Attestation

of executed security audit on the basis of OWASP ASVS for:

Live Engage Sp. z o. o.
ul. Św. Marcina 29/8; 61-806 Poznań
NIP: 1133075233

TestArmy Group S.A.
ul. Petuniowa 9/5, 53-238 Wrocław
NIP: 8992754194
REGON: 022426575, KRS: 0000873700

no 025/2025/74

1. ATTESTATION DETAILS

TestArmy Group S.A. confirms that we have conducted a security audit, penetration testing on the **Live Engage Sp. z o. o.** – test made on system **TrueEngage**
Penetration Testing:
· Web application testing
· Available API interface testing
· External application infrastructure testing
using the black-box and grey-box method following the OWASP ASVS methodology standard.

All above mentioned showed the correct security on level 2 ASVS on the day of 15.07.2025

*Medium, high and critical vulnerabilities detected during the audit were repaired and verified during the retest, and for low vulnerabilities, the adequate risk acceptance was applied or an improvement plan was presented to secure them.

ABOUT THE STANDARD ASVS AND LEVELS

The ASVS is a community-effort to establish a framework of security requirements and controls that focus on normalising the functional and non-functional security controls required when designing, developing and testing modern web applications.
The Application Security Verification Standard is a list of application security requirements or tests that can be used by architects, developers, testers, security professionals, and even consumers to define what a secure application is.

Using the Application Security Verification Standard ASVS has two main goals:
- to help organizations develop and maintain secure applications
- to allow security service, security tools vendors, and consumers to align their requirements and offerings Application Security Verification Levels

The Application Security Verification Standard defines three security verification levels, with each level increasing in depth.

- ASVS Level 1 is meant for all software.
- ASVS Level 2 is for applications that contain sensitive data, which requires protection.
- ASVS Level 3 is for the most critical applications - applications that perform high value transactions, contain sensitive medical data, or any application that requires the highest level of trust.

Each ASVS level contains a list of security requirements. Each of these requirements can also be mapped to security-specific features and capabilities that must be built into software by developers.

1.1 Level 2: Standard

An application achieves ASVS Level 2 (or Standard) if it adequately defends against most of the risks associated with software today. Level 2 ensures that security controls are in place, effective, and used within the application. Level 2 is typically appropriate for applications that handle significant business-to-business transactions, including those that process healthcare information, implement business critical or sensitive functions, or process other sensitive assets. Threats to Level 2 applications will typically be skilled and motivated attackers focusing on specific targets using tools and techniques that are highly practiced and effective at discovering and exploiting weaknesses within applications.

2. APPROACH TO VERIFICATION

2.1. Automatic tools

• Burp Proxy Professional - it's a set of many cooperating tools supporting the performance of penetration tests of web applications. On the one hand, it has some built-in automatisms improving repetitive works and, on the other hand, it provides the pentester with rather large area of freedom of actions. It is composed of the following modules:

- http proxy (to hijack communication from the internet browsers or other communication software with the use of HTTP protocol);
- repeater (to manually modify hijacked HTTP requests or to create any completely new HTTP requests);
- scanner (automatic scanner detecting vulnerabilities in web applications);
- intruder (http fuzzer to semi-automatic penetration tests);
- sequencer (to analyze randomness of generated session identifiers);
- decoder (to convert between popular encryptions - e.g. Base64, URL, HEX) –
spider ("classic" http spider to automatically collect information on the resources used by a given application - subpage / style files / javascript files/ etc.)
- comparer (to compare the differences between HTTP requests or replies)

• SOAP UI,

• DirBuster - tool for testing web applications to detect folders and files normally inaccessible from the website navigation level.

- Nmap - open source tool to explore the network and security audits. It was designed to quickly scan big networks; it also operates well with individual addresses. Nmap uses low-level IP packages - to detect which addresses are accessible in the network, which ones provide access to services (application name and version), which operating systems they use (version system), what types of firewall systems are used and dozens of other features.
- SQLmap - automatic scanner of SQL Injection vulnerabilities
- Metasploit - tool used for penetration tests and for breaching ICT security systems
- Nessus Professional - tool to examine resources quickly, audit configuration, profile objectives, detect malicious software, sensitive data and many other. It scans operating systems, network devices, hyper supervisors, databases, web servers and sensitive infrastructure for vulnerabilities, threats and violations of compliance principles. With the world's biggest and constantly updated library of vulnerabilities and tests configuration as well as with the support of a team of experts for vulnerabilities this tool is a standard of speed and precision in scanning vulnerabilities.
- and our own programming framework

Their use results in more reliable scanning findings to compare and eliminate false alarms (false positive). Furthermore, the use of many tools also reduces the risk of missing a security gap by one of the programs.

2.2. Manual Tests

Major part of the security tester's work consisted in manual verification of the website and checking vulnerabilities. The automatic tools do not detect e.g. logical errors or implemented functionality of application. Carrying out attacks manually provides a possibility of a better evasion/analysis of security filters implemented in the application and firewall systems.

3. AREAS OF VERIFICATION DETAILS

V1. Architecture, design and threat modelling
V2. Authentication
V3. Session management
V4. Access control
V5. Malicious input handling
V6: Output encoding / escaping
V7. Cryptography at rest
V8. Error handling and logging
V9. Data protection
V10. Communications
V11. HTTP security configuration
V12: Security configuration verification requirements
V15. Business logic

V16. File and resources

V17. Mobile

V18. Web services

V19. Configuration

CONTACT DETAILS:

I remain at your disposal for all questions,

clarifications and look forward to your input,

Szymon Chruścicki

mobile: (+48) 505 372 810

email: szymon.chruscicki@testarmy.com