



OWASP

Open Web Application  
Security Project

# Breaking Fraud & Bot Detection Solutions

Mayank Dhiman  
Stealth Security



**Stealth**  
Security™

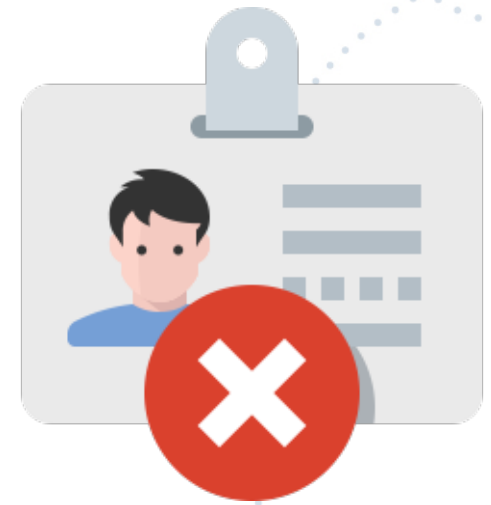
# Agenda

- Architectural Overview
- Threat Model
- Issues & Attacks
- Takeaways



# Fraud Detection

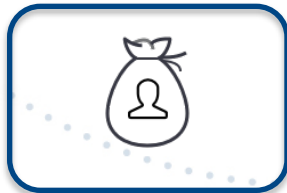
- Defend against fraudulent logins, payments etc.
- Look for anomalies in activity of a user, given past activity.



# Bot Detection

- Defend against bots trying to test credential dumps, scraping etc.
- Bot detection solutions look for anomalies across entire populations and time periods.

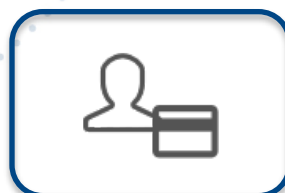
Account Take Over



Fake Accounts



PII / PHI Theft

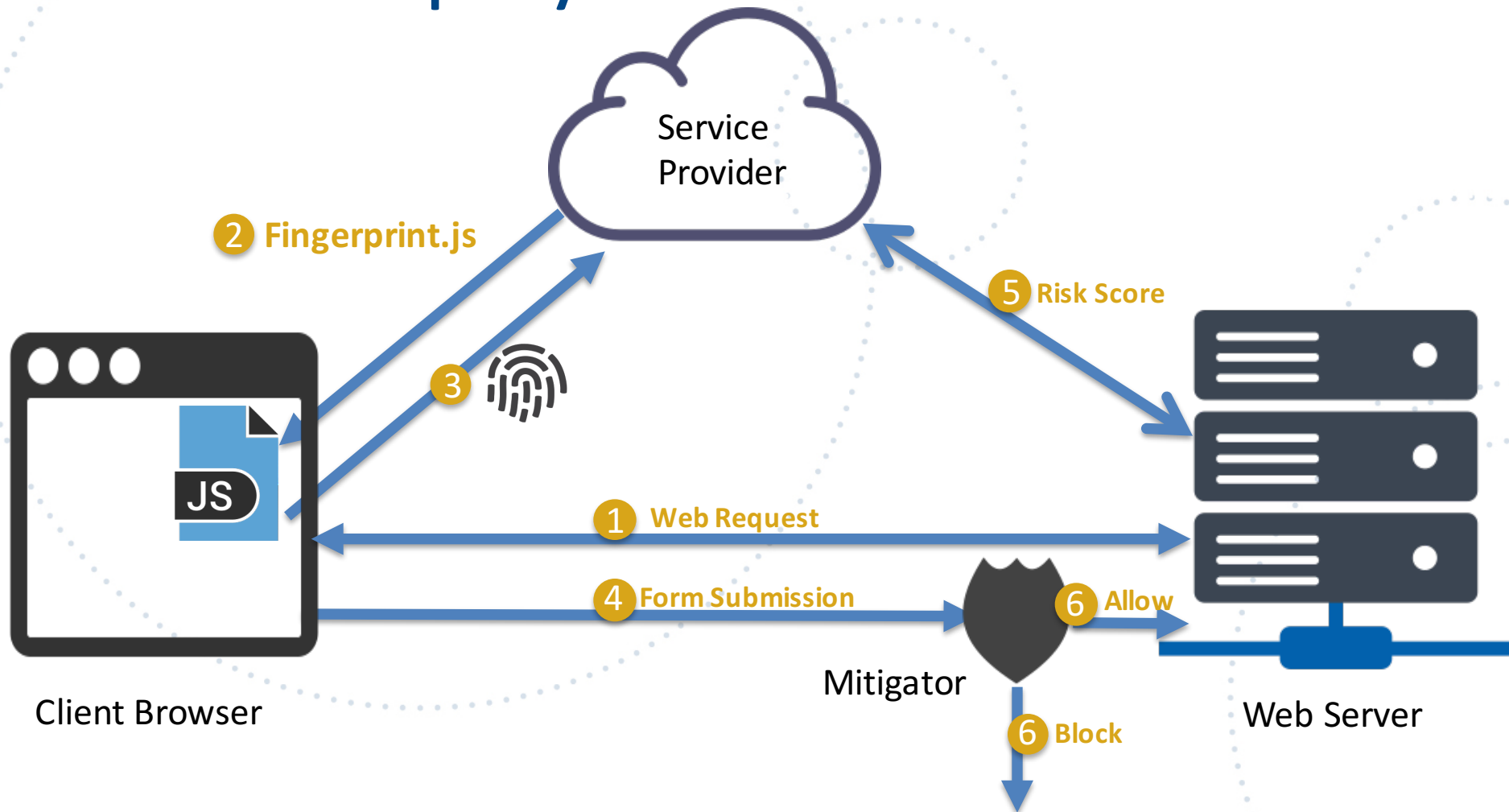


OWASP

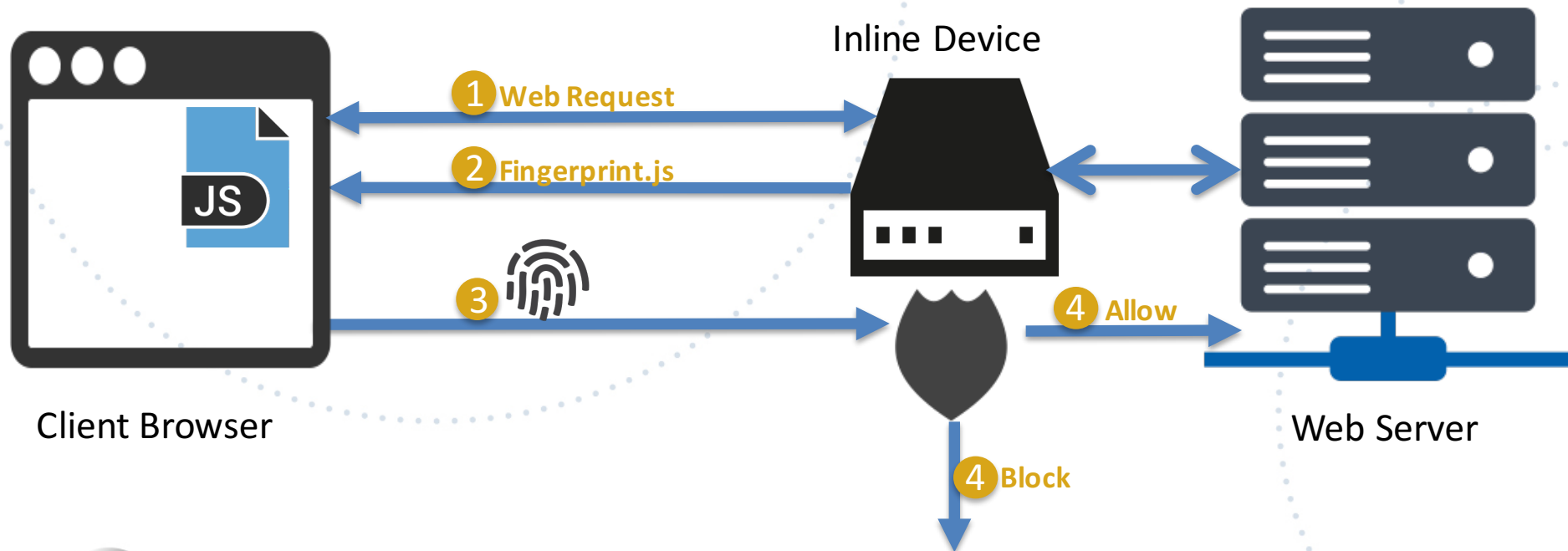
Open Web Application  
Security Project

[WWW.OWASP.ORG](http://WWW.OWASP.ORG)

# Cloud Deployment



# Inline Deployments



# Threat Model

- Attacker has full control over the browser.
- Attacker can craft requests and modify responses according to the responses from the web server.



# Fundamental Issue I

- Attacker can reverse engineer the entire sensor



# Browser Fingerprinting

Browser Characteristic	bits of identifying information	one in $x$ browsers have this value	value
Limited supercookie test	0.39	1.31	DOM localStorage: Yes, DOM sessionStorage: Yes, IE userData: No
Hash of canvas fingerprint	14.2	18792.03	d68a8387d4f25241664deb501fe16855
Screen Size and Color Depth	4.29	19.6	1440x900x24
Browser Plugin Details	9.25	610.61	Plugin 0: Chromium PDF Viewer; ; mhjfbmdgcfjbbpaeojofohoefgihjai; (; application/pdf; pdf). Plugin 1: Chromium PDF Viewer; Portable Document Format; internal-pdf-viewer; (Portable Document Format; application/x-google-chrome-pdf; pdf). Plugin 2: Native Client; ; internal-nacl-plugin; (Native Client Executable; application/x-nacl; ) (Portable Native Client Executable; application/x-pnacl; ).
Time Zone	5.66	50.46	480
DNT Header Enabled?	0.8	1.74	True
HTTP_ACCEPT Headers	7.22	148.8	text/html, */*; q=0.01 gzip, deflate en-US,en;q=0.8
Hash of WebGL fingerprint	18.06	272484.5	e3cea23259d3808d0672f33457a79450
Language	0.91	1.88	en-US
System Fonts	4.61	24.34	Andale Mono, Arial, Arial Black, Arial Hebrew, Arial Narrow, Arial Rounded MT Bold, Arial Unicode MS, Comic Sans MS, Courier, Courier New, Geneva, Georgia, Helvetica, Helvetica Neue, Impact, LUCIDA GRANDE, Microsoft Sans Serif, Monaco, Palatino, Tahoma, Times, Times New Roman, Trebuchet MS, Verdana, Wingdings, Wingdings 2, Wingdings 3 (via javascript)
Platform	3.06	8.33	MacIntel
User Agent	20.06	1089938.0	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2694.0 Safari/537.36
Touch Support	0.58	1.49	Max touchpoints: 0; TouchEvent supported: false; onTouchStart supported: false
Are Cookies Enabled?	0.2	1.15	Yes



<https://panopticlick.eff.org/>



**OWASP**

Open Web Application  
Security Project

WWW.OWASP.ORG

# Browser Fingerprinting

- Hardware
  - CPU Architecture & Device Memory
  - GPU Canvas Fingerprinting
  - Audio Stack Fingerprinting
- Software
  - UserAgent
  - OS Version
- Storage
  - LocalStorage
  - SessionStorage
- Display
  - Color Depth
  - Screen Size
- Browser Customizations
  - Fonts
  - Plugins
  - Codecs
  - Mime Types
  - Time zone
  - User Language
- Misc.
  - Floating point calculations
  - Add behavior/callbacks/objects to DOM to check a real JS execution engine



OWASP

Open Web Application  
Security Project

[WWW.OWASP.ORG](http://WWW.OWASP.ORG)

# Browser Fingerprinting (Fingerprintjs2)

Your browser fingerprint: **ecf9e4942d57d4067112531d4f97e4fc**

## Detailed information:

```
user_agent = Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2
language = en-US
color_depth = 24
device_memory = -1
pixel_ratio = 1
hardware_concurrency = 8
resolution = 1920,1080
available_resolution = 1920,1076
timezone_offset = 480
session_storage = 1
local_storage = 1
indexed_db = 1
open_database = 1
cpu_class = unknown
navigator_platform = MacIntel
do_not_track = unknown
regular_plugins = Chromium PDF Viewer::::application/pdf-pdf,Native Client::::application/x-nacl~,application/x-pnacl~
canvas = canvas winding:yes~canvas fp:data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAB9AAAADICAYAAACwGnoBAAAgA
webgl = data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAASwAAACWCAYAAABkW7XSAAAMz0lEQVR4Xu2dX4hkRxWHT/UseQiiPg
webgl_vendor = ATI Technologies Inc.~AMD Radeon R9 M370X OpenGL Engine
adblock = false
has_lied_languages = false
has_lied_resolution = false
has_lied_os = false
has_lied_browser = false
touch_support = 0,false,false
js_fonts = Andale Mono,Arial,Arial Black,Arial Hebrew,Arial Narrow,Arial Rounded MT Bold,Arial Unicode MS,Comic
```

<https://github.com/Valve/fingerprintjs2>



**OWASP**

Open Web Application  
Security Project

WWW.OWASP.ORG

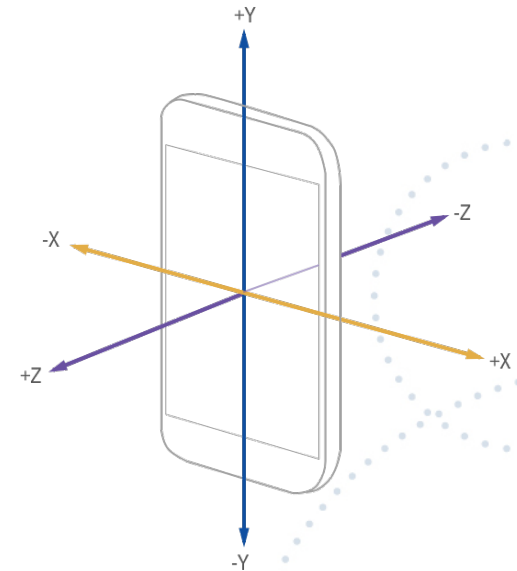
# User Behavior

- Mouse
  - Coordinates of where the mouse moved
  - Coordinates of clicks
- Keyboard
  - Stream of key presses
- Touchpad
  - Coordinates of where the screen was touched



# User Behavior

- Device Orientation
  - 3D angle of device whenever the orientation changes
- Device Position
  - Record speed of change of device's position.



Timing information along with event type can be used to create a very accurate picture of what interactions took place on the webpage.



OWASP

Open Web Application  
Security Project

[WWW.OWASP.ORG](http://WWW.OWASP.ORG)

# Anti-Tampering & Anti-Reversing

- JavaScript Obfuscation
- XOR based packed code
- Randomize name/location of the JavaScript file to load
- Dynamic Fields

# Payload

- Payload Encoding (Base64)
- Symmetric Encryption (DES)
- Custom Encryption Schemes



OWASP

Open Web Application  
Security Project

[WWW.OWASP.ORG](http://WWW.OWASP.ORG)

# Fundamental Issue II

- There are no guarantees of the correct execution of JavaScript



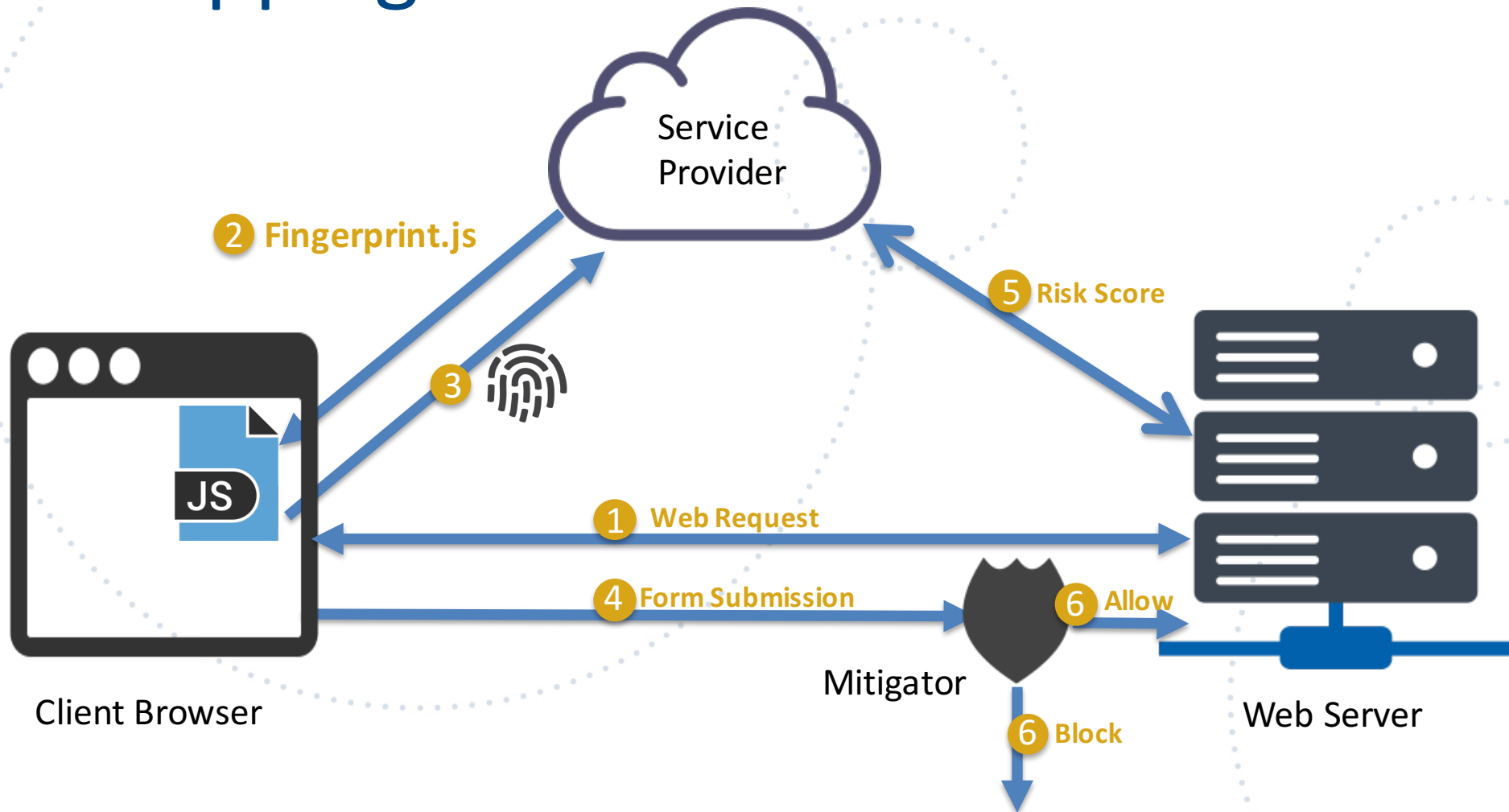


# Headless Browsers

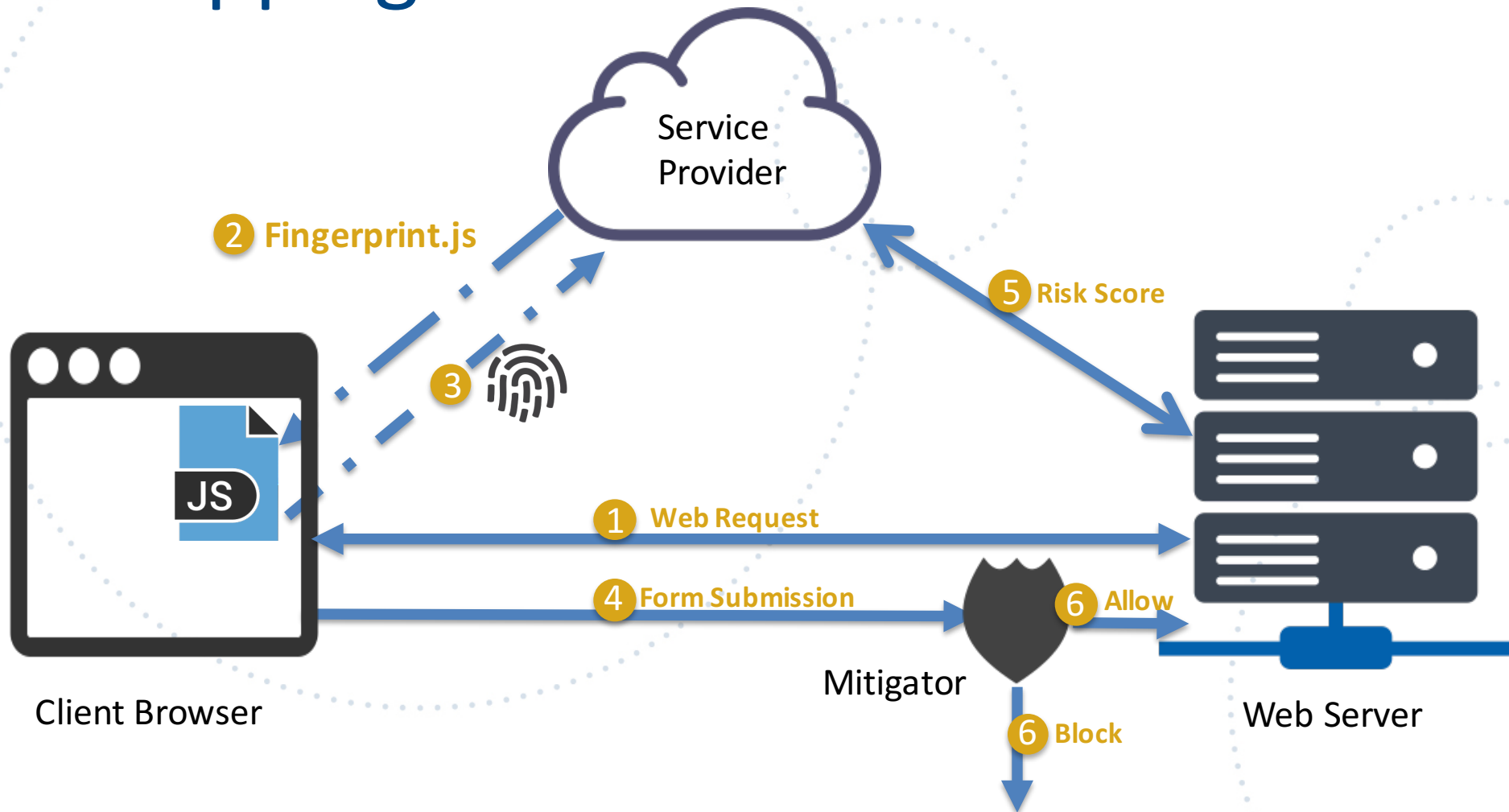
- Browser without a GUI, often used for automation and testing.
- Either render full JS or run JS in a virtual DOM.



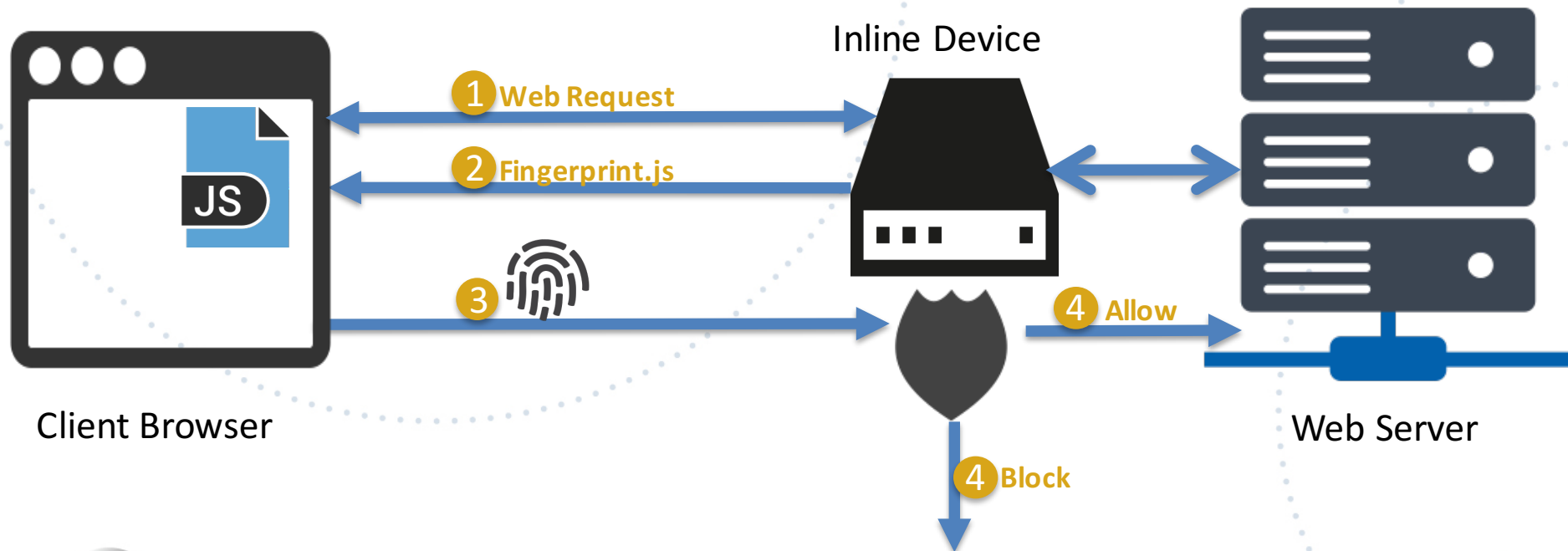
# Stripping Attack



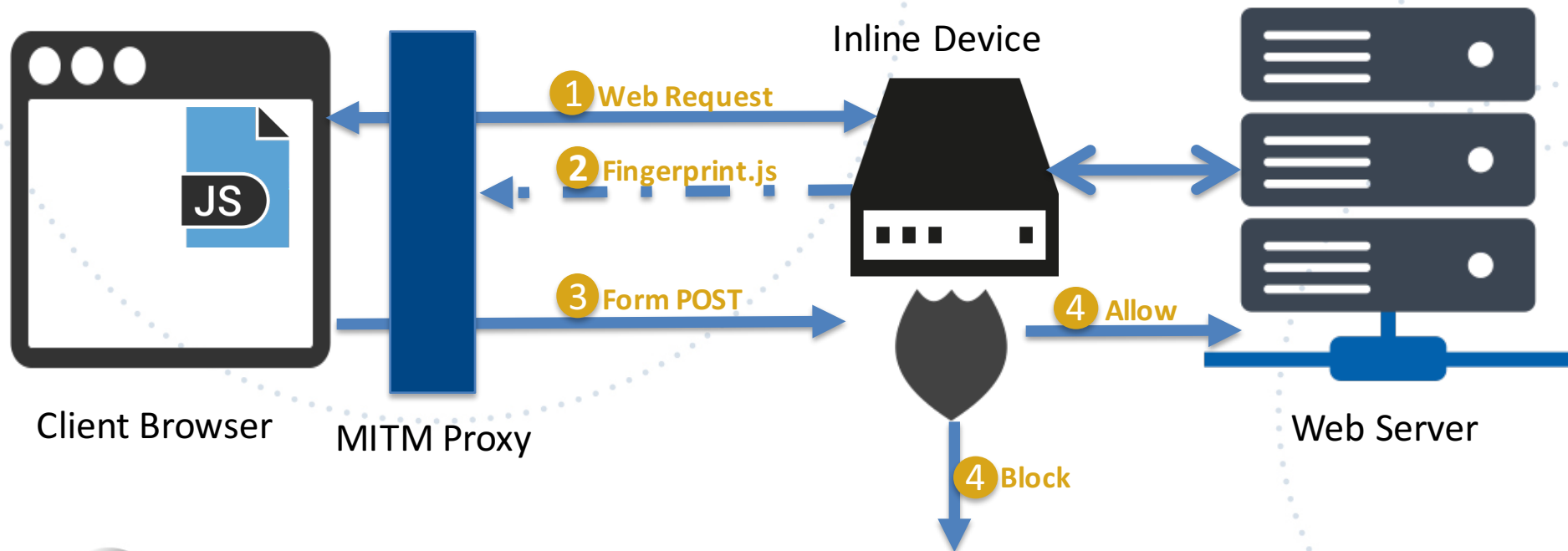
# Stripping Attack



# Stripping Attack



# Stripping Attack



# Replay Attacks

- No check on freshness of payload.

```
while :
do
  curl 'http://example.com:8080/login.php?user=testusername&pass=testpwd\
    &fingerprint=user_agent+%3D+Mozilla%2F5.0+%28Macintosh%3B+Intel+Mac+OS+X+10_12_6%29+AppleWebKit\
    %2F537.36+%28KHTML%2C+like+Gecko%29+Chrome%2F51.0.2%26language+%3D+en-US%26color_depth+%3D+24%26\
    device_memory+%3D+-1%26hardware_concurrency+%3D+826navigator_platform+%3D+MacIntel%26' \
    -H 'Pragma: no-cache' -H 'Accept-Encoding: gzip, deflate, sdch' \
    -H 'Accept-Language: en-US,en;q=0.8' -H 'Upgrade-Insecure-Requests: 1' \
    -H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) \
      Chrome/51.0.2694.0 Safari/537.36' \
    -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8' \
    -H 'Referer: http://example.com:8080/' -H 'Connection: keep-alive'

  sleep 1
done
```



# Dynamic Tokens

- A dynamic token is generated, which is derived from the timestamp.
- Same logic can be replicated in a script.

# Fundamental Issue III

- There are no guarantees of the legitimacy of the data collected by the JavaScript sensors.



# Forging Browser Fingerprints

- FPRANDOM – Modified browser which introduces noise during browser fingerprint.
- OpenWPM – Web Privacy Measurement software.
- Database of Normal Fingerprints

<https://github.com/plaperdr/fprandom>

<https://github.com/citp/OpenWPM>



**OWASP**

Open Web Application  
Security Project

[WWW.OWASP.ORG](http://WWW.OWASP.ORG)

chrome://google.com

About Store Gmail Images Sign in

14

Elements Network Resources Timelines Debugger Storage Console

Filter Console Log

All Errors Warnings Logs

```
> // original source: https://github.com/Valve/fingerprintjs2
function getCanvasFp() {
  var result = ""
  var canvas = document.createElement('canvas')
  var ctx = canvas.getContext('2d')
  ctx.textBaseline = 'alphabetic'
  ctx.fillStyle = '#f60'
  ctx.fillRect(125, 1, 62, 20)
  ctx.fillStyle = '#069'
  ctx.font = '11pt Arial'
  ctx.fillText('Cwm fjordbank glyphs vext quiz, \ud83d\ude03', 2, 15)
  ctx.fillStyle = 'rgba(102, 204, 0, 0.2)'
  ctx.font = '18pt Arial'
  ctx.fillText('Cwm fjordbank glyphs vext quiz, \ud83d\ude03', 4, 45)

  result = 'Canvas Fingerprint is: ' + canvas.toDataURL()
  return result
}
```

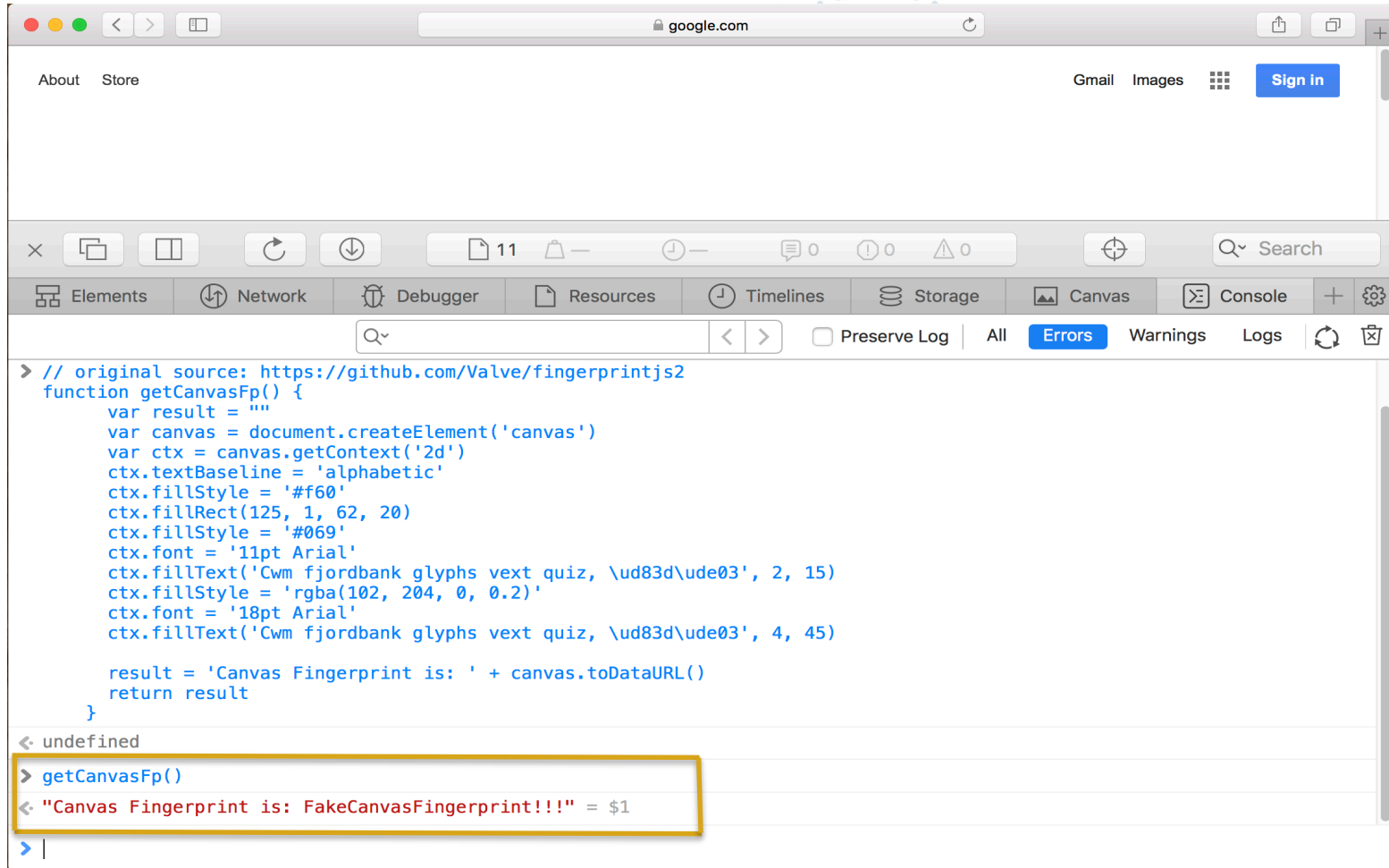
< undefined

> getCanvasFp()

< "Canvas Fingerprint is: data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAASwAAACWCAYAAABkw7XSAAAAAXNSR0IArs4c6QAALexJREFUeAHtnQl8VdW1/3/3ZiQkhAQSEhKBMCoIiDghilGg6k0lti/P0ryKctj/07ZSxKptNfq0Wq21pdo+EBz6cUJ8zyr0qEWccUAmFQyJJAwhIZdA5uT+f+ucu+899+bc5AYZoqz9gZxz9ny+Z++11157n3M9UKcELMBhJ7DmwrNzunfZ/pusPnX/mZDXJx3J3YH6ajSVbtqzY1PC02VVve84aebHW1lR/2Gv7GGsg0cwLq1FHyYC/ulHZqP3zEWna+/PzBjTZVz/db/peVLuz+IGX9UDaacB3h5sGQn83wS0VAL7lqH560d2VywreeSLnSnuPbN4yd7D1HQ0e7Gd7gEediJHQAUVYHW0h/xIcWHyBfkrZ2eePGIajr4WiMuk/uTl/3hWMI7/wBPI/8387QaKJmP3e+9/8THla0vPnvW6/s6x10c2loImWj0g4HnJiK/thLLlDUH3bH+gxMRElJA2tyRKvFh/1JaAU0CIHCLmtuyTy27zQMuJgtugw7169AfU0Ljho2lAKKuoTHj7K1a+H1NiJnQG+gzyRk7PNdeuxby3azAr/g/yOuH1CcR7iB5ybhstlPYvqcFpz1wzoMvrgR027FMfFHYyNiHrrLCb8bbtdnVh2m39aCK+euwOXXv32AK+BhGX4UntLPyvfglBFe5QNdhut30xkvhheyf1dbcA/nT7Px2f4LP+Sp9uIqPIkLDma54e3jW5b0f9efmt27oPInvD0BqF1D4VSGtx5+Gi//d04aG0oosL5GU3MJXvnbPCx+6GmWrvNV7ZdA1iBkD0z56Ut3nNrnW1bh05k8XGAVFsdb0iol6WLUVv5F6eqJ2F15A5AEFJz9LiadPeG



# Forging Browser Fingerprints



```
// original source: https://github.com/Valve/fingerprintjs2
function getCanvasFp() {
  var result = ""
  var canvas = document.createElement('canvas')
  var ctx = canvas.getContext('2d')
  ctx.textBaseline = 'alphabetic'
  ctx.fillStyle = '#f60'
  ctx.fillRect(125, 1, 62, 20)
  ctx.fillStyle = '#069'
  ctx.font = '11pt Arial'
  ctx.fillText('Cwm fjordbank glyphs vext quiz, \ud83d\ude03', 2, 15)
  ctx.fillStyle = 'rgba(102, 204, 0, 0.2)'
  ctx.font = '18pt Arial'
  ctx.fillText('Cwm fjordbank glyphs vext quiz, \ud83d\ude03', 4, 45)

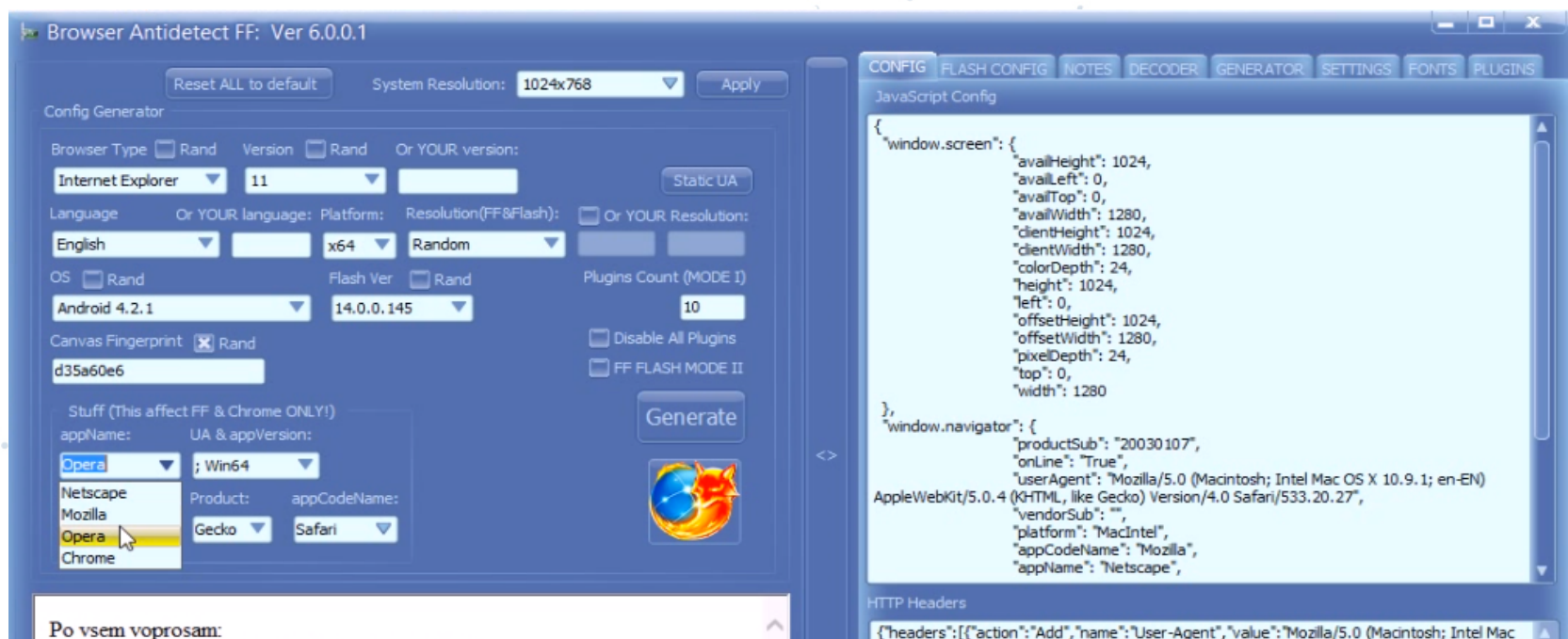
  result = 'Canvas Fingerprint is: ' + canvas.toDataURL()
  return result
}

> undefined
> getCanvasFp()
< "Canvas Fingerprint is: FakeCanvasFingerprint!!!" = $1
> |
```



# Bad Guys Are Already Doing this

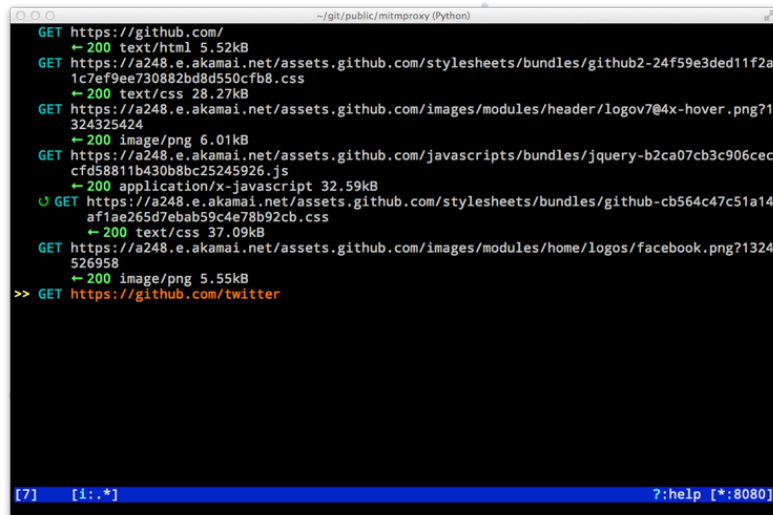
- Anti-Detect\* \$399 in the underground



<https://krebsonsecurity.com/2015/03/antidetect-helps-thieves-hide-digital-fingerprints/>

# User Behavior

- Replay with changed timestamps
- Add ripples and disturbances
- Use MITM Proxy



```
~/git/public/mitmproxy (Python)
GET https://github.com/
  ← 200 text/html 5.52kB
GET https://a248.e.akamai.net/assets.github.com/stylesheets/bundles/github2-24f59e3ded11f2a1c7ef99ee730882bd8d550c-fb8.css
  ← 200 text/css 28.27kB
GET https://a248.e.akamai.net/assets.github.com/images/modules/header/logov7@4x-hover.png?1324325424
  ← 200 image/png 6.01kB
GET https://a248.e.akamai.net/assets.github.com/javascripts/bundles/jquery-b2ca07cb3c906cecf58811b430b8bc25245926.js
  ← 200 application/x-javascript 32.59kB
GET https://a248.e.akamai.net/assets.github.com/stylesheets/bundles/github-cb564c47c51a14af1ae265d7ebab59c4e78b92cb.css
  ← 200 text/css 37.09kB
GET https://a248.e.akamai.net/assets.github.com/images/modules/home/logos/facebook.png?1324526958
  ← 200 image/png 5.55kB
>> GET https://github.com/twitter

[7] [1:.*] ? :help [*:8080]
```



# Fundamental Issue IV

- JavaScript can't protect all flows.



# Fundamental Issue V

- The mitigative action acts as an oracle for the attacker.



**OWASP**

Open Web Application  
Security Project

[WWW.OWASP.ORG](http://WWW.OWASP.ORG)

# Other Issues

- Fraud/Bot Detection Solutions are themselves Fingerprintable.
- Similar issues exist for mobile app SDK based solutions.





# Takeaways

- Implementation and Architectural Issues in multiple deployments.
- JavaScript runs in an attacker controlled environment.
- Understand the limitations of such solutions.
- Protect all flows.

# Questions?



**OWASP**

Open Web Application  
Security Project

[WWW.OWASP.ORG](http://WWW.OWASP.ORG)