CONFIABILIDADE E SEGURANÇA CIBERNÉTICA
(DACYS)
PL06
*Jorge Pinto Leite (**JPL**)*
*Pedro Sousa Rodrigues (**DCR**), Nuno Peralta* (**NSP**)
{@isep.ipp.pt}

Exercise 1: Application Security - Setup

The Internet grew exponentially with browsers. The browser war was to no avail, they rule the Internet and with them, the HTTP protocol became ever so relevant. With the evolution of the protocol and the Internet altogether, accompanied by the need for user interaction, a new attack surface surged.
In the following exercises, we will train the exploitation of such vulnerabilities.

1. In your Kali Virtual machine install Docker.io by using: **apt install docker.io**;
2. Run **WebGoat** [1] by issuing the following command: **docker run -it -p 127.0.0.1:8081:8080 -p 127.0.0.1:9090:9090 -e TZ=Europe/Amsterdam webgoat/webgoat**
3. Navigate your browser to `http://127.0.0.1:8081/WebGoat/` and create a new user account
4. Start **BurpSuite** [2] (already available in your Virtual Machine);
5. Configure **FoxyProxy** [3] to route your browser traffic through BurpSuite.

Exercise 2: Application Security - Execution

Now that you have your environment ready the next step is to train the exploitation of vulnerabilities.

1. Solve the A1 section - Broken Access Control;
   i) How should you protect an authentication mechanism? What security-sensitive features should you include to fix these exercises?
2. Solve the A3 section - Injection
   i) What is a **SQL Injection** [4] vulnerability?
   ii) Does using a **TLS-enabled Database Connector** prevents SQL injection attacks from happening?
   iii) Does the use of a **WAF** [5] fixes the danger of the SQL injection vulnerability?
   iv) What is the danger of a **Path Traversal** [6] Vulnerability?

v) What is **XSS's** [7] main target?

3. Solve the A7 section - Identity & Auth Failure

   i) What is **JWT** [8]?

   ii) State two standard misuse of **JWT tokens**

   iii) Imagine that you are a developer tasked to implement Secure passwords. What should be your requirements to consider a secure password?

Exercise 3: Protection

Now that you know how some attacks are performed, you also need to think about how to detect and protect.

1. Imagine that you want to expose your application to the Internet in a secure and reliable way, what should be your considerations (taking into account logging, resilience, and attack prevention).

2. Investigate on security features such as **SELinux** [9] and **AppArmour** [10]. In what ways can it help in mitigating security problems?

3. How **Secure Communications** align with application security? Debate the necessary considerations on assuring secure communications to better protect clients and your application.

References

[1] WebGoat, "Webgoat/webgoat: Webgoat is a deliberately insecure application." [Online]. Available: https://github.com/WebGoat/WebGoat

[2] "Burp suite - application security testing software." [Online]. Available: https://portswigger.net/burp

[3] "Foxyproxy standard," Apr 2006. [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/

[4] "Sql injection." [Online]. Available: https://owasp.org/www-community/attacks/SQL_Injection

[5] "Web application firewall." [Online]. Available: https://owasp.org/www-community/Web_Application_Firewall

[6] "Path traversal." [Online]. Available: https://owasp.org/www-community/attacks/Path_Traversal

[7] "Cross site scripting (xss)." [Online]. Available: https://owasp.org/www-community/attacks/xss/

[8] M. Jones, J. Bradley, and N. Sakimura, "Json web token (jwt)," May 1970. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7519

[9] "What is selinux?" [Online]. Available: https://www.redhat.com/en/topics/linux/what-is-selinux

[10] "Apparmor." [Online]. Available: https://apparmor.net/