Continuous and Non-Intrusive Reauthentication of Web Sessions based on Mouse Dynamics

Eric Medvet Alberto Bartoli Francesca Boem Fabiano Tarlao

Department of Engineering and Architecture
University of Trieste
Italy

September 10th, 2014

http://machinelearning.inginf.units.it

Table of Contents

MACHINE LEARNING LAB

(Re)Authentication

Credentials stealing is *not* an exceptional event

- *bad* current user with *good* credentials, possibly for a long time

(Re)Authentication

Credentials stealing is *not* an exceptional event

- *bad* current user with *good* credentials, possibly for a long time

$\rightarrow$ verify the user identity over the time

(Re)Authentication

Credentials stealing is *not* an exceptional event

- *bad* current user with *good* credentials, possibly for a long time

→ verify the user identity over the time

- by other means than credentials
- possibly non-intrusively

Behavioral biometrics

Non-intrusive continuous verification of the user identity
$\rightarrow$ Behavioral biometrics:

- keystrokes
- ...
- mouse trajectories

MACHINE LEARNING LAB

Behavioral biometrics

Non-intrusive continuous verification of the user identity
$\rightarrow$ Behavioral biometrics:

- keystrokes
- ...
- mouse trajectories

Behavioral biometrics

Non-intrusive continuous verification of the user identity
$\rightarrow$ Behavioral biometrics:

- keystrokes
- ...
- mouse trajectories

Machine instrumentation for collecting biometrics may be unpractical for large distributed organizations

MACHINE LEARNING LAB

Scenario

So, we are concerned in:

- continuous reauthentication
- using mouse dynamics
- collected w/o specific software installed on client machine

Scenario

We chose to address:

- web
- full transparency to server and client

Suitable for:

- large organizations w/ user web access
- (private) cloud hosted enterprise applications

Example

Large organizations w/ user web access:

Example

Large organizations w/ user web access:

1. X authenticates with Alice's credentials on her organization

Example

Large organizations w/ user web access:

1. X authenticates with Alice's credentials on her organization
2. X browses the web (any website) and...

Example

Large organizations w/ user web access:

1. X authenticates with Alice's credentials on her organization
2. X browses the web (any website) and...
3. ...if X's behaviour is different enough from Alice's known behavior, an alert is eventually raised

MACHINE LEARNING LAB

Example

Large organizations w/ user web access:

1. X authenticates with Alice's credentials on her organization
2. X browses the web (any website) and...
3. ...if X's behaviour is different enough from Alice's known behavior, an alert is eventually raised

Authentication

Example

Large organizations w/ user web access:

1. X authenticates with Alice's credentials on her organization
2. X browses the web (any website) and...
3. ...if X's behaviour is different enough from Alice's known behavior, an alert is eventually raised

Authentication, then reauthentication in the web using mouse dynamics.

Example

Large organizations w/ user web access:

1. X authenticates with Alice's credentials on her organization
2. X browses the web (any website) and...
3. ...if X's behaviour is different enough from Alice's known behavior, an alert is eventually raised

Authentication, then reauthentication in the web using mouse dynamics. Aim at detecting long lasting systematic fraudulent account usage (*defense-in-depth*).

Table of Contents

Our contribution

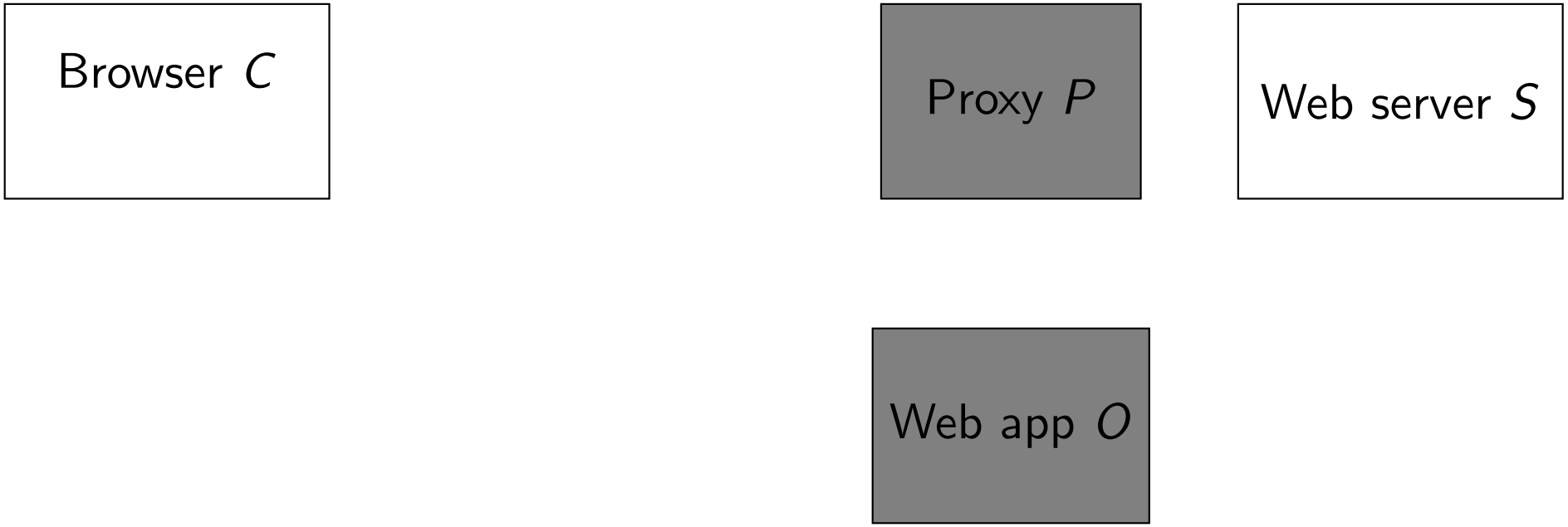
In a nutshell:

- a system for capturing web GUI-related events transparent for user and web site
- a procedure for performing continuous reauthentication using mouse-generated events

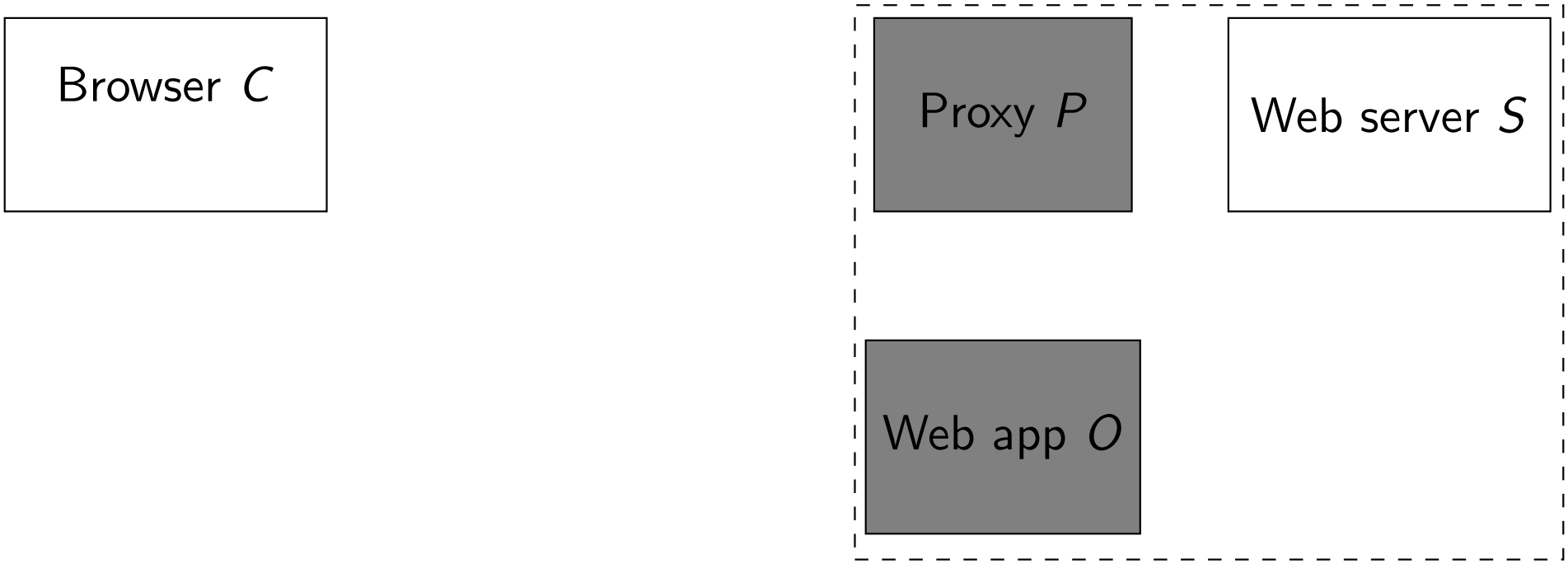
Data capture system: overview

- a web proxy
- a js (collects data)
- a web app (receives and analyzes data)

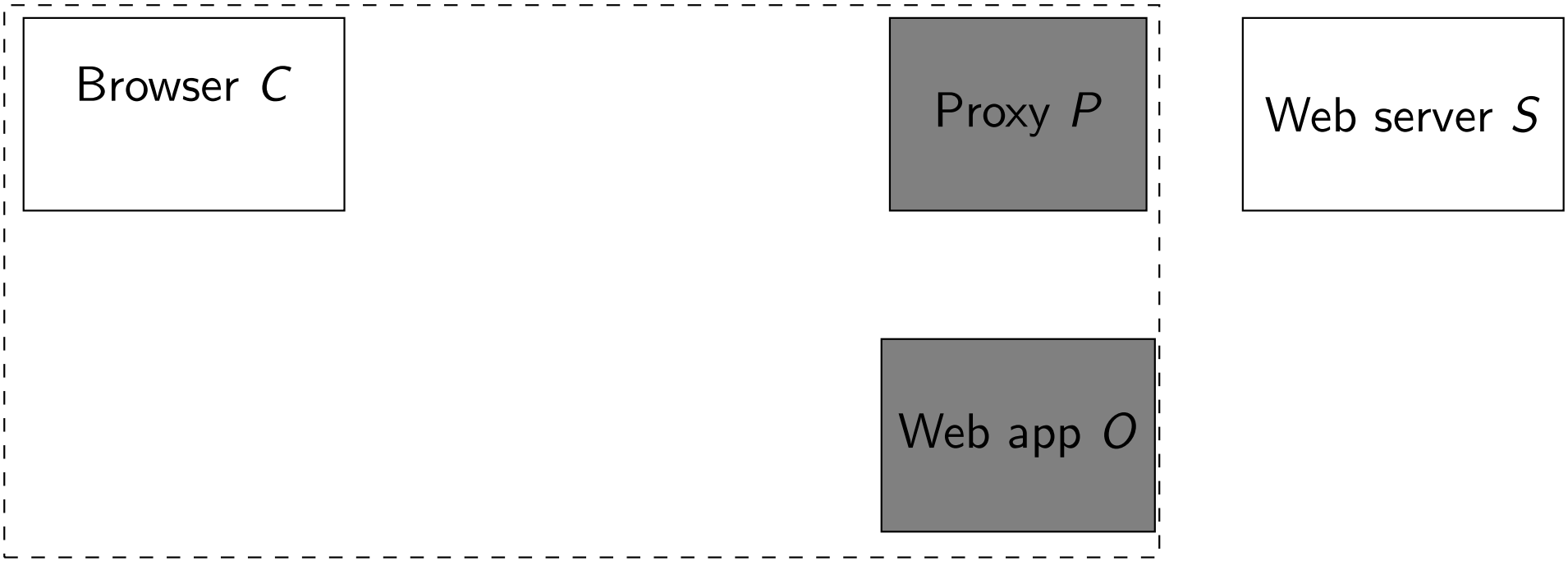
How it works



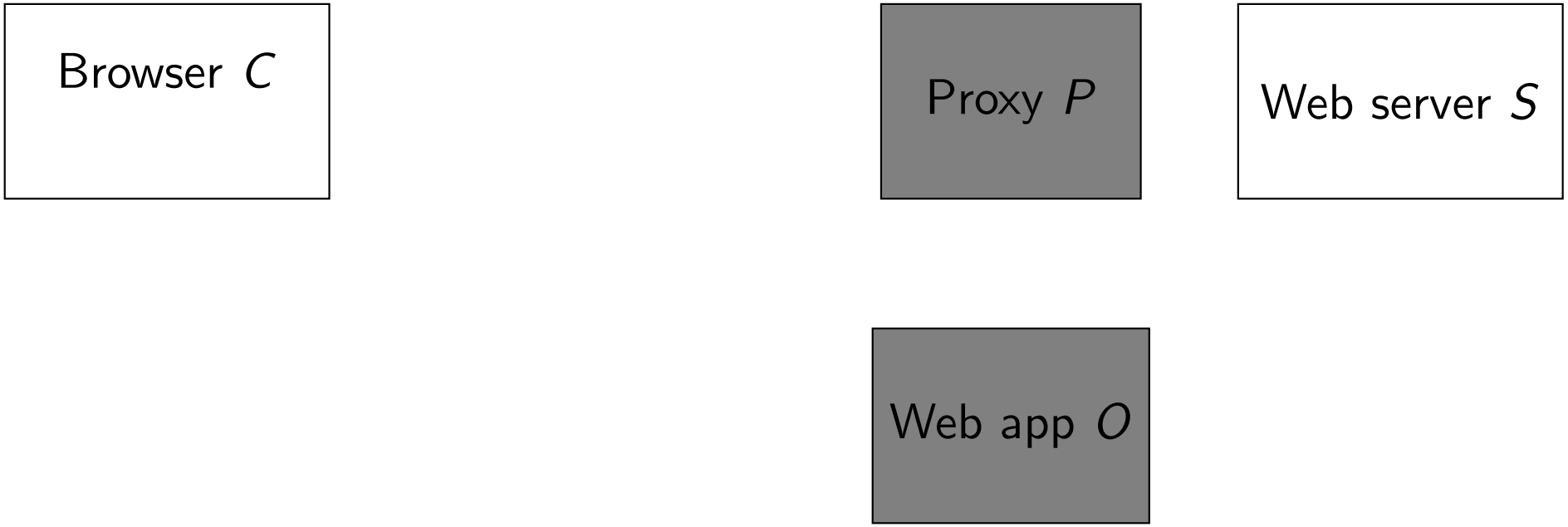
How it works



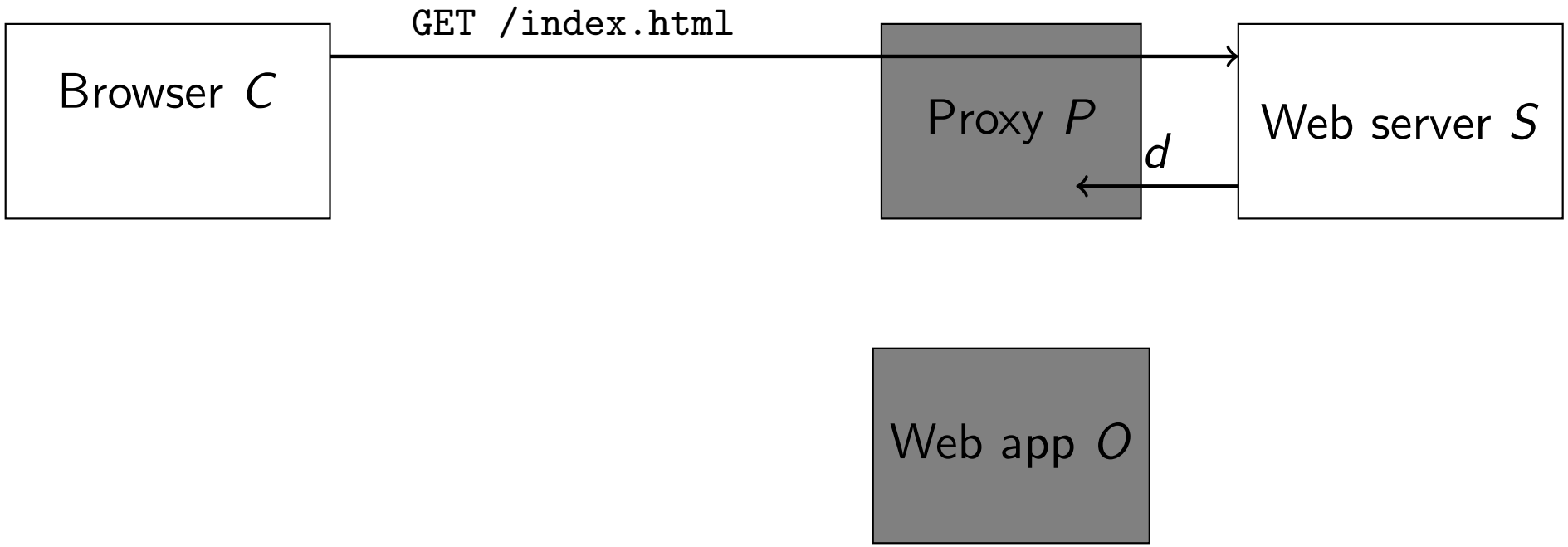
How it works



How it works

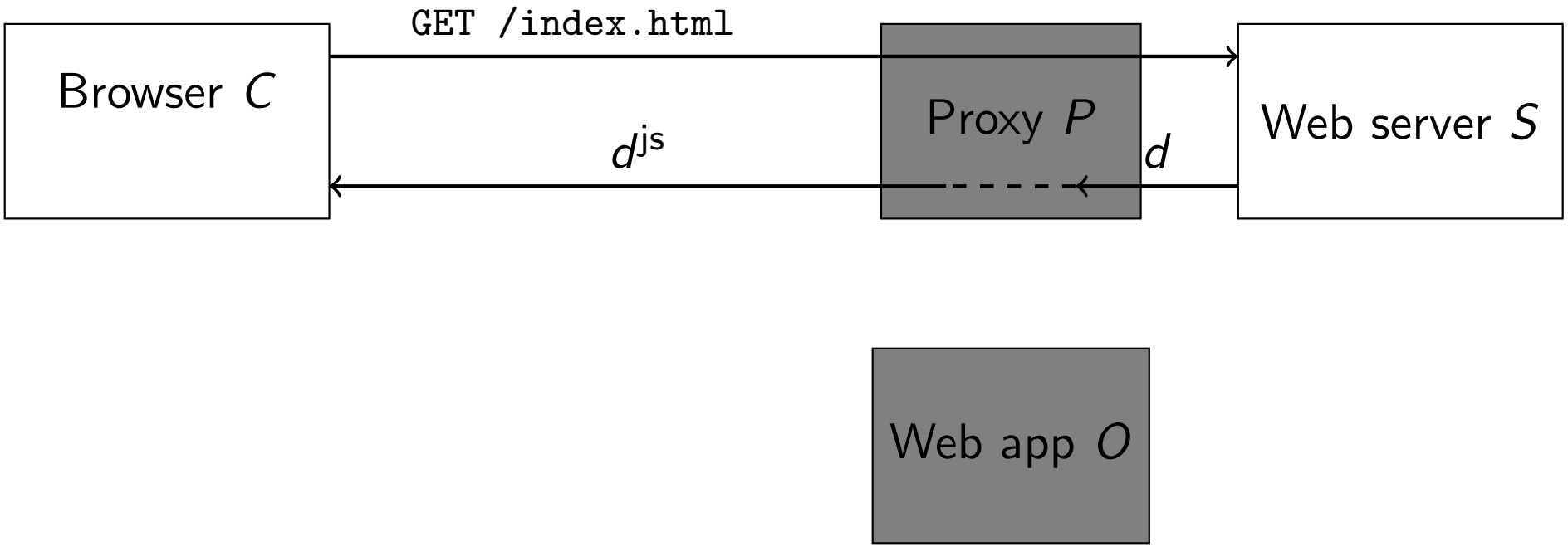


How it works



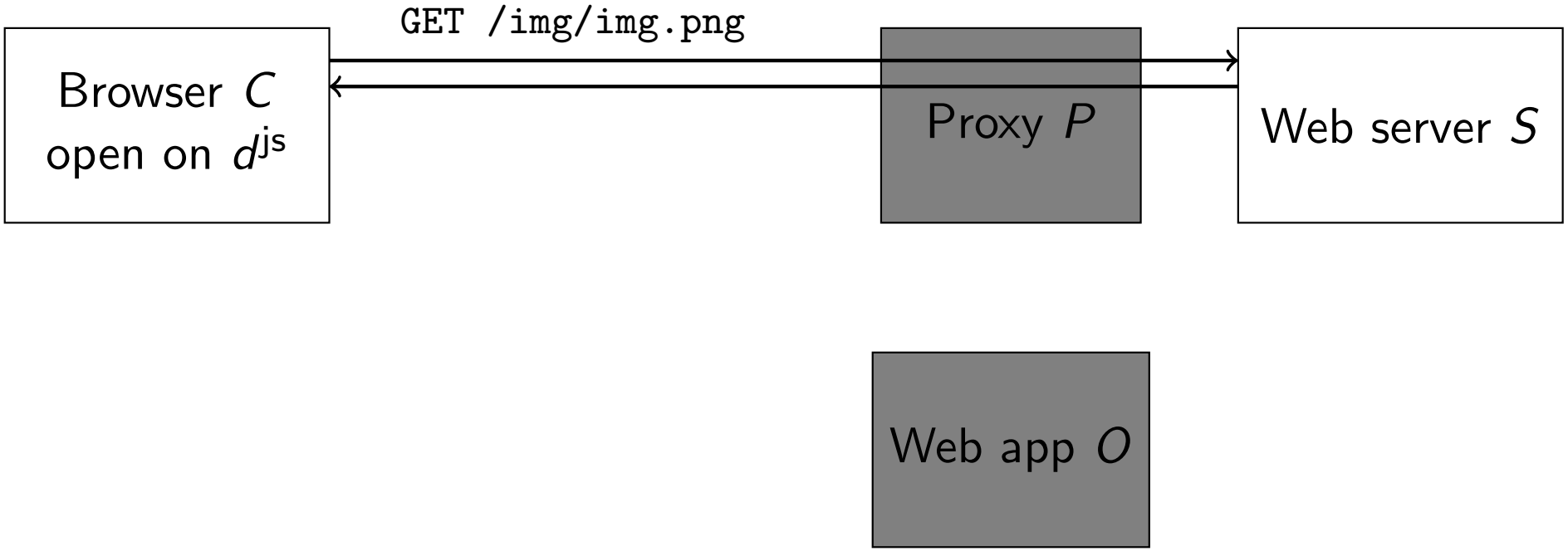
1. C requests HTML document to S, S responds with d

How it works



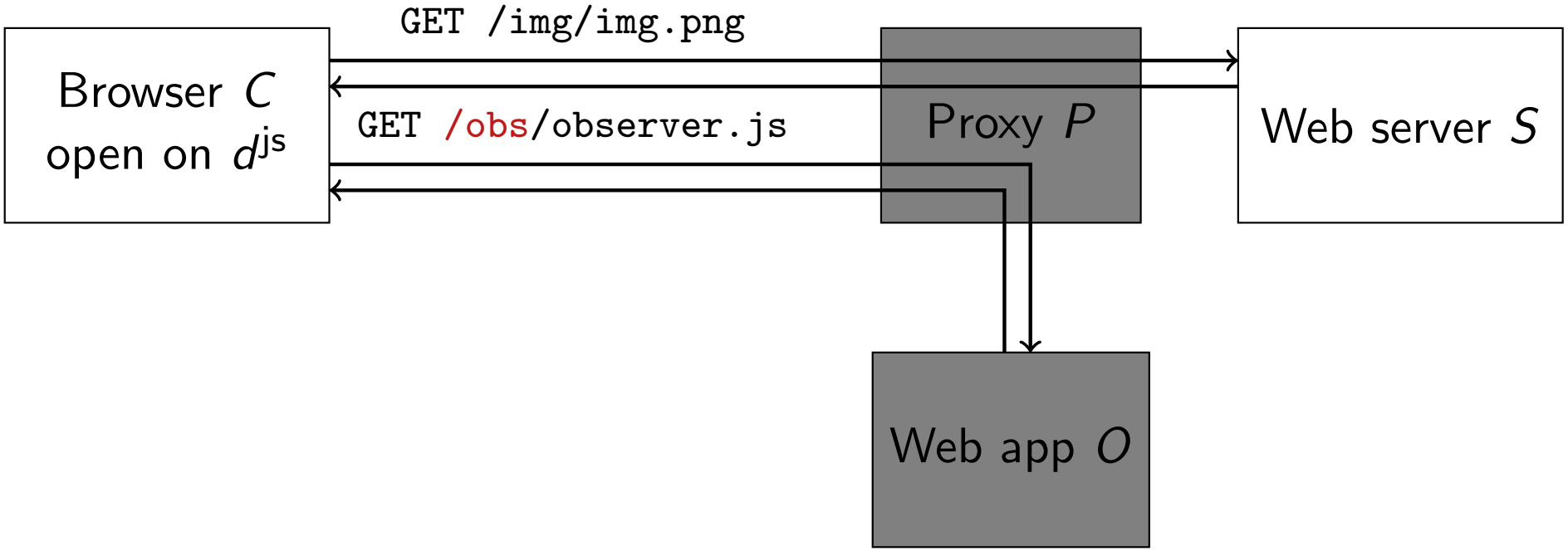
1. C requests HTML document to S, S responds with d
2. P injects in d our js URL (`src="/obs/observer.js"`)

How it works



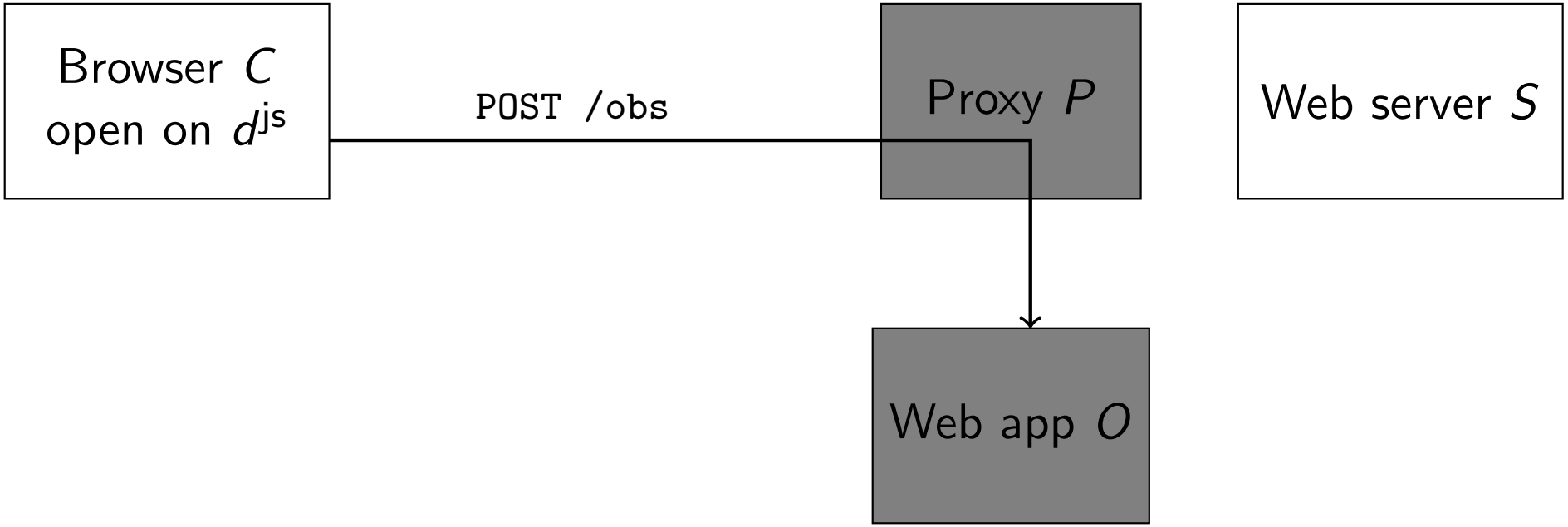
1. C requests HTML document to S, S responds with d
2. P injects in d our js URL (`src="/obs/observer.js"`)
3. C requests resources mentioned in d^{js}

How it works



1. C requests HTML document to S, S responds with d
2. P injects in d our js URL (`src="/obs/observer.js"`)
3. C requests resources mentioned in d^{js}: our js comes from O (rather than S) via P

How it works



1. C requests HTML document to S, S responds with d
2. P injects in d our js URL (`src="/obs/observer.js"`)
3. C requests resources mentioned in d^{js}: our js comes from O (rather than S) via P
4. our js on C sends mouse events data to `/obs`, i.e., O

Data capture system

- fully transparent to both user and web sites, requires only to set the proxy
- redirection of `/obs/*` traffic allows to circumvent Same Origin Policy
- low bandwidth usage ($\approx 2.5\,\mathrm{kB\,s^{-1}}$)
- can work with HTTPS (w/ self-signed certificate)

1

Data capture system

- fully transparent to both user and web sites, requires only to set the proxy
- redirection of `/obs/*` traffic allows to circumvent Same Origin Policy
- low bandwidth usage ($\approx 2.5\,\mathrm{kB\,s^{-1}}$)
- can work with HTTPS (w/ self-signed certificate)
- could be used also for other purposes: web app testing[1], web app misuse detection, ...

[1] Bartoli, Medvet, Mauri, *Recording and Replaying Navigations on AJAX Web Sites*, Int. Conf. on Web Engineering (ICWE), 2012

MACHINE LEARNING LAB

Procedure: overview

Data capture system generates an event $e = (x, y, t)$ every $\approx 25\,\text{ms}$, then we:

1. split sequence of events on pauses $\geq 500\,\text{ms}$ and consider the last 10 events before a pause (*trajectory*)
2. transform a trajectory T into a vector $\mathbf{f}(T) \in \mathbb{R}^{39}$
3. classify $\mathbf{f}(T)$ as anomalous/normal, w.r.t. current authenticated user

MACHINE LEARNING LAB

Features

$\mathbf{f}(T)$ includes:

- directions and direction changes
- speeds
- accelerations
- x- and y-extents

Classification

Two phases:

- training
- actual classification

Classification

Two phases: (U^- is the authenticated user)

- training based on trajectories of U^- and other users $U_1^+, U_2^+, \ldots$
- actual classification

Classification

Two phases: (U^- is the authenticated user)

- training based on trajectories of U^- and other users $U_1^+, U_2^+, \ldots$
- actual classification based on trajectories of current unknown user U claiming to be U^-

Training phase

Once, at the beginning:

1. train a SVM_{U-} on the training set

Actual classification phase

For each T trajectory of U:

1. apply SVM_{U^-} to $\mathbf{f}(T)$
2. consider last w trajectories and...
3. ...if too many positives, raise an alert

Last w trajectories

Aggregation of several classifier outcomes:

- often used with mouse dynamics
- the higher w,
  - the higher the accuracy and
  - the longer the *Time to Detection* (TtD)

Table of Contents

Dataset

Two groups of users, each observed for several working days:

- 6 users, with different hardware equipment
- 18 users, with homogeneous hardware

Results

w	TtD (min)	Dataset 1 Acc.	Dataset 1 FAR	Dataset 1 FRR	Dataset 2 Acc.	Dataset 2 FAR	Dataset 2 FRR
50	13.5	83.3	16.6	16.7	76.4	21.8	25.4
100	27.1	88.5	12.8	10.2	81.4	17.5	19.6
200	54.1	93.5	9.2	3.8	86.6	13.5	13.3
350	94.7	95.6	7.9	1.0	90.6	10.8	8.0
500	135.3	96.5	6.1	0.8	92.2	9.5	6.1

Results

w	TtD (min)	Dataset 1 Acc.	Dataset 1 FAR	Dataset 1 FRR	Dataset 2 Acc.	Dataset 2 FAR	Dataset 2 FRR
50	13.5	83.3	16.6	16.7	76.4	21.8	25.4
100	27.1	88.5	12.8	10.2	81.4	17.5	19.6
200	54.1	93.5	9.2	3.8	86.6	13.5	13.3
350	94.7	95.6	7.9	1.0	90.6	10.8	8.0
500	135.3	96.5	6.1	0.8	92.2	9.5	6.1

- accuracy up to 96%

Results

w	TtD (min)	Dataset 1 Acc.	Dataset 1 FAR	Dataset 1 FRR	Dataset 2 Acc.	Dataset 2 FAR	Dataset 2 FRR
50	13.5	83.3	16.6	16.7	76.4	21.8	25.4
100	27.1	88.5	12.8	10.2	81.4	17.5	19.6
200	54.1	93.5	9.2	3.8	86.6	13.5	13.3
350	94.7	95.6	7.9	1.0	90.6	10.8	8.0
500	135.3	96.5	6.1	0.8	92.2	9.5	6.1

- accuracy up to 96%
- works better if attacker uses different hardware

Results

w	TtD (min)	Dataset 1 Acc.	Dataset 1 FAR	Dataset 1 FRR	Dataset 2 Acc.	Dataset 2 FAR	Dataset 2 FRR
50	13.5	83.3	16.6	16.7	76.4	21.8	25.4
100	27.1	88.5	12.8	10.2	81.4	17.5	19.6
200	54.1	93.5	9.2	3.8	86.6	13.5	13.3
350	94.7	95.6	7.9	1.0	90.6	10.8	8.0
500	135.3	96.5	6.1	0.8	92.2	9.5	6.1

- accuracy up to 96%
- works better if attacker uses different hardware
- time to detection of tens of minutes

Time to detection

Time to detection of tens of minutes: is it practical?

- fits the threat model
- we can only monitor web usage (browser)
  - user could unfocus the browser for minutes
  - we consider sessions without pauses ≥ 10 minutes

MACHINE LEARNING LAB

Thanks!