Talk overview

AdTech relies on request chains for tracking & bypassing privacy protections

Existing solutions are ineffective against advertising & tracking request chains

Khaleesi:
A purpose-built approach to protect against advertising & tracking request chains
What are request chains?
What are request chains?

GET NEWS.COM

3XX REDIRECT: LOCAL-NEWS.COM

GET LOCAL-NEWS.COM

200 OK

NEWS.COM

LOCAL—NEWS.COM
What are request chains?

GET NEWS.COM

3XX REDIRECT: LOCAL-NEWS.COM

GET LOCAL-NEWS.COM

200 OK

REQUEST

RESPONSE

REQUEST

RESPONSE
Background: Cookie syncing
Background: Cookie syncing

ADVERTISER.COM

ACCESS TRACKER.COM COOKIE

Same-origin policy

USER_ID 123

ACCESS ADVERTISER.COM COOKIE

USER_ID ABC

TRACKER.COM
Background: Cookie syncing

Tracking information shared across origin

(currentUser: USER_ID=123, PARTNER=ADVERTISER.COM)

(cookie: USER_ID=123, PARTNER=TRACKER.COM)

(USER_ID=123)

(currentUser: USER_ID=ABC, PARTNER=TRACKER.COM)

(cookie: USER_ID=ABC, PARTNER=ADVERTISER.COM)

(USER_ID=ABC)

Same-origin policy
Background: Bounce tracking

Has user visited this domain as a website?

NEWS.COM
ADVERTISER.COM
TRACKER.COM
Background: Bounce tracking

Has user visited this domain as a website?
Background: Bounce tracking

Has user visited this domain as a website?

Third-party cookie access granted

NEWS.COM
ADVERTISER.COM
TRACKER.COM

NEWS.COM
ADVERTISER.COM
TRACKER.COM
NEWS.COM/STORY
Talk overview

AdTech relies on request chains for tracking & bypassing privacy protections

Existing solutions are ineffective against advertising & tracking request chains

Khaleesi: A purpose-built approach to protect against advertising & tracking request chains
Current solutions: Ad/Tracker blocking extensions

Widely used solution

Not equipped to detect request chains : ( 
Detect known “trackers” through manually curated filter lists

| TRACKER.COM |
| EXAMPLE.COM |
Current solutions: Ad/Tracker blocking extensions

- Widely used solution
- Not equipped to detect request chains:
- Operate at the level of individual requests
- Detect known “trackers” through manually curated filter lists
- Cannot detect “tracking”
Current solutions: Heuristic based detection

Detect “known” tracking

[Cookie syncing]
If requests share cookies
[BLOCK]
Current solutions: Heuristic based detection

Detect “known” tracking

[Cookie syncing]
If requests share cookies
[BLOCK]

Cookie Syncing

REQUEST
ADVERTISER.COM
COOKIE: USER_ID= 123

RESPONSE
TRACKER.COM?
ADVERTISER_ID=123

REQUEST
TRACKER.COM
COOKIE: USER_ID= 123

RESPONSE
ADVERTISER.COM?
TRACKER_ID=ABC

Bounce tracking

REQUEST
ADVERTISER.COM
COOKIE: NONE

RESPONSE
TRACKER.COM

REQUEST
TRACKER.COM
COOKIE: NONE

RESPONSE
ADVERTISER.COM
Detect “known” tracking

[Cookie syncing]
If requests share cookies
[BLOCK]

Cannot detect “unknown” tracking

Current solutions: Heuristic based detection

**Cookie Syncing**

**REQUEST**
ADVERTISER.COM
COOKIE: USER_ID= 123

**RESPONSE**
TRACKER.COM
COOKIE: USER_ID= 123

**REQUEST**
ADVERTISER.COM?
TRACKER_ID=ABC

**Response**
ADVERTISER.COM

**Bounce tracking**

**REQUEST**
ADVERTISER.COM
COOKIE: NONE

**REQUEST**
TRACKER.COM
COOKIE: NONE

**Response**
TRACKER.COM
COOKIE: NONE

**Response**
ADVERTISER.COM
Current solutions: Machine learning based detection

Capture wide range of tracking behaviors

HTML  JavaScript  Network
Current solutions: Machine learning based detection

Capture wide range of tracking behaviors

HTML → JavaScript

Network

Not specialized to detect request chains

Do not utilize and capitalize on the sequential nature
Current solutions: Machine learning based detection

Capture wide range of tracking behaviors

Not specialized to detect request chains

Do not utilize and capitalize on the sequential nature
Current solutions: Machine learning based detection

Capture wide range of tracking behaviors

Not purpose-built → Accuracy issues!

Not specialized to detect request chains

Do not utilize and capitalize on the sequential nature
AdTech relies on request chains for tracking & bypassing privacy protections

Existing solutions are ineffective against advertising & tracking request chains

Khaleesi:
A purpose-built approach to protect against advertising & tracking request chains
Khaleesi: Motivation & Key idea

- Capitalize on the sequential nature of request chains
- The purpose of request chains becomes clear as they grow
Khaleesi: Motivation & Key idea

REQUEST
ADVERTISER.COM
COOKIE: USER_ID= 123

Third party request
Reading cookie
Khaleesi: Motivation & Key idea

REQUEST
ADVERTISER.COM
COOKIE: USER_ID=123

RESPONSE
TRACKER.COM?
ADVERTISER_ID=123

- Third party request
- Reading cookie
- Third party request
- Sharing identifiers
Khaleesi: Motivation & Key idea

**REQUEST**
ADVERTISER.COM
**COOKIE:** USER_ID= 123

**RESPONSE**
TRACKER.COM?
ADVERTISER_ID=123

**REQUEST**
TRACKER.COM
**COOKIE:** USER_ID= 123

**RESPONSE**
ADVERTISER.COM?
TRACKER_ID=123

Third party request
Third party request
Bilateral data exchange between third parties
Reading cookie
Sharing identifiers
Khaleesi: Motivation & Key idea

Third party request

Reading cookie

Third party request

Sharing identifiers

Bilateral data exchange between third parties
Web Extension APIs

HTTP + JavaScript request chains
Khaleesi: Overview

Web Extension APIs

HTTP + JavaScript request chains

Extract request, response, and sequence features

Labeling using Easylist + EasyPrivacy
Web Extension APIs

HTTP + JavaScript request chains

Extract request, response, and sequence features

Labeling using Easylist + EasyPrivacy

Machine learning classifier training

Automated detection
Request chain organization

Khaleesi captures both request chains at Network and JavaScript layer
Request chain organization

Khaleesi captures both request chains at Network and JavaScript layer

3XX HTTP redirects → Network-layer request chains
Request chain organization

Khaleesi captures both request chains at Network and JavaScript layer

3XX HTTP redirects

Top level navigations with JavaScript

Same script-initiated requests (with common identifiers)

Network-layer request chains

JavaScript-layer request chains
Request chain organization

Khaleesi captures both request chains at Network and JavaScript layer

3XX HTTP redirects

Network-layer request chains

Top level navigations with JavaScript

JavaScript-layer request chains

Same script-initiated requests (with common identifiers)

Cookie Syncing

Bounce Tracking
Khaleesi’s features

- Request features
- Response features
- Sequence features
Khaleesi’s features

Request features
Response features
Sequence features

What the request might do

Example feature

**Long URL length** → Trackers trying to embed identifiers
Khaleesi’s features

- Request features
  - What the request might do
  - Example feature: **Long URL length** → Trackers trying to embed identifiers

- Response features
  - What the response will do
  - Example feature: **Length of the response** → loading of a tracking pixel

- Sequence features
Khaleesi’s features

Request features
- What the request might do

Response features
- What the response will do

Sequence features
- What the chain has been doing

Example feature
- **Long URL length** → Trackers trying to embed identifiers

Example feature
- **Length of the response** → loading of a tracking pixel

Example feature
- **Unique domains in a chain** → Trackers sending data to other trackers
Khaleesi: Classification & Accuracy

Random forest ensemble
Khaleesi: Classification & Accuracy

Random forest ensemble

EasyList/EasyPrivacy ground truth
Alexa top-10K websites
Khaleesi: Classification & Accuracy

Random forest ensemble

EasyList/EasyPrivacy ground truth
98.63% accuracy

Alexa top-10K websites
1,259 new ad/tracking domains identified
Key Takeaways

- AdTech relies on request chains for tracking & bypassing privacy protections
- Existing solutions are ineffective against advertising & tracking request chains
- Khaleesi: A purpose-built approach to protect against advertising & tracking request chains
- Lots of additional analysis in the paper (e.g., Robustness, Performance, etc.)

More details in the paper!