Activity 9 : Web Security Scanner

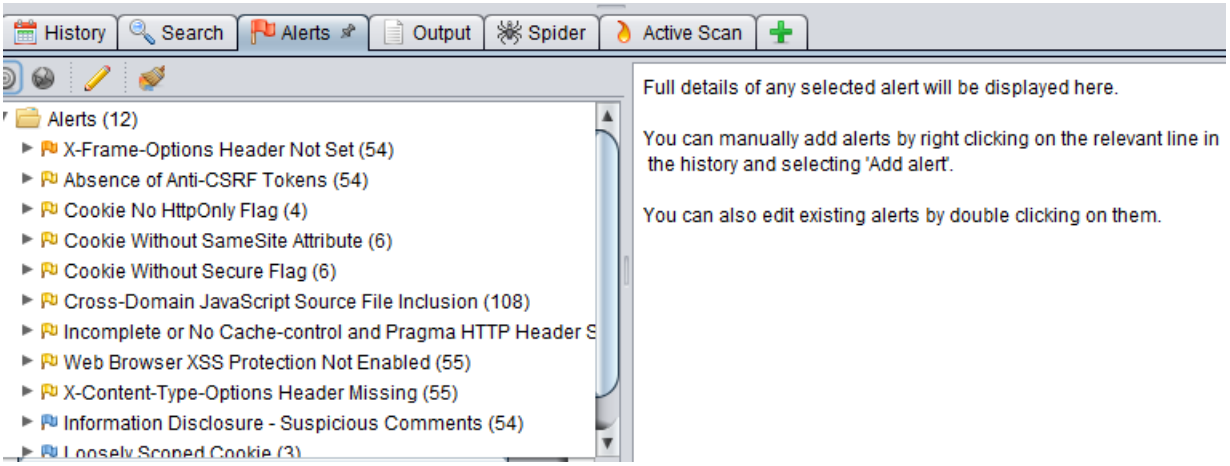
Instructors : Krerk Piromsopa, Ph.D

Overview

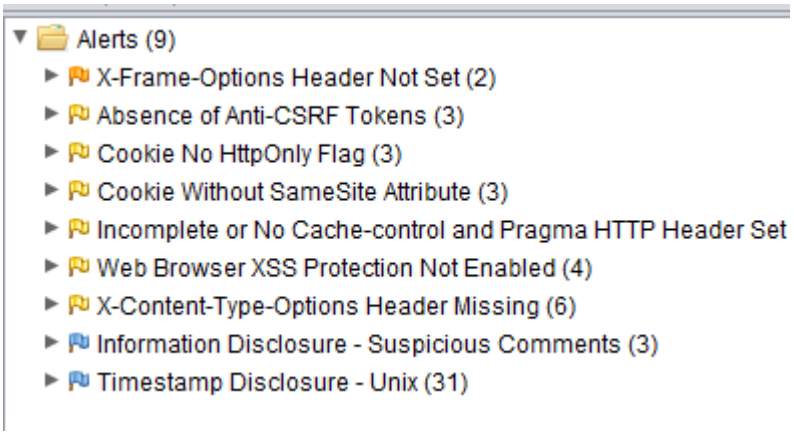
With the web everywhere, we should learn the fundamentals of secure web application. In this activity, we will use OWASP Zed Attack Proxy[1] to test web applications.

Exercise

1. We will try to test sites with OWASP ZAP. Run OWASP in ZAP in standard mode. *** Don't try attack mode with real sites. *** Test the following sites:
    a. https://www.chula.ac.th/

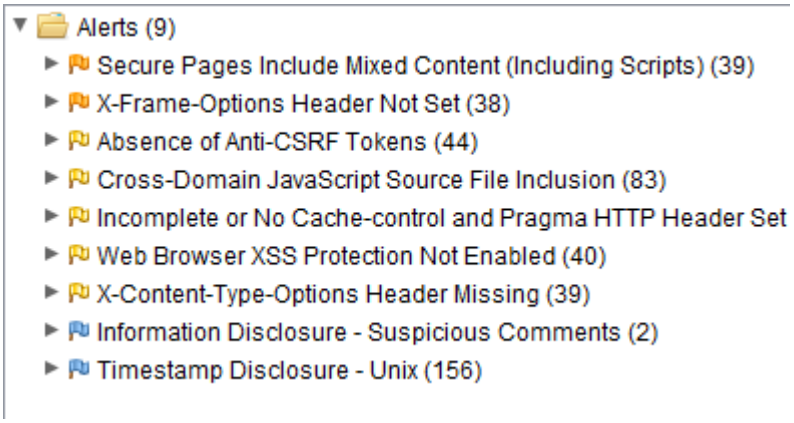


    b. https://www.eng.chula.ac.th/

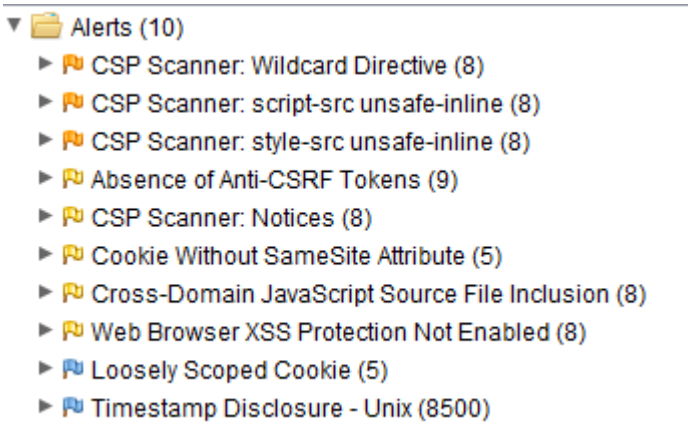


[1] https://www.zaproxy.org/

c. https://www.cp.eng.chula.ac.th/



d. https://www.facebook.com/ (You name it.) Facebook, Google ? What do you see?



Alerts from facebook is detected slowly compare to chula-ish website

Computer Security
Dept. of Computer Engineering,
Chulalongkorn University.

---

2. Use OWASP ZAP in proxy mode. (Install Web Driver Add-ons) Browse a web site (e.g. facebook) What do you see?

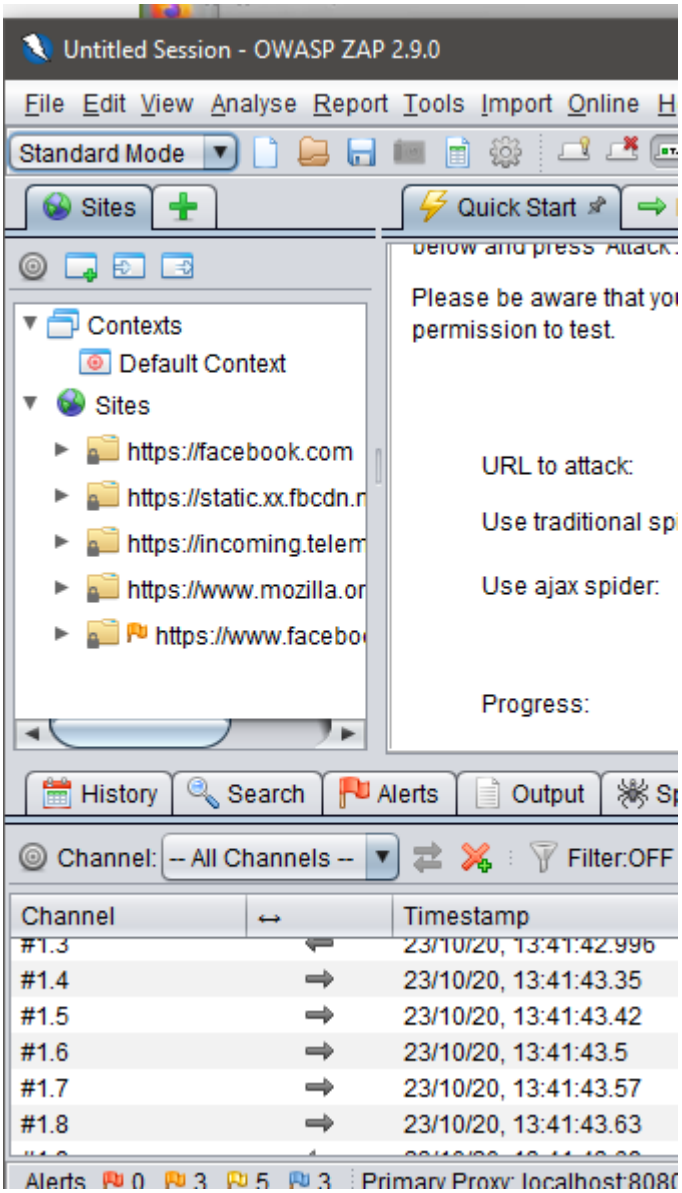
   ZAP as proxy. By default, ZAP uses:

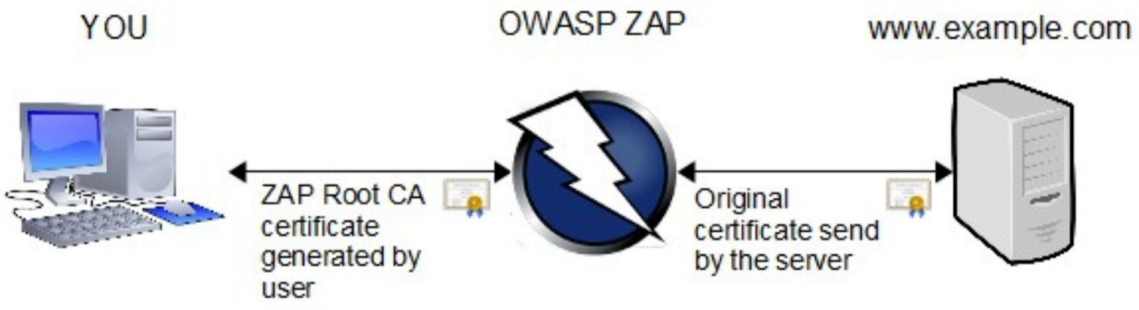
   Address: localhost

   Port: 8080

   Do ZAP certificate (if needed)

   The browsing website appear on OWASP ZAP





3. Suppose that you are now a security analysis. Please scan 3 sites in the domain **chula.ac.th** . Identify possible security issues. Write a report (1-page A4) to suggest to the admin how to improve the security.

1. www.chula.ac.th
    a. X-Frame-Options Header Not Set
        i. <Risk: Medium > *given from zap hence omit impact/likelihood
        ii. Description
        This website could be at risk of a clickjacking attack. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames. Detect on every endpoint.
        iii. Recommendation
            1. Sending the proper X-Frame-Options in HTTP response headers
            2. Employing defensive code in the UI to ensure that the current frame is the most top level window.
    b. Absence of Anti-CSRF Tokens
        i. <Risk: Medium >
        ii. Description
        A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. Anti-CSRF tokens are used in web applications to prevent (CSRF/XSRF).
        Evidence: <form class="searchsite-form clearfix" role="search" method="get" action="https://www.chula.ac.th/" autocomplete="off">
        iii. Recommendation
        Use Anti-CSRF token for specific form
2. www.eng.chula.ac.th
    a. Cookie Without SameSite Attribute
        i. <Risk: Low>
        ii. Description
        This can be abused to do CSRF attacks. Recently a new cookie attribute named SameSite was proposed to disable third-party usage for some cookies, to prevent CSRF attacks. Same-site cookies allow servers to mitigate the risk of CSRF and information leakage attacks.
        iii. Recommendation
        Can be ignore or adopt the use of samesite attribute
3. www.cp.eng.chula.ac.th
    a. Secure Pages Include Mixed Content (Including Scripts)
        i. <Risk: Medium>
        ii. Description
        Mixed content is HTTPS page loads HTTP content – HTTP is insecure, and attackers can read/modify HTTP traffic. Hence, Implement secure pages using mixed content is not secure from abuse through HTTP. In example, When the image is loaded over HTTP, the attacker can change this image.
        Evidence: http://www.cp.eng.chula.ac.th/wp-content/plugins/bxslider-integration/assets/js/bxslider-integration.min.js?ver=5.4.2
        iii. Recommendation
        Implement secure page over HTTPS page only