FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Kiosk browser pro svobodný operační systém
Student:	Martin Horský
Vedoucí:	Ing. Miroslav Prágl, MBA
Studijní program:	Informatika
Obor / specializace:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

- Analyzujte stávající dostupná řešení pro realizaci technologie on-premise kiosk browseru pro svobodný (Linux a pod.) operační systém.
- S použitím technologií dle svého výběru navrhněte, realizujte a otestuje vlastní řešení s důrazem na:
  - kompatibilitu (co nejlepší použitelnost při omezení funkčnosti vstupního rozhraní nebo prezentační vrstvy kiosku) při zachování robustnosti, bezpečnosti a stability,
  - centralizaci správy,
  - škálovatelnost instalace a údržby,
  - rozšiřitelnost a svobodnou licenci výsledného řešení.

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 25. ledna 2021 v Praze.

Bakalářská práce

KIOSK BROWSER PRO SVOBODNÝ OPERAČNÍ SYSTÉM

Martin Horský

Fakulta informačních technologií ČVUT v Praze
Katedra počítačových systémů
Vedoucí: Ing. Miroslav Prágl, MBA
12. května 2021

Obsah

Seznam výpisů kódu

Chtěl bych poděkovat Ing. Miroslavu Práglovi, MBA za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat. Také bych chtěl poděkovat svým rodičům a blízkým za podporu při mém studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů připřípravě 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. §2373 odst. 2 zákona č.89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou jejísoučástí 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 aza jakýmkoli účelem (včetně užití k výdělečnýmúčelům). Toto oprávnění je časově, teritoriálně imnožstevně 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 sjiným dílem, zařazením Díla dodíla 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 10. května 2020 .....

Abstrakt

Tato práce se zabývá implementací kiosk systému nainstalovaném na místním zařízení. Cílem práce je vytvořit systém, který přijímá vstup buď z klávesnice a myši, nebo pouze z dotykové obrazovky. Nejprve byla zanalyzována existující volně dostupná i komerční řešení. Dále byly prozkoumány možnosti sestavení vlastního systému za pomoci svobodného software. Následně bylo pomocí těchto zjištění implementováno vlastní řešení a připraven instalační skript, který nastavuje čistou instalaci systému Debian na kiosk. Také byla připravena role do systému Ansible pro dosažení stejného výsledku škálovatelným způsobem.

Výsledný systém zobrazuje pouze prohlížeč s administrátorem nastavenou stránkou. Veškeré grafické elementy prostředí (jako spodní lišta nebo hlavička okna prohlížeče) jsou skryty a systém je zabezpečen proti spuštění jiných programů uživatelem.

Klíčová slova kiosk systém, operační systém, linuxová distribuce, souborový systém, zavaděč, vzdálená správa, orchestrační systém

Abstract

This thesis is about implementing a locally installed kiosk system. The goal was to create a system that would accept input from either a mouse and keyboard or just a touch screen. Firstly, existing freely available and proprietary solutions were analyzed. Also, options for implementing a custom solution using free software were investigated. Afterwards a custom kiosk system was implemented using these findings. An installation script, which would configure a clean install of Debian as a kiosk, was created. A role definition for the Ansible orchestration system was also prepared to achieve a more scalable enrollment solution.

Finished system displays only a browser with a website configured by an administrator. All graphical elements of the desktop user interface (like the task bar or window borders) are disabled and the system is secured accordingly to prevent users from executing other programs.

Keywords kiosk system, operating system, linux distribution, filesystem, boot loader, remote management, orchestration system

Seznam zkratek

API	Application Interface
BSD	Berkley Software Distribution
DM	Display Manager
DNS	Domain Name System
DRM	Digital Rights Management
FUSE	Filesystem in Userspace
GNOME	GNU Network Object Model Environment
GNU	GNU's Not Unix!
GPL	General Public License
GRUB	Grand Unified Boot Loader
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ISC	Internet Systems Consortium
JSON	JavaScript Object Notation
LAN	Local Area Network
LGPL	Lesser General Public License
LTS	Long Term Support
MIT	Massachusetts Institute of Technology
MPL	Mozilla Public License
NTP	Network Time Protocol
OS	Operační systém
OSK	On Screen Keyboard
RDP	Remote Desktop Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
VESA	Video Electronics Standards Association
VNC	Virtual Network Computing
VTY	Virtual Teletypewriter
YAML	YAML Ain't Markup Language

Kapitola 1
Úvod

Internetový kiosk je vestavné zařízení určené pro použití kýmkoliv, zpravidla za účelem prohlížení určené webové stránky. Typicky se takový kiosk používá v obchodech, kde slouží pro prohlížení e-shopu obchodu, aby si mohl zákazník prohlédnout i sortiment, který kupříkladu není na pobočce naskladněn. Kiosk zprostředkovává uživateli – zákazníku prodejny – veškeré možnosti, které mu poskytují webové stránky obchodu, jako zakládání uživatelských účtů, jejich editace, objednání zboží a kontrola vystavených faktur z předchozích objednávek. Kiosk může také sloužit jako informační stanice, např. v metru nebo turistických oblastech. Dokonce i některé staré bankomaty fungovali na podobném principu. Konkrétně byly některé postaveny na systému Windows XP.

Jelikož je kiosk součástí sítě prodejny, je žádoucí, aby byl náležitě zabezpečen a nemohl být použit jako bod vniku do sítě, do které běžný uživatel kiosku nesmí mít přístup. Musí tedy znemožňovat prohlížení jiných internetových stránek a spouštění či instalaci programů. Zároveň nesmí být možné, aby s kioskem mohl cílový uživatel manipulovat fyzicky, např. připojit externí úložiště, připojit nebo odpojit kabely nebo jej náhle vypnout. Toho je dosaženo zpravidla zabezpečením mechanickým, běžný uživatel tedy nemá ke konektorům a hardware tlačítkům pro vypnutí a reset fyzický přístup. Toto však nemusí být jediná ochrana hardware části kiosku. Jelikož kreativita potenciálních útočníků na síť může být veliká a mohou překonat mechanické zabezpečení bez povšimnutí personálu obchodu, je vhodné do kiosku přidat ještě další stupeň ochrany implementovaný do jeho software.

Jelikož není cílem této práce zkoumat fyzickou podobu kiosku a jeho mechanické zabezpečení, bude se tato práce věnovat pouze software části.

V současné době jsou k dispozici komerční řešení kiosk systémů. Tyto systémy jsou však obtížně rozšířitelné dle potřeb poptávajících společností. Přidat nové funkce do těchto systémů je často velmi složité. Krom toho jsou tato komerční řešení finančně nákladná. Proto je vytvoření vlastního kiosku pomocí svobodných operačních systémů a nástrojů jednodušší variantou.

Existující svobodná řešení mají všechny placené verze. Ty se od neplacených liší klíčovými funkcionalitami. Tento problém není tedy jen technická záležitost, ale také licenční.

Na mnoha stránkách existují návody pro vytvoření kiosk systému. Tyto návody se ale zpravidla zaměřují pouze na vytvoření funkčního kiosku, s minimálním zaměřením na zabezpečení. Slouží tedy pouze jen jako základ pro již zkušené administrátory.

Kapitola 2
Cíl práce

Cílem této bakalářské práce je vytvoření administrativních skriptů pro některý open-source operační systém, které jej nastaví na tzv. kiosk systém a budou jej dále spravovat. Tyto skripty budou jednak v podobě jednorázových skriptů a jako modul pro některý orchestrační nástroj. Výsledný systém by měl být funkční s klávesnicí a myší nebo s dotykovou obrazovkou včetně virtuální klávesnice.

V teoretické části se zaměřuji na analýzu existujících řešení jednotlivých dílčích částí kiosk systému. Jednak budu vybírat vhodný operační systém a distribuci pro takováto vestavná zařízení, způsoby omezení uživatelského rozhraní a konfigurace systému uživatelem a také omezení internetového přístupu, jelikož kiosk se běžně používá s webovým prohlížečem.

Praktická část se věnuje testování, psaní skriptů pro správu a vytvoření modulu pro orchestrační nástroj. Výsledná sada skriptů by měla obsahovat instalační skript, který systém nastaví během nebo po instalaci, skripty pro konfiguraci prohlížeče a skripty pro údržbu a aktualizaci systému.

Kapitola 3
Existující implementace

3.1 Porteus Kiosk

Porteus Kiosk je systém určený pro použití jako kiosk systém a je momentálně v aktivním vývoji. Je založený na systému Porteus a rozšiřuje ho jen nástroji pro instalaci, nastavení a aktualizace. Zdroje těchto nástrojů a skriptů nejsou ale z bezpečnostních důvodů k dispozici. [1][2]

Je vydáván jak ve variantě místního systému, tak ve variantě tenkého klienta (dále jen *thin-client*), který se po spuštění připojí na vzdálenou plochu. *Thin-client* podporuje protokoly Citrix, RDP, NX, VNC a SSH. [3]

Při instalaci je zobrazen průvodce nastavení systému. Všechna tato nastavení jdou také určit konfiguračním souborem přiloženým v obraze instalačního média. Tento konfigurační soubor jde vytvořit buď ručně nebo je možné jej vygenerovat v průvodci nastavení. Konfigurační soubor má formát „klíč-hodnota“. V rámci konfigurace je možné nastavit typické vlastnosti jako nastavení sítě, Wake-on-LAN, firewall, povolné stránky, jazyk a lokalizaci atd. Také jde určit, který prohlížeč má systém používat. Na výběr jsou Google Chrome a Mozilla Firefox. [4]

Systém jde rozšířit za pomoci `.xzm` modulů. Tyto moduly jsou archívy ve formátu Squashfs, obsahující požadované soubory. Tyto soubory se přikládají do obrazu instalačního média do složky `/xzm`. Při instalaci se všechny tyto moduly aplikují v systému.

Ačkoliv je Porteus Kiosk volně dostupný ke stažení a k používání, některé části systému jsou placené, například upravený systém od vydavatelů systému a automatické aktualizace.

3.2 Webconverger

Webconverger je open-source kiosk systém založený na systému Debian. Nenakonfigurovatelná verze je vydávána volně ke stažení. Pro jakékoliv nastavení systému je nutné zakoupit jednorázovou nebo dlouhodobou licenci. Jednorázová licence je vázána na konkrétní hardware konfiguraci. Při změně součástek počítače kiosku je tedy nutné licenci zakoupit znovu. Dlouhodobá licence je vázána pouze na počet aktivních zařízení, nehledě na hardware konfiguraci. Dle aktuálního ceníku je cena za jedno zařízení stanovena na $200 jednorázově nebo $100 ročně. [5][6]

Pro konfiguraci systému je nutná registrace na stránkách vývojářů. Při prvním spuštění vyzývá uživatele k vyplnění registrovaného e-mailu. Tím se počítač přihlásí na servery vývojářů. Poté je možné skrze jejich stránky systém nastavit.

Pro automatické aktualizace využívá systém Git. Celý souborový systém je obsažen v Gitovém repozitáři, který je při aktualizaci klonován.

Kromě dostupných nastavení není možné systém jakkoliv rozšířit. Jediná možnost by byla zduplikovat Git repozitář a provést úpravy v něm.

Kapitola 4

Analýza dílčích částí systému

4.1 Operační systém

Operační systém (dále jen OS) je platforma která umožňuje spuštění služeb a nástrojů. Typicky je součástí OS jádro, správce služeb (často nazývaný „init systém“), knihovny pro běh programů, nějaké rozhraní pro komunikaci s uživatelem a sada nástrojů pro práci na počítači. V rámci kiosk systému je však nutné, aby uživatel mnoho částí systému používat nemohl. To by sice naznačovalo, že jsou tyto části v systému nežádoucí, ale zpravidla jsou tyto nástroje nezbytné pro správce (případně pro spravující orchestrační systém).

Pro tuto práci použiji OS GNU/Linux, jelikož Linux je jeden z nejpoužívanějších OS v odvětví serverů a vestavných zařízení. Teoreticky by nejspíše bylo možné vytvořit kiosk i na platformách BSD nebo Solaris, ale za účelem snazší rozšířitelnosti a správy použiji právě GNU/Linux. [7][8][9]

4.1.1 GNU/Linux a jeho distribuce

GNU/Linux je obecné označení pro systém, který používá Linux jako jádro OS a sadu nástrojů GNU. Neexistuje žádné tzv. „originální vydání“. GNU/Linux je svobodný software, tudíž jej může kdokoliv zkopírovat, upravit a publikovat. Z tohoto důvodu vznikl trend vytváření tzv. „linuxových distribucí“. Tyto distribuce jsou zpravidla pouze mírně upravené jádro a nástroje společně s vlastní sadou nástrojů pro specifickou správu systému. Z pohledu uživatele (a tedy i koncových aplikací) jsou běžně tyto distribuce téměř nerozlišitelné. [10]

Rozdílnosti mezi distribucemi jsou především mezi dostupným software. Software se na Linuxových systémech instaluje v podobě balíčků, což jsou v podstatě archívy obsahující instalovaný software a popř. skripty spouštěné před nebo po rozbalení archívu (často zvané „pre-hooks“ a „post-hooks“). Tyto balíčky jsou z pravidla k dispozici ze serverů zvaných repozitáře.

4.1.1.1 Debian

Debian je jeden z nejčastěji používaných distribucí. Téměř veškerý obsah jeho repozitáře je svobodný software. Skupina zodpovědná za Debian se zároveň zavazuje, že Debian bude vždy volně dostupný a svobodný software. [11]

Debian je udržován vždy ve třech vývojových větvích - *Stable*, *Testing* a *Unstable*. Každá verze Debianu postupně prochází těmito větvemi a každé 2 roky se tyto větvě mezi sebou posouvají. *Stable* je stabilní vydání systému, které již dostává pouze bezpečností aktualizace a nedostává aktualizace na nové verze software. Software dostupný v této větvi většinou bývá starý, což je ale záměr za účelem větší spolehlivosti. *Testing* je verze v testovací fázi. Dostává některé

nové (spíše vedlejší) aktualizace a jejím cílem je odchycení funkčních nebo bezpečnostních chyb v dostupném software. Běžně je dostatečně spolehlivý pro použití jako OS pro pracovní stanici. Na základě zpětné vazby těchto uživatelů se určují problémové verze programů. *Unstable*, jak název napovídá, je nestabilní verze poskládaná z nejnovějších balíčků, které však mohou být vzájemně nekompatibilní. [12]

Ačkoliv tedy každé 2 roky vychází nová stabilní verze, předchozí stabilní verze zvaná *Old stable* vždy dostává dodatečné minimální aktualizace ještě další 3 roky. Životnost jednoho vydání je tedy 5 let. Pro usnadnění údržby existujících systémů je také možné Debian ve stavu *Old stable* aktualizovat na následující verzi *Stable*.

4.1.1.2 Ubuntu

Ubuntu je distribuce založena na větvi Debian *Unstable* a je vydáván společností Canonical Ltd. Je běžněji používán na pracovních stanicích a osobních počítačích než Debian. Typicky mívá novější verze software oproti Debianu, ale v mnoha ohledech je mu velice podobný.

Nové vydání Ubuntu vychází každých 6 měsíců (v dubnu a říjnu) a je podporované následujících 9 měsíců. Každé čtvrté vydání (po dvou letech) ale získává prodlouženou podporu na 5 let, jako tzv. LTS vydání. [13]

4.1.1.3 Alpine Linux

Alpine Linux je distribuce určená pro vestavná zařízení. Cílem této distribuce je jednoduchost a bezpečnost. Nainstalovaný systém má typicky okolo 130MiB. Obsahuje pouze nutné nástroje a na rozdíl od většiny systémů používá *init systém* OpenRC namísto SystemD. Tyto rozdílnosti existují za účelem jednoduchosti, ale zároveň to může být komplikace pro administrátory, kteří jsou zvyklí na plně vybavené systémy se SystemD. [14][15]

4.1.2 Výběr systému

Pro vypracování bakalářské práce jsem vybral systém Debian. Je to univerzální distribuce a má bohatou dokumentaci a komunitu. Ačkoliv bude komplikovanější takový systém zabezpečit, správa a úprava systému bude na takové distribuci značně jednodušší.

4.2 Prohlížeč

Webový prohlížeč je zásadní část celého systému, jelikož je to ta část, se kterou bude uživatel interagovat. Schopnosti zabezpečení v rámci prohlížeče diktují i jaké části systému je nutné zabezpečit samostatně. Je nutné, aby prohlížeč uměl blokovat některé stránky (resp. povoloval pouze některé stránky). Zároveň je nutné, aby v rámci systémového nastavení prohlížeče bylo možné zamezit uživatelskému nastavení prohlížeče.

4.2.1 Mozilla Firefox

Mozilla Firefox (dále jen Firefox) je svobodný webový prohlížeč od společnosti Mozilla, vyvíjený pod licencí MPL 2.0 (Mozilla Public License). MPL dovoluje využití bez velkého omezení a to i s vlastním proprietárním kódem (omezení copyleft platí pouze pro úpravy souborů existujících ve zdrojích Firefoxu). [16]

Firefox má kiosk režim, ve kterém se zobrazí přes celou obrazovku bez hlavičky okna, tedy i bez adresního řádku nebo přístupu do nastavení a přizpůsobení. Tento režim se také zpravidla používá společně s anonymním režimem, aby po ukončení nebyla uložena historie nebo

soubory cookies. Firefox v kiosk režimu se tedy spustí `firefox --kiosk --private-window "https://www.example.com/"`. [17]

Firefox je možné nastavit za pomoci politik. Tyto politiky se nastavují odlišně na různých systémech, ale mají na všech platformách ekvivalentní strukturu. Na Linuxu se zapisují do `/etc/firefox/policies/policies.json` v níže uvedeném formátu 4.1. Těmito politikami lze nastavit seznam povolených stránek, blokování uživatelského nastavení a nastavení nebo vypnutí doplňkových funkcí prohlížeče. [18]

Výpis kódu 4.1 Firefox - Formát politik

```
{
    "policies": {
        "Jméno politiky": {
            "Vlastnost1": "hodnota",
            "Vlastnost2": ["hodnota1", "hodnota2"]
        }
    }
}
```

4.2.1.1 Blokování stránek

Firefox dovoluje blokování stránek za pomoci politiky `WebsiteFilter`. Tato politika obsahuje seznam zakázaných adres a seznam výjimek ze zákazu na principu „Allow all, Deny some, except“, tedy že povolí přístup na všechny stránky, ale zakáže všechny stránky v seznamu „Block“ kromě stránek v seznamu „Exceptions“ 4.2. [18]

Zároveň může „Block“ obsahovat záznam `<all_urls>`, který zakáže všechny stránky. V takovém případě jsou dostupné pouze stránky v seznamu „Exceptions“ 4.3. V kiosku by se běžně použil pouze seznam povolených stránek, na což by se použila tato varianta.

Záznamy v seznamech blokovaných a povolených stránek se píší v porovnávacích vzorech pro URL. [19]

Výpis kódu 4.2 Firefox - Z `www.example.com` zakáže vše kromě `/internal/*` a `/api/*`

```
{
    "policies": {
        "WebsiteFilter": {
            "Block": ["*://.www.example.com/*"],
            "Exceptions": [
                "*://.www.example.com/internal/*",
                "*://.www.example.com/api/*"
            ]
        }
    }
}
```

Výpis kódu 4.3 Firefox - Zakáže vše kromě `www.example.com`

```
{
    "policies": {
        "WebsiteFilter": {
            "Block": ["<all_urls>"],
            "Exceptions": ["*://.www.example.com/*"]
        }
    }
}
```

4.2.2 Chromium

Chromium je svobodný prohlížeč od společnosti Google. Licencovaný je pod licencemi BSD, MIT, LGPL, GPL, MS-PL a MPL (různé části zdrojů spadají pod různé licence). Známý prohlížeč Google Chrome je nadstavba na prohlížeč Chromium, tedy Chromium buď některé funkce Chromu neobsahuje nebo nejsou dostupné v základním nastavení překladu. Tyto chybějící funkce jsou mimo jiné integrace některých služeb společnosti Google, Widevine DRM modul, Flash modul a sledování a metriky uživatelů. Všechny podstatné funkce Chromu Chromium obsahuje. [20]

Chromium má také kiosk mód, který funguje obdobně jako kiosk režim Firefoxu. Spustí prohlížeč na celou obrazovku bez hlavičky nebo adresního řádku. Také je vhodné spustit prohlížeč v anonymním režimu, dovolí-li to ostatní nastavení. [21]

Chromium lze také nastavit za pomoci politik obdobně jako Firefox. Tyto politiky jsou velmi podobné jako politiky Firefoxu. Mimo jiné obsahují nastavení některých integrovaných funkcí Chromia, které by mohly v kiosku překážet, jako například integrace se službou Google Translate. Na rozdíl od Firefoxu má ale Chromium možnost nastavit jednak vynucené politiky (které uživatel měnit nemůže) a politiky doporučené. Stejně jako u Firefoxu je možné politiky nastavit různými způsoby na různých platformách a na Linuxu jsou opět ve formátu JSON4.4. Vynucené politiky se nastavují v souborech `/etc/chromium/policies/managed/*.json` a doporučené v `/etc/chromium/policies/recommended/*.json`. Pro kiosk by se využily pouze vynucené politiky, jelikož uživatel nemá mít možnost měnit žádná nastavení. [22][23]

Výpis kódu 4.4 Chromium - Formát politik

```
{
    "Jméno politiky": {
        "Vlastnost1": "hodnota",
        "Vlastnost2": ["hodnota1", "hodnota2"]
    }
}
```

4.2.2.1 Blokování stránek

Blokování stránek v Chromiu funguje v stejně jako ve Firefoxu, ale s jiným názvoslovím a syntaxí. Politika `URLBlocklist` je ekvivalentní `WebsiteFilter.Block` a politika `URLAllowlist` je ekvivalentní `WebsiteFilter.Exceptions`. Hlavní rozdíl syntaxe je možnost specifikovat TCP port a `<all_urls>` je nahrazeno samotnou hvězdičkou `*`. Seznam povolených stránek by se tedy sestavil následovně 4.5. [24]

Výpis kódu 4.5 Chromium - Zakáže vše kromě `www.example.com`

```
{
    "URLBlocklist": ["*"],
    "URLAllowlist": [".www.example.com/*"]
}
```

4.2.3 Výběr prohlížeče

Z běžně používaných prohlížečů mají podporu pro Linux hlavně Firefox a Chromium (popř. Google Chrome). Chromium má ale mnohem více funkcí užitečných pro kiosk než ostatní prohlížeče se stejným jádrem. Mezi Chromiem a Firefoxem jsem vybral Chromium. Funkcionálně jsou ekvivalentní, ale Chromium má širší nastavení. Například Firefox nelze nastavit tak, aby nedovolil stahování souborů.

4.3 Nástroje pro spuštění a zabezpečení systému

Kromě nastavení prohlížeče je také nutné zabezpečit zbytek systému a zajistit bezpečné a spolehlivé spuštění prohlížeče.

4.3.1 Zabezpečení souborového systému

Teoreticky by bylo vhodné zabezpečit souborový systém proti změnám. Pokud ale za běhu systému může uživatel pracovat pouze s prohlížečem, nemá taková ochrana smysl. Jelikož prohlížeč bude spuštěn pod neprivilegovaným uživatelským účtem, bude moci upravovat pouze obsah své domovské složky. Tudíž, i kdyby uživatel dokázal zavřít prohlížeč, nemohl by systém významně ovlivnit. Mohl by pouze zaplnit složku daty a tedy zaplnit souborový systém, čemuž se ale dá zabránit kvótami. [25]

Zároveň je vhodné vždy před ukončením systému domovskou složku promazat. Prohlížeč do domovské složky ukládá mezipaměť, která by teoreticky mohla po dlouhé době souborový systém zaplnit. Zároveň je v krajním případě možné, že by se útočníkovi povedlo vytvořit v domovské složce konfigurační soubory, které by byly při následujícím spuštění použity.

Pravidelné promazání domovské složky se dá zajistit službou v SystemD vytvořením *one-shot* služby. Obdobného celkového výsledku jde také docílit připojením domovské složky jako RAMdisk. Jelikož RAMdisk je ukládaný v hlavní paměti, bude promazán při vypnutí systému. Zároveň je možné připojenému RAMdisku určit maximální velikost, což je obdoba kvóty. [26]

4.3.2 Zabezpečení zavedení systému

Debian používá pro spuštění systému zavaděč GRUB2. Ten typicky při spuštění počítače zobrazí menu s možnostmi spuštění a umožní jejich dočasnou úpravu. To je ale nežádoucí. Po spuštění počítače by se měl systém spustit bez možnosti úpravy jeho parametrů. Toho lze triviálně docílit nastavením čekání zavaděče na 0 sekund. Toto nastavení se určí řádkou `GRUB_TIMEOUT=0` v souboru `/etc/default/grub` a následným spuštěním příkazu `update-grub`. [27]

4.3.3 Uživatelský účet

Pro spuštění prohlížeče je vytvořen uživatel *kiosk*, který nemá nastavené heslo, shell ná nastavený na `/usr/sbin/nologin` a je členem pouze skupiny *video*. Pro spuštění grafického prostředí s prohlížečem nic víc nepotřebuje.

4.3.4 Virtuální terminály

Při spuštění systému je nutné, aby se automaticky spustil i prohlížeč. Toho lze docílit za pomoci virtuálního terminálu (dále jen VTY). Základní instalace Debianu obsahuje vícero VTY a je možné mezi nimi přepínat klávesovými zkratkami Ctrl+Alt+F1,F2,... Jejich počet nastavuje služba LoginD. Nastavením jejich počtu na 1 se zamezí přepínání na jiný, což by schovalo spuštěný prohlížeč, a zároveň brání pokusům o přihlášení pod jiným uživatelským účtem. [28]

4.3.5 Spuštění grafického serveru a prohlížeče

Grafický server X11 (dále jen X11) je možné spustit příkazem `startx`. Ten spustí X11 a následně i skript, který spustí grafické rozhraní nebo uživatelskou plochu. Může být také nastaven tak, aby spustil pouze jednu grafickou aplikaci. Skript, který má toto vykonat, je v domovské složce v `~/.xinitrc`. Pokud tento soubor neexistuje, načte X11 `/etc/X11/xinit/xinitrc`. Zároveň je možné mu předat cestu k tomuto skriptu jako argument. [29]

Také je možné v `xinitrc` spustit kompozitor a prohlížeč spustit až následně v něm. Tím je možné se vyvarovat některých nežádoucích stavů, jelikož aplikace typicky neumí komunikovat přímo s X11. Spuštěný kompozitor by ale měl být buď co nejtriviálnější nebo by měl mít možnost svou funkcionalitu co nejvíce omezit. Dobrý kandidát pro takovou situaci je kompozitor OpenBox. V základní konfiguraci nemá žádné dodatečné grafické elementy a veškeré jeho klávesové zkratky jdou snadno zrušit v jeho konfiguračním souboru. Zároveň po načtení spouští skript `/etc/xdg/openbox/autostart`, ve kterém může být spuštěn prohlížeč. Společně s tím je možné v tomto skriptu nastavit vlastnosti X11, jako přemapování kláves a nastavení spořiče obrazovky za pomoci příkazů `xset` a `xmodmap`. [30][31][32]

Další možnost je spustit X11 za pomoci software zvaný *display manager* (dále jen DM). Na pracovních stanicích je DM zodpovědný za přihlašovací dialog po spuštění počítače. Mnohé se ale dají nastavit s automatickým přihlašováním. DM běžně spouští kompozitor, ale může také přímo spustit X11. [33]

Jeden z nejjednodušších DM je *lightdm*. Lze jej s modulem *lightdm-autologin-greeter* nastavit tak, aby přihlašovací dialog neukazoval nikdy a přímo spouštěl X11. *Lightdm-autologin-greeter* spouští některou relaci X11 určenou nastavením v `lightdm-autologin-greeter.conf`. Relace X11 jsou v Debianu definované soubory v `/usr/share/xsessions`. [34]

Pro implementaci kiosku použiji variantu s *lightdm* a OpenBox. Z výše uvedených variant je nejspolehlivější.

4.3.6 Vstupní zařízení

Klávesnice a myš jsou zařízení, které v Linuxu nevyžadují žádné dodatečné ovladače. Hardware klávesnice potřebují pouze určit rozložení. To jde určit v souboru `/etc/default/keyboard` upravením hodnot `XKBLAYOUT` a `XKBVARIANT`. [35]

Dotykové obrazovky jako vstup ukazatele v dnešní době také nevyžadují žádné dodatečné ovladače. Pro případ dotykové obrazovky je ještě nutné zajistit textový vstup klávesnicí na obrazovce (dále jen OSK). OSK může zprostředkovávat buď kompozitor, prohlížeč, nebo samostatný program.

V případě kompozitoru je na výběr pouze GNOME. Ostatní kompozitory nemají funkci OSK aktivně vyvíjenou. Pro GNOME ale nejdou zakázat grafické elementy plochy, které by narušovaly funkčnost kiosku.

Prohlížeče Chromium ani Firefox neimplementují OSK přímo, ale existují doplňky pro OSK. Pro Chromium existuje *xontab/chrome-virtual-keyboard* pod licencí MIT. Již není aktivně vyvíjený, ale zdá se být plně funkční a stále používaný. [36]

OSK jako samostatné programy jsou například již nevyvíjené *onboard* nebo *florence*. Obě možnosti v kombinaci s prohlížečem nefungují. Prohlížeč se vykresluje přes celou obrazovku a nedokáže předat (pravděpodobně záměrně) *focus* jiným oknům. Z tohoto důvodu se tyto programy nezobrazí před oknem prohlížeče, což je dělá nepoužitelnými. [37][38]

Pro OSK jsem vybral doplněk prohlížeče jako jedinou funkční možnost.

4.3.7 USB Úložiště

Běžně operační systém automaticky rozpoznává USB úložiště (flash disky atd.) a snaží se připojit jejich souborový systém. Automatické připojování souborových systémů těchto zařízení vykonává služba FUSE. Tu je možné jednoduše zrušit zakázáním kernel modulu `fuse`. Obdobně i rozpoznávání USB úložiště je možné zamezit zakázáním modulu `usb-storage`. [39]

4.3.8 Automatická obnova prohlížeče

Uživatel kiosku může nechat kiosk na jiné webové stránce, než jaká je nastavená jako prvotní (třeba stránka konkrétního produktu e-shopu). Může se také zapomenout odhlásit ze svého účtu. Je tudíž nutné prohlížeč pravidelně restartovat. To by se mělo dít vždy několik minut poté, co uživatel přestane s kioskem pracovat. K tomu může být použit `xautolock`, který dostává od X11 informace o vstupních zařízeních. Pokud po určenou dobu vstupní zařízení nic neposílají, `xautolock` spustí určený program nebo skript. Ten pak může triviálně ukončovat prohlížeč příkazem `pkill $BROWSER`, jelikož prohlížeč bude mít vždy spuštěnou pouze jednu instanci. [40]

4.3.9 Nastavení času

Pro správné fungování šifrovaných spojení TLS je nutné korektní nastavení času. To vyžaduje správné nastavení služby NTP. Službu NTP je buď možné doinstalovat nebo použít vestavěnou službu NTP v SystemD. Klasický ISC NTP klient je dostupný v balíčku *ntp* a nastavuje se v souboru `/etc/ntp.conf`. NTP služba v SystemD (*systemd-timesyncd*) je součástí SystemD, tedy není nutné ji doinstalovat. Pro jednoduchost bude mé řešení používat *systemd-timesyncd*. [41]

4.3.10 Síť a firewall

Jelikož pro administraci kiosk zařízení bude použitý orchestrační nástroj, nastavené síťové rozhraní je předpoklad pro funkčnost mého řešení. Tudíž budu předpokládat již nastavené síťové rozhraní.

Požadovaná funkcionalita kiosku může také být připojování k webovému serveru v místní síti. Takové konfigurace z pravidla závisí na DNS serveru v místí síti. Ten je možné nastavit v souboru `/etc/resolv.conf`.

Pro zabránění případným pokusům o neoprávněné připojení na kiosk vzdáleně, je nutné nastavit na zařízení firewall. Zároveň se může stát, že některý software (konkrétně Chromium) se bude pokoušet připojovat na různé vzdálené služby. Jediná komunikace, která se na kiosku předpokládá, jsou protokoly HTTP (bez/včetně TLS), DNS, NTP a SSH.

Debian od verze 10 používá implicitně firewall *nftables*, namísto předešlého *iptables*. Jejich funkčnosti jsou velmi podobné. [42][43]

4.3.11 Automatické aktualizace

Ačkoliv stabilní větev Debianu nedostává aktualizace s novými funkcionalitami, je doporučované jej aktualizovat. Aktualizace, které dostává, jsou z pravidla bezpečnostní. Instalace nových aktualizací jde triviálně vyvolat příkazy `apt-get update -y -qq && apt-get upgrade -y -qq`. Občas mohou aktualizace vyžadovat restart systému pro správnou funkčnost. Proto není vhodné aktualizace provést v průběhu používání kiosku. Ideální řešení by tedy bylo provést automatické aktualizace těsně před vypnutím systému.

Pokud je čas vypnutí systému předem známý, mohou být aktualizace provedené službou cron. Služba cron spouští nastavené příkazy v pravidelných časech. Úkony cronu jsou definované v souboru `/etc/crontab`. Tyto pravidelné časy jsou uváděny v níže uvedeném formátu včetně wildcard formátu. [44]

- Minuta z hodiny <0,59>
- Hodina ze dne <0,24>
- Den z měsíce <1,31>

- Měsíc z roku <1,12>
- Den v týdnu <0,6> – 0 pro neděli, 1 pro pondělí, ...
- * – libovolná hodnota
- a,b,c – výčet hodnot
- a-b – rozsah hodnot
- */n – každé n-té opakování

Níže uvedený příklad provede příkaz `/bin/example.sh` každých 10 minut mezi 10:00 a 15:59, každý všední den, celý rok vyjma června a července.

Výpis kódu 4.6 Crontab formát

```
*/10  10-15  *  1-5,8-12  1-5  /bin/example.sh
```

Použití služby cron ale není úplně spolehlivé, obzvlášť pokud čas vypnutí systému není definitivní. Navíc se může stát, že signál pro vypnutí systému přijde v průběhu vykonávání příkazu cronu. Tehdy vypnutí systému tento příkaz přeruší. V případě aktualizací tak může dojít k poškození systému.

Další možnost je použití služby v SystemD, která vykoná příkaz pro aktualizaci systému. Takovou službu jde nadefinovat, aby se spustila před ukončením systému. Při vypínání systému SystemD čeká na úspěšné ukončení spuštěných služeb. Tím se zajistí, že se aktualizace provede korektně a vždy před vypnutím kiosku.

4.3.12 Vzdálená správa

V kiosku by nemělo být možné zobrazit nic jiného než prohlížeč. Tím pádem ale správce nemůže mít možnost spravovat systém kiosku fyzickým přístupem. Jediná možnost je tedy spravovat systém vzdáleně. Nejběžnější způsob vzdálené správy je SSH. Pro vzdálenou správu přes SSH je doporučované používat uživatelský účet jiný než *root*. Při instalaci Debianu je nutné vytvořit uživatelský účet. Budu tedy předpokládat, že už některý administrátorský uživatelský účet v systému existuje.

4.3.13 Orchestrační systém

Kromě samostatných skriptů je cílem práce také využití některého orchestračního systému. Orchestrační systém je řešení pro centralizovanou správu několika zařízení. Na jednom zařízení běží serverová služba systému, která se připojuje na spravované stanice. Tyto stanice nastavuje podle deklarativních vzorů.

4.3.13.1 Puppet

Puppet je jeden z nejpoužívanějších orchestračních systémů. Je vyvíjen pod licencí Apache 2.0 společností Puppet, Inc. Je k dispozici také placená verze, která obsahuje i webové rozhraní pro snazší administraci. Umí spravovat jak Unix-like systémy, tak Microsoft Windows. Pro vzdálený přístup používá službu zvanou agent, která běží na spravovaných zařízeních. Tu je nutné předem na zařízení nainstalovat. [45]

Pro popis vzorů zařízení používá vlastní deklarativní jazyk. 4.7 Pro komplikovanější akce je nutné napsat vlastní modul. Tyto moduly jsou psané v jazyku Ruby. Mnoho modulů je k dispozici z veřejného repozitáře Puppet forge. [46]

Výpis kódu 4.7 Deklarativní jazyk Puppetu

```
typ {'jmeno':
    vlastnost => hodnota
}

# Definice uzivatele
user {'randall':
    ensure => present
    comment => 'Randall Munroe'
    home => '/home/randall',
    password => 'Tr0ub4dor&3',
    shell => '/bin/bash'
}
```

4.3.13.2 Ansible

Ansible je orchestrační systém od společnosti RedHat, vyvíjený pod licencí GPL 3.0. Spravovat umí také Unix-like systémy i Windows, ale nepoužívá agenta. Cílové stanice musí mít pouze interpret jazyku Python verze 2.4 a některý standardní způsob připojení. Na Unix-like stanice se připojuje za pomoci SSH, zatím co na Windows používá protokol WS-Management. Pro Ansible také existuje webové rozhraní Ansible AWX, které je také dostupné pod svobodnou licencí. [47]

Pro vzory používá jazyk datový YAML. Ansible definuje role jako samostatné celky vlastností systému. Dále definuje *Playbook*, který obsahuje parametry rolí a seznamy cílových zařízení. 4.84.9 Každá vlastnost je popis pro některý modul. Některé moduly jsou vestavěné v Ansible, další jsou k dispozici z veřejného repozitáře Ansible galaxy. [48]

Výpis kódu 4.8 Ansible role

```
# roles/common.yml
-   name: Add user randall
    ansible.builtin.user:
        name: randall
        comment: "Randall Munroe"
        home: /home/randall
        password: "correct horse battery staple"
        shell: /bin/bash
-   name: Install web server
    ansible.builtin.apt:
        name: "apache2"
        state: present
```

Výpis kódu 4.9 Ansible playbook

```
-   hosts: all
    roles:
    -   common
```

4.3.13.3 SaltStack

SaltStack je orchestrační systém od Thomase S. Hatche. Je vyvíjený pod licencí Apache 2.0. Obdobně jako Puppet, umí spravovat Unix-like i Windows a používá agenta, ale pro zařízení se službou SSH agenta používat nemusí. [49]

Vzory SaltStack také definuje v jazyku YAML, ale v jiné struktuře 4.10. Zároveň je možné používat již existující moduly dostupné z kolekce *SaltStack Formulas* na GitHub. [50]

Výpis kódu 4.10 SaltStack modul

```
randall:
    user.present:
        - fullname: "Randall Munroe"
        - home: /home/randall
        - shell: /bin/bash
```

4.3.13.4 Výběr orchestračního systému

Pro svou práci jsem vybral systém Ansible, jelikož pracuje bez agenta a má bohatou komunitu. Jeho používání je navíc velmi zjednodušeno volně dostupným webovým rozhraním.

Kapitola 5
Realizace systému

Pro instalaci a správu kiosku předpokládám čistou instalaci Debianu 10 bez grafického rozhraní. Pro správu systémem Ansible je ještě nutné na systém předem nainstalovat a nastavit službu SSH a interpret Python.

Následující informace budou parametry instalace, tedy parametry instalačního skriptu a role Ansible.

- Ovladač grafického adaptéru.
- Použití dotykové klávesnice. Pokud zařízení bude mít klávesnici, tak není OSK nutná.
- Domovská stránka prohlížeče, která bude zobrazovaná po spuštění systému.
- Povolené stránky prohlížeče, jelikož proklikáváním webové stránky se teoreticky může uživatel dostat na jiné.
- Zakázání USB úložiště, jelikož systém může být uložen na flash disku a startovat z něj.
- Rozložení klávesnice.
- Velikost ramdisku domovské složky.

5.1 Instalační skript

K čisté instalaci Debianu jsou ještě doinstalovány následující balíčky. Verze odpovídají aktuálním dostupným balíčkům v Debianu 10 ke dni 15. 4. 2021. Z principu návrhu vývojových větví Debianu ale nezáleží na konkrétních verzích. Zároveň se nainstaluje některý grafický ovladač. Neurčí-li uživatel parametrem, nainstalují se univerzální ovladače VESA. K dispozici jsou také ovladače *nouveau*, *intel* nebo *amdgpu*.

Před instalací nových balíčků se nejprve systém zaktualizuje.

- `chromium 89.0.4389.114-1` – Webový prohlížeč
- `lightdm 1.26.0-4` – Display manager
- `lightdm-autologin-greeter 1.0-3` – Modul pro lightdm zodpovědný za automatické přihlášení bez zobrazení přihlašovacího dialogu
- `nftables 0.9.0-2` – Nástroj pro nastavení firewall systému
- `openssh-server 1:7.9p1-10` – SSH server pro vzdálenou správu

- `sudo 1.8.27-1` – Nástroj pro spouštění programů s právy jiného uživatele
- `xautolock 1:2.2-5.1` – Nástroj pro automatické zamykání (použitý pro restartování prohlížeče)
- `xorg 1:7.7` – Systém X.Org (také zvaný X11)
- `xserver-xorg-input-libinput 0.28.2-2` – Univerzální ovladač pro vstupní zařízení
- `xserver-xorg-video-*` – Grafický ovladač
  - `vesa 1:2.4.0-1` – Univerzální grafický ovladač VESA
  - `amdgpu 18.1.99` – Grafický ovladač pro grafické adaptéry od společnosti AMD/ATI
  - `intel 2:2.99.917` – Grafický ovladač pro grafické adaptéry společnosti Intel
  - `nouveau 1:1.0.16-1` – Grafický ovladač Nouveau pro grafické adaptéry společnosti Nvidia

5.2 Nastavení X11

Mohou existovat funkcionality, které nejdou zakázat nebo vypnout, vyvolávané klávesovými zkratkami. V tom případě je jediné východisko přemapovat tyto klávesové zkratky na některé neškodné klávesy, jako např. klávesu escape. Ačkoliv `xmodmap` dovoluje přemapovat klávesy na nic (tedy je vypnout), tak jsou některé klávesy, které musí být na něco namapované.

Při testování jsem zjistil, že v Chromiu fungují záložky i v kiosk módu. Nefunguje vyvolávání nových záložek nebo oken webovou stránkou, ale ta funkcionalita jako taková vypnutá není. Při stisknutí klávesy F1 se zobrazí nápověda prohlížeče jako nová záložka. Po jejím vyvolání je možné přepínat mezi ní a původní klávesovou zkratkou Ctrl+Tab.

Zároveň je stále možné přepínat na jiné VTY. Nastavení VTY je sice zruší, ale zdá se, že lightdm je opět vytvoří, ačkoliv jen jako prázdné obrazovky. To ale nic nemění na tom, že přepnutím na jiné VTY může uživatel schovat prohlížeč.

Z tohoto důvodu jsem ve skriptu `/etc/xdg/openbox/autostart` přemapoval všechny klávesy F1 až F24 na Escape. Zároveň jsem přemapoval klávesu Tab, aby nebylo možné používat klávesovou zkratku Alt+Tab.

5.3 Nastavení Chromia

I v kiosk módu Chromia je mnoho nežádoucích funkcionalit, které stále fungují. Tyto funkcionality jsou zakázány/vypnuty politikami. Ačkoliv kiosk mód může již některé z těchto funkcionalit zakazovat, jejich dodatečné zakázání neuškodí. Proto jsem rozhodl zakázat níže uvedené funkcionality explicitně v nastavení Chromia.

- Upozornění
- Vyskakovací okna
- Google Cast
- Dialog pro výběr souborů
- Automatické vyplňování formulářů
- Integrace s Google překladačem
- Přihlašování do prohlížeče
- Vestavěný DNS klient (pro případ nutnosti vlastního DNS serveru)

- Snímky obrazovky
- Stahování souborů
- Tisk
- Nahrávání zvuku nebo videa (běžně používané pro videohovory)
- Správce hesel
- Vzdálený přístup
- JavaScript filesystem/sensors/bluetooth/usb/serial API

Pro podporu OSK instaluji do prohlížeče doplněk *xontab/chrome-virtual-keyboard*. Instalaci provádím určením politiky `ExtensionInstallForcelist`.

Pro blokování nepovolených stránek používám politiky `URLBlocklist` a `URLAllowlist`. V případě, že administrátor nezadal blokované adresy, blokuji všechna schémata kromně `http` a `https`. Tím zajišťuji, že se nespustí žádný program pro odbavení těchto schémat. Konkrétně jsem narazil na problém se chématem `mailto`, které otvíralo nové okno prohlížeče.

5.4 Spouštění a obnovování prohlížeče

Prohlížeč spouštím ve skriptu `/etc/xdg/openbox/autostart`. Spouští se v nekonečném cyklu, aby se spouštěl znovu v případě, že bude ukončen nebo spadne. Domovskou stránku zapisuji do souboru `/etc/kiosk/homepage`, ze kterého poté ve skriptu čtu.

Pro obnovování prohlížeče na domovskou stránku jsem použil `xautolock`. Spouštím ho s takovými parametry, aby každých 5 minut neaktivity kiosku zavolal skript `/usr/local/kiosk/restart_chromium.sh`. Aby se prohlížeč neobnovoval každých 5 minut, zkontroluje skript čas posledního volání. Pokud byl naposledy zavolán před 5 minutami (a 3 sekundami kvůli nepřesnosti) tak nic nedělá, pouze zapíše nový čas do pomocného souboru. Pokud byl zavolán před delší dobou než 5 minutami, ukončí všechny instance Chromia. To způsobí opětovné otevření prohlížeče s domovskou stránkou. Čas posledního volání skript ukládá do souboru `/tmp/m_xautolock_time`. Tímto řešením se vyhnu opakovanému obnovování prohlížeče, pokud není dlouho používaný.

5.5 Firewall

Nastavení firewallu jsem zakládal na vzorovém nastavení pro pracovní stanice na ArchWiki. [43]

Toto nastavení:

- přijímá lokální příchozí nebo odchozí komunikaci,
- zahazuje neplatnou komunikaci TCP a UDP,
- přijímá již navázanou příchozí nebo odchozí komunikaci,
- přijímá IPv6 ICMP, jelikož dle standardu pro IPv6 je toto povinné,
- povoluje odchozí komunikaci pro HTTP, HTTPS, DNS a NTP,
- povoluje příchozí komunikaci pro SSH,
- zakazuje veškerou ostatní příchozí, odchozí i průchozí komunikaci.

5.6 Zakázání USB úložiště

V systému zakazuji modul `fuse` a, pokud to určuje nastavení, taky `usb-storage`. Klasicky se moduly zakazují v souborech `/etc/modprobe.d/*.conf` příkazem `blacklist $MODULE`. V takovém případě ale může být stále modul jádra načten, vyžaduje-li ho jiný načítaný modul jádra. Proto moduly zakazuji příkazem `install $MODULE /bin/true`. Tím se zajistí, že se tyto moduly zaručeně nenačtou.

5.7 Ramdisk v domovské složce

Pro nezachování domovské složky napříč běhy systému ji při spuštění připojuji jako ramdisk. Toho jsem docílil definicí v souboru `/etc/fstab`. Připojuji ho s níže uvedenými možnostmi.

- `rw` – Připojení pro čtení i zápis
- `size=$RAMDISK_SIZE` – Limit velikosti na zábranu zaplnění paměti
- `noexec` – Zákaz spuštění programů
- `nodev` – Zákaz interpretace blokových nebo znakových zařízení v podsložkách
- `nosuid` – Zákaz nastavení *setuid*
- `uid=kiosk` a `gid=kiosk` – Nastavení vlastníka a vlastnické skupiny

5.8 Nastavení Openbox

Openbox se nastavuje v souborech `/etc/xdg/openbox/rc.xml` a `/etc/xdg/openbox/menu.xml`. V těchto souborech se nastavují klávesové zkratky a grafické elementy Openboxu. Openbox implicitně nemá téměř žádné klávesové zkratky nebo grafické elementy. Z tohoto důvodu tyto soubory jen přejmenovávám s příponou `*.disable`.

5.9 Realizace systémem Ansible

Pro správu systémem Ansible musí spravované zařízení mít nainstalovaný interpret Python. Tento interpret musí být verze 2.4 a vyšší nebo 3 a vyšší. To ale není problém, jelikož v repozitáři Debianu 10 je k dispozici Python 3.7.3 a 2.7.16. Zároveň musí mít spravovaný systém spuštěnou službu SSH.

Z tohoto důvodu jsem také připravil skript, který na systém nainstaluje Python 3 a OpenSSH server a nastaví a spustí SSH server.

Připravil jsem roli pro systém Ansible. Role bere stejné parametry jako instalační skript. Tyto parametry jsou předávány za pomoci proměnných. Výchozí hodnoty těchto parametrů jsou nastavené stejně jako u instalačního skriptu. Administrátor by tuto roli zavolal ve vlastním Playbook.

Před použitím role je nutné doinstalovat moduly/kolekce modulů z Ansible Galaxy. Používám kolekce modulů *community.general* na úpravu souborů ve formátu INI a *ansible.posix* pro spolehlivou úpravu `/etc/fstab`. Obě kolekce mají licenci GPL 3.0. Tyto požadavky jsou nadefinované v souboru `roles/kiosk/requirements.yml`. Ansible dovoluje nainstalovat moduly a kolekce dle souboru příkazem `ansible-galaxy role install -r requirements.yml`. [51][52][53]

Kapitola 6

Porovnání s existujícími řešeními

6.1 Rozšířitelnost

Problém existujících řešení je limitovaná, až žádná rozšířitelnost přes meze nastavení. Jejich úprava není nemožná díky svobodné licenci těchto řešení. Avšak úprava těchto systémů je nadmíru komplikovaná i pro zkušené administrátory.

Mé řešení je založené na univerzální distribuci Linuxu. To dovoluje administrátorům doinstalovat libovolné balíčky dostupné v repozitářích. Administrátor se může připojit přes SSH a systém jakkoliv upravit instalací dostupného software a změnou nastavení. Použití univerzální distribuce také znamená, že administrátoři budou pravděpodobně schopni takový systém spravovat.

6.2 Licence

Existující řešení vyžadují pro používání některých vlastností systému zakoupení licence. V případě systému Webconverger je nutné zakoupit licenci pro jakékoliv smysluplné používání. To dle aktuálního ceníku činí buď $200 za zařízení jednorázově nebo $100 za zařízení ročně.

Porteus Kiosk je v omezeném rozsahu možné používat zdarma. Placená varianta zahrnuje automatické aktualizace a dle aktuálního ceníku stojí €40 za zařízení ročně s případnými množstevními slevami. Také je možné za poplatek od Porteus Solutions získat na míru upravený systém. Dle jejich stránek se cena této služby určuje až při domluvě a neuvádí ani orientační ceník.

Mé řešení je v rámci Bakalářské práce uvolněno pod licencí obdobnou LGPL, viz prohlášení.

6.3 Rozhraní

Porteus Kiosk i Webconverger mají v rozhraní vidět hlavičku prohlížeče. To sice samo o sobě není nutně bezpečností chyba, ale dle mého názoru to vypadá neprofesionálně. Zároveň to dovoluje uživateli pokusit se otevřít libovolné webové stránky.

Kromě webových stránek může uživatel otevírat stránky schématu *about:*. Ačkoliv jsem zkoušel tyto stránky otevřít, zdá se, že všechny tyto stránky, které dovolují uživateli prohlížeč nastavit, jsou nedostupné. Zobrazují se jako prázdné bílé stránky. Ty které otevřít jdou, jsou *easter egg* stránky a některé pouze informativní stránky jako `about:plugins`. V mém řešení může uživatel otvírat pouze stránky, na které se dostane zobrazenými odkazy.

Dotykový vstup Porteus Kiosk také podporuje. OSK ale nepodporuje oficiálně. Dle Porteus Solutions má pro instalaci OSK administrátor nainstalovat rozšíření do prohlížeče. Systém má

k dispozici samostatný program pro OSK, ten se ale nevyvolá při kliknutí na textové pole. Já jsem pro OSK použil také rozšíření do prohlížeče. [54]

6.4 Centralizace a škálovatelnost

Existující řešení umožňují centralizovanou správu. Webconverger se vždy konfiguruje z administrativního panelu na stránkách Webconvergeru. Konkrétní funkcionalitu tohoto řešení jsem nemohl vyzkoušet, jelikož je placená. Nevýhoda tohoto řešení je závislost na serverech Webcovergeru. V případě (administrátorem neovlivnitelného) výpadku není možné zařízení spravovat.

Porteus Kiosk existuje i ve variantě se serverem, který dodává nainstalovaným instancím konfiguraci. Toto řešení ale vyžaduje jedno zařízení navíc pro server. Systém pro server, který Porteus Solutions vydává, není upravitelný a nedá se použít pro nic jiného. Verze zdarma dovoluje pouze nastavovat spravovaná zařízení, monitorovat, zda jsou připojená, a zobrazovat verze nainstalovaného software a vlastnosti hardware. Funkce jako Wake-on-LAN, SSH, VNC nebo monitorování zdrojů systému vyžadují placenou verzi. Automatické aktualizace serveru jsou také obsažené pouze v placené verzi. Placená verze této varianty dle aktuálního ceníku stojí €300 ročně.

Mé řešení používá pro centrální správu systém Ansible. Ansible může být nainstalován na libovolném linuxovém zařízení, tedy i na již existujícím serveru v síti. Pokud administrátor potřebuje nastavit velké množství kiosků, tedy využít škálovatelnost, dá se předpokládat, že již má v síti zřízený server. V takovém případě stačí, aby administrátor nainstaloval Ansible na tento existující server. U řešení s dedikovaným serverem by musel buď zprovoznit další server, nebo ho spustit ve virtuálním stroji. To je ale časově i výkonnostně náročné.

Kapitola 7
Závěr

Cílem této práce byla primárně analýza a výběr existujících technologií pro sestavení kiosk systému s podporou pro klasický i dotykový vstup. Systém měl umožňovat centralizovanou správu a snadnou škálovatelnost a měl být patřičně zabezpečený. Praktickým výsledkem práce mělo být vytvoření instalačních a konfiguračních skriptů a modulu do některého orchestračního systému.

V první kapitole jsem prozkoumal existující implementace kiosk systému používající svobodné systémy. Zjistil jsem jejich funkčnosti a způsoby zabezpečení.

V druhé kapitole jsem zanalyzoval dostupné řešení dílčích částí kiosk systému. Provedl jsem výběr vhodného operačního systému prohlížeče. Také jsem vybíral vhodné nástroje a řešení zabezpečení systému a omezení funkčnosti pro jeho specializované použití. Obdobný výběr jsem provedl i u orchestračního nástroje zajišťujícího centralizovanou správu a škálovatelnost.

Ve třetí kapitole jsem se věnoval implementaci vlastního řešení. Pro to jsem využil zjištění z druhé kapitoly. Popisoval jsem použité technologie a jejich nastavení pro vytvoření kiosk systému. Z této implementace jsem sestavil instalační skript, konfigurační skript a roli pro orchestrační systém Ansible.

V poslední kapitole jsem srovnával své řešení s již existujícími implementacemi kiosk systému.

V těchto krocích jsem splnil zadání své práce. Výsledný systém je použitelný jak s klávesnicí a myší, tak s dotykovou obrazovkou a je škálovatelný s centralizovanou správou za pomoci systému Ansible. Zároveň je díky použití univerzálních technologií poměrně snadno rozšiřitelný zdatnými Linuxovými administrátory.

V budoucnu by mohlo být výhodné vytvořit pro centralizovanou správu i nějaké grafické rozhraní. Také by přicházelo v potaz rozšíření o možnost používat jinou aplikaci než Chromium bez omezení na vstupních zařízeních.

Bibliografie

1. PORTEUS SOLUTIONS. *Porteus Kiosk* [online]. 2021 [cit. 2021-04-15]. Dostupné z: `https://porteus-kiosk.org/`.
2. FANTHOM. *Isn't Porteus kiosk open source?* [Online]. 2015 [cit. 2021-04-15]. Dostupné z: `https://forum.porteus.org/viewtopic.php?t=4851`.
3. PORTEUS SOLUTIONS. *Porteus Kiosk variant „ThinClient"* [online]. 2021 [cit. 2021-04-15]. Dostupné z: `https://porteus-kiosk.org/thinclient.html`.
4. PORTEUS SOLUTIONS. *Kiosk Wizard* [online]. 2021 [cit. 2021-04-15]. Dostupné z: `https://porteus-kiosk.org/wizard.html`.
5. WEBCONVERGER LTD. *Webconverger* [online]. 2020 [cit. 2021-04-15]. Dostupné z: `https://webconverger.com/`.
6. WEBCONVERGER LTD. *Webconverger Pricing* [online]. 2020 [cit. 2021-04-15]. Dostupné z: `https://webconverger.com/pricing/`.
7. ASPENCORE. *2019 Embedded Markets Study* [online]. 2019 [cit. 2021-04-15]. Dostupné z: `https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf`.
8. W3TECHS. *Usage statistics of operating systems for websites* [online]. 2021 [cit. 2021-04-15]. Dostupné z: `https://w3techs.com/technologies/overview/operating_system`.
9. W3TECHS. *Usage statistics of Unix for websites* [online]. 2021 [cit. 2021-04-15]. Dostupné z: `https://w3techs.com/technologies/details/os-unix`.
10. TORVALDS, L. *torvalds/linux/COPYING* [online]. Github, 2020 [cit. 2021-04-15]. Dostupné z: `https://github.com/torvalds/linux/blob/COPYING`.
11. DEBIAN PROJECT. *Společenská smlouva Debianu* [online]. 2004 [cit. 2021-04-15]. Dostupné z: `https://www.debian.org/social_contract.cs.html`.
12. HERTZOG, R.; ROLAND, M. *The Debian Administrator's Handbook, Debian Buster from Discovery to Mastery*. Dominique Levy Galerie Perroti, 2020. ISBN 979-10-91414-19-7.
13. CANONICAL LTD. *Ubuntu - Releases* [online]. 2021 [cit. 2021-04-15]. Dostupné z: `https://wiki.ubuntu.com/Releases`.
14. ALPINE LINUX DEVELOPMENT TEAM. *Alpine Linux* [online]. 2021 [cit. 2021-04-15]. Dostupné z: `https://www.alpinelinux.org/about/`.
15. ALPINE LINUX DEVELOPMENT TEAM. *Alpine Linux Init System* [online]. 2021 [cit. 2021-04-15]. Dostupné z: `https://wiki.alpinelinux.org/wiki/Alpine_Linux_Init_System`.

16. MOZILLA FOUNDATION. *Mozilla Public Licence Version 2.0* [online]. 2012 [cit. 2021-04-15]. Dostupné z: https://www.mozilla.org/en-US/MPL/2.0/.
17. RODARO, M.; KAPLY, M.; GARDENHIRE, L. *Firefox Enterprise Kiosk mode* [online]. 2020 [cit. 2021-04-15]. Dostupné z: https://support.mozilla.org/en-US/kb/firefox-enterprise-kiosk-mode.
18. MOZILLA FOUNDATION. *Policy Templates for Firefox* [online]. Github, 2021 [cit. 2021-04-15]. Dostupné z: https://github.com/mozilla/policy-templates/blob/README.md.
19. MDN CONTRIBUTORS. *Match patterns in extension manifests* [online]. MDN Web Docs, 2021. Dostupné také z: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Match_patterns.
20. WOOLYSS. *Chromium vs Google Chrome* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://chromium.woolyss.com/.
21. BEVERLOO, P. *List of Chromium Command Line Switches* [online]. 2019 [cit. 2021-04-15]. Dostupné z: https://peter.sh/experiments/chromium-command-line-switches/.
22. GOOGLE. *Configuring Apps and Extensions by Policy* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://www.chromium.org/administrators/configuring-policy-for-extensions.
23. GOOGLE. *Chromium - Linux Quick Start* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://www.chromium.org/administrators/linux-quick-start.
24. GOOGLE. *URL Blocklist filter format* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://www.chromium.org/administrators/url-blocklist-filter-format.
25. ARCHWIKI CONTRIBUTORS. *Disk quota* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Disk_quota&oldid=660855.
26. ARCHWIKI CONTRIBUTORS. *tmpfs* [online]. ArchWiki, 2020 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Tmpfs&oldid=637974.
27. ARCHWIKI CONTRIBUTORS. *GRUB* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=GRUB&oldid=661221.
28. ARCHWIKI CONTRIBUTORS. *getty* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Getty&oldid=659767.
29. ARCHWIKI CONTRIBUTORS. *xinit* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Xinit&oldid=650234.
30. ARCHWIKI CONTRIBUTORS. *Openbox* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Openbox&oldid=662104.
31. DEBIAN CONTRIBUTORS. *Openbox* [online]. Debian wiki, 2020 [cit. 2021-04-15]. Dostupné z: https://wiki.debian.org/Openbox.
32. ARCHWIKI CONTRIBUTORS. *xmodmap* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Xmodmap&oldid=662633.
33. ARCHWIKI CONTRIBUTORS. *LightDM* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=LightDM&oldid=661714.
34. ZINI, E. *lightdm-autologin-greeter* [online]. Github, 2019 [cit. 2021-04-15]. Dostupné z: https://github.com/spanezz/lightdm-autologin-greeter.
35. DEBIAN CONTRIBUTORS. *Keyboard* [online]. Debian wiki, 2019 [cit. 2021-04-15]. Dostupné z: https://wiki.debian.org/Keyboard.
36. TABONE, S. *Virtual Keyboard for Google Chrome™* [online]. Github, 2018 [cit. 2021-04-15]. Dostupné z: https://github.com/xontab/chrome-virtual-keyboard.

37. ONBOARD DEVEL TEAM. *Onboard* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://launchpad.net/onboard.

38. AGRECH, F. *Florence Virtual Keyboard* [online]. 2014 [cit. 2021-04-15]. Dostupné z: http://florence.sourceforge.net/english.html.

39. ARCHWIKI CONTRIBUTORS. *Kernel module* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Kernel_module&oldid=637992.

40. ARCHWIKI CONTRIBUTORS. *Session lock* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Session_lock&oldid=659101.

41. ARCHWIKI CONTRIBUTORS. *systemd-timesyncd* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Systemd-timesyncd&oldid=655607.

42. DEBIAN CONTRIBUTORS. *DebianFirewall* [online]. Debian wiki, 2019 [cit. 2021-04-15]. Dostupné z: https://wiki.debian.org/DebianFirewall.

43. ARCHWIKI CONTRIBUTORS. *nftables* [online]. ArchWiki, 2021 [cit. 2021-04-15]. Dostupné z: https://wiki.archlinux.org/index.php?title=Nftables&oldid=649410.

44. ARCHWIKI CONTRIBUTORS. *Cron* [online]. ArchWiki, 2021 [cit. 2021-04-21]. Dostupné z: https://wiki.archlinux.org/index.php?title=Cron&oldid=660670.

45. PUPPET INC. *Puppet* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://puppet.com/.

46. PUPPET INC. *Puppet* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://forge.puppet.com/modules.

47. REDHAT INC. *Ansible* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://www.ansible.com/.

48. REDHAT INC. *Ansible Galaxy* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://galaxy.ansible.com/home.

49. SALTSTACK. *SaltStack Documentation* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://docs.saltproject.io/en/latest/.

50. SALTSTACK. *SaltStack Formulas* [online]. Github, 2021 [cit. 2021-04-15]. Dostupné z: https://github.com/saltstack-formulas.

51. ANSIBLE COMMUNITY. *ansible-collections/community.general* [online]. Github, 2021 [cit. 2021-04-21]. Dostupné z: https://github.com/ansible-collections/community.general.

52. REDHAT INC. *ansible-collections/ansible.posix* [online]. Github, 2021 [cit. 2021-04-21]. Dostupné z: https://github.com/ansible-collections/ansible.posix.

53. REDHAT INC. *Galaxy User Guide: Installing roles and collections from the same requirements.yml file* [online]. 2021 [cit. 2021-04-15]. Dostupné z: https://docs.ansible.com/ansible/latest/galaxy/user_guide.html#installing-roles-and-collections-from-the-same-requirements-yml-file.

54. FANTHOM. *On Screen Keyboard* [online]. 2018 [cit. 2021-04-15]. Dostupné z: https://forum.porteus.org/viewtopic.php?p=66684#p66684.

Obsah přiloženého média

- `readme.txt` ..... stručný popis obsahu média
- `ansible` ..... zdroje pro použití s Ansible
  - `ansible-pre.sh` ..... skript pro instalaci nutných programů pro Ansible
  - `roles` ..... vzor role kiosku pro Ansible
- `scripts` ..... administrativní skripty bez použití Ansible
  - `configure.sh` ..... skript pro nastavení prohlížeče
  - `install.sh` ..... skript pro instalaci a nastavení kiosku
  - `update.sh` ..... skript pro ruční aktualizaci systému
- `latex` ..... zdrojová forma práce ve formátu LaTeX
- `thesis.pdf` ..... text práce ve formátu PDF