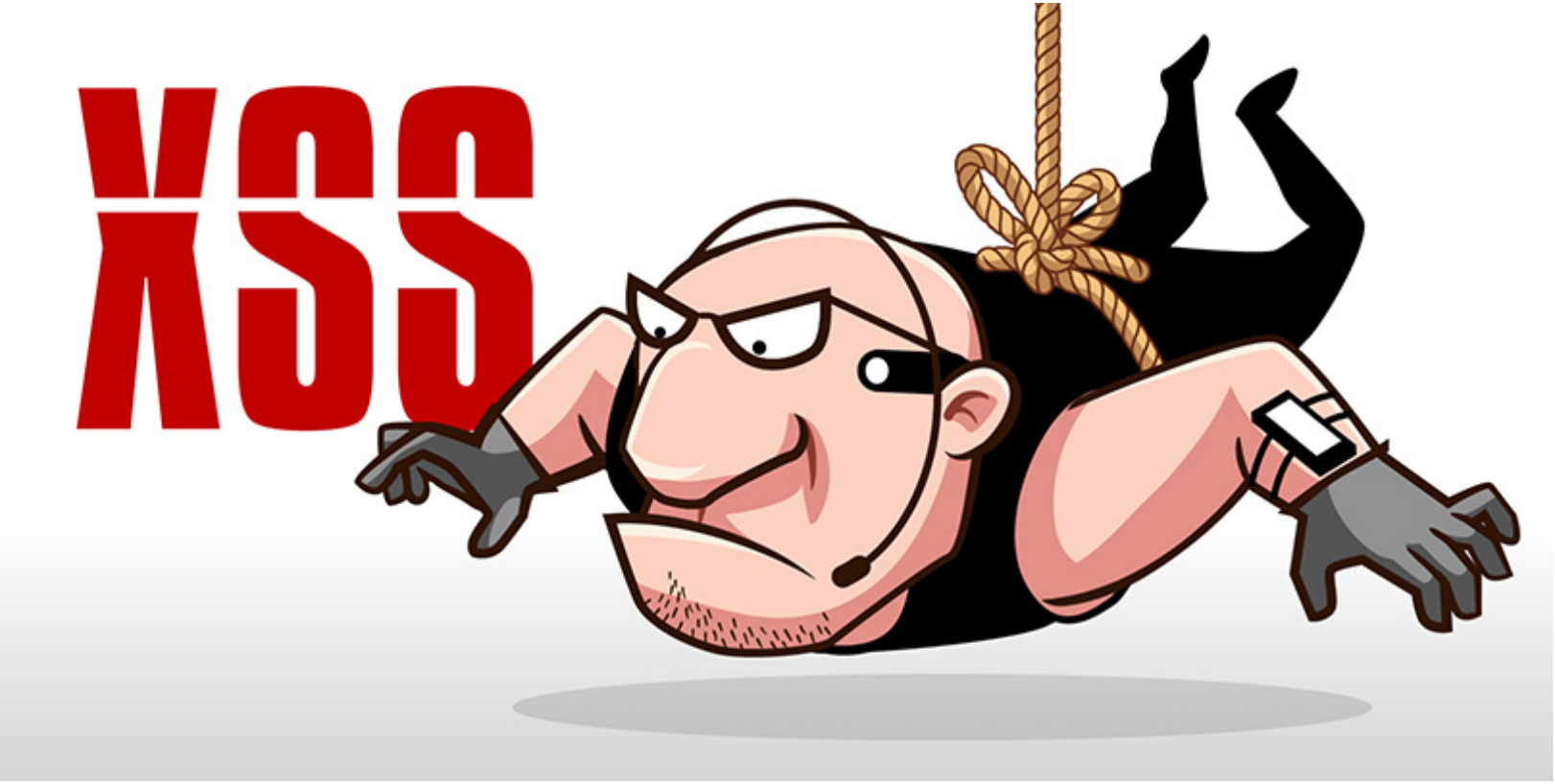
Documenting the impossible: Unexploitable XSS labs

Gareth Heyes | 22 May 2020 at 13:08 UTC
Updated: 26 May 2020 at 09:30 UTC



Have you ever found some risky behavior, but couldn't quite prove it was exploitable? Our XSS cheat sheet contains virtually every exploit technique we know of, but what should you do if you can't find a technique for your scenario? Did we just forget to mention the right technique, or is it actually unexploitable?

Here at PortSwigger we think anything is hackable, so when we come across a problem that isn't solvable we don't like it. We frequently invest a significant amount of time in trying to crack these challenges, and we know many of you do the same. That's why we decided to create an 'impossible labs' section on our XSS cheat sheet, offering two key benefits:

1) If you're up for a challenge, you can try and solve these labs and win eternal fame

2) If you find a scenario in the wild that matches one of our impossible labs and can't exploit it, you can have reasonable confidence that nobody else knows how to exploit it either.

This section will grow in time but for the time being we have 8 labs. Each lab defines a small problem that our research team has tried to solve but have been unable to pop an alert, limiting us to scriptless attacks like dangling markup injection which are often less powerful. Can you do the impossible and solve these labs?

Rules:

- Vectors must work in modern browsers only - no Internet Explorer! Or Old Edge
- The alert function must be called

Basic HTML context but WAF blocks <[a-zA-Z]

This lab captures the scenario when you can't use an open tag followed by an alphanumeric character. Sometimes you can solve this problem by bypassing the WAF entirely, but what about when that's not an option? Certain versions of .NET have this behaviour, and it's only known to be exploitable in old IE with <%tag.

View lab

Script based injection but quotes, forward slash and backslash are escaped

We often encounter this situation in the wild: you have an injection inside a JavaScript variable and can inject angle brackets, but quotes and forward/backslashes are escaped so you can't simply close the script block.

The closest we've got to solving this is when you have multiple injection points, the first within a script based context and the second in HTML.

View lab

innerHTML context but no equals allowed

You have a site that processes the query string and URL decodes the parameters but splits on the equals then assigns to innerHTML. In this context <script> doesn't work and we can't use = to create an event.

View lab

Basic context length limit

This lab's injection occurs within the basic HTML context but has a length limitation of 15. We came up with a vector that could execute JavaScript in 16 characters: <q oncut=alert`` but can you beat it?

View lab

Attribute context length limit

The context of this lab inside an attribute with a length limitation of 14 characters. We came up with a vector that executes JavaScript in 15 characters:"oncut=alert``+ the plus is a trailing space. Do you think you can beat it?

View lab

Injection occurs inside a frameset but before the body

We received a request from twitter about this next lab. It occurs within a frameset but before a body tag with equals filtered. You would think you could inject a closing frameset followed by a script block but that would be too easy.

View lab

Basic context length limit arbitrary code

It's all well and good executing JavaScript but if all you can do is call alert what use is that? In this lab we demonstrate the shortest possible way to execute arbitrary code.

View lab

Attribute context length limit arbitrary code

Again calling alert proves you can call a function but we created another lab to find the shortest possible attribute based injection with arbitrary JavaScript.

William Bowling @wcbowling solved this lab. The limit is now 20.

View lab

Think you have solved an impossible lab? You are my new hero. Be ready to accept fame and glory, please report your epic accomplishment on twitter or file a new issue on Github.

To view the impossible labs please consult the XSS cheat sheet.

Gareth Heyes

@garethheyes

Latest Posts

JavaScript without parentheses using DOMMatrix

Top 10 web hacking techniques of 2019

DOM Clobbering strikes back

Related stories

DDLS and Open Colleges partner to expand online cyber training in Australia

25 May 2020

Google Cloud security find earns South American researcher $31k bug bounty payout

22 May 2020

How advances in machine learning can improve DDoS attack detection

22 May 2020