VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA SKRIPTOV PRE ZAZNAMENÁVANIE SEDENIA

ANALYSIS OF SESSION REPLAY SCRIPTS

BAKALÁRSKA PRÁCA
BACHELOR'S THESIS

AUTOR PRÁCE — AUREL STRIGÁČ
AUTHOR

VEDÚCI PRÁCE — Ing. LIBOR POLČÁK, Ph.D.
SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

162588

Ústav: Ústav informačních systémů (UIFS)
Student: **Strigáč Aurel**
Program: Informační technologie
Název: **Analýza skriptů pro zaznamenávání sezení**
Kategorie: Web
Akademický rok: 2024/25

Zadání:

1. Nastudujte problematiku skriptů zaznamenávajících sezení ve webovém prohlížeči. Seznamte se s problematikou ochrany soukromí na webu a dřívějšími studiemi zaměřenými na skripty pro zaznamenávání sezení.
2. Zjistěte možnosti bezplatného vyzkoušení služeb jako jsou FullStory, UserReplay, SessionCam, Hotjar, Yandex Metrika, Smartlook aj., případně najděte vzorové stránky obsahující tyto skripty. Prostudujte a porovnejte dokumentaci a uživatelské návody poskytované těmito službami.
3. Analyzujte činnost skriptů nalezených v bodě 1. Zaměřte se např. na typy obsluhy událostí, které skripty využívají a možnosti jejich detekce. Zaměřte se na problémy detekované dřívějšími studiemi a validujte jejich přítomnost. Popište změny, které se od těchto studií udály. Zamyslete se nad různými typy stránek a jejich obsahem a problémy, které by skripty pro zaznamenávání sezení mohly způsobovat. Navrhněte a implementujte nástroje, které vám v analýze pomohou. Své pokroky průběžně diskutujte s vedoucím práce.
4. Pokuste se nalézt stránky využívající poskytovatele služeb z bodu 1. Analyzujte nasazení těchto služeb a jejich individuální nastavení. Např. pro skripty HotJar zmapujte využívání atributů data-hj-allow a data-hj-suppress.
5. Shrňte dosažené výsledky a navrhněte další pokračování práce.

Literatura:

- Acar, G., Englehardt, S., and Narayanan, A. (2020). No boundaries: data exfiltration by third parties embedded on web pages. Proceedings on Privacy Enhancing Technologies, 2020, s. 220–238
- Grodzinsky, F. S., Miller, K. W., and Wolf, M. J. (2022). Session replay scripts: A privacy analysis. The Information Society, 38(4), s. 257–268.
- Polčák, L., Slezáková, A. (2023). Data Exfiltration by Hotjar Revisited. International Conference on Web Information Systems and Technologies (WEBIST). Řím: SciTePress - Science and Technology Publications, 2023, s. 347-354. ISBN 978-989-758-672-9.

Při obhajobě semestrální části projektu je požadováno:
Plně vypracovaný bod 1, zjištění možností bezplatného užívání služeb v bodě 2, katalogizovaná dokumentace těchto služeb. Bod 3 hotový pro nejméně dva poskytovatele služeb. Výsledky popište v technické zprávě.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Polčák Libor, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2024
Termín pro odevzdání: 14.5.2025
Datum schválení: 22.10.2024

Abstrakt

Skripty pre zaznamenávanie sedenia poskytujú prevádzkovateľom webových stránok vhľad do toho, ako používatelia interagujú s ich webovými stránkami. Ich použitie je avšak spojené s rizikom nechceného zachytenia osobných údajov. Táto práca analyzuje aktuálne správanie, redakčné mechanizmy a možnosti modifikácie skriptov pre zaznamenávanie sedenia od poskytovateľov Hotjar, Fullstory, Smartlook, Yandex Metrica, Mouseflow a PostHog. Dynamická analýza, vykonaná v kontrolovanom prostredí na vlastných testovacích stránkach, odhalila zmeny v predvolenom maskovaní dát, kde skript od poskytovateľa Hotjar, spolu so skriptom poskytovateľa Smartlook, teraz predvolene rediguje aj vybrané typy zobrazovaného obsahu, rozširujúc tak pôvodné zameranie na vstupné polia. Výsledky statickej analýzy skriptov potvrdili možnosť úplne obísť tieto predvolené redakčné mechanizmy cielenými úpravami kódu zaznamenávacieho skriptu. Práca ďalej skúmala implementáciu reakcie na signály DNT/GPC a analyzovala nasadenie samotných skriptov na reálnych, verejne dostupných, webových stránkach.

Abstract

Session recording scripts provide website operators with insights into how users interact with their websites. However, their use is associated with the risk of unintentionally capturing personal data. This thesis analyzes the current behavior, redaction mechanisms, and modification possibilities of session recording scripts from providers Hotjar, Fullstory, Smartlook, Yandex Metrica, Mouseflow, and PostHog. Dynamic analysis, conducted in a controlled environment on custom test pages, revealed changes in default data masking behaviour, where the script from provider Hotjar, along with the script from provider Smartlook, now also redacts selected types of displayed content by default, thereby expanding the original focus on input fields. The results of static script analysis confirmed the possibility of completely bypassing these default redaction mechanisms through targeted modifications to the recording script's code. The thesis further examined the implementation of responses to DNT/GPC signals and analyzed the deployment of the scripts on real, publicly available websites.

Kľúčové slová

Súkromie na webe, Ochrana osobných údajov, Skripty pre zaznamenávanie sedenia

Keywords

Web privacy, Personal data protection, Session replay scripts

Citácia

STRIGÁČ, Aurel. *Analýza skriptov pre zaznamenávanie sedenia.* Brno, 2025. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedúcí práce Ing. Libor Polčák, Ph.D.

Analýza skriptov pre zaznamenávanie sedenia

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Libora Polčáka, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

. . . . . .
Aurel Strigáč
14. mája 2025

Poďakovanie

Ďakujem vedúcemu mojej práce, Ing. Liborovi Polčákovi, Ph.D., za všetky jeho cenné rady a odporúčania v procese vypracovávania tejto práce.

Obsah

1 Úvod **3**

2 Zhrnutie doterajšieho stavu **4**
2.1 Identifikátor sedenia, používateľa a cookie 4
2.2 Princíp skriptov pre zaznamenávanie sedenia 5
2.3 Prehľad poskytovaných služieb 6
2.4 Množstvo zbieraných dát 9
2.5 Bezpečnostné riziká 10
2.6 Spôsoby ochrany užívateľa 12

3 Prehľad analyzovaných poskytovateľov služieb pre zaznamenávanie sedenia **15**
3.1 Proces výberu poskytovateľov 15
3.2 Dostupnosť bezplatnej verzie 16
3.3 Ponuka služieb 16
3.4 Nasadenie skriptu na sledovanie sedenia 18
3.5 Súlad s nariadeniami GDPR 26

4 Metodika **28**
4.1 Popis testovacích scenárov 28
4.2 Proces analýzy a úpravy zaznamenávacieho skriptu 31
4.3 Proces analýza nasadenia skriptov na reálnych webových stránkach 32

5 Výsledky **33**
5.1 Dynamická analýza 33
5.2 Statická analýza 43
5.3 Analýza nasadenia skriptov na reálnych webových stránkach 56

6 Záver **66**

Literatúra **68**

Zoznam obrázkov

2.1 Ukážka agregovaných sedení užívateľa (Fullstory) 5
2.2 Teplotné mapy 7
2.3 Prehrávanie sedenia 8
2.4 Konverzný lievik 9
2.5 Ukážka nepriameho úniku dát 11

3.1 Ukážka snippetov 20
3.2 Diagram procesu zaznamenávania sedenia používateľa 21
3.3 Ukážka redigovanej nahrávky sedenia v prehrávacom paneli (Posthog) . . . 23

4.1 Ukážky testovacej stránky 29
4.2 Prihlasovací formulár 30
4.3 Formulár s poštovnými údajmi 30

5.1 Maskovanie zobrazovaných čísiel (Hotjar) 36
5.2 Ukážka východzích nastavení redakcie zaznamenávacieho skriptu (Hotjar) . 37
5.3 Funkcionalita Form Privacy (Fullstory) 39
5.4 Funkcionalita Element data capture rule (Fullstory) 39
5.5 Ukážka zachytenia osobných informácií upraveným zaznamenávacím skriptom (Mouseflow) 53

Kapitola 1

Úvod

Skripty pre zaznamenávanie sedenia (session replay scripts) sú nástroje tretích strán, ktoré umožňujú prevádzkovateľom webových stránok sledovať a rekonštruovať interakcie návštevníkov s ich stránkami. Tieto skripty zaznamenávajú dáta o aktivitách používateľa, ako sú napríklad pohyby myšou, kliknutia, posúvanie stránky a vypĺňanie formulárov, ktoré sú následne spracované a prevedené do rôznych vizuálnych foriem, umožňujúcich analýzu správania používateľov. Medzi hlavné služby, ktoré tieto skripty ponúkajú, patria prehrávanie jednotlivých používateľských sedení, vizualizácia používateľského správania prostredníctvom teplotných máp a často aj ďalšie doplnkové funkcie ako napríklad zber spätnej väzby používateľov skrz dotazníky.

Aby mohli tieto služby poskytovať presnú rekonštrukciu sedenia používateľa, skripty pre zaznamenávanie sedenia musia zaznamenávať rozsiahle množstvo dát. To zahŕňa nielen sledovanie spomínaných akcií používateľa, ale aj zaznamenávanie samotného obsahu a štruktúry webovej stránky, vrátane jej dynamických zmien a rôznych metadát. Takýto rozsiahly zber dát so sebou prináša riziko zachytenia osobných alebo inak citlivých údajov používateľov. Tieto údaje môžu byť zachytené buď priamo, napríklad zadaním do vstupných polí stránky, alebo nepriamo, odvodením z iných zaznamenaných aktivít. Problematika zachytenia osobných údajov týmito skriptami bola už v minulosti predmetom viacerých analýz a výskumov, a táto práca na získané poznatky nadväzuje.

Táto práca je štruktúrovaná nasledovne: Kapitola 2 sa venuje zhrnutiu doterajšieho stavu v oblasti zaznamenávania sedení, poskytovaným službám, množstvu zbieraných dát a bezpečnostným rizikám spojeným s ich nasadením, vrátane známych spôsobov zachytenia osobných informácií a možností ochrany používateľa pred takýmto sledovaním. Kapitola 3 predstavuje prehľad analyzovaných poskytovateľov služieb pre zaznamenávanie sedenia, ich ponuku služieb, proces nasadenia ich skriptov na webovú stránku, dostupné redakčné nástroje a ich prístup k nariadeniam GDPR. V kapitole 4 je popísaná metodika použitá pri analýze, vrátane popisu testovacích scenárov, konfigurácií testov a procesu dynamickej aj statickej analýzy zdrojového kódu skriptov, ako aj analýzy ich nasadenia na reálnych webových stránkach. Nasledujúca kapitola 5 prezentuje výsledky a zistenia získané prostredníctvom týchto analýz pre jednotlivých poskytovateľov a webové stránky.

Kapitola 2

Zhrnutie doterajšieho stavu

Sedenie na webovej stránke predstavuje sekvenciu po sebe nasledujúcich interakcií vykonaných jedným návštevníkom v rámci definovaného časového úseku. Tieto interakcie môžu zahŕňať rôzne aktivity, ako napríklad prehliadanie jednotlivých podstránok, vyplnenie formulára alebo pridanie produktov do nákupného košíka. Sedenie teda predstavuje časový interval a zahŕňa všetky aktivity návštevníka od jeho prvého príchodu na webovú lokalitu až po jej opustenie alebo ukončenie aktivity.

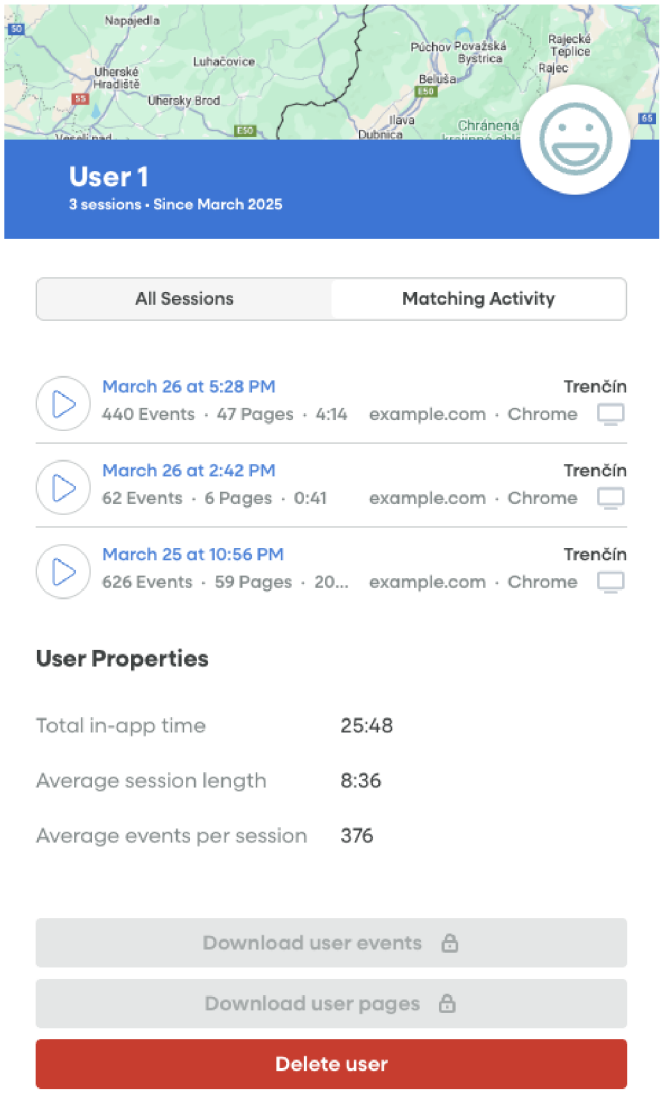
2.1 Identifikátor sedenia, používateľa a cookie

Na účely sledovania každého sedenia sa využíva unikátny identifikátor, známy ako ID sedenia alebo token sedenia. Tento identifikátor generuje webový server pri iniciácii sedenia a priraďuje ho konkrétnemu klientovi (prehliadaču). ID sedenia tak umožňuje serveru identifikovať aktivitu používateľa počas konkrétneho sedenia. Z bezpečnostných dôvodov sa tieto identifikátory generujú ako náhodné (pseudonáhodné) reťazce, aby sa znížila pravdepodobnosť ich uhádnutia pomocou útoku hrubou silou (brute-force attack) [12]. Identifikátory sedení sú kľúčové pre udržiavanie stavu používateľa medzi jednotlivými HTTP požiadavkami, čo je nevyhnutné pre funkcie ako zachovanie prihlásenia, správa položiek v online nákupnom košíku alebo vypĺňanie viacstranových formulárov.

Na rozdiel od ID sedenia, ktoré identifikuje iba jedno konkrétne sedenie, sa ID používateľa[1] (User ID) používa na rozpoznanie a prepojenie sedení uskutočnených z jedného konkrétneho profilu prehliadača naprieč viacerými, aj časovo vzdialenými, sedeniami. Hoci sa často používa na sledovanie jedného používateľa, čo implikuje aj názov tohto identifikátoru, technicky identifikuje unikátny profil prehliadača, nie samotnú osobu. Tento trvalejší, taktiež náhodne generovaný, identifikátor sa zvyčajne priradí konkrétnemu profilu v prehliadači používateľa po jeho prvej návšteve webovej stránky, a následne ho skripty pre zaznamenávanie sedenia využívajú na agregáciu všetkých sedení z daného profilu prehliadača [22]. To znamená, že ak ten istý používateľ navštívi stránku z iného profilu prehliadača, bude mu pridelené nové ID používateľa. Ukážku využitia ID používateľa pre agregáciu sedení zobrazuje obrázok 2.1. Navyše, môžu nastať aj situácie, keď jeden profil prehliadača (a teda jedno ID používateľa) je zdieľaný viacerými fyzickými osobami, napríklad v rámci jednej domácnosti používajúcej spoločný počítačový účet, čím sa ďalej komplikuje priradenie zachytených sedení konkrétnej osobe.

[1] ID používateľa môžete nájsť niekedy aj pod názvom ID klienta (ClientID) avšak funkcionalitou sú tieto identifikátory identické

Všetky tieto identifikátory sa často ukladajú do cookies [45] v prehliadači používateľa. Cookies sa odosielajú serveru pri každej HTTP požiadavke na doménu, ktorá ich pôvodne vytvorila. Niektoré cookies sú nastavené tak, aby vypršali (boli vymazané) okamžite po zatvorení prehliadača. Na druhej strane, trvalé cookies trvajú aj po skončení aktuálneho sedenia používateľa. Príkladom je situácia, keď používateľ na webovej stránke vyberie možnosť „Zapamätať si ma“. Trvalé cookies sa zvyčajne ukladajú na pevný disk používateľa na miesto podľa konkrétneho operačného systému a prehliadača [12]. Používajú sa na rozpoznanie používateľov pri opakovaných návštevách danej stránky, prípadne aj na ich identifikáciu medzi viacerými webovými lokalitami [43]. ID používateľa sa často ukladá práve do týchto trvalých cookies.



Obr. 2.1: Ukážka agregovaných sedení užívateľa (Fullstory)

2.2 Princíp skriptov pre zaznamenávanie sedenia

V prostredí moderného webového vývoja hrajú skripty pre zaznamenávanie sedenia, alebo session replay scripts, kľúčovú úlohu v porozumení používateľskej skúsenosti (UX). Tieto

skripty tretích strán umožňujú prevádzkovateľom webových stránok podrobne sledovať interakcie používateľov s ich stránkou [19]. Vďaka tomuto sledovaniu môžu prevádzkovatelia webových stránok zlepšovať používateľský zážitok, urýchľovať identifikáciu a riešenie chýb na stránke alebo optimalizovať mieru konverzie. Príkladom môže byť tvrdenie „Spoločnosť Jacaranda Finance zvyšuje mieru konverzie o 20 percent“, ktoré uvádza spoločnosť Fullstory na svojich stránkach [25].

Existuje niekoľko spôsobov, ako tretia strana môže sledovať jednotlivé akcie užívateľa. Jedným z prvých, zdanlivo jednoduchším, je periodické zachytávanie snímok obrazovky používateľa. Tento prístup zvolila napríklad spoločnosť Microsoft pri implementácii ich nástroja Recall [44], ktorý používateľovi umožňuje rýchle nájdenie a návrat k obsahu, ktorý v minulosti videl na počítači. Skripty pre zaznamenávanie sedenia však žiadny videozáznam nevyužívajú. Namiesto toho sú záznamy, ktoré poskytujú, presnou rekonštrukciou toho, čo používateľ skutočne videl, počul a s čím na webovej stránke interagoval. Tieto informácie skripty získavajú zaznamenávaním štruktúry stránky, všetkých jej zmien a používateľských interakcií s prvkami stránky [48]. Skripty pre zaznamenávanie sedenia zbierajú tieto dáta počas aktuálneho sedenia na stránke a po jeho ukončení (ukončenie zo strany užívateľa, poprípade vypršanie časového limitu) ich odosielajú na stranu servera poskytovateľa služby. Tam sú dáta agregované do štatistík [56], ktoré sú následne sprístupnené prevádzkovateľom webových stránok prostredníctvom prehrávacieho panela poskytovateľa služby. Tieto štatistiky môžu mať rôznu podobu [25] a poskytujú detailný prehľad o správaní používateľov.

2.3 Prehľad poskytovaných služieb

Poskytovatelia služieb pre analýzu používateľského správania spravidla ponúkajú širokú škálu funkcií [33, 25], pričom každá z nich ponúka prevádzkovateľovi webovej lokality jedinečný vhľad do správania používateľov.

2.3.1 Teplotné mapy

Teplotné mapy (Heatmaps) predstavujú vizualizáciu interakcií používateľov s webovou stránkou. Na znázornenie frekventovanosti využívaných oblastí používajú farebnú škálu (často od chladných po teplé odtiene, pričom čím je farba „teplejšia“, tým viac interakcií sa tam vyskytlo). Teplotná mapa tak poskytuje rýchlu a intuitívnu vizuálnu reprezentáciu agregovaných dát [38] bez potreby zobrazenia číselných hodnôt, čo uľahčuje analýzu a pochopenie spôsobov, akými používatelia s danou stránkou pracujú.

Existuje niekoľko základných typov teplotných máp [21]. Nasledujúci prehľad zahŕňa tie najbežnejšie, ktoré ponúka väčšina poskytovateľov služieb pre analýzu správania:

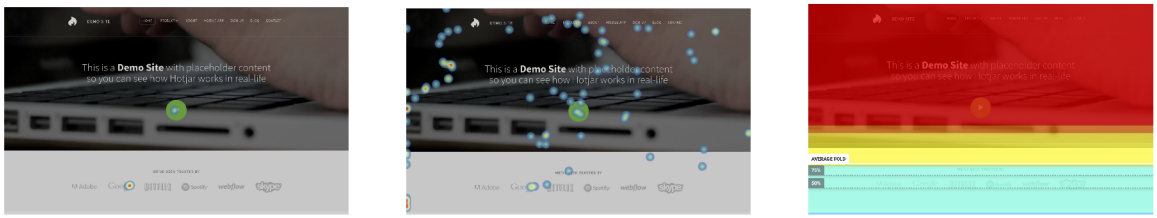
- **Mapy kliknutí (Click Maps)** – Vizualizujú agregované údaje o tom, kam používatelia na stránke klikajú. Pomáhajú identifikovať najviac a najmenej klikané prvky, čo môže poukázať na záujem používateľov alebo odhaliť problémy v navigácii. Teplotné mapy kliknutí sa generujú na základe zaznamenaných súradníc kliknutia, cieľového prvku, rozlíšenia obrazovky zariadenia a obsahu webovej stránky [19]. Filtráciou dát, ktoré poskytujú teplotné mapy klikaní, je možné vytvárať špecifické teplotné mapy zamerané na konkrétne problémy. Príkladom sú situácie, keď používateľ kliká na nekliknuteľné časti stránky (mŕtve kliky).

- **Mapy posúvania (Scroll Maps)** – Ponúkajú vizuálne znázornenie toho, ako hlboko sa používatelia na stránke dostanú. Zvyčajne farebne odlišujú oblasti stránky

podľa percenta používateľov, ktorí danú časť ešte videli [21]. Pomáhajú tak určiť, či kľúčový obsah nie je umiestnený príliš nízko, alebo kde používatelia strácajú záujem. Teplotné mapy pozornosti sú odvodené od máp posúvania a vyzerajú podobne, ale namiesto posúvania zobrazujú čas strávený v jednotlivých sekciách stránky.

- **Mapy pohybu myši (Mouse Movement Maps)** – Sledujú a vizualizujú všeobecný pohyb kurzora myši po stránke. Ukazujú oblasti, kde sa používatelia kurzorom pohybujú najčastejšie, kde sa zastavujú alebo váhajú. Keďže výskum naznačuje koreláciu medzi pohybom očí a pohybom kurzora myši [34], tieto mapy môžu poskytnúť nepriame informácie o tom, kam sa používateľ v danom čase približne pozeral.

Okrem uvedených typov teplotných máp takmer každý poskytovateľ ponúka aj svoje jedinečné varianty, vďaka ktorým sa odlišujú od konkurencie [21].



(a) Kliknutia (b) Pohyby myši (c) Posúvania

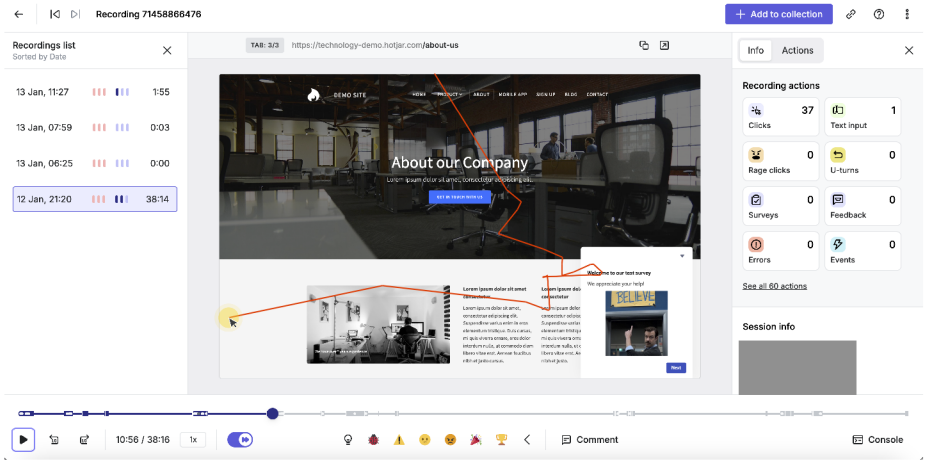
Obr. 2.2: Teplotné mapy[1]

2.3.2 Prehrávanie sedenia

Prehrávanie sedenia (Session replay) umožňuje prevádzkovateľom webových stránok prehrávať interakciu používateľov so stránkou ako videozáznam. Analýza týchto záznamov poskytuje pohľad na individuálne správanie používateľov. Prevádzkovatelia stránok tak môžu napríklad identifikovať sedenia, počas ktorých používatelia boli zmätení alebo prejavovali známky frustrácie, čo umožňuje cielené vylepšenia webovej stránky [56].

Prínosom prehrávania sedení je tiež zjednodušenie a urýchlenie procesu reprodukcie a odstraňovania technických chýb na webovej stránke [24]. Toto je možné dosiahnuť vďaka prepojeniu záznamov sedení používateľov s nástrojmi na nahlasovanie a správu chýb. Podobne môžu tieto záznamy zefektívniť prácu zákazníckej podpory. Poskytnutím záznamu relevantného sedenia získavajú pracovníci podpory kontext k problému nahlásenému používateľom, čo im umožňuje efektívnejšie poskytnúť pomoc.

[1] Obrázky boli získané z prehrávacieho panelu referenčnej stránky Hotjar



Obr. 2.3: Prehrávanie sedenia[1]

2.3.3 Konverzný lievik

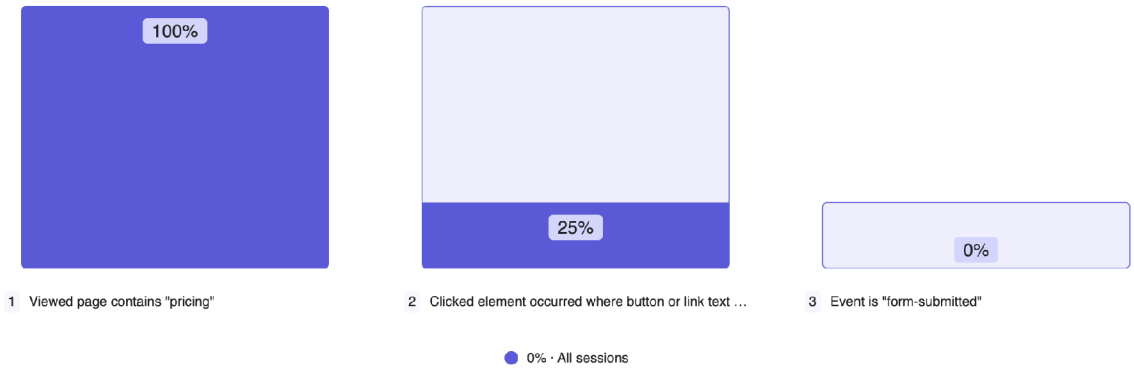
Konverzný lievik (Conversion funnel) predstavuje cestu, ktorou používateľ prechádza na webovej stránke smerom k dosiahnutiu konkrétneho cieľa, ako je napríklad dokončenie nákupu, odoslanie formulára alebo registrácia. Táto cesta je typicky rozdelená do viacerých fáz (krokov) a postupne v každom nasledujúcom kroku počet používateľov prirodzene klesá. Optimalizácia konverzného lievika teda spočíva v identifikácii a následnom znižovaní počtu dôvodov, prečo používatelia v konkrétnej fáze web opúšťajú [61].

Dôležitou metrikou, ktorá pomáha vyhodnotiť efektivitu konverzného lievika, je miera konverzie (Conversion rate). Táto metrika udáva percento návštevníkov, ktorí v sledovanom období úspešne dokončili požadovanú akciu (konverziu), v pomere k celkovému počtu návštevníkov [10]. Definícia konkrétnej konverznej akcie už závisí od obchodných cieľov danej webovej stránky (napr. nákup alebo kliknutie na reklamu).

S rozvojom internetu a firemných webových stránok sa vyvíjali aj ich ciele a metriky úspešnosti. V počiatočných fázach sa dôraz kladol hlavne na šírenie informácií o stránke a úspech sa často meral počtom unikátnych návštevníkov [9]. Súčasťou týchto snáh bola aj optimalizácia internetových prehliadačov (SEO) na zvýšenie dosahu [4]. Ako sa však online trh vyvíjal a webové stránky začali plniť úlohu hodnotného predajného kanála, zameranie sa presunulo z čistej návštevnosti na „premenu“ návštevníkov na zákazníkov, teda na zvyšovanie miery konverzie [11, 36].

Detailné sledovanie správania používateľov, ktoré skripty pre zaznamenávanie sedenia umožňujú, napomáha pri identifikácii vzorcov vedúcich ku konverzii alebo naopak k opusteniu stránky. Získané dáta umožňujú aplikovať stratégie na optimalizáciu jednotlivých fáz konverzného lievika a tým zvyšovať mieru konverzie.

[1]Obrázok bol získaný z prehrávacieho panelu referenčnej stránky Hotjar



Obr. 2.4: Konverzný lievik[1]

2.3.4 Dodatočné služby

Okrem hlavných služieb, ako sú teplotné mapy alebo prehrávanie sedení, poskytovatelia ponúkajú aj ďalšie dodatočné služby. Medzi tieto služby patria napríklad nástroje pre zber spätnej väzby prostredníctvom krátkych prieskumov integrovaných priamo do stránky. Ďalšou službou je analýza formulárov, ktorá umožňuje analyzovať (čas vyplnenia, poradie, miera nedokončenia) a zlepšovať výkonnosť formulárov používaných na webovej stránke. Niektoré platformy tiež ponúkajú funkcie, ako je sledovanie používateľských ciest pre lepšie pochopenie navigačných vzorcov, prípadne nástroj pre A/B testovanie. Ten umožňuje porovnať dve verzie aplikácie alebo webovej stránky s cieľom identifikovať tú lepšiu [27].

2.4 Množstvo zbieraných dát

Pre správnu rekonštrukciu používateľského sedenia je nevyhnutné zachytiť nielen interakcie používateľa, ale aj vizuálnu podobu webovej stránky a jej zmeny v čase. Pre získanie štruktúry a obsahu stránky, ktorú používateľ navštívil, skripty využívajú prístup k DOM, alebo Objektového Modelu Dokumentu. DOM je štandardizované programovacie rozhranie (API) pre HTML [72] a XML [68] dokumenty, vyvinuté a vydané World Wide Web Consortium (W3C) [71]. Jeho hlavným účelom je uľahčiť programátorom prístup k prvkom webovej stránky, aby mohli dynamicky pridávať, odstraňovať a upravovať obsah (všetky texty a vstupné polia, vrátane užívateľom zadávaných hodnôt), atribúty a štýly stránky. DOM funguje ako platformovo nezávislá a jazykovo neutrálna reprezentácia dokumentu, s ktorou interagujú skriptovacie jazyky ako JavaScript [51]. V praxi sa často hovorí o „stromovej štruktúre DOM“, kde každý uzol (prvok) stránky je nezávislým, vyberateľným objektom. Skripty pre zaznamenávanie sedenia môžu získať počiatočný stav stránky napríklad serializáciou (typicky do formátu JSON) celej DOM štruktúry a následne monitorovať jej dynamické zmeny (mutácie) počas sedenia, napríklad prechádzaním jednotlivých uzlov a sledovaním ich vlastností [1].

Popri štruktúre stránky je rovnako potrebné zaznamenávať aj priame interakcie používateľa. Medzi sledované akcie patria pohyby kurzora myši, kliknutia, posúvanie (rolovanie) stránky, stláčanie klávesov či navigácia medzi jednotlivými stránkami alebo sekciami [24]. Na monitorovanie týchto interakcií používateľa so stránkou skripty používajú odposluchy

[1] Obrázok bol získaný z prehrávacieho panelu referenčnej stránky Hotjar

udalostí (event listeners) pre akcie myši a klávesnice. Skripty dokážu sledovať aj interakcie s formulárovými prvkami ako `<input>`, `<textarea>` alebo `<select>`, pričom zachytávajú udalosti ako zmeny zadaných hodnôt, aktiváciu alebo opustenie poľa [1].

Skripty pre zaznamenávanie sedenia zvyčajne zhromažďujú aj metadáta o prostredí používateľa. Tieto dáta typicky zahŕňajú informácie ako názov a verzia použitého webového prehliadača, názov operačného systému, rozlíšenie obrazovky alebo typ zariadenia (napríklad stolný počítač alebo mobil) [48].

2.5 Bezpečnostné riziká

Aby mohli skripty tretích strán poskytovať služby ako napríklad práve prehrávanie sedení, musia zaznamenávať rozsiahle spektrum informácií o interakciách používateľa a obsahu stránky z jeho zariadenia [1, 28]. Tento proces, ak nekontrolovaný, predstavuje riziko. Môže viesť k neúmyselnému zachyteniu a prenosu dát, ktoré môžu ohroziť súkromie, a dokonca aj bezpečie užívateľa.

Príkladom tohto rizika je výskum z roku 2017 [13], ktorý na farmaceutickej stránke preukázal a zdokumentoval únik citlivých informácií užívateľov, vrátane zdravotného stavu a lekárskych predpisov. Tieto lekárske predpisy navyše mohli byť priamo priradené ku jednotlivým užívateľom, vďaka súvislému úniku mien užívateľov prameniacemu z tej istej stránky. Tretím stranám tak boli sprístupnené detailné informácie o zdravotnom stave návštevníkov webu skrz informácie o liekoch, ktoré jednotliví užívatelia užívajú.

Ďalšie riziko nastáva počas používania ID používateľa, ktoré umožňuje pripojiť viacero sedení jedného návštevníka z konkrétneho prehliadača k jeho profilu v prehrávacom paneli poskytovateľa. Ak počas niektorého z týchto sedení dôjde k úniku osobných alebo citlivých informácií, potenciálne to umožňuje deanonymizáciu používateľa. To by umožnilo rekonštruovať kompletný obraz o dlhodobej aktivite používateľa naprieč všetkými jeho zaznamenanými sedeniami (aj tými bez úniku), ktoré sú viazané na dané ID. Prepojené sedenia, obohatené o zaznamenané metadáta ako sú lokalita pôvodu jednotlivých sedení (za predpokladu, že používateľ nepoužíva VPN alebo iné anonymizačné techniky) ako je vidno v obrázku 2.1, by mohli potenciálne umožniť aj sofistikovanejšie sledovanie, napríklad rekonštrukcie geografickej polohy a pohybu používateľa v čase.

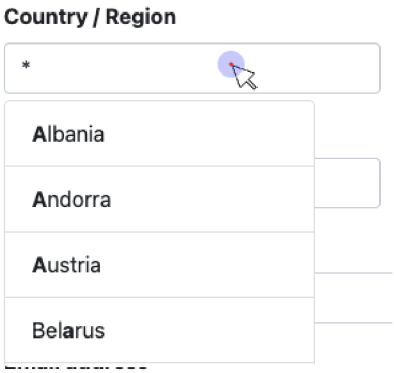
Je dôležité uviesť, že mnohí poskytovatelia analytických služieb v skutočnosti vyjadrujú svôj postoj k danej tématike. Vo svojich podmienkach používania výslovne zakazujú zhromažďovanie citlivých informácií o používateľoch prostredníctvom svojich služieb a ponúkajú nástroje, a odporúčania, na ich maskovanie či vylúčenie zo záznamov [28]. Mojím cieľom nie je posudzovať, nebodaj spochybňovať zámery spoločností tretích strán, ktoré sa zaoberajú zaznamenávaním sedení. Ani na to nemám dôvod, keďže sa zatiaľ nenašli žiadne dôkazy, ktoré by naznačovali, že získané údaje sa v niektorej z analyzovaných spoločností vedome zneužívali na aktivity ako personalizácia reklamy alebo sledovanie medzi stránkami [1].

2.5.1 Známe spôsoby úniku informácií

Skripty tretích strán, vložené priamo do hlavného kontextu webovej stránky, majú v podstate prístup k rovnakému obsahu DOM, aký sa zobrazuje používateľovi [1]. Keďže DOM však často obsahuje citlivé informácie, ako sú napríklad údaje o platobných kartách, adresy pobytu, zdravotné údaje či heslá, tento obsah by mal byť prostredníctvom redakčných (maskovacích) nástrojov zo záznamov vylúčený. Táto ochrana však nemusí byť vždy účinná, nakoľko nie je zaručené správne pokrytie všetkých citlivých polí týmito nástrojmi, čo môže

následne viesť k priamemu úniku informácií. Zneužitie nástrojov ako je v prípade Hotjar atribút `data-hj-allow`, pomocou ktorého môžeme zámerne zaznamenať obsah pôvodne maskovaného obsahu polí, môže taktiež viesť k takémuto úniku informácií.

K úniku môže dôjsť aj nepriamou cestou, odvodením pôvodnej informácie z iných, nemaskovaných dát. Príkladom sú formuláre s výberom z viacerých možností. Aj keď text samotných možností môže byť zo záznamu vylúčený, presné sledovanie pohybu a kliknutí myši používateľa môže prezradiť, ktorú konkrétnu možnosť označil [1]. Podobne, hoci samotný text zadaný do vyhľadávacieho poľa môže byť maskovaný, zaznamenané výsledky vyhľadávania, ktoré sa zobrazia na stránke a sú súčasťou záznamu, často umožňujú s vysokou pravdepodobnosťou spätne určiť, čo sa užívateľ snažil vyhľadať [56].



Obr. 2.5: Ukážka nepriameho úniku dát

Nemaskované heslá môžu uniknúť najmä v súvislosti s bežnou funkcionalitou „Ukáž Heslo“. Typická implementácia tejto funkcie spočíva v použití dvoch prekrývajúcich sa vstupných polí – jedno typu `password` (ktorého obsah je štandardne maskovaný a skriptami ignorovaný) a druhé typu `text` (obsahujúce rovnaké heslo). Pri použití funkcionality „Ukáž Heslo“ sa pole typu `text` dočasne zobrazí, čím sa jeho obsah (čitateľné heslo) stáva viditeľným nielen pre používateľa, ale aj pre skript zaznamenávajúci sedenie [1, 56, 13]. Šanca úniku hesla týmto spôsobom sa môže ďalej zvýšiť používaním nástrojov alebo rozšírení prehliadača, ktoré automaticky menia typ poľa na text s cieľom uľahčiť používateľovi proces zadávania hesla (napr. nástroj Unmask Password [53]). Okrem zmeny typu vstupných polí existujú aj iné, menej bežné spôsoby úniku hesiel, ktoré však, vzhľadom na citlivosť uniknutej informácie, sú po odhalení rýchlo opravované [62].

2.5.2 Spôsob implementácie prehrávacieho panelu

Bezpečnosť prenosu zaznamenaných dát sedenia je jedným z aspektov správnej implementácie prehrávacieho panelu. Použitie nešifrovaného protokolu HTTP [18], protokol aplikačnej vrstvy referenčného modelu ISO/OSI [35], na tento účel predstavuje významné bezpečnostné riziko, nakoľko prenášané dáta môžu byť ľahko odchytené a prečítané treťou stranou v rámci útoku typu Man-in-the-Middle [54].

Tento útok na protokol HTTP bol využiteľný v bývalej implementácii prehrávacieho panelu služby Hotjar [33], kedy prehrávací panel prenášal dáta sedení prostredníctvom protokolu HTTP, a to aj v prípadoch, keď pôvodné nahrávanie prebiehalo na stránkach zabezpečených pomocou protokolu HTTPS (HTTP cez TLS, zabezpečená verzia protokolu HTTP) [13]. Momentálna implementácia prehrávacieho panelu spoločnosti Hotjar už av-

šak používa HTTPS a aby sa ešte viac zabránilo možnosti odposluchu komunikácie treťou stranou, skript od spoločnosti Hotjar nie je možné používať na webových stránkach, ktoré nepodporujú HTTPS [56]. Tento posun k zabezpečenej komunikácii je v súlade so všeobecným trendom na webe, kde podľa dostupných informácií využíva HTTPS už približne 86% všetkých webových stránok [70].

2.6 Spôsoby ochrany užívateľa

Blokovanie zo strany užívateľa často predstavuje hľadanie rovnováhy medzi užívateľským zážitkom a ochranou súkromia používateľa. Zatiaľ čo príliš striktné alebo agresívne blokovanie môže narušiť alebo úplne znemožniť fungovanie častí alebo aj celých webových stránok, základné alebo benevolentnejšie nastavenie ochrany nemusí poskytovať dostatočnú úroveň ochrany súkromia podľa predstáv používateľa [62]. Napriek tomu predstavuje používanie nástrojov na blokovanie sledovačov stále významnú a relatívne efektívnu stratégiu na zvýšenie ochrany online súkromia. Navyše, ako ukázal výskum, môže takéto blokovanie viesť aj k zlepšeniu výkonnosti internetových prehliadačov, napríklad vďaka zníženiu množstva prenášaných dát [40].

2.6.1 Mechanizmy ochrany pred sledovaním v internetových prehliadačoch

Ochrana súkromia a blokovanie neželaného sledovania sa stali súčasťou väčšiny moderných internetových prehliadačov. Tieto aplikácie dnes štandardne obsahujú funkcie a nastavenia zamerané na posilnenie súkromia používateľov [55].

Ako príklad spomeniem prehliadač Safari od spoločnosti Apple [3], patriaci medzi najpoužívanejšie prehliadače [67]. Safari používa niekoľko mechanizmov ochrany súkromia známych pod názvom Intelligent Tracking Prevention (ITP). Patria medzi ne funkcie ako blokovanie cookies tretích strán na sledovanie medzi stránkami, poskytovanie prehľadov o zablokovaných pokusoch o sledovanie a v určitých prípadoch aj vyžadovanie explicitného súhlasu používateľa so sledovacími aktivitami aplikácií a webov. Podobné služby a funkcie poskytujú aj mnohé ďalšie internetové prehliadače [55].

Moderné prehliadače navyše ponúkajú vstavanú podporu vynucovania použitia protokolu HTTPS, takže prístup na internetové stránky využívajúce nezabezpečený protokol HTTP môže byť obmedzený, blokovaný alebo aspoň označený ako nebezpečný.

2.6.2 Nástroje tretích strán na zabránenie sledovania

Internetové prehliadače obsahujú rôzne zabudované funkcie ochrany súkromia, avšak dodatočné zvýšenie ochrany si často vyžaduje využitie blokovačov sledovania tretích strán, ako je napríklad nástroj uBlock Origin [29]. Tieto nástroje majú zvyčajne podobu rozšírení prehliadača, ktoré rozlišujú medzi sledovacími a nesledujúcimi požiadavkami HTTP vďaka používaniu komunitou alebo organizáciami udržiavaných zoznamov adries URL (filter lists alebo blocklists) [6]. Tieto zoznamy obsahujú vzory adries URL alebo celé domény identifikované ako zdroje sledovacích skriptov alebo iného nežiaduceho obsahu. Aby bol požiadavok podľa zoznamu vyhodnotený ako sledovací, prvotne musí byť pridaný na zoznam sledovacích domén. Z tohto vyplýva jedna zo slabín blokovania na základe zoznamov adries URL, ktorou je, že úplne nové hrozby môžu určitý čas zostať neodhalené a tým pádom neblokované [56]. Existuje ešte niekoľko spôsobov, ako sa môžu sledovatelia vyhnúť detekcii.

Príkladom je zmena adresy URL skriptu sledovateľa tak, že zmenená adresa URL sledovateľa sa už nezhoduje so žiadnym predpisom v zozname blokovania [43, 56]. Tieto zoznamy adries URL sú často doplnené o filtre CSS [43].

2.6.3 Signalizovanie preferencií sledovania

Hlavička DNT (Do Not Track) [49] v požiadavke HTTP bola navrhnutá ako štandardizovaný mechanizmus, pomocou ktorého mohli používatelia signalizovať webovým serverom svoju preferenciu nebyť sledovaní. Zaslanie hodnoty `DNT: 1` malo indikovať serveru požiadavku na zastavenie sledovania. Hodnota `DNT: 0` signalizovala súhlas so sledovaním, zatiaľ čo absencia hlavičky alebo hodnota `null` znamenala nevyjadrenie preferencie. Tento mechanizmus je dnes považovaný za zastaraný a jeho úlohu má prebrať štandard Global Privacy Control (GPC) [66], signalizovaný prostredníctvom hlavičky `Sec-GPC`.

Hlavička DNT je vo veľkej miere webovými stránkami ignorovaná [5], hoci niektoré analytické nástroje, vrátane určitých skriptov pre zaznamenávanie sedenia, ju môžu stále čiastočne rešpektovať [56]. Zaujímavý rozmer však DNT nadobúda v kontexte európskeho nariadenia GDPR. Podľa GDPR [16] má subjekt údajov právo namietať proti spracúvaniu osobných údajov aj prostredníctvom automatizovaných prostriedkov. Signál DNT môže predstavovať práve takúto automatizovanú námietku. Ignorovanie tohto signálu (úmyselné či neúmyselné) by tak mohlo potenciálne viesť k porušeniu nariadení Európskej únie. Nedávne rozhodnutie Berlínskeho okresného súdu ešte potvrdilo dôležitosť rešpektovania DNT, keď zakázalo spoločnosti LinkedIn ignorovať signály DNT v kontexte GDPR [60].

Paradoxne, aj napriek tejto potenciálnej právnej relevancii, podpora pre zasielanie DNT signálu zo strany internetových prehliadačov klesá. Prehliadač Safari ho nepodporuje už dlhodobo a Firefox od neho upustil po verzii 134 [50]. Hlavným dôvodom je práve rozsiahle nerešpektovanie DNT signálu zo strany webových stránok.

	Chrome	**Edge**	**Firefox**	**Opera**	**Safari**
DNT	✓	✓	✗	✓	✗

Tabuľka 2.1: Podpora DNT internetovými prehliadačmi [49]

Štandard GPC sa od signálu DNT líši v niekoľkých aspektoch. Jeho zameranie je užšie: namiesto všeobecného sledovania signalizuje používateľov nesúhlas s predajom alebo zdieľaním osobných údajov [66]. Sémantika GPC je tiež rozdielna od signálu DNT – hodnota `Sec-GPC: 1` znamená nesúhlas, zatiaľ čo absencia hlavičky `Sec-GPC` značí chýbajúcu preferenciu alebo súhlas [52]. Širokoplošná implementácia signálu GPC v prehliadačoch je stále prebiehajúci proces.

	Chrome	**Edge**	**Firefox**	**Opera**	**Safari**
Sec-GPC	✗	✗	✓	✗	✗

Tabuľka 2.2: Podpora Sec-GPC internetovými prehliadačmi [52]

2.6.4 Stratégia ochrany používateľa pred sledovaním

Efektívna ochrana používateľa pred sledovaním si vyžaduje kombinovaný prístup, čo znamená použitie technických nástrojov spolu s dostatočným informovaním používateľa o tejto

problematike. Po technickej stránke to zahŕňa výber internetového prehliadača a jeho následné doplnenie o nástroje na blokovanie sledovania (ako sú rozšírenia tretích strán) alebo využívanie signalizačných mechanizmov (napr. DNT), aj napriek tomu, že ich rešpektovanie zo strany webov nie je zaručené [5, 56].

Po stránke informovanosti používateľa, kľúčová je obozretnosť pri zadávaní osobných alebo citlivých informácií na webových stránkach. Používatelia by si mali byť vedomí, že na webových stránkach môžu byť ich dáta spracúvané nielen prevádzkovateľom stránky, ale aj rôznymi integrovanými službami tretích strán, ktorých prítomnosť nemusí byť na prvý pohľad zrejmá [1]. Preto by mali užívatelia zvážiť používanie unikátnych a silných hesiel pre jednotlivé online služby alebo využívanie dočasných aliasov pre e-maily pri registráciách. Toto môže výrazne znížiť potenciálne škody v prípade úniku údajov z niektorej služby. Okrem tohto by si mal používateľ rozmyslieť, či bude súhlasiť s cookies, ktoré nie sú nevyhnutné (esenciálne) pre základnú funkčnosť stránky. Cookies, označované niekedy ako analytické alebo štatistické, môžu slúžiť na ukladanie identifikátorov, ako je napríklad spomínané ID používateľa [22], čím sa len prehlbuje miera sledovania.

Kapitola 3

Prehľad analyzovaných poskytovateľov služieb pre zaznamenávanie sedenia

Táto kapitola sa zameriava na predstavenie a porovnanie nasledujúcich šiestich poskytovateľov služieb zaznamenávania sedení, analyzovaných v tejto práci:

- Hotjar [33],
- Fullstory [25],
- Smartlook [65],
- Yandex Metrica [75],
- Mouseflow [46],
- PostHog [57].

Po úvodnom zdôvodnení výberu týchto konkrétnych platforiem nasleduje prehľad ich ponúkaných služieb, porovnanie doby uchovávania dát, analýza procesu nasadenia ich skriptov a dostupnosti bezplatných verzií. Kapitola taktiež rozoberá ponúkané redakčné nástroje na vynechanie alebo zahrnutie údajov v nahrávkach a nakoniec rozoberá súlad s nariadeniami GDPR.

3.1 Proces výberu poskytovateľov

Skripty na zaznamenávanie sedenia od spoločností Hotjar, Fullstory, Smartlook a Yandex patria k najobľúbenejším a najpoužívanejším vo svojom obore [1] a v minulosti už boli predmetom viacerých analýz [1, 19, 28]. Ich popularitu a vysokú mieru nasadenia posudzujem najmä na základe výskumu z minulosti [1]. V ňom sa tieto skripty umiestnili na najvyšších priečkach v nasadenosti na stránkach, nasledované skriptami spoločností ako sú SessionCam a UserReplay.

Aktuálna situácia na trhu sa však čiastočne zmenila. Dáta o nasadenosti analytických nástrojov na vzorke Top 1 milióna webových stránok (poskytované spoločnosťou Built-

With[1]) naznačujú pokles popularity nástrojov ako SessionCam a UserReplay [7]. Preto boli z mojej strany do výberu pre túto prácu zaradené iné, momentálne relevantné, alternatívy.

Konkrétne bol vybraný skript pre zaznamenávanie sedenia od dánskej spoločnosti Mouseflow, ktorý som zvolil aj vďaka jeho širokej ponuke analytických funkcií a pozitívnych referencií. Nasadený ho dokážete nájsť napríklad aj na oficiálnej stránke automobilovej spoločnosti Ford [20]. Šesticu dopĺňa skript od spoločnosti PostHog, ktorý ma zaujal vďaka svojmu dostupnému zdrojovému kódu (open-source, MIT licencia), možnosti self-hostingu (napr. cez Docker Compose[2]) a poskytovaním rozsiahlej bezplatnej verzie [57].

3.2 Dostupnosť bezplatnej verzie

Vzhľadom na charakter mojej práce, ktorej cieľom je analýza skriptov na zaznamenávanie sedení, ma najviac zaujímalo, či poskytovatelia analyzovaní v rámci tejto práce vôbec ponúkajú bezplatnú verziu ich služieb.

	Hotjar	**Fullstory**	**Smartlook**	**Yandex Metrica**	**Mouseflow**	**PostHog**
Bezplatná verzia	✓	✓	✓	✓	✓	✓
Obmedzenie (sedenia/mesiac)	1050	10000	3000	žiadne[3]	500	5000

Tabuľka 3.1: Porovnanie bezplatných verzií služieb

Pre potreby analýzy skriptov pre zaznamenávanie sedenia a testovania, či daný nástroj korektne maskuje rôzne typy údajov, sú tieto bezplatné verzie úplne postačujúce a nepredpokladám, že by limity uvedené v tabuľke 3.1 ovplyvnili moju prácu negatívnym spôsobom.

Okrem toho, viacerí poskytovatelia, ktorí ponúkajú aj platené verzie svojich produktov, umožňujú vyskúšať plnú verziu bezplatne, aj keď len na časovo obmedzenú skúšobnú dobu. Príkladom je Hotjar, ktorý takéto rozšírenie ponúka počas prvých 15 dní využívania bezplatnej verzie.

3.3 Ponuka služieb

Ponuka analytických nástrojov sa medzi jednotlivými analyzovanými poskytovateľmi líši. Základné funkcie, ako je prehrávanie sedení, teplotné mapy či konverzné lieviky, sú síce spoločné pre väčšinu poskytovateľov, avšak poskytovanie nástrojov, ako je analýza formulárov, používateľské prieskumy, sledovanie používateľských ciest či A/B testovanie, sa už u jednotlivých poskytovateľov rozchádza. Detailný prehľad ponúkaných služieb poskytuje tabuľka 3.2.

Pri skúmaní aktuálnej ponuky som si nemohol nevšimnúť rastúci vplyv trendov v oblasti umelej inteligencie (AI). Niektorí poskytovatelia skriptov pre zaznamenávanie sedenia už začali integrovať technológie AI aj do svojich analytických nástrojov. Príkladom je funkcia

[1] Pôvodný výskum [1] využíval pre hodnotenie najpoužívanejších skriptov rebríček Alexa. Keďže služba Alexa bola v roku 2022 ukončená, pre aktuálnejšie posúdenie som použil dáta zo služby BuiltWith [7].

[2] https://github.com/PostHog/posthog

[3] Yandex Metrika pri ich službe Session Replay 1.0 a 2.0 nemá verejne známe žiadne obmedzenie zo strany počtu zaznamenatelných sedení, avšak sedenia sú dostupné po kratšiu dobu, ako u konkurencie, a majú obmedzenie vo veľkosti [73]

Hotjar AI, ktorá využíva generatívnu AI na asistenciu pri vytváraní vlastných používateľských dotazníkov [31].

	Hotjar	Fullstory	Smartlook	Yandex Metrica	Mouseflow	PostHog
Prehrávanie sedenia	✓	✓	✓	✓	✓	✓
Teplotné mapy	Klikanie Posúvanie Pohyby myši Zapojenie Zúrivé klikanie	Klikanie Posúvanie Chybové klikanie Zúrivé klikanie Prázdne klikanie	Klikanie Posúvanie Pohyby myši	Klikanie Posúvanie Prepojenie	Klikanie Posúvanie Pohyby myši Pozornosť Geografická Dynamické	Klikanie Posúvanie Pohyby myši
Konverzné lieviky	✓	✓	✓	✓	✓	✓
Analýza formulárov	✗	✗	✗	✓	✓	✗
Používateľské prieskumy	✓	✗	✗	✗	✓	✓
Sledovanie užívateľských ciest	✗	✓	✓	✗	✓	✓
A/B testovanie	✓	✓	✗	✓	✗	✓

Tabuľka 3.2: Porovnanie ponuky služieb

Nie každý poskytovateľ služieb pre zaznamenávanie sedení zahŕňa v svojej ponuke služieb funkcionalitu ID používateľa. Tabuľka 3.3 ponúka prehľad o tom, kto túto funkcionalitu poskytuje, a kto zas nie.

	Hotjar	Fullstory	Smartlook	Yandex Metrica	Mouseflow	PostHog
ID používateľa	✓	✓	✓	✓	✗	✓

Tabuľka 3.3: Porovnanie doby uloženia záznamov sedení

3.3.1 Doba uchovania údajov

Poskytovatelia služieb pre zaznamenávanie sedení neuchovávajú zhromaždené dáta na svojich serveroch neobmedzene. Tieto dáta sú uchované do ich automatického zmazania alebo kým sa zákazník nerozhodne ich manuálne odstrániť. Konkrétna dĺžka tohto obdobia sa medzi jednotlivými poskytovateľmi líši (ako je zrejmé z tabuľky 3.4) a často závisí od zvoleného predplatného plánu – dlhšia doba uchovávania je zvyčajne spojená s vyššími poplatkami. V prípadoch, kde tabuľka 3.4 uvádza rozsah (napr. 1-12 mesiacov), spodná hranica zvyčajne zodpovedá najnižšiemu, často bezplatnému, predplatnému plánu, zatiaľ čo horná hranica platí pre vyššie platené úrovne. Je tiež potrebné spomenúť, že mnohí poskytovatelia ponúkajú aj individuálne riešenia pre firmy, pri ktorých môže byť doba uchovávania dát dohodnutá na mieru a líšiť sa od štandardných plánov. Tieto detaily však zvyčajne nie sú verejne dostupné. Niektoré platformy taktiež ponúkajú možnosť cieleného archivovania vybraných záznamov pre ich dlhodobejšie zachovanie [58].

	Hotjar	Fullstory	Smartlook	Yandex Metrica	Mouseflow	PostHog
Doba uchovania	12 mesiacov	1-12 mesiacov	1-3 mesiace	15 dní	1-12 mesiacov	1-12 mesiacov[1]

Tabuľka 3.4: Porovnanie doby uloženia záznamov sedení

[1] Neplatí ak zákazník prevádzkuje vlastnú inštanciu PostHog

Dĺžka doby uchovávania súvisí s časovým obdobím, počas ktorého sú akékoľvek nechtiac zaznamenané osobné alebo citlivé údaje používateľov vystavené potenciálnym rizikám. Počas celého obdobia uchovávania existuje riziko nielen ich možného zneužitia (napr. neoprávneným prístupom v rámci organizácie zákazníka či poskytovateľa), ale aj riziko ich odcudzenia pri možnom úniku dát. Takýto únik sa môže uskutočniť rôznymi spôsobmi, či už elektronicky (napr. kybernetický útok) alebo fyzicky (napr. ukradnutím dátových nosičov ako USB kľúče či externé disky) [37].

Obavy o bezpečnosť dát nielen u zákazníka, ale aj priamo u poskytovateľa služby, nie sú len teoretické. V januári 2023 došlo k úniku interného zdrojového kódu (takmer 45 GB) spoločnosti Yandex na platformu BreachForums [42]. Zverejnený materiál obsahoval aj zdrojový kód týkajúci sa ich analytickej služby Yandex Metrica. Podľa dostupných zdrojov tento konkrétny únik nezahŕňal žiadne používateľské dáta, avšak demonštruje potenciálnu zraniteľnosť infraštruktúry aj u veľkých poskytovateľov analytických služieb.

3.4 Nasadenie skriptu na sledovanie sedenia

Nasadenie skriptu pre zaznamenávanie sedenia na webovú stránku zahŕňa niekoľko krokov, a to od prvotnej registrácie u zvoleného poskytovateľa služby až po samotnú implementáciu sledovacieho kódu.

3.4.1 Registračný proces

Prvým krokom pre správcu webovej stránky, ktorý má záujem o služby zaznamenávania sedení, je typicky vytvorenie účtu u vybraného poskytovateľa. Tento proces je pri väčšine poskytovateľov priamočiary. Na základe mojej analýzy poskytovateľov služieb, ktorí sú predmetom tejto práce, môžem konštatovať, že počas registrácie zvyčajne nedochádza k podrobnejšiemu overeniu účelu, na ktorý bude skript nasadený, a ani k validácii použitej e-mailovej adresy. Súčasťou registrácie je akceptovanie zmluvných podmienok poskytovateľa služby, a dodatočne môže byť zákazník požiadaný o uvedenie kategórie svojej webovej stránky (napr. e-shop, vzdelávanie, bankovníctvo), ale hlbšia kontrola často chýba.

Čiastočnú výnimku predstavuje registračný proces spoločnosti Fullstory, ktorá pri registrácii neakceptuje adresy z bežných mailových služieb (ako napr. `@gmail.com`). Okrem toho spoločnosť Fullstory dáva zákazníkovi na výber aj rozsah automatickej redakcie. Zákazník pri registračnom procese definuje množstvo osobných a citlivých údajov, ktoré jeho stránka obsahuje, a tým ovplyvní mieru automatického maskovania obsahu.

3.4.2 Implementácia a mechanizmus fungovania sledovacieho kódu

Po registrácii u poskytovateľa je ďalším krokom implementácia sledovacieho kódu na webovú stránku zákazníka. Poskytovatelia skriptov pre zaznamenávanie sedenia zvyčajne ponúkajú viacero spôsobov inštalácie [23]. Medzi bežné metódy patria:

- **Priame vloženie kódu:** Manuálne pridanie poskytnutého kódu (snippetu) priamo do zdrojového kódu stránky.
- **Manažéry značiek (Tag Managers):** Implementácia prostredníctvom nástrojov ako Google Tag Manager[1], kde sa kód pridá cez rozhranie manažéra bez priamej úpravy kódu stránky.

[1] https://support.google.com/tagmanager/answer/6102821?hl=en

- **NPM balíčky:** Inštalácia pomocou Node Package Manager[1] (NPM), vhodná napríklad pre aplikácie postavené na frameworkoch ako React, Angular alebo Vue.
- **Integrácie pre platformy:** Použitie špecifických pluginov alebo rozšírení pre populárne e-commerce platformy (napr. Shopify, WooCommerce, Magento).

Hoci tieto alternatívne metódy existujú a môžu zjednodušiť samotné nasadenie sledovacieho skriptu na webovú stránku, základným a univerzálne ponúkaným spôsobom zostáva priame vloženie krátkeho JavaScript kódu, tzv. snippetu. Tento snippet poskytne služba zákazníkovi po jeho registrácii a je určený na vloženie do HTML kódu každej stránky, ktorú chce zákazník monitorovať. Mnohé spoločnosti zdôrazňujú práve jednoduchosť tohto kroku, pričom často dodávajú, že na nasadenie nie sú potrebné pokročilé technické znalosti, keďže proces implementácie zvyčajne predstavuje len skopírovanie a vloženie poskytnutého snippetu do hlavičky (`<head>`) alebo tela (`<body>`) dokumentu HTML [56]. Ukážky týchto snippetov sú zobrazené na obrázku 3.1.

[1] https://www.npmjs.com/package/npm

Tag Verify installation Site ID: -------

This tag is unique to **https://example.com/**.

Paste the tag into the `<head>` of every page you wish to track users and collect feedback. And then verify your installation.

Copy

```
<!-- Hotjar Tracking Code for https://example.com/ -->
<script>
    (function(h,o,t,j,a,r){
        h.hj=h.hj||function(){(h.hj.q=h.hj.q||[]).push(arguments)};
        h._hjSettings={hjid:-------,hjsv:6};
        a=o.getElementsByTagName('head')[0];
        r=o.createElement('script');r.async=1;
        r.src=t+h._hjSettings.hjid+j+h._hjSettings.hjsv;
        a.appendChild(r);
    })(window,document,'https://static.hotjar.com/c/hotjar-','.js?sv=');
</script>
```

(a) Hotjar

Install tracking code for website

Add script below to the website header in the <head> tags:

```
<script type='text/javascript'>
  window.smartlook||(function(d) {
    var o=smartlook=function(){ o.api.push(arguments)},h=d.getElementsByTagName('head')
[0];
    var c=d.createElement('script');o.api=new Array();c.async=true;c.type='text/javascrip
t';
    c.charset='utf-8';c.src='https://web-sdk.smartlook.com/recorder.js';h.appendChild(c);
    })(document);
    smartlook('init', '-----------', { region: 'eu' });
</script>
```

Copy script to clipbord Share instructions

(b) Smartlook

Web snippet

PostHog's configurable web snippet allows you to (optionally) autocapture events, record user sessions, and more with no extra work. Place the following snippet in your website's HTML, ideally just above the </head> tag.

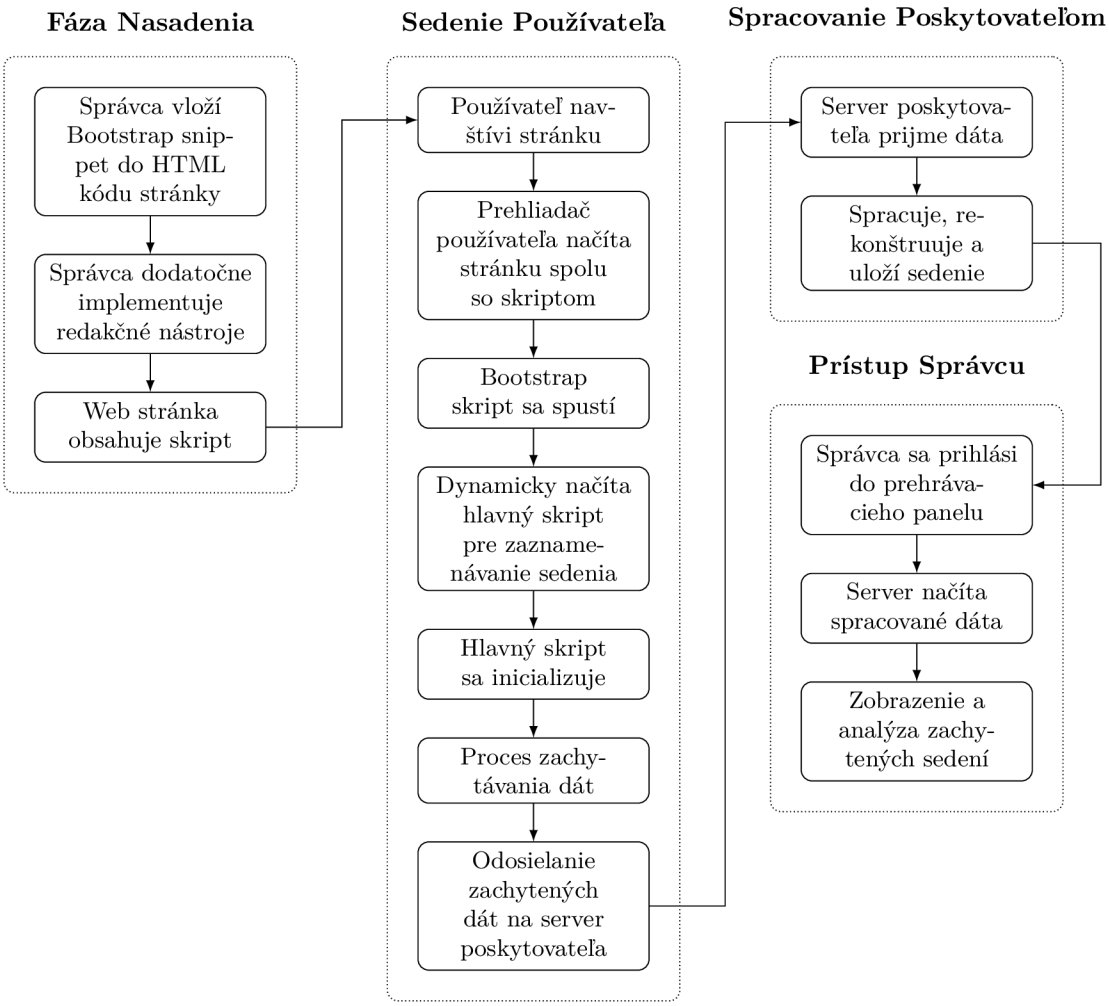
For more guidance, including on identifying users, see PostHog Docs.

```
<script>
    !function(t,e){var o,n,p,r;e.__SV||(window.posthog=e,e._i=[],e.init=function(i,s,a){function g(t,e){var o=e.split(".");2==o.length&
    posthog.init('phc_---------------', {
        api_host: 'https://us.i.posthog.com',
        person_profiles: 'identified_only', // or 'always' to create profiles for anonymous users as well
    })
</script>
```

(c) PostHog

Obr. 3.1: Ukážka snippetov

Tento snippet nezahŕňa kompletný záznamový mechanizmus sám o sebe, ale skôr funguje ako inicializačný (bootstrap) skript. Jeho hlavnou úlohou po načítaní stránky v prehliadači používateľa je dynamicky načítať hlavný zaznamenávací skript zo serverov poskytovateľa služby. Dynamické načítanie je často implementované pomocou štandardných techník jazyka JavaScript [51], ako je vytvorenie elementu `<script>`, nastavenie jeho atribútu `src` na URL adresu hlavného skriptu a následné vloženie tohto elementu do DOM stránky pomocou funkcie `appendChild()`. Až tento externe načítaný skript následne preberá zodpovednosť za zaznamenávanie sedenia, ako je znázornené v diagrame 3.2.



Obr. 3.2: Diagram procesu zaznamenávania sedenia používateľa

3.4.3 Vplyv na výkonnosť stránky

Neoddeliteľnou súčasťou užívateľského zážitku je samotná výkonnosť webu. Zhoršenú výkonnosť webu môžu používatelia vnímať napríklad ako pomalé načítavanie alebo oneskorené reakcie stránky. Dá sa aj merať, a to napríklad zvýšením sieťového prenosu. Rýchlejšia webová stránka je zvyčajne menšia, čo znamená, že má menej kódu a spotrebuje menej zdrojov [19]. Nástroje tretích strán vyžadujú pridanie vlastných zdrojov (JavaScript súborov), ktoré sa starajú o spracovanie údajov vo webovom prehliadači používateľa. To tiež zahŕňa odosielanie zachytených údajov na server tretej strany.

Mnohé spoločnosti, poskytujúce služby pre prehrávanie sedení, avšak tvrdia, že ich skripty sú optimalizované tak, aby neznižovali rýchlosť načítania stránky a ani neovplyvňovali interaktivitu pre používateľa [48]. Príklad takejto optimalizácie je samotná funkcionalita snippetu a to tak, že sa najprv načíta len inicializačný skript, ktorý až následne asynchrónne stiahne a spustí hlavný, oveľa rozsiahlejší zaznamenávací skript, čím sa odloží načítanie väčšiny kódu. Tento prístup môže výrazne znížiť negatívny dopad na počiatočnú rýchlosť načítania stránky.

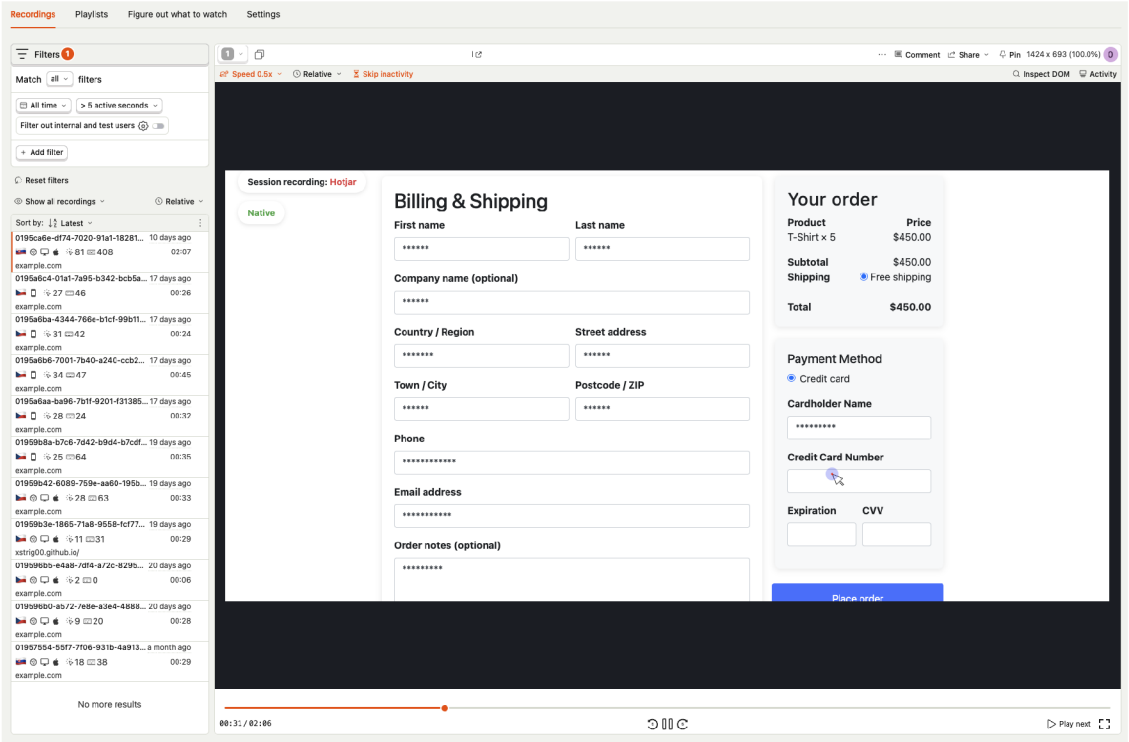
Pri skriptoch na zaznamenávanie sedenia sa často využíva minifikácia. Tento proces zahŕňa jednak odstránenie nepotrebných znakov, ako sú medzery, nové riadky a komentáre, a jednak premenovanie identifikátorov [8]. V rámci minifikácie sú teda premenné, funkcie a parametre premenované na krátke a často nezmyselné názvy, napríklad jednopísmenové alebo skladajúce sa iba z čísel. Benefitom minifikácie pre výkonnosť je zmenšenie veľkosti súboru, čo znamená rýchlejší prenos dát a potenciálne aj rýchlejšie spracovanie skriptu prehliadačom. Hoci primárne slúži na zmenšenie veľkosti súboru, výsledný zhustený kód je tiež ťažšie čitateľný. Štúdia ukazuje, že transformácie kódu ako práve minifikácia sú v bežnom webovom kóde veľmi rozšírené, zatiaľ čo pokročilejšie techniky obfuskácie (napr. enkódovanie častí kódu, získavanie reťazcov z globálneho poľa) sú menej časté [63].

3.4.4 Redakčné nástroje

Aby nedochádzalo v procese zbierania dát k únikom osobných, poprípade citlivých dát, spoločnosti tretích strán ponúkajú kombináciu manuálnych a automatických redakčných nástrojov.

Redakčné nástroje umožňujú správcom webovej stránky manuálne vylúčiť, prípadne zahrnúť, z nahrávok obsah prvkov, ktoré môžu obsahovať citlivé informácie. Tieto prvky môžu zahŕňať napríklad vstupné polia alebo zobrazovaný obsah. Výsledok použitia redakčných nástrojov môže byť rôzny. Pôvodný text môže byť maskovaný iným textom rovnakej dĺžky. V prípade hesiel sa môže jednať o text inej dĺžky, aby sa zabránilo úniku vedomosti o dĺžke hesla. Poprípade môže byť redigovaný text úplne vynechaný zo zaznamenávania [1, 56, 28].

Aby sa však úniku informácií zabránilo, správcovia webových stránok by museli dôkladne kontrolovať a aktualizovať všetky stránky, ktoré zobrazujú alebo prijímajú informácie o používateľoch. Takáto pracná manuálna redakcia je úloha náchylná na chyby, najmä v prípade zložitých a dynamických webových stránok, ktorá môže nepriamo viesť ku úplnej ignorácii redakčných nástrojov [56]. Pravdepodobne bezpečnejším prístupom by bolo maskovanie alebo redakcia všetkých elementov HTML ako východzie nastavenie. S týmto prístupom by správca stránky manuálne vyberal polia, ktorých obsah môže byť odoslaný. Manuálne povolovanie polí však môže viesť k opačnému extrému, kde správca stránky povolí odoslanie dát pre všetky existujúce polia.



Obr. 3.3: Ukážka redigovanej nahrávky sedenia[1] v prehrávacom paneli (Posthog)

Proces redakcie je zo strany správcu webovej stránky možné spravovať dvoma spôsobmi:

1. **Úpravou samotného kódu webovej stránky**, čo prináša väčšiu kontrolu nad tým, ktoré údaje sa budú alebo nebudú zaznamenávať. Prehľad nástrojov používaných na takúto manuálnu redakciu ponúka tabuľka 3.5. Navyše, služba PostHog umožňuje spravovať redigovanie polí aj priamo v snippete (inicializačnom kóde), ktorý sa počas nasadenia vkladá do každej sledovanej stránky.

2. **Prostredníctvom ovládacieho panela**, ktorý poskytovateľ služby ponúka priamo v účte zákazníka pre príslušnú webovú stránku. V ňom je napríklad možné zvoliť polia, elementy HTML a podobne, ktoré budú automaticky redigované, poprípade presne označiť časti stránky, ktoré sa nemajú zaznamenávať. Niektorí poskytovatelia skriptov pre zaznamenávanie sedenia, ako napríklad Mouseflow[2], umožňujú zákazníkovi definovať vlastné názvy tried CSS alebo atribútov HTML, ktoré budú následne v kóde stránky slúžiť na označenie prvkov určených na redakciu. Tento prístup môže byť výhodný, nakoľko umožňuje potenciálne využiť už existujúce triedy používané pri vývoji stránky. Celkovo možno konštatovať, že tento prístup, teda správy redakcie prostredníctvom ovládacieho panelu, je používateľsky prívetivejší a ponúka ho väčšina poskytovateľov analyzovaných v rámci tejto práce.

Mnohé spoločnosti automaticky redigujú všetky vstupné polia na webovej stránke, keďže tieto polia často obsahujú osobné údaje používateľa [32]. Obsah takto označených polí obvykle nemožno zaznamenať ani pomocou nástrojov na povolenie zaznamenávania. Príkladom je použitie atribútu `data-hj-allow` v kombinácii s poľom typu `password`, ktorého

[1] Stránka je nedostatočne redigovaná keďže nahrávka obsahuje informáciu o zakúpených položkách

[2] https://intercom.help/mouseflow/en/articles/4263776-excluding-masking-and-whitelisting-content-with-the-visual-privacy-tool

obsah sa aj tak nezaznamenáva. Proces manuálnej redakcie zobrazovaného obsahu však zostáva vo výsledku na správcovi webovej stránky, čiže na zákazníkovi.

Dôkladný proces redakcie je nevyhnutnou požiadavkou pre väčšinu služieb zaznamenávajúcich sedenie [28]. Po tom, čo skript nazbierané dáta zo sedenia odošle na stranu servera poskytovateľa služby, dané dáta nie je možné opätovne získať alebo redigovať (potlačiť) ich obsah a aktualizácia nastavení redakcie sa často neuplatní so spätnou platnosťou [32]. V skutočnosti stačí niekoľko zle redigovaných polí, ktoré môžu spôsobiť malé, prvotne nevinné úniky. Malé úniky na niekoľkých stránkach môžu následovne viesť k veľkému nahromadeniu osobných údajov v rámci jedného záznamu sedenia [1].

3.4.5 Riziká jednoduchosti nasadenia

Jednoduchosť implementácie snippetu môže byť z pohľadu ochrany súkromia dvojsečnou zbraňou. Ak prevádzkovateľ webovej stránky zanedbá dôkladnú implementáciu redakčných nástrojov, môže byť používateľ neúmyselne vystavený riziku úniku osobných alebo iných citlivých informácií tretím stranám.

Zároveň, pri samotnom rozhodovaní o nasadení skriptu pre zaznamenávanie sedenia by mala vzniknúť otázka, či je jeho použitie pre danú stránku skutočne nevyhnutné a opodstatnené [28]. Je dôležité zvážiť, či problémy, ktoré má nástroj pomôcť riešiť, nie je možné identifikovať a odstrániť už počas fázy návrhu a testovania webovej aplikácie. Pokiaľ je však použitie skriptu pre zaznamenávanie sedení vyhodnotené ako prínosné, malo by byť vždy použité na jasne definované ciele, pričom samotné nasadenie by malo byť časovo ohraničené.

	Nástroje na manuálnu redakciu	Použitie
Hotjar	`data-hj-suppress`	Atribút HTML alebo trieda CSS; potláča všetok textový i obrazový/video obsah daného prvku (vrátane vnorených elementov).
	`data-hj-allow`	Atribút HTML alebo trieda CSS; explicitne povoľuje zaznamenanie obsahu (text, klávesy) len pre označený prvok (nefunguje rekurzívne).
Fullstory	`fs-exclude`	Atribút HTML; vylučuje obsah prvoku z nahrávania (vrátane vnorených elementov), nahrádza ich v zázname šedými pruhmi.
	`fs-mask`	Atribút HTML; nahrádza text nevratným placeholderom (zachová rozloženie a dĺžku textu); vybrané vnorené elementy možno odmaskovať.
	`fs-unmask`	Atribút HTML; povoľuje zaznamenanie obsahu prvku (text, obrázky); zachytávajú sa všetky interakcie s daným prvkom.
Smartlook	Record API	Rozhranie API na manuálne ovládanie zaznamenávania vstupov formulára, IP adries, emailových adries a čísel.
Yandex Metrica	`ym-record-keys`	Trieda CSS; povoľuje zaznamenávanie obsahu vstupného poľa (textarea, input).
	`ym-disable-keys`	Trieda CSS; zabraňuje zaznamenávaniu obsahu vstupného poľa a obsah nahrádza hviezdičkami (textarea, input).
	`ym-hide-content`	Trieda CSS; zabraňuje zaznamenávaniu zobrazovaného prvku; obrázky nahradzuje šedými pruhmi, text rozmazanými znakmi.
	`ym-show-content`	Trieda CSS; umožňuje zaznamenávanie zobrazovaného prvku iba v rámci prvkov označených `ym-hide-content`.
Mouseflow	`mf-masked mf-listen`	Trieda CSS; maskuje obsah prvku ale zaznamenáva interakcie (napr. kliknutia).
	`mf-excluded`	Trieda CSS; úplne vylučuje obsah prvku z nahrávania a ignoruje všetky interakcie s daným prvkom.
	`data-mf-replace`	Atribút HTML; nahrádza obsah prvku placeholderom na zachovanie kontextu bez zobrazenia citlivých údajov.
PostHog	`ph-no-capture`	Trieda CSS; zabraňuje zaznamenávaniu obsahu prvku; prvok je nahradený kvádrom rovnakých rozmerov pri prehrávaní záznamov.

Tabuľka 3.5: Prehľad ponuky redakčných nástrojov a ich použitie

3.5 Súlad s nariadeniami GDPR

Keďže je hlavnou témou mojej práce analýza skriptov pre zaznamenávanie sedenia používateľov, považujem za dôležité tiež preskúmať, či spoločnosti ponúkajúce tieto služby dodržiavajú nariadenia GDPR. Ako občana krajiny v rámci Európskej únie, kde GDPR predstavuje základný právny rámec na ochranu osobných údajov [16], ma zaujíma, ako tieto spoločnosti zabezpečujú ochranu osobných údajov svojich používateľov. Aj napriek tomu, že táto práca je hlavne technického charakteru, verím, že krátke nahliadnutie do právnej stránky spracovania osobných údajov bude mať pozitívny prínos.

3.5.1 Základné pojmy

V tejto časti uvediem definície základných pojmov podľa GDPR [16]:

- **Osobný údaj:** Informácia, ktorá priamo identifikuje jednotlivca (konzistentné dáta ako napríklad meno, adresa, identifikačné číslo alebo kód, telefónne číslo, e-mailová adresa, IP adresa [41, 62], atď.) alebo pomocou ktorej spoločnosť zamýšľa identifikovať konkrétnych jednotlivcov v spojení s inými dátovými prvkami (nepriama identifikácia). Dokonca aj zaheslované alebo zašifrované informácie, ako napríklad e-mailové adresy, sú považované za osobné údaje, ak ich je možné spojiť s konkrétnou osobou [16, 15].

- **Spracovanie osobných údajov:** Akákoľvek operácia alebo súbor operácií vykonávaných s osobnými údajmi, či už ide o ich získavanie, uchovávanie, používanie, zverejňovanie alebo inú formu spracovania, a to manuálne alebo automatizovane vykonanú [14].

- **Prevádzkovateľ údajov:** Subjekt, ktorý určuje účely a prostriedky spracovania osobných údajov. Môže ísť o právnickú osobu, orgán verejnej moci, agentúru alebo akýkoľvek iný subjekt [14].

- **Spracovateľ údajov:** Subjekt, ktorý spracúva osobné údaje výlučne na základe pokynov prevádzkovateľa [14].

3.5.2 Aplikácia GDPR na skripty pre zaznamenávanie sedenia

Služby pre zaznamenávanie sedení spracúvajú údaje o návštevníkoch, pretože zachytenie týchto informácií je základným predpokladom pre samotné fungovanie tejto služby. Bez spracovania širokého spektra dát, ako je popísané v sekcii 2.4 (zahŕňajúce stav a zmeny DOM, používateľské interakcie či rôzne metadáta), by nebolo možné poskytnúť zákazníkovi rekonštrukciu návštevy a ďalšie analytické nástroje, ktoré sú popísané v sekcii 2.3.

Spoločnosti analyzované v rámci tejto práce v tomto prípade teda zastupujú stranu spracovateľa údajov. Všetky z týchto spoločností tvrdia, že ako spracovatelia sú v súlade s nariadeniami GDPR [59, 64, 74, 30, 26, 47]. Je už len na zákazníkovi, aby ako prevádzkovateľ údajov dodržal povinnosti vychádzajúce z nariadení GDPR [16, 47].

Špeciálny prípad je, ak si zákazník sám prevádzkuje PostHog. V takom prípade je zákazník zodpovedný za svoju inštanciu PostHog. Spoločnosť PostHog vtedy nemá prístup k údajom jeho používateľov, takže tu nemá žiadne špecifické povinnosti vzhľadom na nariadenia GDPR [59].

Osobné údaje môžu byť uchovávané buď v papierovej podobe alebo na iných médiách. Pri skriptoch pre zaznamenávanie sedenia sú tieto informácie najčastejšie uschovávané na vzdialenom serveri poskytovateľa skriptov pre zaznamenávanie sedenia, respektíve jeho spracovateľov, takže v elektronickej podobe.

3.5.3 Geografická filtrácia a súlad s GDPR

Keďže súlad s nariadeniami GDPR je právnou požiadavkou pre akýkoľvek subjekt spracúvajúci osobné údaje občanov EÚ (bez ohľadu na sídlo subjektu) [16], relevantnou funkciou, ktorú ponúkajú všetci z poskytovateľov analyzovaných v rámci tejto práce, je možnosť filtrovania alebo úplného blokovania zaznamenávania sedení na základe IP adresy návštevníka. Táto funkcionalita tak teoreticky umožňuje zákazníkom aktívne zamedziť zberu dát od návštevníkov z Európskej únie a tým potenciálne obmedziť rozsah svojich povinností, ako prevádzkovateľ údajov, vyplývajúcich z GDPR v kontexte týchto nástrojov.

Kapitola 4

Metodika

Moja metodológia pozostávala z troch hlavných častí. Prvá časť, dynamická analýza, sa zameriavala na správanie skriptov pre zaznamenávanie sedenia v kontrolovanom prostredí na mnou vytvorených testovacích stránkach (sekcia 4.1). V rámci nej som sledoval, aké dáta sa automaticky redigujú, akým spôsobom je možné tento proces upraviť pomocou nástrojov poskytovaných službami (sekcia 4.1.3), a následne som vyhodnocoval rozsah zachytených údajov v prehrávacom paneli danej služby.

Druhá časť metodológie bola statická analýza zdrojového kódu zaznamenávacích skriptov od analyzovaných poskytovateľov. Tu som skúmal, ako sú implementované interné mechanizmy ako napríklad obsluha manuálnych redakčných nástrojov, logika maskovania obsahu alebo prispôsobenie sa signalizovaným preferenciám používateľa ohľadom sledovania, ktoré sú popísané v sekcii 2.6.3. Následne som sa v rámci statickej analýzy pokúsil modifikovať logiku maskovania obsahu stránky s cieľom dosiahnuť, aby upravený skript zaznamenal stránku v jej pôvodnej, nemaskovanej podobe, identickej s tou, ktorú vidí používateľ. Proces statickej analýzy a úpravy zaznamenávacieho skriptu je popísaný v sekcii 4.2. Zároveň som sa v oboch týchto analytických častiach zameral na problémy zistené v predchádzajúcich štúdiách [1, 19, 28] a overoval som ich stálu prítomnosť.

Treťou časťou mojej práce bola analýza nasadenia a konfigurácie skriptov pre zaznamenávanie sedenia na reálnych webových stránkach. Cieľom tejto časti bolo zistiť, ako sú tieto služby a ich redakčné nástroje (tabuľka 3.5) reálne využívané v praxi.

4.1 Popis testovacích scenárov

Moje testovacie scenáre boli inšpirované metodológiou uvedenou v štúdii zaoberajúcou sa analýzou skriptu od spoločnosti Hotjar [56]. Testovacie scenáre boli zároveň založené na známych spôsoboch úniku informácií zo sekcie 2.5.1. Jednotlivé testovacie scenáre vyzerali nasledujúco:

1. **Prihlasovací formulár** – obsahuje vstupné polia pre hodnoty ako heslo a používateľský email. Prihlasovacie formuláre často umožňujú zobraziť heslo ktoré používateľ zadal, a preto formulár obsahuje tlačidlo, ktoré zmení typ vstupného poľa z `password` na `text` a naopak.

2. **Formulár s poštovými údajmi** – obsahuje vstupné polia pre hodnoty ako meno používateľa, jeho adresa, kontaktné údaje a platobné údaje. Súčasťou stránky je aj rekapitulácia objednávky, aby som mohol otestovať, či sa líši spôsob zachytávania

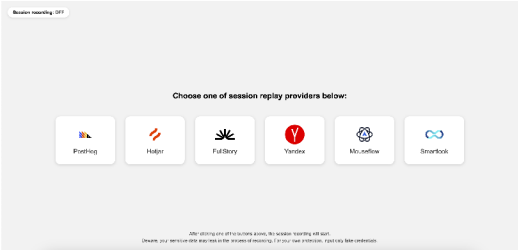
zobrazeného obsahu od toho vo vstupných poliach. Táto rekapitulácia obsahuje aj zoznam zakúpených predmetov. Na stránke sa taktiež nachádza textové pole, do ktorého používateľ zadáva názov krajiny. Stránka priebežne zobrazuje návrhy podľa zadávaného textu a na základe zobrazeného obsahu môže následovne skript prezradiť obsah vstupného poľa aj bez toho aby obsah tohoto vstupného poľa zaznamenával.

Počas testovania mi napadla ďalšia možnosť testovacieho scenára – chatovacia aplikácia. V posledných rokoch zaznamenávame výrazný nárast používania aplikácií so end-to-end šifrovaním, čo je poháňané potrebou bezpečných komunikácií [2]. End-to-end šifrovanie (E2EE) slúži na ochranu súkromia komunikácie tak, aby ju nemohli prečítať ani samotní poskytovatelia služby. Takýto spôsob zabezpečenia je avšak zbytočný, ak vám nejaká služba zaznamenáva obsah obrazovky, čiže ešte pred zašifrovaním alebo po dešifrovaní komunikácie. Nakoniec som tento testovací scenár do svojej testovacej sady nezaradil, pretože by nepriniesol žiadne nové poznatky oproti dvom už použitým scenárom a bol by len plytvaním času počas testovania.

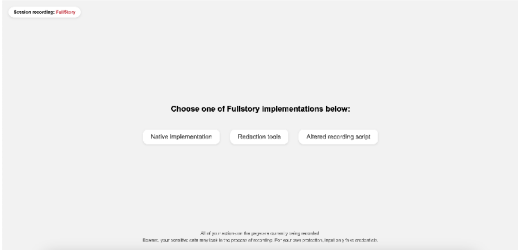
4.1.1 Testovacie Konfigurácie

V rámci praktických experimentov som každý testovací scenár pozoroval v nasledujúcich konfiguráciách:

1. Zaznamenávací skript je nasadený na stránku bez akýchkoľvek ďalších zásahov, teda bez použitia dodatočných redakčných nástrojov.
2. Snažil som sa zaznamenať obsah čo najviac polí pomocou nástrojov, ktoré služba poskytuje na umožnenie zaznamenania pôvodne redigovaných polí.
3. Využil som úpravy zaznamenávajúceho kódu tak, aby dokázal zachytiť obsah pôvodne redigovaných polí, a to bez použitia dostupných nástrojov na povolenie zaznamenania.

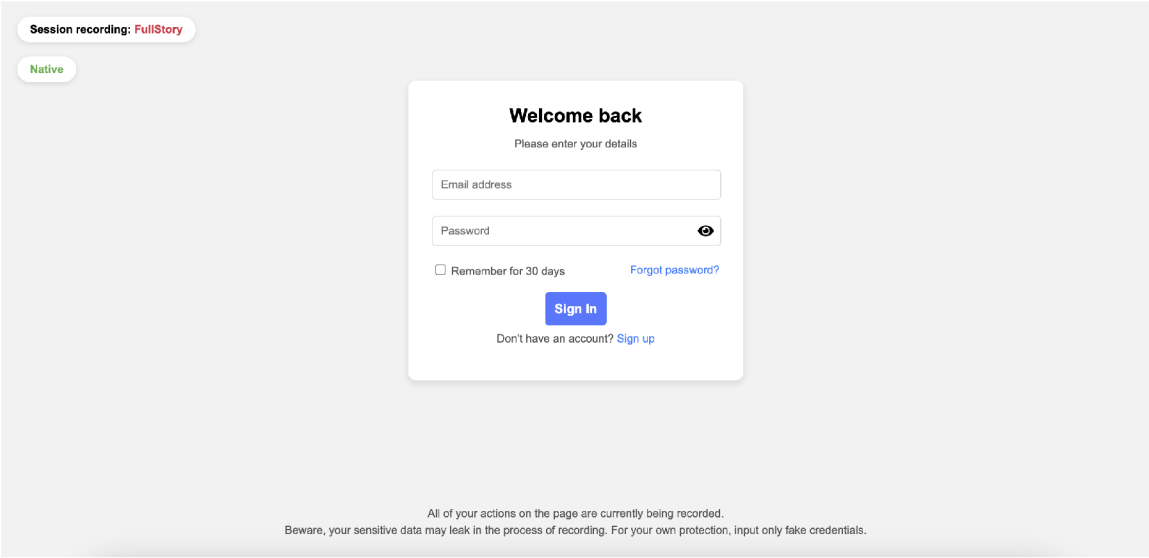


(a) Domovská stránka

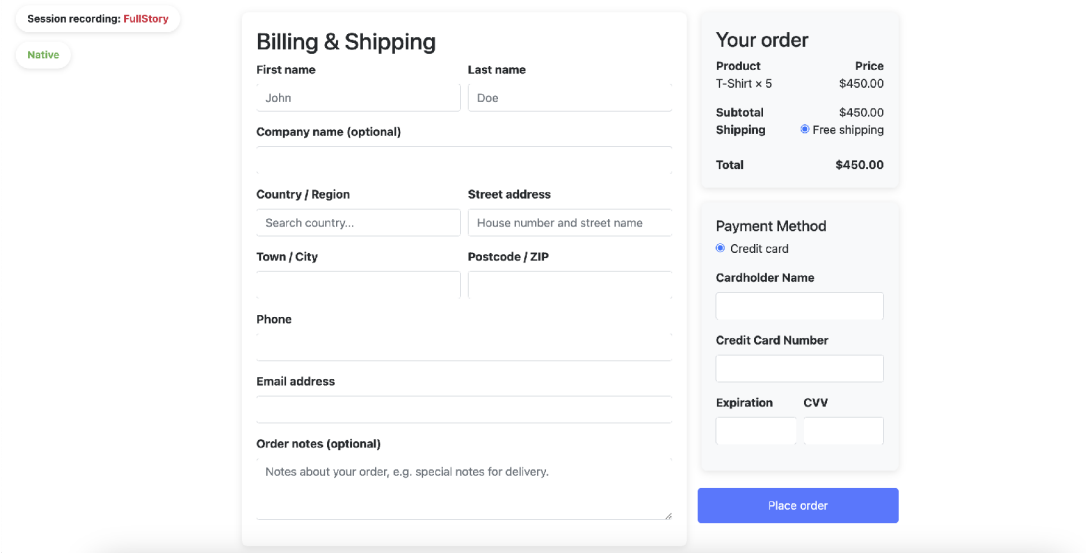


(b) Výber konfigurácie

Obr. 4.1: Ukážky testovacej stránky



Obr. 4.2: Prihlasovací formulár



Obr. 4.3: Formulár s poštovnými údajmi

4.1.2 Implementácia testov

Pri implementácii testovacích scenárov som využil nasledujúce nástroje:

- **GitHub Pages**[1] z dôvodu jednoduchého hosťovania webovej stránky a rýchlej implementácie dodatočných zmien.
- **Google Chrome** ako testovací prehliadač a to bez dodatočného použitia nástrojov pre ochranu užívateľa pred sledovaním (signalizovanie DNT, GPC alebo nástroje tretích strán). Počas testovania som tento prehliadač používal v režime Inkognito so zapnutou možnosťou používania cookies tretích strán.

[1] https://pages.github.com/

4.1.3 Použité redakčné nástroje

Keďže každý poskytovateľ služieb pre zaznamenávanie sedenia implementuje automatickú redakciu vstupných polí a zobrazovaného obsahu odlišne a ponúka iný rozsah nástrojov na manuálnu redakciu, bolo potrebné testovaciu stránku počas testovania dynamicky upravovať. Tabuľka 4.1 ukazuje konkrétne použité redakčné nástroje použité pri testovaní na odmaskovanie zachytávaného obsahu.

	Hotjar	Fullstory	Smartlook	Yandex Metrica	Mouseflow	PostHog
Redakčný nástroj	`data-hj-allow`	`fs-unmask`	Record API	`ym-record-keys`	`capture`[1]	API

Tabuľka 4.1: Redakčné nástroje použité na „odmaskovanie“ obsahu

Tieto triedy CSS a atribúty HTML z tabuľky 4.1 som vždy aplikoval na všetky vstupné polia obidvoch testovacích formulárov – prihlasovací formulár (obrázok 4.2) a formulár s poštovými údajmi (obrázok 4.3). V prípade skriptu od spoločnosti PostHog som využil možnosť spravovania automatickej redakcie priamo z inicializačného kódu pomocou ich programovacieho rozhrania. Nastavil som premennú `maskAllInputs` na hodnotu `false` a to isté som spravil aj pri premenných spadajúcich do objektu `maskInputOptions`. Record API spoločnosti Smartlook som zasa nastavil následovne: premenné `forms`, `numbers`, `emails` a `ips` na hodnotu `true`, čím som explicitne povolil zaznamenávanie týchto typov údajov.

4.2 Proces analýzy a úpravy zaznamenávacieho skriptu

Pri tejto konfigurácii som sa inšpiroval štúdiou z roku 2023 [56], ktorá načrtla taký spôsob úpravy zaznamenávacieho skriptu, ktorý už nemaskoval obsah pôvodne redigovaných polí. Prvým a spoločným krokom pri každom zaznamenávacom skripte je zabrániť inicializačnému snippetu v dynamickom načítaní hlavného zaznamenávacieho skriptu zo vzdialeného servera poskytovateľa. Toto typicky zahŕňa zakomentovanie alebo odstránenie funkcie `appendChild()` v inicializačnom snippete poskytovateľa a vytvorenie si lokálnej kópie hlavného zaznamenávacieho skriptu, ktorú som následne upravoval.

Nasledovné kroky úpravy skriptu sa už, vzhľadom na odlišnú implementáciu zaznamenávacieho skriptu medzi jednotlivými poskytovateľmi, líšia. Keďže tieto skripty sú pomerne rozsiahle a overovanie kódu riadok po riadku nie je efektívne, zameral som sa hlavne na identifikáciu tých sekcií v kóde, ktoré naznačujú možnosť maskovania obsahu. Po identifikácii potenciálnej maskovacej logiky som sa ju snažil upraviť tak, aby k maskovaniu už nedochádzalo. Tento prístup som dosiahol kombináciou nasledujúcich postupov:

- **Vyhľadávanie maskovacích znakov**: Prvým, veľmi jednoduchým, krokom bolo hľadanie znakov, ktorými je maskovaný obsah nahradený (napríklad znak „*“), čo mi umožnilo identifikovať potenciálne oblasti maskovania.
- **Obsluha redakčných nástrojov**: Zároveň som hľadal a skúmal časti kódu, kde sa spracúvajú redakčné nástroje, pretože tieto môžu tiež spúšťať maskovaciu funkciu.
- **Kontrola typov polí**: Ak predchádzajúce metódy nepriniesli požadované výsledky, zameral som sa na hľadanie kontrolných podmienok overujúcich typy polí, ktoré nebolo

[1] Vlastne zadefinovaná trieda CSS ktorá povoluje zaznamenávanie obsahu vstupného poľa, zadefinovaná v nastaveniach prehrávacieho panelu spoločnosti Mouseflow

možné „odmaskovať“ redakčnými nástrojmi (napríklad cez atribút `data-hj-allow`). Zvyčajne išlo o polia obsahujúce heslá (kľúčové slovo `password`) alebo o polia s platobnými údajmi.

4.2.1 Detekcia spracovania DNT/GPC signálov

Pre zistenie, či skript reaguje na signály Do Not Track alebo Global Privacy Control, som v kóde vyhľadával kľúčové slová ako „`navigator.globalPrivacyControl`“ [52] alebo „`navigator.doNotTrack`“ [49]. Následne som analyzoval ich spracovanie.

4.3 Proces analýza nasadenia skriptov na reálnych webových stránkach

Na identifikáciu webových stránok, ktoré využívali skripty od analyzovaných poskytovateľov, som použil verejne dostupné nástroje ako už spomenutý nástroj BuiltWith (sekcia 3.1) alebo Wappalyzer[1]. Tieto nástroje dokážu nielen detegovať prítomnosť rôznych technológií na konkrétnej webovej stránke, ale poskytujú aj zoznamy webových lokalít, ktoré danú technológiu alebo službu využívajú. Pri následnom výbere stránok pre analýzu som primárne vyberal tie s vysokým potenciálom spracovania osobných údajov (napr. e-shopy, bazáry). Taktiež som zohľadnil rozsah nasadenia sledovacieho skriptu na danej stránke a moju prípadnú predchádzajúcu znalosť jej obsahu.

Po identifikácii relevantných stránok som manuálne preskúmal ich zdrojový kód HTML, prípadne dostupné súbory JavaScript. Počas tejto fázy som sa zameral na detekciu a analýzu rozsahu použitia redakčných nástrojov implementovaných priamo v kóde stránky. Medzi sledované redakčné nástroje patria tie z tabuľky 3.5. Pri analýze zdrojového kódu som taktiež kontroloval konfiguráciu inicializačného snippetu, aby som overil, či je funkcia zaznamenávania sedení pre účely prehrávania sedení vôbec aktívna.

Uvedomoval som si však obmedzenia tohto prístupu analýzy. Zistiť úplný rozsah redakcie aplikovanej na zaznamenávané dáta je často nemožné, pretože nastavenia redakcie môžu byť vykonané aj cez ovládací panel poskytovateľa služby (sekcia 3.4.4). Tieto nastavenia sa nemusia odraziť priamo v kóde stránky alebo v súboroch JavaScript zaznamenávacieho skriptu. Problematická je niekedy aj samotná detekcia prítomnosti skriptu, keďže metódy implementácie sú rôznorodé (viď sekcia 3.4.2), čo sťažovalo identifikáciu len na základe analýzy hlavného kódu HTML.

[1] https://www.wappalyzer.com/

Kapitola 5

Výsledky

V tejto kapitole prezentujem výsledky dynamickej a statickej analýzy skriptov pre zaznamenávanie sedenia od analyzovaných poskytovateľov. Táto kapitola obsahuje aj moje zistenia z analýzy reálneho nasadenia týchto skriptov na verejne dostupných webových stránkach. Výsledky vychádzajú z metodológie popísanej v kapitole 4.

5.1 Dynamická analýza

V tejto sekcii prezentujem moje výsledky dynamickej analýzy, v rámci ktorej som skúmal reálne správanie skriptov pre zaznamenávanie sedenia na testovacích stránkach (sekcia 4.1) pri použití prvých dvoch testovacích konfigurácií definovaných v sekcii 4.1.1: *základná implementácia* bez dodatočných úprav redakcie pomocou redakčných nástrojov, kedy som len vložil snippet do kódu testovacej stránky, a *implementácia s použitím redakčných nástrojov na povolenie záznamu*, kde okrem vloženia snippetu do kódu testovacej stránky som použil aj mechanizmy poskytované analyzovanými službami na odmaskovanie zaznamenávaných dát (ako je popísané v sekcii 4.1.3).

Pri prezentácii výsledkov dynamickej analýzy pre jednotlivé konfigurácie (základná implementácia a použitie povoľovacích nástrojov) sa hlavne zameriavam na dva typy udalostí: *zachytený obsah vstupných polí* a *maskovaný zobrazovaný obsah*. Pri zobrazovanom obsahu sa teda sústredím len na prípady, kde k maskovaniu došlo. Tento prístup som zvolil hlavne z dôvodu prehľadnosti výsledku, keďže automatická redakcia (maskovanie) sa typicky sústreďuje hlavne na obsah vstupných polí, zatiaľ čo automatické maskovanie bežného zobrazovaného obsahu je skôr výnimkou. V preklade to znamená, že zaznamenávanie zobrazovaného obsahu v jeho pôvodnej podobe (pokiaľ nie je explicitne maskovaný zákazníkom pomocou redakčných nástrojov) je štandardným správaním a potenciálny únik informácií z neho teda je očakávaný.

Všetky súbory HTML obsahujúce kód testovacích scenárov (popísané v sekcii 4.1) pre jednotlivé analyzované konfigurácie (základná implementácia a implementácia s použitím redakčných nástrojov na povolenie záznamu) sú priložené ako súčasť odovzdaných dát v adresári `resources/src/`.

5.1.1 Sledované dáta

Testovacie scenáre – prihlasovací formulár (obrázok 4.2) a formulár s poštovými údajmi (obrázok 4.3) – obsahovali konkrétne nasledujúce typy vstupných polí, ktorých potenciálne zaznamenanie obsahu som sledoval pri obidvoch konfiguráciách:

- **E-mailová adresa** (typ `email`, id `email`)
- **Heslo** (typ `password`, s možnosťou prepnutia na `text`, id `password`)
- **Meno** (typ `text`, id `firstname`)
- **Priezvisko** (typ `text`, id `lastname`)
- **Názov firmy** (typ `text`, id `company`)
- **Krajina** (typ `text`, id `country`, vyhľadávanie s dynamickým zobrazovaním návrhov)
- **Ulica a číslo domu** (typ `text`, id `address`)
- **Mesto** (typ `text`, id `town`)
- **PSČ** (typ `text`, id `postcode`)
- **Telefónne číslo** (typ `tel`, id `phone`)
- **Poznámky k objednávke** (element `textarea`, id `notes`)
- **Meno držiteľa karty** (typ `text`, id `ccname`, autocomplete `cc-name`)
- **Číslo kreditnej karty** (typ `text`, id `ccnumber`, autocomplete `cc-number`)
- **Dátum expirácie karty** (typ `text`, id `ccexp`, autocomplete `cc-exp`)
- **CVC kód karty** (typ `text`, id `cccvc`, autocomplete `cc-csc`)

Okrem samotných vstupných polí som sa ďalej zameral aj na sledovanie viditeľnosti zobrazovaných prvkov ako sú:

- **Rekapitulácia objednávky**, a to konkrétne:
  - Názov produktu
  - Počet kusov
  - Cenu jednotlivých produktov
  - Celkovú sumu objednávky
- **Dynamicky zobrazované návrhy krajín** (v rámci vyhľadávania krajiny)

Zobrazované prvky vymenované vyššie (rekapitulácia objednávky, návrhy krajín) sú avšak len príkladmi, keďže som v rámci analýzy sledoval viditeľnosť v podstate všetkých zobrazovaných elementov na stránke.

5.1.2 Výsledky dynamickej analýzy podľa poskytovateľa

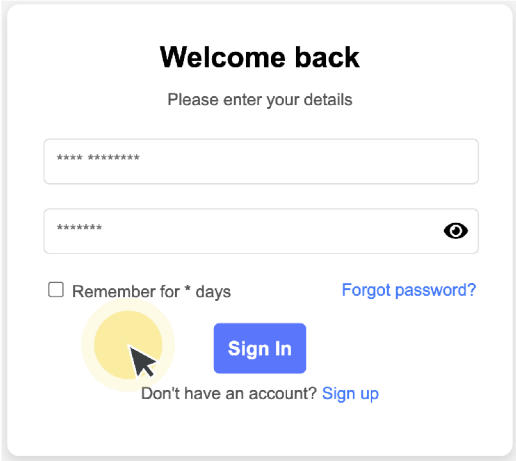
V nasledujúcich sekciách rozoberám výsledky dynamickej analýzy pre každého poskytovateľa zvlášť, pričom som pre každú konfiguráciu (základná implementácia a použitie povoľovacích nástrojov) dodržal štruktúru: cieľ, metodika, výsledky a prípadné porovnanie s predchádzajúcimi štúdiami. Toto štruktúrovanie výsledkov som zvolil za cieľom zlepšiť prehľadnosť zistení a potenciálne zjednodušiť budúcu reprodukciu jednotlivých testov.

Hotjar

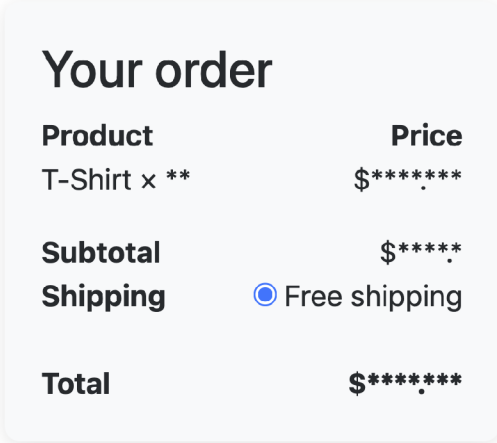
Výsledky dynamickej analýzy skriptu pre zaznamenávanie sedenia od poskytovateľa Hotjar pre obe konfigurácie (sekcia 5.1) sú popísané v nasledujúcich odsekoch.

Základná implementácia:

- **Cieľ:** Určiť predvolený (automatický) rozsah maskovania skriptu poskytovateľa Hotjar pri základnej implementácii a porovnať zistené správanie s výsledkami štúdie z roku 2023 [56].
- **Metodika:** Štandardný snippet poskytovateľa Hotjar som vložil do kódu testovacích stránok (sekcia 4.1) bez aplikovania akejkoľvek dodatočnej manuálnej redakcie nad rámec predvoleného správania skriptu.
- **Testovacie súbory:** `resources/src/Hotjar/native-login.html`, `resources/src/Hotjar/native-shipping.html`
- **Výsledky:** Zistil som, že skript od poskytovateľa Hotjar pri základnej implementácii maskoval obsah všetkých vstupných polí vymenovaných v sekcii 5.1.1. Zaujímavým zistením bolo *maskovanie zobrazovaných číselných údajov*, bez ohľadu na ich umiestnenie, ako ukazuje obrázok 5.1. Okrem číselných údajov nebol maskovaný žiadny iný zobrazovaný obsah na stránke. Pozoroval som, že dynamické návrhy krajín boli počas písania zachytávané. Keďže tieto návrhy reflektujú zadávaný text, ich zaznamenanie môže odhaliť zvolenú krajinu aj v prípade, že samotné vstupné pole bolo maskované.
- **Porovnanie:** Maskovanie zobrazovaných čísel (obrázok 5.1) predstavuje zmenu oproti správaniu zaznamenávacieho skriptu zdokumentovanému v štúdii z roku 2023 [56], kde zobrazované čísla podľa výsledkov maskované neboli. Technický aspekt tejto zmeny správania skriptu rozoberám ďalej v sekcii 5.2.1. Čo sa týka spôsobu maskovania vstupných polí, pri použití typu `tel` pre telefónne číslo (podľa definície v sekcii 5.1.1) som pozoroval maskovanie znakmi `*`. Správanie opísané v štúdii z roku 2023 [56] (maskovanie telefónneho čísla reťazcom "1111") sa mi podarilo reprodukovať až po zmene typu tohto vstupného poľa na `number`. V takom prípade bolo číslo skutočne nahradené jednotkami, čo je v súlade aj s dokumentáciou poskytovateľa Hotjar [32] pre vstupné polia typu `number`.



(a) "Pamätať si prihlásenie"funkcionalita



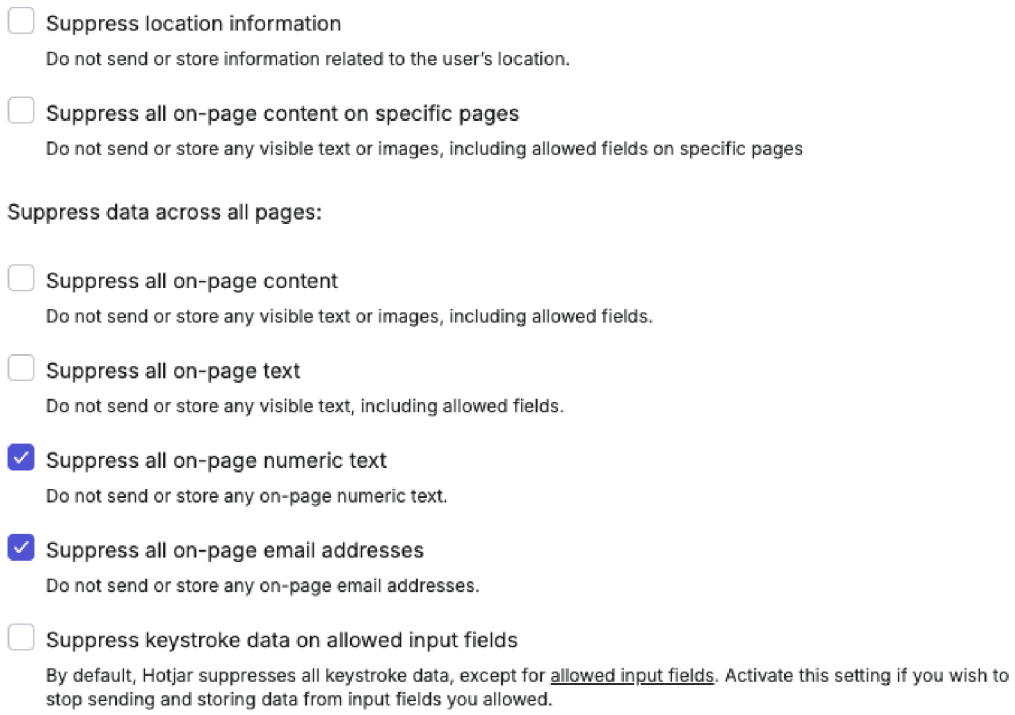
(b) Rekapitulácia objednávky

Obr. 5.1: Maskovanie zobrazovaných čísiel (Hotjar)

Použitie redakčných nástrojov na povolenie záznamu:

- **Cieľ:** Otestovať vplyv atribútu HTML `data-hj-allow` (tabuľka 3.5) na povolenie záznamu obsahu polí a porovnať zistené správanie s výsledkami štúdie z roku 2023 [56].
- **Metodika:** Na všetky relevantné vstupné polia testovacích stránok (vymenované v sekcii 5.1.1) som aplikoval atribút HTML `data-hj-allow`. Tento atribút som aplikoval tiež na elementy obsahujúce číselné údaje v rekapitulácii objednávky a aj v prihlasovacom formulári, ktoré boli pri základnej implementácii maskované (ako je vidieť na obrázku 5.1).
- **Testovacie súbory:** `resources/src/Hotjar/unredact-login.html`, `resources/src/Hotjar/unredact-shipping.html`
- **Výsledky:** Zistil som, že použitie atribútu `data-hj-allow` umožnilo zachytenie obsahu vstupných polí názov firmy, krajina, mesto a poznámky k objednávke. Obsah zvyšných vstupných polí zostal maskovaný. Ďalej som zistil, že použitie atribútu `data-hj-allow` nemalo žiadny vplyv na maskovanie zobrazovaných číselných údajov, ktoré zostali maskované rovnako ako v základnej implementácii (obrázok 5.1). Toto naznačuje, že atribút `data-hj-allow` v analyzovanej verzii skriptu neovplyvňuje logiku maskovania zobrazovaného obsahu riadenú pravdepodobne nastaveniami v ovládacom paneli Hotjar (obrázok 5.2).
- **Porovnanie:** Oproti štúdii z roku 2023 [56], kde atribút `data-hj-allow` umožnil zachytiť napríklad aj údaje o platobnej karte, moja analýza ukázala obmedzenejšie odmaskovanie. Logika zaznamenávacieho skriptu poskytovateľa Hotjar bráni odmaskovaniu polí, pokiaľ ich atribúty HTML (`id`, `name`) zodpovedajú preddefinovaným vzorom pre citlivé dáta [32], a to aj v prípade, že je na nich použitý atribút `data-hj-allow`. Keď som však upravil pôvodné identifikátory vstupných polí (popísané v sekcii 5.1.1) tak, aby týmto vzorom už neodpovedali, atribút `data-hj-allow` úspešne umožnil zachytenie ich obsahu, čo odpovedá správaniu skriptu Hotjar popísanému v štúdií z roku

2023 [56]. Toto potvrdzuje, že účinnosť atribútu `data-hj-allow` je podmienená nezhodou s interne definovanými identifikátormi pre osobné alebo citlivé údaje. Technické detaily tejto podmienenej logiky odmaskovania popisujem v sekcii 5.2.1.



Obr. 5.2: Ukážka východzích nastavení redakcie zaznamenávacieho skriptu (Hotjar)

Fullstory

Výsledky dynamickej analýzy skriptu pre zaznamenávanie sedenia od poskytovateľa Fullstory pre obe konfigurácie (sekcia 5.1) sú popísané v nasledujúcich odsekoch.

Základná implementácia:

- **Cieľ:** Určiť predvolený (automatický) rozsah maskovania skriptu poskytovateľa Fullstory. Nástroj od tohoto poskytovateľa využíval poskytovateľom odporúčané nastavenia pre používaný typ stránky, čiže strednú úroveň automatickej redakcie (funckionalita *Form Privacy* aktívna).
- **Metodika:** Štandardný snippet poskytovateľa Fullstory som vložil do kódu testovacích stránok (sekcia 4.1) bez aplikovania akejkoľvek dodatočnej manuálnej redakcie nad rámec predvoleného správania skriptu.
- **Testovacie súbory:** `resources/src/Fullstory/native-login.html`, `resources/src/Fullstory/native-shipping.html`
- **Výsledky:** Faktorom ovplyvňujúcim predvolené správanie skriptu poskytovateľa Fullstory bola automaticky aktívna funkcionalita *Form Privacy*[1]. Táto funkcionalita, ktorej

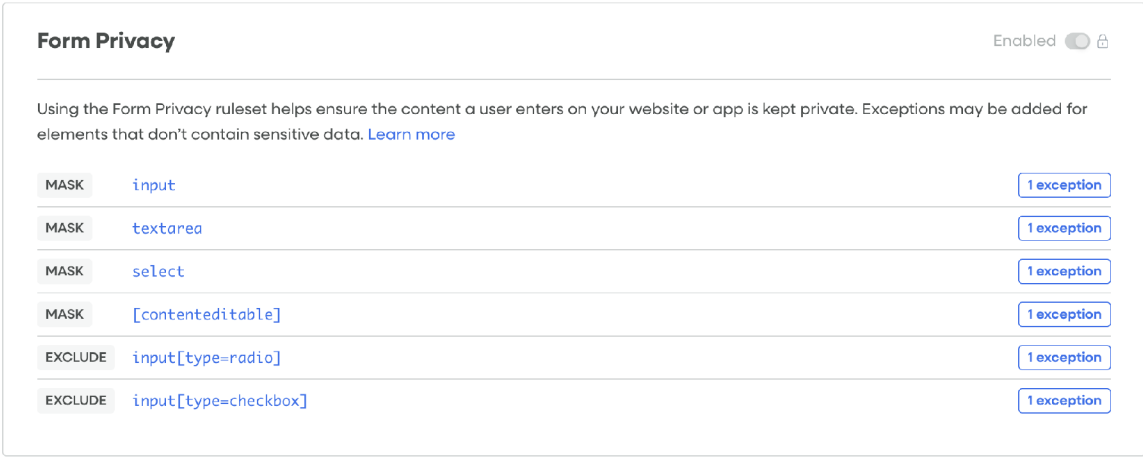
[1] https://help.fullstory.com/hc/en-us/articles/4408633932439-Form-Privacy

predvolené nastavenia sú zobrazené na obrázku 5.3, má za cieľ automaticky maskovať obsah väčšiny bežných formulárových vstupov (ako `input`, `textarea`, `select`). V dôsledku toho som zistil, že pri základnej implementácii bol skriptom poskytovateľa Fullstory maskovaný obsah všetkých vstupných polí na testovacích stránkach (vymenované v sekcii 5.1.1). Okrem toho som pozoroval, že boli maskované aj voľby vykonané pomocou vstupného poľa typu *checkbox* a zo záznamov boli taktiež vynechané voľby vykonané pomocou tlačidiel typu *radio button*. Zobrazovaný obsah stránky maskovaný nebol. Dynamické návrhy krajín boli zachytené, vďaka čomu môže byť nepriamo zistené čo používateľ zadal do tohto poľa aj pri maskovaní samotného vstupného poľa pre krajinu.

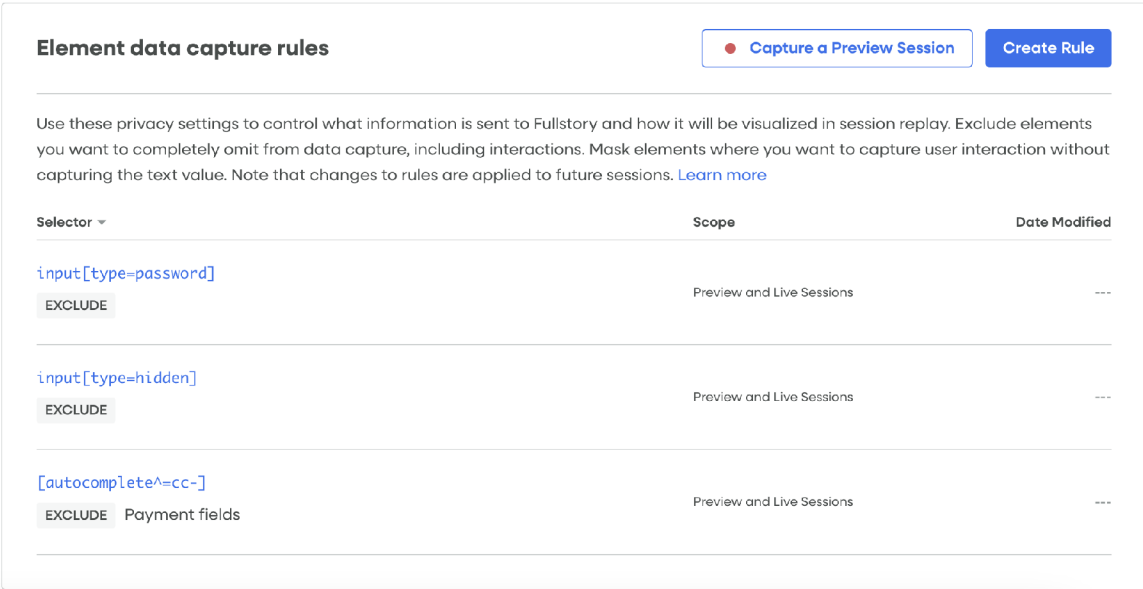
Použitie redakčných nástrojov na povolenie záznamu:

- **Cieľ:** Otestovať vplyv atribútu HTML `fs-unmask` (tabuľka 3.5) na povolenie záznamu obsahu polí.
- **Metodika:** Na všetky vstupné polia testovacích stránok (sekcia 5.1.1) som aplikoval atribút HTML `fs-unmask`.
- **Testovacie súbory:** `resources/src/Fullstory/unredact-login.html`, `resources/src/Fullstory/unredact-shipping.html`
- **Výsledky:** Zistil som, že pridanie atribútu `fs-unmask` k vstupným poliam nemalo žiadny viditeľný efekt, pokiaľ zostala aktívna funkcionalita *Form Privacy* v ovládacom paneli poskytovateľa Fullstory. Všetky vstupné polia zostali maskované ako v základnej implementácii, čo potvrdzuje nadradenosť nastavení ovládacieho panela (obrázok 5.3 a obrázok 5.4) nad implementáciou redakcie v kóde.
- **Doplňujúce výsledky (vypnutie Form Privacy):** Keď som následne v paneli deaktivoval funkcionalitu *Form Privacy*, došlo k zachyteniu obsahu väčšiny predtým maskovaných vstupných polí. Konkrétne boli zaznamenané emailová adresa, meno, priezvisko, názov firmy, krajina, ulica a číslo domu, mesto, PSČ, tel. číslo a poznámky k objednávke. Voľby vykonané pomocou vstupného poľa typu *checkbox* boli zaznamenané taktiež. Zaznamenané boli aj voľby vykonané pomocou tlačidiel typu *radio button*, ktoré boli v základnej implementácií maskované. Zobrazovaný obsah stránky nebol maskovaný. Heslo a údaje o kreditnej karte zostali maskované vďaka pravidlám definovaným v *Element data capture rules*[2] (obrázok 5.4). Táto funkcionalita sa avšak bohužial v bezplatnej verzii deaktivovať nedá.

[2] https://help.fullstory.com/hc/en-us/articles/1500002888501-How-do-I-create-exceptions-to-element-data-capture-rules



Obr. 5.3: Funkcionalita *Form Privacy* (Fullstory)



Obr. 5.4: Funkcionalita *Element data capture rule* (Fullstory)

Smartlook

Výsledky dynamickej analýzy skriptu pre zaznamenávanie sedenia od poskytovateľa Smartlook pre obe konfigurácie (sekcia 5.1) sú popísané v nasledujúcich odsekoch.

Základná implementácia:

- **Cieľ:** Určiť predvolené maskovacie správanie skriptu poskytovateľa Smartlook.
- **Metodika:** Štandardný snippet poskytovateľa Smartlook som vložil do kódu testovacích stránok (sekcia 4.1) bez aplikovania akejkoľvek dodatočnej manuálnej redakcie nad rámec predvoleného správania skriptu.
- **Testovacie súbory:** `resources/src/Smartlook/native-login.html`, `resources/src/Smartlook/native-shipping.html`

- **Výsledky:** Pozoroval som, že skript od poskytovateľa Smartlook pri základnej implementácii maskoval obsah všetkých vstupných polí vymenovaných v sekcii 5.1.1. Zistil som aj maskovanie obsahu vstupných polí typu *checkbox*, tak isto ako to bolo pri skripte od poskytovateľa Fullstory. Ďalej som pozoroval, že skript od poskytovateľa Smartlook, presne tak isto ako Hotjar, štandardne maskoval všetky zobrazované číselné údaje na testovacích stránkach. Okrem týchto číselných údajov som nezistil maskovanie žiadneho iného zobrazovaného obsahu (texty, názvy produktov atď.). Dynamické návrhy krajín boli zachytené, vďaka čomu môže byť nepriamo zistené čo používateľ zadal do tohto poľa aj pri maskovaní samotného vstupného poľa pre krajinu.

Použitie redakčných nástrojov na povolenie záznamu:

- **Cieľ:** Otestovať vplyv konfigurácie rozhrania Record API (tabuľka 3.5) na povolenie záznamu špecifických typov dát.
- **Metodika:** Rozhranie Record API od poskytovateľa Smartlook som nakonfiguroval tak, aby explicitne povolilo zaznamenávanie formulárov, čísel, emailov a IP adries, ako je podrobne opísané v sekcii 4.1.3.
- **Testovacie súbory:** `resources/src/Smartlook/unredact-login.html`, `resources/src/Smartlook/unredact-shipping.html`
- **Výsledky:** Zistil som, že takéto explicitné povolenie záznamu viedlo k zachyteniu obsahu nasledovných vstupných polí: email, meno, priezvisko, názov firmy, krajina, ulica a číslo domu, mesto, PSČ, telefónne číslo a poznámky k objednávke. Údaje o kreditnej karte a heslo zostali maskované. Obsah vstupných polí typu *checkbox* už nebol naďalej maskovaný. Zároveň, vďaka povoleniu zaznamenávania čísel, prestalo byť uplatňované aj maskovanie zobrazovaných číselných údajov na stránke. Zobrazovaný obsah stránky tým pádom nebol vôbec maskovaný.

Yandex Metrica

Výsledky dynamickej analýzy skriptu pre zaznamenávanie sedenia od poskytovateľa Yandex Metrica pre obe konfigurácie (sekcia 5.1) sú popísané v nasledujúcich odsekoch.

Základná implementácia:

- **Cieľ:** Určiť predvolený (automatický) rozsah maskovania skriptu poskytovateľa Yandex Metrica.
- **Metodika:** Štandardný snippet poskytovateľa Yandex Metrica som vložil do kódu testovacích stránok (sekcia 4.1) bez aplikovania akejkoľvek dodatočnej manuálnej redakcie nad rámec predvoleného správania skriptu.
- **Testovacie súbory:** `resources/src/Yandex/native-login.html`, `resources/src/Yandex/native-shipping.html`
- **Výsledky:** Pozoroval som, že skript od poskytovateľa Yandex Metrica ako jediný z analyzovaných poskytovateľov zachytával obsah niektorých vstupných polí už pri základnej implementácií. Kvôli závislosti maskovania obsahu vstupného poľa od jeho

identifikátora (detailne popísané v sekcii 5.2.4) som zistil, že bol zachytený obsah nasledovných vstupných polí: názov firmy, krajina, mesto, PSČ, poznámky k objednávke, meno držiteľa karty, číslo kreditnej karty, dátum expirácie karty a CVC kód karty. Maskované boli len heslo, email, meno, priezvisko, tel. číslo a ulica s číslom domu. Nevýhodou prístupu maskovania založeného na identifikátoroch je jeho závislosť na správnej implementácii týchto identifikátorov správcom stránky, čo môže viesť k neúplnému maskovaniu. Dynamické návrhy krajín boli zachytené, avšak tie už sú pri zachytenom obsahu vstupného poľa krajina zbytočné. Zobrazovaný obsah stránky nebol vôbec maskovaný.

Použitie redakčných nástrojov na povolenie záznamu:

- **Cieľ:** Otestovať vplyv triedy CSS `ym-record-keys` (tabuľka 3.5) na povolenie záznamu obsahu polí.
- **Metodika:** Na všetky vstupné polia testovacích stránok (sekcia 5.1.1) som aplikoval triedu CSS `ym-record-keys`.
- **Testovacie súbory:** `resources/src/Yandex/unredact-login.html`, `resources/src/Yandex/unredact-shipping.html`
- **Výsledky:** Zistil som, že použitie triedy `ym-record-keys` viedlo k zachyteniu obsahu všetkých vstupných polí vymenovaných v sekcii 5.1.1, vrátane vstupných polí, ktorých obsah bol maskovaný v základnej implementácii. Zobrazovaný obsah zostal nezmenený (nebol maskovaný).

Mouseflow

Výsledky dynamickej analýzy skriptu pre zaznamenávanie sedenia od poskytovateľa Mouseflow pre obe konfigurácie (sekcia 5.1) sú popísané v nasledujúcich odsekoch.

Základná implementácia:

- **Cieľ:** Určiť predvolený (automatický) rozsah maskovania skriptu poskytovateľa Mouseflow.
- **Metodika:** Štandardný snippet poskytovateľa Mouseflow som vložil do kódu testovacích stránok (sekcia 4.1) bez aplikovania akejkoľvek dodatočnej manuálnej redakcie nad rámec predvoleného správania skriptu.
- **Testovacie súbory:** `resources/src/Mouseflow/native-login.html`, `resources/src/Mouseflow/native-shipping.html`
- **Výsledky:** Zistil som, že skript poskytovateľa Mouseflow pri základnej konfigurácii maskoval obsah všetkých vstupných polí vymenovaných v sekcii 5.1.1, a to bez výnimky. Zobrazovaný obsah stránky nebol vôbec maskovaný. Dynamické návrhy krajín boli zachytené, vďaka čomu môže byť nepriamo zistené čo používateľ zadal do tohto poľa aj pri maskovaní samotného vstupného poľa pre krajinu.

Použitie redakčných nástrojov na povolenie záznamu:

- **Cieľ:** Otestovať vplyv použitia triedy CSS `capture`, definovanej v ovládacom paneli poskytovateľa Mouseflow, na zmenu rozsahu maskovania obsahu polí. Spôsob definovania tejto triedy je popísaný v sekcii 4.1.3.
- **Metodika:** Na všetky vstupné polia testovacích stránok (sekcia 5.1.1) som aplikoval triedu CSS `capture`.
- **Testovacie súbory:** `resources/src/Mouseflow/unredact-login.html`, `resources/src/Mouseflow/unredact-shipping.html`
- **Výsledky:** Pozoroval som unikátnu závislosť rozsahu maskovania od spôsobu zadávania dát. Pri manuálnom zadaní dát bol obsah vstupných polí krajina, ulica a číslo domu, mesto a PSČ zachytený. Ak som však pre tieto vstupné polia použil funkciu automatického dopĺňania (autocomplete), ich obsah bol v nahrávke maskovaný. Toto isté avšak neplatí pre údaje o kreditnej karte (meno držiteľa, číslo, expirácia, CVC), ktoré som zistil, že boli vždy zachytené (nemaskované), bez ohľadu na spôsob zadania. Vstupné pole s poznámkami k objednávke som testoval len s manuálnym zadaním údajov, a jeho obsah bol zachytený. Obsah vstupných polí e-mail, heslo, meno, priezvisko, názov firmy a telefónne číslo zostal vždy maskovaný, bez ohľadu na spôsob zadania. Zobrazovaný obsah nebol vôbec maskovaný.

PostHog

Výsledky dynamickej analýzy skriptu pre zaznamenávanie sedenia od poskytovateľa PostHog pre obe konfigurácie (sekcia 5.1) sú popísané v nasledujúcich odsekoch.

Základná implementácie:

- **Cieľ:** Určiť predvolený (automatický) rozsah maskovania skriptu poskytovateľa PostHog.
- **Metodika:** Štandardný snippet poskytovateľa PostHog som vložil do kódu testovacích stránok (sekcia 4.1) bez aplikovania akejkoľvek dodatočnej manuálnej redakcie nad rámec predvoleného správania skriptu.
- **Testovacie súbory:** `resources/src/Posthog/native-login.html`, `resources/src/Posthog/native-shipping.html`
- **Výsledky:** Zistil som, že skript poskytovateľa PostHog pri základnej konfigurácii maskoval obsah všetkých vstupných polí vymenovaných v sekcii 5.1.1, bez výnimky. Zobrazovaný obsah stránky nebol vôbec maskovaný. Dynamické návrhy krajín boli zachytené, vďaka čomu môže byť nepriamo zistené čo používateľ zadal do tohto poľa aj pri maskovaní samotného vstupného poľa pre krajinu.

Použitie redakčných nástrojov na povolenie záznamu:

- **Cieľ:** Otestovať vplyv konfiguračných volieb rozhrania API na povolenie záznamu v snippete poskytovateľa PostHog.

- **Metodika:** Inicializačný snippet poskytovateľa PostHog som upravil pomocou rozhrania API (tabuľka 4.1) tak, aby explicitne povolil zachytávanie obsahu vstupných polí (detailné nastavenie tohoto rozhrania API je popísané v sekcií 4.1.3).
- **Testovacie súbory:** `resources/src/Posthog/unredact-login.html`, `resources/src/Posthog/unredact-shipping.html`
- **Výsledky:** Zistil som, že takéto explicitné povolenie záznamu cez rozhranie API viedlo k zachyteniu obsahu všetkých vstupných polí vymenovaných v sekcii 5.1.1. Zobrazovaný obsah na stránke nebol vôbec maskovaný.

5.2 Statická analýza

Popri dynamickej analýze správania skriptov pre zaznamenávanie sedenia na testovacích stránkach som vykonal aj statickú analýzu ich zdrojového kódu. Cieľom tejto analýzy a následných úprav bolo modifikovať zaznamenávacie skripty tak, aby prestali vykonávať automatickú redakciu obsahu stránky a namiesto toho zachytávali kompletný obsah stránky v nemaskovanej podobe. Pri väčšine analyzovaných poskytovateľov sa mi podarilo úspešne identifikovať a upraviť tieto časti kódu zodpovedné za proces redakcie obsahu stránky. Takéto úpravy v podstate viedli k zachyteniu stránky v takej podobe, ako ju vidí používateľ. V závere tejto sekcie taktiež zhrniem, ako jednotlivé skripty pristupujú k signalizácii preferencií používateľa ohľadom sledovania.

V podsekciach 5.2.1 až 5.2.6 približne popisujem konkrétne úpravy, ktoré som vykonal na lokálnych kópiách zaznamenávacích skriptov, aby som dosiahol požadované správanie. Možnosť reprodukcie týchto úprav v budúcnosti, avšak, nie je zaručená, keďže poskytovatelia tieto zaznamenávacie skripty priebežne upravujú a ich budúca verzia sa môže líšiť od tej, ktorú som mal možnosť analyzovať v rámci tejto práce.

Takýto proces pravidelných úprav zdrojového kódu som mal možnosť pozorovať priamo počas obdobia testovania pri skriptoch spoločností Hotjar a PostHog. Tieto dva skripty som testoval na začiatku procesu statickej analýzy. Pri neskoršom overení som zaznamenal, že ich aktuálne verzie, dynamicky načítavané zo serverov poskytovateľov, sa už líšili od pôvodne analyzovaných verzií.

Po bližšej analýze novšej verzie skriptu PostHog som odhalil predovšetkým zmeny v názvoch premenných a funkcií, zatiaľ čo základná logika zaznamenávania a maskovania zostala z veľkej časti nezmenená. Sedenia zaznamenané pomocou pôvodnej (upravenej) verzie skriptu spoločnosti PostHog boli naďalej prehrateľné v prehrávacom paneli poskytovateľa. Naopak, skript spoločnosti Hotjar prešiel za toto obdobie rozsiahlejšími úpravami, ktoré pravdepodobne zahŕňali aj zmeny v dátovom formáte alebo internej logike skriptu, nakoľko sedenia zaznamenané pôvodnou (upravenou) verziou sa už vôbec v prehrávacom paneli nezobrazovali.

Všetky upravené zaznamenávacie skripty od analyzovaných poskytovateľov sú súčasťou odovzdaných dát, konkrétne sa nachádzajú v adresári `resources/js/`.

5.2.1 Hotjar

Vychádzajúc z poznatkov získaných v rámci predchádzajúcej štúdie z roku 2023 [56], som úspešne reprodukoval spôsob modifikácie zaznamenávacieho skriptu spoločnosti Hotjar. Hlavnou úpravou bola zmena funkcie `getSuppressedText`, ktorá je zodpovedná za samotný

akt nahradenia pôvodného obsahu tým maskovaným. Pôvodná implementácia tejto funkcie (výpis 5.1) vracala potenciálne maskovanú verziu vstupného textu.

```
getSuppressedText: function(e, t) {
    var n = P[e];
    return n ? n(t) : R(t)
}
```

Výpis 5.1: Pôvodná funkcia `getSuppressedText` (Hotjar)

Túto funkciu som upravil (výpis 5.2) tak, aby namiesto maskovaného reťazca vždy priamo vracala pôvodný vstupný reťazec (parameter `t`), čím som jej maskovaciu logiku úplne odstránil.

```
getSuppressedText: function(e, t) {
    return t
}
```

Výpis 5.2: Upravená funkcia `getSuppressedText` s odstráneným maskovaním (Hotjar)

Ďalej som nastavil premennú `suppression` na hodnotu `"none"` (výpis 5.4), zatiaľ čo pôvodne jej hodnota závisela od inej premennej (`p`) a mohla byť aj `"full"` (výpis 5.3).

```
suppression: p ? "full" : "none"
```

Výpis 5.3: Pôvodné nastavenie premennej `suppression` (Hotjar)

```
suppression: "none"
```

Výpis 5.4: Upravené nastavenie premennej `suppression` (Hotjar)

Výsledkom týchto úprav je, že skript pre zaznamenávanie sedenia teraz zachytáva v pôvodnej, nemaskovanej podobe nielen obsah vstupných polí, ale aj bežný text zobrazovaný na stránke.

Maskovanie zobrazeného obsahu na stránke

Pri analýze zaznamenávacích skriptov Hotjar (verzie z roku 2023 a tej aktuálnej, ktorá je súčasťou odovzdaných dát) som hľadal konkrétnu zmenu v kóde, ktorá by vysvetľovala rozdielne maskovanie zobrazovaných číslic medzi týmito verziami, ako bolo popísané v sekcii 5.1.2. Preskúmal som funkcie zodpovedné ako za proces serializácie DOM (napr. `serializeNode`), tak aj za logiku samotného maskovania (napr. `getSuppressedTextNode`, `shouldSuppressTextContent`, `getSuppressedNode`, `textContentHandler`). Napriek tomu sa mi v kóde aktuálneho skriptu nepodarilo identifikovať úpravu, ktorá by bola za pozorovanú zmenu priamo zodpovedná, ak by boli ostatné nastavenia zaznamenávacieho skriptu rovnaké.

Rozdielne správanie teda s najväčšou pravdepodobnosťou vyplýva z odlišnej hodnoty premennej `anonymize_digits` v konfigurácii zaznamenávacieho skriptu. Predpokladám, že kým v staršej verzii zaznamenávacieho skriptu analyzovanej v roku 2023 [56] bola táto hodnota nastavená na `false` (čo viedlo k nemaskovaniu bežných číslic), v aktuálnej verzii,

resp. v tej analyzovanej, je jej východzia hodnota nastavená na hodnotu `true`, ako je vidieť aj na predvolených nastaveniach v ovládacom paneli poskytovateľa (obrázok 5.2).

Pri hodnote `true` totiž skript identifikuje textové uzly obsahujúce číslice ako vyžadujúce maskovanie. Toto som potvrdil aj experimentálne. Keď som nastavil v aktuálnom zaznamenávacom skripte premennú `anonymize_digits` na hodnotu `false` sa správanie vrátilo k stavu z roku 2023, kedy zobrazované čísla maskované neboli [56]. Možnosť zmeny tohto nastavenia je dostupná pre správcu stránky priamo v ovládacom paneli služby Hotjar (obrázok 5.2).

Podmienené zachytenie obsahu vstupného poľa

Účinnosť atribútu HTML `data-hj-allow` (tabuľka 3.5) pri odmaskovaní vstupných polí v skripte poskytovateľa Hotjar je podmienená jeho internou logikou. Skript obsahuje preddefinovaný zoznam identifikátorov (atribúty HTML `id` alebo `name`), ktoré skript používa na rozpoznanie polí s citlivými dátami. Ak vstupné pole obsahuje identifikátor z tohto zoznamu, jeho obsah zostane v nahrávke maskovaný aj napriek prítomnosti odmaskovacieho atribútu `data-hj-allow`.

Tento zoznam citlivých identifikátorov som našiel v kóde zaznamenávacieho skriptu, v premennej označenej ako `s` a je znázornený vo výpise 5.5. Zoznam identifikátorov obsahuje dokonca aj slovenský (poprípade český) identifikátor `heslo`.

```
var s = ["address", "address1", "address2", "addressline1", "addressline2",
    "cell", "cellphone", "dateofbirth", "dob", "email", "familyname",
    "firstname", "fullname", "lastname", "mobile", "name", "phone",
    "postalcode", "postcode", "surname", "tel", "telephone", "username",
    "zip", "zipcode", "nationalinsurancenumber", "nin", "ppsn", "security",
    "securitynum", "socialsec", "socialsecuritynumber", "socsec", "ssn",
    "adgangskode", "authpw", "contrasena", "contrasenya", "contrasena",
    "contrasinal", "cyfrinair", "fjalekalim" , "focalfaire", "geslo",
    "haslo" , "heslo", "jelszo" , "kennwort", "kodikos",
    "kodikosprosvasis", "lozinka", "lykilord" , "losenord", "motdepasse",
    "parakalo", "parola", "paroladordine", "parole", "parool", "pasahitza",
    "pass", "passord", "password", "passwort", "pw", "pwd", "pword",
    "pwrd", "salasana", "sapwd", "senha", "sifre", "slaptazodis", "userpw",
    "userpwd", "wachtwoord", "lozinka", "parola", "parol", "kalimatalsir",
    "pasvard", "pasuwado", "mima", "amho", "cc", "cccsc", "cccvc",
    "cccvv", "ccexp", "ccexpiry", "ccexpmonth", "ccexpyear", "ccname",
    "ccnum", "ccnumber", "cctype", "creditcard", "csc", "cvc", "cvv",
    "exp", "accountnum", "accountnumber", "bic", "iban", "ibanaccountnum",
    "ibanaccountnumber", "pin", "pinno", "pinnum", "secq", "secret",
    "secretq", "secretquestion", "securityq", "securityquestion",
    "sortcode", "swift"];
```

Výpis 5.5: Zoznam identifikátorov ovplyvňujúcich zachytenie obsahu poľa (Hotjar)

5.2.2 Fullstory

V rámci úprav zdrojového kódu zaznamenávacieho skriptu poskytovateľa Fullstory som vykonal viacero zmien s cieľom zaznamenať obsah všetkých vstupných polí bez ich masko-

vania alebo vylúčenia, ktoré implementujú funkcionality ako *Form Privacy* a *Element data capture rules*, a to bez ich manuálneho vypnutia.

Hoci som musel zasiahnuť do viacerých častí kódu, hlavné úpravy som vykonal v dvoch kľúčových funkciách. Prvou bola funkcia zodpovedná za spracovanie udalostí zmeny hodnoty vstupného poľa (udalosť `VALUECHANGE`), spracovávaná funkciou `t.prototype.un`. Pôvodná implementácia tejto metódy, ktorá obsahovala logiku pre podmienené maskovanie hodnoty poľa, je zobrazená vo výpise 5.6. Uvádzam celý (minifikovaný) kód funkcie, aby ju bolo možné identifikovať podľa jej štruktúry aj v prípade, že sa budúce verzie skriptu budú líšiť v názvoch premenných, samotnej funkcie alebo iným spôsobom.

```
t.prototype.un = function(t, n, i) {
           var r = this;
           void 0 === n && (n = !1),
           void 0 === i && (i = !1);
           var e = ks(t);
           if (t && e && !this.ln(e, t)) {
               var s = vo(this.W, t);
               if (lo(t)) {
                   var u = ho(t);
                   "false" === s && this.Gt[u] === e ? delete this.Gt[u] :
                       "true" === s && (this.Gt[u] = e)
               } else
                   this.on(e, no.Value, s);
               this.W.measurer.enqueue(function() {
                   var u = t.getBoundingClientRect()
                     , o = u.width > 0 && u.height > 0
                     , a = gs(t) ? $e(s) : s;
                   r.zt.enqueue({
                       Kind: O.VALUECHANGE,
                       Args: [e, a, n, o, i]
                   })
               })
           }
       }
```

Výpis 5.6: Pôvodná funkcia `t.prototype.un` pre spracovanie zmien hodnoty (FullStory)

V tejto funkcii som odstránil podmienku `gs(t) ? $e(s) : s`, ktorá rozhodovala o aplikovaní maskovania (pomocou funkcie `$e` ktorá sa na rôznych miestach v kóde používa v kontexte maskovania) na hodnotu `s`. Namiesto toho teraz premenná `a`, ktorá sa zapisuje do dát udalosti `VALUECHANGE`, vždy obsahuje pôvodnú, nemaskovanú hodnotu poľa. Časť upravenej funkcie je zobrazená vo výpise 5.7.

```
t.prototype.un = function(t, n, i) {
    ...
    this.W.measurer.enqueue(function() {
        var u = t.getBoundingClientRect()
            , o = u.width > 0 && u.height > 0
```

```
            , a = s;
        r.zt.enqueue({
    ...
}
```

Výpis 5.7: Upravená časť funkcie `t.prototype.un` s odstráneným maskovaním (FullStory)

Druhou kľúčovou modifikáciou prešla funkcia určujúca stav prvku z hľadiska sledovania pri spracovaní DOM stromu (metóda identifikovaná ako `t.prototype.getWatchState`), kde som po získaní pôvodného stavu prvku (pomocou volania `this.Wn.isWatched(t)`) pridal kontrolu, či ide o element typu `INPUT`. Ak áno, v príznakoch stavu (state flags) som odstránil príznaky pre maskovanie (`Ge.Mask`) a vylúčenie (`Ge.Exclude`), čím som zabránil internému označeniu týchto polí ako maskované alebo vylúčené z nahrávky. Vzhľadom na rozsah úprav v rámci tejto funkcie tu neuvediem jej priamu ukážku kódu, ale jej kompletná podoba je dostupná v priložených dátach.

Tieto úpravy, v spojení s ďalšími menšími zásahmi, deaktivujú mechanizmy pre automatickú redakciu pre vstupné polia a umožňujú ich zaznamenanie v nezmenenej podobe.

5.2.3 Smartlook

Počas úprav zaznamenávacieho skriptu poskytovateľa Smartlook som sa zameral hlavne na logiku zodpovednú za serializáciu prvkov DOM (v rámci funkcie `serializeNode`) a spracovanie udalostí zo vstupných polí (vo funkcii `handleInput`).

Vo funkcii spracúvajúcej textové uzly (identifikovanej ako `Ie`) som odstránil podmienené maskovanie a volanie externej maskovacej funkcie `Ne`. Pôvodná implementácia funkcie `Ie` je zobrazená vo výpise 5.8.

```
Ie = (e, t, i, s) => {
    let n = e.nodeValue;
    return null === n ? "" : (n = s.isMaskedNode(e) ?
        n.replace(/\S/gi, "*") : Ne(n, t),
    n)
}
```

Výpis 5.8: Pôvodná funkcia `Ie` pre spracovanie textových uzlov (Smartlook)

Funkciu `Ie` som upravil tak, aby priamo vracala hodnotu textového uzla (`n`) bez akéhokoľvek maskovania alebo volania externej funkcie. Výsledná upravená podoba tejto funkcie je viditeľná vo výpise 5.9.

```
Ie = (e, t, i, s) => {
    let n = e.nodeValue;
    return null === n ? "" : n;
}
```

Výpis 5.9: Upravená funkcia `Ie` s odstráneným maskovaním (Smartlook)

Samotnú maskovaciu funkciu `Ne`, určenú na redakciu e-mailov a čísel, som upravil tak, aby vždy vrátila vstupnú hodnotu (premenná `e`) bez modifikácie. Pôvodná verzia funkcie `Ne` je uvedená vo výpise 5.10.

```
Ne = (e, t) => {
    let i = e;
    return t.record.emails || (i = Te(i)),
    t.record.numbers || (i = (e => e.replace(we, "*"))(i)),
    i
}
```

Výpis 5.10: Pôvodná maskovacia funkcia `Ne` (Smartlook)

Upravená funkcia `Ne`, ktorá teraz jednoducho vracia pôvodný vstup, je zobrazená vo výpise 5.11.

```
Ne = (e, t) => {
    return e
}
```

Výpis 5.11: Upravená funkcia `Ne` vracajúca vstup bez modifikácie (Smartlook)

V hlavnej serializačnej logike zaznamenávacieho skriptu (vo funkcii `serializeNode`) som upravil spracovanie elementov `INPUT` a `TEXTAREA`. Odstránil som volanie maskovacej funkcie `Ne` pri zaznamenávaní obsahu uzla (vlastnosť aj atribút `value`). Zároveň som odstránil časti kódu, ktoré zodpovedali za odstraňovanie obsahu pre vstupné polia.

Následne som upravil funkciu reagujúcu na udalosti `input` a `change` (`handleInput`), kde som zmenil logiku tak, aby sa aktuálna hodnota poľa zaznamenala priamo do udalosti (`NODE_MUTATION` alebo `INPUT`) bez aplikovania maskovania, ktoré sa pôvodne aplikovalo na výslednú premennú (`i_val`) pred jej odoslaním. Kvôli rozsahu úprav vykonaných na týchto funkciách (`serializeNode` a `handleInput`) neuvádzam ukážky ich pôvodných ani upravených podôb. Kompletná upravená verzia skriptu je však dostupná ako súčasť dát odovzdaných spolu s touto prácou.

Týmito zásahmi do funkcií `serializeNode`, `handleInput`, `Ie` a `Ne` som docielil vypnutie automatickej redakcie pre obsah vstupných polí, čím som dosiahol kompletné zaznamenávanie ich obsahu. Keďže moje úpravy zasiahli aj všeobecné spracovanie textových uzlov (`Ie`) a maskovaciu funkciu (`Ne`), deaktivoval som tým aj automatickú redakciu bežného zobrazovaného textového obsahu.

5.2.4 Yandex Metrica

Pri úprave zaznamenávacieho skriptu služby Yandex Metrica som najprv upravil tú časť logiky skriptu, ktorá umožňovala vypnúť zaznamenávanie obsahu polí pomocou tried CSS `ym-disable-keys` a `-metrika-nokeys`. Toto som dosiahol úpravou funkcie `fp` (výpis 5.12), ktorá bola zodpovedná za detekciu týchto tried. Zmenil som ju tak, aby bez ohľadu na prítomnosť uvedených tried vždy vrátila hodnotu `false` (výpis 5.13), čím som efektívne znemožnil vypnutie zaznamenávania obsahu polí pomocou týchto tried CSS.

```
function fp(a, b) {
    var c = a && (Na(a.className, "ym-disable-keys") ||
                  Na(a.className, "-metrika-nokeys"));
    return b && a ? c || !!Nj(a).length : c
}
```

Výpis 5.12: Pôvodná funkcia `fp` (Yandex Metrica)

```
function fp(a, b) {
    return false;
}
```

Výpis 5.13: Upravená funkcia `fp` (Yandex Metrica)

Druhou úpravou bolo to, aby skript považoval pole za explicitne označené na zaznamenanie (ako keby malo triedu CSS `ym-record-keys`). Vo funkcii `ip`, ktorá okrem iného zisťovala prítomnosť tejto triedy (výpis 5.14), som priamo nastavil výsledok tejto kontroly (premenná `f`) na hodnotu `true` (výpis 5.15).

```
function ip(a, b, c) {
        c = void 0 === c ? !1 : c;
        if (!b)
            return {
                Wa: !1,
                hb: !1,
                qb: !1
            };
        var d = b.getAttribute("type") || b.type;
        if ("button" === d)
            return {
                Wa: !1,
                hb: !1,
                qb: !1
            };
        var e = mb(ep, [b.className, b.id, b.name])
          , f = b && Of("ym-record-keys", b);
        d = d && K(d, Ro) || Nb(Wa($o), e);
        var g;
        (g = d) || (g = b.placeholder,
        g = Nb(Wa(Zo), e) || ep(g) && bp.test(g || ""));
        e = g;
        return {
            Wa: !f && (hp(a, b) || e && c || e && !d && !c),
            hb: f,
            qb: e
        }
    }
```

Výpis 5.14: Pôvodná funkcia `ip` (Yandex Metrica)

```
function ip(a, b, c) {
    ...
    var e = mb(ep, [b.className, b.id, b.name])
      , f = true;
    d = d && K(d, Ro) || Nb(Wa($o), e);
    ...
}
```

Výpis 5.15: Časť upravenej funkcie `ip` (Yandex Metrica)

Nakoniec som upravil aj funkciu `qq`, kde v časti kódu, ktorá pôvodne kontrolovala prítomnosť triedy CSS `ym-record-keys` (výpis 5.16), som nahradil volanie funkcie `Of()` s parametrami `"ym-record-keys", b` priamo hodnotou `true` (výpis 5.17).

```
function qq(a, b, c, d, e, f) {
    ...
    g.pb = !a && (e || b)) : (g.pb = e,
    c = !(b && Of("ym-record-keys", b))),
    c || e) && (d = "" + d,
    ...
}
```

Výpis 5.16: Časť pôvodnej funkcie `qq` kontrolujúca triedu `ym-record-keys` (Yandex Metrica)

```
function qq(a, b, c, d, e, f) {
    ...
    g.pb = !a && (e || b)) : (g.pb = e,
    c = !(b && true)),
    c || e) && (d = "" + d,
    ...
}
```

Výpis 5.17: Časť upravenej funkcie `qq` s vynúteným `true` namiesto kontroly triedy (Yandex Metrica)

V dôsledku úprav týchto funkcií (`fp`, `ip` a `qq`) zaznamenávací skript nevykonáva žiadne maskovanie obsahu polí, vďaka čomu skript pre zaznamenávanie sedenia teraz zachytáva obsah vstupných polí v ich pôvodnej, nemaskovanej podobe.

Implementácia maskovania na základe identifikátorov

Maskovanie vstupných polí na základe ich identifikátorov je v skripte Yandex Metrica implementované pomocou preddefinovaného zoznamu kľúčových slov a regulárneho výrazu. Najprv je zadefinované pole reťazcov (v kóde identifikované ako premenná `So`), ktoré obsahuje anglické termíny bežne spojené s osobnými údajmi, ako ukazuje výpis 5.18.

```
var So = "first(-|\\.|_|\\s){0,2}name last(-|\\.|_|\\s){0,2}name zip
    postal address passport (bank|credit)(-|\\.|_|\\s){0,2}card
    card(-|\\.|_|\\s){0,2}number card(-|\\.|_|\\s){0,2}holder cvv
    card(-|\\.|_|\\s){0,2}exp card(-|\\.|_|\\s){0,2}name card.*month
    card.*year card.*month card.*year password
    birth(-|\\.|_|\\s){0,2}(day|date) second(-|\\.|_|\\s){0,2}name
    third(-|\\.|_|\\s){0,2}name patronymic middle(-|\\.|_|\\s){0,2}name
    birth(-|\\.|_|\\s){0,2}place house street city flat state
    contact.*".split(" ")
```

Výpis 5.18: Pole `So` s kľúčovými slovami pre citlivé polia (Yandex Metrica)

Následne sa toto pole spojí s ďalším poľom obsahujúcim ruské ekvivalenty a vytvorí sa z nich rozsiahly regulárny výraz (premenná `bp`), ako je vidieť vo výpise 5.19.

```
var bp = new RegExp("(" + So.concat(\u0438\u043c\u044f...
    \u043e\u0431\u043b\u0430\u0441\u0442\u044c).join("|") + ")","i");
```

Výpis 5.19: Regulárny výraz `bp` pre detekciu citlivých identifikátorov (Yandex Metrica)

Tento regulárny výraz `bp` je potom použitý vo funkcii zodpovednej za analýzu vstupných polí (funkcia `ip`, ktorej pôvodnú podobu už ukazuje výpis 5.14).

5.2.5 Mouseflow

Moje úpravy zaznamenávacieho skriptu spoločnosti Mouseflow, ktoré boli podstatne rozsiahlejšie v porovnaní s ostatnými analyzovanými skriptami, začali nastavením globálnej premennej `mouseflowDisableKeyLogging = false;` v snahe ovplyvniť maskovanie dát zadávaných používateľom. Táto úprava sama o sebe nemala žiadny vplyv na správanie zaznamenávacieho skriptu.

Následne som modifikoval funkciu `serializeNode`, ktorá zodpovedá za transformáciu DOM štruktúry stránky do formátu vhodného na prenos a uloženie dát pre nahrávku. V rámci tejto funkcie som upravil alebo odstránil logiku volajúcu kontrolné funkcie ako `_441` (kontrolujúca triedu CSS `mf-excluded`), `_506` (kontrolujúca triedu CSS `mf-masked`), a kontrolnú funkciu `_847` (zahŕňajúcu kontrolu tried `mf-excluded`, `mf-masked`, `no-mouseflow` a iné podmienky pre ochranu súkromia užívateľa).

Taktiež som zasiahol do funkcie `_600`, ktorá rozhoduje o maskovaní hodnoty vstupných polí na základe ich typu a rôznych príznakov (napr. `_3.gdprEnabled`, `_3._446`), a funkcie `_889`. Navyše som upravil funkcie ako `_232` (získavanie hodnoty z elementu typu `input`) a `_706` (získavanie textového obsahu elementu) tak, aby vždy vracali pôvodné, nemaskované hodnoty. Tým, že upravené funkcie (`_232` a `_706`) teraz vracali priamo pôvodné hodnoty, som sa snažil obísť aj mechanizmy ako automatické maskovanie polí pre heslá alebo detekciu a maskovanie potenciálnych čísel kreditných kariet (kontrolované napr. funkciou `_263`).

Úpravy, ktoré som vykonal na tomto zaznamenávacom skripte boli rozsiahle a dohromady prevýšili počet 20 modifikovaných miest v zdrojovom kóde skriptu. Z dôvodu tohto rozsahu zásahov preto pri skripte od poskytovateľa Mouseflow neprikladám konkrétne ukážky upraveného kódu. Kompletná upravená verzia skriptu je však dostupná ako súčasť dát odovzdaných spolu s touto prácou.

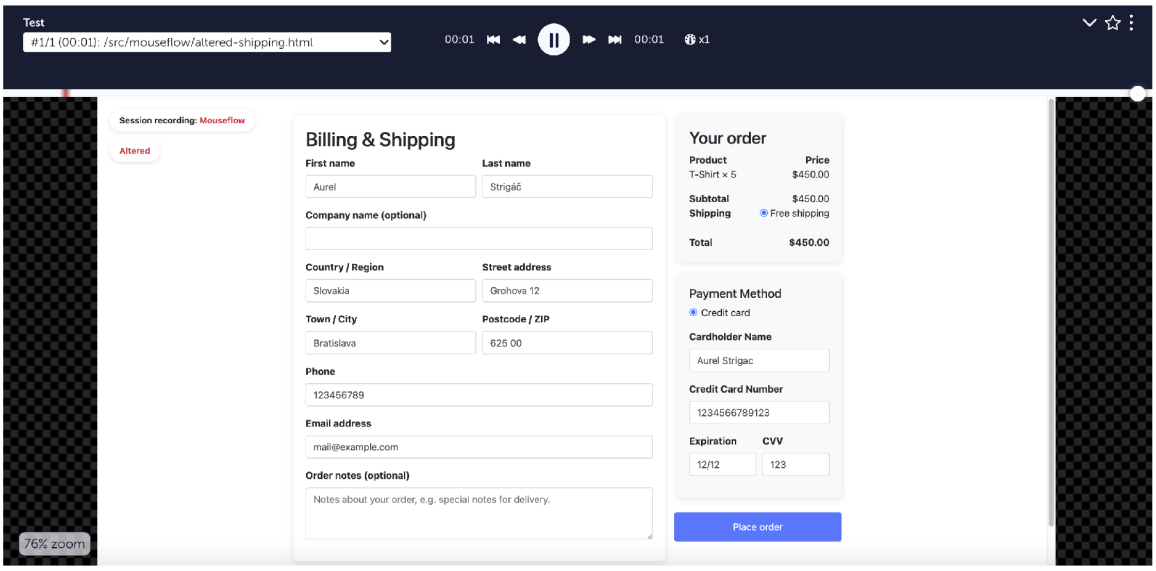
Úprava zaznamenávacieho skriptu avšak dosiahla len čiastočný úspech – obsah polí bol odkrývaný iba na začiatku nahrávania, neskôr maskovanie ďalej prebiehalo. Napriek tomu, že úprava nezabezpečila odkrývanie obsahu vstupných polí po celú dobu sedenia, aj tento počiatočný únik informácií možno využiť. Keďže Mouseflow spracúva každú návštevu podstránky ako samostatnú nahrávku, ak sa upravený skript spustí tesne pred interakciou vedúcou k opusteniu stránky (napríklad pred odoslaním formulára), práve jeho počiatočný záznam – ktorý obsahuje ešte nemaskovaný stav vstupných polí – sa stane touto samostatnou nahrávkou pre danú podstránku. Výsledkom je, že takto neskoro zachytená nahrávka bude obsahovať aktuálne vyplnené (a vďaka úprave dočasne nemaskované) údaje používateľa tesne pred jeho odchodom.

Tento scenár som demonštroval implementáciou skriptu, ktorého kód je zobrazený vo výpise 5.20. Po kliknutí na tlačidlo „Place order“ (s id `place-order-btn`) nedošlo k okamžitému presmerovaniu používateľa. Namiesto toho sa aktivovala vizuálna indikácia načítavania (zmena kurzora) a s dvojsekundovým oneskorením sa dynamicky spustil upravený zaznamenávací skript. Ihneď po uplynutí tohto intervalu, ktorý dáva skriptu dostatok času na inicializáciu a zachytenie momentálneho stavu stránky, bol používateľ presmerovaný na nasledujúcu stránku.

```
<script>
    document.addEventListener('DOMContentLoaded', function() {
        const placeOrderBtn = document.getElementById('place-order-btn');
        const targetUrl = 'altered-forms.html';

        if (placeOrderBtn) {
            placeOrderBtn.addEventListener('click', function(event) {
                event.preventDefault();

                document.body.classList.add('loading');

                const script = document.createElement('script');
                script.src = "../../js/Mouseflow/ef6ef5b9-cd69-43f2-
                    90d3-fc6cfbd9faa8.js";
                script.async = true;
                document.head.appendChild(script);

                setTimeout(function() {
                    document.body.classList.remove('loading');

                    window.location.href = targetUrl;
                }, 2000);
            });
        }
    });
</script>
```

Výpis 5.20: Kód implementujúci oneskorené načítanie upraveného skriptu Mouseflow

Pre používateľa sa tento proces javí ako spomalenie odozvy stránky. Zachytenie všetkých dát na stránke spôsobené takouto implementáciou znázorňuje obrázok 5.5.



Obr. 5.5: Ukážka zachytenia osobných informácií upraveným zaznamenávacím skriptom (Mouseflow)

Časová značka

Pri analýze skriptu poskytovateľa Mouseflow som si všimol, že obsahuje reťazec reprezentujúci časovú značku podľa štandardu ISO 8601 [39] uložený v premennej `_979`. Taktiež som si všimol, že táto časová značka menila hodnotu len zriedkavo – konkrétne dvakrát počas môjho testovania.

Zmena tejto značky priamo korelovala so zmenou stavu premennej `_750`. Z mojej analýzy vyplynulo, že táto premenná reprezentuje stav aktivácie procesu zaznamenávania sedení pre danú webovú stránku v systéme Mouseflow. Počas počiatočnej fázy nasadenia skriptu na testovaciu stránku, kedy bola hodnota `_750` nastavená na `false` (indikujúc prebiehajúcu aktiváciu procesu zachytávania sedení používateľov na danej stránke), skript používal prvú verziu časovej značky. Počas tohto obdobia bolo zaznamenávanie sedení používateľov nefunkčné.

Približne do pol hodiny od prvotného nasadenia skriptu na stránku sa tento serverový proces dokončil, hodnota premennej `_750` sa zmenila na `true` (signalizujúc aktiváciu zaznamenávacieho procesu) a práve vtedy došlo k druhej (a počas môjho testovania poslednej) zmene časovej značky v zdrojovom kóde zaznamenávacieho skriptu poskytovateľa Mouseflow. Od tohto momentu bolo zaznamenávanie sedení plne funkčné a časová značka hodnotu nezmenila.

Počas testovania som zistil, že ak sa táto hodnota v odosielaných dátach nezhodovala s tou, ktorú server očakával, príslušná nahrávka sedenia sa v prehrávacom paneli vôbec nezobrazila. Overil som to tak, že po dokončení serverovej inicializácie (kedy bola premenná `_750` už nastavená na `true`) som použil v lokálnej kópii skriptu staršiu verziu časovej značky (zodpovedajúcu stavu `_750 = false`) – záznamy sedení zachytené takýmto skriptom sa v prehrávacom paneli vážne nezobrazili. Tento mechanizmus teda reálne slúži ako základná ochrana pred prijímaním dát z neautorizovaných, napríklad lokálne uložených, upravených alebo zastaraných verzií zaznamenávacieho skriptu Mouseflow.

Keďže sa však táto hodnota mení len zriedkavo, prakticky som potvrdil, že túto kontrolu je možné relatívne ľahko obísť. Stačilo zistiť aktuálne platnú časovú značku (ktorú

som získal zo zaznamenávacieho skriptu dynamicky získaného zo servera poskytovateľa) a manuálne ju vložiť do mojej lokálnej upravenej kópie zaznamenávacieho skriptu. Týmto som kontrolu efektívne obišiel a mohol som tak využívať moju upravenú verziu skriptu pre zaznamenávanie používateľských sedení až do nasledovnej zmeny tejto časovej značky.

5.2.6 PostHog

Pri práci so zaznamenávacím skriptom od spoločnosti PostHog som úspešne identifikoval funkciu `O`, ktorá sa nachádza v súbore `recorder.js`, ako zodpovednú za proces maskovania obsahu vstupných polí. Pôvodná implementácia tejto funkcie je zobrazená vo výpise 5.21.

```
function O(e) {
    var {element: t, maskInputOptions: r, tagName: n, type: o, value: a,
         maskInputFn: i} = e
      , s = a || ""
      , l = o && x(o);
      return (r[n.toLowerCase()] || l && r[l]) &&
             (s = i ? i(s, t) : "a".repeat(s.length)),
              s
}
```

Výpis 5.21: Pôvodná funkcia `O` pre maskovanie vstupov (PostHog)

Následne som upravil túto funkciu `O` tak, aby vždy vracala pôvodnú hodnotu vstupného poľa (v tomto prípade premennú `s`). Logiku, ktorá pôvodne aplikovala maskovanie (nahradenie znakov hviezdičkami), som odstránil. Výsledná upravená funkcia je zobrazená vo výpise 5.22.

```
function O(e) {
    var {element: t, maskInputOptions: r, tagName: n, type: o, value: a,
         maskInputFn: i} = e
      , s = a || "";
    return s
}
```

Výpis 5.22: Upravená funkcia `O` s odstráneným maskovaním (PostHog)

Výsledkom mojej úpravy je, že táto funkcia už nevykonáva žiadne maskovanie obsahu polí, vďaka čomu skript pre zaznamenávanie sedenia teraz zachytáva obsah vstupných polí v ich pôvodnej, nemaskovanej podobe.

5.2.7 Reakcia na signály DNT/GPC

Zistil som, že iba polovica analyzovaných poskytovateľov implementuje rešpektovanie hodnoty signálu Do Not Track (DNT), ako ukazuje tabuľka 5.1. Žiadny z analyzovaných skriptov však nerešpektuje novší signál Global Privacy Control (GPC).

Kontrola hodnoty signálu DNT nie je vykonávaná v inicializačnom snippete, ale až v hlavnom zaznamenávacom skripte. Tento prístup, pri ktorom sa najprv musí načítať a spustiť hlavný skript predtým, než sa môže zistiť používateľova preferencia o sledovaní,

	Hotjar	Fullstory	Smartlook	Yandex Metrica	Mouseflow	PostHog
Signál DNT	✓	✗	✗	✗	✓	✓

Tabuľka 5.1: Rešpektovanie signálu DNT v zaznamenávacm skripte poskytovateľa

som pozoroval pri všetkých troch poskytovateľoch, ktorí tento signál rešpektujú (Hotjar, Mouseflow, PostHog).

Skutočnosť, že signál DNT je spracovaný až v hlavnom skripte, bola zaznamenaná už pre Hotjar v štúdii z roku 2023 [56]. Dôsledkom je, že aj prehliadač používateľa s aktívnym signálom DNT (hodnota `DNT: 1`, indikujúc nesúhlas používateľa so sledovaním) musí najprv získať hlavný zaznamenávací skript zo serverov poskytovateľa, aby mohla byť používateľova preferencia vôbec rozpoznaná a nahrávanie prípadne zastavené.

Zaznamenávací skript spoločnosti Hotjar (výpis 5.23) rešpektoval tento signál už v roku 2023 [56] a moja analýza potvrdila, že sa na tom odvtedy nič nezmenilo. Nezmenil sa ani samotný mechanizmus kontroly, ktorý naďalej overuje iba striktnú zhodu hodnoty `"1"` pre atribúty `navigator.doNotTrack`, `window.doNotTrack` a `navigator.msDoNotTrack`, ako bolo zistené už v roku 2023 [56]. Táto implementácia je síce v súlade s hodnotou definovanou pre aktívny signál DNT v štandarde W3C [69], avšak je najmenej flexibilná. Nekontroluje numerickú hodnotu `1`, ktorú by niektoré implementácie mohli teoreticky použiť, a hlavne ignoruje akékoľvek prípadné rozšírenia za znakom `"1"` (`DNT-extension`), ktoré štandard technicky umožňuje (napr. hodnota `"1xyz"` by bola nesprávne vyhodnotená ako neaktívne DNT [56]), aj keď ich praktické využitie štandard neodporúča.

```
if ("1" !== navigator.doNotTrack && "1" !== window.doNotTrack &&
    "1" !== navigator.msDoNotTrack) {
}
```

Výpis 5.23: Kontrola hodnoty signálu DNT v skripte poskytovateľa Hotjar

Implementácia poskytovateľa Mouseflow (výpis 5.24) používa na kontrolu hodnoty DNT porovnanie `== 1`, čím je tolerantnejšou voči typu hodnoty signálu – rozpozná reťazec `"1"` a aj numerickú hodnotu `1`. Stále však, podobne ako Hotjar, ignoruje prípadné rozšírenia DNT signálu (`DNT-extension`), čím môže dôjsť k jeho nesprávnemu vyhodnoteniu (napr. hodnota `"1xyz"` by nebola rozpoznaná ako aktívne DNT). Zahrnutie kontroly na reťazec `'yes'` navyše nezodpovedá špecifikácii DNT v štandarde W3C [69].

```
_769: (navigator.doNotTrack === 'yes' || navigator.doNotTrack == 1 ||
       window.doNotTrack == 1 || navigator.msDoNotTrack == 1) ? 1 : 0
```

Výpis 5.24: Kontrola hodnoty signálu DNT v skripte poskytovateľa Mouseflow

Poslednú a najtolerantnejšiu kontrolu hodnoty signálu DNT obsahuje skript od poskytovateľa PostHog (výpis 5.25). Po overení, či je v konfigurácii zaznamenávacieho skriptu zapnuté rešpektovanie DNT (`this.config.respect_dnt`) skript iteruje cez možné atribúty DNT (pomocou funkcie `ie`). Pre nájdenú hodnotu (`e`) overí (pomocou funkcie `f`), či patrí do zoznamu `[true, 1, "1", 'yes']`. Implementácia teda akceptuje ako štandardnú hodnotu (`"1"`), tak aj numerickú hodnotu (`1`).

Podobne ako skripty od poskytovateľov Hotjar a Mouseflow, ani táto implementácia však nepodporuje rozšírenia signálu DNT — hodnota ako `"1xyz"` by nebola nájdená v zozname akceptovaných hodnôt a DNT by bolo vyhodnotené ako neaktívne. Zahrnutie boolean

hodnoty `true` a reťazca `'yes'` však prekračuje rámec špecifikácie signálu DNT [69]. Celkovo je teda kontrola v skripte pre zaznamenávanie sedenia poskytovateľa PostHog najviac tolerantná k rôznym možným (aj neštandardným) hodnotám signálu DNT.

```
getDnt() {
          return !!this.config.respect_dnt &&
              !!ie([null == o ? void 0 : o.doNotTrack,
                  null == o ? void 0 : o.msDoNotTrack, _.doNotTrack],
              (e => f([!0, 1, "1", "yes"], e)))
      }
```

Výpis 5.25: Kontrola hodnoty signálu DNT v skripte poskytovateľa PostHog

Na rozdiel od signálu DNT, ako ukazuje tabuľka 5.2, signál Global Privacy Control (GPC) počas môjho skúmania nerešpektoval žiadny z analyzovaných poskytovateľov. Ich zaznamenávacie skripty počas mojej analýzy neobsahovali logiku na detekciu ani reakciu na hodnotu `navigator.globalPrivacyControl`.

	Hotjar	Fullstory	Smartlook	Yandex Metrica	Mouseflow	PostHog
Signál GPC	✗	✗	✗	✗	✗	✗

Tabuľka 5.2: Rešpektovanie signálu GPC v zaznamenávacm skripte poskytovateľa

5.3 Analýza nasadenia skriptov na reálnych webových stránkach

V tejto sekcii prezentujem výsledky analýzy nasadenia a konfigurácie skriptov pre zaznamenávanie sedenia od analyzovaných poskytovateľov na vybraných, verejne dostupných, webových stránkach. Do výberu pre analýzu nasadenia sa dostali nasledujúce webové stránky, ktoré som vyhodnotil ako zaujímavé pre analýzu, keďže vzhľadom na ich charakter (ide napríklad o elektronické obchody, internetové bazáre, verejnú stránku banky či streamovaciu službu) majú potenciál obsahovať osobné údaje:

- alza.cz
- sbazar.cz
- eyerim.sk
- canyon.com
- slsp.sk
- autobazar.eu
- iprima.cz
- paulaschoice-eu.com

Metodológiu analýzy nasadenia som popísal v sekcii 4.3. Pri analýze som postupoval nasledovne:

1. Identifikoval som stránky obsahujúce skripty od analyzovaných poskytovateľov.
2. Overil som, či je na identifikovanej stránke aktívne zaznamenávanie dát pre prehrávanie sedení (podrobnosti v sekcii 5.3.1). Všetky stránky analyzované v rámci sekcie 5.3.3 majú tento proces aktívny.
3. Analyzoval som zdrojový kód stránky (HTML) s cieľom zistiť rozsah použitia manuálnych redakčných nástrojov (tabuľka 3.5).
4. Analyzoval som konfiguráciu samotného zaznamenávacieho skriptu (pomocou Chrome DevTools[1]) s cieľom zistiť nastavenia redakcie vykonané cez ovládací panel poskytovateľa (podrobnosti v sekcii 5.3.2).
5. Porovnal som zistenia z bodov 3 a 4 s cieľom určiť predpoklad celkového rozsahu redakcie obsahu stránky v procese zaznamenávania.

Výsledkom týchto krokov je môj odhad rozsahu dát zachytávaných zaznamenávacím skriptom na analyzovaných stránkach. Ide len o odhad, keďže presný rozsah zachytených dát nedokážem s istotou určiť len na základe výsledkov tejto analýzy.

5.3.1 Overenie aktivácie prehrávania sedenia

Pred samotnou analýzou rozsahu použitia redakčných nástrojov som musel overiť, či identifikovaný skript pre zaznamenávanie sedenia bol na danej stránke aj aktívny pre funkciu prehrávania sedení (sekcia 2.3.2). Prítomnosť snippetu, respektíve samotného zaznamenávacieho skriptu, totiž automaticky neznamená, že nahrávanie prebieha aj pre účely prehrávania sedení.

Počas výberu stránok pre túto analýzu som narazil aj na prípady, kedy bol skript prítomný, ale nahrávanie nebolo aktívne. Konkrétne išlo o poskytovateľa Mouseflow na stránkach moebelix.cz a xxxlutz.cz (obe stránky sa špecializujú na predaj nábytku a bytových doplnkov). Keďže na týchto stránkach nebol proces zaznamenávania sedení aktivovaný, nezahrnul som ich do finálnej analýzy implementácie.

Napriek tomu by som rád spomenul, ako si používateľ môže overiť aktiváciu skriptu od poskytovateľa Mouseflow. Ďalej, vzhľadom na to, že väčšina analyzovaných stránok, ktoré som nakoniec v sekcii 5.3.3 zahrnul do analýzy (a mali teda aktívne nahrávanie), obsahuje skript buď od poskytovateľa Hotjar alebo Smartlook, rozoberiem overenie aktivácie aj pre nich:

- **Hotjar**: Pri skripte od tohto poskytovateľa som musel preskúmať jeho konfiguračný súbor, ktorý som v Chrome DevTools našiel v záložke „Sources“, konkrétne v adresári `static.hotjar.com/c/`. V tomto súbore som hľadal premennú s názvom `record`. Ak bola jej hodnota nastavená na `true`, proces zaznamenávania pre účely prehrávania sedenia je aktivovaný.
- **Smartlook**: Stav nahrávania som zisťoval kontrolou konzoly prehliadača. Ak sa v konzole objaví nasledovná správa:

[1] https://developer.chrome.com/docs/devtools

```
[Smartlook] Smartlook not recording. Reason: Project is paused.
```

indikuje to, že nahrávanie je pozastavené. V opačnom prípade, ak táto správa chýba a v záložke „Sources“ (Chrome DevTools) je prítomný hlavný zaznamenávací skript `bundle.js` (nachádzajúci sa v adresári `web-sdk.smartlook.com/es6/`), nahrávanie je aktívne.

- **Mouseflow**: Aktivitu som zistil spustením príkazu `mouseflow.isRecording()`[1] v konzole prehliadača. Hodnota **`true`** znamená, že Mouseflow aktívne nahráva aktuálne sedenie. Hodnota **`false`** indikuje, že skript Mouseflow je na stránke prítomný, ale sedenie nezaznamenáva. Hodnota **`undefined`** znamená, že skript Mouseflow nie je na stránke inicializovaný alebo prítomný.

Táto verifikácia mi umožnila zamerať analýzu implementácie redakčných nástrojov primárne na tie stránky, kde bolo zaznamenávanie sedení s vysokou pravdepodobnosťou aktívne.

5.3.2 Analýza konfigurácie redakcie skriptu v ovládacom paneli

Kľúčom k odhadu skutočného rozsahu zachytávaných dát, v procese zaznamenávania sedenia, je analýza konfiguračných nastavení redakcie, nie len samotného rozsahu implementácie nástrojov pre manuálnu redakciu (tabuľka 3.5). Tieto nastavenia sú často definované v ovládacom paneli konkrétneho poskytovateľa.

Keďže žiadna zo stránok obsahujúcich skript od poskytovateľa Mouseflow nie je zahrnutá vo výsledkoch analýzy nasadenia z dôvodu neaktívneho zaznamenávania (ako bolo spomenuté v sekcii 5.3.1), konfiguráciu skriptu tohto poskytovateľa v tejto sekcii ďalej nerozoberám. Poskytovateľ Smartlook, podobne ako poskytovateľ Fullstory pri jeho funkcionalite *Element data capture rule* (obrázok 5.4), umožňuje definovať vlastné pravidlá redakcie cez svoj ovládací panel. Tieto pravidlá sa však neodzrkadľujú priamo v kóde zaznamenávacieho skriptu spôsobom, ktorý by umožňoval ich jednoduchú detekciu a analýzu konfigurácie.

Z týchto dôvodov som sa pri detailnejšom skúmaní konfiguračných možností zameral hlavne na analýzu skriptu od poskytovateľa Hotjar. Nastavenia redakcie vykonané v ovládacom paneli Hotjar sa totiž odrážajú v hodnotách špecifických premenných v konfiguračnom súbore skriptu (identifikovanom v sekcii 5.3.1). Tieto premenné priamo korelujú s možnosťami zobrazenými na obrázku 5.2. Atribúty, ktoré som sledoval, boli:

- `anonymize_digits`: Určuje, či majú byť maskované všetky číslice na stránke.
- `anonymize_emails`: Určuje, či majú byť maskované všetky e-mailové adresy na stránke.
- `recording_capture_keystrokes`: Určuje, či sa má zaznamenávať obsah polí explicitne povolených atribútom `data-hj-allow` (tabuľka 3.5). Hodnota `false` znamená, že povolenie záznamu cez `data-hj-allow` je ignorované.
- `suppress_text`: Určuje, či má byť maskovaný všetok viditeľný text na stránke, vrátane textu v elementoch povolených cez `data-hj-allow`.
- `suppress_all`: Určuje, či má byť maskovaný všetok viditeľný obsah (text aj obrázky), vrátane obsahu v elementoch povolených cez `data-hj-allow`.

[1] https://help.mouseflow.com/en/articles/9971958-verify-if-mouseflow-is-recording

- `suppress_all_on_specific_pages`: Obsahuje pole adries URL. Ak je aktuálna URL stránky zhodná s niektorou z adries v tomto poli, aplikuje sa rovnaké správanie ako pri `suppress_all=true` (maskuje sa všetok viditeľný obsah vrátane povolených polí), ale len pre tieto špecifické stránky.

5.3.3 Výsledky analýzy nasadenia podľa stránky

V tejto sekcii prezentujem výsledky mojej analýzy nasadenia skriptov pre zaznamenávanie sedenia, od analyzovaných poskytovateľov, na vybraných webových stránkach. Pre každú analyzovanú stránku uvediem jej stručný popis, skript od ktorého poskytovateľa obsahuje, mnou zistený rozsah nasadenia zaznamenávacieho skriptu, identifikovaný rozsah použitia manuálnych redakčných nástrojov implementovaných v kóde HTML stránky (podľa tabuľky 3.5), mnou získané nastavenia redakcie nakonfigurované cez ovládací panel poskytovateľa a záverečný odhad rozsahu zachytávaných dát.

alza.cz

- **Popis stránky:** Elektronický obchod so širokým sortimentom tovaru.
- **Poskytovateľ:** Hotjar
- **Rozsah nasadenia:** Skript som detegoval iba na stránke prihlasovacieho formulára[1].
- **Použitie atribútu `data-hj-suppress`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Použitie atribútu `data-hj-allow`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Nastavenia redakcie z ovládacieho panela (sekcia 5.3.2):**

```
"recording_capture_keystrokes": true,
"anonymize_digits": true,
"anonymize_emails": true,
"suppress_all": false,
"suppress_all_on_specific_pages": [],
"suppress_text": false
```

- **Odhadovaný rozsah redakcie a zachytených dát:** Vzhľadom na charakter analyzovanej stránky (prihlasovací formulár) a na základe výsledkov dynamickej analýzy (sekcia 5.1.2), kde som zistil, že skript poskytovateľa Hotjar pri základnej implementácii maskuje obsah všetkých vstupných polí, nepredpokladám že dochádza k zachytávaniu osobných údajov.

sbazar.cz

- **Popis stránky:** Online bazár umožňujúci používateľom pridávať, prezerať a reagovať na inzeráty.
- **Poskytovateľ:** Hotjar

[1] https://identity.alza.cz/Account/Login

- **Rozsah nasadenia:** Skript som detegoval na všetkých stránkach, ktoré som počas návštevy webovej lokality navštívil, vrátane prezerania inzerátov a profilu používateľa.
- **Použitie atribútu `data-hj-suppress`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Použitie atribútu `data-hj-allow`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Nastavenia redakcie z ovládacieho panela (sekcia 5.3.2):**

```
"recording_capture_keystrokes": true,
"anonymize_digits": true,
"anonymize_emails": true,
"suppress_all": false,
"suppress_all_on_specific_pages": [],
"suppress_text": false
```

- **Odhadovaný rozsah redakcie a zachytených dát:** Vzhľadom na konfiguráciu z ovládacieho panela (maskovanie zobrazovaných číslic, nezapnuté globálne potlačenie textu/obrázkov) a absenciu manuálnych redakčných atribútov v kóde predpokladám, že rozsah maskovania dát bude zodpovedať zisteniam zo základnej implementácie skriptu poskytovateľa Hotjar popísanej v sekcii 5.1.2. To znamená, že obsah všetkých vstupných polí bude maskovaný. Keďže však nie je aktívne žiadne potlačenie zobrazovaného obsahu (okrem číslic), je vysoko pravdepodobné, že v nahrávkach budú zachytené osobné údaje zobrazované na stránke, ako napríklad fotografia používateľa a jeho celé meno (zobrazované na profile), spolu s informáciami o jeho inzerátoch, ktoré konkrétne inzeráty si prehliada a na ktoré reaguje. Pri reakcii na inzerát cez funkcionalitu „Napísať predajcovi“ bude pravdepodobne zachytená samotná interakcia s touto funkcionalitou, avšak obsah odosielanej správy by mal byť maskovaný, keďže ide o vstup do vstupného poľa.

eyerim.sk

- **Popis stránky:** Elektronický obchod špecializovaný hlavne na predaj kontaktných šošoviek, dioptrických a slnečných okuliarov.
- **Poskytovateľ:** Hotjar
- **Rozsah nasadenia:** Skript som detegoval na všetkých stránkach, ktoré som počas návštevy webovej lokality navštívil, vrátane procesu výberu a konfigurácie dioptrických okuliarov[1].
- **Použitie atribútu `data-hj-suppress`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Použitie atribútu `data-hj-allow`:** Tento atribút som na analyzovaných stránkach nenašiel.

[1] https://eyerim.sk/lens-configurator/

- **Nastavenia redakcie z ovládacieho panela (sekcia 5.3.2):**

```
"recording_capture_keystrokes": true,
"anonymize_digits": true,
"anonymize_emails": true,
"suppress_all": false,
"suppress_all_on_specific_pages": [],
"suppress_text": false
```

- **Odhadovaný rozsah redakcie a zachytených dát:** Vzhľadom na konfiguráciu z ovládacieho panela (maskovanie číslic, nezapnuté globálne potlačenie textu/obrázkov) a absenciu manuálnych redakčných atribútov predpokladám, že rozsah maskovania dát bude zodpovedať zisteniam zo základnej implementácie skriptu Hotjar (sekcia 5.1.2). Obsah všetkých vstupných polí by mal byť teda maskovaný. Najcitlivejšou informáciou, ktorá by mohla byť potenciálne v procese zaznamenávania zachytená, sú hodnoty dioptrií zvolené používateľom počas procesu konfigurácie okuliarov. Aj keď sú číselné hodnoty dioptrií v ponuke pravdepodobne maskované vďaka nastaveniu `anonymize_digits`, samotný výber dioptrie je implementovaný ako rozbaľovacie menu (drop-down), nie ako textové vstupné pole. Používateľ si hodnotu vyberá kliknutím, nezadáva ju z klávesnice. Vzhľadom na statický charakter položiek v tomto menu existuje riziko nepriameho úniku informácie o zvolenej dioptrii: aj keď samotné číslo v menu môže byť v nahrávke maskované, zaznamenanie kliknutia na konkrétnu položku (napr. tretiu v poradí) by mohlo umožniť spätné zistenie zvolenej hodnoty. Ďalej, ak je používateľ prihlásený do svojho účtu, jeho krstné meno sa zobrazuje v pravom hornom rohu stránky. Keďže nie je aktívne potlačenie textu, predpokladám, že toto meno bude zachytené.

- **Poznámka:** Spoločnosť eyerim má zverejnené pravidlá zodpovedného zverejňovania bezpečnostných zraniteľností [17]. Tieto pravidlá sa vzťahujú na systémy vlastnené, prevádzkované alebo udržiavané spoločnosťou eyerim a zároveň vylučuje z rozsahu účinnosti služby tretích strán („Out of scope: Third-party services“). Keďže moja analýza sa zameriava na implementáciu a konfiguráciu služby tretej strany (skriptu pre zaznamenávanie sedenia od poskytovateľa Hotjar), a nie na priame testovanie bezpečnosti systémov eyerim, domnievam sa, že moja činnosť nespadá pod obmedzenia týchto pravidiel[1].

canyon.com

- **Popis stránky:** Oficiálny internetový obchod výrobcu bicyklov Canyon, ktorý predáva svoje produkty primárne priamo koncovým zákazníkom (direct-to-consumer) online.

- **Poskytovateľ:** Hotjar

- **Rozsah nasadenia:** Skript som detegoval na všetkých stránkach, ktoré som počas návštevy webovej lokality navštívil, vrátane produktových stránok, používateľského profilu a procesu objednávky.

[1] Konfigurácia skriptu poskytovateľa Hotjar na stránke eyerim.sk je zodpovednosťou spoločnosti eyerim, pre ktorú Hotjar pôsobí ako spracovateľ údajov (teda nie je v tomto kontexte úplne „treťou stranou“), čiže aplikovateľnosť týchto pravidiel na analýzu konfigurácie externej služby je otázna.

- **Použitie atribútu `data-hj-suppress`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Použitie atribútu `data-hj-allow`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Nastavenia redakcie z ovládacieho panela (sekcia 5.3.2):**

```
"recording_capture_keystrokes": false,
"anonymize_digits": true,
"anonymize_emails": true,
"suppress_all": false,
"suppress_all_on_specific_pages": [],
"suppress_text": false
```

- **Odhadovaný rozsah redakcie a zachytených dát:** Vzhľadom na konfiguráciu z ovládacieho panela (maskovanie zobrazovaných číslic, nezapnuté globálne potlačenie textu/obrázkov) a absenciu manuálnych redakčných atribútov predpokladám, že rozsah maskovania dát bude zodpovedať zisteniam zo základnej implementácie skriptu Hotjar (sekcia 5.1.2). Obsah všetkých vstupných polí by mal byť teda maskovaný. Potenciálne môžu byť zaznamenané osobné údaje zobrazované na stránke používateľského profilu, ako napríklad dodacie údaje (adresa). Hoci číselné časti adresy (číslo domu, PSČ) by mali byť maskované vďaka nastaveniu `anonymize_digits`, názvy ulice, mesta a krajiny pravdepodobne zachytené budú.

slsp.sk

- **Popis stránky:** Verejné webové stránky Slovenskej sporiteľne.
- **Poskytovateľ:** Hotjar
- **Rozsah nasadenia:** Skript som detegoval na všetkých verejne dostupných stránkach, ktoré som počas analýzy navštívil, vrátane stránok s refinančnou kalkulačkou[1] a hypotekárnou kalkulačkou[2]. V momente, keď som prešiel na prihlasovací formulár do sekcie internet bankingu, skript už prítomný nebol.
- **Použitie atribútu `data-hj-suppress`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Použitie atribútu `data-hj-allow`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Nastavenia redakcie z ovládacieho panela (sekcia 5.3.2):**

```
"recording_capture_keystrokes": true,
"anonymize_digits": true,
"anonymize_emails": true,
"suppress_all": false,
"suppress_all_on_specific_pages": [],
"suppress_text": false
```

[1] https://www.slsp.sk/sk/kalkulacky/refinancna-kalkulacka
[2] https://www.slsp.sk/sk/kalkulacky/kalkulacka-uveru-na-byvanie

- **Odhadovaný rozsah redakcie a zachytených dát:** Na stránkach ako je refinančná alebo hypotekárna kalkulačka je väčšina polí implementovaná ako vstupné polia. Vzhľadom na to, že neboli použité atribúty `data-hj-allow` ani `data-hj-suppress`, a na základe výsledkov dynamickej analýzy základnej implementácie skriptu poskytovateľa Hotjar (sekcia 5.1.2), predpokladám, že obsah týchto vstupných polí je štandardne maskovaný. Ďalej, všetky zobrazované číselné údaje v týchto kalkulačkách (napr. výška úveru, splátky) sú maskované vďaka nastaveniu `anonymize_digits: true`. Aj keď niektoré voľby v kalkulačkách sú implementované ako rozbaľovacie menu (drop-down, napr. počet detí, vek), a samotné číselné hodnoty týchto volieb by mali byť maskované, existuje riziko nepriameho zachytenia týchto informácií skrz zachytené kliknutie používateľa na konkrétnu položku v tomto menu. Keďže skript nie je prítomný na prihlasovacej stránke do internet bankingu, riziko zachytenia prihlasovacích údajov je eliminované.

autobazar.eu

- **Popis stránky:** Internetový bazár s autami a príslušenstvom k nim.
- **Poskytovateľ:** Hotjar
- **Rozsah nasadenia:** Skript som detegoval na všetkých stránkach, ktoré som počas návštevy navštívil, vrátane jednotlivých inzerátov, profilu používateľa a aj v celom procese vytvárania inzerátu.
- **Použitie atribútu `data-hj-suppress`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Použitie atribútu `data-hj-allow`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Nastavenia redakcie z ovládacieho panela (sekcia 5.3.2):**

```
"recording_capture_keystrokes": false,
"anonymize_digits": true,
"anonymize_emails": true,
"suppress_all": false,
"suppress_all_on_specific_pages": [],
"suppress_text": false
```

- **Odhadovaný rozsah redakcie a zachytených dát:** Zobrazované číselné údaje na stránke sú maskované vďaka nastaveniu `anonymize_digits: true`. Po prihlásení používateľa do jeho účtu sa po celú dobu jeho pobytu na stránke zobrazuje jeho používateľské meno v pravom hornom rohu stránky. Keďže nastavenia `suppress_text` a aj `suppress_all` majú hodnotu `false` a tento element neobsahuje ani maskovací atribút `data-hj-suppress`, táto informácia je potenciálne zachytená. Podobne, v sekcii „Moje konto“ je zobrazované meno a priezvisko používateľa, ktoré je tiež potenciálne zachytené. V sekcii „Nastavenia konta“, kde si používateľ môže upraviť informácie o svojom účte, je väčšina úprav vykonávaná pomocou vstupných polí. Obsah týchto vstupných polí je v základnej implementácii skriptu Hotjar maskovaný, ako je popísané v sekcii 5.1.2. Avšak, výber krajiny, kraja, okresu a mesta je realizovaný pomocou rozbaľovacích menu (drop-down). Keďže tieto menu obsahujú textové názvy lokalít,

samotné názvy týchto lokalít sú potencionálne zachytené v nahrávke, a zvolenú lokalitu je možné odvodiť z pozície kliknutia používateľa.

iprima.cz

- **Popis stránky:** Streamovacia služba televíznej stanice TV Prima.
- **Poskytovateľ:** Smartlook
- **Rozsah nasadenia:** Skript som detegoval na všetkých navštívených stránkach, vrátane domovskej stránky s odporúčaniami na filmy, pri samotnom sledovaní relácie a aj v sekcii s nastaveniami o účte.
- **Použitie Record API:** Použitie tohoto rozhrania na manuálnu konfiguráciu redakcie (popísané v tabuľke 3.5) som nedetegoval na žiadnej z analyzovaných stránok.
- **Odhadovaný rozsah redakcie a zachytených dát:** V sekcii, kde si používateľ upravuje údaje o účte, sú tieto informácie (ako e-mail, meno, priezvisko alebo rok narodenia) najprv zobrazené a až po voľbe úpravy sa menia na vstupné polia. Na základe zistení zo základnej implementácie skriptu Smartlook (sekcia 5.1.2), obsah týchto vstupných polí je s najväčšou pravdepodobnosťou v nahrávke maskovaný. Avšak, prvotne zobrazované informácie o účte sú s najväčšou pravdepodobnosťou zachytávané. Keďže skript poskytovateľa Smartlook v základnej implementácii maskuje aj čísla, číselné údaje z týchto zobrazených informácií by mali byť maskované. Medzi potenciálne zachytené údaje (pokiaľ nie sú špecificky maskované pravidlami v ovládacom paneli, ako je popísané v sekcii 5.3.2) by tak mohli patriť: meno a priezvisko, pohlavie, ulica a mesto pobytu, dosiahnuté vzdelanie a profesné postavenie. Je však dôležité zdôrazniť, že Smartlook umožňuje v ovládacom paneli definovať vlastné pravidlá pre maskovanie obsahu, takže všetok tento zobrazený textový obsah môže byť v skutočnosti maskovaný.

paulaschoice-eu.com

- **Popis stránky:** Internetový obchod s produktami na starostlivosť o pleť.
- **Poskytovateľ:** Hotjar
- **Rozsah nasadenia:** Skript som detegoval na všetkých stránkach, ktoré som počas analýzy navštívil, vrátane profilu používateľa, jednotlivých produktových stránok a aj v nákupnom košíku.
- **Použitie atribútu `data-hj-suppress`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Použitie atribútu `data-hj-allow`:** Tento atribút som na analyzovaných stránkach nenašiel.
- **Nastavenia redakcie z ovládacieho panela (sekcia 5.3.2):**

```
"recording_capture_keystrokes": false,
"anonymize_digits": false,
"anonymize_emails": false,
```

```
"suppress_all": false,
"suppress_all_on_specific_pages": [],
"suppress_text": null
```

- **Odhadovaný rozsah redakcie a zachytených dát:** Zaujímavým zistením je nastavenie `anonymize_digits: false`, čo znamená, že na rozdiel od ostatných analyzovaných nasadení v rámci tejto sekcie, zobrazované číselné údaje na tejto stránke by nemali byť v záznamoch maskované. Používateľ si na svojom profile môže uložiť osobné údaje, ako celú adresu, meno a telefónne číslo, čo mu uľahčuje vypĺňanie údajov pri budúcich objednávkach. Tieto uložené informácie sú potom viditeľné priamo na profile používateľa. Vzhľadom na nastavenia `suppress_text: null`, `anonymize_digits: false`, `suppress_all: false` a absenciu redakčného atribútu (`data-hj-suppress`), predpokladám, že všetky tieto zobrazené osobné údaje, vrátane telefónneho čísla a číselných častí adresy, sú v procese zaznamenávania zachytené. Vstupné polia pri zadávaní alebo úprave týchto údajov by mali byť maskované vďaka správaniu základnej implementácie skriptu poskytovateľa Hotjar (sekcia 5.1.2), avšak samotné zobrazené údaje na profile sú vysoko pravdepodobne zaznamenávané v plnom rozsahu.

Kapitola 6

Záver

Výsledkom tejto práce je analýza skriptov pre zaznamenávanie sedenia od šiestich vybraných poskytovateľov, konkrétne Hotjar, Fullstory, Smartlook, Yandex Metrica, Mouseflow a PostHog. Výsledky dynamickej analýzy skriptov od týchto poskytovateľov ukázali, ako rozdielny prístup k automatickej redakcii, tak aj rôznu reakciu na povolenie zaznamenávania obsahu pomocou manuálnych redakčných nástrojov.

Väčšina analyzovaných skriptov v základnej implementácii maskuje obsah všetkých vstupných polí, zatiaľ čo napríklad Yandex Metrica zachytáva obsah niektorých vstupných polí už pri tejto implementácii, teda bez použitia dodatočných redakčných nástrojov. Niektoré skripty, menovite Hotjar a Smartlook, dokonca maskujú aj časti (aj keď nie rozsiahle) zobrazovaného obsahu. Reakcia na nástroje určené pre povolenie záznamu obsahu stránky je taktiež rôzna: zatiaľ čo niektoré skripty (Yandex Metrica, PostHog) umožňujú po aplikácii týchto nástrojov zachytiť kompletný obsah stránky, ostatné skripty naďalej nepovoľujú zachytenie obsahu určitých typov vstupných polí, aj keď je na nich aplikovaný povoľovací nástroj.

Statická analýza kódu potvrdila, že zaznamenávacie skripty je možné upraviť natoľko, že proces maskovania obsahu nebudú vykonávať vôbec. Analýza zdrojového kódu ďalej priniesla zistenie, že zatiaľ čo signál DNT (Do Not Track) rešpektuje polovica z analyzovaných poskytovateľov (konkrétne Hotjar, Mouseflow a PostHog), signál GPC (Global Privacy Control) v čase analýzy nerešpektoval žiadny z analyzovaných poskytovateľov. Analýza nasadenia týchto skriptov na reálnych stránkach, vykonaná síce na obmedzenej vzorke stránok, poukázala na rôzne spôsoby, akými môžu byť osobné údaje v procese zaznamenávania sedenia zachytené práve v dôsledku nedostatočne implementovanej redakcie obsahu stránky.

Potenciálne pokračovanie práce na tému analýzy skriptov pre zaznamenávanie sedenia by sa mohlo potenciálne uberať niekoľkými smermi. Mohla by sa vykonať podobná analýza aj pre ďalších poskytovateľov týchto služieb, ako je napríklad Microsoft Clarity. Zároveň by bolo vhodné sa po určitom časovom odstupe vrátiť k poskytovateľom analyzovaným v rámci tejto práce a preskúmať, ako sa ich prístup k redakcii od tejto analýzy zmenil.

Ďalej by mohol byť uskutočnený plošnejší prieskum konfigurácie zaznamenávacích skriptov a rozsahu použitia ich redakčných nástrojov na reálnych webových stránkach, čo by mohlo načrtnúť širší pohľad na reálny stav ochrany súkromia používateľa v tejto oblasti. Taktiež by bolo vhodné v budúcnosti rozšíriť testovacie scenáre o implementáciu elementov HTML typov ako `<img>`, `<video>` alebo `<iframe>`, aby sa experimentálne overilo, ako je obsah týchto elementov vo výslednej nahrávke spracovaný. V neposlednom rade by bolo užitočné detailnejšie otestovať spracovanie rôznych zobrazovaných typov osobných údajov

(ako sú napríklad adresa, meno, e-mail) na stránke, aby bolo možné detegovať ich prípadné maskovanie.

Literatúra

[1] Acar, G.; Englehardt, S. a Narayanan, A. No boundaries: data exfiltration by third parties embedded on web pages. In: *Proceedings of the 20th Privacy Enhancing Technologies Symposium (PETS)*. Sciendo, Júl 2020.

[2] Alatawi, M. S. M. *On the Security of End-to-End Encrypted Messaging and Calling Applications*. 2024. Dizertačná práca. Texas A&M University.

[3] Apple. *Safari White Paper – November 2019* online. Dostupné z: https://www.apple.com/safari/docs/Safari_White_Paper_Nov_2019.pdf. [cit. 2024-10-21].

[4] Ayanso, A. a Yoogalingam, R. Profiling retail web site functionalities and conversion rates: A cluster analysis. *International Journal of Electronic Commerce*. Taylor & Francis, 2009, zv. 14, č. 1, s. 79–114.

[5] Balebako, R.; Leon, P.; Shay, R.; Ur, B.; Wang, Y. et al. Measuring the effectiveness of privacy tools for limiting behavioral advertising. In: Web. 2012.

[6] Brave. *Filter List Glossary* online. Dostupné z: https://brave.com/glossary/filter-list/. [cit. 2024-10-21].

[7] BuiltWith. *Analytics Usage Distribution in the Top 1 Million Sites* online. Dostupné z: https://trends.builtwith.com/analytics. [cit. 2024-11-29].

[8] Cloudflare. *Why minify JavaScript code?* online. Dostupné z: https://www.cloudflare.com/learning/performance/why-minify-javascript-code/. [cit. 2025-04-27].

[9] Daniel, E.; Wilson, H. a Myers, A. Adoption of e-commerce by SMEs in the UK: towards a stage model. *International small business journal*. SAGE Publications Ltd 6 Bonhill Street, London EC2A 4PU, UK., 2002, zv. 20, č. 3, s. 253–270.

[10] Di Fatta, D.; Patton, D. a Viglia, G. The determinants of conversion rates in SME e-commerce websites. *Journal of Retailing and Consumer Services*, Marec 2018, zv. 41.

[11] Drew, S. Strategic uses of e-commerce by SMEs in the east of England. *European management journal*. Elsevier, 2003, zv. 21, č. 1, s. 79–88.

[12] Endler, D. Brute-Force Exploitation of Web Application Session IDs. *IDEFENSE Inc.*, December 2001, s. 1–40.

[13] Englehardt, S.; Acar, G. a Narayanan, A. *No Boundaries: Exfiltration of Personal Data by Session-Replay Scripts* Freedom to Tinker Blog Post. 15. november 2017. [cit. 2024-09-20].

[14] European Data Protection Board. *Data controller and data processor* online. Dostupné z: https://www.edpb.europa.eu/sme-data-protection-guide/data-controller-data-processor_en. [cit. 2024-11-27].

[15] European Data Protection Board. *Opinion 05/2014 on Anonymisation Techniques* online. Dostupné z: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf. [cit. 2024-10-20].

[16] European Parliament and Council of the European Union. *The General Data Protection Regulation (GDPR, 2016/679).* Dostupné z: https://data.europa.eu/eli/reg/2016/679/oj.

[17] eyerim. *Vulnerability Disclosure Policy* online. Dostupné z: https://eyerim.sk/researchers-security-policy/. [cit. 2025-05-07].

[18] Fielding, R. T.; Gettys, J.; Mogul, J. C.; Nielsen, H. F.; Masinter, L. et al. *Hypertext Transfer Protocol – HTTP/1.1.* Request for Comments 2616. Internet Engineering Task Force, June 1999. Dostupné z: https://www.rfc-editor.org/rfc/rfc2616.txt.

[19] Filip, P. a Čegan, L. Comparing Tools for Web-session Recording and Replaying. In: *2019 International Conference on Sustainable Information Engineering and Technology (SIET).* 2019, s. 257–260.

[20] Ford Motor Company. *US Privacy Notice* online. Dostupné z: https://www.ford.com/help/privacy/#USprivacynotice. [cit. 2024-04-27].

[21] FullStory. *Heatmaps: What They Are and How to Use Them* online. Dostupné z: https://www.fullstory.com/blog/heatmap/. [cit. 2024-11-14].

[22] FullStory. *Identifying users* online. Dostupné z: https://help.fullstory.com/hc/en-us/articles/360020828113-Identifying-users. [cit. 2025-03-10].

[23] FullStory. *Installing the Fullstory Script* online. Dostupné z: https://help.fullstory.com/hc/en-us/articles/360020623514-Installing-the-Fullstory-Script. [cit. 2025-04-20].

[24] FullStory. *Session Replay* online. Dostupné z: https://www.fullstory.com/blog/session-replay/. [cit. 2024-10-20].

[25] FullStory. *Session Replay Platform* online. Dostupné z: https://www.fullstory.com/platform/. [cit. 2024-10-15].

[26] Fullstory. *Trust Center* online. Dostupné z: https://trust.fullstory.com/. [cit. 2024-11-28].

[27] Gallo, A. *A Refresher on A/B Testing* online. Dostupné z: https://hbr.org/2017/06/a-refresher-on-ab-testing. [cit. 2024-11-21].

[28] Grodzinsky, F.; Miller, K. a Wolf, M. Session replay scripts: A privacy analysis. *The Information Society*, Jún 2022, zv. 38, s. 1–12.

[29] Hill, R. *UBlock - GitHub Repository* online. Dostupné z: https://github.com/gorhill/uBlock. [cit. 2024-10-21].

[30] Hotjar. *Compliance at Hotjar* online. Dostupné z: https://help.hotjar.com/hc/en-us/articles/360045447214-Compliance-at-Hotjar. [cit. 2024-11-28].

[31] Hotjar. *Fast-track your research with Hotjar AI* online. Dostupné z: https://www.hotjar.com/product-ai-surveys/. [cit. 2024-11-29].

[32] Hotjar. *How to Suppress Text, Images, Videos and User Input from Collected Data* online. Dostupné z: https://help.hotjar.com/hc/en-us/articles/115012439167-How-to-Suppress-Text-Images-Videos-and-User-Input-from-Collected-Data. [cit. 2024-11-21].

[33] Hotjar. *Website Heatmaps & Behavior Analytics Tools* online. Dostupné z: https://www.hotjar.com/. [cit. 2024-10-15].

[34] Huang, J.; White, R. a Buscher, G. User see, user point: gaze and cursor alignment in web search. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2012, s. 1341–1350. CHI '12. ISBN 9781450310154.

[35] ISO. *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. International Standard ISO/IEC 7498-1:1994. Geneva, Switzerland: ISO/IEC, 1994.

[36] Jelassi, T. a Leenen, S. An E-Commerce Sales Model for Manufacturing Companies:: A Conceptual Framework and a European Example. *European Management Journal*. Elsevier, 2003, zv. 21, č. 1, s. 38–47.

[37] Kaspersky. *Data Leakage: How to Prevent Data Breaches?* online. Dostupné z: https://www.kaspersky.com/resource-center/definitions/data-leakage. [cit. 2024-11-29].

[38] Kaur, K. a Singh, H. Click analytics: What clicks on webpage indicates? In: IEEE. *2016 2nd international conference on next generation computing technologies (NGCT)*. 2016, s. 608–614.

[39] Klyne, G. a Newman, C. *Date and time on the internet: Timestamps*. 2002.

[40] Kontaxis, G. a Chew, M. Tracking protection in firefox for privacy and performance. *ArXiv preprint arXiv:1506.04104*, 2015.

[41] Lah, F. Are IP addresses personally identifiable information. *ISJLP*. HeinOnline, 2008, zv. 4, s. 681.

[42] McCrea, K. *The Yandex Leak: How a Russian Search Giant Uses Consumer Data* online. Dostupné z: https://www.blackhat.com/us-23/briefings/schedule/#the-yandex-leak-how-a-russian-search-giant-uses-consumer-data-33301. [cit. 2024-11-29].

[43] Merzdovnik, G.; Donko Huber, M.; Buhov, D.; Nikiforakis, N.; Neuner, S. et al. Block Me If You Can: A Large-Scale Study of Tracker-Blocking Tools. In:. Apríl 2017.

[44] Microsoft. *Retrace your steps with Recall* online. Dostupné z: https://support.microsoft.com/en-us/windows/retrace-your-steps-with-recall-aa03f8a0-a78b-4b3e-b0a1-2eb8ac48701c. [cit. 2024-04-27].

[45] Moore, P. *HTTP State Management Mechanism*. Request for Comments 2965. Internet Engineering Task Force, October 2000. Dostupné z: https://www.rfc-editor.org/rfc/rfc2965.txt.

[46] Mouseflow. *Feature Overview* online. Dostupné z: https://mouseflow.com/features/. [cit. 2024-11-27].

[47] Mouseflow. *GDPR (EEA and UK)* online. Dostupné z: https://mouseflow.com/legal/gdpr/. [cit. 2024-11-28].

[48] Mouseflow. *How Session Replays Work* online. Dostupné z: https://mouseflow.com/blog/how-session-replays-work/. [cit. 2024-10-14].

[49] Mozilla. *DNT* online. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/DNT. [cit. 2024-10-16].

[50] Mozilla. *How do I turn on the Do Not Track feature?* online. Dostupné z: https://support.mozilla.org/en-US/kb/how-do-i-turn-do-not-track-feature. [cit. 2025-01-01].

[51] Mozilla. *JavaScript* online. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript. [cit. 2024-10-13].

[52] Mozilla. *Sec-GPC* online. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Sec-GPC#directives. [cit. 2025-03-11].

[53] nkuma. *UnmaskPassword* online. Dostupné z: https://github.com/nkuma/UnmaskPassword. [cit. 2024-10-20].

[54] OWASP. *Manipulator in the Middle Attack* online. Dostupné z: https://owasp.org/www-community/attacks/Manipulator-in-the-middle_attack#:~:text=The%20MITM%20attack%20is%20very,also%20in%20the%20data%20transferred. [cit. 2024-10-15].

[55] Pan, R. a Ruiz Martínez, A. Evolution of Web Tracking Protection in Chrome. *Journal of Information Security and Applications*, 2023, zv. 79, s. 103643. ISSN 2214-2126.

[56] Polčák, L. a Slezáková, A. Data Exfiltration by Hotjar Revisited. In: *Proceedings of the 19th International Conference on Web Information Systems and Technologies*. SciTePress - Science and Technology Publications, 2023, s. 347–354. ISBN 978-989-758-672-9.

[57] PostHog. *How developers build successful products* online. Dostupné z: https://posthog.com/. [cit. 2024-11-27].

[58] PostHog. *Replay data retention* online. Dostupné z: https://posthog.com/docs/session-replay/data-retention. [cit. 2024-11-29].

[59] PostHog. *Security and Privacy* online. Dostupné z: https://posthog.com/handbook/company/security. [cit. 2024-11-27].

[60] Regional Court of Berlin. *16 O 420/19* online. Dostupné z: https://gdprhub.eu/index.php?title=LG_Berlin_-_16_O_420/19#Facts. [cit. 2024-10-21].

[61] Rácz, E. *7 Effective Ways to Optimize Your E-commerce Conversion Funnel* online. Dostupné z: https://www.ppchero.com/7-effective-ways-to-optimize-your-e-commerce-conversion-funnel/. [cit. 2024-11-14].

[62] Senol, A.; Acar, G.; Humbert, M. a Borgesius, F. Z. Leaky Forms: a study of email and password exfiltration before form submission. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022, s. 1813–1830.

[63] Skolka, P.; Staicu, C.-A. a Pradel, M. Anything to Hide? Studying Minified and Obfuscated Code in the Web. In: *The World Wide Web Conference*. New York, NY, USA: Association for Computing Machinery, 2019, s. 1735–1746. WWW '19. ISBN 9781450366748. Dostupné z: https://doi.org/10.1145/3308558.3313752.

[64] Smartlook. *Data Processing Agreement* online. Dostupné z: https://www.smartlook.com/dpa. [cit. 2024-11-28].

[65] Smartlook. *Getting started with Smartlook* online. Dostupné z: https://help.smartlook.com/docs/getting-started. [cit. 2024-11-27].

[66] Soltani, A. et al. *Global Privacy Control* online. Dostupné z: https://globalprivacycontrol.org/. [cit. 2024-10-21].

[67] Statcounter. *Browser market share worldwide* online. Dostupné z: https://gs.statcounter.com/. [cit. 2024-10-21].

[68] The World Wide Web Consortium (W3C). *Extensible Markup Language (XML)* online. Dostupné z: https://www.w3.org/XML/. [cit. 2024-10-14].

[69] The World Wide Web Consortium (W3C). *Tracking Preference Expression (DNT)* online. 2019. Dostupné z: https://www.w3.org/TR/tracking-dnt/. [cit. 2025-04-20].

[70] W3Techs. *Usage of HTTPS for Websites* online. Dostupné z: https://w3techs.com/technologies/details/ce-httpsdefault. [cit. 2024-10-15].

[71] WHATWG. *DOM Standard* online. Dostupné z: https://dom.spec.whatwg.org/. [cit. 2024-09-28].

[72] WHATWG. *HTML Living Standard* online. 2024. Dostupné z: https://html.spec.whatwg.org/multipage/. [cit. 2024-10-13].

[73] Yandex. *Data transfer restrictions* online. Dostupné z: https://yandex.com/support/metrica/general/limits.html. [cit. 2024-12-30].

[74] Yandex. *GDPR compliance* online. Dostupné z: https://yandex.com/support/metrica/general/gdpr.html. [cit. 2024-11-28].

[75] Yandex. *Yandex Metrica* online. Dostupné z: https://yandex.com/support/direct/technologies-and-services/metrika-in-direct.html. [cit. 2024-11-27].