GUIDEPOINT
SECURITY

ContractPod Technologies Inc.

Application Security Assessment

Copilot

Letter of Attestation
January 9, 2024
Version 1.0

CONFIDENTIAL

Table of Contents

Project Contacts and Document History

ContractPod Contacts
Primary
Michael Weaver
Director of Information Security
+44 2070961401
michael.weaver@contractpodai.com

GuidePoint Security Contacts	
Primary	**Secondary**
Justin Tharpe	J. David Bressler
Senior Application Security Consultant	Practice Director, Application Security
(877) 889-0132 x8133	(877) 889-0132 x7432
justin.tharpe@guidepointsecurity.com	david.bressler@guidepointsecurity.com

Report Version History

Version	Date	Author	Comments
1.0	January 9, 2024	Justin Tharpe	Letter of Attestation

Disclaimer

This document contains and constitutes the proprietary and confidential information of GuidePoint Security, LLC, ("GuidePoint"). It is provided to ContractPod Technologies Inc. ("ContractPod") subject to and in accordance with the terms of any agreement between GuidePoint and ContractPod regarding treatment of confidential information and/or licensing of proprietary information. This document also contains information that is the highly sensitive confidential information of ContractPod and should be treated by representatives of ContractPod accordingly. The recipient, without the expressed permission of GuidePoint and ContractPod, may not distribute this document.

The contents of this document do not constitute legal advice. GuidePoint's offers of services or deliverables that relate to compliance, litigation, or other legal interests are not intended as legal counsel and should not be taken as such.

Independent Security Assessment Report

To Whom It May Concern:

GuidePoint Security LLC (“GuidePoint”) has performed the Application Security Assessment for ContractPod Technologies Inc. (“Client”) while acting as an independent security assessor. This assessment was performed with the intent of evaluating the scope, security, and resiliency of Client’s Copilot application environments.

The methodology utilized during this assessment is detailed in Appendix B - Methodology. GuidePoint developed this methodology based on extensive professional experience and information system security assessment best practices gathered from the Open Source Security Testing Methodology Manual (“OSSTMM”), the National Institute of Standards and Technology (“NIST”) Special Publication 800-115: Technical Guide to Information Security Testing and Assessment, the Penetration Testing Execution Standard (“PTES”), and the Open Web Application Security Project (“OWASP”) Testing Guide v4.0.

While this type of assessment is intended to mimic a real-world attack scenario, GuidePoint is bound by rules-of-engagement, defined scope, allocated time, and additional related constraints. GuidePoint has made every effort to perform a thorough and comprehensive analysis and to provide appropriate remedial advice. However, inherit limitations, errors, misrepresentations, and changes to the Client environment may have prevented GuidePoint from identifying every security issue that was present in the Client environment at the time of testing. Therefore, the findings included in this report should be considered to be representative of what a similarly skilled attacker could achieve with comparable resources, constraints, and time frame.

Additionally, it is worth emphasizing that the findings and remediation recommendations are the result of a point-in-time assessment based on the state of the Client environment as of November 6, 2023. GuidePoint therefore does not provide any assurance related to configuration or control modifications in the Client environment, changes in regulatory or compliance requirements, discoveries of new vulnerabilities and attack techniques, or any other future event that may impact the Client’s security posture.

The information contained in this report represents a fair and unbiased assessment of the Client’s environment based on the agreed upon criteria as defined in the Statement of Work. This report is provided to the Client as notification of outstanding security risks that threaten the confidentiality, integrity, and availability of sensitive information, as well as to provide assistance and direction with remediation. The evidence and references provided for each finding serve as the basis for our qualified opinions in this report.

GuidePoint has provided this report solely for private and internal use by the Client, and it may not be shared or redistributed without GuidePoint’s express written consent. GuidePoint’s assessments focus exclusively on information security and the conclusions arrived at in this report should not be considered to be a representation or endorsement of the Client’s products or services.

Bryan Orme
Principal
GuidePoint Security, LLC

Executive Summary

BACKGROUND

ContractPod Technologies Inc. (“ContractPod”) engaged GuidePoint Security (“GuidePoint”) to perform an Application Security Assessment of ContractPod’s Copilot application. An Application Security Assessment is comprised of an assessment of the dynamic (runtime) functionality of an application. This assessment uses both automated and manual security testing techniques in order to identify weakness in the application from an attacker’s perspective.

APPROACH

GuidePoint performed assessment activities against ContractPod’s Copilot application while hosted in the QA environment. As part of the Application Security Assessment, GuidePoint utilized multiple testing techniques to evaluate the different components of the ContractPod Copilot application and its supporting infrastructure.

GuidePoint performed testing from both unauthenticated (anonymous) and authenticated perspectives. Unauthenticated testing identifies vulnerabilities and weaknesses available to anyone that possesses network connectivity to the Copilot environment. Authenticated testing identifies vulnerabilities and weaknesses in functionality that is only available to valid, authenticated users. Since most applications commonly limit anonymous access and provide the majority of their functionality to authenticated users, authenticated testing often provides the best insight into the security posture of the application.

ContractPod provided GuidePoint with credentials for the following roles in the Copilot application:

- Standard User

Testing application functionality from multiple authenticated perspectives increases coverage and the likelihood of identifying horizontal escalation (accessing functionality or information belonging to similarly or identically configured user accounts) and vertical escalation (accessing more privileged functionality or information than the testing account has been granted).

GuidePoint conducted the Application Security Assessment between October 30 and November 6, 2023. All testing activities were performed remotely from GuidePoint’s security testing lab.

LIMITATIONS

Note: Denial-of-Service (DoS) testing was not performed during this engagement.

The assessment was a time-boxed engagement and prioritizes the most security sensitive functions and features, but a comprehensive review of the entire application was not possible within these time limits.

REMEDIATION VERIFICATION

Remediation verification was performed on December 1, 2023 and January 2, 2024. Remediation verification testing was performed for the following vulnerabilities in ContractPod’s Copilot application:

- Failure to Invalidate Sessions
- Valid User Account Enumeration
- HTML Markup Returned in JSON Response
- Last Login History Not Displayed
- Detailed Error Messages
- Response Contains Platform Information

Results of the remediation verification are provided for each finding in the Technical Analysis section.

FINDINGS SUMMARY

Through the use of automated and manual techniques, GuidePoint identified a total of **8** findings within ContractPod's Copilot application and supporting environment. These weaknesses threaten the confidentiality, integrity, and availability of the application, the environment, and the data contained within it. The following table summarizes the quantity and severity of the findings identified during this assessment:

	Finding Severity Summary			
	High	**Medium**	**Low**	**Total**
Findings Identified	0	2	6	8
Findings After Remediation Verification	0	0	4	4
Findings After 2nd Remediation Verification	0	0	0	0

Table 1: Finding Severity Summary

RISK ANALYSIS

The following table summarizes the severity and potential business impact associated with the identified findings. Refer to Appendix C for Severity Definitions.

ID	Severity	Finding Title	Business Impact
M1	~~Medium~~ Remediated	Failure to Invalidate Sessions	Session guessing or replay attacks may be more successful, because sessions are not terminated at logout.
M2	~~Medium~~ Remediated	Valid User Account Enumeration	Attackers can discover valid system accounts.
L1	~~Low~~ Remediated	HTML Markup Returned in JSON Response	Attacker controlled script execution is possible but is dependent on an exploitable web browser vulnerability.
L2	~~Low~~ Remediated	Last Login History Not Displayed	A user may not be able to detect malicious usage of their account.
L3	~~Low~~ Remediated	Detailed Error Messages	Information leakage can lead to more effective and targeted attacks.
L4	~~Low~~ Remediated	Response Contains Platform Information	Detailed platform information could allow for more accurate attacks.
I1	Informational	Weak SSL/TLS Server Configuration	The secured network communications can be attacked due to server configuration weaknesses.

Table 2: Finding Summary List

SUMMARY OPINION

As a result of conducting this engagement and remediation verification, GuidePoint has determined that cumulatively the vulnerabilities identified pose a **Minimal** risk to ContractPod. This evaluation was determined by assessing the severity and number of vulnerabilities identified throughout the application as well as GuidePoint's experience in assessing similar applications.

Appendix A: Scope Overview

GuidePoint evaluated the security of the Copilot application as defined by ContractPod. The following table details the application URL, version, and provided accounts that the assessor used for each application.

URL	Version	User Name	Role
pentest-qa.leahcopilot.com	Unknown	justin.tharpe@guidepointsecurity.com JKT.TestAcct@gmail.com	Standard User Standard User

Table 3: Applications Tested

The following table lists the supporting infrastructure components that GuidePoint analyzed as part of this Application Security Assessment. These components may include items related to hosting the application, authentication, or remote access.

Host Name	Description
pentest-qa.leahcopilot.com	QA
api-qa.leahcopilot.com	QA

Table 4: Application Supporting Infrastructure

All other applications and supporting components were outside of the scope of this assessment.

Appendix B: Methodology

APPLICATION SECURITY ASSESSMENT

An Application Security Assessment (ASA) is focused on the running application and its environment. The objective of the assessment is to identify vulnerabilities in the application and use manual testing techniques to verify their existence. These assessments are most successful when clients share all known information with the consultant; however, the client can elect to share less information.

The ASA is a comprehensive assessment that identifies vulnerabilities ranging from High to Low severity. GuidePoint's Application Security Team identifies, verifies, and reports anything that raises the attack surface of the application. We use multiple techniques to simulate attacks from an unauthenticated and, when applicable, authenticated perspective, exposing the greatest amount of attack surface and providing the most value from the testing efforts.

GuidePoint follows a highly-structured methodology to ensure a thorough test of the application and its environment is conducted. Our methodology uses a phased approach, consisting of information gathering, testing, verification, and notification. GuidePoint employs a comprehensive and careful methodology in order to identify any potentially dangerous functionality. Prior to performing tests against these functions, GuidePoint shares any potential impacts with the client. These steps ensure the least amount of business impact possible.

In cases where automated testing cannot be performed safely, multiple manual testing techniques may be used to confirm the existence of a vulnerability. If a deep level of exploitation of a database, application server, or host platform weakness is necessary to gather evidence of a vulnerability, GuidePoint's consultant will contact the engagement POC, as well as any applicable customer personnel. The GuidePoint Team will discuss a plan of attack as well as any potential concerns, and then will seek explicit approval from the client in order to proceed with the exploitation of any vulnerabilities that have the potential to impact production operations.

The GuidePoint Team will communicate all verified vulnerabilities identified throughout the engagement that present significant danger to the client's organization. This will allow the client to begin planning remediation activities sooner, potentially closing the window on further exploitation by an attacker prior to the delivery of the final report.

GuidePoint follows industry best practice methodologies when performing application security testing activities. Such methodologies include:

- Open Source Security Testing Methodology Manual (OSSTMM)
- Open Web Application Security Project (OWASP) Testing Guide
- The National Institute of Standards and Technology (NIST) SP 800-115

Information Gathering

At the beginning of the assessment, the GuidePoint Team reviews documentation and meets with client representatives to gather background information on the applications we will test. Typically, this takes place during the Kickoff Meeting between GuidePoint and designated client contacts.

The background information we gather includes items such as URLs, account credentials, testing timelines, and any additional technical information shared at the outset of the engagement. The client also can provide this information after the Kickoff Meeting via additional phone calls or communications, as necessary.

After the client provides the initial information, the GuidePoint Team will test the credentials and make notes of their privilege levels, any technology in use, and anything that could cause issues with automated testing.

Testing

GuidePoint conducts testing using both automated and manual testing methods. GuidePoint leverages multiple automated testing techniques to ensure full coverage, including commercial, open source, and custom-built tools.

Some examples of tools that GuidePoint uses include:

- Burp Suite Pro w/Various add-ons
- Netsparker
- Hopper, x64dbg, ILSpy, JD-GUI, Wireshark
- ProcessHacker, Sysinternals Suite
- SQLmap

The GuidePoint Team configures the automated tools using information obtained during the information-gathering phase. This ensures the highest level of success, by removing obstacles that typically hinder their functionality and ability to navigate the application properly.

Manual testing typically comprises approximately 80% of the overall level of effort of the assessment. During this portion of the testing, the GuidePoint consultant scours the application, analyzing the communication, functions, and data the application sends and receives. The GuidePoint Team tests complex interactions, workflows, and business logic. Additionally, they manually evaluate areas of the application and specific vulnerabilities that automated tools either have difficulty with or are unable to identify and analyze properly.

GuidePoint Security’s ASA includes, but is not limited to, identification of the following risks:

Application Profiling and Information Disclosure	Platform and Third-Party Misconfiguration	Cookie and Session Handling
• Default Banners • Unhandled Error Conditions • HTML/JavaScript Comment Information Leakage • Extraneous Content in Web Root • Source Code Disclosure • Robots.txt Path Disclosure • Content Expiration and Cache Control • Bit Bug/Referer Header Leakage • Account Enumeration • Backup/Archive Content	• Default Administrative Credentials • Default Content and Scripts • Application Script Engine • Web Server • Weak SSL Implementation • Flawed Use of Cryptography	• Session Fixation/Hijacking • Set-Cookie Weaknesses • Sensitive Information Disclosure • Cookie Poisoning • Multiple Simultaneous Login Allowed • Session Timeout • Explicit/Implicit Logout Failures • Cookieless Sessions • Custom Session Management
Command Injection Flaws	**Logic Flaws**	**Client-Side Flaws**
• SQL Injection • XXE, XPath, and XML Injection • SSI/OS Command Injection • Server Script Injection/Upload • Cross-Site Scripting (XSS) • Buffer Overflow	• Privilege Escalation • Sensitive Information Disclosure • Data Mining/Inference • Functional Bugs • Application-Specific Control Failures • Cross-Site Tracing (XST) • Weak Data Validation • Race Conditions • CPU-Intensive Functions	• Exposure of Sensitive Business • Logic • Reliance on Client-Side Validation • AJAX/Web Service Flaws • Java Applet/ActiveX • Control/Flash Weaknesses
Authentication and Authorization		
• Unauthenticated Sensitive Content • Poor Separation of Privilege • Brute-Force Login • Weak Password Policy • Account Lockout/Denial of Service • SSO Weaknesses • Security Question Weaknesses • CAPTCHA Flaws		

Validation

The GuidePoint Team will validate all vulnerabilities identified throughout the course of the assessment. This starts at the outset, with the validation of vulnerabilities identified by automated tools, and continues throughout the course of the assessment. The GuidePoint Team employs multiple techniques and manual methods to eliminate false positives. We use care during testing and validation activities to avoid stability issues with the application and its environment.

Appendix C: Finding Severity Definitions

GuidePoint's Severity levels are determined by the evaluation of multiple factors surrounding the vulnerability and the environment the vulnerability was identified in. The rating of a vulnerability can change based on these factors. In general, the factors that make up the risk of an identified vulnerability include:

- Likelihood of vulnerability being identified
- Public availability of technical details
- Attack tools
- Requisite attacker skill
- Potential and likelihood for successful exploitation
- Attack surface and users affected

Severity levels as well as specific factors documented with levels in the finding are further defined in the tables below. While these definitions encompass common scenarios, a vulnerability's Severity level may be adjusted based on the specific circumstances in which it was encountered.

Severity	Defining Characteristics
High	**Vulnerabilities that may have an immediate impact and could result in arbitrary code execution, data loss, user compromise, or affect system availability.** • A systems or user compromise is likely based on successful exploitation • Technical vulnerability details and/or exploit code are publicly available • Controls to prevent exploitation do not exist or are ineffective • If additional attack vectors are required to perform exploitation, they are trivial to leverage • May result in significant and costly loss of major tangible assets or resources • May significantly violate, harm, or impede the organization's mission, reputation, or interest • A strong need for corrective measures exists
Medium	**Vulnerabilities that have a low probability of arbitrary code execution or data loss, but a high potential to affect individual users or system availability.** • Exploitation does not result in system compromise • Controls are in place that may impede successful exploitation of the vulnerability • An additional attack vector may be required to exploit this vulnerability • May result in costly loss of tangible assets or resources • May violate, harm, or impede the organization's mission, reputation, or interest • Corrective actions should be performed within a reasonable period of time
Low	**Vulnerabilities that could provide excessive information or increase the attack surface, but do not result in code execution, direct user compromise, data loss, or affect system availability.** • Exploitation is difficult or only results in minor information disclosure • Controls implemented prevent, or significantly impede, the vulnerability from being exploited • The conditions under which this vulnerability can be exploited are possible, but less likely • May result in the loss of some tangible assets or resources • May noticeably affect the organization's mission, reputation, or interest • Correct actions are recommended, but this determination is dependent upon the system owner

Table 5: Vulnerability Severity Definitions

Rating	Defining Characteristics
Easy	**Vulnerabilities in this category are trivial to identify in an application or environment.** • Publicly available tools • Skill not needed by attacker • Discovery methods are well known • Controls not in place to prevent discovery
Moderate	**Vulnerabilities in this category require effort by an attacker but are still easy to identify.** • Information on vulnerability is publicly available • May require modification of public tools or frameworks • May require modification of request that is outside the norm • Research on particular conditions may be necessary prior to discovery
Difficult	**Vulnerabilities in this category are hard to identify and require a skilled attacker.** • Skilled attacker necessary • Limited information available • May require multiple vulnerabilities to exist as a prerequisite • Tools may not be available publicly • Custom tools may need to be written in order to identify vulnerability • Control bypass may be necessary in order to conduct attacks

Table 6: Ease of Exploit Definitions

Rating	Defining Characteristics
High	**Vulnerabilities in this category are widely known and common targets for attackers and provide high value for their effort.** • Information publicly available • Exploitation provides a high value for an attacker • Vulnerabilities are well known • Tools are easily accessible for exploitation • Techniques are easily known for exploitation • Skilled attacker not necessary • No protection mechanisms in place
Medium	**Vulnerabilities in this category are known and provide a good balance of effort for attack value.** • Exploitation requires a knowledgeable attacker • Tools can be modified to provide successful exploitation • Vulnerability information is known • Protection mechanisms in place can be bypassed
Low	**Vulnerabilities in this category are not well known and provide a significant amount of effort and may yield less value.** • Skilled attacker required to provide value from exploitation of the vulnerability • Protection mechanisms are in place that stifle exploitation • Tools may need to be constructed to exploit vulnerability • Manual attack techniques may be necessary • Protection mechanisms in place difficult to bypass

Table 7: Probability of Attack Definitions

GUIDEPOINT

SECURITY