Protect • Comply • Thrive

[Client name]

Level 2 External Infrastructure Penetration Test Report (Redacted)

Release date: 9 April 2020

Version control

Version	Author	Stage
0.1	Report author	Draft issue
0.2	Report reviewer	Technical review
0.3	Report author	Author updates
0.4	Copy editor	Copy edit
0.5	Report author	Author updates
1.0	Testing manager	Approve for issue

Contact information

[Client name] details	
Primary contact	[Client name]
Primary contact's email	[Client email]
Relationship manager details	
Name	[Account manager's name]
Contact number	[Account manager's number]
Email address	[Account manager's email]
Consultant details	
Name	[Consultant's name]
Title	Penetration Tester

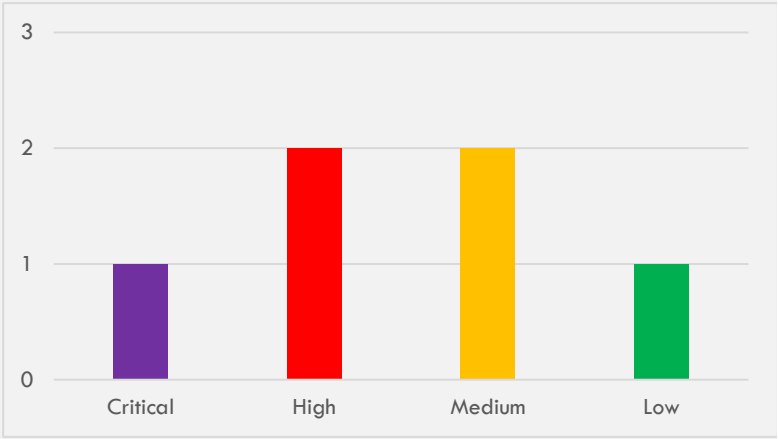
Table of contents

1. Executive summary

[Client name] has commissioned IT Governance Ltd (IT Governance) to perform a level 2 penetration test of its external infrastructure. A total of one critical, two high-, two medium- and one low-level vulnerabilities have been identified.

Vulnerability overview	Overall rating	Critical

Severity	No. of issues
Critical	1
High	2
Medium	2
Low	1



IT Governance considers vulnerabilities rated 'critical' or 'high' to be problems that require urgent remediation.

The vulnerabilities identified could allow an Internet-based attacker to:

- Access large amounts of sensitive and confidential data through a publicly accessible directory listing. The information consists of applicants' CVs, which contain personally identifiable information, along with usernames and hashed passwords that could be cracked offline and used to gain further access into [Client Name]'s network;
- Exploit known vulnerabilities affecting PHP, Apache and OpenSSL to compromise the confidentiality, integrity and availability of any information stored within the application or server; and
- Obtain information about software versions and the underlying infrastructure, helping the attacker gain an understanding of the systems and create a more sophisticated attack.

If these vulnerabilities were exploited by an attacker, [Client name] could be in breach of the Data Protection Act (DPA) 2018 and the General Data Protection Regulation (GDPR) as personal information is at risk. The resulting fines and/or data leaks could have legal, financial and reputational implications for [Client name].

All issues should be investigated and remediated, prioritising the critical- and high-risk vulnerabilities and then addressing all other vulnerabilities in a timely manner.

Further details on all vulnerabilities can be found in section 3 of this report.

2. Testing details

A level 2 penetration test is designed to help mitigate the threat from determined attackers or skilled, disgruntled employees. Automated scans are run alongside advanced manual testing techniques to identify and exploit vulnerabilities in the infrastructure.

IT Governance has scored the identified vulnerabilities against Common Vulnerability Scoring System (CVSS) version 3, which is summarised in Appendix A of this report. This classifies vulnerabilities as critical, high, medium or low.

The penetration test was performed using a proprietary security testing methodology, which is closely aligned with the SANS, Open Source Security Testing Methodology Manual (OSSTMM) and Open Web Application Security Project (OWASP) methodologies, as set out in Appendix B of this report.

Before testing, the test tools were updated to the latest versions and databases to make sure the latest known vulnerability signatures were being used. The test machines were also scanned to make sure they were free from malware.

We process the limited personal data given to us by client staff and representatives for the sole purpose of reporting on data gathered for this report. This data is not stored anywhere in our systems other than in this report and associated meeting records. It is retained until we are legally able to delete the records.

2.1 Scope

The penetration test was conducted from IT Governance's network ranges between [assessment date] and [assessment date] inclusive. The specific details of the scope of the testing, as provided by [Client name], are set out below.

Scope			
xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx
xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx
xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx

2.2 Objectives

The penetration test involves:

- Identifying vulnerabilities in the defined infrastructure;
- Attempting to exploit any identified vulnerabilities;
- Creating an ordered list of issues and their associated risk; and
- Providing remediation advice for identified vulnerabilities.

2.3 Testing limitations and restrictions

It should be noted that IT systems are subject to regular change. Consequently, the current security posture should be considered a 'snapshot' of specific aspects of security at a given point in time. New risks and vulnerabilities emerge constantly, and these, together with any changes made by [Client name], may result in increased levels of risk over time unless they are carefully managed.

3. Vulnerability findings

This section provides a technical summary and overview of the testing performed, the vulnerabilities identified and recommended remediation, as well as a summary detailing each specific issue.

3.1 Vulnerability summary

Vulnerability summary			
Ref	**Title**	**Severity**	**Status**
VUL-01	Directory listing	**Critical**	Open
VUL-02	Vulnerable version of Apache	**High**	Open
VUL-03	Vulnerable version of PHP	**High**	Open
VUL-04	Default PHP config file information disclosure	**Medium**	Open
VUL-05	Administrative interface publicly available	**Medium**	Open
VUL-06	Insecure TLS protocols supported	**Low**	Open

3.2 Consultant's commentary

A full network scan was performed, covering all TCP ports and the top 1,000 UDP ports across all hosts provided by [Client name]. The overall exposure of services across the external infrastructure was very limited, consisting entirely of HTTP and HTTPS over TCP ports 80 and 443 respectively.

A browsable directory listing was discovered on host xxx.xxx.xxx.xxx over port 80. This directory listing granted access to large amounts of sensitive and confidential data. Within the "/core/cache" directory, many files were titled CV_UserID_X, where X is a numeric value. These files contain large amounts of information about many applicants, including some personally identifiable information such as full name, residential address and contact information that, if leaked, could reveal a failure to properly secure the personal data, which is a breach of the GDPR. This issue was raised during testing so that [Client name] could start organising a remediation plan for the vulnerability.

In addition, two .sql files were discovered in the "/db" directory. These files contain records of past SQL commands, some of which contain sensitive data. One example was the command to write to the table "modx_users", which discloses usernames and hashed passwords of multiple user accounts along with their salt. This would allow an attacker to attempt to crack the passwords offline. If successful, the attacker could use these credentials to gain further access into [Client name]'s network. It is important to ensure that any directories listed by the server are intended to be publicly facing and do not contain sensitive information; otherwise, remove the directory listing.

The directory listing also contained "phpinfo" and "Security Information About PHP" files. Both of these disclose large amounts of information about the software installed on the underlying server and their associated weaknesses. These files revealed that PHP version 7.1.17 is installed. This version has multiple known security vulnerabilities that could be exploited by an attacker.

Additional vulnerable versions of software on multiple hosts were disclosed through their HTTP response headers. These headers identified Apache 2.4.33 and OpenSSL 1.1.0h. Default splash pages also revealed that IIS 8.5 was in use. While this version is not affected by any known vulnerabilities, an attacker can use the information to develop larger, more sophisticated attacks. Vulnerable software should be updated to the latest supported releases; review the patching policy to ensure that all software is kept up to date. Ensure that default splash pages are disabled and that HTTP response headers do not leak information regarding the software in use on the server.

Using various tools and manual verification, the consultant found that many of the servers hosting web services use a weak implementation of TLS protocols and cipher suites. These allow an attacker appropriately situated on the network to intercept and decrypt network traffic, allowing them to access sensitive information. It is important to ensure that best practice is followed, which dictates that only secure ciphers and protocols should be used to communicate sensitive information. The nature of the test meant it was not possible to exploit this issue, because a connection between a valid user and the server would have to be compromised. Configure the hosts to use TLSv1.1 or above, using strong cipher suites only.

3.3 Vulnerability details

This section provides a technical overview of the vulnerabilities identified and recommended remediation.

VUL-01	**Directory listing**		
Risk rating	**Critical**	**CVSS score**	**7.5**
CVSS v3 vector	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N		
Affected host			
xxx.xxx.xxx.xxx:80 (TCP)			
Description			

Web servers can be configured to automatically list the contents of directories that do not have an index page present. This can help an attacker to quickly identify the resources at a given path and proceed directly to analysing and attacking those resources. It particularly increases the exposure of sensitive files within the directory that are not intended to be accessible to users, such as temporary files and system logs.

Directory listings themselves do not necessarily constitute a security vulnerability. Any sensitive resources within the web root should in any case be properly access-controlled and should not be accessible by an unauthorised party who knows or guesses the URL. Even when directory listings are disabled, an attacker may guess the location of sensitive files using automated tools.

The images below show the root of the directory listing and an example of the files stored within:

[Image]

Due to the sensitivity of the information disclosed in this vulnerability, as described in section 3.2 of this report, the risk posed by this vulnerability has been upgraded to 'critical'.

Remediation advice

Disable the use of directory listings. There is not usually any good reason to provide directory listings and disabling them may place additional hurdles in the path of an attacker. This can normally be achieved in two ways:

1. Configure your web server to prevent directory listings for all paths beneath the web root.
2. Place into each directory a default file (such as index.htm) that the web server will display instead of returning a directory listing.

References

https://portswigger.net/kb/issues/00600100_directory-listing
https://www.siteground.com/kb/how_to_prevent_directory_listing/
https://www.owasp.org/index.php/Top_10-2017_A6-Security_Misconfiguration

https://cwe.mitre.org/data/definitions/548.html

VUL-02	**Vulnerable version of Apache**		
Risk rating	**High**	**CVSS score**	**7.8**
CVSS v3 vector	AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H		
Affected hosts			
xxx.xxx.xxx.xxx:80 (TCP) xxx.xxx.xxx.xxx:443 (TCP)			
Description			
The servers are running Apache version 2.4.33. This version of software has multiple known security vulnerabilities, which could be exploited by an attacker. A sample of weaknesses includes: • Code execution – with MPM (event, worker or prefork), code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. (CVE-2019-0211) • Denial of service (DoS) – by specially crafting HTTP/2 requests, workers would be allocated 60 seconds longer than necessary, leading to worker exhaustion and a DoS. (CVE-2018-1333)			
Remediation advice			
Update to the latest secure version of Apache and review the patching policy.			
References			
https://httpd.apache.org/download.cgi https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-252825/Apache-Http-Server-2.4.33.html https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities			

VUL-03	**Vulnerable version of PHP**		
Risk rating	**High**	**CVSS score**	**7.3**
CVSS v3 vector	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L		
Affected host			
xxx.xxx.xxx.xxx:80 (TCP)			
Description			

The server is running PHP version 7.1.17. This version of software has multiple known security vulnerabilities, which could be exploited by an attacker. A sample of weaknesses includes:

- Overflow – invalid input to the function xmlrpc_decode() can lead to an invalid memory access (heap out of bounds read or read after free). This is related to xml_elem_parse_buf in ext/xmlrpc/libxmlrpc/xml_element.c. (CVE-2019-9020)
- Overflow – a heap-based buffer over-read in PHAR reading functions in the PHAR extension may allow an attacker to read allocated or unallocated memory past the actual data when trying to parse the file name, a different vulnerability than CVE-2018-20783. This is related to phar_detect_phar_fname_ext in ext/phar/phar.c. (CVE-2019-9021)

Remediation advice

Update to the latest secure version of PHP and review the patching policy.

This branch version of PHP (7.1) is no longer actively supported, though it did receive security support until 1 December 2019. Consider migrating to the latest supported branches, versions 7.2 or 7.3.

References

https://www.cvedetails.com/vulnerability-list.php?vendor_id=74&product_id=128&version_id=257423
http://php.net/supported-versions.php
https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities

VUL-04	**Default PHP config file information disclosure**		
Risk rating	**Medium**	**CVSS score**	**5.3**
CVSS v3 vector	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N		
Affected host			
xxx.xxx.xxx.xxx:80 (TCP)			
Description			
The web application is configured to use default PHP config files. The "phpinfo" file displays sensitive information about the software and underlying host while the "Security Information About PHP" file displays the output of a security assessment highlighting potential vulnerabilities. An unauthenticated Internet-based attacker could use this information to perform further attacks against the web application host. This could lead to DoS attacks, remote code execution and privilege escalation. **Example screenshot of phpinfo:** [Image] **Security information about PHP:** [Image]			
Remediation advice			
Remove access to all default and config files to limit the amount of information publicly available.			
References			
https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A3-Sensitive_Data_Exposure *https://perishablepress.com/htaccess-secure-phpinfo-php/*			

VUL-05	Administrative interface publicly available		
Risk rating	**Medium**	**CVSS score**	**5.8**
CVSS v3 vector	AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:N/A:N/		

Affected hosts

xxx.xxx.xxx.xxx:443 (TCP)
xxx.xxx.xxx.xxx:443 (TCP)

Description

Administrative interfaces are publicly available. These interfaces control access to important services such as firewalls. The service is running on a standard port and is externally available to the Internet; an attacker could attempt to brute-force the credentials with the intention of gaining access. The following image shows an example of the web application interface.

[Image]

If an attacker gained access, they would be able to make network changes that would impact the integrity of the network and could put internal systems and data at risk.

Remediation advice

Review the need to have an administrative interface made publicly available. All interfaces that manage services pertaining to the firewall should be restricted to an authorised network.

Consider changing the firewall rules to restrict access to the administrative interface or contact the vendor documentation about disabling the interface.

All management interfaces should be configured with two-factor authentication enabled.

References

https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A5-Broken_Access_Control
https://www.ncsc.gov.uk/blog-post/protect-your-management-interfaces

VUL-06	**Insecure TLS protocols supported**		
Risk rating	**Low**	**CVSS score**	**3.1**
CVSS v3 vector	AV:A/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N/		

Affected hosts

xxx.xxx.xxx.xxx:443 (TCP)
xxx.xxx.xxx.xxx:443 (TCP)

Description

The server accepts connections using TLSv1.0 and TLSv1.1 encryption. These versions of TLS are affected by several cryptographic flaws, including Insecure session renegotiation and resumption schemes.

TLSv1.0 is heavily based on the vulnerable SSL protocol and has a number of flaws, such as a lack of support for various strong ciphers, enabling the downgrading of the connection to SSLv3.0 and being vulnerable to BEAST attacks, which would allow an attacker situated between a user's browser and the server to decrypt communication between the two parties. A successful exploit will lead to a loss in confidentiality of the data in transit.

TLSv1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLSv1.1.

An attacker appropriately situated on the network can exploit these flaws to conduct MITM attacks or decrypt communications between the affected service and clients. Although TLS has a secure means for choosing the highest-supported version of the protocol, many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as through the POODLE vulnerability).

Remediation advice

Disable the use of TLSv1.0. Only use TLSv1.1 or above with strong cipher suites.

References

https://www.nist.gov/oism/tls-10-being-turned-wwwnistgov
https://payment-services.ingenico.com/int/en/ogone/support/products/tls
https://support.microsoft.com/en-gb/help/187498/how-to-disable-pct-1-0-ssl-2-0-ssl-3-0-or-tls-1-0-in-internet-informat
https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html
https://alpacapowered.wordpress.com/2014/10/20/ssl-poodle-attack-what-is-this-scsv-thingy/
https://www.entrustdatacard.com/blog/2018/november/deprecating-tls

3.4 Supporting material

The following supporting material provides additional information relating to the associated finding.

3.4.1 Network information

Nmap

The supplied IP addresses were scanned using Nmap and Nessus; all TCP and common UDP ports were scanned. The results of the scans have been compiled, and the following table shows the number of identified ports and the results of OS fingerprinting.

Host	Protocol	Port	Service version
xxx.xxx.xxx.xxx	TCP	80	Firewall
xxx.xxx.xxx.xxx	TCP	443	Firewall
xxx.xxx.xxx.xxx	TCP	80	Apache 2.4.33
xxx.xxx.xxx.xxx	TCP	443	Apache 2.4.33
xxx.xxx.xxx.xxx	TCP	443	Microsoft-IIS/8.5

Appendix A – Vulnerability rating key

CVSS version 3 defines vulnerability bands according to the following definitions.

Vulnerability rating key		
Rating	**Description**	**CVSS band**
Critical	An attacker could use well-known methods and exploits to gain full control over the system or application, render it unusable by legitimate users, or access significant personal or sensitive information.	9.0 – 10.0
High	An attacker could gain full control over the system or application, render it unusable by legitimate users or access limited sensitive information.	7.0 – 8.9
Medium	An attacker could gain some level of interactive control or access to data held on the system.	4.0 – 6.9
Low	An attacker could gain information about the system or application that could be used to facilitate further access.	0.1 – 3.9
Info	Informational only; of limited use to an attacker.	0

Appendix B – External infrastructure testing methodology

IT Governance uses various tools and techniques to complete the agreed testing methodology. The high-level overview is outlined below with a brief description of what is assessed during each section.

Secure configurations – Reviewing open ports and their services to ensure that appropriate firewall configurations have been implemented. Assessing available services to ensure that they have gone through suitable hardening and that default configurations are not still in place.

Patching – Researching software versions to ensure that they are not affected by any publicly known vulnerabilities and that they are still under support by the vendor.

Secure authentication – Ensuring appropriate mechanisms are in place to confirm a user's identity. Understanding how the authentication process works and using that information to circumvent the authentication mechanism. Assessments include, but are not limited to, ensuring default credentials are not in use, and username enumeration.

Encryption – Assessing the implementation of encryption security around the transmission of communication. This includes checking for common weaknesses in SSL/TLS configurations and verifying that all sensitive data is being securely transferred.

Information leakage – Reviewing server configurations to ensure that information is not being leaked. This is assessed by reviewing configurations and examining how the server communicates to discover any information disclosure that could cause a security risk.