Automated Web Attacks Using Headless Browsers

Dima Bekerman & Ben Herzberg

September 2016
ben.about()

> ben.history
< · ["PT","Dev"]
> ben.employer
< · "Imperva Incapsula"
> ben.positionX
< · "Sec. Research Manager"
> ben.social
< · {"TWT": "@KernelXSS", "LNK": "Ben Herzberg"}
dima.about()

> dima.history
<: ["Machine Learning", "Sec. Analysis"]

> dima.employer
<: "Imperva Incapsula"

> dima.position
<: "Security Researcher"

> dima.social
<: {"TWT": "@_unxmaster", "LNK": "Dima Bekerman"}
Remember…

What happens in Novi Sad stays in Novi Sad
Headless browser

A headless browser is a web browser without a graphical user interface. It accesses web pages but doesn’t show them to any human being, it used to provide the content of web pages to other programs.

https://en.wikipedia.org/wiki/Headless_browser
No JavaScript
No Dom
No Styling
No Layouts

HTTP library

Command line tool

Web Browser

Headless Browser

@@Incapsula_com  @KernelXSS  @_unxmaster
Web browser engine

A web browser engine is a program that renders marked up content and formatting information.

- **Blink**
  - Chrome 28+
  - Opera 15+
  - Android 4.4+

- **WebKit**
  - Safari
  - iOS
  - Chrome <27

- **Gecko**
  - Firefox
  - IceWeasel
  - SeaMonkey

- **Trident**
  - Internet Explorer
  - Windows Mobile
  - EdgeHTML

- **Presto**
  - Opera
  - Opera Mini
  - Opera Mobile
What is DOM – Document Object Model?

Is the HTML you write DOM?

Nope

The HTML you write is parsed by the browser and turned into the DOM.

Is the ‘View-Source’ DOM?

Nope

View Source just shows you the HTML that makes up that page. It's probably the exact HTML that you wrote.

Is the code in DevTool DOM?

Yep

DevTools panel shows you a visual representation of the DOM.
Source Code

```html
<div id="container"></div>

<script>
    var container = document.getElementById("container");
    container.innerHTML = "Hello World";
</script>
```

DOM representation

```html
<div id="container">Hello World</div>
```
Page processing

- JavaScript
- CSS
- SVG
- MathML
- WebGL
Modernizer web features test
ECMAScript test262

Failures in test262

- PhantomJS 2: 2
- Safari 7: 2
- Chrome 36: 4
- Firefox 31: 42
- Internet Explorer 11: 1
PhantomJS

- Based on WebKit
- Cross Platform
- Portable
- Scriptable with JavaScript
- Good for WebUI automation

SlimerJS

- Based on Gecko

TrifleJS

- Based on Trident

Awesomium

- Chromium based
- V8 – JavaScript Engine
- Programmable with .NET, C++
- Compiled to cross platform app
- Can be used as a module

CEF

Chromium Embedded Framework

WebKit .NET
Selenium / WebDriver

- Selenium is a browser control and automation system
- Manipulating a real web browser using a browser driver
- Manipulated browser must be installed
- Host OS should have a GUI framework
The world of two contexts

Browser Context

- Language depends on framework
- Compiled / evaluated inside framework interpreter
- Has access to OS

Page Context

- JavaScript
- Evaluated by the web engine
- Has access to DOM objects
The world of two contexts

Browser Context

my_script.js

Page Context

title

example.com

page.open("example.com");

page.evaluate(function(){
    return document.title;
});

WebKit

function () {
    return document.title;
}

Control
Page event
Inject
Callback
Use Cases

• Test automation
• Crawl websites
• Render dynamic web page to static HTML
• Inject and execute arbitrary JavaScript code
• Perform actions
• Taking screen shots
• Monitor performance
Use Cases

- Content scraping
- Click fraud
- Fuzzing
- Bidding abuse
- Login brute force
- Botnet
- Sophisticated DDoS attacks
Capture screen to image

Source Code

```javascript
var page = require('webpage').create();

page.open('http://www.google.com', function (status) {
    page.render('google.png');
    phantom.exit();
});
```
Grabbing text

Source Code

```javascript
// Page title
console.log(page.title);

// View-source
console.log(page.content);

// Content without HTML tags
console.log(page.plainText);
```
Handle console and alerts

Source Code

```javascript
//Handle console
page.onConsoleMessage = function(msg) {
    console.log('CONSOLE: ' + msg);
};

//Handle alert
page.onAlert = function(msg) {
    console.log('ALERT: ' + msg);
};
```
Spot the differences
Awesomium
webView.setWebViewClient(new WebViewClient() {
    @Override
    public void onPageFinished(WebView view, String url) {
        browser.loadUrl("javascript:window.HTMLOUT.processHTML(
                document.getElementsByTagName('html'));");
    }
});
Scenarios from the wild

- SEO tools scrape for dynamic content
- E-commerce abuse
- DDoS attacks from a mobile botnet
Хвала вам!

THANK YOU!

@KernelXSS
Ben Herzberg
hben@incapsula.com

@_unxmaster
Dima Bekerman
dima@incapsula.com