

Web Tracking Forensics: Detecting and Analyzing How Cookies and Scripts Track Users Across the Web

Biškup Dorian
Babić Lovro
Blažek Ilan
Puklek Dino

Cilj projekta

Nadzire mrežni promet
pregledavanja kako bi se otkrilo
praćenje temeljeno na kolačićima i
praćenje bez kolačića.



Bilježi i klasificira metode
praćenja, uključujući kolačiće
trećih strana, beacone i pokušaje
fingerprintinga.

Pohranjuje prikupljene
dokaze u bazu podataka radi
naknadne analize

Vizualizira veze između web
stranica i trackera, odnosno
prikazuje koje treće strane imaju
uvid u korisnikovo pregledavanje.

Podjela rada:



Bavio se snimanjem mrežnog prometa web stranica pomoću alata za presretanje HTTP/HTTPS zahtjeva te identifikacijom domena koje primaju podatke o korisnicima.

Ilijan Blažek



Zadužen za detekciju i klasifikaciju trackera, razlikovanje first-party i third-party praćenja te procjenu intenziteta praćenja po web stranici

Biškup Dorian



Radio je na dizajnu baze podataka i backend skriptama za pohranu i obradu zapisa o praćenju, uključujući kolačiće i mrežne zahtjeve.

Dino Puklek



Izradio je grafičku vizualizaciju odnosa između web stranica i trackera te analizirao rezultate za završno izvještavanje i prezentaciju.

Babić Lovro

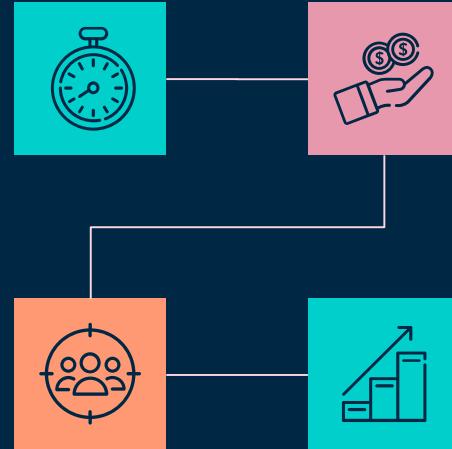
Programi korišteni u sklopu projekta:

MITMProxy

Snimanje HTTP/HTTPS
prometa
Presretanje zahtjeva,
kolačića i trackera

SQLITE

Pohrana podataka o praćenju
Baza zahtjeva, domena i
kategorija



PYTHON

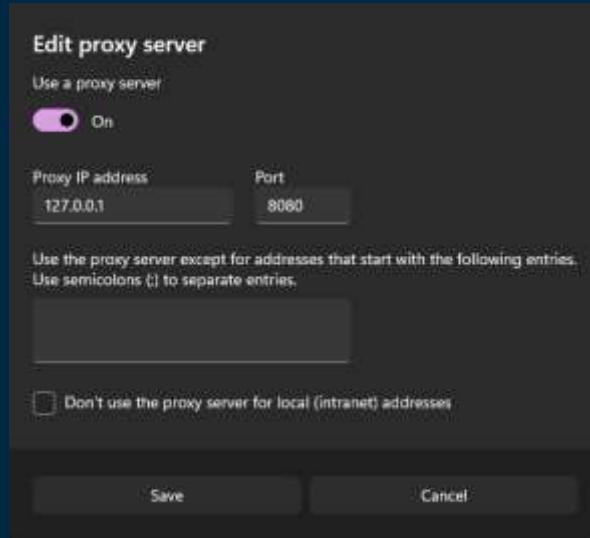
Obrada i analiza podataka
Klasifikacija trackera i izvoz
podataka

GEPHI

Vizualizacija mreže trackera
Graf odnosa web stranica i
third-party domena

Postavljanje radnog okruženja

```
pip install mitmproxy
```



Skripta za snimanje HTTP/HTTPS prometa

```
1  from mitmproxy import http
2  import json
3
4  def request(flow: http.HTTPFlow):
5      referer = flow.request.headers.get("referer", "")
6      site = referer.split("/")[2] if "://" in referer else ""
7
8      entry = {
9          "visited_site": site,
10         "request_domain": flow.request.host,
11         "url": flow.request.pretty_url,
12         "method": flow.request.method,
13         "cookies": dict(flow.request.cookies),
14         "user_agent": flow.request.headers.get("user-agent", "")
15     }
16
17     with open("traffic_log.json", "a", encoding="utf-8") as f:
18         f.write(json.dumps(entry) + "\n")
```

Tracker_logger.py

Sprema sirove podatke o prometu (URL, domena, cookies, headers...)

Presreće HTTP/HTTPS zahtjeve

Radi s **mitmproxyjem**

Skripta za čišćenje i pripremu logova

```
1 import json
2
3 INPUT_FILE = "traffic_log.json"
4 OUTPUT_FILE = "traffic_log_sanitized.json"
5
6 def sanitize_entry(entry):
7     sanitized = {
8         "visited_site": entry.get("visited_site", ""),
9         "request_domain": entry.get("request_domain", ""),
10        "url": entry.get("url", "").split("/")[-1], # uklanja every string
11        "method": entry.get("method", ""),
12        "cookies": list(entry.get("cookies", {}).keys()),
13        "user_agent": entry.get("user_agent", "")
14    }
15
16    sanitized["is_third_party"] = {
17        sanitized["visited_site"] != "" and
18        sanitized["visited_site"] != sanitized["request_domain"]
19    }
20
21    return sanitized
22
23
24 with open(INPUT_FILE, "r", encoding="utf-8") as infile, \
25     open(OUTPUT_FILE, "w", encoding="utf-8") as outfile:
26
27     for line in infile:
28         entry = json.loads(line)
29         clean = sanitize_entry(entry)
30         outfile.write(json.dumps(clean) + "\n")
31
32 print("Sanitized log created:", OUTPUT_FILE)
```

sanitize_log.py

Čisti sirove logove

Uklanja nepotrebne ili
osjetljive dijelove

Priprema podatke za
daljnju obradu

Dobiveni rezultati



A screenshot of a JSON file named "traffic_log.json". The file contains a large array of objects, each representing a single network request. The objects have properties like "method", "url", "headers", and "body". The data is presented in a monospaced font, with many lines of text representing different parts of the request and response bodies.

traffic_log.json

Sadrži sirove podatke snimljenog mrežnog prometa – sve HTTP/HTTPS zahtjeve, dugačke URL-ove, kolačice i tehničke detalje koji nisu odmah pogodni za analizu.



A screenshot of a JSON file named "traffic_log_sanitized.json". This file is similar in structure to the first one but contains significantly less data. It only includes the essential fields for analysis, such as "method", "url", and "body", while omitting sensitive information like full URLs and detailed headers.

traffic_log_sanitized.json

Sadrži očišćene i strukturirane podatke – uklonjeni su nepotrebni i osjetljivi dijelovi, a zadržane su samo informacije važne za analizu web praćenja.

Kreiranje baze podataka

```
1 import json
2 import sqlite3
3
4 DB_NAME = "tracking.db"
5 INPUT_FILE = "traffic_log_sentences.json"
6
7 # 1. Dati su fucci
8 conn = sqlite3.connect(DB_NAME)
9 cursor = conn.cursor()
10
11 # 2. Kreiraju se tablice
12 cursor.execute(
13     CREATE TABLE IF NOT EXISTS requests (
14         id INTEGER PRIMARY KEY AUTOINCREMENT,
15         visited_page TEXT,
16         visited_time TEXT,
17         request_domain TEXT,
18         url TEXT,
19         method TEXT,
20         cookies TEXT,
21         user_agent TEXT,
22         is_main_page INTEGER
23     )
24 )
25
26 # 3. Učitavaju se linije iz fajla u listu
27 with open(INPUT_FILE, "r", encoding="utf-8") as file:
28     for line in file:
29         entry = json.loads(line)
30
31         cursor.execute(
32             INSERT INTO requests (
33                 visited_page,
34                 visited_time,
35                 request_domain,
36                 url,
37                 method,
38                 cookies,
39                 user_agent,
40                 is_main_page
41             )
42             VALUES (?, ?, ?, ?, ?, ?, ?, ?)
43         )
44         entry["visited_page"] = entry.get("visited_page")
45         entry["visited_time"] = entry.get("visited_time")
46         entry["url"] = entry.get("url")
47         entry["request_domain"] = entry.get("request_domain")
48         entry["cookies"] = entry.get("cookies", "")
49         entry["user_agent"] = entry.get("user_agent")
50         entry["is_main_page"] = entry.get("is_main_page", 0)
51
52         conn.commit()
53         conn.close()
54
55 print("Fajl je uspešno spremljen u trakingu.db")
```

log_to_db.py

Skripta učitava očišćene podatke iz JSON datoteke i sprema ih u SQLite bazu podataka.

Kreira tablicu za mrežne zahtjeve i bilježi osnovne informacije potrebne za analizu web praćenja.

Provjera baze podataka

```
1 import sqlite3
2
3 conn = sqlite3.connect("tracking.db")
4 cur = conn.cursor()
5
6 print("== OSNOVNE STATISTIKE ==")
7 cur.execute("SELECT COUNT(*) FROM requests")
8 print("Ukupan broj zapisa:", cur.fetchone()[0])
9
10 print("\n== PRIMJER ZAPISA ==")
11 cur.execute("""
12     SELECT visited_site, request_domain, is_third_party, category
13     FROM requests
14     LIMIT 10
15 """)
16 for row in cur.fetchall():
17     print(row)
18
19 print("\n== TRACKERI PO KATEGORIJI ==")
20 cur.execute("""
21     SELECT category, COUNT(*)
22     FROM requests
23     WHERE is_third_party = 1
24     GROUP BY category
25 """)
26 for row in cur.fetchall():
27     print(row)
28
29 conn.close()
```

check_db.py

```
PS C:\Users\snaxy\Desktop\sis> python check_db.py
>>
== OSNOVNE STATISTIKE ==
Ukupan broj zapisa: 7556

== PRIMJER ZAPISA ==
('', 'mobile.events.data.microsoft.com', 1, 'telemetry')
('www.bing.com', 'www.bing.com', 0, None)
('www.bing.com', 'www.bing.com', 0, None)
('www.bing.com', 'www.bing.com', 0, None)
('www.bing.com', 'fp.msedge.net', 1, 'other')
('', 'clients2.google.com', 0, None)
('', 'www.google.com', 0, None)
('', 'clientservices.googleapis.com', 0, None)
('', 'accounts.google.com', 0, None)
('www.google.com', 'www.gstatic.com', 1, 'other')

== TRACKERI PO KATEGORIJI ==
('advertising', 222)
('analytics', 88)
('other', 5110)
('social', 109)
('telemetry', 2)
```

Prikaz baze podataka

DB Browser (SQLite)

Detekcija i klasifikacija trackera

classify_trackers.py

sustav automatski prepoznaće i klasificira third-party zahtjeve prikupljene tijekom pregledavanja web-stranica.

Analiza se temelji na poznatim domenama trackera (npr. Google, Facebook, TikTok) i pravilima koja ih svrstavaju u kategorije poput *analytics*, *advertising*, *social* i *telemetry*.

Rezultati klasifikacije spremaju se u bazu podataka, čime se podaci pripremaju za daljnju analizu, bodovanje web-stranica i vizualizaciju u grafovima.

Detekcija i klasifikacija trackera

```
1 import sqlite3
2
3 conn = sqlite3.connect("tracking.db")
4 cur = conn.cursor()
5
6 cur.execute("""
7 UPDATE requests
8 SET is_third_party = 1
9 WHERE visited_site = ''
10 AND (
11     request_domain LIKE '%microsoft.com%'
12     OR request_domain LIKE '%facebook.com%'
13     OR request_domain LIKE '%tiktok.com%'
14 )
15 """)
16
17 conn.commit()
18 conn.close()
19
20 print("Third-party oznake ispravljene za telemetry trackere.")
```

fix_third_party.py

Označava zahtjeve prema **third-party domenama**.

Ispravlja klasifikaciju za poznate **telemetry i tracking servise**.

Priprema podatke za **daljnju analizu i vizualizaciju**.

Procjena intenziteta praćenja web-stranica

score_sites.py

```
PS C:\Users\snaxy\Desktop\sis> python score_sites.py
== TOP 10 web stranica po tracking_intensity_score ===
('edition.com.hr', 1890.0, 423, 153)
('www.theguardian.com', 1883.0, 445, 110)
('www.tiktok.com', 1864.0, 408, 23)
('edition.cnn.com', 1890.0, 423, 153)
('www.theguardian.com', 1883.0, 445, 110)
('www.tiktok.com', 1864.0, 408, 23)
('www.jutarnji.hr', 1276.0, 324, 60)
('www.theguardian.com', 1883.0, 445, 110)
('www.tiktok.com', 1864.0, 408, 23)
('www.jutarnji.hr', 1276.0, 324, 60)
('www.jutarnji.hr', 1276.0, 324, 60)
('www.hbc.com', 1271.0, 279, 46)
('www.amazon.de', 1286.0, 307, 8)
('www.bbc.com', 1271.0, 279, 46)
('www.amazon.de', 1286.0, 307, 8)
('search.brave.com', 1128.0, 255, 24)
('search.brave.com', 1128.0, 255, 24)
('www.gloria.hr', 1187.0, 226, 59)
('www.liexpress.com', 1088.0, 324, 19)
('www.khanacademy.org', 1022.0, 244, 9)
```

Računa tracking intensity score za svaku web stranicu na temelju broja i vrste trackera.

U obzir uzima third-party zahtjeve, broj jedinstvenih tracker domena i kategorije (advertising, analytics, social, telemetry).

Omogućuje rangiranje web stranica prema razini praćenja (najviše → najmanje).

Prikaz podataka

```
1 import sqlite3
2 import csv
3
4 # SQL no name
5 conn = sqlite3.connect("tracking.db")
6 cur = conn.cursor()
7
8 # Iznosi iz baze podataka u kategorije
9 # da se dosta vise kategorija, one se skupuju
10 cur.execute('''
11     SELECT request_domain,
12             COUNT(category) AS category,
13             COUNT(*) AS visit
14     FROM requests
15     WHERE request_domain IS NOT NULL AND request_domain != ''
16     GROUP BY request_domain, category
17     ORDER BY request_domain, visit DESC
18     ''')
19
20 rows = cur.fetchall()
21
22 # Da jedna domenka spada u vise kategorija
23 domain_category = {}
24 for domain, category, visit in rows:
25     if domain not in domain_category:
26         domain_category[domain] = category
27
28 # Spajanje u CSV
29 with open("graph_nodes.csv", "w", newline="", encoding="utf-8") as f:
30     writer = csv.writer(f)
31     writer.writerow(["id", "category"])
32     for domain, category in domain_category.items():
33         writer.writerow([domain, category])
34
35 conn.close()
36
37 print("graph_nodes.csv uspostavljen je!")
```

export_nodes.py

```
1 import sqlite3
2 import csv
3
4 conn = sqlite3.connect("tracking.db")
5 cur = conn.cursor()
6
7 cur.execute('''
8     SELECT visited_site, request_domain
9     FROM requests
10    WHERE is_third_party = 1
11      AND visited_site != ''
12      ''')
13
14 rows = cur.fetchall()
15
16 with open("graph_edges.csv", "w", newline="", encoding="utf-8") as f:
17     writer = csv.writer(f)
18     writer.writerow(["source", "target"])
19     for site, domain in rows:
20         writer.writerow([site, domain])
21
22 conn.close()
23
24 print("graph_edges.csv uspostavljen je!")
```

export_graph.py

Prikaz podataka

```
ad_gpt_adwords.com
ad_i_category
153413879169.0.adclickshare.net_advertising
2235598796579464627494519.netfarm.googlemediation.com_other
343588685012673913646485322.safefarm.googlemediation.com_other
35957.net_adgroup.net_other
477500105961676237999684998.safefarm.googlemediation.com_other
4794148.adload.com_other
78912.788.allsites.com_other
91520548-2306-4576-9758-02f6d2b6dedge.perspective.cc_other
991000141.doubleclick.net_advertising
4.ams010.us_other
a.leads_tv_other
a.triadicradius.zen_other
al.al1.bn.com_other
al201375989.com_optically.com_other
ad206.casualmedia.com_other
ad207.0415126.com_optically.com_other
ad208.casualmedia.com_other
adv-us.amazon-adsystems.com_other
adv-us.amazon_de_other
adv-us.amazon-asynsys.com_advertising
adv-us-finder.com_other
adv-us-finding.com_other
adv_us_finding.com_other
accounts.google.com_other
accounts.mtaa.com_other
adv.advova.com_other
adv.alltoppress.com_other
ad-delivery.net_other
ad_360field.com_other
ad_doubleclick.net_advertising
ad_tur.com_other
adv_drometrics.net_other
adv_hexagondigital.com_other
adv_persmatic.com_other
adv_stickyants.tv.com_other
adv_us_a_cleaning.net_other
```

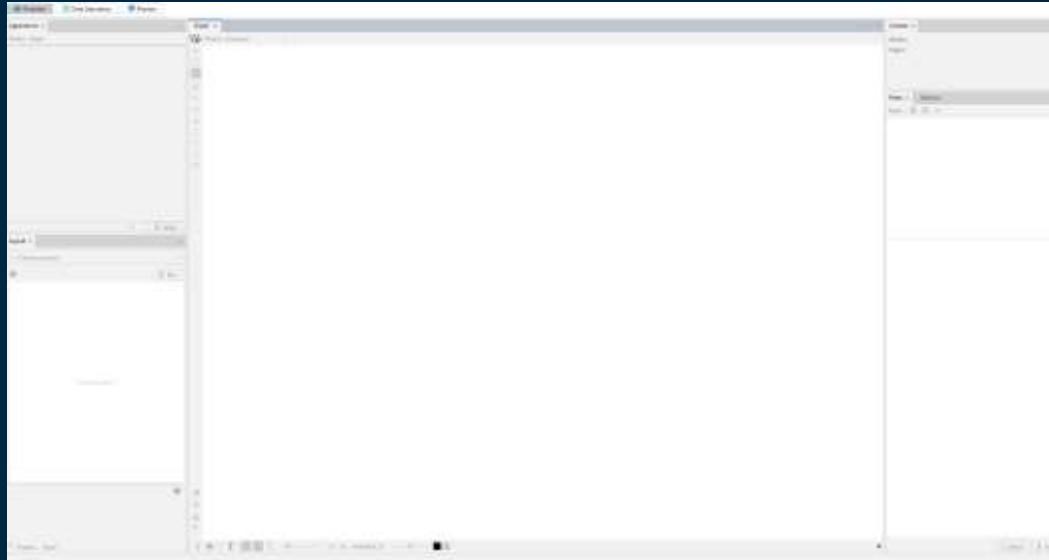
Graph_nodes.csv

Sadrži čvorove grafa – web stranice i tracker domene, zajedno s njihovom kategorijom (analytics, advertising, social, other...).
→ Definira *kto* se pojavljuje u mreži i *što* predstavlja.

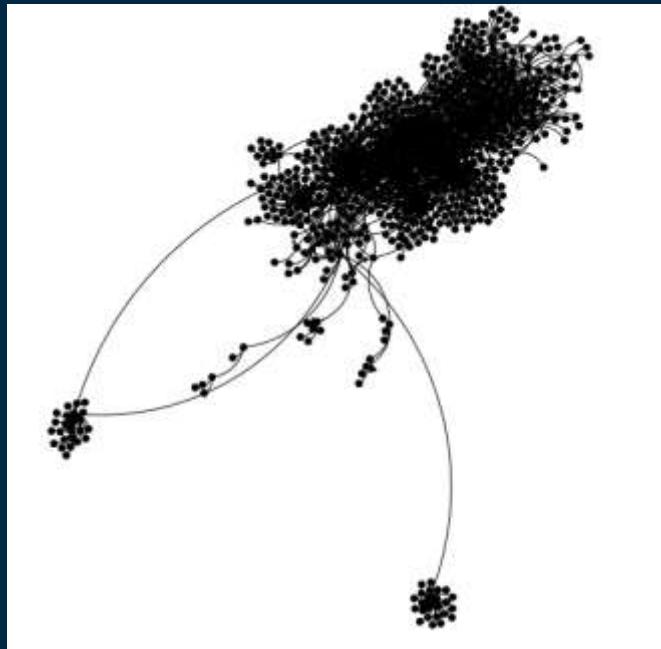
Graph_edges.csv

Sadrži veze između čvorova – pokazuje koja web stranica komunicira s kojim trackerom.
→ Definira *odnose* i tokove praćenja.

Gephi



Vizualizacija mreže trackera (Graph_edges.csv)



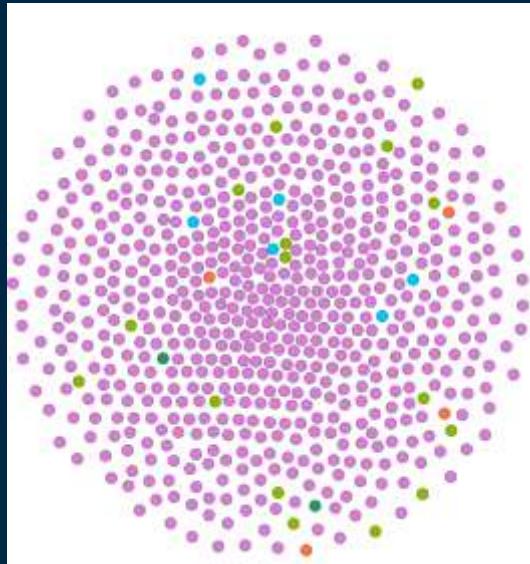
Graph_edges.csv

Graf prikazuje odnose između posjećenih web stranica i third-party tracker domena.

Svaki čvor predstavlja web stranicu ili tracker, a veze (bridovi) označavaju da je tijekom posjeta ostvaren mrežni zahtjev.

Gusti centralni dio pokazuje trackere koji se pojavljuju na velikom broju stranica, dok su izolirani klasteri specifični za pojedine web stranice.

Vizualizacija čvorova mreže (Graph_nodes.csv)



Graph_nodes.csv

Svaki čvor predstavlja web stranicu ili tracker domenu detektiranu tijekom analize.

Boje označavaju kategorije trackera (analytics, advertising, social, telemetry, other).

Gušći centar pokazuje najčešće korištene i najpovezanije trackere koji se pojavljuju na velikom broju stranica.

Vizualizacija omogućuje brzu identifikaciju dominantnih trackera i njihove uloge u ekosustavu praćenja.

Hvala na
Pažnji!