Web Application Penetration Testing Design Document

Platform: https://www.daedalus-systems.co.uk/ **Domain**: technology & ecommerce website

1. Introduction

Penetration testing is a sequence of activities undertaken to diagnose, catalogue and exploit weaknesses in a system's security (Bacudio et. al, 2011; Cohen, 1997). Effective web penetration testing (WPT) can result in: a) a long-term reduction in costs, b) improved technical efficiency c) increased confidence in the security of your IT environments, d) better awareness of the necessity for appropriate technical controls (Creasey, 2017).

The scope of this design document is to present a comprehensive WPT plan, identifying a testing methodology and the regulations which need to be met. The document lists the potential security vulnerabilities within Daedalus Systems platform, and the tools to be used to detect and exploit these vulnerabilities. The limitations of our approach and any assumptions we have made are identified, and a timeline to complete the testing and final report is included.

We will perform a black box penetration test, as we only have access to public information about the system (Conrad, 2015).

2. Chosen Methodology

Our security testing methodology combines PTES (Penetration Testing Execution Standard) and the OWASP Web testing framework. PTES provides a structured approach to penetration testing, detailing a seven phase procedure (PTES, 2014). We will use a combination of both manual and automatic testing as both approaches have their benefits (Stefinko, 2016). After detecting vulnerabilities, security flaws or threats in the application, we examine each vulnerability in detail and provide suggestions and recommendations for improved security. The timeline is presented in figure 3.

3. Threat Modelling

We will follow the Microsoft Threat Modelling Process, which is used to identify and list potential threats, security vulnerabilities and inappropriate defence mechanisms. To identify the security objectives of the site and threats to it, we use the STRIDE modelling process, shown in the appendix, figure 1. We use the DREAD risk analysis model to identify vulnerabilities, and assess their seriousness. (Meier, 2003) A table showing the scale used to measure the vulnerabilities is included in the appendix, figure 3.

4. Regulatory Compliance

Our executive report will provide guidance for our client compliance with the three most relevant regulatory compliance regulations as shown in Table 1: (ISO, 2013; PCI, 2008; Voigt and Bussche, 2017)

Table1: Regulatory Compliance

- **GDPR (General Data Protection Regulation):** EU legislation giving consumers control over how their personal data is gathered and stored.
  Potential vulnerabilities: weak and stolen credentials (due to the use of off-the-shelf software, applications and plugins containing vulnerabilities), application vulnerabilities (such as potential for injection, privilege escalation and cross-site scripting).

- **Iso271001 (formally known as ISO/IEC 27001:2005):** a framework for managing IT security that helps keep consumer data safe in the private and public sector.
  Potential vulnerabilities: Data breaches.

- **PCI DSS (Payment Card Industry - Data Security Standard):** technical requirements set by the Payment Card Industry Security Standards Council to protect cardholder data.
  Potential vulnerabilities: theft of cardholder data.

5. List of Tools and Justifications

Based on PTES methodology, we here present a chronological list of the tools we will use, along with justifications:

- **Pre-engagement**: This document outlines the goals and maps out the scope of the Pentest.
- **Information Gathering:** We find available information about the target in various ways: a) we use Network Utilities (whois, ping, nslookup) to identify the owner, hosts, location of servers, IP address, server type. b) we use Wappalyzer to identify technologies of web application; c) we discover sensitive files with Dirbuster, an old tool to discover files/folders of a web application (Dahle, 2020).
- **Threat Modelling:** In this pre-attack phase, we will: a) discover unused or hidden sub-domains registered with the organisation using Knockpy; b) we will identify the Operating System, any open ports, MAC addresses etc. using NMAP, an open-source tool for network investigation and security auditing. It can efficiently scan large networks but also functions adequately on single hosts (Dahle, 2020).
- **Vulnerability Analysis:** We identify potential vulnerabilities (the OWASP Top 10) using Burp Suite, an efficient discovery tool that looks for directories and files (Dahle, 2020).
- **Exploitation phase:** We use a combination of automated and manual penetration tools and techniques to try to breach the website through the entry-points identified in the threat modelling phase. Here we actively exploit security weaknesses, trying to gain control of the host server or elevating our user privilege to access sensitive information, or conduct a denial-of-service attack. For this, we use Burp Suite, OWASP ZAP and Arachni. Figure 5 presents an exploits diagram showing the entire process (Ami & Hasan, 2012).
- **Post Exploitation:** We document the methods and keep notes and screenshots of the attacks. We will gather information about the exploited system, identify and document interesting files, attempt to elevate our privileges where necessary, explore other machines or applications on the network (Weidman, 2014).
- **Reporting:** Our Pentest Report will contain the generally recommended sections, such as Summary, Technical Detail, Findings, Risk Level Indication and Time Estimation (Ami & Hasan, 2012)

7. Potential impacts on normal operations

Active penetration attacks which send data to the target system can potentially damage or disrupt the system (Conrad, 2015). Detailed below are the most relevant potential negative impacts of the pentesting tools we will use.

- **Complications with Availability (DoS)** – The website may become temporarily unavailable to visitors during our tests.
- **Unintended disclosure of sensitive information** – Our team ensures System Integrity and Data Integrity of any sensitive data that is encountered during the testing process.
- **Disturb the system performance** - We may temporarily disrupt or negatively impact the performance of the website.
- **Email floods and database junk** - To minimise the risks of this occurring, we will mainly conduct manual tests. Automated tests will be restricted to the end of the testing window..

Research experiments highlight how important the choice of input parameters is when using pentesting tools. These allow testers to identify what is to be considered by the tool, assisting in its specification and standardisation (De Lima et. al., 2020).

8. Assumptions & limitations

Table 2: Assumptions and Limitations

- **Assumption of availability:** The web application will always be available to be tested.
- **Assumption of effectiveness:** A single penetration testing will not be enough so the company will schedule a penetration test regularly and take necessary actions.
- **Limitation of scope:** It is impossible to predict all attack scenarios, so only the most common attacks will be considered in our testing.
- **Limitation of time:** the specified time period is 6 weeks.

REFERENCES

Ami, P., & Hasan, A. (2012). Seven phrase penetration testing model. International Journal of Computer Applications, 59(5), 16-20.

Bacudio, A. G., Yuan, X., Chu, B. T. B., & Jones, M. (2011). An overview of penetration testing. International Journal of Network Security & Its Applications, 3(6), 19.

Cohen, F. (1999). Managing network security: attack and defence strategies. Network Security, 1999(7), 7-11.

Conrad, E., Misenar, S., & Feldman, J. (2015). *CISSP study guide*. Syngress.

Creasey, J. (2017). A guide for running an effective Penetration Testing programme. Crest, (April), 1-64.

Dahle, T. K. (2020). Large scale vulnerability scanning (Master's thesis).

de Lima, L. F., Horstmann, M. C., Neto, D. N., Grégio, A. R., Silva, F., & Peres, L. M. (2020, September). On the Challenges of Automated Testing of Web Vulnerabilities. In 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (pp. 203-206). IEEE.

Meier, J. D. (2003). Improving web application security: threats and countermeasures. Microsoft press. Available from: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)

PTES (2014). PTES Technical guideline. Available from: http://www.pentest-standard.org/index.php/Main_Page

ISO/IEC (2013) Information technology — Security techniques — Information security management systems — Requirements. Available from: https://www.iso.org/standard/54534.html

PCI (2008). Payment Card Industry Security Standards. Available from: https://www.pcisecuritystandards.org/pdfs/pcissc_overview.pdf

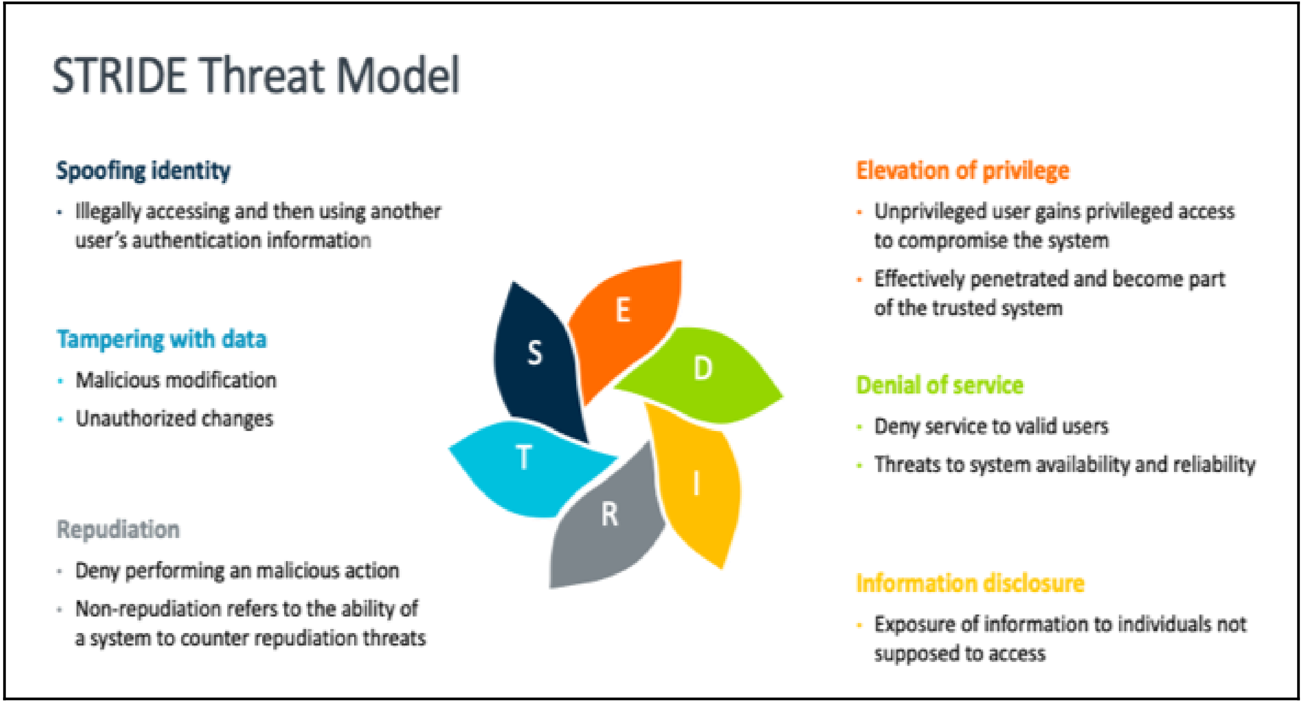
Stefinko, Y., Piskozub, A., & Banakh, R. (2016, February). Manual and automated penetration testing. Benefits and drawbacks. Modern tendency. In 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET) (pp. 488-491). IEEE.

Voigt, P., & Von dem Bussche, A. (2017). The EU general data protection regulation (GDPR). A Practical Guide, 1st Ed., Cham: Springer International Publishing, 10, 3152676.

Weidman, Georgia. (2014). Penetration testing: a hands-on introduction to hacking. San Francisco: No Starch Press.

APPENDIX

Figure 1 : STRIDE threat model



STRIDE Threat Model

Spoofing identity

- Illegally accessing and then using another user's authentication information

Tampering with data

- Malicious modification
- Unauthorized changes

Repudiation

- Deny performing an malicious action
- Non-repudiation refers to the ability of a system to counter repudiation threats

Elevation of privilege

- Unprivileged user gains privileged access to compromise the system
- Effectively penetrated and become part of the trusted system

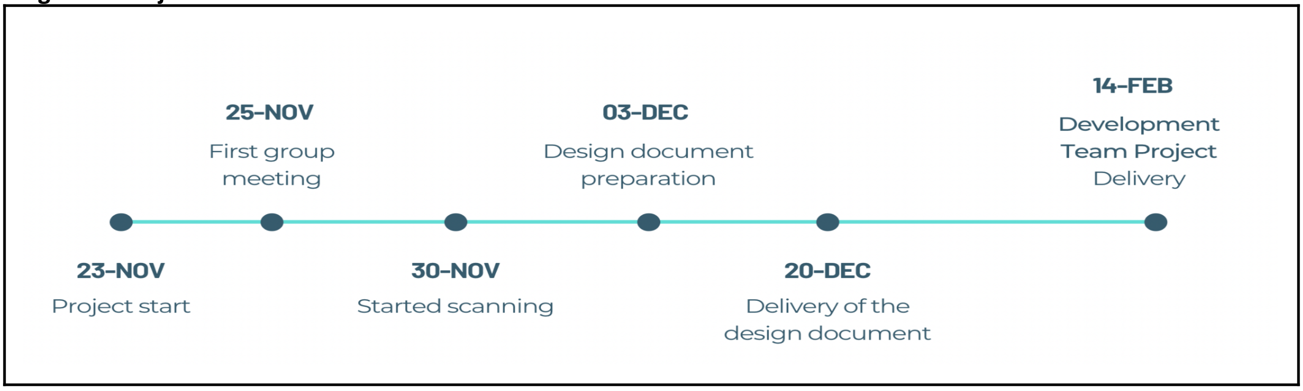
Denial of service

- Deny service to valid users
- Threats to system availability and reliability

Information disclosure

- Exposure of information to individuals not supposed to access

Figure 2: Project Timeline



23-NOV Project start

25-NOV First group meeting

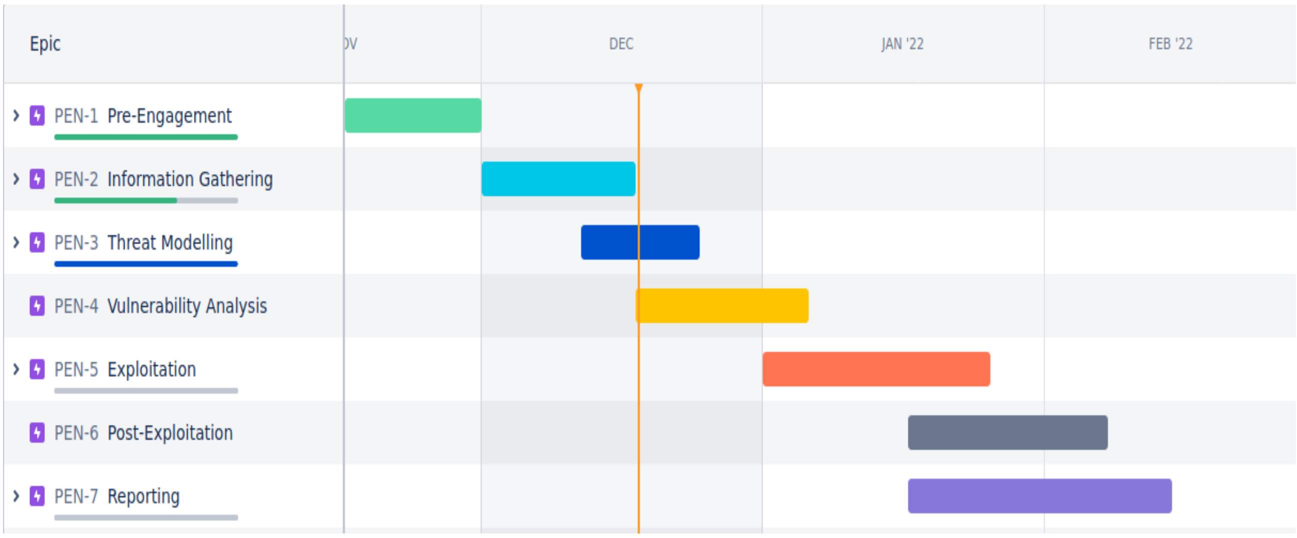
30-NOV Started scanning

03-DEC Design document preparation

20-DEC Delivery of the design document

14-FEB Development Team Project Delivery

Figure 3: Phases Timeline



PEN-1 Pre-Engagement

PEN-2 Information Gathering

PEN-3 Threat Modelling

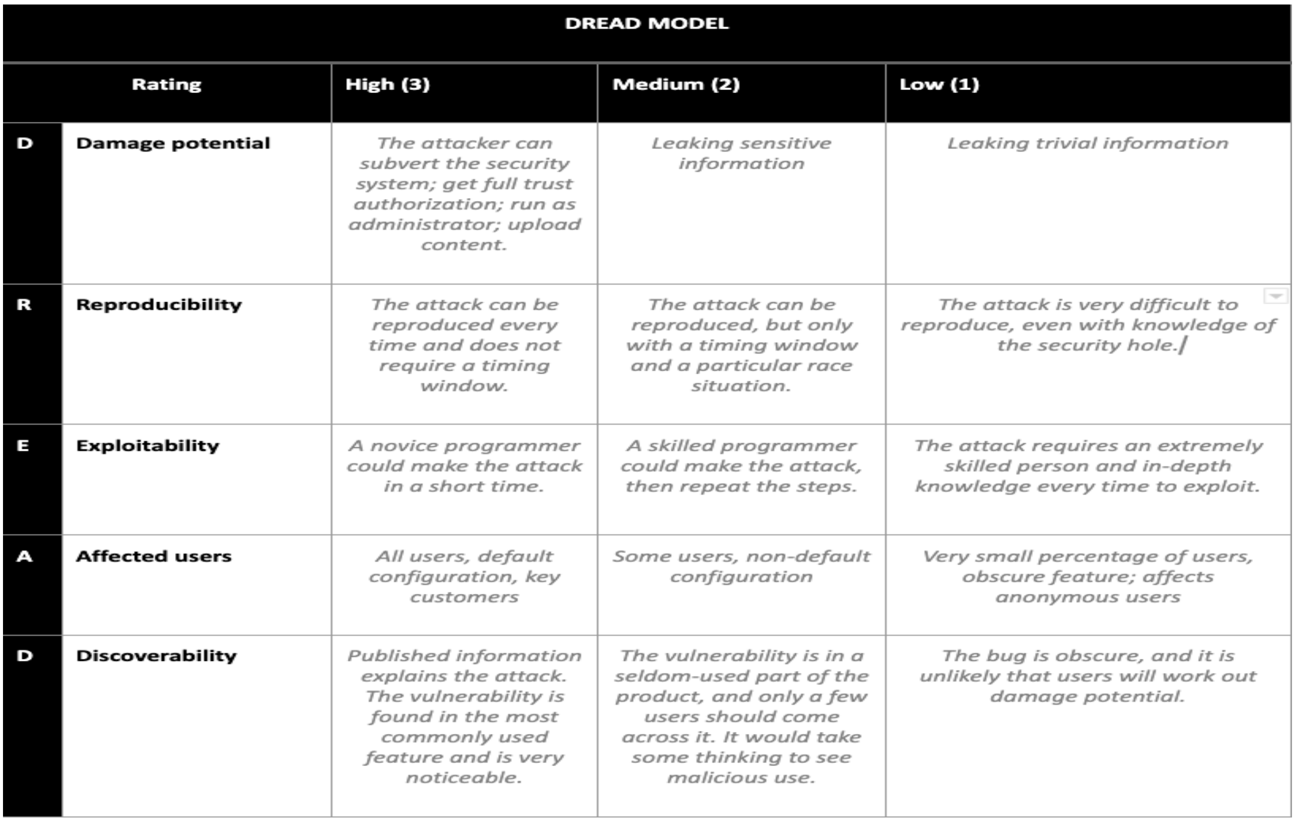
PEN-4 Vulnerability Analysis

PEN-5 Exploitation

PEN-6 Post-Exploitation

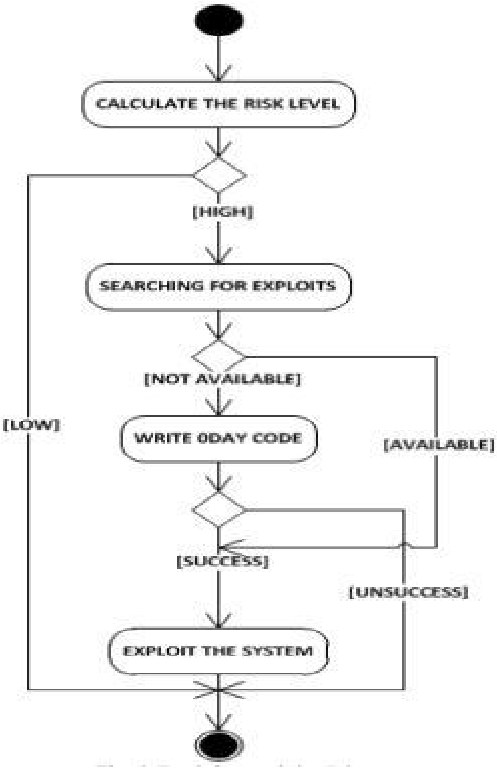
PEN-7 Reporting

Figure 4: DREAD ratings



DREAD MODEL				
Rating		High (3)	Medium (2)	Low (1)
D	Damage potential	*The attacker can subvert the security system; get full trust authorization; run as administrator; upload content.*	*Leaking sensitive information*	*Leaking trivial information*
R	Reproducibility	*The attack can be reproduced every time and does not require a timing window.*	*The attack can be reproduced, but only with a timing window and a particular race situation.*	*The attack is very difficult to reproduce, even with knowledge of the security hole.*
E	Exploitability	*A novice programmer could make the attack in a short time.*	*A skilled programmer could make the attack, then repeat the steps.*	*The attack requires an extremely skilled person and in-depth knowledge every time to exploit.*
A	Affected users	*All users, default configuration, key customers*	*Some users, non-default configuration*	*Very small percentage of users, obscure feature; affects anonymous users*
D	Discoverability	*Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable.*	*The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use.*	*The bug is obscure, and it is unlikely that users will work out damage potential.*

Figure 5: Exploitation Phase


CALCULATE THE RISK LEVEL
[HIGH]
SEARCHING FOR EXPLOITS
[NOT AVAILABLE]
[LOW]
WRITE 0DAY CODE
[AVAILABLE]
[SUCCESS]
[UNSUCCESS]
EXPLOIT THE SYSTEM