

# 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.

**Pohranjuje prikupljene dokaze u bazu podataka** radi naknadne analize

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

**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.

**Ilan 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 izvješćavanje i prezentaciju.

**Babić Lovro**

# Programi korišteni u sklopu projekta:

## MITMPROXY

Snimanje HTTP/HTTPS prometa  
Presretanje zahtjeva, kolačića i trackera



## PYTHON

Obrada i analiza podataka  
Klasifikacija trackera i izvoz podataka

## SQLITE

Pohrana podataka o praćenju  
Baza zahtjeva, domena i kategorija



## GEPHI

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

# Postavljanje radnog okruženja

```
pip install mitmproxy
```

**Edit proxy server**

Use a proxy server

☒ On

Proxy IP address: 127.0.0.1      Port: 8080

Use the proxy server except for addresses that start with the following entries.  
Use semicolons (;) to separate entries.

☐ Don't use the proxy server for local (intranet) addresses

Save      Cancel

# 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(">")[0], # uklanja query 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 with open(INPUT_FILE, "r", encoding="utf-8") as infile, \
24      open(OUTPUT_FILE, "w", encoding="utf-8") as outfile:
25
26     for line in infile:
27         entry = json.loads(line)
28         clean = sanitize_entry(entry)
29         outfile.write(json.dumps(clean) + "\n")
30
31 print("Sanitized log created:", OUTPUT_FILE)
```

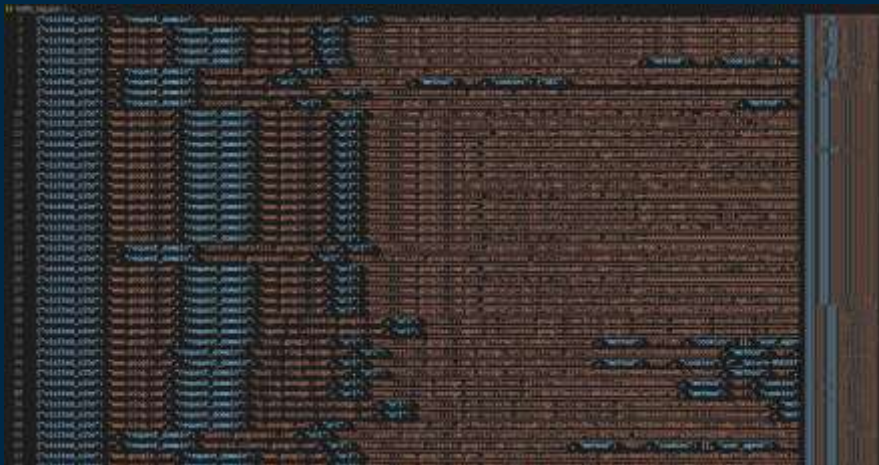
sanitize\_log.py

Čisti sirove logove

Uklanja nepotrebne ili  
osjetljive dijelove

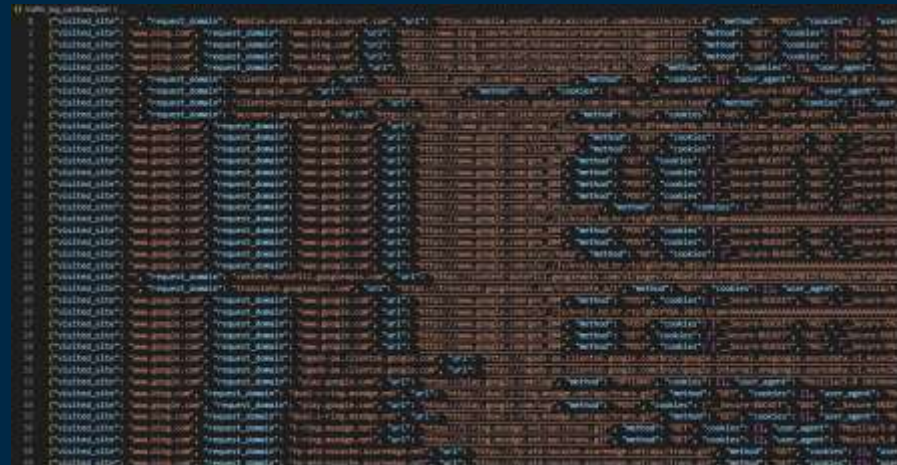
Priprema podatke za  
daljnju obradu

# Dobiveni rezultati

A screenshot of a text file named traffic\_log.json. The file contains a large number of lines of raw, unstructured log data. The text is dense and repetitive, with many lines starting with "request\_line" and "status\_line". The data is not easily readable due to its raw and unstructured nature.

traffic\_log.json

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

A screenshot of a text file named traffic\_log\_sanitized.json. The file contains a large number of lines of structured log data. The data is organized into a consistent format, with fields like "request\_line", "status\_line", and "response\_line" clearly visible. The data is more readable than the raw log file.

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_analitics.json"
6
7 # 1. Test na bazu
8 conn = sqlite3.connect(DB_NAME)
9 cursor = conn.cursor()
10
11 # 2. Kreiranje tablice
12 cursor.execute("""
13 CREATE TABLE IF NOT EXISTS requests (
14     id INTEGER PRIMARY KEY AUTOINCREMENT,
15     visited_site TEXT,
16     request_domain TEXT,
17     url TEXT,
18     method TEXT,
19     cookies TEXT,
20     user_agent TEXT,
21     is_third_party INTEGER
22 )
23 """)
24
25 # 3. Učitavanje podataka iz JSON u bazu
26 with open(INPUT_FILE, "r", encoding="utf-8") as f:
27     for line in f:
28         entry = json.loads(line)
29
30         cursor.execute("""
31             INSERT INTO requests (
32                 visited_site,
33                 request_domain,
34                 url,
35                 method,
36                 cookies,
37                 user_agent,
38                 is_third_party
39             ) VALUES (?, ?, ?, ?, ?, ?, ?)
40             """,
41             [
42                 entry.get("visited_site"),
43                 entry.get("request_domain"),
44                 entry.get("url"),
45                 entry.get("method"),
46                 ", ".join(entry.get("cookies", [])),
47                 entry.get("user_agent"),
48                 int(entry.get("is_third_party", False))
49             ]
50         )
51
52     conn.commit()
53     conn.close()
54
55 print("Podaci iz logova spremljeni u tracking.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\snaaxy\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)
```

7

10

[illegible]

1

7

# Detekcija i klasifikacija trackera

```
def main():
    # URL stranice
    URL = "https://www.example.com"

    # Postavke za klijent
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36",
        "Referer": URL,
        "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
        "Accept-Language": "en-US,en;q=0.9",
        "Accept-Encoding": "gzip, deflate, br",
        "Connection": "keep-alive",
        "Cache-Control": "max-age=0",
        "Pragma": "no-cache",
        "Expires": "0",
        "Host": "www.example.com",
        "Cookie": ""
    }

    # Izvrši GET zahtjev
    response = requests.get(URL, headers=headers)

    # Analiza sadržaja stranice
    soup = BeautifulSoup(response.text, 'html.parser')

    # Pronalazak trackera
    trackers = []
    for script in soup.find_all('script'):
        if script.get('src') and 'track' in script.get('src').lower():
            trackers.append(script.get('src'))

    # Klasifikacija trackera
    for tracker in trackers:
        category = classify_tracker(tracker)
        trackers.append((tracker, category))

    # Izpis rezultata
    print("Detekcija trackera:")
    for tracker, category in trackers:
        print(f"Tracker: {tracker} | Kategorija: {category}")

def classify_tracker(url):
    """Klasifikacija trackera na osnovu URL-a"""
    # Analiza URL-a
    domain = urlparse(url).netloc
    path = urlparse(url).path

    # Prepoznavanje trackera
    if 'google' in domain:
        return 'analytics'
    elif 'facebook' in domain:
        return 'social'
    elif 'tiktok' in domain:
        return 'social'
    elif 'ad' in domain:
        return 'advertising'
    elif 'telem' in domain:
        return 'telemetry'
    else:
        return 'unknown'

if __name__ == '__main__':
    main()
```

classify\_trackers.py

sustav automatski prepoznaje 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 service**.

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

score\_sites.py

7

# Prikaz podataka

```
1 import sqlite3
2 import csv
3
4 # SQL query
5 conn = sqlite3.connect("tracking.db")
6 cur = conn.cursor()
7
8 # Select distinct domain + category
9 # Also domain has also category, can be selected
10 cur.execute("""
11 SELECT request_domain,
12        COUNT(category, 'other') AS category,
13        COUNT(*) AS cnt
14 FROM requests
15 WHERE request_domain IS NOT NULL AND request_domain != ''
16 GROUP BY request_domain, category
17 ORDER BY request_domain, cnt DESC
18 """)
19
20 rows = cur.fetchall()
21
22 # In table domain have 1 category
23 domain_category = {}
24 for domain, category, cnt in rows:
25     if domain not in domain_category:
26         domain_category[domain] = category
27
28 # Create a dict
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 success generated")
```

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 created")
```

export\_graph.py

# Prikaz podataka

```
graph nodes
1 id,category
2 13361218,fin,doubleclick.net,advertising
3 211376676cf37d1c486a27a7f4c1,saferoom.googleexpedition.com,other
4 2d03d862d1f486c39f136c64a8312,saferoom.googleexpedition.com,other
5 18357,com,algarvea.net,other
6 4f2b6a186f81f1c1c72f996a26f6d,saferoom.googleexpedition.com,other
7 4,html-load.com,other
8 7uq1z,tom.alibaba.com,other
9 85126546-2486-4876-4376-4886f2,code.edge.perspective.app,other
10 9918991,fin,doubleclick.net,advertising
11 a,month.com,other
12 a,month.com,other
13 a,month.com,other
14 a,month.com,other
15 a,month.com,other
16 a,month.com,other
17 a,month.com,other
18 a,month.com,other
19 a,month.com,other
20 a,month.com,other
21 a,month.com,other
22 a,month.com,other
23 a,month.com,other
24 a,month.com,other
25 a,month.com,other
26 a,month.com,other
27 a,month.com,other
28 a,month.com,other
29 a,month.com,other
30 a,month.com,other
31 a,month.com,other
32 a,month.com,other
33 a,month.com,other
34 a,month.com,other
35 a,month.com,other
36 a,month.com,other
37 a,month.com,other
38 a,month.com,other
39 a,month.com,other
40 a,month.com,other
41 a,month.com,other
42 a,month.com,other
43 a,month.com,other
44 a,month.com,other
45 a,month.com,other
46 a,month.com,other
47 a,month.com,other
```

Graph\_nodes.csv

Sadrži čvorove grafa – web stranice i tracker domene, zajedno s njihovom kategorijom (analytics, advertising, social, other...).

→ Definira *tko* se pojavljuje u mreži i *što* predstavlja.

```
graph edges
1 source,target
2 www.king.com,saferoom.googleexpedition.com
3 www.google.com,saferoom.googleexpedition.com
4 www.google.com,saferoom.googleexpedition.com
5 www.google.com,saferoom.googleexpedition.com
6 www.google.com,saferoom.googleexpedition.com
7 www.king.com,saferoom.googleexpedition.com
8 www.king.com,saferoom.googleexpedition.com
9 www.king.com,saferoom.googleexpedition.com
10 www.king.com,saferoom.googleexpedition.com
11 www.king.com,saferoom.googleexpedition.com
12 www.king.com,saferoom.googleexpedition.com
13 www.king.com,saferoom.googleexpedition.com
14 www.king.com,saferoom.googleexpedition.com
15 www.king.com,saferoom.googleexpedition.com
16 www.king.com,saferoom.googleexpedition.com
17 www.king.com,saferoom.googleexpedition.com
18 www.king.com,saferoom.googleexpedition.com
19 www.king.com,saferoom.googleexpedition.com
20 www.king.com,saferoom.googleexpedition.com
21 www.king.com,saferoom.googleexpedition.com
22 www.king.com,saferoom.googleexpedition.com
23 www.king.com,saferoom.googleexpedition.com
24 www.king.com,saferoom.googleexpedition.com
25 www.king.com,saferoom.googleexpedition.com
26 www.king.com,saferoom.googleexpedition.com
27 www.king.com,saferoom.googleexpedition.com
28 www.king.com,saferoom.googleexpedition.com
29 www.king.com,saferoom.googleexpedition.com
30 www.king.com,saferoom.googleexpedition.com
31 www.king.com,saferoom.googleexpedition.com
32 www.king.com,saferoom.googleexpedition.com
33 www.king.com,saferoom.googleexpedition.com
34 www.king.com,saferoom.googleexpedition.com
35 www.king.com,saferoom.googleexpedition.com
36 www.king.com,saferoom.googleexpedition.com
37 www.king.com,saferoom.googleexpedition.com
38 www.king.com,saferoom.googleexpedition.com
39 www.king.com,saferoom.googleexpedition.com
40 www.king.com,saferoom.googleexpedition.com
41 www.king.com,saferoom.googleexpedition.com
42 www.king.com,saferoom.googleexpedition.com
43 www.king.com,saferoom.googleexpedition.com
44 www.king.com,saferoom.googleexpedition.com
45 www.king.com,saferoom.googleexpedition.com
46 www.king.com,saferoom.googleexpedition.com
47 www.king.com,saferoom.googleexpedition.com
```

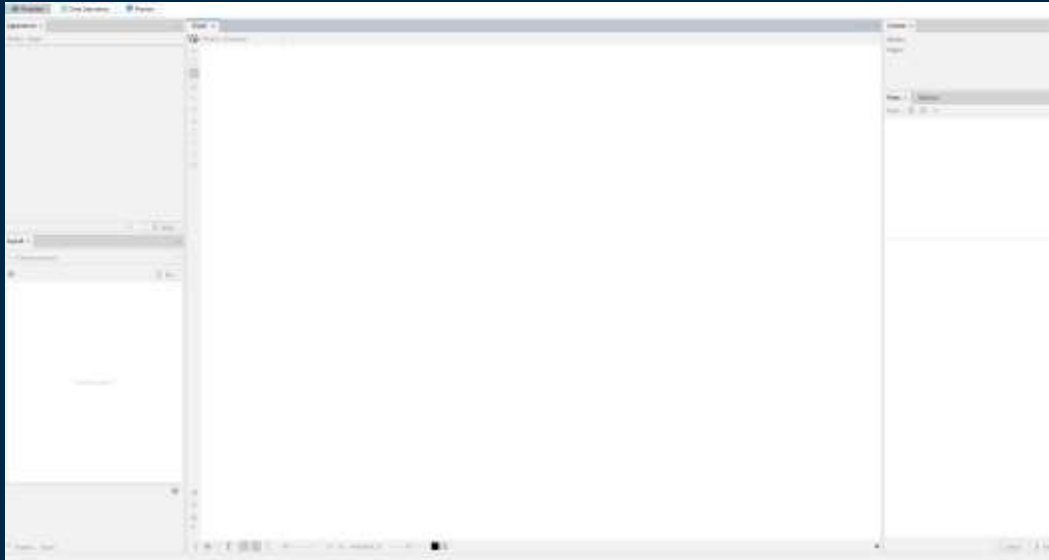
Graph\_edges.csv

Sadrži veze između čvorova – pokazuje koja web stranica komunicira s kojim trackerom.

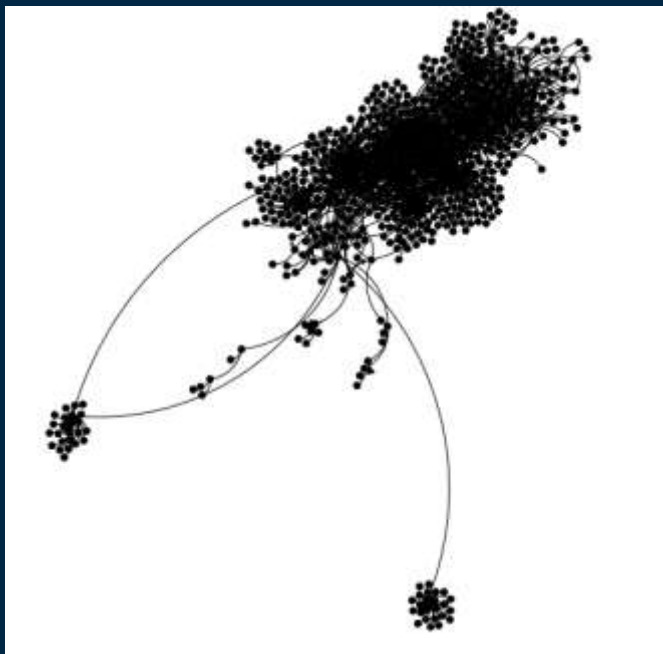
→ Definira *odnose* i tokove praćenja.



# Gephi



# Vizualizacija mreže trackera (Graph\_edges.csv)



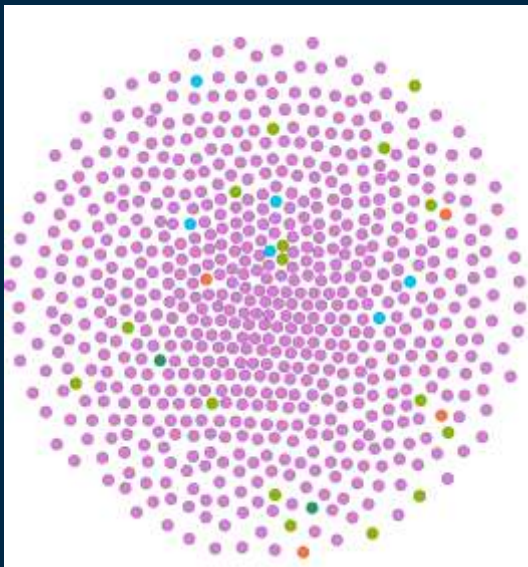
Graph\_edges.csv

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!