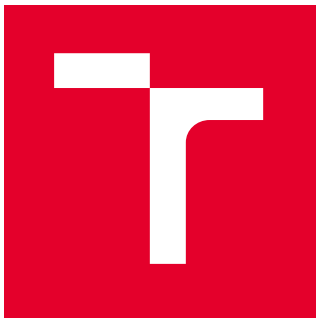


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

WEBOVÉ STRÁNKY JAKO ELEKTRONICKÝ DŮKAZ

WEBPAGE AS DIGITAL EVIDENCE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — Jan Rezek

AUTHOR

VEDOUCÍ PRÁCE — JUDr. Mgr. Jakub Harašta, Ph.D.

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Jan Rezek *ID:* 227374
Ročník: 3 *Akademický rok:* 2022/23

NÁZEV TÉMATU:

Webové stránky jako elektronický důkaz

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je (i) identifikovat požadavky na zajištění obsahu webové stránky v důkazní kvalitě (analýza právních předpisů, rozhodovací praxe) a (ii) vytvořit nástroj s funkcionalitami reflektujícími tyto požadavky. Výstupem semestrálního projektu bude softwarová implementace nástroje pro získání důkazního materiálu v rozpracované formě se základní funkcionalitou. Výstupem bakalářské práce bude kompletní nástroj včetně dokumentace, instalačního a uživatelského manuálu a testovacím protokolem.

DOPORUČENÁ LITERATURA:

podle pokynů vedoucího práce

Termín zadání: 6.2.2023 *Termín odevzdání:* 26.5.2023

Vedoucí práce: JUDr. Mgr. Jakub Harašta, Ph.D.

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

Abstrakt

Tato bakalářská práce se zabývá problematikou zajištění webových stránek jako elektronických důkazů. Zaměřuje se na analýzu právních předpisů a rozhodovací praxe týkající se elektronických důkazů, a na základě toho identifikuje požadavky na jejich uchování a zajištění. Práce také zkoumá roli hashovacích funkcí pro udržení integrity a autenticity těchto důkazů. Jako výsledek je navržen a implementován nástroj pro získání a uchování obsahu webových stránek v důkazní kvalitě, který je v souladu s identifikovanými požadavky.

Klíčová slova

Elektronické důkazy, Zajištění webových stránek, Hashovací funkce, Právní předpisy a rozhodovací praxe, Forenzní analýza webových stránek, Nástroj pro získání důkazního materiálu, Důvěryhodnost elektronických důkazů, Webové technologie

Abstract

This bachelor’s thesis deals with the issue of securing websites as electronic evidence. It focuses on the analysis of legal regulations and decision-making practice concerning electronic evidence, and on this basis identifies the requirements for its preservation and securing. The thesis also examines the role of hashing functions in maintaining the integrity and authenticity of such evidence. As a result, a tool is designed and implemented for the retrieval and preservation of evidence-quality web content that is consistent with the identified requirements.

Keywords

Electronic Evidence, Website Capture, Hash Functions, Legal Regulations and Judicial Practice, Forensic Analysis of Websites, Tool for Gathering Evidential Material, Reliability of Electronic Evidence, Web Technologies

Bibliografická citace

REZEK, Jan. Webové stránky jako elektronický důkaz. Brno, 2023. Dostupné také z: https://www.vut.cz/studenti/zav-prace/detail/151212. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Jakub Harašta.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	*Jan Rezek*
VUT ID studenta:	*227374*
Typ práce:	*Bakalářská práce*
Akademický rok:	*2022/23*
Téma závěrečné práce:	*Webové stránky jako elektronický důkaz*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

.

V Brně dne: ----------

podpis autora

Poděkování

Chtěl bych vyjádřit upřímnou vděčnost panu JUDr. Mgr. Jakubu Haraštovi, Ph.D., za jeho významnou podporu a vedení během tvorby mé bakalářské práce. Jeho konstruktivní rady a odborné zkušenosti hrály nezastupitelnou roli a vedly mě k úspěšnému dokončení této bakalářské práce.

Dále bych chtěl vyjádřit srdečné poděkování mé rodině za jejich neustálou podporu během mého studia a při tvorbě této práce. Zvláště pak bych chtěl poděkovat své mamince za její trpělivost při opakovaném čtení a oprav mé práce.

V Brně dne: ----------

podpis autora

Obsah

SEZNAM OBRÁZKŮ ... **8**

SEZNAM VÝPISŮ ... **9**

ÚVOD ... **10**

1. ELEKTRONICKÝ DŮKAZ ... **11**

1.1 OBECNÉ POŽADAVKY NA ELEKTRONICKÉ DŮKAZY ... 11
1.2 ZAJIŠŤOVÁNÍ ELEKTRONICKÉHO DŮKAZU ... 12
1.2.1 Zajištění zařízení nebo jeho datových nosičů ... *13*
1.2.2 Získání přístupu k počítačovým datům na vzdáleném uložišti ... *14*
1.2.3 Získání dat od poskytovatelů služeb ... *14*
1.3 ZÁVĚR KAPITOLY ELEKTRONICKÉ DŮKAZY ... 15

2. INTEGRITA DAT ... **16**

2.1 HASHOVACÍ FUNKCE ... 17
2.1.1 SHA-2 ... *18*

3. WEBOVÉ STRÁNKY JAKO ELEKTRONICKÝ DŮKAZ ... **19**

3.1 ZPŮSOBY ZAJIŠŤOVÁNÍ WEBOVÝCH STRÁNEK ... 19

4. VÝVOJ NÁSTROJE ... **23**

4.1 POŽADAVKY NA NÁSTROJ ... 23
4.2 SOUHRN POUŽITÝCH TECHNOLOGIÍ A KNIHOVEN ... 24
4.2.1 Backend ... *25*
4.2.2 Frontend ... *26*
4.3 NÁVRH NÁSTROJE ... 26
4.3.1 Návrh frontendu ... *27*
4.3.2 Návrh backendu ... *28*
4.3.3 Návrh struktury elektronického důkazu ... *29*
4.4 IMPLEMENTACE NÁSTROJE ... 31
4.4.1 Důležité funkce nástroje ... *31*
4.4.2 Implementace nástroje na produkční server ... *36*
4.5 TESTOVÁNÍ NÁSTROJE ... 37
4.5.1 Jednotkové testy ... *37*
4.5.2 Testování komunikace mezi serverem a frontendem ... *38*
4.5.3 Testování generování elektronických důkazů ... *38*
4.5.4 Testování zpětného ověření integrity ... *39*
4.5.5 Závěr testování ... *39*

5. ZÁVĚR ... **41**

SEZNAM ZKRATEK ... **44**

SEZNAM PŘÍLOH ... **45**

SEZNAM OBRÁZKŮ

Obr. 3.1: Ukázka zachycení webové stránky vytištěním 20
Obr. 3.2: Ukázka zachycení webové stránky metodou printscreen 20
Obr. 4.1: Ukázka obsahu souboru capture-details.html 30
Obr. 4.2: Ukázka výsledku jednotkového testu modulu updateLinks..... 38

SEZNAM VÝPISŮ

Výpis 4.1: Ukázka kódu z funkce donwnloadHtml 31
Výpis 4.2: Ukázka kódu z funkce parseHtml..... 33
Výpis 4.3: Ukázka kódu pro extrakci obrázků z CSS 34
Výpis 4.4: Ukázka kódu pro výpočet HASH funkce 35
Výpis 4.5: Ukázka kódu jednotkového testu pro modul updateLinks 37

Úvod

Webové stránky se staly nedílnou součástí moderního života a významným zdrojem informací. V současné době hrají klíčovou roli nejen v komerčních a sociálních sítích, ale také ve vědeckém výzkumu, vzdělávání a veřejné správě. Vzhledem k tomu, že internetový obsah je snadno měnitelný a volně dostupný, mohou vznikat právní spory a problémy spojené s autorskými právy, ochranou osobních údajů, plagiátorstvím, kybernetickým zločinem nebo obchodními spory. V tomto kontextu se webové stránky stávají elektronickým důkazem, který je nezbytný pro řešení těchto sporů a zajištění spravedlnosti.

Tato bakalářská práce si klade za cíl identifikovat požadavky na zajištění obsahu webové stránky v důkazní kvalitě a vytvořit nástroj s funkcionalitami reflektujícími tyto požadavky. Práce se zaměřuje na analýzu právních předpisů a rozhodovací praxi, které určují, jakým způsobem lze získat a uchovávat elektronické důkazy z webových stránek, aby byly uznány jako důvěryhodné a platné v soudním řízení.

V první části práce bude provedena rešerše právních předpisů a rozhodovací praxe týkající se elektronických důkazů s důrazem na webové stránky. Tato analýza poskytne přehled o kritériích a postupech, které je třeba dodržet při zajišťování a uchovávání obsahu webových stránek jako důkazního materiálu. Dále bude zkoumána role hashovacích funkcí v kontextu zajišťování integrity a autenticity elektronických důkazů.

Ve druhé části práce bude navržen a implementován nástroj pro získání důkazního materiálu z webových stránek, který bude reflektovat požadavky zjištěné v první části práce. Tento nástroj bude schopen získat obsah webové stránky, zajistit jeho integritu a autenticitu a uchovávat důkazní materiál v souladu s právními předpisy a rozhodovací praxí. Nástroj bude navržen tak, aby byl snadno použitelný, přizpůsobitelný a kompatibilní s různými webovými technologiemi.

Výsledky této práce přispějí k lepšímu pochopení problematiky elektronických důkazů a způsobů zajišťování integrity a autenticity webových stránek. Navržený nástroj může sloužit jako užitečný zdroj pro právníky, forenzní experty, výzkumníky a další zainteresované strany, které se zabývají zajištěním elektronických důkazů v souvislosti s webovými stránkami. Tato práce tak přispěje k posílení důvěryhodnosti elektronických důkazů a zajištění spravedlnosti v digitálním prostředí.

1. ELEKTRONICKÝ DŮKAZ

Pojem *elektronický důkaz* není v trestním řádu České republiky přímo vymezen. Trestní řád sice obsahuje některé postupy a instituty, avšak tyto často neodpovídají technickým realitám potřebným pro zajištění takového důkazu. Obecně lze elektronický důkaz chápat jako jakákoliv data v elektronické podobě, která mohou posloužit jako důkaz [1]. Tyto důkazy se mohou skládat z fotografií, videí, textových souborů, e-mailů, sociálních médií, webových stránek a mnoha dalších.

Elektronické důkazy nabývají na významu v procesu soudního řízení, protože díky stále dostupnější technologii se jejich využití stává častějším. Stále více a více dat se nachází v elektronickém prostoru a jejich použití pro dokazování v soudním řízení je v určitých případech nezbytné. Avšak jsou na ně kladeny určité požadavky, které umožňují použitelnost důkazu v soudním řízení. Tyto požadavky budou zkoumané v následující podkapitole.

1.1 Obecné požadavky na elektronické důkazy

Pro přijetí elektronického důkazu, jako je například webová stránka, soudem je nezbytné splnění několika kritérií. V první řadě je důležitá relevance elektronického důkazu, která znamená, že musí mít přímý vztah k otázkám, které jsou předmětem sporu. Bez relevance by důkaz nebyl pro rozhodnutí případu užitečný.

Dále je klíčovou vlastností elektronického důkazu jeho autenticita. Autentický důkaz musí pocházet z důvěryhodného zdroje a nesmí být falšován nebo zmanipulován. Soud musí mít jistotu, že důkaz je skutečně tím, čím se tvrdí, že je. To zahrnuje ověření pravosti zdroje, z něhož důkaz pochází, a důkaz, že nebyl poškozen nebo pozměněn.

Neporušenost neboli integrita elektronického důkazu je dalším zásadním kritériem, které musí být splněno. Neporušený důkaz musí být chráněn proti neoprávněnému přístupu, úpravám nebo zničení, aby bylo zajištěno, že zachovává svou původní podobu a hodnověrnost. To může zahrnovat použití různých technologií a postupů pro zabezpečení důkazu a jeho uchování.

Čitelnost a srozumitelnost jsou také důležitými aspekty elektronických důkazů. Musí být prezentovány takovým způsobem, aby byly snadno pochopitelné pro soud i ostatní účastníky řízení. To zahrnuje jasnou a stručnou prezentaci důkazů a vysvětlení technických aspektů, které mohou být pro laického čtenáře obtížně srozumitelné.

Posledním kritériem je přiměřenost a zákonnost elektronického důkazu. Přiměřenost zahrnuje získání a použití důkazu v souladu s právními předpisy a etickými normami. Důkaz nesmí být opatřený nezákonně nebo protizákonně. Je nezbytné respektovat práva účastníků řízení, ochranu soukromí a osobních údajů. Bez splnění tohoto kritéria by mohl být důkaz považován za neplatný a nepřijatelný pro použití v soudním řízení [1].

Při hodnocení elektronických důkazů je třeba zohlednit výše uvedená kritéria, a to nejen samostatně, ale také v jejich souhrnném kontextu. Každý důkaz by měl být posuzován individuálně a vzhledem k okolnostem konkrétního případu.

V praxi musí právníci, soudci a ostatní účastníci řízení pečlivě zvážit splnění všech uvedených kritérií, aby zajistili, že elektronické důkazy budou řádně přijaty a použity v soudních řízeních. Tím se zvyšuje pravděpodobnost spravedlivého a efektivního rozhodnutí případu, který zohledňuje všechny relevantní informace a důkazy.

Vzhledem k rostoucímu významu elektronických důkazů v právním procesu je důležité, aby právníci a soudci byli obeznámeni s jejich specifiky a s kritérii pro jejich přijetí. To může zahrnovat další vzdělávání a školení v oblasti elektronických důkazů, stejně jako spolupráci s odborníky na technologie a kybernetickou bezpečnost, kteří mohou poskytnout důležité informace a podporu při získávání a hodnocení těchto důkazů.

1.2 Zajišťování elektronického důkazu

Zajišťování elektronického důkazu je klíčovým procesem, který musí být proveden v souladu se zákonem, aby nedošlo k významným chybám, jež by mohly způsobit absolutní nebo relativní nepřípustnost důkazu. Podobně jako u ostatních důkazů jsou i elektronické důkazy nepřípustné v případě, že byly získány nezákonným donucením podle § 89 odst. 3 TŘ. Pokud je důkaz označen jako nepřípustný, nelze jej použít pro dokazování u soudu.

V trestním řádu není pro zajištění elektronického důkazu uveden specifický procesní postup, což vede k tomu, že se pro zajištění důkazů často používají procesní nástroje, které původně nebyly pro tento účel zamýšleny [1]. Tato situace může vést ke komplikacím, nejasnostem a riziku nepřípustnosti důkazu v soudním řízení.

Dle autorů publikace *Elektronické důkazy v trestním řízení* lze elektronická data získat zásadně třemi způsoby [1]:

1) Zajištěním samotného zařízení nebo jeho datových nosičů
2) Získáním přístupu k počítačovým datům uchovaným na vzdáleném uložišti
3) Získáním důkazního materiálu od poskytovatelů služeb, u kterých se potřebná data nachází

Je důležité zdůraznit, že zajištění elektronických důkazů vyžaduje značnou odbornost a znalosti právních, technických a etických aspektů. Právníci a vyšetřovatelé musí být obeznámeni s nejnovějšími technologiemi, postupy a bezpečnostními opatřeními, aby byli schopni účinně získávat a zachovávat elektronické důkazy v souladu se zákony a etickými normami.

Navíc je nezbytné, aby byly procesy zajištění elektronických důkazů transparentní. To zahrnuje udržování řádné dokumentace o všech krocích zajištění, získání a uchování důkazů, jakož i řádné vyřizování povolení a souhlasů od příslušných stran, pokud je to vyžadováno zákonem. Tím se zajišťuje, že získané důkazy budou respektovat práva všech zúčastněných stran a budou přijatelné pro použití v soudním řízení.

Vzhledem k rychlému rozvoji technologií a narůstající závislosti na elektronických datech v různých aspektech života je nezbytné, aby se právní předpisy a procesní postupy neustále aktualizovaly a přizpůsobovaly novým výzvám a možnostem, které přináší digitální svět. To může zahrnovat revizi a doplnění stávajících zákonů a postupů, aby lépe reflektovaly specifika a nároky spojené se zajištěním a použitím elektronických důkazů v trestním řízení.

1.2.1 Zajištění zařízení nebo jeho datových nosičů

Zajištění zařízení nebo datových nosičů je v praxi velmi efektivní způsob pro zajišťování elektronických důkazů. Tento postup je upraven zejména v § 78 a § 79 Trestního řádu (TŘ), které stanovují povinnost vydat věci důležité pro trestní řízení orgánům činným v trestním řízení na vyzvání státního zástupce nebo policejního orgánu. V kontextu elektronických důkazů jsou zařízení nebo datové nosiče obsahující potřebná data považovány za takové věci, na které se tyto ustanovení mohou vztahovat.

Při zajišťování zařízení nebo datových nosičů je klíčové dodržet správný postup a zaznamenat všechny kroky v důkladném protokolu. Tento protokol by měl obsahovat podrobnosti o průběhu zajištění, účastníků, přítomných svědcích a případných zvláštnostech, které nastaly během procesu. Zajištění musí být provedeno tak, aby se minimalizovalo riziko porušení integrity a autenticity dat uložených na zařízení nebo nosiči. Osoba, která zařízení nebo nosič vydává nebo jí je odnímáno, má právo na potvrzení o vydání/odnětí věci nebo přímo opis protokolu z procesu zajištění.

Je důležité chránit zajištěná zařízení a datové nosiče proti statické elektřině, manipulaci nebo poškození. K tomu je vhodné použít antistatický vak a následně uložit zařízení do zapečetěného boxu nebo jiného obalu, který zajišťuje ochranu.

Zajištění zařízení nebo datového nosiče může proběhnout také v rámci domovní prohlídky nebo prohlídky jiných prostor nařízené předsedou senátu nebo soudcem na návrh státního zástupce. V takovém případě je třeba také vést podrobný protokol o celém procesu prohlídky. Protokol slouží jako důkazní materiál pro soudní řízení a zároveň poskytuje záruku dodržení procesních norem.

Během zajišťování zařízení nebo datových nosičů je důležité respektovat základní práva a svobody účastníků řízení, včetně práva na soukromí a ochranu osobních údajů. To zahrnuje například zohlednění účelu a proporcionality zásahu při zajišťování zařízení a datových nosičů a ujištění, že získané důkazy budou použity pouze v souladu s právními předpisy a etickými normami.

1.2.2 Získání přístupu k počítačovým datům na vzdáleném uložišti

Získávání přístupu k počítačovým datům uloženým na vzdálených uložištích je důležitou součástí trestního vyšetřování, zvláště v případech, kdy se jedná o elektronické důkazy. Ačkoli existují různé způsoby, jak získat přístup k těmto datům, je zásadní, aby byl zajištěn soulad se zákonnými předpisy a dodržovány procesní normy, které omezují zásahy do soukromí a zaručují ochranu práv účastníků řízení.

Volně dostupná data, například obsah veřejně přístupných webových stránek, lze získat bez nutnosti předchozího povolení soudce. Avšak i při zajišťování těchto dat je třeba dodržovat předpisy stanovené v § 112 TŘ a vytvářet podrobný protokol, který dokumentuje způsob a čas získání informací. Kvalitní dokumentace zdrojového kódu webové stránky je nezbytná pro správné zachování důkazní hodnoty získaných dat.

Pokud se jedná o data, ke kterým je nutné mít přístupové údaje, ať už se jedná o hesla nebo jiné identifikační informace, je nutné postupovat v souladu s ustanoveními o sledování osob a věcí podle § 158d TŘ. Přístup k takovým datům může být povolen pouze soudcem nebo na základě výslovného souhlasu osoby, do jejíž soukromí je zasahováno. V naléhavých případech lze zahájit sledování bez povolení, ale policejní orgán je povinen bezodkladně požádat o dodatečné povolení. Pokud povolení není uděleno do 48 hodin, musí být získaná data zničena.

Přístup k vzdáleným zabezpečeným datovým zdrojům může být zajištěn také prostřednictvím zařízení, které obsahuje údaje nutné k přístupu k těmto službám. V takovém případě je opět aplikováno ustanovení § 158d TŘ o sledování osob a věcí. Je důležité rozlišovat mezi různými typy služeb, ke kterým je přistupováno. Pokud se jedná o elektronickou komunikaci v reálném čase, je třeba postupovat v souladu s § 88 TŘ, který se týká odposlechu a záznamu telekomunikačního provozu. V případě, že služba má charakter kombinovaného uložiště a komunikačního prostředku, je nutné zajistit uložená data podle § 158d TŘ a provádět odposlech v souladu s § 88 TŘ. [1]

1.2.3 Získání dat od poskytovatelů služeb

Jednou z možností, jak získat důležitá data potřebná pro elektronické důkazy v trestním řízení, je získání těchto dat přímo od poskytovatele služby, na kterém se daná data nachází. Způsob získání dat od poskytovatele závisí na povaze poskytovatele a povaze dat. Poskytovatel může mít charakter poskytovatele telekomunikační služby nebo poskytovatele služby informační společnosti. Data mohou obsahovat informace vytvořené uživatelem během používání služby nebo informace o komunikaci uživatele, lokalizační údaje a metadata.

Získání dat, která nejsou chráněna telekomunikačním tajemstvím nebo jinou povinností mlčenlivosti, je možné pomocí dožádání dle § 8 odst. 1 TŘ. Mezi tato data mohou patřit různé logy, metadata nebo obsahová data zveřejněná uživatelem. V případě potřeby zajištění dat chráněných bezpečnostním opatřením je nutné postupovat podle již zmíněného § 158d odst. 3 TŘ.

Poskytovatelé telekomunikačních služeb mají povinnost uchovávat provozní údaje podle § 90 ZEK a lokalizační údaje podle § 91 ZEK po dobu a za podmínek stanovených ZEK. Přístup k těmto údajům je možný za podmínek upravených v § 88a TŘ v případě nařízení soudce na žádost státního zástupce, ale pouze při stíhání trestných činů vyjmenovaných v § 88 TŘ [1].

Data, která jsou obsahem telekomunikačního provozu a na která se váže telekomunikační tajemství, je možné odposlouchávat pouze za podmínek upravených v § 88 TŘ. Tento nástroj lze opět využít pouze v případě stíhání vyjmenovaných trestných činů.

V rámci trestního řízení mohou orgány činné v trestním řízení také uplatňovat dožádání dle § 8 odst. 1 o uchování zálohy dat u poskytovatele služeb. Tento postup je vhodný v případě, že ještě nebylo vydáno povolení pro přístup k datům, ale existuje obava, že by se pro trestní stíhání důležitá data mohla ztratit. Policejní orgán si takto může zajistit zálohu dat až do doby, než bude vydáno příslušné povolení.

Je třeba zdůraznit, že spolupráce s poskytovateli služeb je klíčová pro úspěšné zajištění důkazů v trestním řízení. Při žádání o získání dat od poskytovatele služeb je nutné respektovat jejich interní procesy a dodržovat zákonné požadavky, aby nedošlo k narušení důkazní hodnoty získaných dat.

V některých případech může být také nezbytné spolupracovat s poskytovateli služeb na mezinárodní úrovni, což může zahrnovat právní spolupráci mezi státy. Pro zajištění dat od zahraničních poskytovatelů služeb je často nutné využít mezinárodních právních nástrojů, jako jsou vzájemná právní pomoc nebo evropský zatýkací rozkaz [1].

1.3 Závěr kapitoly elektronické důkazy

Tato kapitola byla zaměřena na různé způsoby získávání elektronických důkazů v trestním řízení, včetně zajišťování dat ze vzdálených uložišť a získávání dat od poskytovatelů služeb. Nyní, když byly prozkoumány metody pro získání těchto důkazů, je důležité se zaměřit na integritu dat – jednu z klíčových složek elektronických důkazů.

Integrita dat je zásadní pro úspěch trestního řízení, protože bez ní by mohla být důkazní hodnota získaných dat zpochybněna. Následující kapitola bude podrobněji představovat principy a postupy, které zajistí zachování integrity dat a jejich důkazní hodnoty v průběhu celého procesu zajišťování, uchovávání a prezentace elektronických důkazů.

2. INTEGRITA DAT

Elektronické důkazy hrají stále větší roli v soudních řízeních a trestním právu, což zdůrazňuje důležitost jejich integrity během celého procesu shromažďování, přepravy a analýzy. Integrita dat zajišťuje, že elektronické důkazy nebyly poškozeny, pozměněny nebo zmanipulovány, což je klíčové pro jejich přípustnost před soudem a důvěryhodnost v rámci právního procesu.

Pro udržení integrity elektronických důkazů je nezbytné dodržovat několik základních principů a postupů. Tyto principy a postupy zahrnují řádné protokolování procesů zajištění, používání nástrojů pro ověřování integrity, jako jsou hashovací funkce, a vytváření záložních kopií zajištěných dat [1].

Hashovací funkce, které si blíže představíme v následující podkapitole, představují zásadní nástroj pro zajištění integrity elektronických důkazů. Jedná se o jednosměrné algoritmy, které slouží k vytvoření unikátního otisku (hash) dat, který je později možné porovnat s otiskem původních dat za účelem ověření jejich integrity. Vytvoření hash hodnoty je nezbytné při zajištění elektronických důkazů, aby bylo možné později ověřit jejich nezměněnost.

Kromě hashovacích funkcí je pro zachování integrity elektronických důkazů klíčové také jejich bezpečné uložení. Elektronické důkazy by měly být uschovány na místech chráněných před neoprávněným přístupem, aby se minimalizovalo riziko manipulace či ztráty dat. Bezpečnost dat lze zvýšit pomocí šifrování, které znesnadní neoprávněný přístup k uloženým důkazům.

Důležitým krokem pro zajištění integrity dat je rovněž vytváření záložních kopií elektronických důkazů. Zálohy by měly být uloženy na oddělených místech, aby se snížilo riziko jejich ztráty či poškození. Tímto způsobem je zajištěno, že důkazy budou vždy k dispozici pro případ potřeby během soudního řízení.

Ve výsledku je integrita dat zásadní pro úspěšné použití elektronických důkazů v soudních řízeních a pro zajištění spravedlivého a transparentního právního procesu. Kombinace vhodných nástrojů, jako jsou hashovací funkce, bezpečné uložení dat a zálohování, spolu s dodržováním řádných postupů a protokolů, umožňuje zajistit integritu elektronických důkazů a jejich přípustnost u soudu. V následující podkapitole se podrobněji zaměřím na udržování integrity elektronických důkazů pomocí hashovacích funkcí.

2.1 Hashovací funkce

Hashovací funkce hrají klíčovou roli v mnoha oblastech informatiky, včetně kryptografie, ukládání dat a zabezpečení. Tyto algoritmy přijímají vstup libovolné délky a generují výstup pevné délky, známý jako hash nebo digest zprávy. Díky jejich schopnosti vytvořit jedinečný "otisk" pro každý vstup se hashovací funkce používají k zajištění integrity a autenticity dat. V této kapitole se zaměříme na některé běžné hashovací funkce, jako jsou MD5, SHA-1, SHA-2 a SHA-3, a jejich vhodnost pro použití v kontextu elektronických důkazů [2].

MD5 (Message Digest Algorithm 5) je hashovací funkce, která byla široce používána v devadesátých letech 20. století. MD5 generuje 128bitový hash, ale během let byly nalezeny zranitelnosti, které snižují jeho bezpečnost a důvěryhodnost. Tyto zranitelnosti zahrnují možnost kolizí, kdy dva odlišné vstupy generují stejný hash, což zpochybňuje integritu výstupu. Z těchto důvodů již MD5 není považováno za bezpečné řešení pro zajištění integrity dat [3].

SHA-1 (Secure Hash Algorithm 1) je další hashovací funkce, která byla původně navržena jako náhrada MD5. SHA-1 generuje 160bitový hash a byla považována za bezpečnější než MD5 [4]. Avšak v průběhu času byly nalezeny i u SHA-1 zranitelnosti, které umožňují kolize a zpochybňují integritu výstupních hashů. Proto se od SHA-1 začalo upouštět ve prospěch bezpečnějších alternativ [5].

SHA-2 (Secure Hash Algorithm 2) je rodina hashovacích funkcí, která poskytuje lepší zabezpečení než MD5 a SHA-1, díky svým silnějším kryptografickým vlastnostem a odolnosti vůči kolizím. V kontextu elektronických důkazů je SHA-256 často považována za ideální volbu, protože nabízí vyvážený kompromis mezi výpočetní efektivitou a zabezpečením. SHA-256 generuje 256bitový hash, což poskytuje dostatečnou úroveň zabezpečení pro většinu aplikací, včetně právních a forenzních účelů [6].

SHA-3 (Secure Hash Algorithm 3) je další generace hashovacích funkcí, která je navržena tak, aby poskytovala ještě silnější kryptografické vlastnosti než SHA-2. SHA-3 byla vyvinuta jako důsledek hledání alternativy, která by byla odolná vůči potenciálním slabostem v předchozích algoritmech, jako jsou MD5, SHA-1 a SHA-2 [7]. I když SHA-3 poskytuje vysokou úroveň zabezpečení, v kontextu elektronických důkazů je často preferována varianta SHA-256 ze SHA-2 rodiny.

Důvodem, proč je v tomto případě lepší použít SHA-2 než SHA-3, je, že SHA-256 je již zavedeným standardem s dostatečnou úrovní zabezpečení pro většinu aplikací, včetně právních a forenzních účelů. Navíc, výpočetní efektivita SHA-256 je větší než u SHA-3, což znamená, že zpracování elektronických důkazů pomocí SHA-256 je rychlejší a efektivnější. SHA-3 je stále vhodnou volbou pro specifické aplikace, které vyžadují vyšší úroveň zabezpečení, ale pro běžné účely v kontextu elektronických důkazů je SHA-256 často považována za dostačující.

Pro zajištění integrity elektronických důkazů je důležité používat vhodné a bezpečné hashovací funkce, jako je SHA-256. Tímto způsobem lze zajistit důvěryhodnost elektronických důkazů a udržet jejich integritu v průběhu celého procesu shromažďování, přepravy a analýzy. Následující kapitola bude podrobněji zaměřena na SHA-2, zejména na variantu SHA-256, a bude uvedeno, jak tato hashovací funkce může být použita k zajištění integrity elektronických důkazů v kontextu této bakalářské práce.

2.1.1 SHA-2

SHA-2 je rodina hashovacích funkcí, která byla vyvinuta Národním institutem pro standardy a technologie (NIST) jako nástupce SHA-1. SHA-2 zahrnuje šest variant hashovacích funkcí, které se liší velikostí výstupního hashe: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 a SHA-512/256. V této kapitole se zaměřím na variantu SHA-256, která je v této bakalářské práci použita pro zajištění integrity elektronických důkazů.

SHA-256 je iterativní hashovací funkce, která zpracovává vstupní data v blokách o velikosti 512 bitů. Algoritmus zahrnuje několik matematických operací, které se opakují v každé iteraci a transformují vstupní blok na 256bitový hash. Některé z těchto operací zahrnují bitové posuny, XOR operace a modulární sčítání. Tyto operace se provádějí tak, aby byly splněny požadavky na hashovací funkce, jako jsou determinismus, neinvertibilita a silné lavinové vlastnosti [6].

SHA-256 má několik výhod oproti starším hashovacím funkcím, jako jsou MD5 a SHA-1. Jednou z hlavních výhod je zvýšená bezpečnost. SHA-256 poskytuje větší odolnost proti kolizím, což znamená, že je obtížnější najít dva různé vstupy, které by měly stejný hash. Tato vlastnost je důležitá pro aplikace, kde je integrita dat klíčová, jako je právní a forenzní sféra.

Další výhodou SHA-256 je jeho odolnost proti tzv. preimage útokům. Tyto útoky spočívají v hledání vstupu, který má stejný hash jako nějaký daný hash. Díky svým kryptografickým vlastnostem je pro SHA-256 obtížné najít takový vstup, což zvyšuje důvěryhodnost elektronických důkazů [8].

SHA-256 je tedy vhodná hashovací funkce pro zajištění integrity elektronických důkazů. Při shromažďování důkazů lze vytvořit hash každého souboru nebo datového bloku pomocí SHA-256. Tyto hash hodnoty lze pak uložit a použít jako referenční body pro ověření integrity důkazů v průběhu celého procesu shromažďování, přepravy a analýzy. Kdykoliv v průběhu tohoto procesu je možné porovnat aktuální hash hodnotu s původní uloženou hash hodnotou, což umožňuje zjistit, zda byla data změněna či poškozena.

3. WEBOVÉ STRÁNKY JAKO ELEKTRONICKÝ DŮKAZ

Webové stránky se staly nepostradatelnou součástí naší každodenní reality, a to nejen v osobních, ale i v profesních sférách. Jsou využívány ke sdílení informací, zprostředkování obchodu, komunikaci a mnoha dalším účelům. Díky svému rozsahu, dostupnosti a různorodosti obsahu jsou webové stránky často využívány jako důkaz v trestním řízení a občanskoprávních sporech. Z tohoto důvodu je důležité pochopit, jak webové stránky mohou být použity jako elektronické důkazy, a především, jak mají být zajištěny jako důkazní materiál pro soudní řízení.

Jedním z nejdůležitějších aspektů použití webových stránek jako elektronických důkazů je jejich správné zajištění a dokumentace. Webové stránky mají specifickou vlastnost, a to takovou, že většina moderních webových stránek je dynamická a interaktivní, což znamená, že se jejich obsah může měnit v závislosti na uživatelských interakcích nebo okolnostech. To představuje výzvu při zajišťování webových stránek jako důkazního materiálu, protože je třeba zajistit, že všechny relevantní informace a interakce jsou správně zaznamenány a zachovány.

Webová stránka by měla být zaznamenána s co největším množstvím podrobností, které umožní soudnímu orgánu posoudit její autenticitu a relevanci pro konkrétní případ. Tyto podrobnosti by měly zahrnovat údaje o tom, kdy byla webová stránka zaznamenána, kdo ji zaznamenal a jak byla zaznamenána.

Kromě toho je nezbytné zajistit, aby zajištěné webové stránky byly pečlivě chráněny před úpravami, ztrátou nebo zneužitím. To zahrnuje vytvoření kopií webové stránky a jejího zdrojového kódu. Současně je důležité zaznamenat hash hodnotu webové stránky, aby bylo možné později ověřit její integritu a prokázat, že nebyla manipulována.

3.1 Způsoby zajišťování webových stránek

V současné době existuje několik metod zajišťování webových stránek, které se v praxi používají v různé míře. Mezi nejčastější neprofesionální způsoby patří vytvoření výtisku stránky a použití tzv. printscreenů neboli vyfocení obrazovky počítače. Tyto metody mohou vést k neúplným nebo zkresleným informacím, neboť na výtisku často chybí skryté prvky stránky a většina ostatních prvků, které jsou normálně viditelné. Printscreeny zase často nezachytí odkazy tlačítek nebo další interaktivní prvky stránky.

Pro představu o rozdílném vzhledu těchto dvou způsobů lze uvést dva obrázky reprezentující totožnou část webové stránky, kdy každý vypadá zcela odlišně. Na obr. 3.1 je vyobrazeno vytištění stránky a na obr. 3.2 je vyobrazeno použití printscreenu obrazovky. Ukázka rozdílu mezi způsoby zajištění byla provedena na webové stránce https://printfarm.cz [9].

Obr. 3.1: Ukázka zachycení webové stránky vytištěním

Obr. 3.2: Ukázka zachycení webové stránky metodou printscreen

Jak je možné si všimnout, tak rozdíl těchto dvou způsobů je docela dramatický a dokazování v trestním řízení pomocí metody vytištění by v tomto případě bylo opravdu nevhodné. Každá stránka samozřejmě může vypadat jinak. V tomto případě je to způsobeno i celkovým rozložením této konkrétní stránky a u jiných stránek může být výsledek o něco lepší. Ale pro ukázku nevhodnosti zajištění webové stránky tímto způsobem je to ideální příklad.

I když se na první pohled může zdát, že metoda printscreenu je ideální pro zajišťování stránek, nesmíme opomenout její potenciální nedostatky. U printscreenu například není možné zachytit odkazy skryté v tlačítkách, které uživatel na stránce přímo nevidí. Kromě toho by bylo nutné pořizovat printscreeny celé stránky postupně, neboť jeden snímek obvykle nezachytí vše. Dále je třeba uvést, že tato metoda nedisponuje vlastním mechanismem pro ověřování integrity, což vyžaduje nasazení dalších nástrojů.

Správný způsob zajištění webových stránek spočívá ve stáhnutí celého zdrojového kódu a uložení stránky ve formátu HTML, který lze následně otevřít v běžných webových prohlížečích, jako jsou Google Chrome, Mozilla Firefox nebo prohlížeč Edge. Tento způsob umožňuje prozkoumat veškerý zobrazovaný obsah, včetně částečně skrytých prvků a odkazů, které pro uživatele nejsou přímo viditelné. Nicméně, stahování webových stránek má své omezení, jako například v případě webových aplikací, které vyžadují přihlašovací údaje pro přístup k určitým informacím. Avšak v případech, kdy stáhnutí webové stránky proběhne v pořádku, bude tento způsob zajištění nejvhodnější.

Je důležité zmínit, že uložení webové stránky ve formátu HTML nemá samo o sobě dostatečnou důkazní hodnotu, neboť neexistuje mechanismus zajišťující integritu takto uložených informací. V reakci na tuto problematiku bude vytvořen nástroj, který integritu a důkazní hodnotu uložených stránek zajišťuje. Tento nástroj poskytne řešení, které umožňuje dokazování v trestním řízení s větší mírou přesnosti a spolehlivosti.

Kromě zmíněných metod existuje ještě další způsob zajištění webových stránek, a to pomocí notářského zápisu. Notář na počítači zobrazí webovou stránku a o jejím obsahu provede notářský zápis. Tento zápis má ze zákona presumpci správnosti, protože se jedná o veřejnou listinu. Tento způsob zajištění zaručuje vysokou důkazní hodnotu a spolehlivost uložených informací.

Adekvátnost těchto způsobů zajištění webových stránek byla hodnocena Nejvyšším správním soudem v rozsudku ze dne 24. 8. 2016 sp. zn. 1 As 80/2016-30 ve věci: Společnost s ručením omezeným CARWEST proti České obchodní inspekci o uložení pokuty. Soud řešil způsob zachycení webových stránek pomocí vytištění stránky pro dokazování v soudním řízení.

Právní věta zmíněného rozhodnutí uvádí, že pokud je dokazovanou skutečností neuvedení určité obsahové informace na internetové stránce, případně grafická nebo vizuální podoba stránky, tak listina vzniklá jejím vytištěním, které mohlo určité části stránky potenciálně obsahující předmětné informace nezachytit, není sama o sobě ke zjištění věrného obsahu stránky dostatečná. Soud tak potvrdil, že se při způsobu

zachycení webové stránky pomocí vytištění mohou kromě grafických a jiných vizuálních prvků vytratit i textové prvky. Toto tvrzení doprovodil ukázkou na svých stránkách www.nssoud.cz, kde bylo zřejmé, že se určitá část textu při způsobu zajištění vytištěním vytratila [10].

4. VÝVOJ NÁSTROJE

Praktická část této bakalářské práce se zaměří na návrh, implementaci a testování nástroje, který bude hlavním výstupem této práce. Nástroj bude navržen tak, aby reflektoval požadavky na elektronické důkazy popsané v teoretických částech práce, a jeho hlavním účelem bude zajistit spolehlivé a efektivní získání webových stránek jako elektronických důkazů.

Nástroj bude umožňovat zajištění webových stránek a současně poskytovat možnost ověření integrity zajištěných elektronických důkazů kdykoliv během jejich existence. V rámci praktické části budou především probrány následující aspekty:

- **Použité technologie a knihovny**: Budou představeny technologie a knihovny, které budou využity pro vývoj nástroje, a bude zdůvodněn jejich výběr z hlediska jejich výhod, nevýhod a kompatibility s požadavky na elektronické důkazy.
- **Architektura nástroje**: Bude navržena vhodná architektura nástroje, která bude flexibilní a škálovatelná, aby mohla být snadno adaptována na různé použití a budoucí vývoj.
- **Proces zajištění webové stránky a ověření integrity důkazu**: Budou popsány kroky, které nástroj podnikne při zajištění webové stránky jako elektronického důkazu, a také proces ověření integrity zajištěného důkazu.
- **Klíčové funkce nástroje**: Budou představeny zásadní funkce nástroje, které umožní zajištění webových stránek jako elektronických důkazů, a také následné ověření integrity těchto důkazů.
- **Testování nástroje**: Bude provedeno komplexní testování nástroje, aby bylo zajištěno, že splňuje všechny stanovené požadavky na elektronické důkazy a funguje správně a efektivně v různých scénářích.

4.1 Požadavky na nástroj

Vzhledem k významu elektronických důkazů a potřebě splnit uvedená kritéria, je nezbytné, aby nástroj pro zachycení webových stránek jako elektronického důkazu splňoval následující požadavky:

1) **Autenticita a integrita**: Autenticita a integrita elektronických důkazů jsou základními kameny jejich právní platnosti. Nástroj, který bude použit pro zajištění webových stránek, musí proto zabezpečit, že uložený obsah stránky je autentický a jeho integrita zůstává nedotčena. Toto může zahrnovat mechanismy pro ověření původu stránky, zaznamenání časových údajů a procesů garantujících, že obsah stránky nebyl poškozen nebo upraven. Použití kryptografických technik, jako je digitální podepisování a hashování, může sloužit jako důkaz autenticity a integrity zachycené webové stránky.

2) **Čitelnost a srozumitelnost**: Srozumitelnost elektronických důkazů je klíčová pro jejich efektivní využití v právním procesu. Nástroj pro zajištění webových stránek by měl proto uchovávat stránky tak, aby byly čitelné a srozumitelné pro všechny účastníky soudního procesu, včetně soudců, právníků a jiných zainteresovaných stran. To může zahrnovat zachování všech vizuálních a interaktivních prvků stránky, stejně jako její struktury a odkazů.
3) **Kompletnost**: Pro účely právního procesu je důležité, aby nástroj byl schopen zachytit co nejvíce informací na webové stránce. To zahrnuje jak viditelné prvky, tak také skryté elementy a meta informace, které mohou být relevantní pro soudní případ. Kompletní zachycení webové stránky zajišťuje, že žádný potenciálně důležitý důkaz nezůstane přehlédnutý.
4) **Nezávislost na online zdrojích**: Webové stránky často obsahují prvky, které jsou načítány až po jejich otevření v prohlížeči, a tyto prvky se někdy mohou měnit nebo být odstraněny. Nástroj pro zachycení webových stránek by měl proto být schopen stáhnout a uložit tyto prvky spolu se zdrojovým kódem stránky, aby bylo zajištěno, že žádný potenciálně důležitý důkaz nebude ztracen.
5) **Přiměřenost**: Při zajišťování webových stránek jako elektronických důkazů je nutné dodržovat právní předpisy a etické normy. To zahrnuje respektování práv jednotlivců, ochranu soukromí a osobních údajů, a zákaz získávání důkazů nelegálními způsoby.
6) **Jednoduchost použití**: Nástroj pro zajištění webových stránek by měl být snadno použitelný pro všechny účastníky soudního procesu. To znamená, že by měl mít intuitivní uživatelské rozhraní a jasné pokyny pro použití, aby mohl být efektivně využíván i lidmi bez technických znalostí.
7) **Univerzálnost**: Vzhledem k různorodosti webových stránek a prohlížečů je důležité, aby nástroj pro zajištění webových stránek byl kompatibilní s co největším množstvím různých formátů a technologií. To zajišťuje, že nástroj bude schopen zachytit jakékoli potenciální elektronické důkazy, bez ohledu na to, jak jsou prezentovány na webu.

4.2 Souhrn použitých technologií a knihoven

Pro realizaci nástroje vytvořeného v rámci této práce byly použity různé technologie a knihovny, které jsou popsány v této kapitole. Výběr těchto technologií a knihoven byl založen na jejich schopnostech, popularitě a kompatibilitě s požadavky projektu. Společně umožnily úspěšnou realizaci nástroje pro zajištění webových stránek jako elektronického důkazu a ověřování jejich integrity.

4.2.1 Backend

- **Node.js** – Node.js je open-source běhové prostředí založené na JavaScriptu, které umožňuje vytvářet a provozovat serverovou část webových aplikací. Díky své univerzálnosti a možnosti napojení na řadu knihoven, byl Node.js ideální volbou pro implementaci serverové části nástroje. Dále, díky event-driven architektuře, je Node.js efektivní při zpracování I/O operací, což je zásadní při práci se soubory a komunikaci po síti [11].
- **Express.js** – Express.js je minimalistický webový framework pro Node.js, který poskytuje nástroje a funkce pro usnadnění vývoje webových aplikací. Jelikož je navržen tak, aby byl snadno použitelný a flexibilní, umožnil rychle a efektivně vytvořit a spravovat HTTP server [12].
- **Socket.io** – Socket.io je knihovna pro Node.js, která umožňuje realtime komunikaci mezi klientem a serverem prostřednictvím webových soketů. Použití této knihovny zajišťuje rychlou a spolehlivou komunikaci mezi serverovou a klientskou částí nástroje, což je klíčové pro správné fungování aplikace [13].
- **fs** (filesystem) - je vestavěná knihovna v Node.js, která poskytuje funkce pro práci se soubory a adresáři. V projektu byla použita pro čtení a zápis souborů, například při ukládání elektronických důkazů. Její integrace do Node.js a přímočaré použití přispělo k efektivitě práce se soubory.
- **path** – je vestavěná knihovna v Node.js, která umožňuje manipulaci s cestami k souborům a adresářům. Byla klíčová pro správné vytváření a získávání cest k souborům, což je důležité pro zachování konzistence a přehlednosti v souborové struktuře projektu.
- **crypto** – je vestavěná knihovna v Node.js, která poskytuje kryptografické funkce. V této práci byla použita pro vytváření kryptografických hash hodnot z HTML a jiných souborů, které budou součástí elektronického důkazu, což je klíčový prvek pro ověření integrity zajištěných stránek.
- **uuid** – je knihovna pro generování univerzálních unikátních identifikátorů (UUID). Použití této knihovny zajišťuje, že každý elektronický důkaz má jedinečný identifikátor, což usnadňuje jeho správu a vyhledávání.
- **axios** – Axios je populární knihovna pro provádění HTTP požadavků. V této práci byla použita pro stahování aktiv (obrázků, scriptů atd.), protože je rychlejší než knihovna puppeteer, která je spíše vhodná pro stahování zdrojového kódu stránky.
- **cheerio** – Cheerio je knihovna pro manipulaci a extrakci dat z HTML souborů. V práci byla použita pro analýzu a extrakci dat ze zdrojového kódu.
- **url** – URL je vestavěná knihovna v Node.js, která poskytuje funkce pro práci s URL adresami. V projektu byla použita pro zpracování a validaci URL adres.
- **Puppeteer** – Puppeteer je knihovna pro Node.js, která poskytuje vysokoúrovňové API pro ovládání prohlížečů Chrome nebo Chromium. Tato knihovna byla klíčová pro zajišťování webových stránek jako elektronických důkazů. Díky své

schopnosti zpracovat a načíst všechny skripty na stránce před stažením zdrojového kódu zajišťuje, že veškerý obsah stránky je správně zaznamenán. Navíc umožňuje automatizovat webové prohlížeče a provádět akce jako načítání stránek, generování snímků obrazovky, manipulaci s DOM a mnoho dalších, což zvyšuje efektivitu a přesnost celého procesu [14].

4.2.2 Frontend

- **HTML** – Hypertext Markup Language (HTML) je základní značkovací jazyk používaný pro vytváření webových stránek. V projektu byl použit pro tvorbu struktury klientské části nástroje.
- **CSS** – Cascading Style Sheets (CSS) je jazyk používaný pro popis vzhledu a formátování webových stránek. V práci byl použit pro stylování klientské části nástroje.
- **JavaScript** – JavaScript je skriptovací jazyk používaný pro tvorbu interaktivních webových stránek. V projektu byl použit pro implementaci klientské části nástroje a interakci s uživatelem. Díky své všestrannosti a možnostem, které nabízí, je nezbytný pro tvorbu dynamických a responzivních webových stránek.
- **jQuery** – jQuery je rychlá a lehká JavaScript knihovna, která usnadňuje manipulaci s HTML dokumentem, práci s událostmi a animace. V této práci byla použita pro zjednodušení manipulace s DOM a práci s událostmi v klientské části nástroje, což výrazně zjednodušilo a zrychlilo vývoj.
- **Socket.io** (klientská část) – Klientská část Socket.io knihovny byla použita pro navázání realtime komunikace mezi klientskou a serverovou částí nástroje prostřednictvím webových soketů. To umožnilo rychlou a plynulou interakci mezi uživatelem a serverem, což zvyšuje uživatelskou zkušenost a efektivitu celého nástroje. [13]

4.3 Návrh nástroje

Cílem této kapitoly je poskytnout podrobný přehled o návrhu nástroje, který umožňuje uživatelům zajišťovat webové stránky jako elektronické důkazy a ověřovat jejich integritu. Nástroj je koncipován tak, aby byl schopen efektivně a bezpečně zpracovávat požadavky uživatelů a poskytovat kvalitní a spolehlivé služby. Tato kapitola se zaměřuje na klíčové aspekty návrhu nástroje zahrnující frontendové a backendové komponenty, komunikaci mezi nimi a strukturu elektronického důkazu.

Návrh nástroje zahrnuje rozdělení na frontend a backend, které umožňuje jasně oddělit uživatelské rozhraní od logiky aplikace. Toto rozdělení zvyšuje modularitu a usnadňuje údržbu a vývoj aplikace. Frontend zajišťuje intuitivní uživatelské rozhraní, které poskytuje uživatelům možnost snadno pochopit a ovládat nástroj, zatímco backend se stará o zpracování uživatelských požadavků a řízení celkového chodu aplikace.

V kontextu návrhu nástroje je důležité také zdůraznit strukturu elektronického důkazu. Struktura elektronického důkazu je navržena tak, aby byl snadno čitelný a poskytoval kompletní informace o zachycené webové stránce. Základní struktura důkazu zahrnuje všechny potřebné komponenty, jako jsou HTML soubory, externí zdroje, metadata a hashe, které zajišťují integritu a autenticitu informací.

4.3.1 Návrh frontendu

Návrh frontendu webové aplikace hraje klíčovou roli v poskytování uživatelsky přívětivého rozhraní, které usnadňuje zadávání nezbytných informací a poskytuje zpětnou vazbu o procesu zajišťování elektronických důkazů. Frontend navržený v této práci se zaměřuje na snadnou orientaci a intuitivní použití, přičemž využívá knihovnu socket.io pro komunikaci s backendem, jak bylo popsáno v předchozí kapitole.

Aplikace nabízí uživatelům dvě hlavní funkce: zajištění webové stránky jako elektronického důkazu a ověření integrity již zajištěné webové stránky. Při vstupu do nástroje uživateli aplikace nabídne dvě tlačítka, která reprezentují zmíněné možnosti.

Pokud si uživatel zvolí možnost zajištění webové stránky, zobrazí se formulář s jednoduchým rozhraním obsahujícím dvě vstupní pole. První pole slouží k zadání URL adresy webové stránky, která má být zachycena, zatímco druhé pole je určeno pro e-mailovou adresu uživatele, kam budou zaslány informace o elektronickém důkazu. Po odeslání formuláře pomocí tlačítka se spustí proces zajištění webové stránky. Po úspěšném dokončení procesu má uživatel možnost stáhnout soubor se zajištěnou webovou stránkou nebo pokračovat v zajištění dalších webových stránek.

Pro ověření integrity již zajištěné webové stránky se uživateli zobrazí jiný formulář s jedním vstupním polem, do kterého lze nahrát soubor s elektronickým důkazem. Po nahrání souboru a odeslání formuláře pomocí tlačítka aplikace provede ověření integrity a zobrazí uživateli výsledky tohoto procesu.

Celkový vizuální vzhled uživatelského rozhraní je navržen tak, aby byl jednoduchý a moderní, což zajišťuje přehlednost a snadnou orientaci pro uživatele. Tento návrh frontendu tak usnadňuje interakci s aplikací a umožňuje uživatelům efektivně využívat její funkce pro zajištění a ověřování webových stránek jako elektronických důkazů.

Při vývoji frontendu je důležité zohlednit i dostupnost a kompatibilitu s různými webovými prohlížeči a zařízeními. Aplikace by měla být responzivní, což znamená, že se automaticky přizpůsobí velikosti obrazovky a rozlišení uživatelova zařízení. Tím se zajišťuje optimální zobrazení a použitelnost na široké škále zařízení, včetně stolních počítačů, notebooků, tabletů a chytrých telefonů.

4.3.2 Návrh backendu

V této kapitole je poskytnut detailní pohled na návrh a architekturu backendu nástroje. Správná funkčnost, zpracování dat a komunikace s frontendem jsou zajišťovány backendem, aby bylo uživatelům umožněno plynulé a bezproblémové použití aplikace.

Backend byl postaven na platformě Node.js, která je považována za ideální pro vývoj škálovatelných a výkonných webových aplikací. Pro efektivní routování byla využita knihovna Express.js a pro real-time komunikaci mezi backendem a frontendem byla použita knihovna Socket.io.

V rámci nástroje byla jako databáze použita MongoDB. MongoDB je špičková NoSQL databáze, která je ideální pro tento typ projektu z více důvodů. MongoDB používá formát JSON pro ukládání dat, což je velmi příznivé pro JavaScriptové prostředí Node.js, protože nevyžaduje žádné převody dat. Toto zajišťuje rychlejší a plynulejší práci s daty a usnadňuje komunikaci mezi databází a aplikačním serverem. Dalším důvodem je flexibilita schématu MongoDB, která umožňuje ukládat data bez nutnosti předdefinování pevných struktur, což může být výhodné v případě, kdy se struktura dat může měnit nebo rozšiřovat v průběhu vývoje aplikace [15].

V průběhu vývoje bylo implementováno několik doplňkových funkcí pro zajištění kompletního zachycení webové stránky. Jedna z nich je využití knihovny Puppeteer pro stažení zdrojového kódu stránky. Puppeteer umožňuje čekání na kompletní načtení a běh všech skriptů, což zajišťuje, že všechny prvky stránky jsou správně viditelné. Puppeteer byl také využit pro vytvoření snímků webových stránek ve formátu PNG. Konkrétně snímků mobilní verze (s využitím rozlišení obrazovky iPhone X) a také desktop verze. Tak jako u aktiv a HTML kódu, i u snímků je vytvářen hash pro ověření integrity.

Struktura backendu aplikace je následující:

- app.js: hlavní soubor aplikace
- routes.js: soubor obsahující definice cest pro směrování pomocí Express.js
- /controllers/: složka modulů pro kontrolování logiky aplikace
- /helpers/: složka obsahující funkce pro stahování zdrojových kódů a aktiv, analýzu HTML a CSS a aktualizaci odkazů v HTML pro offline použití
- /sockets/: složka obsahující modul socketManager.js pro správu komunikace mezi backendem a frontendem
- /config/: složka obsahující moduly pro konfiguraci databáze a definici cest (path)
- /utils/: složka s moduly pro různé účely, jako je vytváření hashů, odesílání e-mailů, TSA modul, modul pro vytváření ZIP souborů a modul pro kontrolu URL.

V aplikaci jsou implementovány dva základní procesy: zajištění webové stránky jako elektronického důkazu a ověření integrity již zajištěné webové stránky. Proces zajištění webové stránky zahrnuje kontrolu URL, stažení zdrojového kódu, analýzu a extrakci dat z HTML a CSS, stažení aktiv, aktualizaci odkazů pro offline použití, vytvoření hashů,

vytvoření snímků webové stránky a uložení dat do lokální databáze MongoDB. Po dokončení těchto kroků je uživateli nabídnut soubor ZIP ke stažení.

Proces ověření integrity zahrnuje kontrolu názvu nahraného souboru, zjištění záznamu v databázi, extrakci hashů z databáze, rozbalení ZIP souboru do dočasné složky, vytvoření hashů jednotlivých souborů (HTML, aktiv a snímků) a porovnání s hashemi uloženými v databázi. Pokud je hash celého ZIP souboru shodný s tím, který je uložen v databázi, další hashe samostatných souborů se neověřují a ZIP soubor nemusí být rozbalován. Výsledky ověření jsou poté zaslány uživateli na frontend.

Celkový návrh backendu byl zaměřen na efektivitu a přehlednost, s důrazem na modularitu a znovu-použitelnost kódu. To usnadňuje údržbu aplikace a její rozšiřování a modifikaci v budoucnu. Současně byla zajištěna snadná integrace s frontendem, což umožňuje plynulou komunikaci mezi oběma částmi aplikace a vede k lepšímu uživatelskému zážitku.

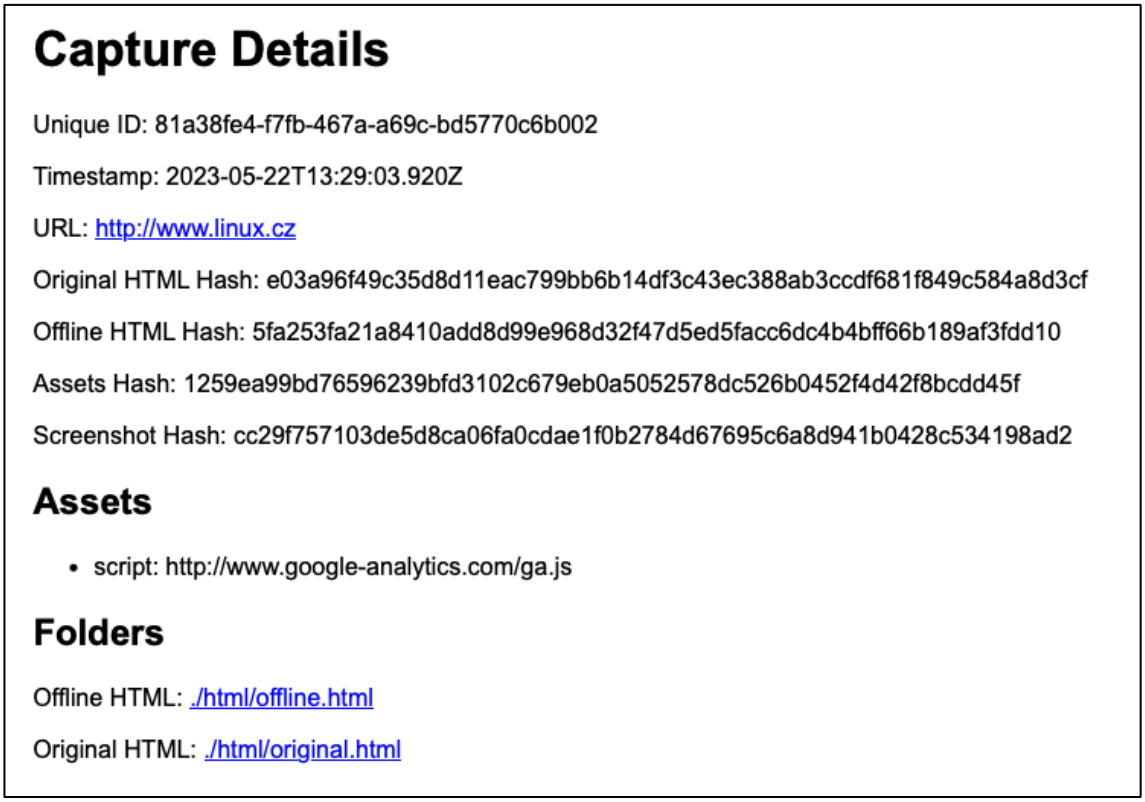
4.3.3 Návrh struktury elektronického důkazu

V této podkapitole bude podrobněji probrána struktura elektronického důkazu, který obsahuje zajištěnou webovou stránku a související informace. Elektronický důkaz je uložen ve formě komprimovaného ZIP souboru, který poskytuje uživatelům snadný způsob, jak stahovat a přenášet důkaz jako jednotný celek, aniž by museli stahovat každou součást zvlášť. Tento přístup je zvláště důležitý v případě, kdy důkaz zahrnuje velké množství souborů.

Struktura elektronického důkazu je následující:

- ZIP soubor: Soubor je pojmenován podle unikátního identifikátoru, který jednoznačně označuje konkrétní zajištění webové stránky.
- /assets/: Tato složka obsahuje další podsložky, které jsou rozděleny podle typu souborů (např. stylesheets, scripts, img, background-image atd.). Složky se vytvářejí dle potřeby, v závislosti na obsahu zajištěné webové stránky. V podsložkách jsou pak samotná aktiva stažena z webové stránky.
- /html/: Tato složka obsahuje dva soubory - offline.html a original.html. Offline.html má upravené odkazy na aktiva tak, aby odkazovaly na soubory uložené ve složce /assets/. Original.html obsahuje původní zdrojový kód stránky tak, jak byl stažen.
- /screenshots/: Složka, která obsahuje dva PNG soubory se snímky obrazovky zajištěné stránky – jeden snímek pro mobilní verzi a druhý snímek pro desktop verzi.
- capture-details.html: Tento HTML soubor je automaticky generován při každém zajištění webové stránky a obsahuje řadu informací o důkazu. Zahrnuje hash hodnoty aktiv, hash hodnoty HTML, časové razítko, unikátní identifikátor procesu zajištění a kompletní seznam stažených aktiv pro danou stránku.

Ukázku obsahu capture-details.html ze zajištění webové stránky http://www.linux.cz [16], je možné vidět na níže přiloženém obr. 4.1. Přidání snímků webové stránky do struktury elektronického důkazu poskytuje další vrstvu důkazních informací ke zdrojovému kódu stránky. Snímky webové stránky poskytují rychlý a jednoduchý způsob, jak vizuálně potvrdit obsah a vzhled stránky v době zajištění, což může být zvláště užitečné pro osoby, které nejsou technicky zdatné a nemohou snadno číst nebo interpretovat zdrojový kód.



Capture Details

Unique ID: 81a38fe4-f7fb-467a-a69c-bd5770c6b002

Timestamp: 2023-05-22T13:29:03.920Z

URL: http://www.linux.cz

Original HTML Hash: e03a96f49c35d8d11eac799bb6b14df3c43ec388ab3ccdf681f849c584a8d3cf

Offline HTML Hash: 5fa253fa21a8410add8d99e968d32f47d5ed5facc6dc4b4bff66b189af3fdd10

Assets Hash: 1259ea99bd76596239bfd3102c679eb0a5052578dc526b0452f4d42f8bcdd45f

Screenshot Hash: cc29f757103de5d8ca06fa0cdae1f0b2784d67695c6a8d941b0428c534198ad2

Assets

- script: http://www.google-analytics.com/ga.js

Folders

Offline HTML: ./html/offline.html

Original HTML: ./html/original.html

Obr. 4.1: Ukázka obsahu souboru capture-details.html

4.4 Implementace nástroje

Tato kapitola se zaměřuje na klíčové aspekty implementace nástroje, který byl vyvinut jako součást této bakalářské práce. Popisuje, jak je nástroj konstruován a jak funguje. Nástroj, který byl v rámci bakalářské práce vyvíjen, je komplexní systém pro zajištění webových stránek a jejich následnou analýzu.

V následující podkapitole budeme zkoumat klíčové funkce tohoto nástroje, které jsou nezbytné pro jeho funkčnost a které představují jedinečné a inovativní řešení problémů spojených s tímto konkrétním oborem.

4.4.1 Důležité funkce nástroje

Stažení zdrojového kódu pomocí knihovny Puppeteer

Jedním z kritických kroků v procesu extrakce dat je stažení zdrojového HTML kódu webové stránky. V tomto projektu je tato operace provedena pomocí funkce downloadHtml, která využívá knihovnu Puppeteer. Puppeteer je knihovna poskytující vysokoúrovňové API pro ovládání prohlížeče Chrome nebo Chromium přes protokol DevTools. Ukázku funkce donwloadHtml pro stažení zdrojového kódu stránky lze vidět ve výpisu 4.1.

Výpis 4.1: Ukázka kódu z funkce donwnloadHtml

```
const downloadHtml = async (url, screenshotFolderPath, socket) => {
  try {
    const browser = await puppeteer.launch({
      headless: 'new',
    });
    const page = await browser.newPage();
    await page.setViewport({ width: 1440, height: 900 });
    await page.goto(url, {waitUntil: 'networkidle2'});
    await page.screenshot({ path: `${screenshotFolderPath}/normal-
screenshot.png`, fullPage: true });
    const html = await page.content();

    await page.emulate(iPhone);
    await page.goto(url, {waitUntil: 'networkidle2'});
    await page.screenshot({ path: `${screenshotFolderPath}/mobile-
screenshot.png`, fullPage: true });

    await browser.close();

    return html;
  } catch (error) {
    socket.emit('error', 'Server is not able to access that website.',
'website-capture' );
    console.error(`Error downloading HTML: ${error.message}`);
  }
};
```

Funkce downloadHtml začíná spuštěním instance prohlížeče pomocí puppeteer.launch(). Následuje vytvoření nové stránky (browser.newPage()) a nastavení viewportu, což ovlivňuje zobrazení obsahu webové stránky. Poté funkce naviguje na cílovou URL pomocí page.goto(url, {waitUntil: 'networkidle2'}). Parametr {waitUntil: 'networkidle2'} určuje, že funkce počká, až bude dokončena většina síťových požadavků, což zajišťuje, že je obsah stránky kompletně načten.

Dále funkce vytvoří screenshot celé stránky pomocí page.screenshot(), což umožňuje vizuální kontrolu obsahu stránky. Proces vytvoření screenshotu se děje dvakrát, jednou pro normální zobrazení a podruhé pro zobrazení na mobilním zařízení (iPhone X v tomto případě) po emulaci zařízení pomocí page.emulate(iPhone).

Nakonec je získán HTML obsah stránky pomocí page.content(), prohlížeč je uzavřen pomocí browser.close() a HTML obsah je vrácen jako výstup funkce. Veškeré chyby zachytává blok catch, který vyšle chybovou zprávu přes socket komunikaci a zaznamená chybu do konzole.

Funkce downloadHtml je důležitou součástí procesu extrakce dat, protože zajišťuje, že je HTML obsah stránky k dispozici pro další zpracování. Je také flexibilní v tom, že umožňuje emulaci různých zařízení pro získání různých verzí stránky.

Analýza HTML kódu a extrakce dat

Jedním z klíčových aspektů tohoto projektu je analýza HTML kódu a extrakce dat, kterou zajišťuje funkce parseHtml. Tato funkce přijímá HTML obsah a základní URL jako parametry a vrací seznam aktiv, které jsou v HTML dokumentu obsaženy. Jedná se o sofistikovanou funkci, která využívá několik pomocných funkcí a technologií. Ukázku části kódu z této funkce lze vidět ve výpisu 4.2.

Funkce parseHtml začíná načítáním HTML pomocí balíčku cheerio, což je rychlá, flexibilní a efektivní implementace jádra jQuery pro server. Vytváří se prázdné pole assets, které bude postupně naplněno objekty reprezentujícími zdroje nalezené v HTML dokumentu.

Funkce resolveUrl je pomocná funkce použitá k určení úplné URL adresy z relativních cest, zatímco funkce isValidUrl ověřuje, zda je daná URL adresa platná.

Následující část kódu pak prochází HTML dokumentem a vyhledává různé typy elementů, jako jsou styly (<style>), odkazy na styly (<link rel="stylesheet">), skripty (<script>) a obrázky (<img>). Pro každý nalezený element extrahuje URL zdroje, ověří její platnost a přidá ji do pole assets.

Výsledkem této funkce je tedy seznam objektů, kde každý objekt reprezentuje jeden zdroj nalezený v HTML dokumentu, a to včetně typu zdroje a jeho URL adresy.

Funkce parseHtml je tak klíčová pro výkonnost tohoto projektu, protože umožňuje efektivně analyzovat a extrahovat data z HTML dokumentů, čímž zajišťuje, že všechny potřebné zdroje jsou k dispozici pro další zpracování.

Výpis 4.2: Ukázka kódu z funkce parseHtml

```
function parseHtml(html, baseUrl) {
  const $ = cheerio.load(html);
  let assets = [];

  function resolveUrl(href) {return url.resolve(baseUrl, href);}

  function isValidUrl(url) {
    if (url.startsWith('data:')) {
      return false;
    }
    try {
      new URL(url);
      return true;
    } catch (_) {return false;}
  }
  $('link[rel="stylesheet"]').each((index, element) => {
    const oldHref = $(element).attr('href');
    if (oldHref) {
      const href = resolveUrl(oldHref);
      if (isValidUrl(href)) {
        assets.push({ type: 'stylesheet', url: href, oldUrl: oldHref
});
      }
    }
  });
  //... podobné extrakce pro další elementy ...
  return assets;
}
```

Analýza CSS a extrakce obrázku na pozadí

V procesu zpracování webových stránek je často nutné analyzovat a extrahovat informace z CSS (Cascading Style Sheets). Toto umožňuje například identifikovat obrázky použité na pozadí různých elementů na stránce. V našem nástroji tuto funkcionalitu zajišťuje funkce parseCss. Ukázku funkce parseCss lze vidět ve výpisu 4.3.

Výpis 4.3: Ukázka kódu pro extrakci obrázků z CSS

```
function extractBackgroundImageUrl(style) {
  const regex = /background-image:\s*url\(['"]?(.*?)['"]?\)/;
  const match = regex.exec(style);
  return match ? match[1] : null;
}

function parseCss(cssContent, assets, baseUrl) {
  const regex = /(background(?:-image)?\s*:[^;}]+)/g;
  let match;
  const backgroundImages = [];

  while ((match = regex.exec(cssContent)) !== null) {
    const backgroundImageUrl = extractBackgroundImageUrl(match);
    if (backgroundImageUrl) {
      const oldUrl = backgroundImageUrl;
      const url = resolveUrl(oldUrl, baseUrl);
      if (isValidUrl(url)) {
        backgroundImages.push({ type: 'background-image', url: url,
oldUrl: oldUrl });
      }
    }
  }

  return [...assets, ...backgroundImages];
}
```

Funkce parseCss začíná voláním pomocné funkce extractBackgroundImages, která vyhledává vstupní CSS kód (argument cssContent) pomocí regulárního výrazu. Tento regulární výraz detekuje CSS vlastnosti, které mohou definovat obrázek na pozadí. Pokud je nalezena taková vlastnost, je z ní extrahována URL obrázku pomocí další pomocné funkce extractBackgroundImageUrl.

Tato URL je pak normalizována a ověřena, a pokud je platná, je přidána do seznamu zdrojů. Výsledkem funkce extractBackgroundImages je pak pole se všemi nalezenými obrázky na pozadí.

Tyto obrázky jsou poté spojeny s předchozími zdroji (assets) a vráceny jako výstup funkce parseCss. Tímto způsobem umožňuje funkce parseCss extrahovat a integrovat obrázky na pozadí do celkového zpracování webové stránky.

Vytvoření digitálního otisku pomocí HASH funkce

Hash funkce jsou v informatice a počítačové vědě klíčové pro zajištění integrity a unikátnosti dat. Pomocí hash funkcí můžeme vytvářet jedinečné identifikátory, tzv. otisky, pro libovolná data. Toto je velmi důležité například pro zjištění, zda byla data mezi dvěma body přenesena správně a zda nebyla poškozena nebo modifikována. Funkce calculateFileHash v našem nástroji plní právě tuto roli. Ukázku funkce pro výpočet HASH hodnoty lze vidět ve výpisu 4.4.

Výpis 4.4: Ukázka kódu pro výpočet HASH funkce

```
function calculateFileHash(filePath, algorithm = 'sha256') {
  return new Promise((resolve, reject) => {
    const hash = crypto.createHash(algorithm);
    const stream = fs.createReadStream(filePath);

    stream.on('data', (chunk) => {
      hash.update(chunk);
    });

    stream.on('end', () => {
      resolve(hash.digest('hex'));
    });

    stream.on('error', (error) => {
      reject(error);
    });
  });
}
```

Funkce calculateFileHash začíná vytvořením instance hashovací funkce pomocí modulu crypto v Node.js. Tato hashovací funkce je inicializována s algoritmem, který je definován jako parametr funkce ('sha256').

Dále je vytvořen datový proud (stream) pomocí fs.createReadStream, který umožňuje postupné čtení souboru definovaného vstupní cestou (filePath). Jakmile je proud vytvořen, je zaregistrováno několik událostí.

Při události data, což znamená, že byl přečten kus dat ze souboru, je hashovací funkce aktualizována těmito daty pomocí metody hash.update(chunk).

Při události end, což znamená, že byl celý soubor přečten, je finální hash vygenerován a vrácen jako výsledek Promise pomocí resolve(hash.digest('hex')). Tento hash je v hexadecimální formě.

Při události error, což znamená, že došlo k chybě při čtení souboru, je tato chyba předána dál a Promise je odmítnuta pomocí reject(error).

Funkce calculateFileHash tak umožňuje efektivní a snadné generování hashů souborů, což je základní kámen pro zajištění integrity dat.

4.4.2 Implementace nástroje na produkční server

Po úspěšném vývoji nástroje v lokálním prostředí byl zahájen proces jeho nasazení na produkční server. Tento proces měl za cíl vytvořit stabilní a bezpečné prostředí pro nástroj, které by umožňovalo snadné a pohodlné použití, a také širokou dostupnost pro veřejnost.

Prvním krokem v tomto procesu bylo vytvoření virtuálního privátního serveru (VPS) na platformě Hostinger. Tento server poskytl potřebné prostředí pro běh aplikace. Byl zvolen operační systém Ubuntu 18.04 64bit s Webminem pro jeho spolehlivost, podporu a širokou uživatelskou základnu.

Následovala registrace domény websitecatcher.com, která se stala místem, kde je nástroj volně dostupný pro veřejnost. Tato adresa byla zvolena pro její jednoduchost, snadnou zapamatovatelnost a proto, že výstižně reflektuje podstatu nástroje – zachycování webových stránek. Poté byl kód nástroje přenesen na server a začal proces instalace potřebných nástrojů a knihoven pro jeho běh. Tento proces zahrnoval především instalaci Node.js a ostatních závislostí, které jsou definovány v konfiguračním souboru aplikace.

V rámci snahy o co největší bezpečnost nástroje během jeho provozu na produkčním serveru byl vytvořen speciální uživatel na serveru. Toto rozhodnutí bylo učiněno s cílem zabránit běhu aplikace s právy superuživatele (root), což je obecně považováno za nebezpečné. Pro zajištění bezpečného připojení byl použit nástroj Certbot k vygenerování certifikátu LetsEncrypt a server byl přes Nginx nastaven tak, aby využíval zabezpečené https připojení.

Jako poslední krok byl použit procesový manažer PM2 pro spouštění webové aplikace. PM2 je robustní řešení pro správu aplikací Node.js s mnoha funkcemi, včetně automatického restartu aplikace v případě pádu, udržování logů aplikace a jednoduché správy procesů.

Nakonec, výsledkem celého procesu je plně funkční webový nástroj, hostovaný na vlastním VPS, s vysokou dostupností a zabezpečeným připojením. Toto nasazení nástroje na produkční server bylo klíčovou částí celého projektu a umožnilo, aby nástroj mohl být využíván širokou veřejností.

4.5 Testování nástroje

Tato kapitola se zaměřuje na testování navrženého nástroje pro zajištění webových stránek jako elektronických důkazů a ověřování jejich integrity. Testování je klíčovým krokem vývoje, který umožňuje identifikovat a odstranit chyby a zajišťuje spolehlivou a kvalitní funkčnost nástroje.

4.5.1 Jednotkové testy

Nejdříve byly provedeny jednotkové testy, které cílily na různé moduly nástroje, konkrétně moduly download, parseCss, parseHtml a updateLinks. Pro jednotkové testování byl využit framework Jest [17], který se osvědčil jako vhodné řešení. Tento framework umožňuje automatizované spouštění testů a poskytuje nástroje pro tvorbu asertivních výroků a kontrolu správnosti implementace. Ukázku kódu jednotkového testu pro modul updateLinks je zobrazen ve výpisu 4.5.

Výpis 4.5: Ukázka kódu jednotkového testu pro modul updateLinks

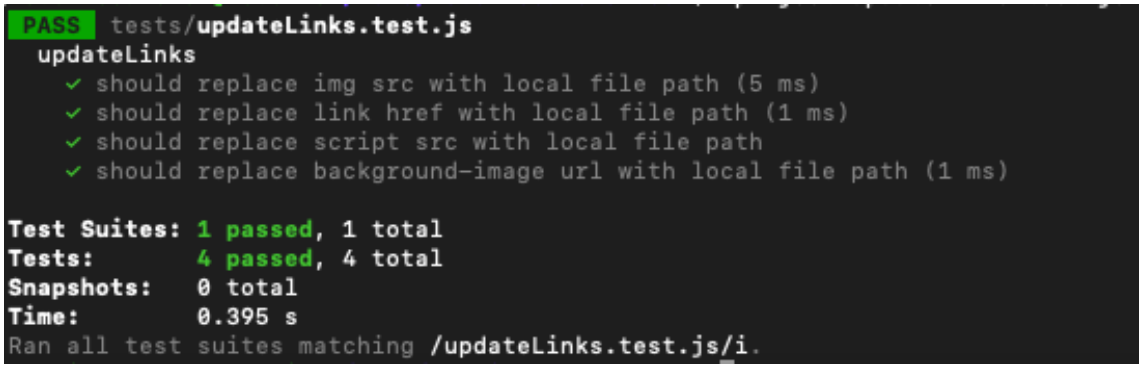
```
describe('updateLinks', () => {
  it('should replace img src with local file path', () => {
    const mockHtml = '<html><body><img
src="http://example.com/old.png"></body></html>';
    const mockAssets = [
      {
        type: 'img',
        url: 'http://example.com/new.png',
        oldUrl: 'http://example.com/old.png',
      }
    ];
    const mockOutputPath = 'mockOutputPath.html';

    fs.writeFileSync.mockImplementation(() => {});

    updateLinks(mockHtml, mockAssets, mockOutputPath);

    expect(fs.writeFileSync).toHaveBeenCalledWith(
      mockOutputPath,
      '<html><body><img src="../assets/img/new.png"></body></html>'
    );
  });
});
```

Výsledky jednotkových testů byly získány prostřednictvím běhu testovacích scénářů, které zahrnovaly různé varianty zápisu odkazů na obrázky a různé formáty stylů background-image. Zmíněné testování umožnilo identifikovat a odstranit chyby v aplikaci a přispělo k jejímu vylepšení. Všechny jednotkové testy po opravě chyb proběhly úspěšně. Na obr. 4.2 je možné vidět výsledek úspěšného jednotkového testu modulu updateLinks.



Obr. 4.2: Ukázka výsledku jednotkového testu modulu updateLinks

Jednotkové testování přineslo významnou přidanou hodnotu nástroji, neboť umožnilo zabezpečit jeho správnou funkčnost a identifikovat potenciální chyby již v rané fázi vývoje. Díky těmto testům byly provedeny nezbytné opravy a optimalizace, což přispělo k vytvoření stabilní a spolehlivé aplikace.

4.5.2 Testování komunikace mezi serverem a frontendem

Po dokončení jednotkových testů bylo provedeno testování celkové komunikace mezi serverem a frontendem, využívající technologii socket.io. Toto testování mělo za cíl ověřit správné propojení obou částí aplikace a zajištění bezproblémového chodu.

Během testování bylo zjištěno, že frontend se od serveru odpojoval hned poté, co uživatel klikl na tlačítko "download" po zajištění webové stránky jako elektronického důkazu. Tento problém byl identifikován a vyřešen přidáním atributu do HTML kódu odkazu na download, který zajišťuje, že se odkaz otevře v novém okně. Tímto způsobem byla navázána správná komunikace mezi frontendem a serverem pomocí socket.io a problém s odpojením byl vyřešen.

Po řešení tohoto problému komunikace mezi frontendem a serverem prostřednictvím socket.io fungovala perfektně a nástroj byl schopen správně přenášet data mezi oběma částmi aplikace.

4.5.3 Testování generování elektronických důkazů

Dalším krokem testování bylo ověření funkcionality nástroje při generování elektronických důkazů z webových stránek. Tato část testování zahrnovala kontrolu správného obsahu vytvořeného elektronického důkazu, stahování potřebných aktiv a generování offline verze webové stránky, která načítá lokální obrázky, skripty atd.

Testování probíhalo na různých typech webových stránek, včetně prezentačních a e-commerce stránek. E-commerce stránky jsou webové stránky, které umožňují obchodní transakce a nákupy zákazníkům. Během testování byly zkoušeny stránky využívající jak protokol HTTPS, tak protokol HTTP, aby byla zajištěna celková funkčnost nástroje v různých prostředích.

Během testování generování elektronických důkazů byla také testována rychlost tohoto procesu. Proces generování u většiny webových stránek trval mezi 15 až 20 sekundami. Avšak u webové stránky, která generuje neustále další obsah při posouvání na konec stránky, byl proces generování delší a trval až minutu nebo déle.

Tato zvýšená doba generování byla způsobena tvorbou screenshotu webové stránky, která obsahovala velké množství obsahu. Přestože proces generování u této stránky trval déle, výsledky byly stále spolehlivé a kompletní.

Výsledky testování generování elektronických důkazů byly velmi úspěšné. Nástroj byl schopen správně zachytit webovou stránku a vytvořit kompletní elektronický důkaz obsahující všechny potřebné komponenty. Testy na různých typech webových stránek byly úspěšné a nástroj se osvědčil při zajišťování jak prezentačních stránek, tak e-commerce stránek.

4.5.4 Testování zpětného ověření integrity

Posledním krokem testování bylo zpětné ověřování integrity již zajištěných webových stránek jako elektronických důkazů. Během tohoto testování byly prováděny různé změny v elektronickém důkazu a ověřovalo se, zda nástroj správně identifikuje tyto změny.

Všechny změny provedené v elektronickém důkazu byly nástrojem správně identifikovány a výsledky testování potvrdily, že nástroj je schopen spolehlivě ověřit integritu již zajištěných webových stránek.

4.5.5 Závěr testování

Testování vyvíjeného nástroje bylo realizováno s maximální důsledností a precizností, aby byla zajištěna jeho kvalita a spolehlivost. Vykonávány byly jednotkové testy, testování komunikace mezi serverem a frontendem, testování generování elektronických důkazů a ověřování jejich integrity, což přineslo cenné výsledky a poznatky.

Jednotkové testy napomohly identifikovat a odstranit chyby v různých modulech nástroje, čímž došlo ke zlepšení jeho stability. Komunikace mezi serverem a frontendem byla důkladně ověřena, potvrzující správnou funkcionalitu tohoto propojení za použití technologie socket.io. Testování generování elektronických důkazů ukázalo, že nástroj je spolehlivý a úspěšný v zachycování webových stránek a generování kompletních důkazů. Ověřování integrity elektronických důkazů prokázalo, že nástroj je schopný správně identifikovat jakékoliv změny v nich.

Nicméně, v průběhu testování byl zaznamenán problém při generování offline verze jedné specifické webové stránky. Tato komplikace vyplynula z nesrovnalostí mezi odkazy, které byly zahrnuty v kódu zajišťované webové stránky, a těmi, které byly určeny k nahrazení v kódu. Hlavní příčinou bylo rozdílné zastoupení znaku "&". Zatímco HTML odkazy obsahovaly HTML entity pro ampersandy ("&"), odkazy v assetech obsahovaly

pouze znaky "&". Tato nesouladnost vedla k selhání vyhledávací funkce RegExp, která požadovala přesnou shodu řetězců.

Problém byl nakonec vyřešen úpravou funkce updateLinks, která byla modifikována tak, aby nahradila znaky ampersandu ("&") v odkazech assetů HTML entitami pro ampersand ("&"). Díky této změně bylo dosaženo shody mezi odkazy v HTML a odkazy v assetech, což umožnilo jejich správnou náhradu.

Je důležité uvést, že tvorba každé stránky může být provedena různými metodami a funkce nahrazující odkazy v offline verzi webové stránky jako elektronického důkazu nemusí být vždy perfektně zvládnuta. Nicméně, tento problém by neměl nutně znamenat, že aktiva jako obrázky nebo další data by nebyla do elektronického důkazu stahována, a tudíž by ztráta důkazních informací neměla nastat. Interpretace elektronického důkazu by však mohla být zkomplikována.

V konečném souhrnu lze konstatovat, že testování nástroje přineslo úspěšné výsledky a poskytlo důležité poznatky o jeho funkčnosti, spolehlivosti a výkonu. Díky provedeným testům je možné mít důvěru v kvalitu a efektivitu nástroje při zajišťování webových stránek jako elektronických důkazů a následné ověřování jejich integrity.

5. ZÁVĚR

Tato bakalářská práce zmapovala klíčové aspekty zachycení obsahu webových stránek pro účely zajištění důkazní kvality. Byla provedena analýza právních předpisů a rozhodovací praxe, na jejímž základě byly stanoveny hlavní požadavky na řešení tohoto problému. Byl vyvinut nástroj, který tyto požadavky reflektuje a umožňuje uživatelům zajišťovat obsah webových stránek jako elektronické důkazy.

Zkoumání a ověřování funkcionalit nástroje odhalilo schopnost zachycovat webové stránky různých formátů a struktur, stejně jako efektivní ověřování integrity zachycených dat. Tato schopnost byla prokázána prostřednictvím série testů, což podpořilo důvěryhodnost nástroje jako nástroje pro ochranu integrity elektronického důkazu.

Nicméně, výzkum také odhalil několik oblastí, které mohou sloužit jako východisko pro další vývoj. Jednou z takových oblastí je zachycování obsahu webových stránek tak, jak ho vidí uživatel, nikoli server. V současné době se nástroj omezuje na serverové zobrazení, což může způsobit problémy při zachycování stránek chráněných heslem nebo stránek s dynamickým obsahem.

Možné řešení tohoto omezení by mohlo spočívat ve vývoji doplňku pro webový prohlížeč, který by umožnil zachycení obsahu stránek tak, jak ho vidí uživatel. Tento přístup by mohl také umožnit zachycení stránek, které jsou přístupné pouze některým uživatelům, jako jsou chráněné příspěvky na Facebooku nebo stránky chráněné heslem.

Přestože tato bakalářská práce představuje úspěšný nástroj v oblasti zajišťování webového obsahu jako elektronického důkazu, je zde stále mnoho oblastí pro další výzkum a vývoj. Je důležité pokračovat v těchto snahách a zdokonalovat nástroje a techniky pro zachycení webového obsahu jako elektronického důkazu, aby bylo možné lépe reagovat na rychle se měnící technologický kontext.

Doufám, že tato práce bude sloužit jako základní kámen pro další práci v této důležité a rychle se rozvíjející oblasti práva a informatiky, poskytne věrohodný základ pro další výzkum a vývoj a přispěje k větší pochopení a řešení problémů spojených s webovým obsahem jako elektronickým důkazem.

LITERATURA

[1] POLČÁK, Radim, František PÚRY a Jakub HARAŠTA. Elektronické důkazy v trestním řízení. Brno: Masarykova univerzita, 2015. ISBN 978-80-210-8073-7.

[2] Teoretický základ a přehled kryptografických hashovacích funkcí. Masarykova univerzita [online]. Masarykova univerzita: Radim Ošťádal, 2012 [cit. 2022-12-11]. Dostupné z: https://is.muni.cz/www/ostadal/hash_overview.pdf

[3] MD5. Tech Target [online]. Tech Target: Mary E. Shacklett, Transworld Data and Peter Loshin, Senior Technology Editor, August 2021 [cit. 2022-12-12]. Dostupné z: https://www.techtarget.com/searchsecurity/definition/MD5

[4] SHA-1: What It Is & How It's Used for Data Verification. LifeWire [online]. LifeWire: Tim Fisher, September 15, 2022 [cit. 2022-12-12]. Dostupné z: https://www.lifewire.com/what-is-sha-1-2626011

[5] Hash functiions. NIST CSRC [online]. NIST: CSRC, Created January 04, 2017 [cit. 2022-12-12]. Dostupné z: https://csrc.nist.gov/projects/hash-functions

[6] What is SHA-2 and how does it work?. Comparitech [online]. Comparitech: Josh Lake, February 17, 2022 [cit. 2022-12-12]. Dostupné z: https://www.comparitech.com/blog/information-security/what-is-sha-2-how-does-it-work/

[7] SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION [online]. Gaithersburg: Information Technology Laboratory, August 2015 [cit. 2022-12-12]. Dostupné z: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

[8] Hash Algorithm Comparison: MD5, SHA-1, SHA-2 & SHA-3. Code Signing Store [online]. Code Signing Store, b. r. [cit. 2022-12-12]. Dostupné z: https://codesigningstore.com/hash-algorithm-comparison

[9] PrintFarm.cz | 3D tisk a modelování na zakázku | Aditivní výroba. PrintFarm.cz | Jan Rezek [online]. Copyright © 2021 Jan Rezek [cit. 11.12.2022]. Dostupné z: https://printfarm.cz

[10] Nejvyšší správní soud. (2016). Rozsudek ze dne 24. 8. 2016, sp. zn. 1 As 80/2016-30 [Online] Dostupné z: https://www.nssoud.cz/stazeni-dokumentu?filepath=SOUDNI_VYKON/2016/0080_1As__1600030_20160919151950_prevedeno.pdf

[11] About Node.js. Node.js [online]. Node.js, r. n. [cit. 2022-12-12]. Dostupné z: https://nodejs.org/en/about/

[12] Express.js. Express.js [online]. StrongLoop, IBM, and other expressjs.com contributors, Copyright © 2017 [cit. 2022-12-12]. Dostupné z: https://expressjs.com

[13] Socket.IO: Dvousměrná a nízkolatenční komunikace pro každou platformu. In: socket.io [online]. [cit. 2023-05-23]. Dostupné z: https://socket.io

[14] Puppeteer: Node.js knihovna poskytující vysokoúrovňové API pro ovládání Chrome/Chromium přes protokol DevTools. In: pptr.dev [online]. [cit. 2023-05-23]. Dostupné z: https://pptr.dev

[15] Why Use MongoDB And When To Use It? | MongoDB. MongoDB: The Developer Data Platform | MongoDB [online]. Copyright © 2022 MongoDB, Inc. [cit. 11.12.2022]. Dostupné z: https://www.mongodb.com/why-use-mongodb

[16] Linux.cz [online]. [cit. 2023-05-25]. Dostupné z: http://www.linux.cz

[17] Jest: A delightful JavaScript Testing Framework with a focus on simplicity. In: jestjs.io [online]. [cit. 2023-05-23]. Dostupné z: https://jestjs.io

SEZNAM ZKRATEK

Zkratky:

TŘ	Trestní Řád
ZEK	Zákon o Elektronických Komunikacích
MD5	Message Digest Algorithm 5
SHA-1	Secure Hash Algorithm 1
SHA-2	Secure Hash Algorithm 2
SHA-3	Secure Hash Algorithm 3
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
DOM	Document Object Model
URL	Uniform Resource Locator
JSON	JavaScript Object Notation

SEZNAM PŘÍLOH

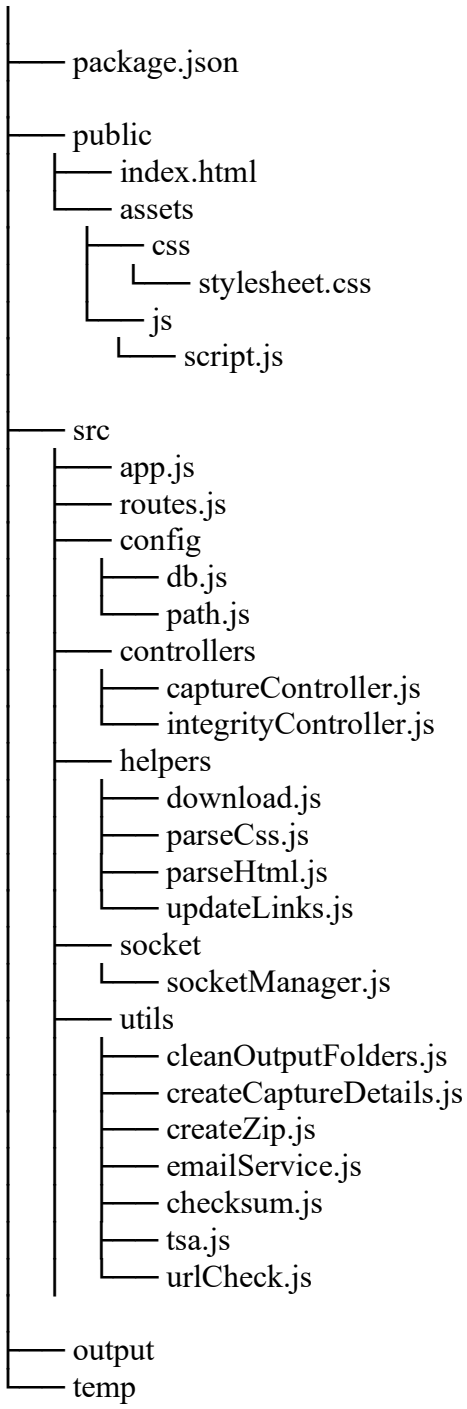
PŘÍLOHA A - OBSAH ELEKTRONICKÉ PŘÍLOHY 46

PŘÍLOHA B - MANUÁL K NÁSTROJI 47

Příloha A - Obsah elektronické přílohy

Elektronická příloha obsahuje soubory, které v souhrnu představují kompletní nástroj pro zajištění webových stránek jako elektronických důkazů.

Struktura elektronické přílohy:



Příloha B - Manuál k nástroji

Zajištění Webové Stránky Jako Elektronického Důkazu

1. Přístup k nástroji: Nástroj je přístupný jako webová aplikace na adrese www.websiteacatcher.com.
2. Výběr akce: Po příchodu na web se uživatel může rozhodnout mezi dvěma možnostmi – zajištění webové stránky jako elektronického důkazu nebo ověření integrity elektronického důkazu. V tomto případě bude zvolena první možnost.
3. Vstupní data: Po kliknutí na tlačítko pro zajištění webové stránky jako elektronického důkazu se zobrazí formulář, do kterého se vloží URL adresa webové stránky, kterou chce uživatel zajistit. Pole pro emailovou adresu je možné nechat prázdné, protože tato část nástroje zatím nebyla zprovozněna.
4. Zahájení procesu: Po vložení URL adresy uživatel klikne na tlačítko "Odeslat" a proces zajištění webové stránky jako elektronického důkazu začne.
5. Stažení elektronického důkazu: Jakmile je proces dokončen, na stránce se objeví zpráva o úspěšném zajištění webové stránky a možnost stáhnout soubor s elektronickým důkazem.
6. Zajištění další webové stránky: Pokud uživatel chce zajistit další webovou stránku, může kliknout na tlačítko "Capture another website" a opakovat kroky 3 až 5.

Ověření Integrity Elektronického Důkazu

1. Výběr akce: Na úvodní stránce nástroje uživatel klikne na tlačítko pro ověření integrity elektronického důkazu.
2. Vstupní data: Následně se zobrazí formulář se vstupním polem pro soubor s elektronickým důkazem a tlačítkem pro odeslání.
3. Příprava souboru pro ověření: Uživatel musí mít k dispozici uložený soubor s elektronickým důkazem. Pokud má původní ZIP soubor, může ho rovnou nahrát. V případě, že původní ZIP soubor byl rozbalen a nyní už není k dispozici, je nutné ho znovu vytvořit se stejným názvem, který obsahuje unikátní identifikátor. Tento identifikátor lze nalézt v souboru capture-details.html obsaženém v elektronickém důkazu. Soubor s elektronickým důkazem by tedy měl vypadat např. takto: b1c1c93d-6a35-4347-b40a-bc968a318874.zip.
4. Odeslání souboru: Po nahrání souboru uživatel klikne na tlačítko "Odeslat" a proces ověření integrity začne.
5. Výsledky ověření: Pokud je soubor původní, tak se ověří hash celého ZIP souboru a už se neověřují samotné části elektronického důkazu. Pokud bylo s ZIP souborem manipulováno, hash původního ZIP souboru se nebude shodovat a musí se ověřit samostatné části elektronického důkazu. Výsledky ověření jsou okamžitě zobrazeny uživateli.