. Dostupné z: https://en.opensuse.org/openSUSE:YaST_Modules

[31] openSUSE Project: *Tumbleweed Portal*. [online]. [cit. 2015-01-28]. Dostupné z: https://en.opensuse.org/Portal:Tumbleweed

[32] openSUSE Project: *Lifetime*. [online]. [cit. 2015-01-28]. Dostupné z: https://en.opensuse.org/Lifetime

[33] openSUSE Project: *Evergreen*. [online]. [cit. 2015-01-28]. Dostupné z: https://en.opensuse.org/openSUSE:Evergreen

[34] Hertzog, R.; Mas, R.: *The Debian Administrator's Handbook*. [online]. [cit. 2015-01-28]. Dostupné z: http://debian-handbook.info/browse/stable/sect.dist-upgrade.html

[35] Ubuntu Documentation Team: *MinimalCD*. [online]. [cit. 2015-01-28]. Dostupné z: https://help.ubuntu.com/community/Installation/MinimalCD

[36] The Chromium projects: *Welcome to Chromium*. 2008, [online]. [cit. 2015-01-18]. Dostupné z: http://blog.chromium.org/2008/09/welcome-to-chromium_02.html/

[37] The Chromium Projects: *Linux Quick Start*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.chromium.org/administrators/linux-quick-start

[38] Opera Software ASA: *Opera Software End-User License Agreement for Opera Desktop Browser*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.operasoftware.com/eula/browser

[39] Opera Software ASA: *Supported extension APIs*. [online]. [cit. 2015-01-28]. Dostupné z: https://dev.opera.com/extensions/apis.html

[40] Opera Software ASA: *Opera Browser for Linux i386*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.opera.com/download/guide/?os=linux-i386&list=all

[41] Schrode, M.: *Opera browser: Blocking unwanted ads*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.schrode.net/opera/url_filtering

[42] Opera Software ASA: *Opera's Kiosk Mode*. [online]. [cit. 2015-01-28]. Dostupné z: http://web.archive.org/web/20130223014915/http://www.opera.com/support/mastering/kiosk/

[43] Mozilla Foundation: *Mozilla Public License Version 2.0*. [online]. [cit. 2015-01-28]. Dostupné z: https://www.mozilla.org/MPL/2.0/

[44] MozillaZine: *UserChrome.css*. [online]. [cit. 2015-01-28]. Dostupné z: http://kb.mozillazine.org/index.php?title=UserChrome.css

[45] Guciek, K.: *Read-only Root Filesystem*. [online]. [cit. 2015-01-28]. Dostupné z: https://sites.google.com/site/linuxpendrive/rorootfs

[46] Friedl, S.: *Best Practices for UNIX chroot() Operations*. [online]. [cit. 2015-01-28]. Dostupné z: http://unixwiz.net/techtips/chroot-practices.html

[47] Simes: *How to break out of a chroot() jail*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.bpfh.net/simes/computing/chroot-break.html

[48] Nguyen, B.: *Linux Filesystem Hierarchy — /home*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/html/home.html

[49] Mydraal: *Linux Magic System Request Key Hacks*. [online]. [cit. 2015-01-28]. Dostupné z: https://www.kernel.org/doc/Documentation/sysrq.txt

[50] Security Research Labs: *BadUSB - On accessories that turn evil*. [online]. [cit. 2015-01-28]. Dostupné z: https://srlabs.de/blog/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf

[51] Hansperl, G.: *Virtual Consoles in Linux*. 1998, [online]. [cit. 2015-01-28]. Dostupné z: http://luv.asn.au/overheads/virtualconsoles.html

[52] Intel Corporation: *Preboot Execution Environment (PXE) Specification*. 1999, [online]. [cit. 2015-01-28]. Dostupné z: http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf

[53] DisklessWorkstations.com, LLC: *Download LTSP*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.ltsp.org/download/

[54] Kraai, M.: *debootstrap(8) - Linux man page*. [online]. [cit. 2015-01-28]. Dostupné z: http://linux.die.net/man/8/debootstrap

[55] Verhelst, W.: *debootstrap(8) - Linux man page*. [online]. [cit. 2015-01-28]. Dostupné z: http://manpages.ubuntu.com/manpages/lucid/man1/nbd-server.1.html

[56] Lougher, P.: *SQUASHFS*. [online]. [cit. 2015-01-28]. Dostupné z: http://squashfs.sourceforge.net/

[57] LTSPedia: *Concepts*. [online]. [cit. 2015-01-28]. Dostupné z: http://wiki.ltsp.org/wiki/Concepts

[58] LTSPedia: *Configuration*. [online]. [cit. 2015-01-28]. Dostupné z: http://wiki.ltsp.org/wiki/Configuration

[59] Macii, E.: *Ultra Low-Power Electronics and Design*. 2004, [online]. [cit. 2015-01-28]. Dostupné z: https://books.google.cz/books?id=k4OBY0RQwTwC&lpg=PA204&ots=WH0M0c0yK5&pg=PA204#v=onepage&q&f=false

[60] Korb, K.: *Network Booting Linux (PXE)*. 2008, [online]. [cit. 2015-01-28]. Dostupné z: http://www.sanitarium.net/golug/netboot.html

[61] BRG Horn: *How to install a Linux Terminal Server*. 2003, [online]. [cit. 2015-01-28]. Dostupné z: http://www.bghorn.ac.at/~chris/ltsp/html/documentation.html

[62] Breuer, P. T.; Lopez, A. M.; Aresi, A. G.: *The Network Block Device*. 2000, [online]. [cit. 2015-01-28]. Dostupné z: http://www.linuxjournal.com/article/3778

[63] Anderson, A.: *The Network File System*. 1996, [online]. [cit. 2015-01-28]. Dostupné z: http://www.tldp.org/LDP/nag/node140.html

[64] Microsoft Corporation: *Operating Systems and PAE Support*. 2009, [online]. [cit. 2015-01-28]. Dostupné z: https://msdn.microsoft.com/en-us/library/windows/hardware/Dn613969%28v=vs.85%29.aspx

[65] Larabel, M.: *Ubuntu 32-bit, 32-bit PAE, 64-bit Kernel Benchmarks*. 2009, [online]. [cit. 2015-01-28]. Dostupné z: http://www.phoronix.com/scan.php?page=article&item=ubuntu_32_pae

[66] Nagios Enterprises: *Nagios Core*. 2013, [online]. [cit. 2015-01-28]. Dostupné z: http://www.nagios.org/download/core/

[67] Nagios Enterprises: *Nagios Exchange*. [online]. [cit. 2015-01-28]. Dostupné z: http://exchange.nagios.org/

[68] Gafton, C.: *pam_wheel(8) - Linux man page*. [online]. [cit. 2015-01-28]. Dostupné z: http://linux.die.net/man/8/pam_wheel

[69] Google Inc.: *Chrome System requirements*. [online]. [cit. 2015-01-18]. Dostupné z: https://support.google.com/chrome/answer/95346?hl=en

[70] Dassen, J. H. M.; Stickelman, C.: *The Debian GNU/Linux FAQ – Chapter 11 - Customizing your installation of Debian GNU/Linux*. 2014, [online]. [cit. 2015-01-28]. Dostupné z: https://www.debian.org/doc/manuals/debian-faq/ch-customizing.en.html

[71] LTSPedia: *Installation*. [online]. [cit. 2015-01-28]. Dostupné z: http://wiki.ltsp.org/wiki/Installation

[72] Layton, J. B.: *Intro to Nested-RAID: RAID-01 and RAID-10*. 2011, [online]. [cit. 2015-01-28]. Dostupné z: http://www.linux-mag.com/id/7928/2/

[73] Oracle Corporation: *Oracle VM VirtualBox User Manual*. 2015, [online]. [cit. 2015-01-28]. Dostupné z: http://download.virtualbox.org/virtualbox/UserManual.pdf

[74] Binary Emotions: *Download Instant WebKiosk/UB, Unrestricted Browsing*. [online]. [cit. 2015-01-28]. Dostupné z: http://www.binaryemotions.com/webkiosk-os/download.html

[75] Porteus Kiosk: *Automatic updates*. [online]. [cit. 2015-01-28]. Dostupné z: http://porteus-kiosk.org/automatic-updates.html

[76] Porteus Kiosk: *Let us build your kiosk*. [online]. [cit. 2015-01-28]. Dostupné z: http://porteus-kiosk.org/builds.html

Příloha A

Použité konfigurace a skripty

Listing A.1: skript /usr/local/bin/kill_browser.sh

```
#!/bin/bash
if [ ! -r /tmp/before.t ]; then
        date +"%s" > /tmp/before.t
        pkill -9 chromium
        exit 0
fi
elapsedTime=$((`date +"%s"`-`cat /tmp/before.t`))
if [ $elapsedTime -gt 122 ]; then
        pkill -9 chromium
fi
date +"%s" > /tmp/before.t
```

Listing A.2: Nastavení politik Prohlížeče Chromium

```
{
"AllowFileSelectionDialogs": false,
"AllowOutdatedPlugins": true,
"AlternateErrorPagesEnabled": true,
"AlwaysAuthorizePlugins": true,
"AudioCaptureAllowed": false,
"AutoFillEnabled": false,
"BackgroundModeEnabled": false,
"BlockThirdPartyCookies": true,
"BookmarkBarEnabled": false,
"BuiltInDnsClientEnabled": false,
"CloudPrintProxyEnabled": false,
"CloudPrintSubmitEnabled": false,
"DefaultBrowserSettingEnabled": true,
"DefaultCookiesSetting": 1,
```

```
"DefaultGeolocationSetting": 2,
"DefaultImagesSetting": 1,
"DefaultJavaScriptSetting": 1,
"DefaultNotificationsSetting": 2,
"DefaultPopupsSetting": 2,
"DefaultSearchProviderEnabled": false,
"DeveloperToolsDisabled": true,
"DeviceAllowNewUsers": false,
"Disable3DAPIs": false,
"DisablePluginFinder": true,
"DisableSafeBrowsingProceedAnyway": true,
"DisableScreenshots": true,
"DisableSpdy": true,
"DisabledPlugins": ["Java", "Shockwave Flash"],
"DiskCacheDir": "/home/${user_name}/Chrome_cache",
"DiskCacheSize": 104857600,
"DnsPrefetchingEnabled": false,
"DownloadDirectory": "/home/${user_name}/Downloads",
"EditBookmarksEnabled": false,
"EnabledPlugins": ["Chrome PDF Viewer"],
"ExtensionInstallBlacklist": ["*"],
"FullscreenAllowed": true,
"HideWebStoreIcon": true,
"HomepageIsNewTabPage": false,
"ImportBookmarks": false,
"ImportHistory": false,
"ImportHomepage": false,
"ImportSavedPasswords": false,
"ImportSearchEngine": false,
"IncognitoModeAvailability": 2,
"MetricsReportingEnabled": false,
"NativeMessagingUserLevelHosts": false,
"PasswordManagerAllowShowPasswords": false,
"PasswordManagerEnabled": false,
"PrintingEnabled": false,
"ProxyMode": "direct",
"RemoteAccessHostAllowClientPairing": false,
"RemoteAccessHostRequireTwoFactor": false,
"RestoreOnStartup": 4,
"SafeBrowsingEnabled": true,
"SavingBrowserHistoryDisabled": true,
"SearchSuggestEnabled": false,
"ShowHomeButton": true,
"SigninAllowed": false,
```

```
"SpellCheckServiceEnabled": false,
"SupervisedUserCreationEnabled": false,
"SyncDisabled": true,
"TranslateEnabled": false,
"VideoCaptureAllowed": false,
"HomepageLocation": "http://katalog.cbvk.cz",
"RestoreOnStartupURLs": ["http://katalog.cbvk.cz"],
"URLBlacklist": ["*"],
"URLWhitelist": ["obalkyknih.cz", "katalog.cbvk.cz", "
   cbvk.cz" ,"books.google.cz", "books.google.com", "
   mojeid.cz" ]
}
```

Listing A.3: Příkazy pro nastavení firewallu pro samostatného klienta

```
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j
   ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-request -j
   ACCEPT
iptables -A OUTPUT -t filter -p tcp --dport http -j
   ACCEPT
iptables -A OUTPUT -t filter -p tcp --dport 8080 -j
   ACCEPT
iptables -A OUTPUT -t filter -p tcp --dport 53 -j ACCEPT
iptables -A OUTPUT -t filter -p udp --dport 53 -j ACCEPT
iptables -A OUTPUT -t filter -p tcp --dport https -j
   ACCEPT
iptables -A OUTPUT -t filter -p udp --dport https -j
   ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -m state --state
   ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -o eth0 -m owner --uid-owner 0 -j
   ACCEPT
iptables -A OUTPUT -o eth0 -m owner --uid-owner guest -j
    REJECT --reject-with icmp-host-prohibited

iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-request -j
   ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j
   ACCEPT
iptables -A INPUT -p tcp --dport 22 -m state --state NEW
   ,ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -
    j ACCEPT

iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Listing A.4: Posloupnost příkazů pro vytvoření AuFS systému nad domovskou složkou

```
mkdir /tempFiles/home
mkdir -p /bkupRO/home
chmod 0700 /bkupRO/home
rsync --delete --archive /home/ /bkupRO/home/
rm -r /home/
mount -t aufs -o noauto,nosuid,noexec,nodev,br:/
    tempFiles/home=rw:/bkupRO/home=ro,udba=reval none /
    home
```

Listing A.5: Skript grafického prostředí $HOME/.xsession

```
#!/bin/bash
LANG=cs_CZ.UTF-8
setxkbmap cz
xset -dpms
xset s off
xautolock -secure -time 2 -locker /usr/local/bin/
    kill_browser.sh &
while true; do
        sudo /usr/bin/rsync -qr --delete --archive /
            bkupRO$HOME/ $HOME/
        sudo /usr/bin/rsync -qr --delete --archive --
            exclude log /bkupRO/var/ /var/
        sed -i 's/"exited_cleanly": false/"
            exited_cleanly": true/' "$HOME/.config/
            chromium/Default/Preferences"
        sed -i 's/"exit_type": "Crashed"/"exit_type": "
            Normal"/' "$HOME/.config/chromium/Default/
            Preferences"
        xmodmap -e "keycode 67 = 0x0000" #disables F1
            used for Help
        xsetroot -cursor_name left_ptr
        chromium-browser --start-maximized
done
```

Listing A.6: Obsah skript `/etc/rc.local`

```
mount /home
mount /var
mount -o remount,ro /
/usr/bin/rsync -qr --delete --archive /bkupRO/home/guest
    / /home/guest/
/usr/bin/rsync -qr --delete --archive --exclude log /
    bkupRO/var/ /var/
service ssh restart
su - -s "/bin/bash" -c "startx" guest
reboot
exit 0
```

Listing A.7: Obsah souboru `lts.conf` obsahující nastavení pro LTSP klienty

```
[default]
LDM_AUTOLOGIN=TRUE
LDM_SESSION=nodm
SCREEN_07 = kiosk
DISABLE_GETTYS = True
```

Listing A.8: Obsah upraveného souboru `/usr/share/ltsp/screen.d/kiosk` nastavující relaci

```
#!/bin/sh
#
# The following script works for LTSP5.
#
# This software is licensed under the Gnu General Public
    License.
# The full text of which can be found at http://www.LTSP
    .org/license.txt
#

PATH=/bin:$PATH; export PATH
. /usr/share/ltsp/screen-x-common

export NODM_XSESSION="/usr/share/ltsp/xinitrc /etc/X11/
    Xsession"
export NODM_X_OPTIONS="${DISPLAY} vt${TTY} ${X_ARGS} -br
    "

[ -n "$2" ] && KIOSK_OPTIONS=$2
```

```
KIOSK_EXE="/usr/bin/chromium-browser --start-maximized"

if boolean_is_true "${KIOSK_DAEMON:-"False"}"; then
    export XINITRC_DAEMON="True"
fi

if [ -x /usr/share/ltsp/xinitrc ]; then
    xinitrc=/usr/share/ltsp/xinitrc
fi

KIOSKUSER=${KIOSKUSER:-"ltspkiosk"}
if [ -z "$(getent passwd ${KIOSKUSER})" ]; then
    # create a ltspkiosk user
    adduser --no-create-home --disabled-password --gecos
        ,,, ${KIOSKUSER}
fi

# Create a tmpdir to be our homedir
TMPDIR=$(mktemp -d /tmp/.kiosk-XXXXXX)
chown ${KIOSKUSER} ${TMPDIR}

# Edit passwd homedir entry for programs that look it up
    from there
sed -i -e '\|'${KIOSKUSER}'|s|[^:]*\(:[^:]*\)$|'$TMPDIR
    '\1|' /etc/passwd

# Change owner of $HOME/Downloads
mkdir ${TMPDIR}/Downloads
chown root:root ${TMPDIR}/Downloads

su - ${KIOSKUSER} -c "XINITRC_DAEMON=${XINITRC_DAEMON}
   KIOSK_WM=${KIOSK_WM} xinit $xinitrc /usr/share/ltsp/
   kioskSession ${KIOSK_EXE} ${KIOSK_OPTIONS} -- ${
   DISPLAY} vt${TTY} ${X_ARGS} -br" >/dev/null

rm -rf ${TMPDIR}
```

Listing A.9: Obsah upraveného souboru `/usr/share/ltsp/kioskSession` spouštějící relaci

```
#!/bin/sh

KIOSK_EXE=$1
KIOSKHOME=/usr/local/share/ltspkiosk/home
```

```
KIOSKSTARTUP=/usr/local/share/ltspkiosk/startup

KIOSKUSER=${KIOSKUSER:-"ltspkiosk"}
KIOSK_WM=${KIOSK_WM:-"/usr/sbin/nodm"}
KIOSK_USER_STARTUP="${HOME}/.kiosk-startup"

mkdir -p ${KIOSK_USER_STARTUP}

if [ -x "${KIOSK_WM}" ]; then
    ln -s ${KIOSK_WM} ${KIOSK_USER_STARTUP}/00-kiosk-wm
fi

# Copy a default homedir if present
if [ -d  "${KIOSKHOME}" ]; then
    cp -r ${KIOSKHOME}/* ${KIOSKHOME}/.??* ${HOME}
fi

for file in ${KIOSKSTARTUP}/* ; do
    if [ -f "${file}" ]; then
        ln -s "${file}" "${KIOSK_USER_STARTUP}/$(
           basename ${file})"
    fi
done

[ -n "${XAUTHORITY}" ] && cp -a ${XAUTHORITY} ${HOME}

for i in ${KIOSK_USER_STARTUP}/* ; do
    [ -x "${i}" ] && eval "${i}" &
done

which chromium-browser 1>/dev/null 2>&1 || {
    ldm-dialog --message "Browser not found."
    [ "$USER" != "root" ] && pkill -u $USER
    exit 0
}

LANG=cs_CZ.UTF-8
setxkbmap cz
xset -dpms
xset s~off
xautolock -secure -time 2 -locker /usr/local/bin/
   kill_browser.sh &
while true; do
        emptyfolder=`mktemp -d`
```

```
        /usr/bin/rsync -qr --delete --archive --exclude
            'Downloads' "$emptyfolder/" "$HOME/"
        rm -rf $emptyfolder/
        sed -i 's/"maximized": false/"maximized": true/'
            "$HOME/.config/chromium/Default/Preferences"
        sed -i 's/"exited_cleanly": false/"
            exited_cleanly": true/' "$HOME/.config/
            chromium/Default/Preferences"
        sed -i 's/"exit_type": "Crashed"/"exit_type": "
            Normal"/' "$HOME/.config/chromium/Default/
            Preferences"
        xmodmap -e "keycode 67 = 0x0000" #disables F1
            used for Help
        xsetroot -cursor_name left_ptr
        chromium-browser --start-maximized
done


[ "$USER" != "root" ] && pkill -u $USER
exit 0
```

Listing A.10: Příklad definujících příkazů pro Nagios

```
define host{
    name                            kiosk-client
    notifications_enabled             1
    event_handler_enabled             1
    flap_detection_enabled            1
    failure_prediction_enabled        1
    process_perf_data                 1
    retain_status_information         1
    retain_nonstatus_information      1
    check_period                      24x7
    check_interval                    5
    retry_interval                    1
    max_check_attempts                10
    check_command             check-host-alive
    notification_period           workhours
    notification_interval             120
    notification_options            d,u,r
    contact_groups                  admin
    register                          0             ; ITS
```

```
        NOT A~REAL HOST, JUST A~TEMPLATE!
         }

define host{
   use             kiosk-client
   host_name       kiosk1
   alias           Kiosk klient 1
   address         192.168.56.101
   hostgroups      kiosk-clients
   contacts        nagiosadmin
   }

define host{
   use             kiosk-client
   host_name       kiosk2
   alias           Kiosk klient 2
   address         192.168.56.102
   hostgroups      kiosk-clients
   contacts        nagiosadmin
   }


define contact{
   contact_name                    nagiosadmin
   service_notification_period     24x7
   host_notification_period        24x7
   service_notification_options    w,u,c,r
   host_notification_options       d,r
   service_notification_commands   notify-service-by-
      email
   host_notification_commands      notify-host-by-email
   alias                           Main Admin
   email                           root@localhost
   }

define contactgroup{
   contactgroup_name       admin
   alias                   Nagios Administrators
   members                 nagiosadmin
   }

define hostgroup{
   hostgroup_name  kiosk-clients
   alias           LTSP Kiosks
```

```
}
```

Listing A.11: Příkazy pro nastavení firewallu pro server

```
# eth0 = External network (Internet connectivity)
# eth1 = Internal network with thin clients

iptables -A OUTPUT -o lo -j ACCEPT
iptables -A OUTPUT -o eth1 -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j
   ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-request -j
   ACCEPT
iptables -A OUTPUT -t filter -p tcp --dport http -j
   ACCEPT
iptables -A OUTPUT -t filter -p tcp --dport 8080 -j
   ACCEPT
iptables -A OUTPUT -t filter -p tcp --dport 53 -j ACCEPT
iptables -A OUTPUT -t filter -p udp --dport 53 -j ACCEPT
iptables -A OUTPUT -t filter -p tcp --dport https -j
   ACCEPT
iptables -A OUTPUT -t filter -p udp --dport https -j
   ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -m state --state
   ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i eth1 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-request -j
   ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j
   ACCEPT
iptables -A INPUT -p tcp --dport 22 -m state --state NEW
   ,ESTABLISHED -j ACCEPT
iptables -A INPUT -m state --state RELATED,ESTABLISHED -
   j ACCEPT

iptables -A FORWARD -o eth0 -i eth1 -j ACCEPT
iptables -A FORWARD -o eth1 -i eth0 -m state --state
   RELATED,ESTABLISHED -j ACCEPT

iptables --table nat --append POSTROUTING --jump
   MASQUERADE --source 192.168.56.0/24
```

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Listing A.12: Natavení DHCP v souboru `/etc/ltsp/dhcpd.conf`

```
authoritative;

subnet 192.168.56.0 netmask 255.255.255.0 {
    option domain-name "ltsp";
    option domain-name-servers 192.168.0.1;
    option broadcast-address 192.168.56.255;
    option routers 192.168.56.1;
    option subnet-mask 255.255.255.0;
    option root-path "/opt/ltsp/i386";
    if substring( option vendor-class-identifier, 0, 9 )
        = "PXEClient" {
       filename "/ltsp/i386/pxelinux.0";
    } else {
       filename "/ltsp/i386/nbi.img";
    }

    pool {
        max-lease-time 28800;
        range 192.168.56.2 192.168.56.254;
        deny unknown-clients;
    }


}
host kiosk1 {
    hardware ethernet 08:00:27:EB:7E:CC;
    fixed-address 192.168.56.101;
}
host kiosk2 {
    hardware ethernet 08:00:27:28:7C:02;
    fixed-address 192.168.56.102;
}
```

Příloha B

Seznam tabulek

Tabulka B.1: Přepínače pro konfiguraci Kiosk režimu v prohlížeči Opera

Přepínač	Popis
kioskmode	Spustí Operu v Kiosk režimu
kioskbuttons	Zobrazí hlavní panel a adresní řádku v celoobrazovkovém režimu
kioskresetstation	Návrat na domovskou stránku po určité zadané době nečinnosti uvedené v nastavení "Go Home Time Out"(implicitně 30 vteřin)
nochangebuttons	Zakáže změnu tlačítek na hlavním panelu
nocontextmenu	Odstraní všechna kontextová menu
nodownload	Zakáže stahovací dialogy a nedovolí stahování souborů
nokeys	Zakáže všechny klávesové zkratky
nomail	Zakáže vestavěného e-mailového klienta a chat
nomenu	Zakáže nabídky Opery (zároveň zakazuje změnu filtrů blokovaných stránek
nominmaxbuttons	Zakáže tlačítka minimalizovat, maximalizovat a zavřít
noprint	Zakáže tlačítko pro tisk
nosave	Zakáže ukládání souborů, stránek, obrázků a odkazů z webových stránek.
nowin	Při startu nečte uloženou relaci (pokud existuje) a vždy vytváří novou
resetonexit	Při ukončení smaže všechna osobní data (historie, cookies a mezipaměť)

Příloha C

Seznam použitých zkratek

AMD Advanced Micro Devices, 18

API Application Program Interface, 3

APT Advanced Packaging Tool, 49

ASA Allmennaksjeselskap, 22

AuFS Another union File System, 2

CD Compact Disc, 21

CLI Command-line Interface, 3

CPU Central Processing Unit, 15

DHCP Dynamic Host Configuration Protocol, 3

DNS Domain Name System, 3

GB Gigabyte, 40

GHz Gigahertz, 44

GiB Gibibyte, 33

GID Group Identifier, 47

GNU GNU's Not Unix!, 9

GPL General Public License, 11

GPU Graphics Processor Unit, 40

GRUB GRand Unified Bootloader, 3

GUI Graphical User Interface, 4

HDD Hard Disk Drive, 40

HTTP Hypertext Transfer Protocol, 44

HTTPS Hypertext Transfer Protocol Secure, 44

I/O Input/Output, 45

ICMP Internet Control Message Protocol, 4

Inc Incorporation, 19

IP Internet Protocol, 3

IPv4 Internet Protocol version 4, 46

MPL Mozilla Public License, 23

TUI Text-based Iser Interface, 4

JSON JavaScript Object Notation, 21

LTD Private Limited Company, 18

LTS Long Term Support, 18

LTSP Linux Terminal Server Project, 29

MAC Media Access Control, 31

MB Megabyte, 29

MHz Megahertz, 29

MiB Mebibyte, 40

NAT Network Address Translation, 4

NBD Network Block Device, 4

NFS Network File System, 4

NIC Network Interface Controller, 45

PAE Physical Address Extension, 33

PAM Pluggable Authentication Module, 38

PC Personal Computer, 18

PCI Peripheral Component Interconnect, 31

PDF Portable Document Format, 37

PPAPI Pepper Plugin API, 22

PXE Preboot Execution Environment, 29

RAM Random Access Memory, 7

RDP Remote Desktop Protocol, 28

RFB Remote FrameBuffer, 28

RPM RPM Package Manager, 5

SSH Secure Shell, 7

TFTP Trivial File Transfer Protocol, 5

UI User Interface, 24

UID User Identifier, 47

URL Uniform Resource Locator, 7

USB Universal Serial Bus, 11

VNC Virtual Network Computing, 7

YaST Yet another Setup Tool, 19

Příloha D

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `scripts` ..... zdrojové kódy skriptů a ukázkových nastavení
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF