Lab name: [Clickjacking with a frame buster script](https://0a7e002a041e77d580b953c8007c003c.web-security-academy.net/login)

Severity: High

Lab description:

* The lab demonstrates a Clickjacking attack where the target website attempts to defend itself using a frame buster script.
* Despite the frame buster, the attacker manages to bypass it and trick the user into interacting with the embedded UI.
* The objective is to successfully clickjack a button (e.g., "Change Email") from a malicious site while the original site is loaded in a transparent iframe.

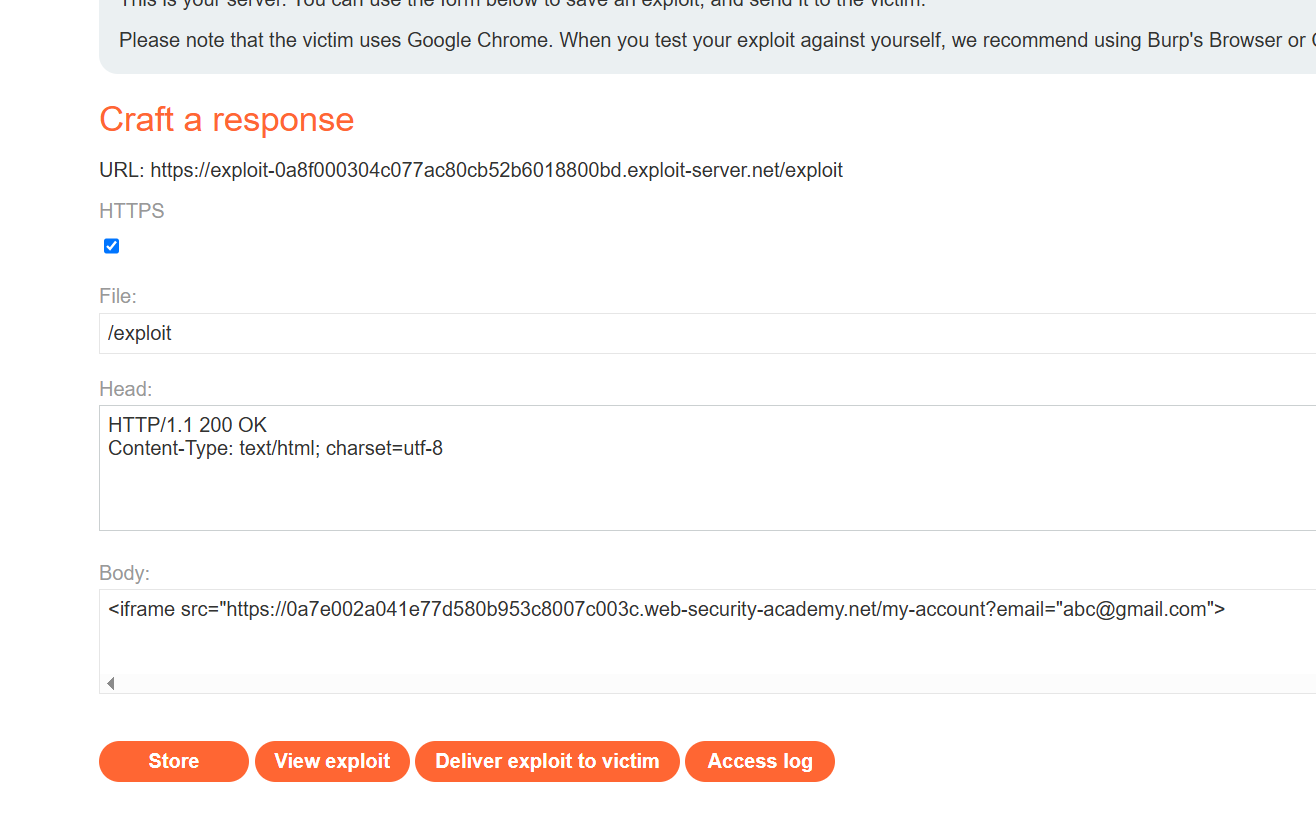
Impact:

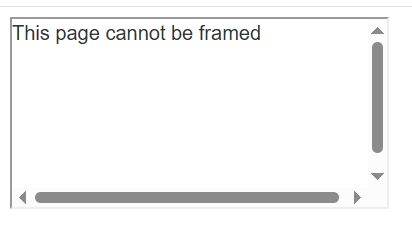
* The attacker can make the user perform actions without knowing, like changing their email.
* It can lead to account changes or even full account takeover.
* The attack breaks user trust and damages the website’s reputation.
* It can also be used with phishing to steal user data.
* The JavaScript protection doesn’t work in all cases, especially if sandboxed iframes are used.
* Security standards may be violated due to weak clickjacking protection.

Recommendations:

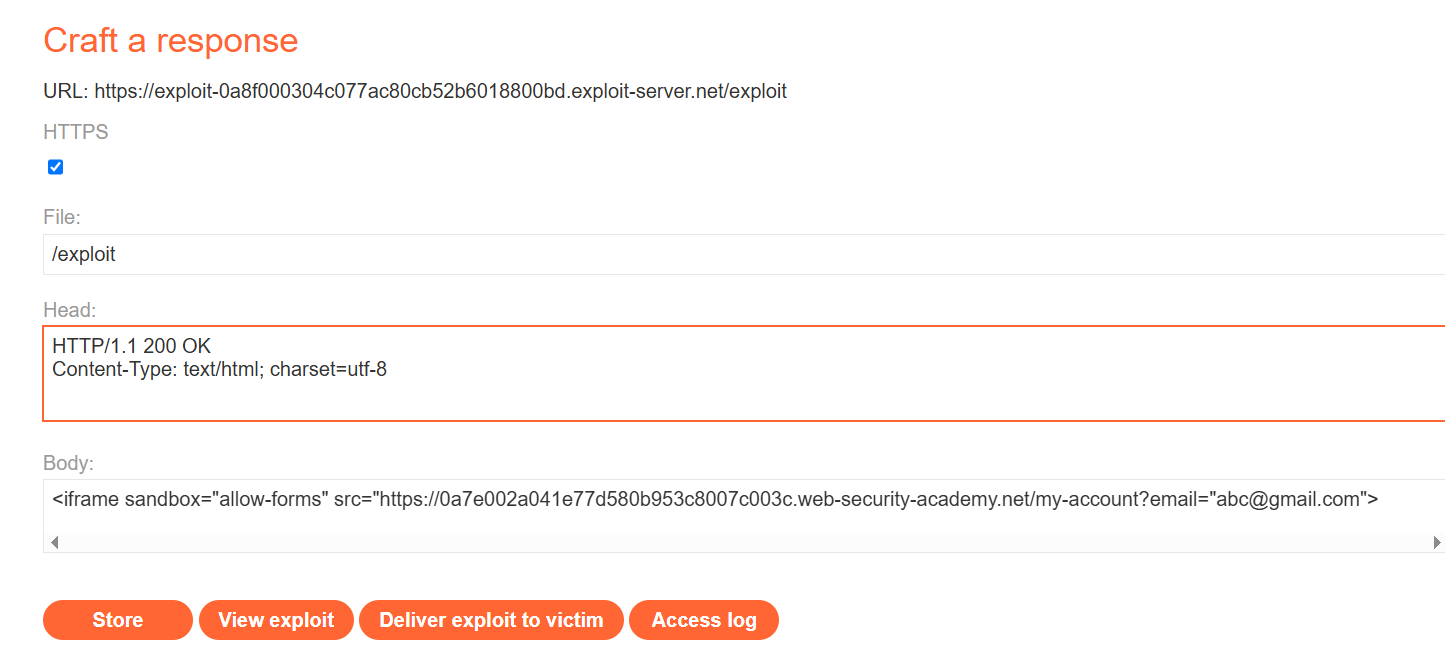
* To mitigate clickjacking risks, the application should implement the X-Frame-Options HTTP header with the value DENY or SAMEORIGIN to completely prevent framing by unauthorized domains. Usage of content-security-policy header also will help in preventing clickjacking.
* Developers should avoid depending on JavaScript-based frame busters, as they can be easily bypassed.
* For highly sensitive actions, incorporating confirmation prompts or requiring user re-authentication adds an additional layer of defense.
* Designing interfaces in a way that requires deliberate user interaction—such as dragging elements or multiple clicks—also helps reduce the risk of clickjacking.
* Lastly, it's important to raise awareness among users about clickjacking tactics and encourage safe browsing practices.

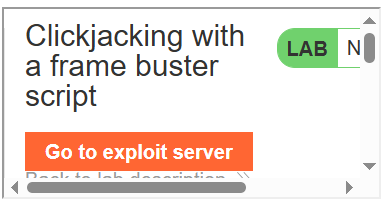
Steps to reproduce:

1. Access the lab and login with the username and password.
2. Notice that the iframe that doesn’t work normally.

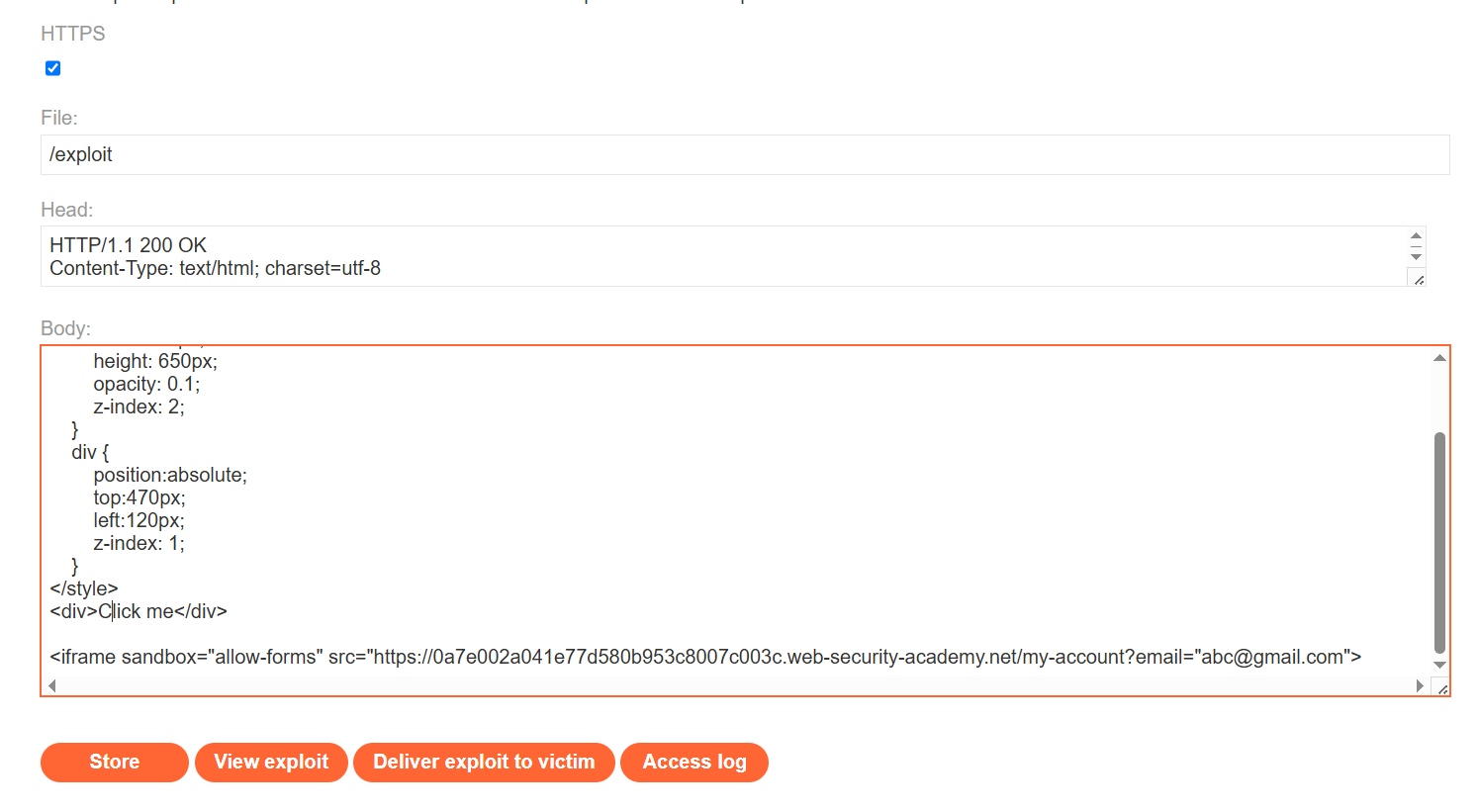


1. Use *sandbox=”allow-forms”* , and the iframe works, this is the vulnerability point.





1. Now the attacker could manipulate the users to click the update email button.



1. Store and deliver the exploit to victim, then the lab is solved.

