Front-end Security

Martin Stoynov

Spas Genov

martin@bsides:~$ whoami

- DevSecOps/AppSec Engineer at DXC Technology
- Former Web Developer
- MSc in Information Security
- Member of CSF (Cyber Security Foundation)
- Web Security Challenges and CTFs

spas@bsides:~$ whoami

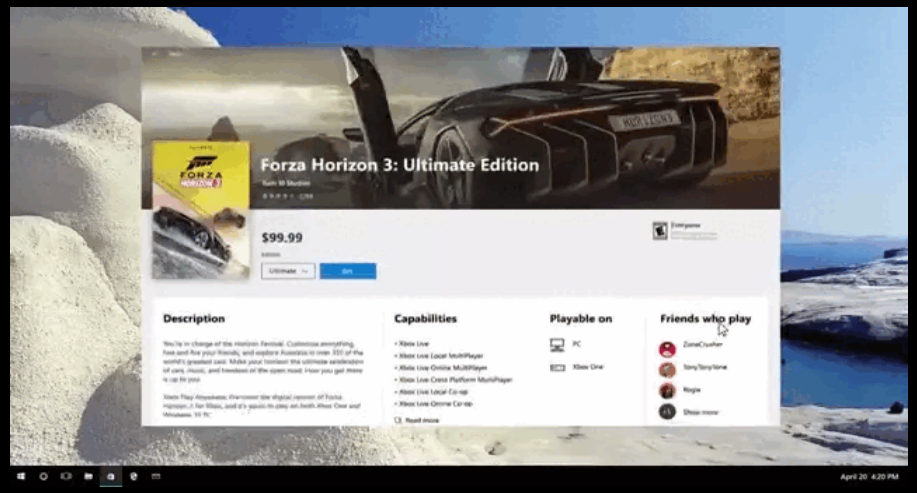
- Engineer (MSc in Civil Engineering), hiker, biker, snowboarder…
- Currently automation QA at NewsUK (The Sun, The Times, The Sunday Times)
- QA should include Security Testing (personal opinion)

Contents

- Evolution of client-side applications
- Common security misconceptions
- Overview of common vulnerabilities
  - What is it?
  - A quick demo.
  - What is the impact?
  - How to prevent?
  - How to test?
  - + more.

Evolution of client-side applications

- Static page stage
  - Back-end MVC model
- AJAX stage
  - 2004 - Gmail, GMaps
- Front-end MVC stage
  - 2010 - Backbone.js (first front-end MVC)
- SPA stage
  - Now - React, Angular, Vue



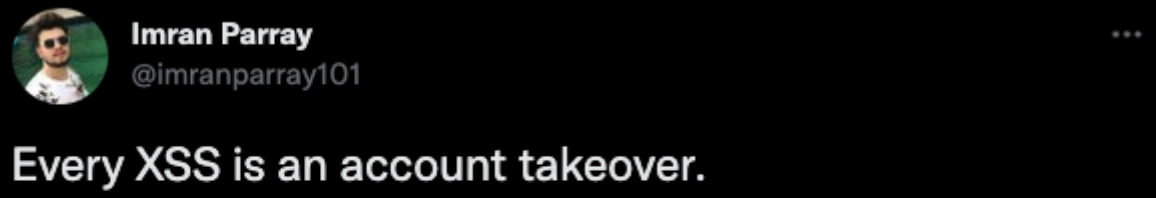
Common security misconceptions

- Myth: The app is secure when we validate input on the client-side.

- Fact: Client-side input validation does nothing. Output sanitization on the other hand will.

Common security misconceptions

- Myth: Client-side vulnerabilities do not pose a threat.
- Fact: Client-side vulnerabilities may sometimes have disastrous effects.



Common security misconceptions

- Myth: We have modern frameworks and modern browsers, so client-side vulnerabilities are dead…

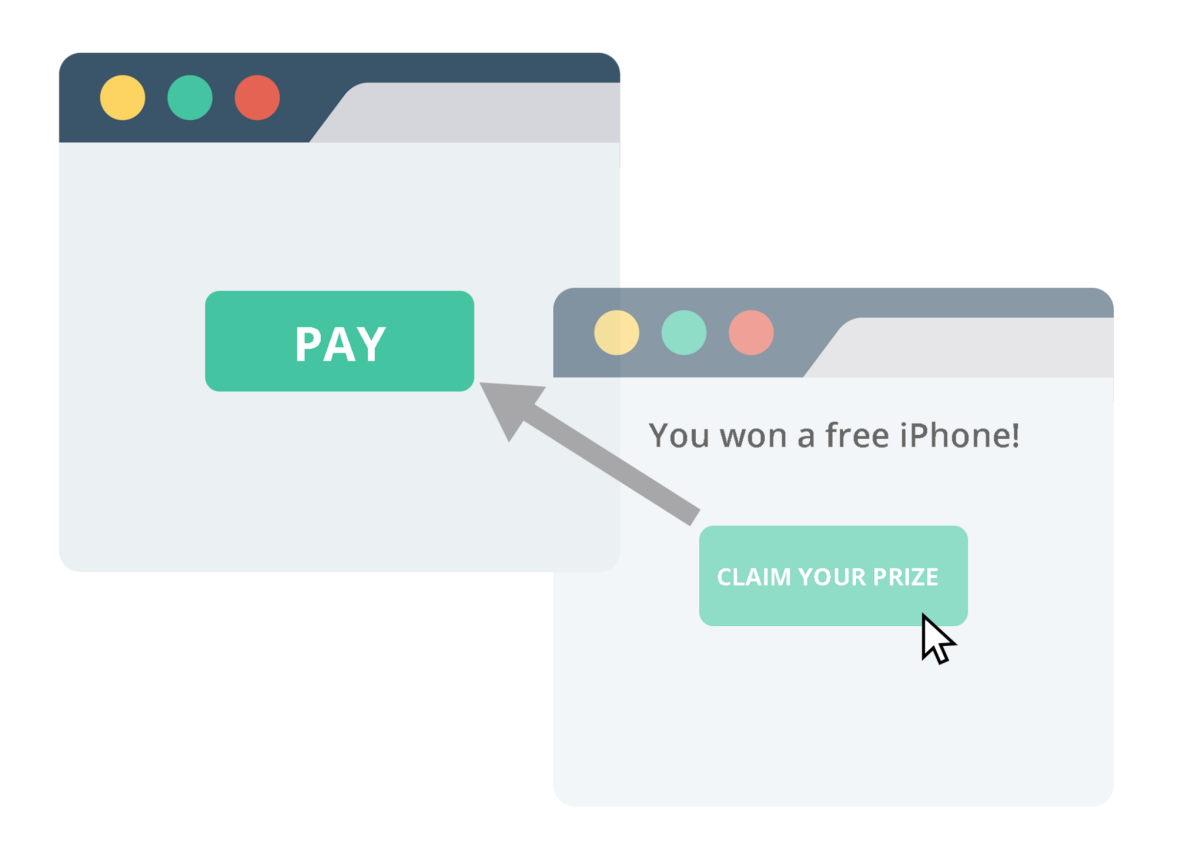
- Fact: They are only dead if the target 1) knows about security measures and 2) implements them correctly

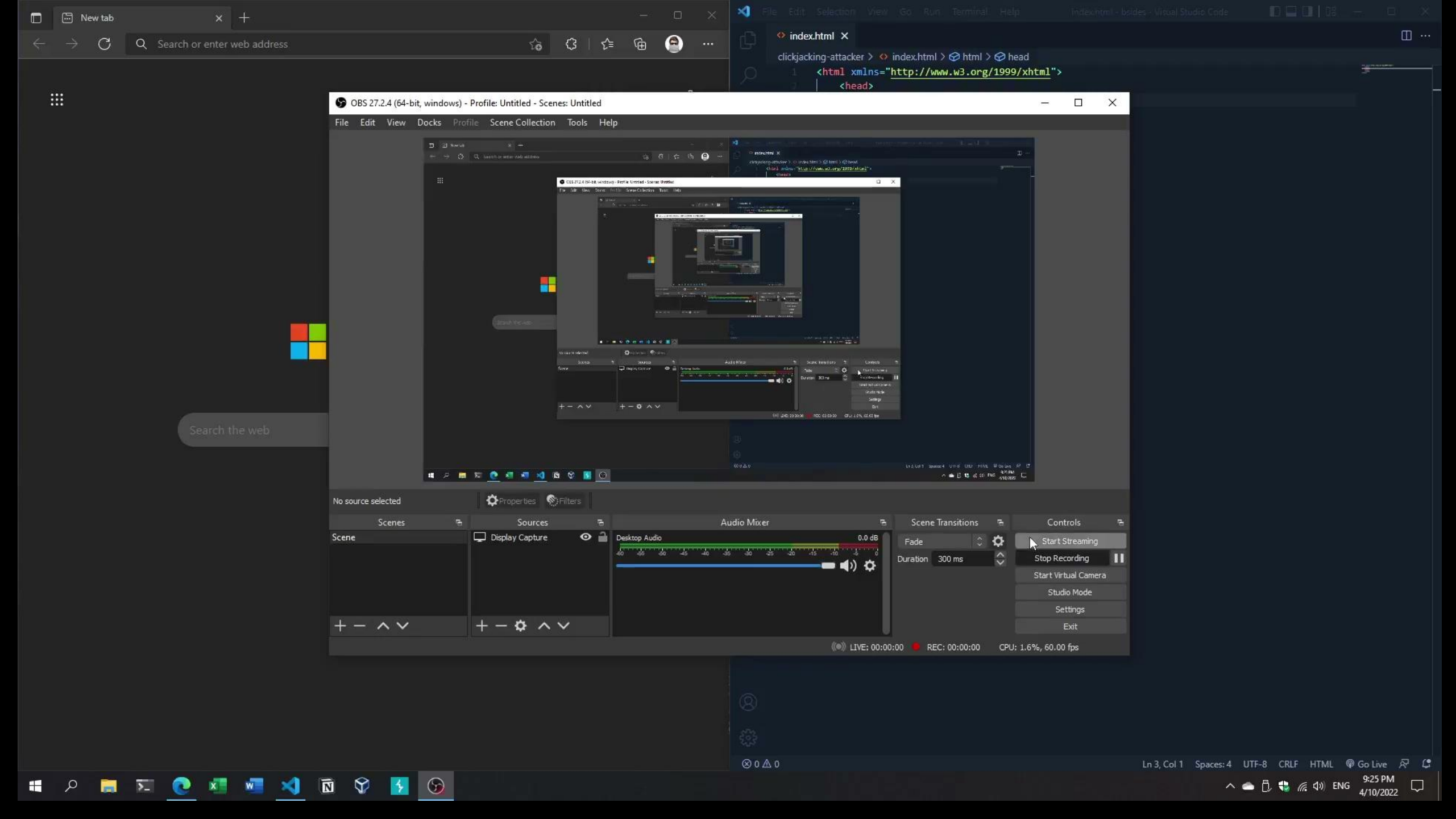
Overview of common vulnerabilities

- Clickjacking
- Cross-site Scripting (XSS)
- Cross-site Leaks (XSL)

Clickjacking

- What is clickjacking?
  - Tricks a user into clicking something.
- How does it work?
  - Frames a vulnerable web pages and overlays fake CSS over it.




New tab
Search or enter web address
index.html
clickjacking-attacker > index.html > html > head
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
OBS 27.2.4 (64-bit, windows) - Profile: Untitled - Scenes: Untitled
File Edit View Docks Profile Scene Collection Tools Help
No source selected
Properties
Filters
Scenes
Scene
Sources
Display Capture
Audio Mixer
Desktop Audio
0.0 dB
Scene Transitions
Fade
Duration 300 ms
Controls
Start Streaming
Stop Recording
Start Virtual Camera
Studio Mode
Settings
Exit
LIVE: 00:00:00
REC: 00:00:00
CPU: 1.6%, 60.00 fps
Search the web
Ln 3, Col 1
Spaces: 4
UTF-8
CRLF
HTML
Go Live
ENG
9:25 PM
4/10/2022

Impact of Clickjacking

- Forces the user to make dangerous requests
  - Payments
  - Account deletion
  - etc.

Prevent Clickjacking Attacks

- Client-side defenses (for example Framekiller JS)
- X-Frame-Options header
- Content Security Policy (CSP)

```
<script type="text/javascript">
  if (top != self) top.location.replace(location);
</script>
```

X-Frame-Options: DENY

CSP Header (Frame-Ancestors) frame-ancestors 'self' https://REDACTED.com

How to test?

- Manual testing - **clickjacker.io**, **online.attacker-site.com**
- Burp Suite -> Burp Clickbandit (PoC with Record mode & Review mode)
- OWASP ZAP (cmd, daemon/headless, jenkins integration, github actions)

web.clickjacker.io

web.clickjacker.io/test?url=RECATED.bg

Clickjacker

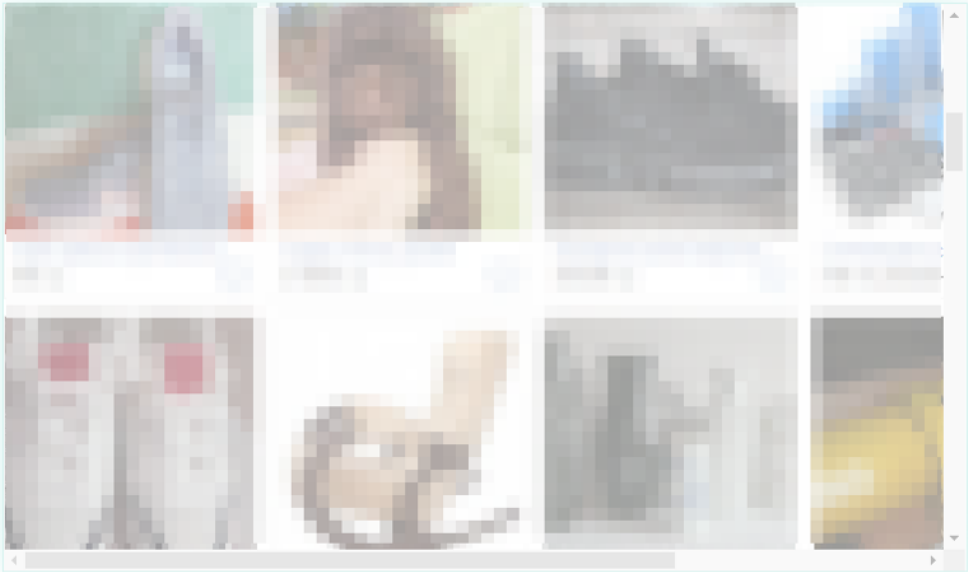
Test both Internet and Intranet sites

HOME ABOUT

Test for Clickjacking now

http:// REDACTED.bg TEST

Share result via url: http://web.clickjacker.io/test?url=http://RECATED.bg COPY



Test Results:

Site:	https://REDACTED.bg/ **(Redirection followed)**
IP Address:	
Time:	Thu Apr 14 2022 20:27:03 GMT+0000 (Coordinated Universal Time)
X-Frame-Options:	Missing header
CSP Header (Frame-Ancestors)	Missing header

Toggle this to show/hide object on Iframe to Capture PoC

It is vulnerable to clickjacking attack.

Burp Suite Community Edition | Burp | Project | Intruder | Repeater | Window | Help | Logger++ | Turbo Intruder | 100% | Fri 15 Apr 01:04

Search [Pro version only]
Configuration library
User options
Burp Infiltrator [Pro version only]
Burp Clickbandit
Burp Collaborator client [Pro version only]
Exit

ite Community Edition v2022.2.4 - Temporary Project

Dashboard | Target | Pr | der | Comparer | Logger | Extender | Project options | User options | Learn | JSON Web Tokens

Tasks

Filter | Running | Paused

New live task

Time to level up? Catch more bugs with Burp Suite Pro | Find out more

Search... | Search...

1. Live passive crawl from Proxy

Add links. Add item itself, same d | dded to site map

Capturing:

ses processed

0 responses queued

vity [Pro version only]

Filter | High | Medium | Lo | Medium | Certain | Firm | Tentative | Search...

Issue type	Host	Path
Suspicious input transformation (reflected)	http://insecure-bank.com	/url-shorten
SMTP header injection	http://insecure-website.c...	/contact-us
Serialized object in HTTP message	http://insecure-bank.com	/blog
Cross-site scripting (DOM-based)	https://insecure-bank.com	/
XML external entity injection	https://vulnerable-websit...	/product/stock
External service interaction (HTTP)	https://insecure-website....	/product
Web cache poisoning	http://insecure-bank.com	/contact-us
Server-side template injection	http://insecure-bank.com	/user-homepage
SQL injection	https://vulnerable-websit...	/
OS command injection	https://insecure-website....	/feedback/submit

Advisory

Event log

Filter | Critical | Error | Info | Debug | Search...

Time	Type	Source	Message
23:51:20 14 Apr 2022	Info	Proxy	Proxy service started on 127.0.0.1:8080

Memory: 104.4MB | Disk: 74.1MB | 00 % | 01:04

Program Statistics

Clickjacking most commonly disclosed on:

- **Legalrobot** - 8 reports - Clickjacking: X-Frame-Options header missing (#163646)
- **Yelp** - 8 reports - CLICKJACKING at Yelp Reservations Resulting in exposure of victim Email info + Victim Credit Card (#163646)
- **Yahoo** - 7 reports - Bypass of the Clickjacking protection on Flickr using data URL in iframes - **$250** (#7264)
- **Mailru** - 7 reports - Clickjacking on o2.mail.ru - **$150** (#8724)
- **Twitter** - 7 reports - Viral Direct Message Clickjacking via link truncation lead to capture Google credentials **$1,120** (#643274)
- **Wordpress** - 7 reports - Account takeover vulnerability by editor role privileged users/attackers via clickjacking (#388254)
- **Automattic** - 4 reports - Exploiting clickjacking vulnerability to trigger self DOM-based XSS **$150** (#953579)
- **Coinbase** - 3 reports OAuth authorization page vulnerable to clickjacking $5,000 (#65825)
- **Gratipay** - 3 reports Bypassing X-frame options (#283951)
- **Gitlab** - 3 reports Gitlab.com is vulnerable to reverse tabnabbing via AsciiDoc links (#213114)

HackerOne Inc. [US] hackerone.com/reports/643274

Login | Contacted by a hacker? | Contact Us

hackerone

SOLUTIONS | PRODUCTS | PARTNERS | COMPANY | HACKERS | RESOURCES

64 #643274

Viral Direct Message Clickjacking via link truncation leading to capture of both Google credentials & installation of malicious 3rd party Twitter App

Share:

TIMELINE

slickrockweb submitted a report to Twitter. Jul 15th (3 years ago)

Summary: [Viral Direct Message Clickjacking via link truncation leading to capture of both Google credentials & installation of malicious 3rd party Twitter App]

Description: [Because very long links in direct messages are truncated after 38 characters the malicious actors were able to provide a malicious link in a direct message that appeared as though it was to an authenticated YouTube video and caused a clickjacking scenario to occur. The link caused any users that were already logged into a Google account to be first logged out and then asked to log back in. A malicious Google app captured the account credentials and then redirected the user to the website getmorefollowers.biz (embedded in the initial link query string) which in turn redirected the user to freefollowers.eu domain. This executed a PHP script and /or a javascript which in turn redirected the user to one of at least 10 different randomized malicious 3rd party Twitter apps (see attached file redirect-sequence-from-start.png for the initial redirect sequence). Depending on whether the user was already logged into their Twitter account, the authentication process was potentially done for the user and/or the user only needed to click on the authenticate button. These apps all did essentially the same thing. They generated a couple of followers to the account but also hijacked the account into sending this same malicious link as a Direct message to everyone that it was able to in that account (open DMs and reciprocal follows). Thus creating the virality of the infection and starting the sequence all over again on hundreds of new victims.

Users that weren't already logged into Google were redirected directly to getmorefollowers.biz and then to freefollowers.eu and then to the malicious Twitter app and sent to one of at least 10 different randomized Oauth screens and encouraged to connect to a 3rd party Twitter app that would supposedly provide you free followers (also provided a paid service to increase your followers).

Here is an example "FULL" link we received from a malicious account that we were investigating named @[redacted]

ONLY FOR YOU Eric JN Ellason {{ https://accounts.youtube.com/accounts/SetSID?89085489=[redacted]&ilo=1&89085489=[redacted]&ils=a4cc1b7ed445598f16cef403bb3b0311&ilc=0&Bi06UejC9N=89085489&continue=https%3A%2F%2Fgoogle.com%2Faccounts%2FLogout%3Fcontinue%3Dhttps%253A%252F%252Fappengine.google.com%252F_ah%252Flogout%253Fcontinue%253Dhttps%25253A%25252F%25252Fwww.google.com%25252Furl%25253Fsa%25253Dt%252526rct%25253Dj%252526q%25253D%252526esrc%25253Ds%252526frm%25253D1%252526source%25253Dweb%252526cd%25253D1%252526cad%25253Drja%252526ved%25253D0CDAQFjAA%252526url%25253Dhttp%2525253A%2525252F%2525252Fwww.getmorefollowers.biz%2525252F%252526ei%25253D3meWUs3fGMun0wWr94CoAg%252526usg%25253DAFQjCNFg9bZvpiCSGCVgdaryfriEHS-XEA%252526sig2%25253D8hAat-jqQCQ0Ciz9ywCbEw%252526bvm%25253Dbv.57155469%25252Cd.bGQ&Bi06UejC9N=43992 ... }} message id: 92439

What gets displaying and hotlinked in the Direct Message is this:

Reported July 15, 2019 11:48am +0300

slickrockweb

Participants

State ● Resolved ()

Reported to Twitter

Disclosed October 31, 2019 7:12pm +0200

Severity High (7 ~ 8.9)

Weakness *None*

Bounty $1,120

CVE ID *None*

Account de... *None*

Cross-site Scripting (XSS)

- What is cross-site scripting?
  - Javascript code injection
- How does it happen?
  - Dangerous functions
  - DOM reflection

Historic Overview

0A.X. = 1999 A.D.

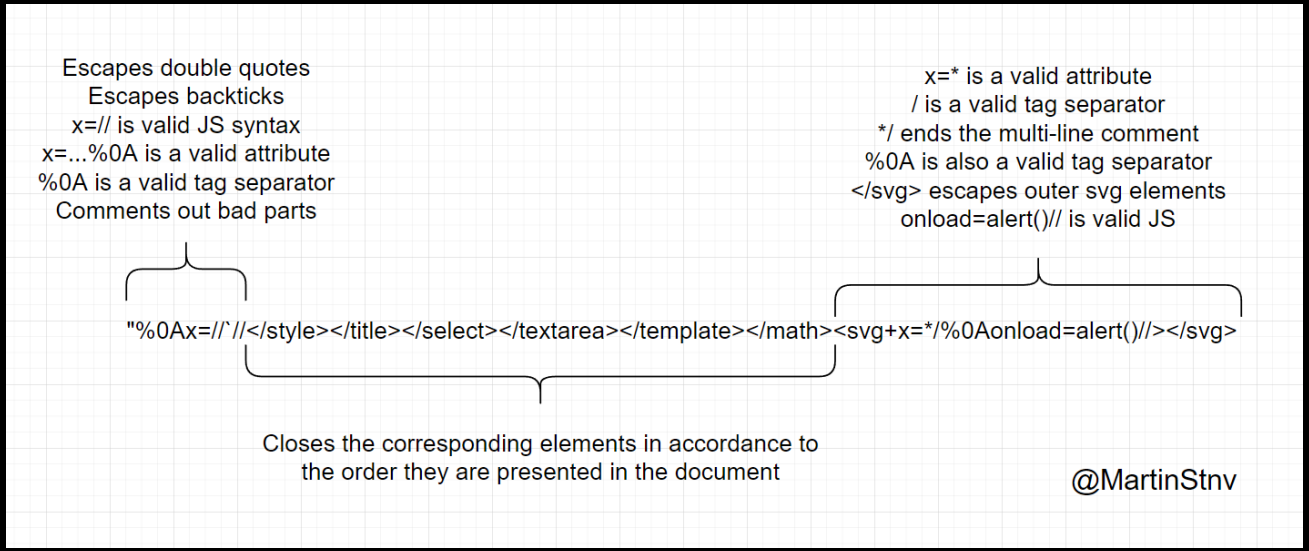
23 years later, the XSS is still here

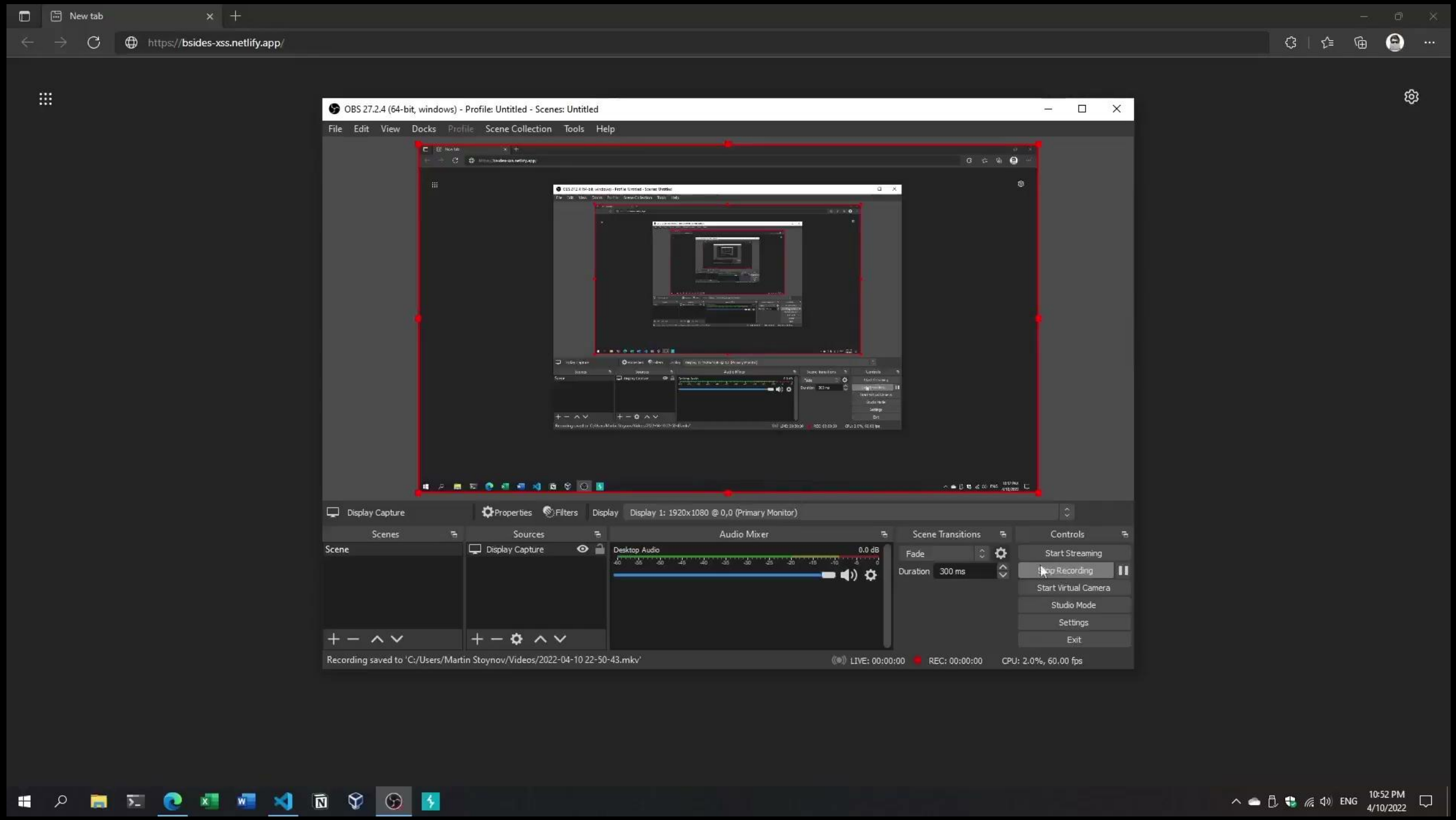
Microsoft coined the term XSS after having a lot of alternatives — some particular, some quite absurd, and one fitting in place.

- Unauthorized Site Scripting
- Unofficial Site Scripting
- URL Parameter Script Insertion
- Cross-Site Scripting (CSS != XSS)
- Synthesized Scripting
- Fraudulent Scripting

Types of XSS

- Persistence-based
  - Stored XSS (Persistent)
  - Reflected XSS (Non-Persistent)
- Context-based
  - DOM XSS
  - mXSS (Mutation XSS)
  - Polyglot




New tab
https://bsides-xss.netlify.app/
OBS 27.2.4 (64-bit, windows) - Profile: Untitled - Scenes: Untitled
File Edit View Docks Profile Scene Collection Tools Help
Display Capture
Properties
Filters
Display
Display 1: 1920x1080 @ 0,0 (Primary Monitor)
Scenes
Scene
Sources
Display Capture
Audio Mixer
Desktop Audio
0.0 dB
Scene Transitions
Fade
Duration 300 ms
Controls
Start Streaming
Stop Recording
Start Virtual Camera
Studio Mode
Settings
Exit
Recording saved to 'C:/Users/Martin Stoynov/Videos/2022-04-10 22-50-43.mkv'
LIVE: 00:00:00
REC: 00:00:00
CPU: 2.0%, 60.00 fps
ENG
10:52 PM
4/10/2022

Impact of XSS

- PII leakage
- Request forgery
- Credential theft
- Account takeover

Basically XSS is the same as somebody who has logged in your account without permission.

Prevent XSS attacks

- Avoid using known dangerous function
- ALWAYS filter user input & UI output
- Use built-in security policies & mechanisms
  - Content Security Policy (CSP)

How to test for XSS?

- Dalfox - https://github.com/hahwul/dalfox (1,7k)
- XSStrike - https://github.com/s0md3v/XSStrike (10,2k)
- XSSHunter - https://xsshunter.com (blind XSS)

```
> python3 xsstrike.py -u https://brutelogic.com.br/multi/js-object3.php?p=d3v

        XSStrike v3.0.5

[~] Checking for DOM vulnerabilities
[+] WAF Status: Offline
[!] Testing parameter: p
[!] Reflections found: 2
[~] Analysing reflections
[~] Generating payloads
[!] Payloads generated: 20
------------------------------------------------------------
[+] Payload: }}}/confirm()//\
[!] Efficiency: 100
[!] Confidence: 7
[?] Would you like to continue scanning? [y/N] |
```

pictures

Not Secure http://testphp.vulnweb.com/listproducts.php?artist=123&asdf=ff&cat=123

Search DuckDuckGo

Private

acunetix acuart

TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art

go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo

Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer

About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

Program Statistics - Generic: 881; Stored: 359; Reflected: 277, DOM: 106;

XSS most commonly disclosed on:

- **Mail.ru** - 71 reports - Home page reflected XSS - **$250** (#9318)
- **Shopify** - 50 reports - XSS in experts.shopify.com - **$500** (#57459)
- **DoD** - 42 reports - XSS vulnerability on an Army website (#187881)

- **Mail.ru** - 63 reports - Stored XSS in e.mail.ru (payload affect multiple users) - **$750** (#217007)
- **GitLab** - 24 reports - Stored XSS on Files overview by abusing git submodule URL - (#218872)
- **NewRelic** - 23 reports - Stored XSS in Brower `name` field reflected in two pages - **$3000** (#348076)

- **Mail.ru** - 54 reports - Reflected XSS on frag.mail.ru (#214642)
- **DoD** - 32 reports - Reflected XSS in a DoD Website (#217108)

- **Mail.ru** - 17 reports - XSS in biz.mail.ru/error **$500** (#268245)
- **Rockstar Games** - 10 reports - dom based xss in http://www.rockstargames.com/GTAOnline/ **$500** (#261571)

A personal research on XSS over the course of 2021 has shown that over 10% of bulgarian websites are vulnerable.

I have personally tested over 1000 bulgarian websites in which I have found 100+ valid XSS vulnerabilities.

These vulnerabilities have all been responsibly disclosed.

Cross-site Leaks (XSL)

- Takes advantage of small pieces of information.
- This information is exposed from cross-site communication.
- The pieces of information usually have a binary form and are referred to as “oracles”.
- Oracles generally answer with YES or NO to cleverly prepared questions in a way that is visible to an attacker.

Example

Oracles generally answer with YES or NO to cleverly prepared questions in a way that is visible to an attacker.

- Does the word secret appear in the user's search results in another web application?
- Does the query ?query=secret return an HTTP 200 status code?
- Does loading a resource from ?query=secret in the application trigger the onload event?

The above query could be repeated by an attacker for many different keywords, and as a result the answers could be used to infer sensitive information about the user's data.

Browsers provide a wide range of different APIs that, while well-intended, can end up leaking small amounts of cross-origin information.

XSL Resources

- xsleaks.dev - wiki
- xsinator.com - browser test suite
- “Massive XS-Search over multiple Google products” - terjanq.medium.com


That's all Folks!