**CVE Submission Request**

**Author:** Jhony Alavez

**Sofware:** Cubex MyCubex (MyQLink Admin)

**Version:** 1.0.0.21

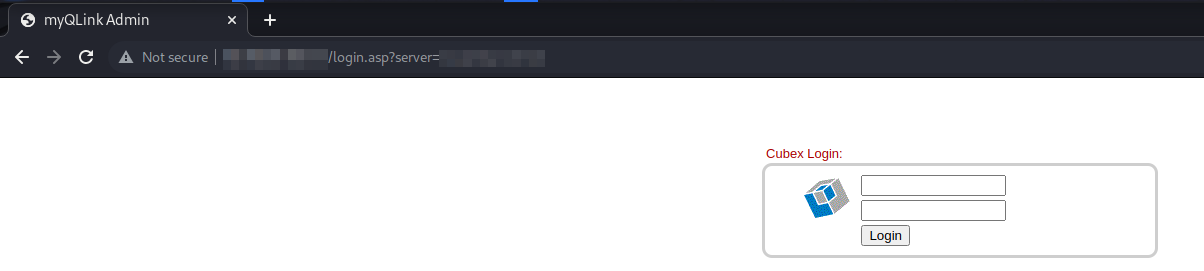
**Vulnerability:** Reflected Cross-Site Scripting (XSS)

**Description:** The Cubex MyQLink software is vulnerable to a Reflected Cross-Site Scripting (XSS) attack. A XSS attack allows attackers to hijack other users' online accounts on the affected website. This vulnerability was found via manually investigating Burp requests, and testing various payloads on vulnerable URL parameters using a parameter analysis and XSS scanning tool called Dalfox (<https://github.com/hahwul/dalfox>). The vulnerability was confirmed afterwards by observing the behavior of the application, in which the webpage reflected the valid XSS payload.

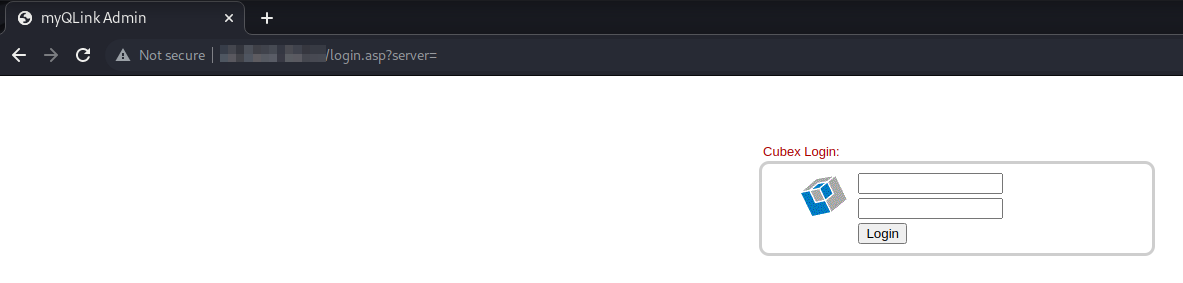
**Impact:** An attacker can use a cross-site scripting vulnerability to inject some malicious script into the vulnerable application. When a victim user encounters the script, it executes in the victim's browser. The XSS script can then perform any action that the victim is able to perform, and access all the victim's data. If the victim has special privileges within the application, or has access to sensitive data, this can constitute a serious vulnerability. If the same application resides on a domain that can access cookies for other more security-critical applications, then the vulnerability could be used to attack those other applications, and so may be considered high risk. Similarly, if the organization that owns the application is a likely target for phishing attacks, then the vulnerability could be leveraged to lend credibility to such attacks, by injecting Trojan functionality into the vulnerable application and exploiting users' trust in the organization in order to capture credentials for other applications that it owns.

**Recommendation:** In most situations where user-controllable data is copied into application responses, cross-site scripting attacks can be prevented using two layers of defenses. Firstly, input should be validated as strictly as possible on arrival, given the kind of content that it is expected to contain. Input which fails the validation should be rejected, not sanitized. Secondly, user input should be HTML-encoded at any point where it is copied into application responses. All HTML metacharacters, including < > " ' and =, should be replaced with the corresponding HTML entities (&lt; &gt; etc).

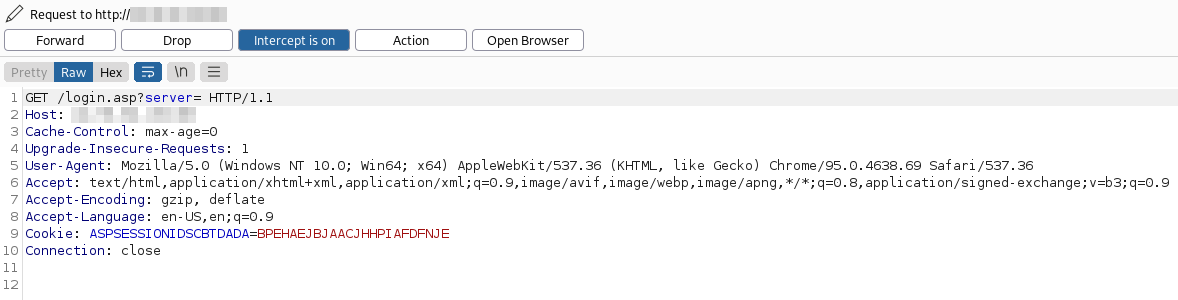
Reproduction Steps:



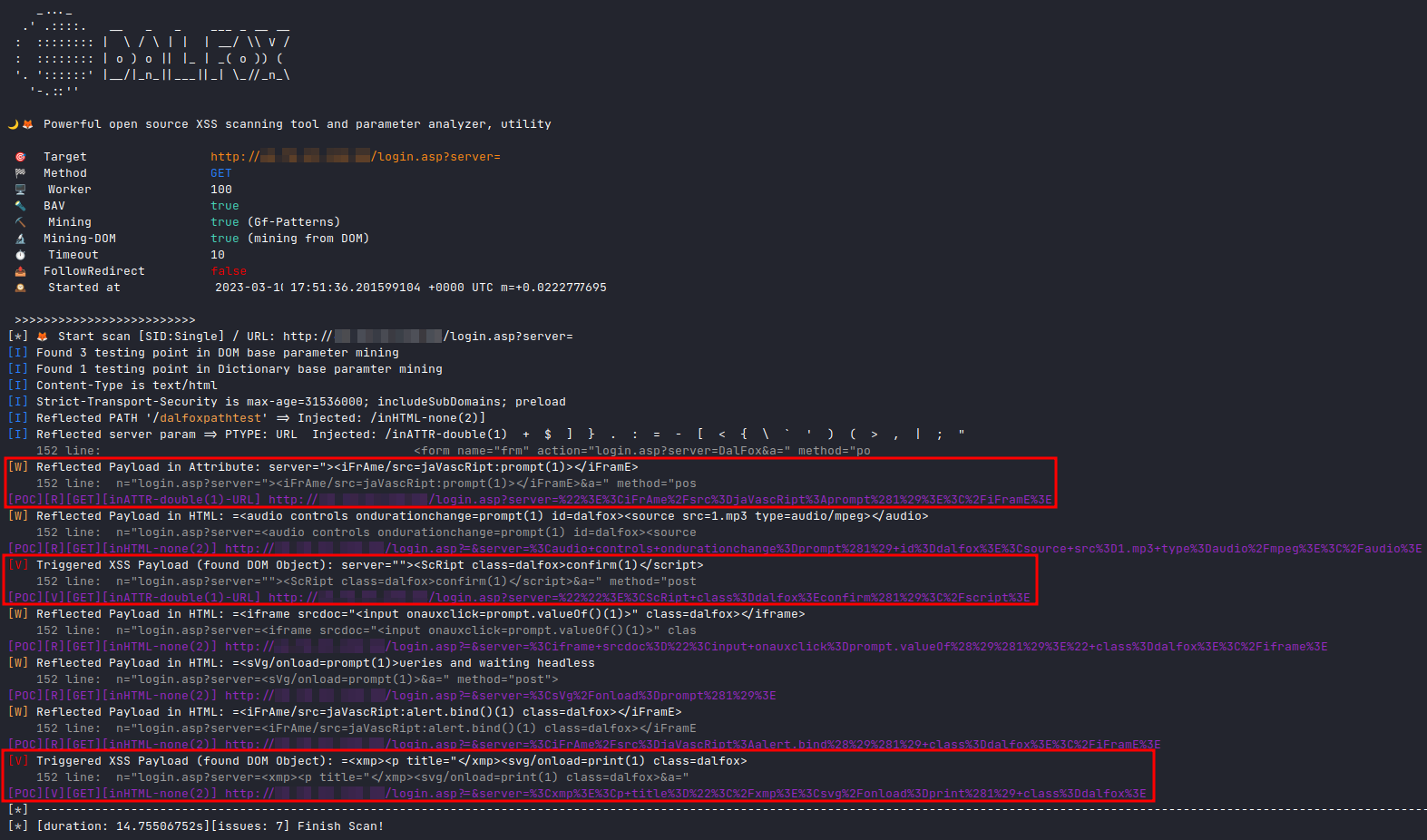
***Figure 1:*** Hosted Login Web Page of Cubex MyQLink



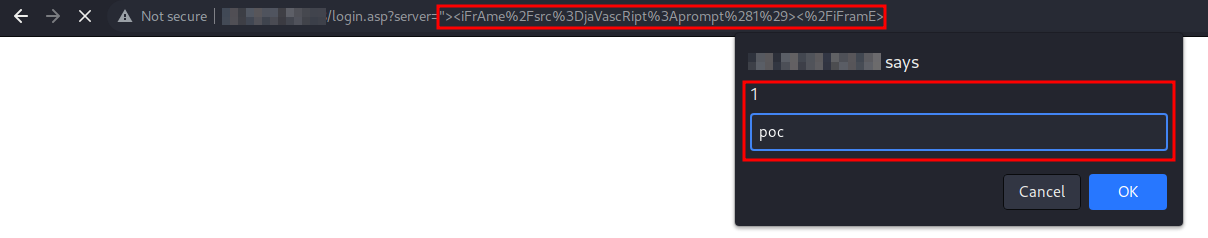
***Figure 2:*** Potential Vulnerable *‘server’* URL Paramenter Identified



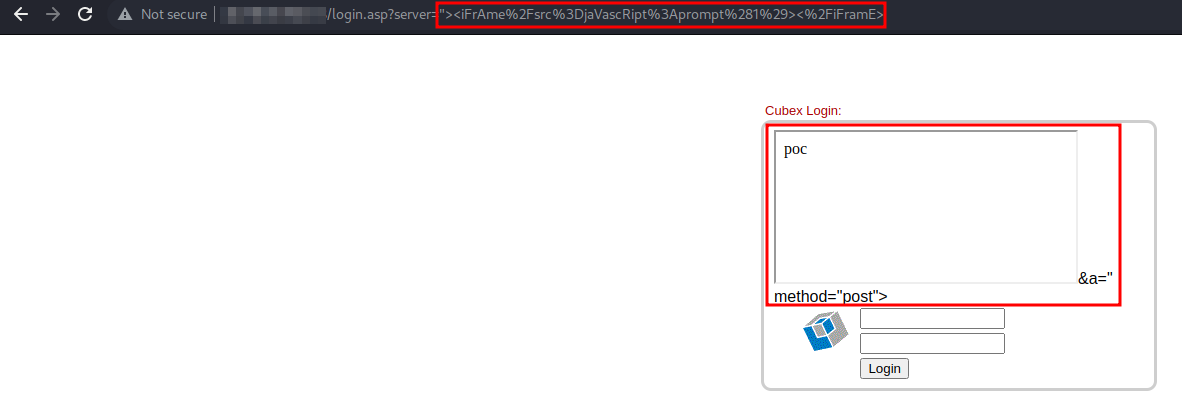
***Figure 3:*** Intercepted Raw HTTP Request of Vulnerable *'server’* Parameter via Burp



***Figure 4:*** Running Dalfox – XSS Scanner Tool – Located Three (3) Successful Reflected XSS Payloads Injected into ‘*server’* Parameter



***Figure 5:*** Proof of Exploited XSS Vulnerability with First Dalfox XSS Payload Result Injected into ‘server’Parameter



***Figure 6:*** Proof of Affected Cubex MyQLink Login Web Page