1on1 Secure

External Penetration Test Report

07/10/2023

example.com

Table of Contents

SYNOPSIS

1on1 Secure was recruited to evaluate **example.com**'s security by engaging in a 1-day penetration test that was conducted on July 10th, 2023. The goal of the "pentest" is to act as a threat-actor by performing cyber-attacks against **example.com.** This will serve to discover any present vulnerabilities that could result in a breach and be leveraged to access **example.com** sensitive data by a real-world attacker. All issues discovered by 1on1 Secure are achieved and verified through network evaluation, system vulnerability scanning and assessment, and both automated and manual exploitation (where applicable) of found vulnerabilities.

FINDINGS OVERVIEW

While conducting the external penetration test, there were several highly critical vulnerabilities discovered on the **example.com** web server. 1on1 Secure was able to obtain a large amount of sensitive data that was not adequately secured by following simple best practices, such as making sensitive files not "world readable". It may have been possible to gain remote system access, then full administrative control. This was only briefly attempted as the budget for this report was insufficient for exhaustive testing.

There was also an opportunity to brute-force credentials, which may have led to further vulnerabilities. This was not attempted due to a limited time frame and budget.

RECOMMENDATIONS:

To increase the security posture of **example.com**, 1on1 Secure recommends the following mitigations and/or remediations be performed:

1. Upgrade all software to the latest versions.
2. Make sensitive files not readable to the public.
3. Implement a reverse proxy service such as cloudflare (a free service).
4. Implement strong password policy including: upper/lower case + special characters + numbers + minimum of 8 characters.
5. Limit login attempts at all login locations.
6. Enable the Plesk extension to automatically upgrade all Wordpress plugins for all websites hosted. https://www.plesk.com/extensions/wp-toolkit-smart-updates-2/

SEVERITY SCALE:

CRITICAL Severity Issue: Poses immediate danger to systems, network, and/or data security and should be addressed as soon as possible. Exploitation requires little to no special knowledge of the target. Exploitation doesn't require highly advanced skill, training, or tools.

HIGH Severity Issue: Poses significant danger to systems, network, and/or data security. Exploitation commonly requires some advanced knowledge, training, skill, and/or tools. Issue(s) should be addressed promptly.

MEDIUM Severity Issue: Vulnerabilities should be addressed in a timely manner. Exploitation is usually more difficult to achieve and requires special knowledge or access. Exploitation may also require social engineering as well as special conditions.

LOW Severity Issue: Danger of exploitation is unlikely as vulnerabilities offer little to no opportunity to compromise system, network, and/or data security. Can be handled as time permits.

INFORMATIONAL Issue: Meant to increase client's knowledge. Likely no actual threat.

METHODOLOGY

1on1 Secure penetration testers employed testing methods that are widely adopted in the cyber security assessment industry. This includes 5 phases: **Information Gathering**, **Enumeration**, **Vulnerability Assessment**, **Exploitation**, and **Reporting/Mitigation**.

During these phases, both automated and manual audit techniques were used to ensure the best possible results.

INFORMATION GATHERING

1on1 Secure was given a scope of host(s) from **example.com** that includes the **example.com** web server. You can see the network details of that device listed below:

- Hostname: **example.com**
- IP Address: **192.168.0.2**

1on1 Secure testers were able to verify the IP address and connectivity of the host/server by connecting to the Internet and performing a ping of **example.com** which returned with IP: **192.168.0.2**

HIGH Severity Issue: Discovery of the server IP could have been prevented by employing a reverse proxy service such as cloudflare (a free service). This single implementation **would have prevented the discovery of the following vulnerabilities on this and the following 2 pages**.

A reverse proxy provides a number of other benefits:

- CDN reduces load times around the world
- Hides true IP address of web server
- Caches images and common files to reduce load and bandwidth cost for the web server
- Provides DDOS protection
- Provides WAF blocking
- Provides additional HTTP Headers such as visitors Country which can be used to block visitors from foreign countries

A port scan for ip address **192.168.0.104** revealed the following ports:
port 21 is open for ftp vsftpd 2.0.8 or later / current ver = 3.0.5
port 80 and 443 are open

CRITICAL Severity Issue: Exposure of the web servers true ip, allowed for a scan to reveal ALL websites hosted on the server:

Other sites:

1 **—.com**
2 **—.club**
3 **—.net**
4 **—.com**
5 **—.com**
Mail servers: mail.---.net (used by 31 domains)
6 **example.com**
7 **—-.com**
8 **—.com** **31 Vulnerabilities allowing server control**
9 **—.com**

10 **—.com**
11 **—.com**
12 **—.com**
13 **—.org**
14 **—.com** **Vulnerable**
Mail servers: mail.---.net (used by 31 domains)

| [!] Title: Contact Form 7 <= 5.0.3 - register_post_type() Privilege Escalation
| Fixed in: 5.0.4
| References:
| - https://wpscan.com/vulnerability/af945f64-9ce2-485c-bf36-c2ff59dc10d5
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20979
| - https://contactform7.com/2018/09/04/contact-form-7-504/
| - https://plugins.trac.wordpress.org/changeset/1935726/contact-form-7
| - https://plugins.trac.wordpress.org/changeset/1934594/contact-form-7
| - https://plugins.trac.wordpress.org/changeset/1934343/contact-form-7
| - https://plugins.trac.wordpress.org/changeset/1934327/contact-form-7
| - https://www.ripstech.com/php-security-calendar-2018/#day-18
|
| [!] Title: Contact Form 7 < 5.3.2 - Unrestricted File Upload
| Fixed in: 5.3.2
| References:
| - https://wpscan.com/vulnerability/7391118e-eef5-4ff8-a8ea-f6b65f442c63
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-35489
| - https://www.getastra.com/blog/911/plugin-exploit/contact-form-7-unrestricted-file-upload-vulnerability/
| - https://www.jinsonvarghese.com/unrestricted-file-upload-in-contact-form-7/
| - https://contactform7.com/2020/12/17/contact-form-7-532/#more-38314

15 **—.org**
16 **—.com**
17 **—.com**
18 **—.com**
19 **—.com**
20 **—.com**
21 **—.net**

NOTE: It is assumed that each of the above listed websites each have their own unique vulnerabilities. Not all URLs listed above were scanned, but a random sampling was performed and resulted in enough **CRITICAL** vulnerabilities to potentially **gain complete control of the web server.**

The scope of this assignment prevented us from attempting penetration of the server THRU other URLs, though we are highly confident this method would have been successful to **gain complete control of the web server.**

RECOMMENDATIONS:

1. Upgrade ALL websites hosted to the latest version of PHP
2. Upgrade ALL websites hosted to the latest version of Wordpress
3. Enable the Plesk extension to automatically upgrade all Wordpress plugins for all websites hosted. https://www.plesk.com/extensions/wp-toolkit-smart-updates-2/
4. Employ a reverse proxy service such as cloudflare (a free service), to hide the web server's true IP address and prevent scanning for all websites hosted on the server. The webserver should be configured to refuse all traffic that does NOT come from the reverse proxy service.

The above scan, revealed an unusual mail server with IP address: **192.168.0.113**
This was interesting as the IP address for integtech.net was found to be: **192.168.0.104**
It was reasonable to assume that these IPs were purchased in a block, and a scan was run from: **192.168.0.104 - 192.168.0.113**

The following was discovered:

```
192.168.0.107
80/tcp   open   http               nginx
443/tcp  open   ssl/http           Microsoft IIS httpd 8.5
500/tcp  closed isakmp
3389/tcp open   ssl/ms-wbt-server?

3389/tcp open  ms-wbt-server
| rdp-enum-encryption:
|   Security layer
|     CredSSP (NLA): SUCCESS
|     CredSSP with Early User Auth: SUCCESS
|     Native RDP: SUCCESS
|     RDSTLS: SUCCESS
|     SSL: SUCCESS
|   RDP Encryption level: Client Compatible
|     40-bit RC4: SUCCESS
|     56-bit RC4: SUCCESS
|     128-bit RC4: SUCCESS
|     FIPS 140-1: SUCCESS
|_  RDP Protocol Version:  RDP 5.x, 6.x, 7.x, or 8.x server
| rdp-ntlm-info:
|   Target_Name: EXAMPLE
|   NetBIOS_Domain_Name: EXAMPLE
|   NetBIOS_Computer_Name: ARGUS
|   DNS_Domain_Name: EXAMPLE.net
|   DNS_Computer_Name: Argus.EXAMPLE.net
|   DNS_Tree_Name: EXAMPLE.net
|   Product_Version: 6.3.9600
|_  System_Time: 2023-07-12T04:45:39+00:00

192.168.0.113
Host is up (0.046s latency).
Not shown: 990 filtered tcp ports (no-response)
PORT     STATE SERVICE  VERSION
80/tcp   open  http     Microsoft IIS httpd 10.0
110/tcp  open  ssl/pop3 MailEnable POP3 Server
143/tcp  open  imap     MailEnable imapd
443/tcp  open  ssl/http Microsoft IIS httpd 10.0
465/tcp  open  ssl/smtp MailEnable smptd 10.43--10.43
587/tcp  open  ssl/smtp MailEnable smptd 10.43--10.43
993/tcp  open  ssl/imap
995/tcp  open  ssl/pop3 MailEnable POP3 Server
8080/tcp open  http     MailEnable httpd 5.0
8081/tcp open  ssl/http MailEnable httpd 5.0
```

NOTE: These items were out of scope for this project, however an attacker is not bound by any scope and there are several open ports of interest here, including RDP.

DOMAIN-WIDE ISSUES:

LOW Severity Issue: GET /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options:

> This makes it possible for an attacker to place your website inside a nefarious website that can perform “Clickjacking”, Phishing, or Frame Injection

NOTE: Normally this would be classified as a **HIGH Severity Issue** but Due to the limited functionality of the website and the fact that no sensitive user data is present, the severity is currently **LOW** This is subject to change if the website functionality is expanded in the future.

RECOMMENDATION: Configure the anti-clickjacking X-Frame-Options HTTP header correctly.

LOW Severity Issue: GET /: The Content-Encoding header is set to "deflate" which may mean that the server is vulnerable to the BREACH attack. See: http://breachattack.com/:

> This makes it possible for an attacker to gain access to sensitive information.

NOTE: Exploitation was not attempted.

RECOMMENDATION: See: http://breachattack.com for information on how to properly configure the webserver.

LOW Severity Issue: GET /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/

RECOMMENDATION: Configure the X-Content-Type-Options header correctly.

HIGH Severity Issue: HEAD PHP/8.0.15 appears to be outdated (current is at least 8.2).
This PHP version is now 3 years old and has a large number of known vulnerabilities.

RECOMMENDATION: Upgrade to the latest version of PHP.

SPECIFIC ISSUES:

MEDIUM **Severity Issue**: XML-RPC seems to be enabled: https://example.com/xmlrpc.php
This is a rarely used feature of Wordpress that allows successive brute forcing attempts with no limit to the number of incorrect logins.

RECOMMENDATION: If XML-RPC is not used, it should be disabled.

MEDIUM **Severity Issue**: Debug Log found: https://example.com/wp-content/debug.log
This file can expose failures in the PHP code running on the system and can possibly provide an attack vector to an attacker.

RECOMMENDATION: This file should not be publicly readable.

MEDIUM **Severity Issue**: Vulnerable and no longer supported WP Plugin found: digiproveblog
| Location: https://example.com/wp-content/plugins/digiproveblog/
| Latest Version: 4.16 (up to date)
| Last Updated: 2022-01-14
|
| [!] 1 vulnerability identified:
|
| [!] Title: Copyright Proof <= 4.16 - Reflected Cross-Site-Scripting
| References:
| - https://wpscan.com/vulnerability/af4f459e-e60b-4384-aad9-0dc18aa3b338
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-1906

NOTE: Attempts to exploit this vulnerability failed.

RECOMMENDATION: This plugin should be replaced with a new plugin that is actively maintained.

MEDIUM **Severity Issue**: Sensitive server data leaked here: https://example.com/phpinfo.php

NOTE: This page lists an extensive amount of data about the server and PHP environment and can provide significant information to an attacker.

RECOMMENDATION: This file should not be publicly readable.

MEDIUM **Severity Issue**: Sensitive server data leaked here: https://example.com/info.php

NOTE: This page lists an extensive amount of data about the server and PHP environment and can provide significant information to an attacker.

RECOMMENDATION: This file should not be publicly readable.

MEDIUM **Severity Issue**: Cookie created without the httponly flag here:
https://example.com/wp-login.php?action=register

This can allow an attacker to gain access to the user's authentication cookie and the attacker can then login as that user.

NOTE: This page lists an extensive amount of data about the server and PHP environment and can provide significant information to an attacker.

RECOMMENDATION: All cookies should use the “httponly flag” to prevent access to the cookie from javascript.

INFORMATIONAL Issue: plesk hosting software present on the web server

INFORMATIONAL Issue: imunify360 is installed to secure the server and harden PHP

NOTE: This is likely why attempts at compromising the server were unsuccessful, despite vulnerabilities being present on the server.

RECOMMENDATION: Continue using immunify360, but also remove all vulnerabilities.

INFORMATIONAL Issue:
User(s) Identified:

- Alex Thornberger - alext2022
- Jadie Russo - jadie1018

HOUSE CLEANING

During a penetration testing engagement, tools, files, user accounts, etc., are created on the client’s system(s) which would compromise the client’s security.

1on1 Secure is diligent to ensure that no potential security issues are introduced to **example.com**’s environment through remnants left on their system(s) after the completion of the engagement. **example.com** had all tools, files, user accounts, etc. that were created by 1on1 Secure testers during the engagement removed.