

Day 6 - Be careful with what you wish on a Christmas night

Scenario

Task 11

[Day 6]

Web Exploitation

Be careful with what you wish on a Christmas night

Watch DarkStar's Video On Solving This Task.

Deploy

This year, Santa wanted to go fully digital and invented a "Make a wish!" system. It's an extremely simple web app that would allow people to anonymously share their wishes with others. Unfortunately, right after the hacker attack, the security team has discovered that someone has compromised the "Make a wish!". Most of the wishes have disappeared and the website is now redirecting to a malicious website. An attacker might have pretended to submit a wish and put a malicious request on the server! The security team has pulled a back-up server for you on `MACHINE_IP:5000`. Your goal is to find the way the attacker could have exploited the application.

By [Swafox](#)

the root webpage

Santa's portal - Mozilla Firefox

Santa's portal

10.10.121.216:5000

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security

Welcome to Santa's official 'Ma

YEAR 2020

Here you can anonymously submit your Christmas wishe

Search query

Showing all wishes:

enter test into search query

Here you can:

test

& it shows 2 thing here:

1st there's a GET param 'q' with our 'test' string as the query

0/?q=test

2nd is that the 'test' string will be shown up on the webpage

Here are all wishes that have "test":

let's try injecting javascript to spawn alert box, see whether it's vulnerable to XSS or not

<svg onload=alert('xss') />|

omg the website is vulnerable to reflected XSS vulnerability

21.216:5000/?q=<svg+onload%3Dalert('xss')+%2F>

[Kali Docs](#) [Kali Forums](#) [NetHunter](#) [Offensive Security](#) [Exploit-DB](#)

Welcome to Santa's official 'Make a Wish

YEAR 2020

anonymously submit

see w

XSS

OK

let's try another one here the wish box to see what will happen here

Enter your wish here:

New book...

enter test string into it

Enter your wish here:

test

& it shows that the 'test' string will be stored in the server webpage



test

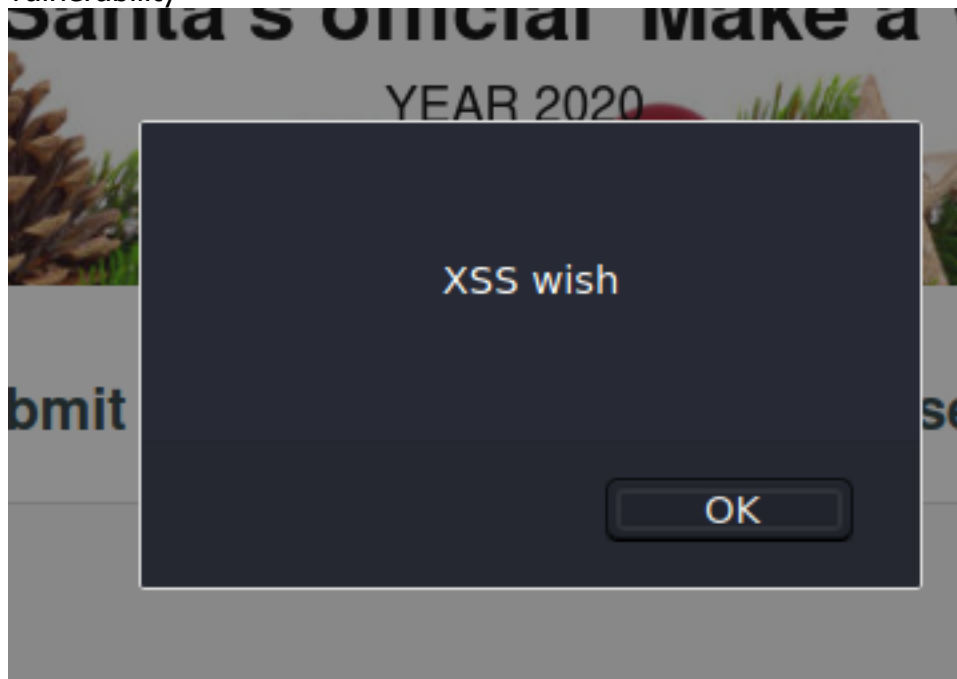
now let's test again to check for XSS vulnerability

Enter your wish here:



```
<svg onload='alert("XSS wish")'|/>
```

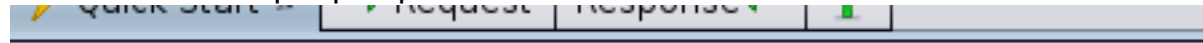
& it works!! everytime we refresh the webpage it'll popup this vulnerability. This is called stored XSS vulnerability



Question: What vulnerability type was used to exploit the application?
-stored cross-site scripting

Question: What query string can be abused to craft a reflected XSS?
-q

now let's run owasp zap to perform automated scan on the santa's website



This screen allows you to launch an automated scan against an application. Please be aware that you should only attack applications that you have

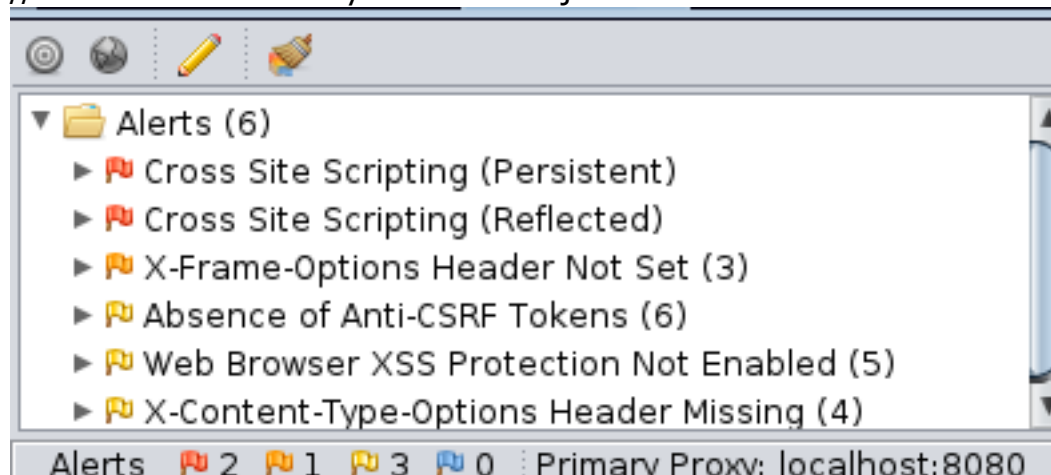
URL to attack:

Use traditional spider: ☒

Use ajax spider: ☐ with

Progress: Actively scanning (attacking) the URLs discovered

now here it shows that we've 6 alerts that owasp zap found here
// the 2 XSS vulnerability that we tried just now it also able to detect too



Question: Run a ZAP (zapproxy) automated scan on the target. How many XSS alerts are in the scan?
-2