Web Security - Session Security Assignment

Stefano Calzavara

Università Ca' Foscari Venezia

Università
Ca'Foscari
Venezia

Breaking Web Session Integrity

Three classic attacks against session integrity:

1. session hijacking: if the attacker can steal the victim's cookies, they can impersonate the victim at the server
2. session fixation: if the attacker can fix the victim's cookies to a known value, they can impersonate the victim at the server
3. cross-site request forgery: if the attacker can forge requests from the victim's browser, such requests might look legitimate to the server

This year, we will experiment with session fixation!

Preliminaries

The activity has an associated web application:

1. Download the corresponding ZIP archive from Moodle
2. Configure the `/etc/hosts` file (or any similar file from your distro) so that `www.vulnerable.com` resolves to the local host 127.0.0.1: this way, you can serve content from it
3. Extract the archive and run the Flask application inside it
4. Ensure you can access `www.vulnerable.com` (port 5000) using a first browser, e.g., Google Chrome: this is the victim's browser
5. Ensure you can access `www.vulnerable.com` (port 5000) using a second browser, e.g., Mozilla Firefox: this is the attacker's browser

Session Fixation

In a session fixation attack:

1. The attacker visits `www.vulnerable.com` from their own browser and acquires a session identifier (without authenticating)
2. The victim visits `www.vulnerable.com` from her browser, passing the attacker's session identifier
3. The victim authenticates at `www.vulnerable.com`, which does not refresh the session identifier upon login
4. The attacker finally uses the known session identifier to access `www.vulnerable.com` with the identity of the victim

Assignment

The web application suffers from a session fixation vulnerability, which the attacker must exploit to delete the victim's account:

1. Exploit the vulnerability following the steps in the previous slide, filling in the missing bits and using the two browsers as appropriate
2. Once you gain enough familiarity, record a short video (2/3 minutes at most) showing the attack at work on your PC
3. Write a short report discussing the attack steps and proposing a simple fix to prevent it
4. Submit a ZIP archive on Moodle including the video, the report and a patched version of the web application implementing your fix