Proxy Vulnerability Report of "Unanimous Website"

Scope:

Hackers often exploit proxy vulnerabilities to make incursions into web application systems. It is shocking to uncover different ways attackers utilize intercepting proxies such as: Burp Suite, Webscarab and Paros to break into websites.

Below is a demonstration of a security login flaw that bad guys can use to illegally access a website. The following demo is put together with Burp Suite to show how a website login form can be hacked without signing-in. The website is a venues search engine, whose name won't be revealed for security purposes. **P.S.** It is highly recommended NOT to use anything here to hack a website without proper written consent of the website's owner. Committing to do otherwise with anything explained here is at your own risk.

Walk Through

The first step is to type any random username and password in acceptable formats and click sign-in. Right after that, obviously with Burp Suite being turned on, Burp Suite intercepts the web traffic to display the post, cookie, username and password that are associated with what information submitted to access the sign-in environment of the website ("Proxy" and "Intercept" tabs display these information).

Next, right-click on the Post intercepted information to send it to intruder. Burp Suite helps configured the details target of the attack by revealing the host, port and hypertext protocol of the website ("Intruder" and "Target" tabs display these information). Identify potential targets information via "Intruder" and "Positions" tabs.

Follow-up by selecting cluster bomb as the type of attack. Afterwards, move to the "Payloads" tab to choose the payload set number and type (could be the number of choice and simple list as the payload type are recommended), then add the initial/random username selected to successfully access the login environment of the website. Repeat the previous steps to add the password entered.

Lastly, click on Start Attack. The success of this attack can be verified by accessing "Intruder," "Payloads" and "Results" tabs to confirm any information discrepancy for the request within the length status column. This information should be the ones associated with the invalid username and password entered to log-into the website. Clicking on this information row reveals what were posted for the request plus the responses received. If ASP.net.ApplicationCookie information is revealed (which is the result from the experiment, but did not post critical portion of info for security purposes), the attack succeeded. Go and try to log-into the website using user@test.com or whatever "username" and "password" that were selected without creating formal credentials, and you should successfully log-in.

Mitigation Strategy Recommendations

I am not yet aware of formal technical solutions to mitigate this risk; however, I'm going to share a couple recommendations to help eliminate this login flaw:

- Make sure your web developers work hand-in-hand with cyber security professionals
- Hire pen-testers to evaluate proxy vulnerability and overall security of your web platforms (Written consents for all parties involved are highly recommended prior to testing sessions)
- Align your web environment with the best and up-to-date security tools
- Perform maintenance and system reviews regularly