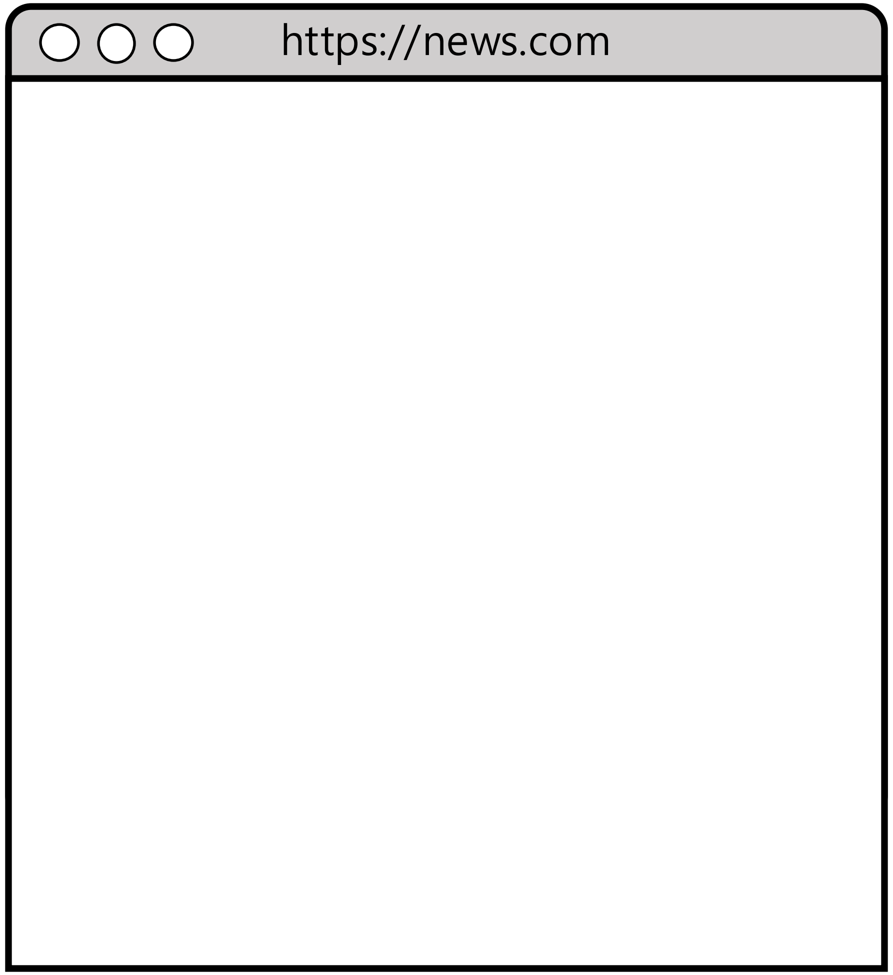
You Only Perturb Once: Bypassing (Robust) Ad-Blockers Using Universal Adversarial Perturbations

Dongwon Shin*, Suyoung Lee*, Sanghyun Hong† and Sooel Son*

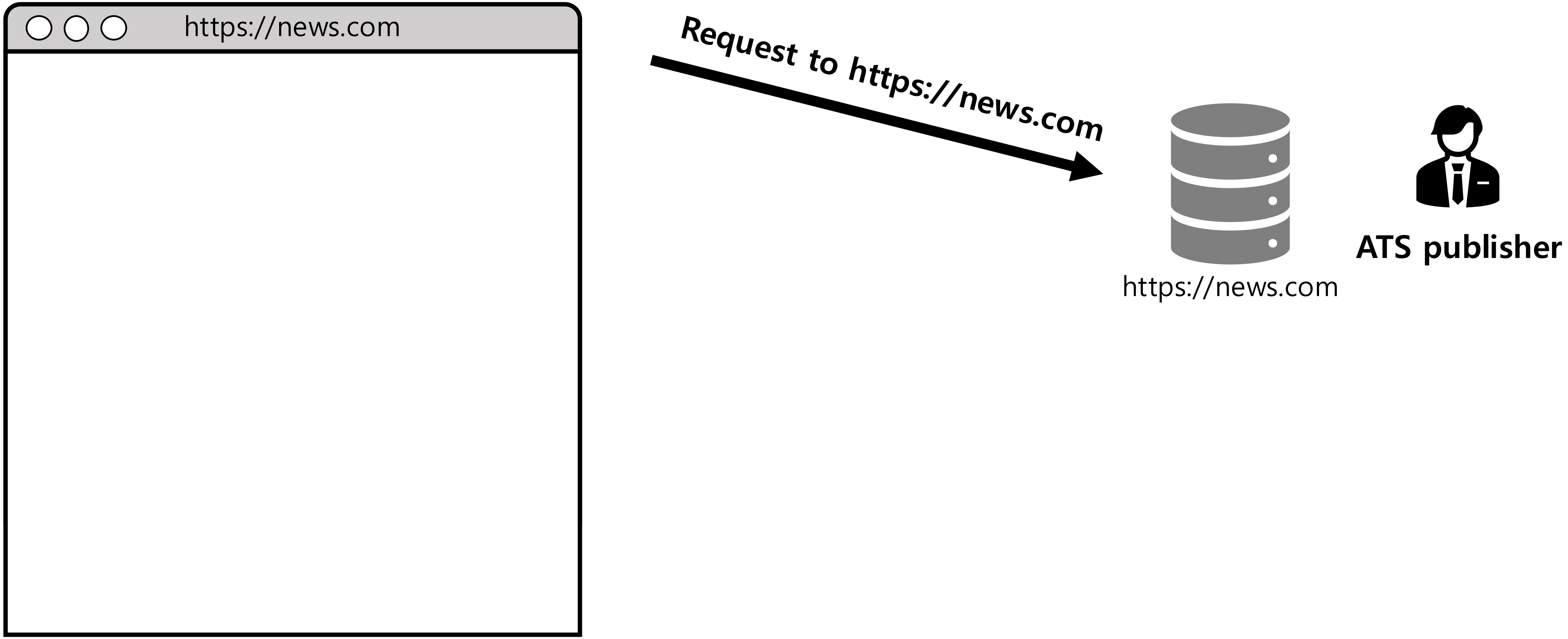
*KAIST, †Oregon State University

ACSAC 2024

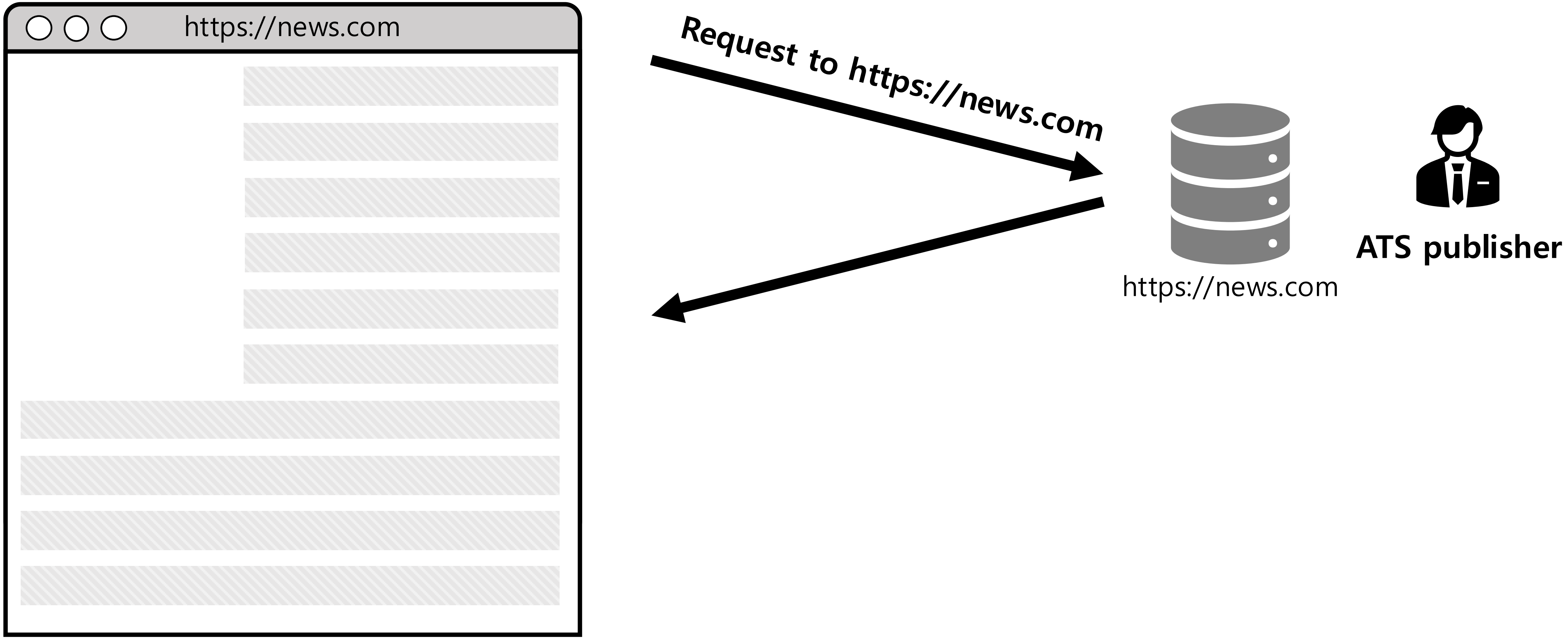
Online Advertising & Tracking Service (ATS)



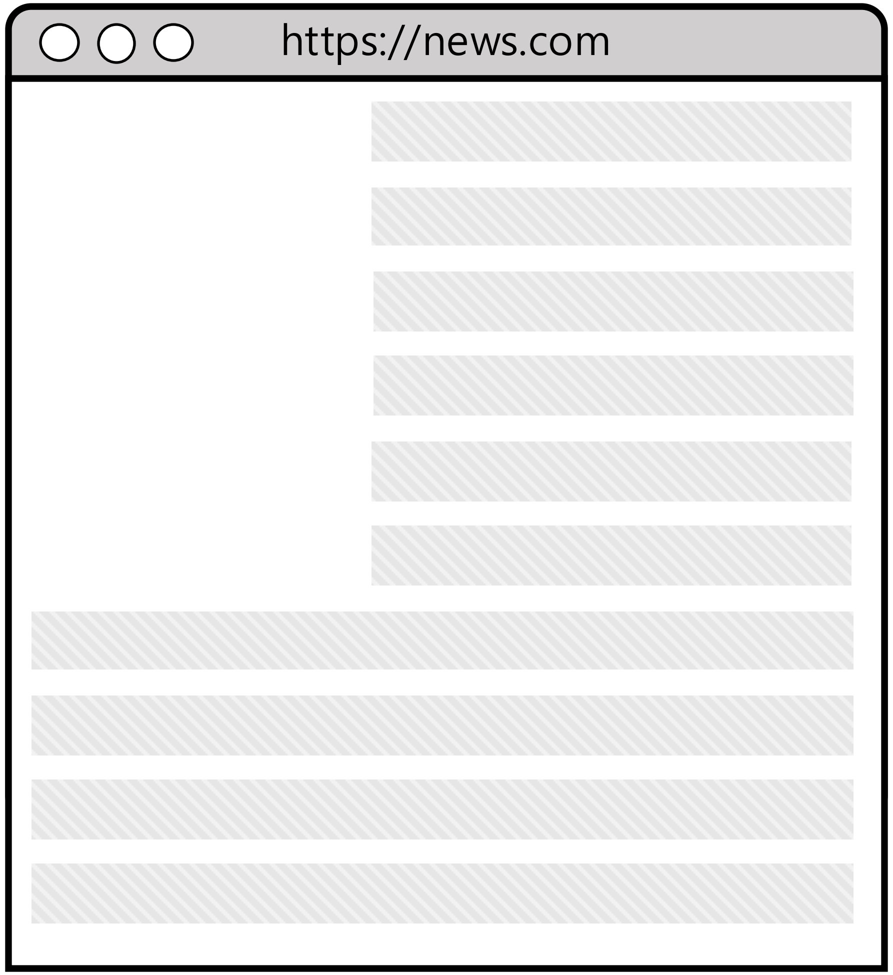
Online Advertising & Tracking Service (ATS)



Online Advertising & Tracking Service (ATS)



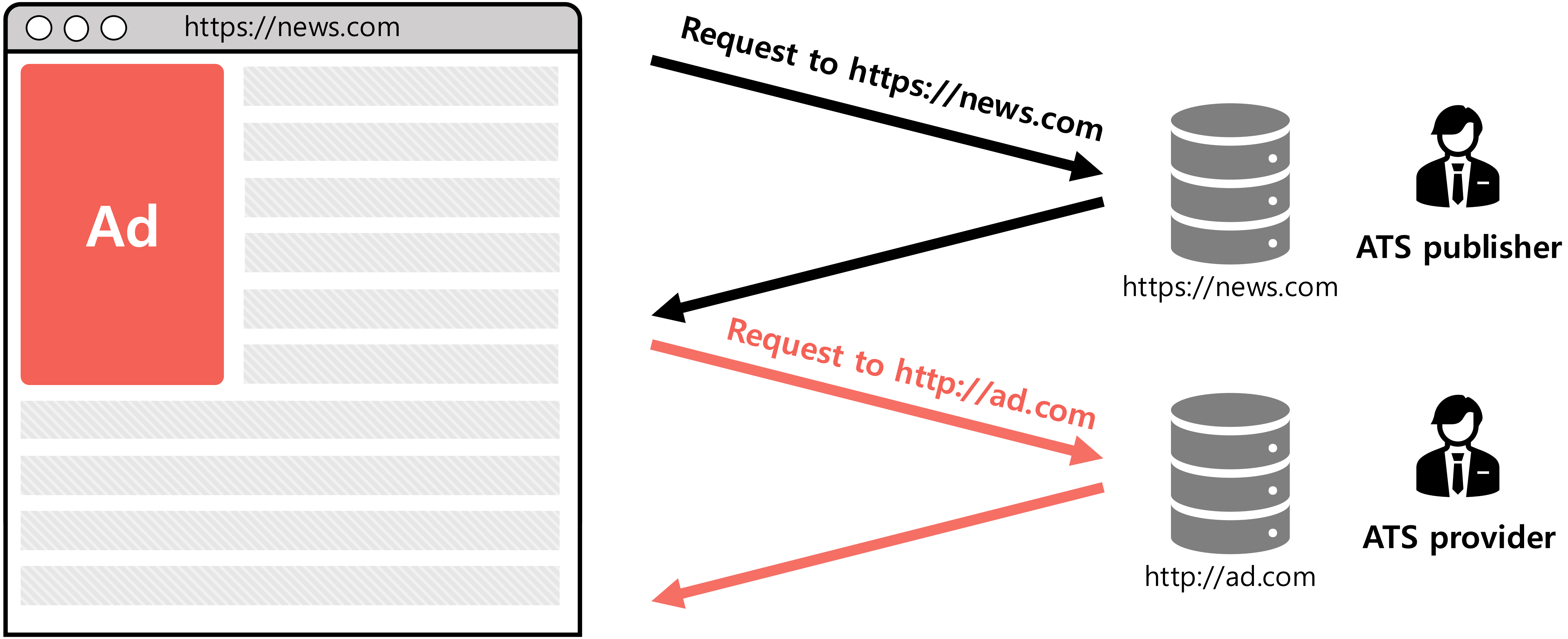
Online Advertising & Tracking Service (ATS)



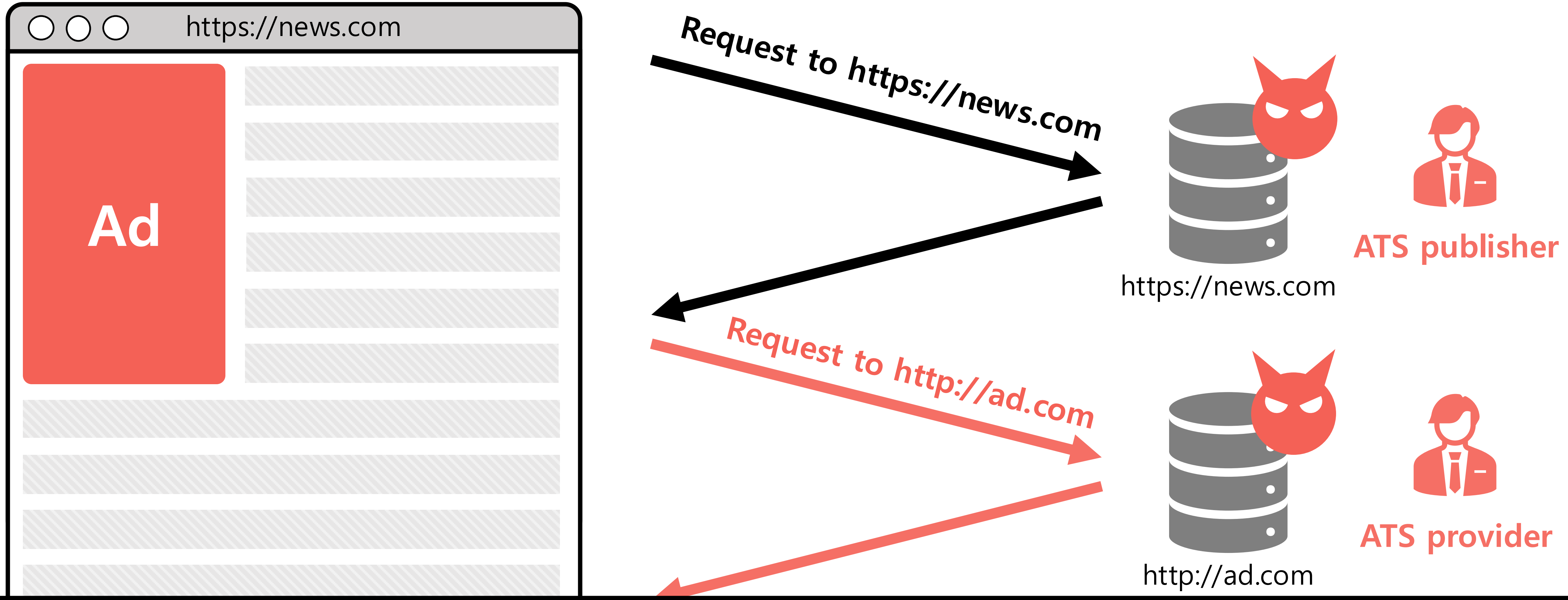
HTML

```
<body>
  <iframe src = ‘http://ad.com/show_ad’>
  …
  <script src = ‘http://ad.com/track_user.js’>
  …
</body>
```

Online Advertising & Tracking Service (ATS)

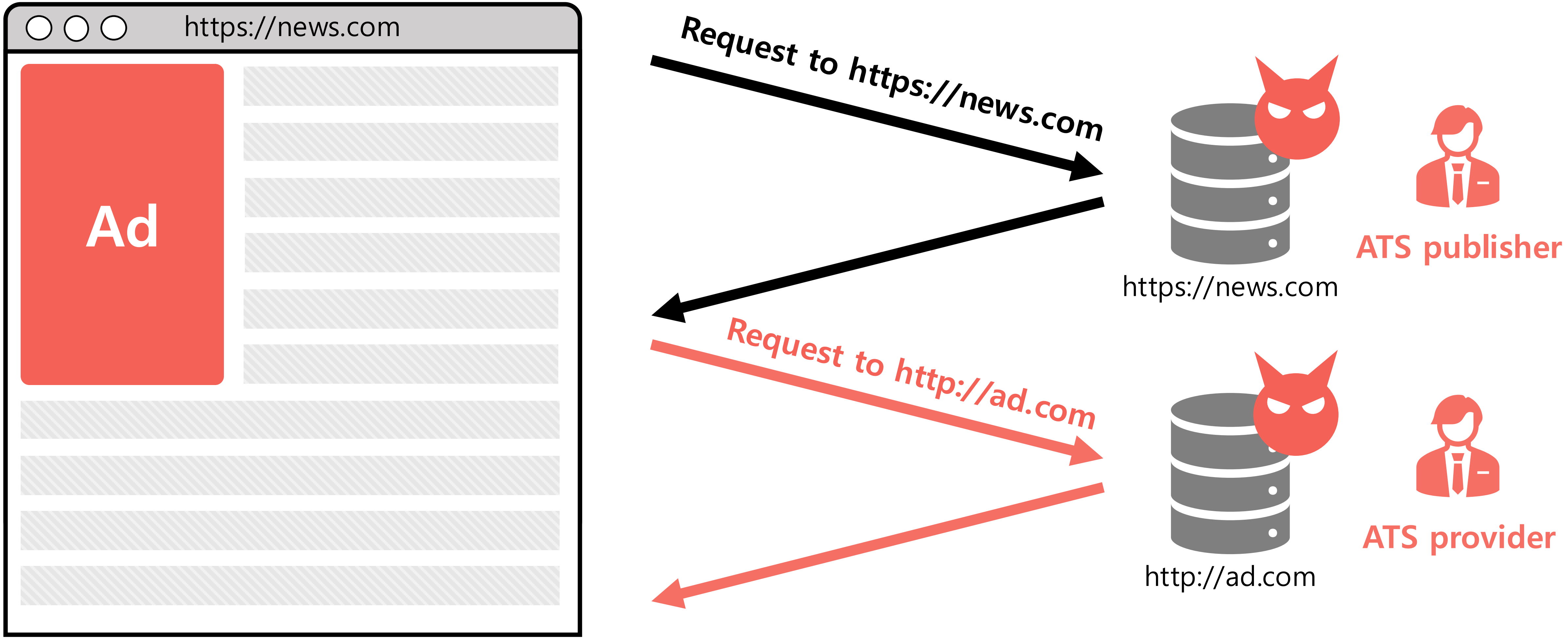


Online Advertising & Tracking Service (ATS)

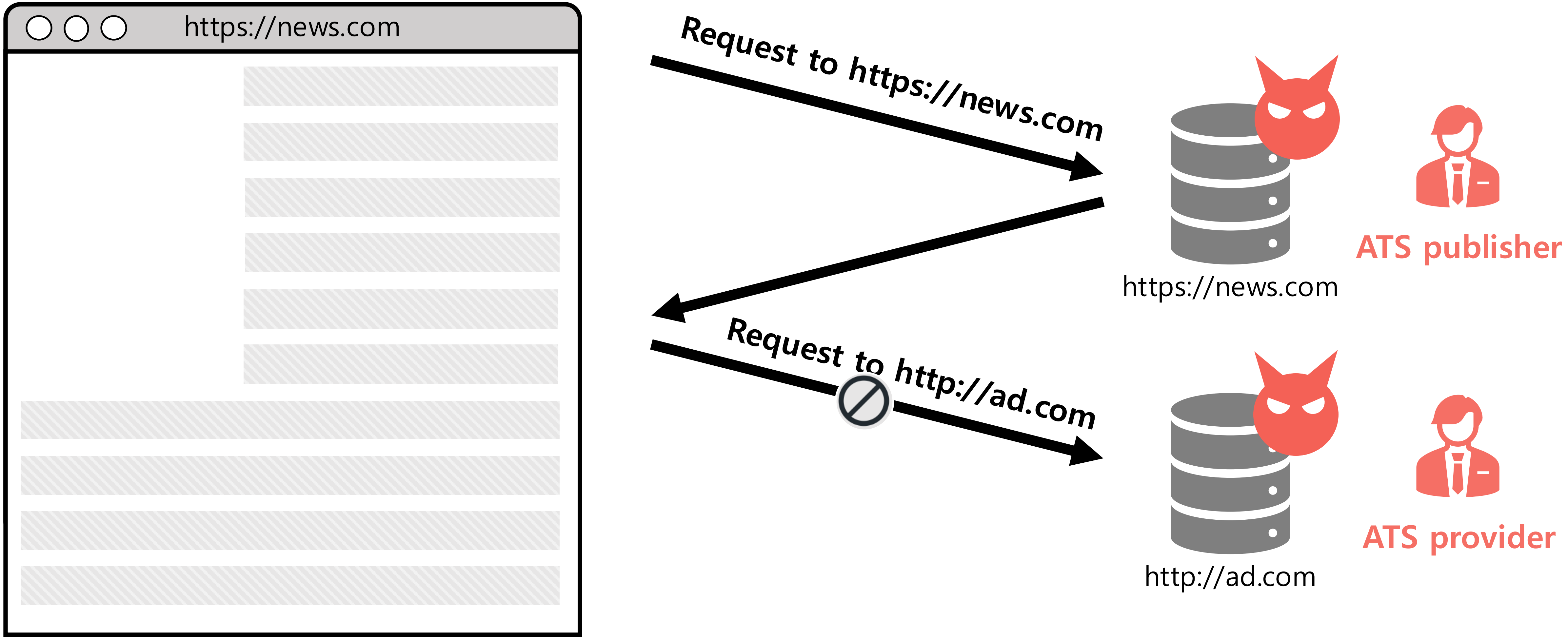


They track users' browsing history!

ATS blockers



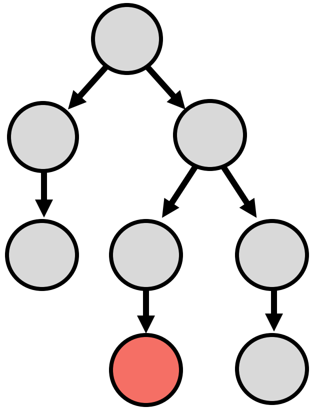
ATS blockers



ATS blockers block resources fetched from ATS providers!

ML-based ATS Blockers

Webpage



Graph representation

Request URL	# of Nodes	URL Length	…
http://ad.com/show_ad	97	21	…

Extracted features

Random forest classifier

Non-ATS

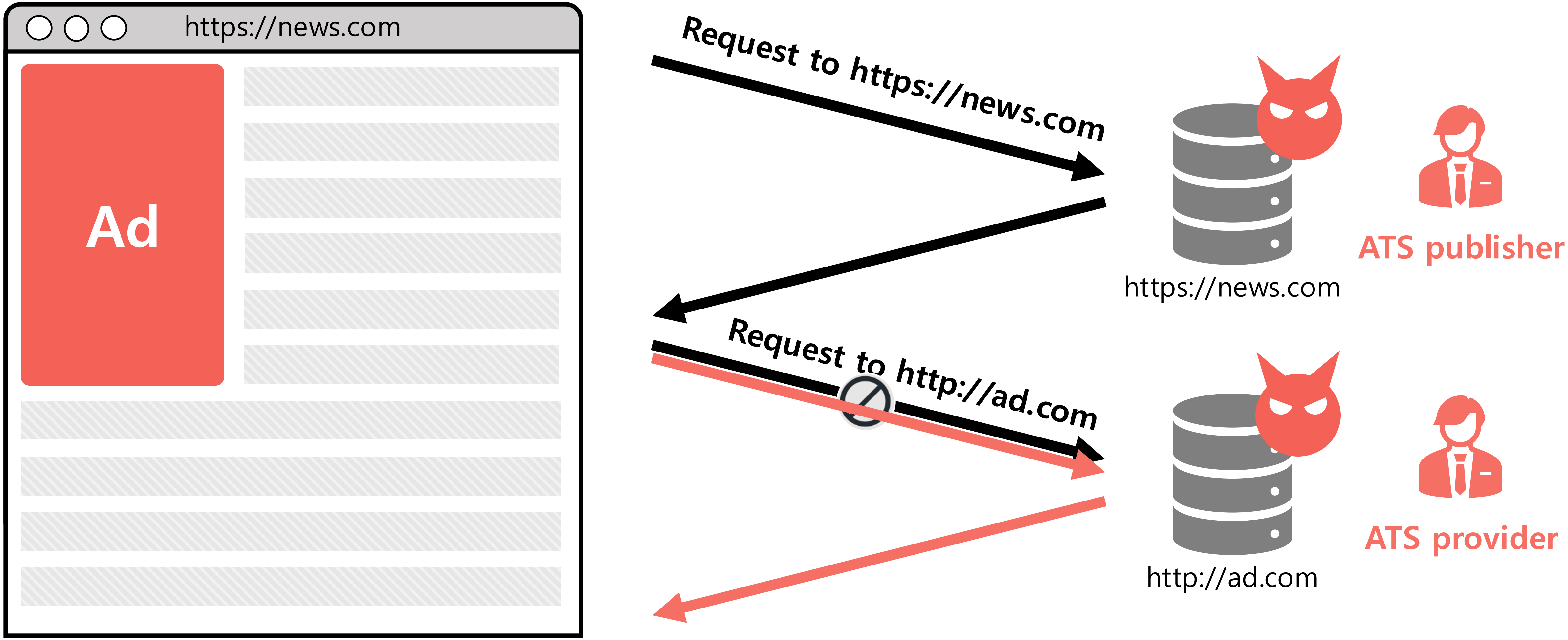
ATS

All these steps take place in the client-side

[1] Iqbal et al. AdGraph: A graph-based approach to ad and tracker blocking. S&P '20
[2] Sjosten et al. Filter list generation for underserved regions. WWW '20
[3] Siby et al. WebGraph: Capturing advertising and tracking information flows for robust blocking. Security '22
[4] Lee et al. AdFlush: A real-world deployable machine learning solution for effective advertisement and web tracker prevention. WWW '24

KAIST Web Security & Privacy Lab Oregon State University SAIL

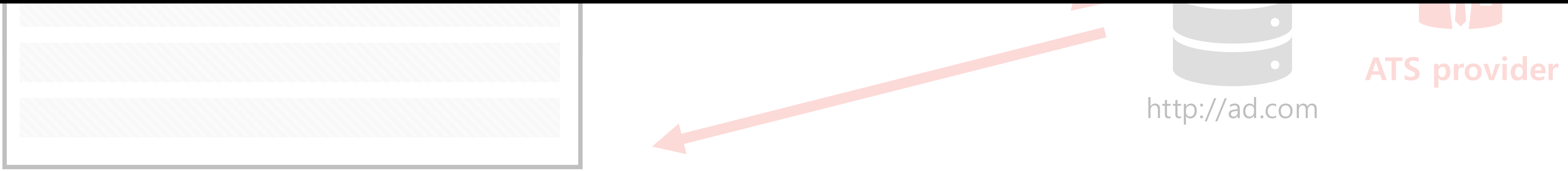
Adversarial Attacks against ML-based ATS Blockers



<u>Abusive ATS publishers and providers</u> may seek to **<u>bypass ATS blockers</u>** to maximize their profit!

Adversarial Attacks against ML-based ATS Blockers

Can ATS publishers/providers bypass these ATS blockers?

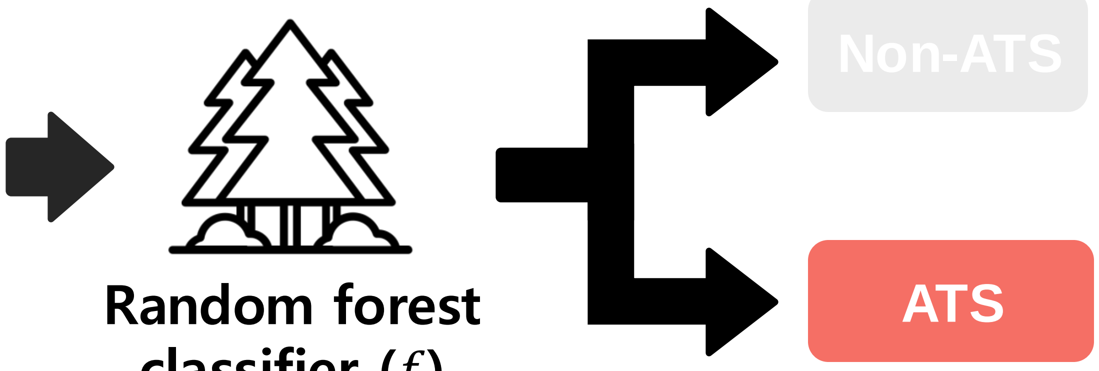


Abusive ATS publishers and providers may seek to bypass ATS blockers to maximize their profit!

Adversarial Attacks against ML-based ATS Blockers

Request URL	# of Nodes	URL Length	···
http://ad.com/show_ad	97	21	···

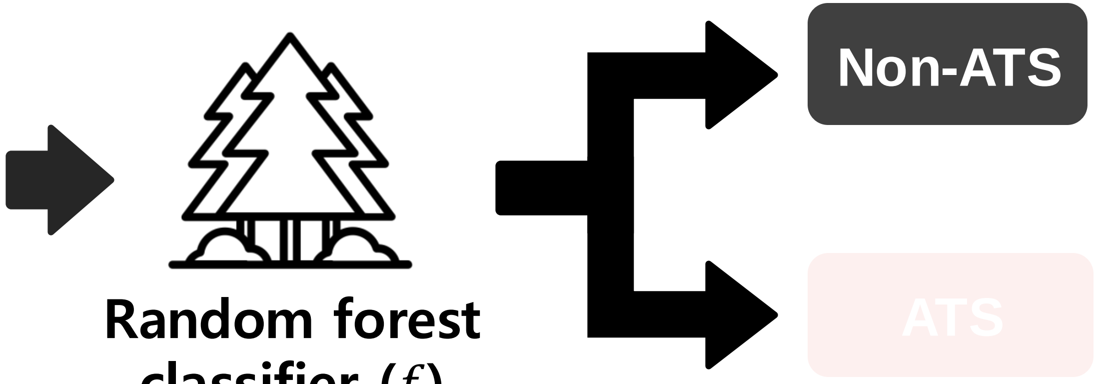
Extracted features (x)



Random forest classifier (f)

Request URL	# of Nodes	URL Length	···
http://ad.com/show_ad	98	81	···

Perturbed features ($x + \delta$)



Random forest classifier (f)

Optimize perturbation (δ) **on this request node** to bypass the ATS blocker!

[5] Zhu et al. Eluding ML-based adblockers with actionable adversarial examples. ACSAC '21

Adversarial Attacks against ML-based ATS Blockers

Request URL	# of Nodes	URL Length	…
http://ad.com/track_user.js	97	27	…

Extracted features (x')

However, what if the adversary wants to attack **another network request**?

Random forest classifier (f)

Non-ATS

ATS

Request URL	# of Nodes	URL Length	…
http://ad.com/show_ad	98	81	…

Perturbed features ($x + \delta$)

Random forest classifier (f)

Non-ATS

ATS

Optimize perturbation (δ) **on this request node** to bypass the ATS blocker!

[5] Zhu et al. Eluding ML-based adblockers with actionable adversarial examples. ACSAC '21

Limitation of Per-Sample Attacks

Request URL	# of Nodes	URL Length	…
http://ad.com/track_user.js	97	27	…

Extracted features (x')

Request URL	# of Nodes	URL Length	…
http://ad.com/track_user.js	101	72	…

Perturbed features ($x' + \delta'$)

The adversary has to optimize perturbation (δ')
<u>**again on this network request node!**</u>

[5] Zhu et al. Eluding ML-based adblockers with actionable adversarial examples. ACSAC '21

Limitation of Per-Sample Attacks

Request URL	# of Nodes	URL Length	…
http://ad.com/track_user.js	97	27	…
http://ad.com/track_user.js	101	72	…

Perturbed features $(x' + \delta')$

Random forest classifier (f)

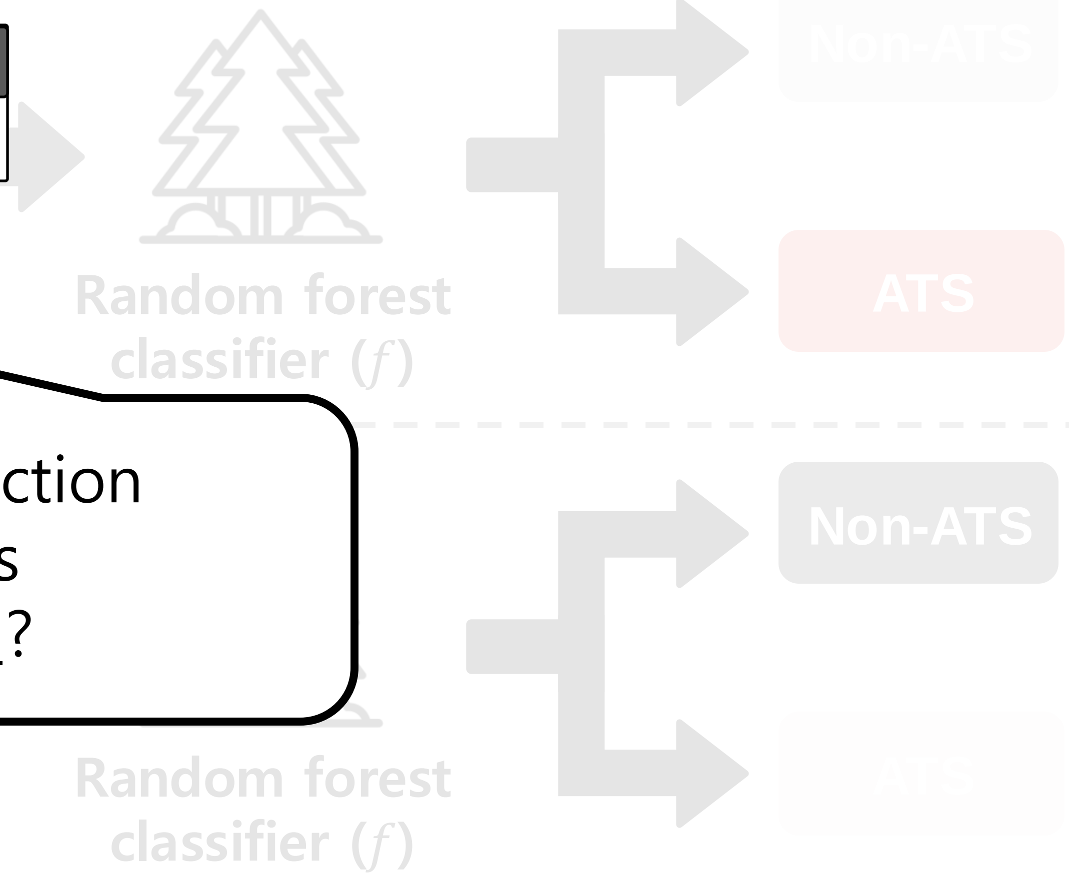
Can adversaries bypass a target ATS blocker at scale?

[5] Zhu et al. Eluding ML-based adblockers with actionable adversarial examples. ACSAC '21

Limitation of Per-Sample Attacks

Request URL	# of Nodes	URL Length	…
http://ad.com/track_user.js	97	27	…

Multiple samples $(x_1, x_2, \ldots, x_n)$



Perturbed features $(x' + \delta')$

Can adversaries bypass the detection of multiple network requests **using a single perturbation**?

[5] Zhu et al. Eluding ML-based adblockers with actionable adversarial examples. ACSAC '21

You Only Perturb Once

Request URL	# of Nodes	URL Length	…
http://ad.com/track_user.js	97	27	…
http://ad.com/track_user.js	101	72	…

Perturbed features ($x' + \delta'$)

Random forest classifier (f)

Non-ATS

ATS

We propose **YOPO**!

[5] Zhu et al. Eluding ML-based adblockers with actionable adversarial examples. ACSAC '21

Our Contributions

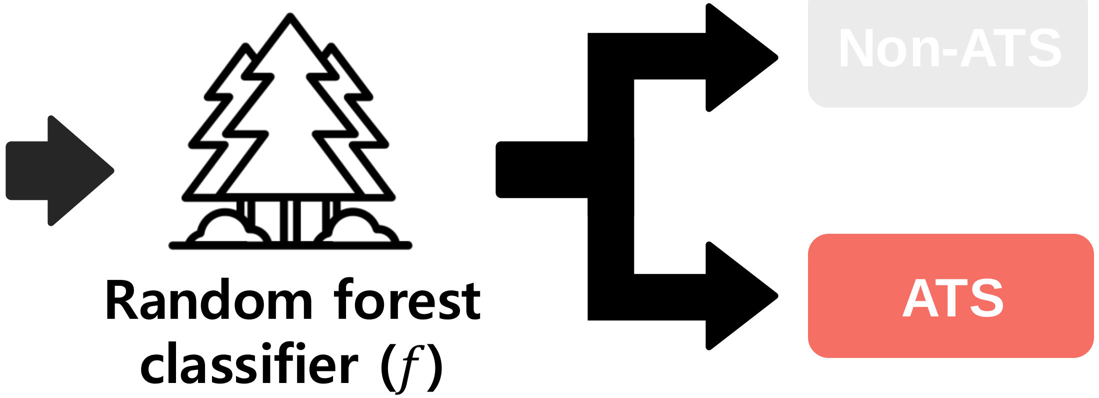
- We show that an adversary can generate **a single and cost-effective universal perturbation** that bypasses recent ML-based ATS blockers.
- We design and implement a novel framework (YOPO) where one can **generate a universal adversarial perturbation (UAP)** against these ATS blockers.
- We propose two new **mitigation strategies** by analyzing the factors attributing to this vulnerability.

Challenge #1: Perturbation Optimization

- Random forest classifiers are not differentiable.

Request URL	# of Nodes	URL Length	⋯
http://ad.com/show_ad	97	21	⋯

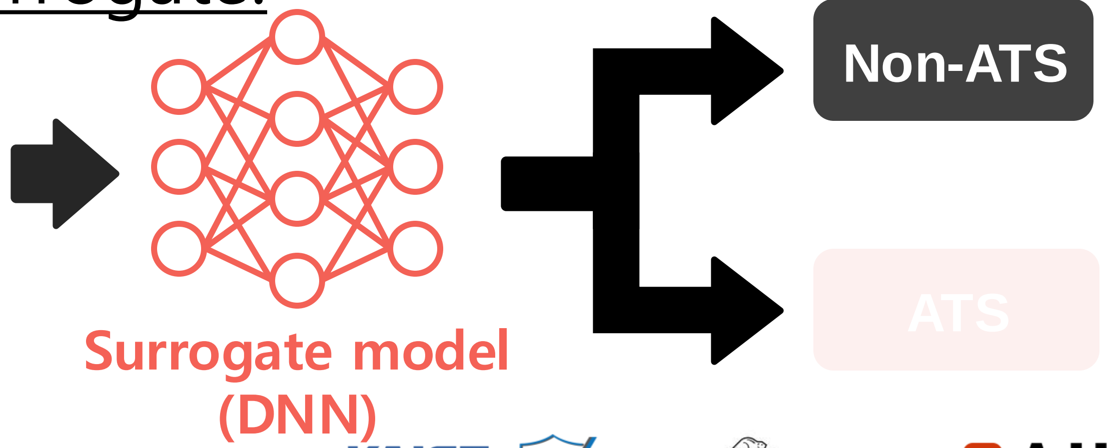
Extracted features (x)



- We train a DNN and use it as a surrogate.

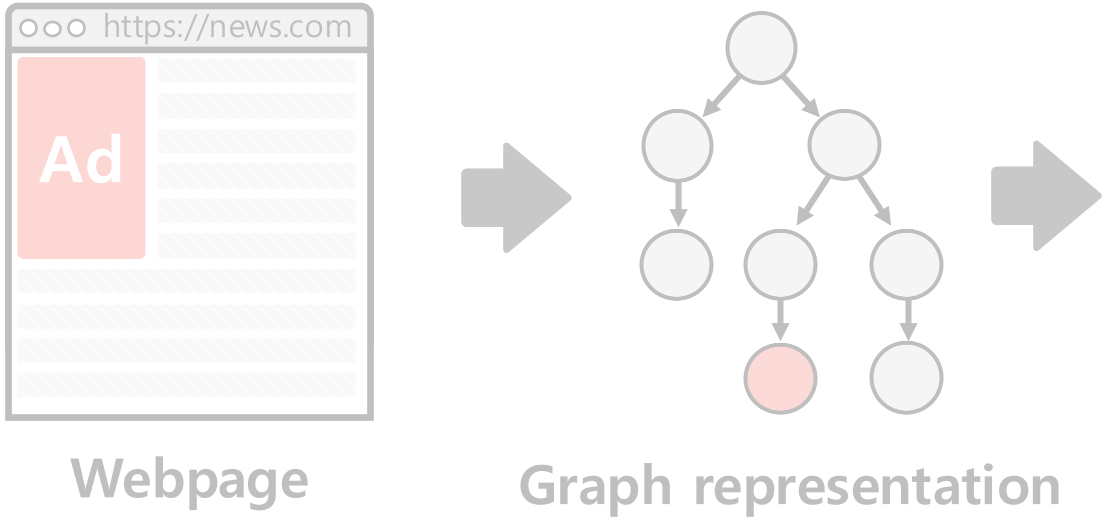
Request URL	# of Nodes	URL Length	⋯
http://ad.com/show_ad	98	21	⋯

Perturbed features ($x + \delta$)



Challenge #2: HTML Manipulation

- Perturbed features should be **reflected in an HTML format**.

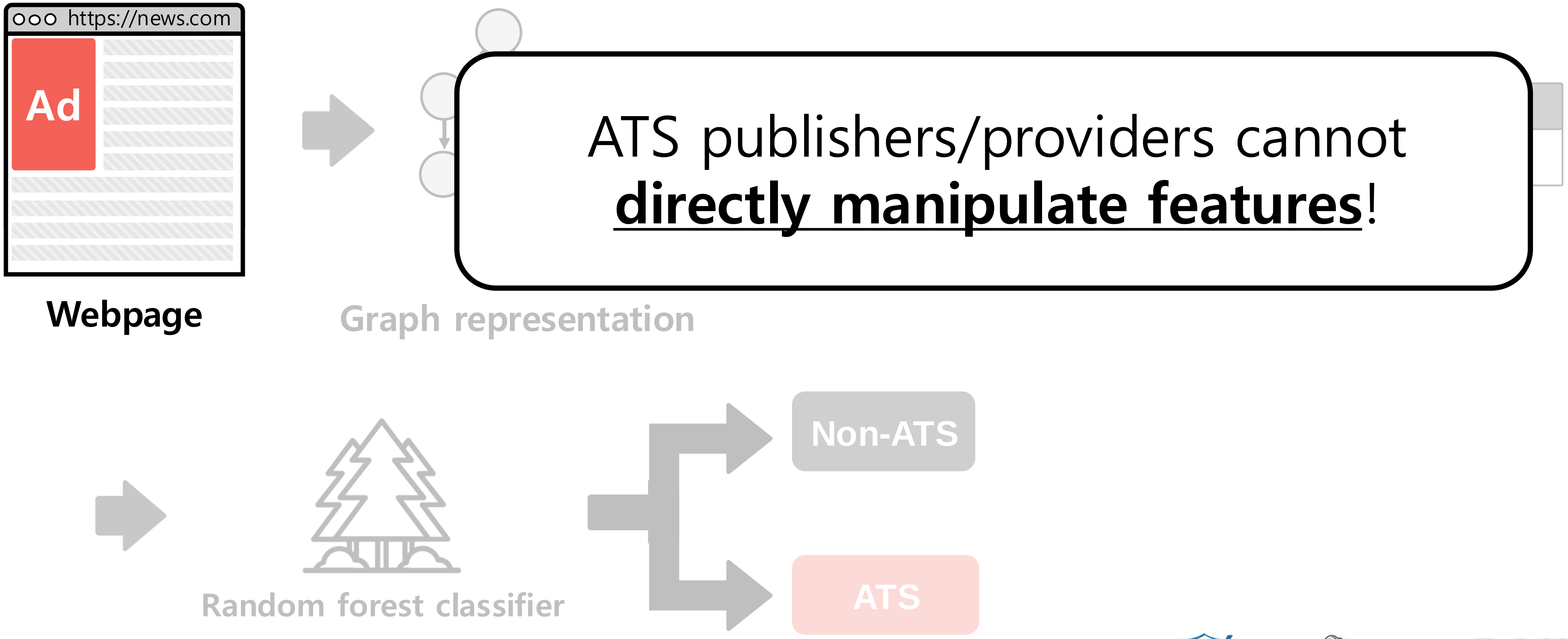


Request URL	# of Nodes	URL Length	…
http://ad.com/show_ad	98	21	…

Perturbed features $(x + \delta)$

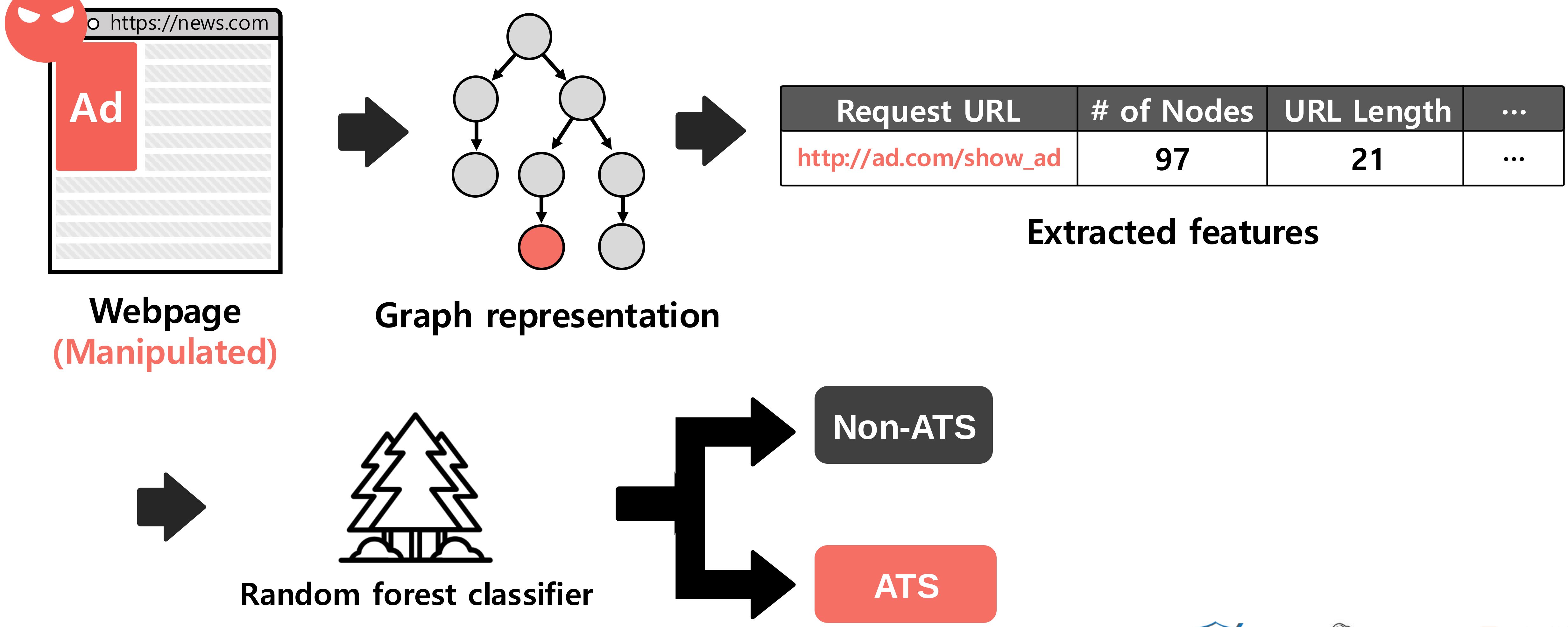
Challenge #2: HTML Manipulation

- Perturbed features should be **reflected in an HTML format**.



Challenge #2: HTML Manipulation

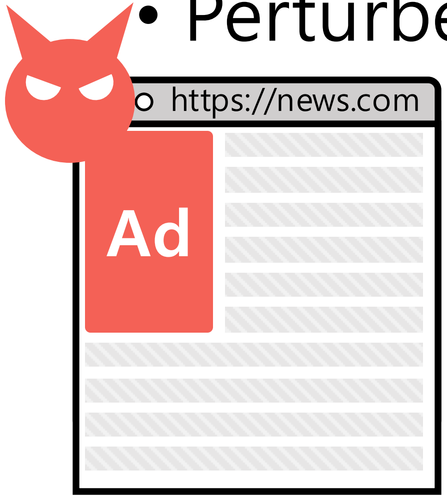
- Perturbed features should be **<u>reflected in an HTML format</u>**.



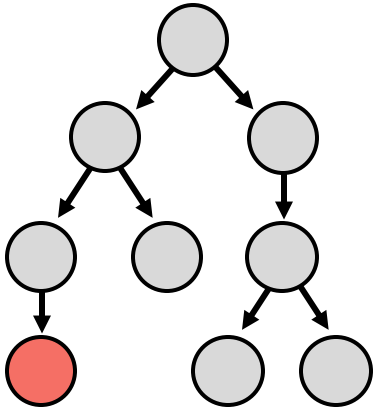
Request URL	# of Nodes	URL Length	…
http://ad.com/show_ad	97	21	…

Challenge #2: HTML Manipulation

- Perturbed features should be **reflected in an HTML format**.



Webpage
(Manipulated)



Graph representation
(Manipulated)

Request URL	# of Nodes	URL Length	…
http://ad.com/show_ad	97	21	…

Extracted features

Random forest classifier

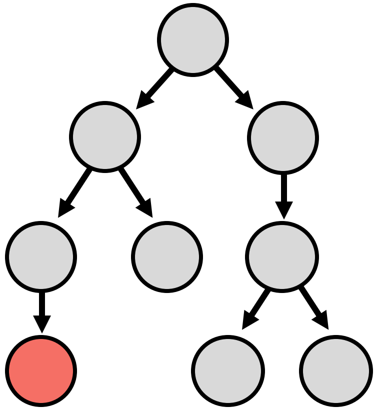
Non-ATS

ATS

Challenge #2: HTML Manipulation

- Perturbed features should be **reflected in an HTML format**.

Webpage
(Manipulated)

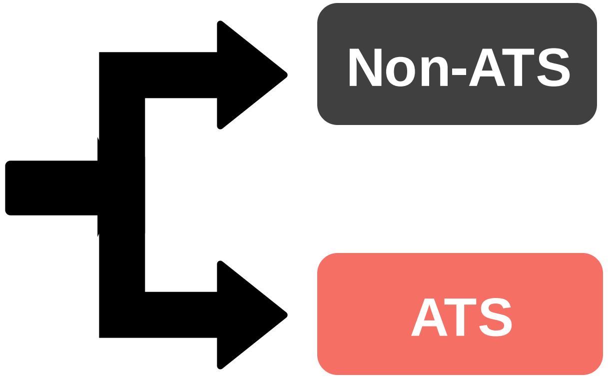


Graph representation
(Manipulated)

Request URL	# of Nodes	URL Length	…
http://ad.com/show_ad	98	21	…

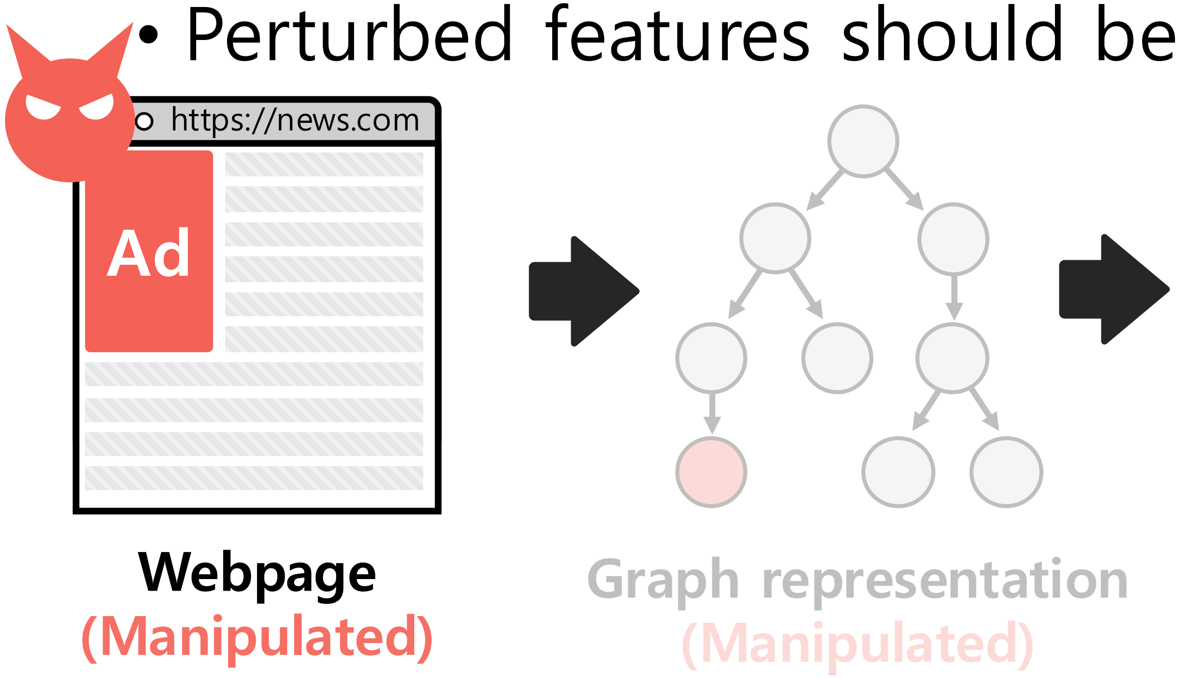
Perturbed features $(x + \delta)$

Random forest classifier



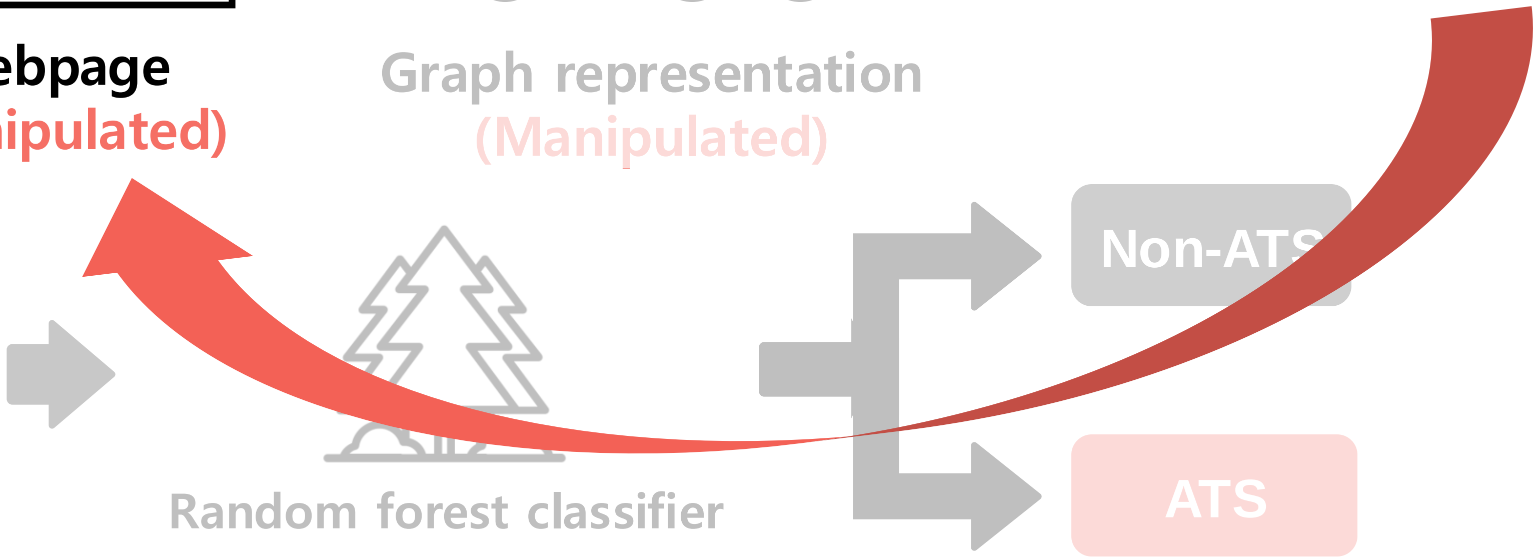
Challenge #2: HTML Manipulation

- Perturbed features should be **reflected in an HTML format**.



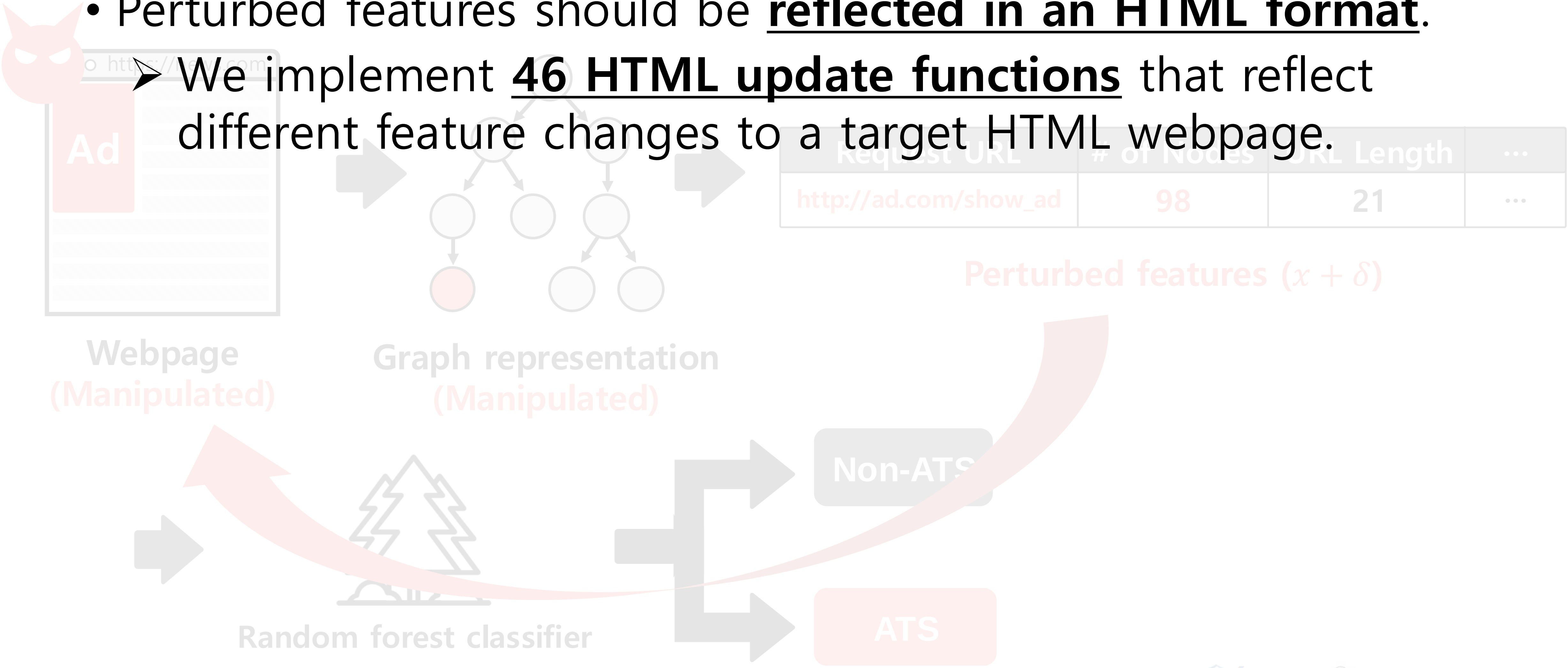
Request URL	# of Nodes	URL Length	...
http://ad.com/show_ad	98	21	...

Perturbed features $(x + \delta)$



Challenge #2: HTML Manipulation

- Perturbed features should be **reflected in an HTML format**.
  - ➢ We implement **46 HTML update functions** that reflect different feature changes to a target HTML webpage.



Challenge #3: Preserving Functionalities

- Each feature has **a different functionality breakage risk** of manipulating it.

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
</body>
```

Original webpage

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad?1234'>
</body>
```

Increased the URL length

☺ **Preserves the functionality!**

Challenge #3: Preserving Functionalities

- Each feature has **a different functionality breakage risk** of manipulating it.
  - ➢ We **designed a cost model** that prioritizes which features to manipulate first.

HTML
<body> <iframe src = 'http://ad.com/show_ad'> </body>

Original webpage

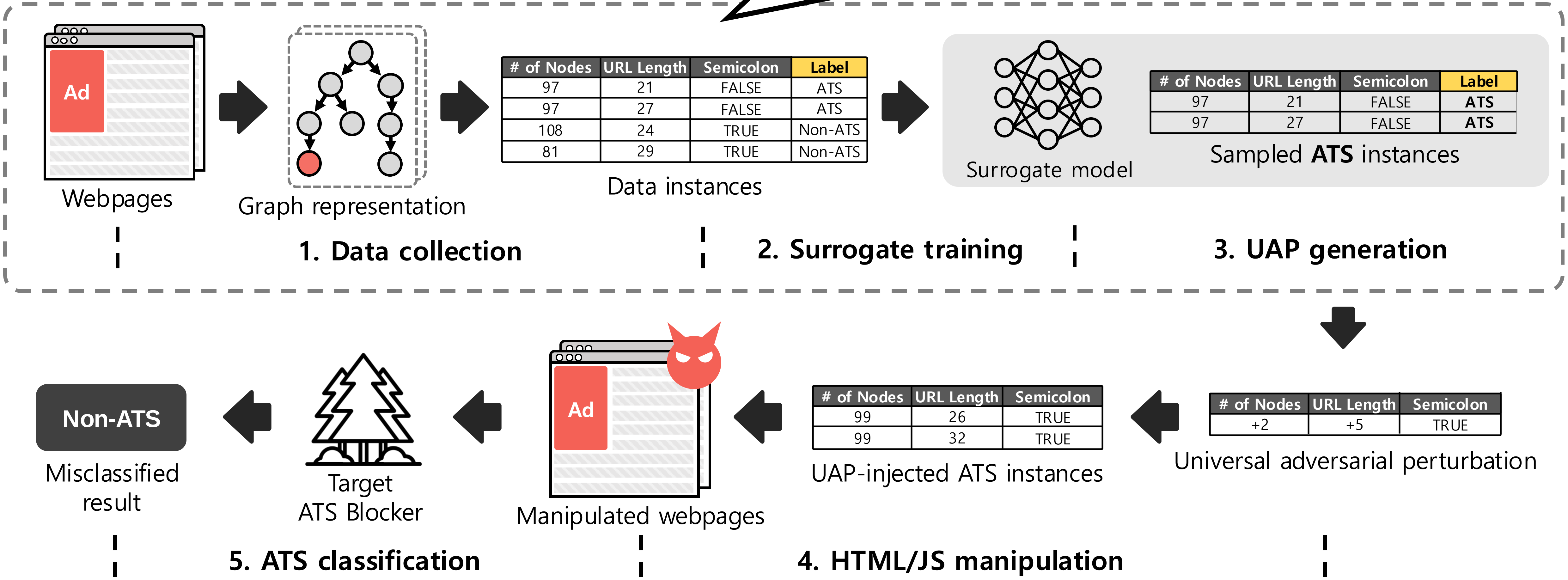
HTML
<script> <iframe src = 'http://ad.com/show_ad'> </script>

Changed its parent tag name

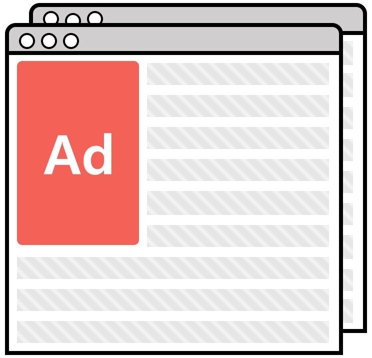
⚠ **Breaks the functionality!**

YOPO Overview

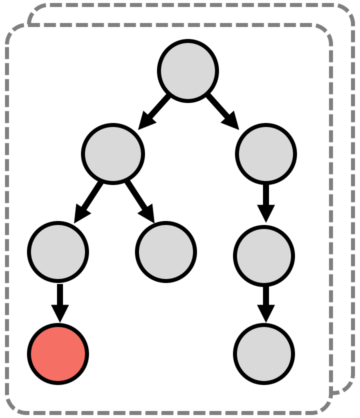
One-time preparation step!



Data Collection for Training a Surrogate Model



Tranco's Top-10K Webpages



Graph representation

ATS blockers classify **network request nodes**!

Request URL	# of Nodes	URL Length	Semicolon	# of Cookie Read
http://ad.com/show_ad	97	21	FALSE	0
http://ad.com/track_user.js	97	27	FALSE	1
https://sec.com/logo.png	108	24	TRUE	0
https://acsac.org/favicon.ico	81	29	TRUE	3

Extracted features

Target ATS Blocker

Request URL	# of Nodes	URL Length	Semicolon	# of Cookie Read	Label
http://ad.com/show_ad	97	21	FALSE	0	ATS
http://ad.com/track_user.js	97	27	FALSE	1	ATS
https://sec.com/logo.png	108	24	TRUE	0	Non-ATS
https://acsac.org/favicon.ico	81	29	TRUE	3	Non-ATS

Labeled features

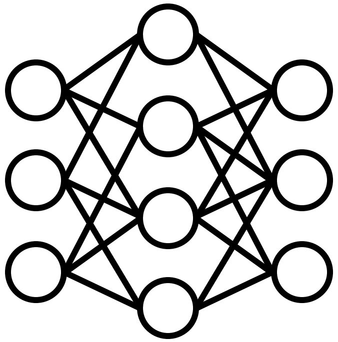
Surrogate Model Training

- We selected a four-layer feed-forward neural network as our surrogate.

Request URL	# of Nodes	URL Length	Semicolon	# of Cookie Read	Label
http://ad.com/show_ad	97	21	FALSE	0	ATS
http://ad.com/track_user.js	97	27	FALSE	1	ATS
https://sec.com/logo.png	108	24	TRUE	0	Non-ATS
https://acsac.org/favicon.ico	81	29	TRUE	3	Non-ATS

Data instances

Train



Surrogate model (DNN)

UAP Generation

# of Nodes	URL Length	Semicolon	# of Cookie Read	Label
97	21	FALSE	0	ATS
97	27	FALSE	1	ATS

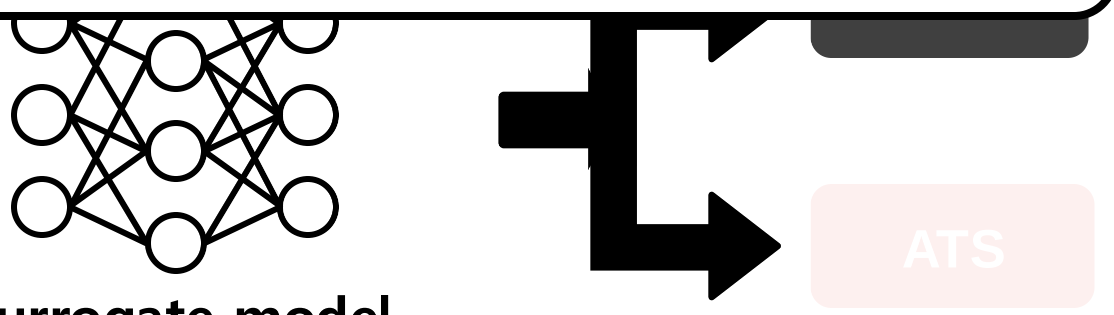
Sampled 40K ATS instances (x)

➕

# of Nodes	URL Length	Semicolon	# of Cookie Read
+0	+0	FALSE	+0

Universal Adversarial Perturbation (δ)

We optimize the perturbation over **<u>all these instances</u>** to make it universal!



Surrogate model (DNN)

UAP Generation

- **Optimization goal**

  1) Bypass the target ATS blocker with a single perturbation

$$\underset{\delta}{\operatorname{argmax}} \; \boldsymbol{E}_{(x,y)\sim D_{ATS}}[L_{CE}(f'(\theta, x+\delta), ATS)]$$

δ: Perturbation

$f'(\theta)$: Surrogate model (DNN)

D_{ATS}: Data instances labeled as ATS

L_{CE} : Cross-Entropy Loss

KAIST Web Security & Privacy Lab Oregon State University SAIL

UAP Generation

- **Optimization goal**
  1) Bypass the target ATS blocker with a single perturbation
  2) Minimize the breakage risk of manipulating each feature

How can we achieve this goal?

$$\underset{\delta}{\operatorname{argmax}} \ \boldsymbol{E}_{(x,y)\sim D_{ATS}}[L_{CE}(f'(\theta, x+\delta), ATS)]$$

δ: Perturbation

D_{ATS}: Data instances labeled as ATS

$f'(\theta)$: Surrogate model (DNN)

L_{CE} : Cross-Entropy Loss

UAP Generation

- **Optimization goal**
  1) Bypass the target ATS blocker with a single perturbation
  2) Minimize the breakage risk of manipulating each feature

Simply minimizing the perturbation size is **insufficient**!

$$\underset{\delta}{\operatorname{argmax}} \; \boldsymbol{E}_{(x,y)\sim D_{ATS}}[L_{CE}(f'(\theta, x+\delta), ATS)] - \|\delta\|$$

δ: Perturbation

D_{ATS}: Data instances labeled as ATS

$f'(\theta)$: Surrogate model (DNN)

L_{CE} : Cross-Entropy Loss

Cost Model

- **Optimization goal**
  1) Bypass the target ATS blocker with a single perturbation
  2) Minimize the breakage risk of manipulating each feature

Considered **a web-specific cost**!

$$\underset{\delta}{\operatorname{argmax}} \; \boldsymbol{E}_{(x,y)\sim D_{ATS}}\left[L_{CE}\left(f'(\theta, x+\delta), ATS\right)\right] - C \cdot \|\delta\|$$

δ: Perturbation	$f'(\theta)$: Surrogate model (DNN)
D_{ATS}: Data instances labeled as ATS	L_{CE} : Cross-Entropy Loss

KAIST Web Security & Privacy Lab Oregon State University SAIL

Cost Model

Prioritize with **a lower cost value**!

Perturbation	Assigned Cost
URL_LENGTH	0.2

Cost model

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
</body>
```

Original webpage

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad?1234'>
</body>
```

Increased the URL length

☺ **Preserves the functionality!**

Cost Model

Perturbation	Assigned
URL_LENGTH	0.2
PARENT_TAG_NAME	3

Cost model

Deprioritize with **a higher cost value**!

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
</body>
```

Original webpage

HTML

```
<script>
  <iframe src = 'http://ad.com/show_ad'>
</script>
```

Changed its parent tag name

Breaks the functionality!

Cost Model

Perturbation	Assigned Cost
URL_LENGTH	0.2
PARENT_TAG_NAME	3
…	…

Cost model

The cost model represents
<u>a relative risk of manipulating it</u>.

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
</body>
```

Original webpage

HTML

```
<script>
  <iframe src = 'http://ad.com/show_ad'>
</script>
```

Changed its parent tag name

⚠ **Breaks the functionality!**

HTML/JS Manipulation

# of Nodes	URL Length	Semicolon	# of Cookie Read
+2	+5	TRUE	+4

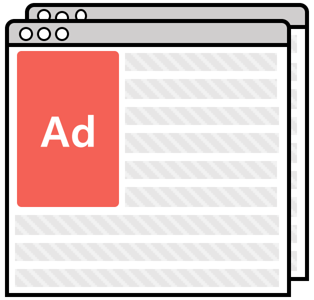
Universal Adversarial Perturbation (δ)

# of Nodes	URL Length	Semicolon	# of Cookie Read
97	21	FALSE	0
97	27	FALSE	1

Target ATS instances (x)

# of Nodes	URL Length	Semicolon	# of Cookie Read
99	26	TRUE	4
99	32	TRUE	5

UAP-injected ATS instances ($x + \delta$)



Webpages

Manipulated Webpages

HTML/JS Manipulation

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
  …
  <script src = 'http://ad.com/track_user.js'>
  …

</body>
```

track_user.js

```
…
// Tracking Users
user_cookie = document.cookie;
```

Webpage

Target network request!

# of Nodes	URL Length	Semicolon	# of Cookie Read
+2	+5	TRUE	+4

UAP

KAIST Web Security & Privacy Lab Oregon State University SAIL

HTML/JS Manipulation

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
  …
  <script src = 'http://ad.com/track_user.js'>
  …

</body>
```

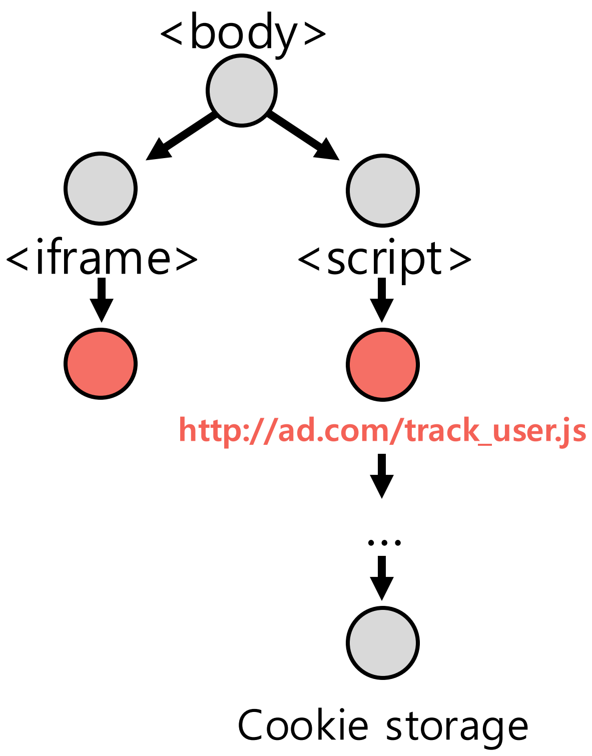
track_user.js

```
…
// Tracking Users
user_cookie = document.cookie;
```

Webpage

# of Nodes	URL Length	Semicolon	# of Cookie Read
+2	+5	TRUE	+4

UAP



Graph representation

HTML/JS Manipulation

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
  …
  <script src = 'http://ad.com/track_user.js'>
  …

</body>
```

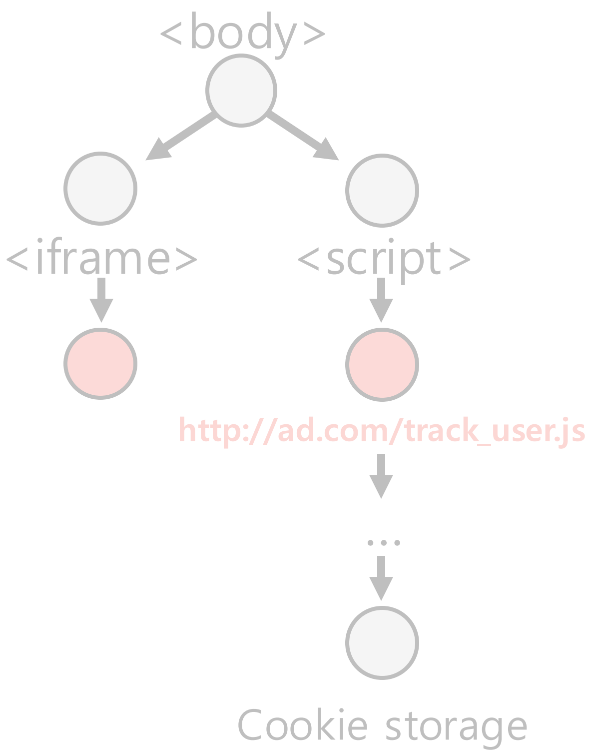
track_user.js

```
…
// Tracking Users
user_cookie = document.cookie;
```

Webpage

# of Nodes	URL Length	Semicolon	# of Cookie Read
+2	+5	TRUE	+4

UAP



Graph representation

HTML/JS Manipulation

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
  …
  <script src = 'http://ad.com/track_user.js'>
  …
  <div hidden=""></div>
  <p hidden=""></p>
</body>
```

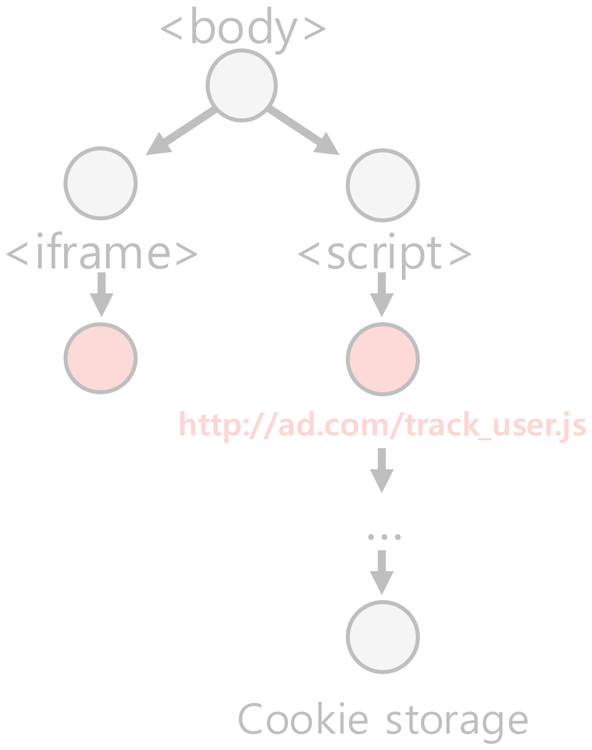
track_user.js

```
…
// Tracking Users
user_cookie = document.cookie;
```

Webpage

# of Nodes	URL Length	Semicolon	# of Cookie Read
+2	+5	TRUE	+4

UAP



Graph representation

HTML/JS Manipulation

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
  …
  <script src = 'http://ad.com/track_user.js'>
  …
  <div hidden=""></div>
  <p hidden=""></p>
</body>
```

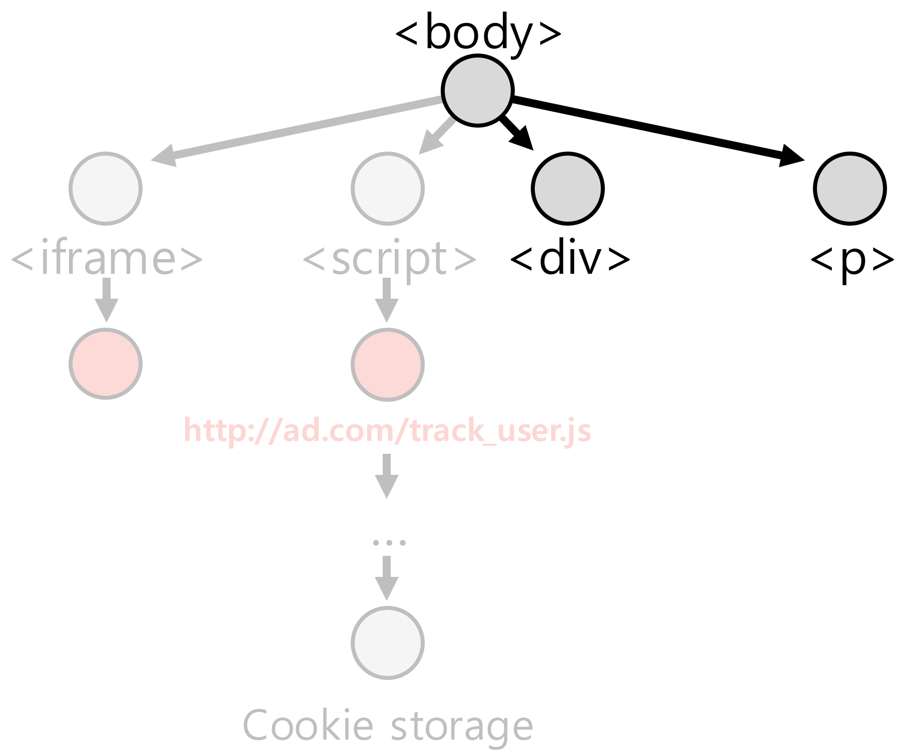
track_user.js

```
…
// Tracking Users
user_cookie = document.cookie;
```

Webpage

# of Nodes	URL Length	Semicolon	# of Cookie Read
+2	+5	TRUE	+4

UAP



Graph representation

HTML/JS Manipulation

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
  …
  <script src = 'http://ad.com/track_user.js?123;'>
  …
  <div hidden=""></div>
  <p hidden=""></p>
</body>
```

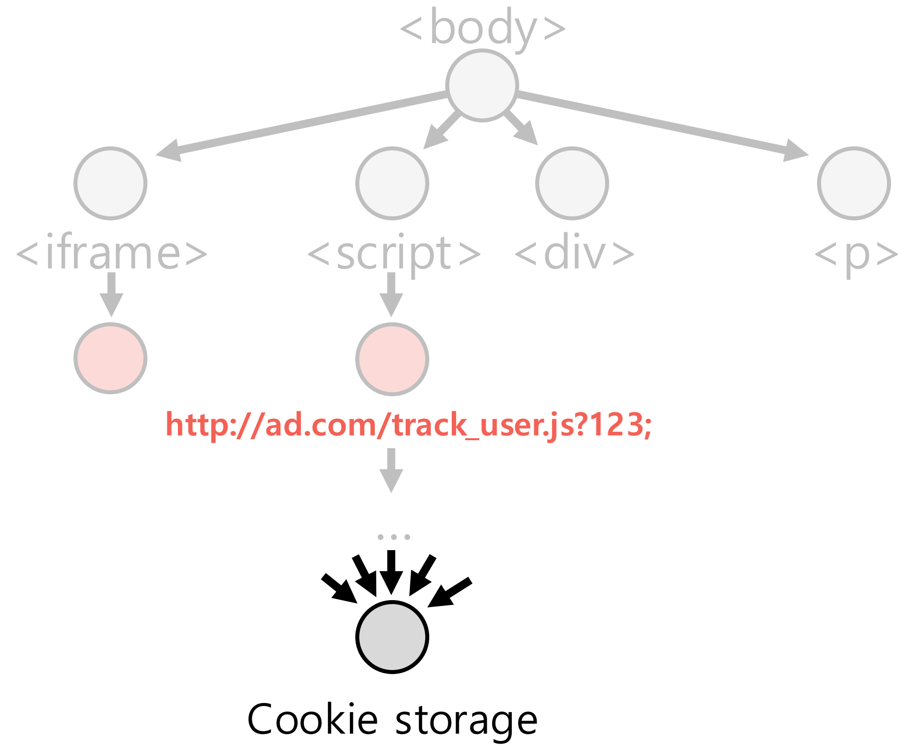
track_user.js

```
…
// Tracking Users
user_cookie = document.cookie;
for (let i = 1; i <= 4; i++) {
  getCookie();
}
```

Webpage

# of Nodes	URL Length	Semicolon	# of Cookie Read
+2	+5	TRUE	+4

UAP



Graph representation

HTML/JS Manipulation

HTML

```
<body>
  <iframe src = 'http://ad.com/show_ad'>
  …
```

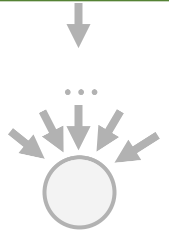
# of Nodes	URL Length	Semicolon	# of Cookie Read
+2	+5	TRUE	+4

UAP

YOPO implements 46 HTML/JS manipulation functions!

```
…
// Tracking Users
user_cookie = document.cookie;
for (let i = 1; i <= 4; i++) {
  getCookie();
}
```

Webpage



Graph representation

Experimental Setup

- Target ATS blockers
  1) **AdGraph** [S&P '20]
  2) **WebGraph*** [Security '22]
  3) **AdFlush** [WWW '24]
  4) **PageGraph**** [WWW '20]

  * We used all content, structural, and flow features for WebGraph.

  ** We revised PageGraph to support all ATS resource types.
- We measured **attack success rate (ASR)** against 2,000 target ATS requests.

Attack Success Rate

- ASRs measured against target ATS blockers

ATS blockers	Attack success rate (%)
AdGraph	89.27
WebGraph	71.21
AdFlush	61.91
PageGraph	84.16

Recent ML-based ATS blockers are **vulnerable to universal attacks** using a single perturbation!

Attack Success Rate

- ASRs measured against target ATS blockers

Adversaries can launch attacks against these ATS blockers <u>at scale</u>!

PageGraph	84.16

Recent ML-based ATS blockers are **<u>vulnerable to universal attacks</u>** using a single perturbation!

Attack Success Rate

- ASRs measured against target ATS blockers

Where does this vulnerability stem from?

PageGraph	84.16

Recent ML-based ATS blockers are **vulnerable to universal attacks** using a single perturbation!

Top-5 Most Influential Features

Features	Type	ASR drop (↓)
PARENT_ATTR_ASYNC	Binary	**-19.87%**
SEMICOLON_IN_URL	Binary	**-8.28%**
PARENT_ATTR_DEFER	Binary	**-5.83%**
DOMAIN_NAME_IN_QS	Binary	**-5.52%**
URL_LENGTH	Numerical	**-1.89%**

Top-5 most influential features for attacking AdGraph

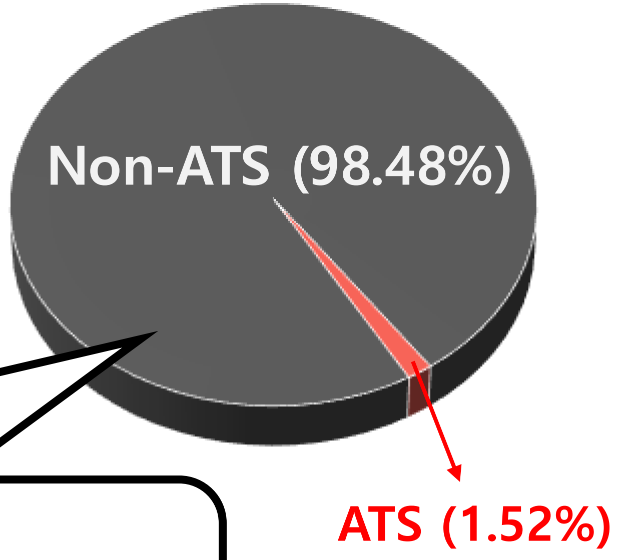
Top-5 Most Influential Features

Features	Type	ASR drop (↓)
PARENT_ATTR_ASYNC	**Binary**	**-19.87%**
SEMICOLON_IN_URL	**Binary**	**-8.28%**
PARENT_ATTR_DEFER	**Binary**	**-5.83%**
DOMAIN_NAME_IN_QS	**Binary**	**-5.52%**
URL_LENGTH	Numerical	**-1.89%**

When applying a UAP, YOPO overwrites binary features to have **a specific combination of values**!

Top-5 Most Influential Features

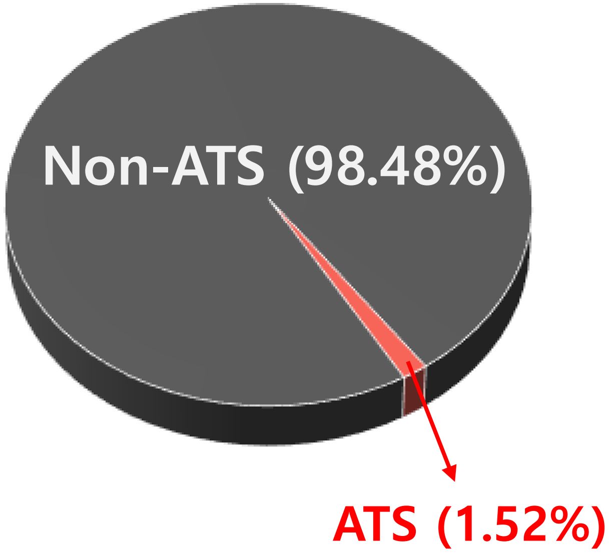
Features	UAP Values	ASR drop (↓)
PARENT_ATTR_ASYNC	TRUE	-19.87%
SEMICOLON_IN_URL	TRUE	-8.28%
PARENT_ATTR_DEFER	FALSE	-5.83%
DOMAIN_NAME_IN_QS	FALSE	-5.52%
URL_LENGTH	+9	[illegible]



<u>98.48%</u> of network requests that have **<u>this combination</u>** in our training set are **<u>non-ATS</u>**!

Top-5 Most Influential Features

Features	UAP Values	ASR drop (↓)
PARENT_ATTR_ASYNC	TRUE	-19.87%
SEMICOLON_IN_URL	TRUE	-8.28%
PARENT_ATTR_DEFER	FALSE	-5.83%
DOMAIN_NAME_IN_QS	FALSE	-5.52%
URL_LENGTH	+9	-1.89%



PARENT_ATTR_ASYNC	SEMICOLON_IN_URL	PARENT_ATTR_DEFER	DOMAIN_NAME_IN_QS
FALSE	FALSE	FALSE	TRUE

ATS

↓

PARENT_ATTR_ASYNC	SEMICOLON_IN_URL	PARENT_ATTR_DEFER	DOMAIN_NAME_IN_QS
TRUE	TRUE	FALSE	FALSE

Non-ATS

Top-5 Most Influential Features

Features	UAP Values	ASR drop (↓)
PARENT_ATTR_ASYNC	TRUE	-19.87%

Non-ATS (98.48%)

This arises from the inherent imbalance of binary features in real-world webpages!

FALSE	FALSE	FALSE	TRUE	ATS

PARENT_ATTR_ASYNC	SEMICOLON_IN_URL	PARENT_ATTR_DEFER	DOMAIN_NAME_IN_QS	
TRUE	TRUE	FALSE	FALSE	Non-ATS

Mitigation #1: Nullifying Binary Features

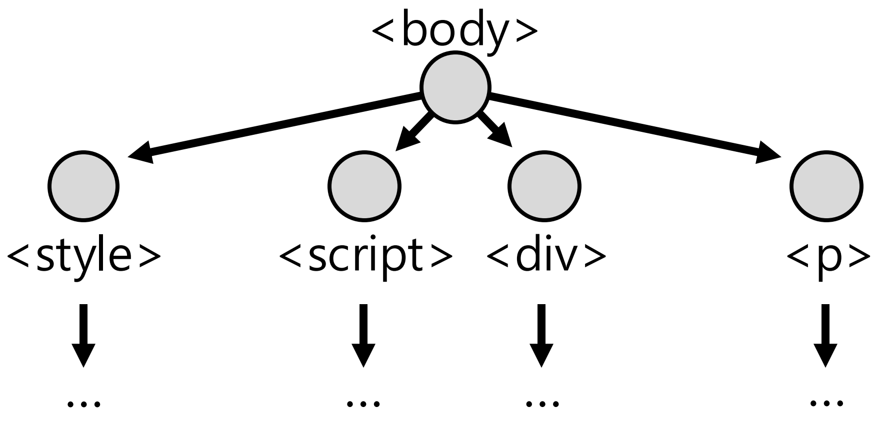
- We **nullified binary feat** … s
  when training each ATS …

Reduced the ASR **by at most 27.52%**
without any performance drop!

ATS blockers	ASR	Accuracy	Precision	Recall
AdGraph	61.75 **(27.52↓)**	92.15 (0.49↓)	89.11 (0.79↓)	84.43 (0.87↓)
WebGraph	63.90 **(7.31↓)**	95.39 (0.29↓)	92.52 (0.74↓)	92.08 (0.19↓)
AdFlush	49.82 **(12.09↓)**	95.79 (0.14↓)	93.99 (0.35↓)	90.77 (0.10↓)
PageGraph	70.28 **(13.88↓)**	95.78 (0.11↓)	92.66 (0.24↓)	93.06 (0.39↓)

Mitigation #2: Misleading Perturbation Directions

- HTML manipulation **<u>decreasing feature values</u>** is more likely to **<u>break webpages</u>**.



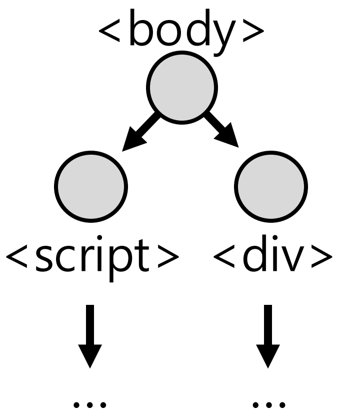
Graph representation

# of Nodes	URL Length	Semicolon	# of Cookie Read
-2	+5	TRUE	+4

UAP

Mitigation #2: Misleading Perturbation Directions

- HTML manipulation **decreasing feature values** is more likely to **break webpages**.



Graph representation

Breaks the functionality!

# of Nodes	URL Length	Semicolon	# of Cookie Read
-2	+5	TRUE	+4

UAP

Mitigation #2: Misleading Perturbation Directions

- HTML manipulation **decreasing feature values** is more likely to **break webpages**.

- We **preprocessed input features** before training ATS blockers, thus **misleading perturbations to decrease feature values**.

- As a result, adversaries **cannot reflect such manipulation** at an HTML level.

Mitigation #2: Misleading Perturbation Directions

Applying both mitigation strategies reduced the ASR **by at most 48.86%** without any performance drop!

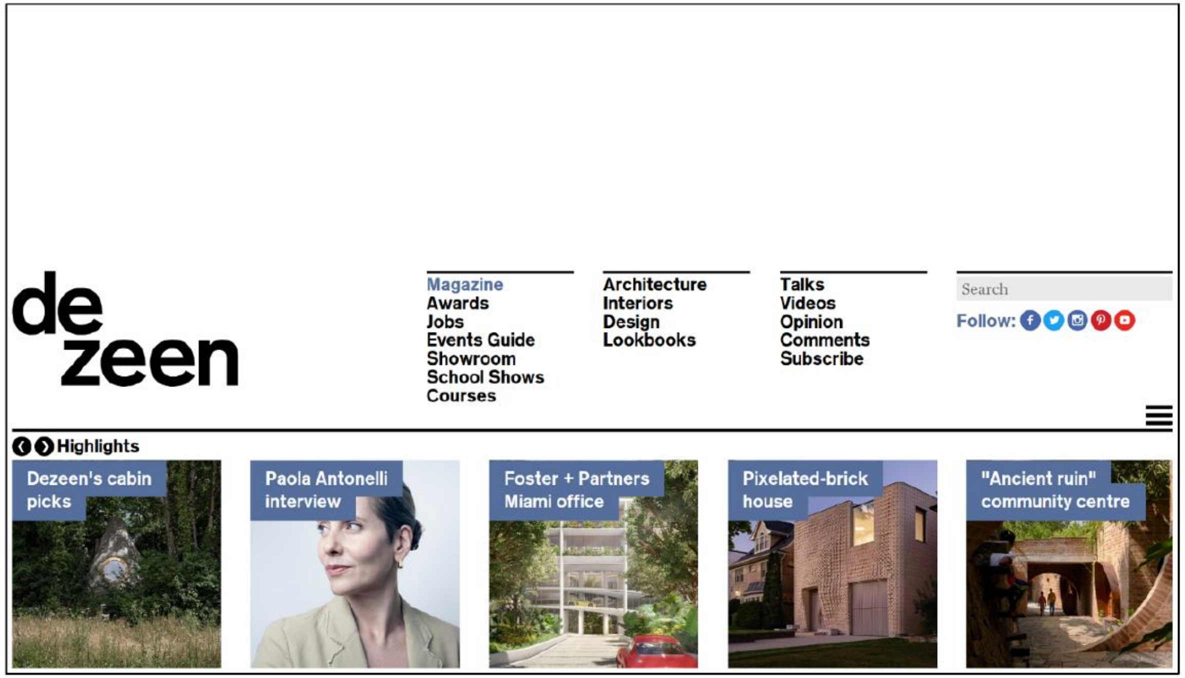
ATS blockers	ASR	Accuracy	Precision	Recall
AdGraph	40.41 **(48.86 ↓)**	91.59 (1.05 ↓)	85.49 (4.41 ↓)	87.05 (1.75 ↑)
WebGraph	48.55 **(22.66 ↓)**	95.19 (0.49 ↓)	91.64 (1.62 ↓)	92.38 (0.11 ↑)
AdFlush	42.74 **(19.17 ↓)**	95.68 (0.25 ↓)	95.00 (0.34 ↓)	90.34 (0.53 ↓)
PageGraph	64.51 **(19.65 ↓)**	95.74 (0.15 ↓)	92.59 (0.31 ↓)	93.26 (0.19 ↓)

Breakage Analysis

- We manually inspected 400 webpages manipulated by YOPO.
- We defined 4 breakage types following prior studies.
- **Only 14 webpages out of 400** exhibited functionality disruption.

Breakage Analysis

Successfully bypassed AdGraph!



Original webpage

Manipulated webpage

No functional breakage

For More Details

- Case study
- Effect of the attack hyperparameters
- Attacking multiple requests
- Different cost models
- https://github.com/WSP-LAB/YOPO

Question?

If you have more questions, please email dongwon.shin@kaist.ac.kr