altus cloud

Cybersecurity Penetration Assessment Report

Grey-Box Findings

An Independent Evaluation and Attack Narrative by Allendevaux & Company

Highly Confidential

Q2 2023

Security | Privacy | Complex Compliance

HIGHLY CONFIDENTIAL

Compiled for	Compiled by
altus cloud	ALLENDEVAUX & COMPANY

This report is intended solely for use by the organisation and is not intended to be consumed by anyone other than those specified parties.

Technical Penetration Assessment Report

altus cloud

GREY-BOX PENTESTING | Q1 2023

ALLENDEVAUX & COMPANY
United States of America | United Kingdom

ABSTRACT

This document summarizes the results of an independent cybersecurity pentest assessment conducted for the organization by ALLENDEVAUX & COMPANY. The purpose of this report is to measure the security posture of the organization's technical environment by (a) discovering vulnerabilities, (b) attempting to exploit vulnerabilities, (c) reporting the results through an attack narrative, and (d) positing recommendations for improvement. When this process is employed regularly, it fosters continued improvement in the hardening of the organization's technical infrastructure.

CAUTION: A vulnerability report and penetration test contains highly confidential information. It may identify ways a service or system may be exploited in order for engineers to improve upon the service. If this document falls into the wrong hands, it could pose a significant threat. For that reason, this document should only be used within the organisation; its distribution outside the organisation should only be done under strict nondisclosure agreement.

Document Revision	1.0
Date of Issue	25 April 2023
Testing Window	17 April 2023 – 23 April 2023
Document Authors	Dr. Scott Allendevaux, CISSP, CIPT, CIPM, CIPP/US, HCISPP Jonny Leage, CREST CPSA, Certified Ethical Hacker (CEH)
Document Contributors	Rebekah Allendevaux, CIPP/US, CIPM, CIS LI, CIS LA Clayton Horstman, OSCP, CREST CRT, CREST CSPA Koushick Prasad, Certified Ethical Hacker (CEH) Mayank Garg, Certified Ethical hacker (CEH)

ALLENDEVAUX & COMPANY
United States of America | United Kingdom

TABLE OF CONTENTS

1 EXECUTIVE SUMMARY 8

1.1 Assessment Objective 8
1.2 Responsibility of Management 8
1.3 Responsibility of the Assessor 8
1.4 Not the Responsibility of the Assessor 9
1.5 Assessment Findings 9
1.6 Recommendations 10
1.6.1 Remove Command Injection Vulnerability (CRITICAL) 10
1.6.2 Review User Input Character Bypass (HIGH) 10
1.6.3 Incorporate TLS for the Application (HIGH) 11
1.6.4 Update or Remove Vulnerability JavaScript Libraries (MED-HIGH) 11
1.6.5 Remove Sensitive Information Exposure (MED-HIGH) 11
1.6.6 Prevent Clear Text Displays of Data in Transport (MED-HIGH) 12
1.7 Conclusion of Executive Summary 12
1.7.1 Confidentiality 12
1.7.2 Integrity 12
1.7.3 Availability 12

2 SCOPE AND METHODOLOGY OF PENTEST 14

2.1 Testing Perspective 14
2.1.1 Black-box 14
2.1.2 Grey-box 14
2.1.3 White-box 14
2.2 Target Methodology 14
2.3 Target Inventory 16
2.4 Tools Used 16
2.5 Engagement Restrictions 18

3 SUMMARY OF PENTRATION TEST FINDINGS 19

3.1 Vulnerability Findings 20
3.1.1 Externally Accessible Hosts 20
3.1.2 Web Applications 21

4 ATTACK NARRATIVE 23

4.1 Web Application Pentest Findings 23
4.1.1 Footprinting and Intelligence Gathering 23
4.1.2 Static Binary Analysis and Command Injection 28
4.1.3 Data Transmitted Insecurely 34
4.1.4 Code Injection 35

5 CONCLUSION & RECOMMENDATIONS 38
5.1 Remove Command Injection Vulnerability (CRITICAL) 38
5.2 Review User Input Character Bypass (HIGH) 39
5.3 Incorporate TLS for the Application (HIGH) 40
5.4 Update or Remove Vulnerability JavaScript Libraries (MED-HIGH) 41
5.5 Remove Sensitive Information Exposure (MED-HIGH) 43
5.6 Prevent Clear Text Displays of Data in Transport (MED-HIGH) 44
5.7 Review Use of the “strcpy” Function (INFO) 45
6 REFERENCES 46
7 SUPPLEMENTAL REPORTS 47
APPENDIX A 48

25 April 2023

HIGHLY CONFIDENTIAL

ALTUS CLOUD
Attn: Wile E. Coyote, Chief Architect
Attn: Road Runner, Head of Engineering
Attn: Bugs Bunny, COO
Looney Tunes Entertainment Division, Warner Bros. Studios
4000 Warner Boulevard, Burbank, CA 91522, United States

RE: ALTUS PENETRATION ASSESSMENT REPORT – Q1 2023

Dear Colleagues,

ALLENDEVAUX & COMPANY LLC conducted an independent penetration test[1] of the AltusCloud Network Guardian (ANG) solution during April 2023. The objective of the test was to determine the system's exposure to a targeted attack from an Internet-facing vector. All activities were conducted in a manner that replicated a malicious actor engaged in a targeted attack against AltusCloud with the goals of:

- identifying if a remote attacker can successfully interrupt and/or penetrate the AltusCloud Network Guardian application; and
- determining the impact of a security breach on the confidentiality of data stored and processed within the service, and potential impacts to availability to the service.

Efforts from a grey-box[2] perspective were taken to identify and target security weaknesses that could allow an attacker to gain unauthorized access to confidential data. The attacks were conducted with the level of access that a general user locally to the Network Guardian solution would have as well with the level of access a AltusCloud authenticated user, or an attacker would have who has successfully compromised the application perimeter and obtained valid credentials.

As senior executives and stakeholders, your understanding and active involvement in cybersecurity is crucial for the organization's overall risk management. Recent laws and regulations require stakeholders and key senior executives to receive, review, and act upon the findings of penetration tests. This audit report serves to inform you of the security posture of your organization; it highlights areas that may require attention to mitigate potential risks.

[1] Penetration testing has more of an emphasis on gaining as much access as possible while vulnerability testing places the emphasis on identifying areas that are vulnerable to a computer attack. An automated vulnerability scanner will often identify possible vulnerabilities based on service banners or other network responses that are not in fact what they seem. A vulnerability assessor will stop just before compromising a system, whereas a penetration tester will go as far as they can within the scope of the contract. (Northcutt, et al., 2006)
[2] Grey-box testing typically involves a methodology wherein the tester has "the access and knowledge levels of a user, potentially with elevated privileges on a system." (Poston 2019)

The assessment was conducted in accordance with the recommendations outlined in ISO/IEC 27032. Results of the assessment are organised into the following sections.

- **Section 1: Executive Summary**: a high-level summary of the results of the various tests conducted by cybersecurity assessors. This section also includes a high-level view of vulnerability scanning results.
- **Section 2: Scope and Methodology of the Pentest**: a view of the assets within scope, including IP subnets or address ranges, URIs or URLs associated with services or portals and associated host systems, and how these assets comprise infrastructure clusters. This section also provides visibility to the tools used during pentesting activities.
- **Section 3: Summary of Vulnerability Findings:** a high-level view of the vulnerability results identified during automated scanning.
- **Section 4: Attack Narrative:** the attack narrative conveys important details associated with pentesting attacks employed by cybersecurity assessors, a sampling of results by assessors.
- **Section 5: Conclusion and Recommendation:** depending upon the security posture observed, recommendations are provided as helpful guidance to further harden systems and services.
- **Section 6: References:** where citations are used, the reference section serves as a bibliography.
- **Section 7: Supplemental Reports:** provides instructions for recipients of this report to request additional details in long-form vulnerability reports which may be used by the Client's technical team members to resolve identified issues.
- **Section 8: Appendix A:** provides a concise, easily digestible summary of each finding. This approach is particularly useful when the goal is to provide a quick, yet more technically precise overview of the subject matter or to compare different sets of data.

Taken together, these sections comprise the entire report. Additional information such as further details regarding findings of the vulnerability scans are provided separately and not attached to this document.

Receiving this audit report and acting responsibly upon the findings is crucial for maintaining the organization's security posture and fulfilling legal and contractual obligations. Please carefully review the report and collaborate with your technical team to implement the necessary measures for ensuring the ongoing security of the solution.

Should you have any questions or require further clarification, please do not hesitate to contact us. We appreciate the opportunity to assist you in enhancing your organization's cybersecurity and look forward to working with you in the future.

1 EXECUTIVE SUMMARY

This section outlines the objectives of the cybersecurity assessment, responsibilities of the associated parties, and provides a high-level summary of the assessment findings.

1.1 Assessment Objective

The objective of this assessment is an expression of opinion concerning the cybersecurity posture of the targeted assets. This assessment activities have been overseen by accredited assessors attesting to the findings in this report, which is addressed to the executive leadership of the organisation and the relevant asset owners and administrators. Circumstances may arise in which it is necessary for ALLENDEVAUX to modify the opinions expressed in this report or add an emphasis-of-matter or other-matter paragraph as additional findings or clarification is uncovered.

1.2 Responsibility of Management

It is the responsibility of the organization's management team, its executive stakeholders, its board of directors and the technical stakeholders to study these findings and ensure the information is well understood. ALLENDEVAUX recommends the executive stakeholders set aside time to attend a presentation to be given to step through the findings where questions may be vetted.

It is also the responsibility of management and engineering to take proper action to prioritize issues and remediate known security issues. UNTREATED ISSUES ARE LIABILITIES, POSING SUBSTANTIAL DANGERS TO THE ORGANISATION AND ITS DATA SUBJECTS (CUSTOMERS, STAFF, CONTRACTORS), INCLUDING LEGAL RISKS, REPUTATIONAL RISKS, OPERATIONAL RISKS, AND FINANCIAL RISKS.

1.3 Responsibility of the Assessor

It is the responsibility of ALLENDEVAUX to conduct an independent assessment and express an opinion regarding the security posture of the targeted assets based on the assessor's findings. A thorough assessment has been completed using updated threat definitions, resulting in the following outcomes:

- scored findings using a vulnerability measurement rubric;
- an attack narrative, detailing the attacks employed;
- actionable advice that, when employed, helps to mitigate findings;
- patching advisories for operating systems and firmware;
- prioritized findings classified by CVSS scale; and
- recommendations and considerations toward an improved security posture.

ALLENDEVAUX notes that this report is its expressed opinion of findings, and the results herein are accurate to the best of the assessor's knowledge. This report also serves as an attestation that an independent assessment has been performed by certified processionals in their field of practice.

1.4 Not the Responsibility of the Assessor

It is beyond the scope of this assessment for ALLENDEVAUX to action upon the findings by employing remediation efforts or otherwise. Taking any mitigating action is the responsibility of the client's I.T. organization.

1.5 Assessment Findings

The table below provides a high-level summary of the identified issues (delineated by whether they were discovered via automated vulnerability scan or manual penetration test). Additional details about issues, delineation between asset groups.

Result Source	Asset Type	Critical (SEV5)	High (SEV4)	Med-High (SEV3)
Vulnerability Scan	Internal Hosts	0	0	0
	Web Applications	0	0	0
Penetration Test	Internal Hosts	0	0	0
	Web Applications	1	2	3
	Totals	**1**	**2**	**3**

The table below provides a high-level summary of the significant findings from this penetration test.

RISK LEVEL	DESCRIPTION	# AFFECTED SYSTEM(S)
CRITICAL	Command Injection	1
HIGH	Character Sanitisation Bypass	1
HIGH	Absence of TLS	2
MED-HIGH	Vulnerable JavaScript Library	1
MED-HIGH	Clear Text Password in HTTP Response	1
MED-HIGH	Sensitive Information Exposure	1

Table 1. High-level overview of Penetration Test's results.

1.6 Recommendations

After performing this grey-box penetration test against the ANG, ALLENDEVAUX identified several findings which merit further investigation by the AltusCloud team. Details regarding the evidence and the process each assessor used can be located: Attack Narrative; a deep dive into the technical data can be found here: Appendix A.

1.6.1 Remove Command Injection Vulnerability (CRITICAL)

ALLENDEVAUX assessors identified that system functions within the binary are vulnerable to command injection via the web application. This can allow threat actors to run commands and create reverse shells; furthermore, to make the matter even more critical, these are run as root by default. It is recommended that AltusCloud ensure that all instances of the "system function" that run user input are sanitised for dangerous characters. In addition to this, it is important that AltusCloud minimize the use of the system function and calling the shell.

1.6.2 Review User Input Character Bypass (HIGH)

During the AltusCloud Network Guardian application pentest, it was identified that it is possible to bypass frontend character restrictions by injecting changes within the captured HTTP request. While assessors

were not able to develop a proof of concept for this vulnerability – it remains an issue that needs further review. It is recommended that the AltusCloud team implement a backend sanitisation check to partner the frontend JS check to ensure that no dangerous symbols are injected.

1.6.3 Incorporate TLS for the Application (HIGH)

ALLENDEVAUX assessors discovered that the application was using HTTP instead of HTTPS to transmit data over the network. This means that all the data being transmitted between the application and the client was not encrypted and could be easily intercepted by anyone with access to the network. This puts the data at risk of being stolen or manipulated. To address this finding, it is recommended that AltusCloud incorporate TLS (Transport Layer Security) for all communication between the application and the server. TLS is a protocol that provides encryption and authentication for network connections, ensuring that all data transmitted over the network is secure and protected from unauthorized access.

1.6.4 Update or Remove Vulnerability JavaScript Libraries (MED-HIGH)

The Network Guardian host (http://192.168.10.1) were discovered to have implemented JavaScript libraries with known vulnerabilities. These libraries should be updated or removed from production to ensure vulnerable components are not in use.

Host	Current JS Library & Version	Recommended Upgrade
http://192.167.10.1/#/login	Angular 1.5.5	≥ Angular 1.8.0

1.6.5 Remove Sensitive Information Exposure (MED-HIGH)

During the testing, two instances of information exposure were discovered on the system. The first instance involved a template page that contained potentially sensitive information. Although the information was not considered critical, it was found to be against best practices to have such information available on the system. The second instance occurred due to improper 403 HTTP responses, which allowed assessors to observe JavaScript files used by the web applications. This vulnerability could potentially allow attackers to access sensitive information. It is recommended that AltusCloud review and remove this page and implement appropriate HTTP response codes.

1.6.6 Prevent Clear Text Displays of Data in Transport (MED-HIGH)

ALLENDEVAUX assessors observed that the application was not preventing clear text displays of data in transport. This means that sensitive information such as usernames, passwords, or other confidential data were being transmitted over the network without any encryption or protection, making it vulnerable to interception and unauthorised access. This finding poses a significant risk to the security of the application and its users, as it could potentially lead to data breaches and compromise of sensitive information. To mitigate this risk, it is recommended that AltusCloud implement encoding, encryption, obfuscation or other security measures to prevent clear text displays of data in transport. This can be achieved by using secure communication protocols such as HTTPS or SSL/TLS, or by implementing data encryption at the application level.

1.7 Conclusion of Executive Summary

In conclusion, the following takeaways summarize the pentesting outcomes across the three security tenets of Confidentiality, Integrity and Availability.

1.7.1 Confidentiality

The assessors discovered that data displayed on the Network Guardian webUI was transmitted through HTTP, which exposes it to MitM packet sniffing attacks that can easily pilfer confidential information, including credentials. This vulnerability poses a direct threat to the confidentiality of the web application, making it highly susceptible to unauthorized access and abuse.

1.7.2 Integrity

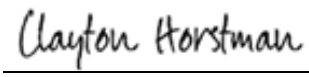
Similarly with the confidentiality issue, the absence of HTTPS and encrypted data in transit can allow threat actors to manipulate data, threatening its integrity. As a result of this, ALLENDEVAUX have deemed this penetration test report acknowledges the impact to integrity.

1.7.3 Availability

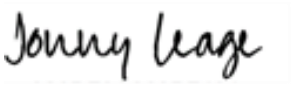
All attempts to interact with the in-scope systems in a manner which would disrupt the availability of the service were unsuccessful.

Thank you for the opportunity to assess the AltusCloud Network Guardian. We trust this report proves to be helpful in your ongoing efforts to ensure the service is hardened. Let's set up a time to walk through the testing together and ensure follow-up actions are understood. For more details, please see the Attack Narrative in Section 4.

Very best regards,



Clayton Horstman
Senior Cybersecurity Analyst
OSCP, CREST CRT, CompTIA Security+



Jonny Leage
Cybersecurity Analyst
CREST CPSA, Certified Ethical Hacker (CEH)

Koushick Prasad
Cybersecurity Analyst
Certified Ethical Hacker (CEH)

ALLENDEVAUX & COMPANY
United States of America | United Kingdom

2 SCOPE AND METHODOLOGY OF PENTEST

This section aims to provide an in-depth understand of the pentest process undertaken by Allendevaux. It covers the targeted systems, the chosen testing perspective, an overview of the tools employed, and any contractual restrictions. Furthermore, this section also elaborates on the PTES methodology followed by Allendevaux, ensuring a thorough and industry-standard approach to penetration testing.

2.1 Testing Perspective

There are various perspectives from which penetration tests may be performed. The pentest was performed from a grey-box perspective. For context, a description of each of the three primary testing perspectives are provided below.

2.1.1 Black-box

OWASP defines black-box testing as follows:

According to the Open Web Application Security Project (OWASP), black-box testing involves assessing a system's security without access to its source code or any internal documentation. The tester treats the system as a "black box," providing input and analyzing output to identify vulnerabilities or potential issues (OWASP, 2021). This approach often involves fuzz testing, which sends different input types, sizes, and patterns to a closed source application to determine its behavior and uncover potential security flaws.

2.1.2 Grey-box

The Infosec Institute defines grey-box testing as follows:

Different than a black-box test where the tester has no knowledge about internal workings, the grey-box tester has partial knowledge of the internal workings of the system, service or web application. In some cases, the grey-box tester may have authentication rights to a service with some privileges and attempts to escalate one’s privileges as part of the test. (Infosec Institute, 2019)

2.1.3 White-box

Redscan defines white-box testing as follows:

White box penetration testing, sometimes referred to as crystal or oblique box pen testing, involves sharing full network and system information with the tester, including network maps and credentials. This helps to save time and reduce the overall cost of an engagement. A white box penetration test is useful for simulating a targeted attack on a specific system utilising as many attack vectors as possible. (Redscan, 2020)

2.2 Target Methodology

Allendevaux assessors adhere to the industry-leading Penetration Testing Execution Standard (PTES) methodology, which provides a comprehensive framework for conducting

penetration tests. Developed collaboratively by a group of information security experts, PTES encompasses seven primary sections: Pre-engagement, Intelligence Gathering, Threat Modeling, Vulnerability Analysis, Exploitation, Post-Exploitation, and Reporting (PTES, 2014).

The PTES methodology aims to standardise penetration testing across industries and cater to businesses requiring such services (Hout, 2019). Allendevaux focuses predominantly on the Intelligence Gathering, Vulnerability Analysis, and Exploitation stages of the methodology, employing both active and passive techniques suggested during the testing phases of each VAPT.

The Penetration Testing Execution Standard (PTES) offers numerous benefits for both pentesting companies and clients. By employing the PTES methodology, pentesting companies can assure their clientele that they are adhering to a well-established, comprehensive, and industry-standard approach to evaluating the security of systems and networks. Below are some of the key benefits of employing the PTES methodology:

- **Consistency and Standardisation:** The PTES methodology provides a structured framework that ensures a consistent approach across various engagements, industries, and organisations. This consistency benefits clients by allowing them to compare the results of different pentests or providers, ensuring they receive reliable and comparable security assessments.
- **Comprehensive Coverage:** PTES was created through a collaborative effort by a group of information security experts who aimed to develop a thorough and inclusive guide for penetration testing (PTES, 2014). By following this methodology, pentesting companies can ensure they cover all relevant aspects of the assessment, reducing the likelihood of overlooking critical vulnerabilities.
- **Continual Improvement:** The PTES methodology is regularly updated by the information security community, ensuring that it stays current with emerging threats, new technologies, and evolving security practices. This ongoing refinement allows pentesting companies to stay ahead of the curve, providing clients with up-to-date assessments and recommendations.
- **Clear Communication and Reporting:** PTES includes guidelines for clear and concise communication between the pentesting company and the client. By adhering to these guidelines, pentesting companies can ensure they deliver actionable, understandable, and well-documented reports, enabling clients to make informed decisions about their security posture and risk mitigation strategies.
- **Trust and Credibility:** The PTES methodology is widely recognised and respected within the information security industry. By employing this methodology, pentesting companies can demonstrate their commitment to maintaining the highest professional standards, instilling confidence in their clients and differentiating themselves from competitors who may not adhere to such rigorous practices.

In conclusion, employing the PTES methodology allows Allendevaux to deliver consistent, comprehensive, and up-to-date security assessments to its customers. By adhering to this well-established standard, Allendevaux demonstrates its commitment to excellence and

instills confidence across its clientele, making PTES a positive differentiator in the competitive landscape of penetration testing services.

2.3 Target Inventory

A set of IP ranges including externally accessible target IP addresses and web applications were provided to the pentest team for the scope of this engagement. The tables below outline the target inventory used for this pentest.

The following web applications were also included in the scope of this engagement.

URI	Business Context	Production?
http://192.168.10.1	Network Guardian Admin Interface	No
http://192.168.11.1	Network Guardian Admin Interface (dupe)	No

Table 2. In scope target Inventory on a private RFC 1918 address.

Introducing a detailed technical overview of the features incorporated into our cutting-edge AltusCloud product, known as the "AltusCloud Network Guardian" (ANG). The arrow symbol represents the direction of service, either inbound (server) or outbound (client). Please note that not all services can be accessed via every network interface (e.g., cellular, ethernet, and Wi-Fi), and their availability can be adjusted through software settings. Additionally, a customizable firewall is integrated into the ANG to regulate access to specific services as needed.

2.4 Tools Used

In any cybersecurity assessment, the choice of tools plays a critical role in effectively identifying vulnerabilities and potential threats. Allendevaux recognises the importance of employing a diverse and robust suite of tools[3] to ensure comprehensive testing and accurate results. Our certified cybersecurity professionals utilise these tools in an ethical manner, adhering to legal and industry standards, while maintaining strict compliance with the terms and conditions of the licensed software. This section outlines the various pentesting tools used by Allendevaux during engagements, offering insight into their specific functions and contributions to the assessment process.

- **Qualys:** A cloud-based vulnerability scanning and management solution, Qualys helps identify system vulnerabilities, manage malware, execute controlled cross-site scripting, perform dictionary attacks, and test for SQL injections. This tool streamlines vulnerability management and ensures a comprehensive approach to system assessment.

[3] This may not be an exhaustive list of all tools which were used during penetration testing. Other tools are used at assessors' discretion during pentesting based on assessment of the target environment. However, this listing provides insight into the primary testing tools which were used.

- **Kali Linux:** A specialised Linux distribution designed for cybersecurity professionals, Kali Linux comes preloaded with a suite of ethical hacking tools, enabling our assessors to perform a wide range of penetration testing activities.
- **Metasploit:** A powerful network discovery and exploitation framework, Metasploit allows our cybersecurity experts to identify vulnerabilities, develop and execute exploits, and simulate real-world attack scenarios.
- **Feroxbuster/Gobuster/Ffuf:** These web application directory traversal and brute force tools enable our cybersecurity experts to discover hidden files, directories, and resources, exposing potential attack vectors.
- **Sqlmap:** An open-source penetration testing tool, Sqlmap automates the detection and exploitation of SQL injection vulnerabilities in web applications, helping our assessors to identify potential database breaches.
- **Nmap:** A network mapping and enumeration tool, Nmap helps our cybersecurity professionals to discover open ports, services, and the overall network topology, providing valuable insights into the target's infrastructure.
- **Cewl:** A custom dictionary generation tool, Cewl allows our assessors to create targeted wordlists for password cracking and brute force attacks based on the target's website content.
- **John/Hydra/Ncrack:** These password cracking tools enable our cybersecurity professionals to test password strength and identify weak or easily guessed credentials, highlighting potential security risks.
- **Burp Suite Pro:** A web traffic analyser and vulnerability scanner, Burp Suite Pro assists our assessors in identifying and exploiting web application vulnerabilities through real-time traffic analysis and manipulation.
- **Nikto/W3af/Skipfish/ZAP:** These web application vulnerability analysis tools allow our assessors to identify common vulnerabilities, such as misconfigurations, outdated software, and insecure coding practices, ensuring a comprehensive evaluation of web application security.
- **WPScan:** Focused on WordPress security, WPScan enables our assessors to identify vulnerabilities within WordPress installations, plugins, and themes, ensuring a thorough evaluation of WordPress-based websites.
- **Wireshark:** A network traffic analyser, Wireshark assists our cybersecurity professionals in capturing and analyzing network packets, providing invaluable insights into the target's network communications and potential vulnerabilities.
- **SIPVicious:** A SIP enumeration and brute forcing tool, SIPVicious helps our assessors to identify vulnerabilities within Voice over IP (VoIP) infrastructure and evaluate the security of SIP-based communication systems.

By employing this diverse suite of pentesting tools, Allendevaux's certified cybersecurity professionals can conduct comprehensive and accurate security assessments, ensuring our clients receive the best possible evaluation of their systems' security posture.

2.5 Engagement Restrictions

There were no restrictions placed upon the engagement.

3 SUMMARY OF PENTRATION TEST FINDINGS

This section presents a high-level overview of the vulnerability findings identified during the penetration test of the service. Before delving into the details, it is essential to understand the distinction between penetration testing and vulnerability scanning.

Penetration testing, as defined by Whitaker and Newman (2005), is an extensive, manual process wherein cybersecurity experts simulate real-world attacks to uncover potential vulnerabilities and exploit paths within a system. This approach surpasses automated vulnerability scanning by incorporating a diverse array of testing techniques to thoroughly assess an organisation's security posture. For example, penetration testing may involve social engineering tactics, such as phishing, to determine an organisation's susceptibility to targeted email attacks. It may also encompass testing web applications for vulnerabilities like SQL injection and cross-site scripting (XSS) that could allow attackers unauthorised access to sensitive information (OWASP Foundation, 2021). Furthermore, penetration testers may attempt to bypass physical security controls, such as locks or access card systems, to gain unauthorised entry to a facility. (Whitaker & Newman, 2005)

Vulnerability scanning, in contrast and as described by Scarfone and Mell (2007), is an automated process that systematically scans and analyses a system for potential vulnerabilities and misconfigurations. Although valuable, vulnerability scanning alone does not offer the same level of in-depth insight as a full penetration test, since it primarily relies on automated tools and predefined vulnerability databases (Scarfone & Mell, 2007). Examples of vulnerability scanning include checking for missing security patches, outdated software versions, and configuration errors that could expose a system to potential attacks (Chappel, Seidl, & Stewart, 2019). These scans can help organisations identify weaknesses in their networks, applications, and infrastructure before attackers can exploit them.

To maintain optimal security, vulnerability scanning should be performed at a regular cadence, such as monthly or quarterly, depending on the organisation's risk appetite and industry best practices (Peltier, 2016). Regular scanning enables organisations to stay informed about their security posture and proactively address identified weaknesses before they can be exploited.

The findings presented in this section are the culmination of both vulnerability scanning and penetration testing activities performed by our cybersecurity professionals. To provide a clear and concise representation of the results, we have employed a rubric for measuring the severity levels of the vulnerabilities discovered. The abbreviation "SEV" represents the word "severity" and has a scoring range from 5 (highest severity) to 1 (minimal severity), based on a scale adopted by Qualys. These severity ratings are also mapped to the broadly accepted Common Vulnerability Scoring System (CVSS[4,5]) standard across a ten-point scale.

[4] According to NIST, the Common Vulnerability Scoring System (CVSS) is an open framework for communicating the characteristics and severity of software vulnerabilities. CVSS consists of three metric groups: Base, Temporal, and Environmental. The Base metrics produce a score ranging from 0 to 10, which can then be modified by scoring the Temporal and Environmental metrics.

[5] This mapping is an estimation for reference purposes only. Qualys does not map its proprietary vulnerability severity ratings directly to CVSS scores. More information is available on the Qualys website.

By combining the automated vulnerability scanning results with the manual penetration testing efforts, this section offers a thorough and insightful summary of the security vulnerabilities identified within the scope.

RISK LEVEL	CVSS 2.0[6]	SEVERITY RATING	DESCRIPTION	SERVICE FINDINGS	WEB APP FINDINGS
HIGH	7.0 to 10.0	SEV5	High probability intruders may gain control of the host.	0	0
		SEV4	Moderate probability intruders may gain control of the host.	0	6
MED	4.0 to 6.9	SEV3	Intruders may gain access to facets of host.	0	21
		SEV2	Intruders may collect some information about a service or host.	3	1
LOW	0.0 to 3.9	SEV1	Intruders may collect information about ports and services.	0	6
		SEV0	Minimal information gathering.	88	30

Table 3. Summary of Vulnerability Scanning findings, including web application and network-based testing.

No SEV5 findings (CVSS 8.0 – 10.0) avenues of attack were discovered during this pentest. However, multiple SEV4s and SEV3s were identified within the web application scans that should be reviewed thoroughly.

3.1 Vulnerability Findings

From an external perspective, the vulnerability findings showed limited exposure to threats. The results of the team's vulnerability assessment of host systems are enumerated in the following sections.

3.1.1 Externally Accessible Hosts

The number of identified vulnerabilities across the externally accessible assets are comprised of a relatively small number of low critical vulnerabilities based around encryption.

[6] The Qualys engine used in this study supports CVSS 2.0 and 3.0; here, the CVSS 2.0 scale is used because CVSS 3.0 scores were not available for all findings.

by Severity				
Severity	**Confirmed**	**Potential**	**Information Gathered**	**Total**
5	0	0	0	0
4	0	0	0	0
3	0	2	5	7
2	3	0	8	11
1	0	0	75	75
Total	**3**	**2**	**88**	**93**

5 Biggest Categories				
Category	**Confirmed**	**Potential**	**Information Gathered**	**Total**
Information gathering	0	0	33	33
TCP/IP	0	0	30	30
General remote services	0	2	12	14
CGI	1	0	6	7
Web server	2	0	4	6
Total	**3**	**2**	**85**	**90**

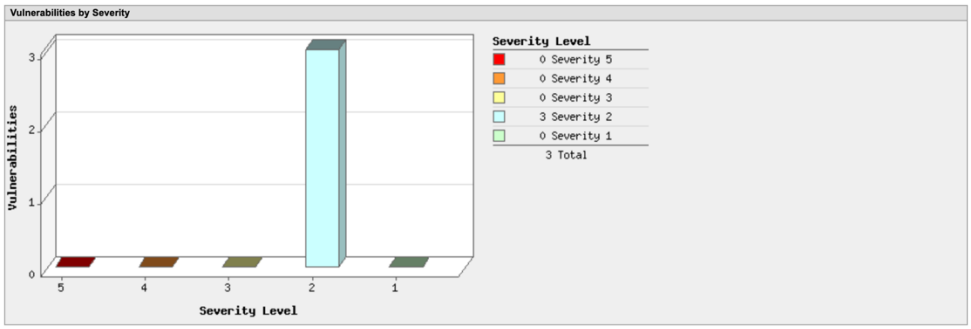


Figure 1. Vulnerability scanning of the IP addressable asset results.

3.1.2 Web Applications

The AltusCloud Appliance web interface were targeted with WAS vulnerability scanning to test for OWASP vulnerabilities. Several areas of the OWASP Top 10:2021 were identified during the test scanning including Cryptographic Failures, Security Misconfigurations, Broken Access and more.

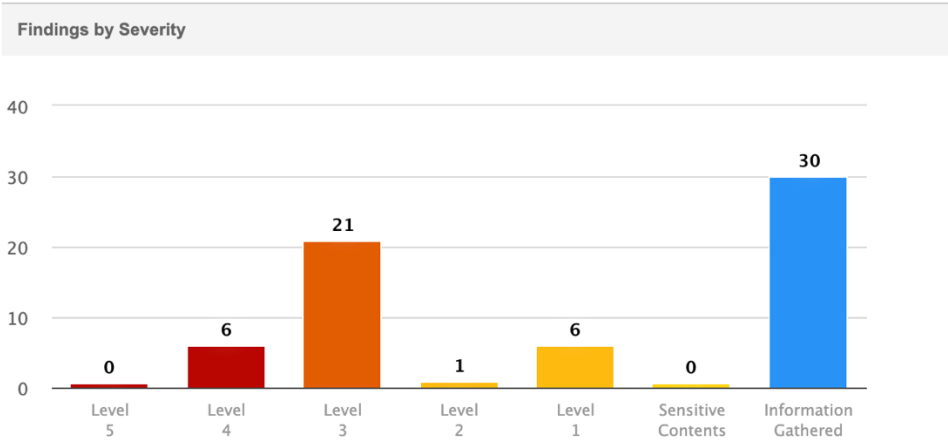


Figure 2. Screen capture from the Qualys WAS Scan results.

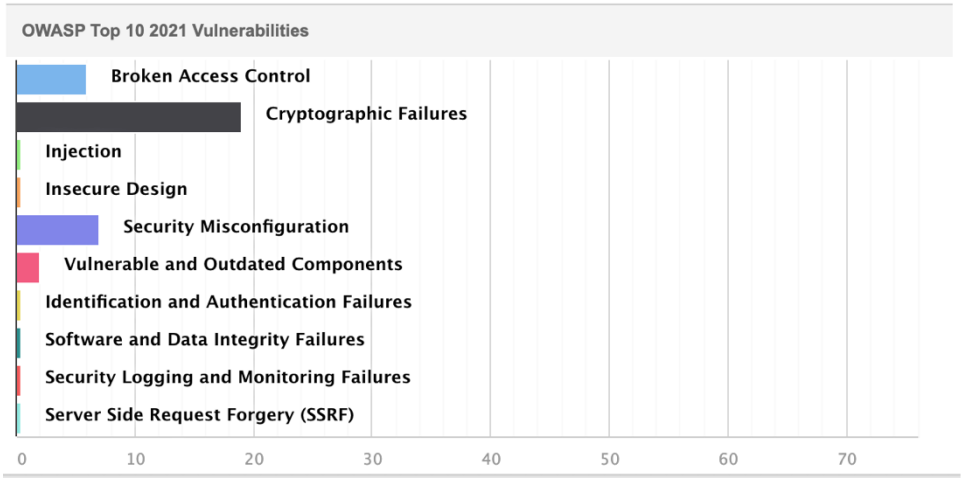


Figure 3. Screen capture from the Qualys WAS Scan OWASP findings.

Testing of the web portals was performed using Qualys for external vulnerability scanning and a number of other tools (e.g. Nikto, Zed Attack Proxy) for internal scanning using test account credentials created specifically for this engagement.

4 ATTACK NARRATIVE

The "Attack Narrative" is a crucial part of the penetration testing report, providing a comprehensive chronicle of the various exploitations attempted by the pentesting team. It provides the reader with a detailed walkthrough of the steps taken, tools used, and methodologies employed during the assessment. This narrative is integral to penetration testing reports for several reasons.

Firstly, it serves as a testament to the thoroughness of the assessment, detailing each stage of the penetration test, including service detection, system and network information extraction, JavaScript and command injection, and automated exploitation. It provides a clear picture of the extent of the testing done, even when a low number of confirmed vulnerabilities are found. This transparency aids in understanding the depth of the penetration testing performed.

The attack narrative is also a best practice in the field of penetration testing, as it provides valuable information to various stakeholders. For the technical audience, primarily engineers and developers, it helps in understanding the precise steps that led to the identified vulnerabilities. This level of detail enables them to reproduce the anomalies discovered, which is crucial for the development and testing of appropriate fixes (OWASP, 2021).

For non-technical stakeholders, the narrative provides an overview of the testing process, which can help in understanding the scope and value of the penetration test, and the importance of addressing the identified vulnerabilities.

However, due to its detailed nature, the attack narrative should be considered highly confidential. It provides a step-by-step guide to the vulnerabilities found in the system, and if it falls into the wrong hands, a malicious actor could reproduce the steps to compromise the system, service, or platform. Therefore, strict controls should be in place to ensure that the report is shared only with authorized individuals and is stored securely (Digital Guardian, 2021).

The subsequent sections of the attack narrative will delineate the attempted attacks based on their targets, whether they were web applications or host systems. This segregation helps in understanding the unique challenges and vulnerabilities associated with different parts of the system, assisting in the development of more targeted and effective countermeasures.

In essence, the Attack Narrative is an essential component of a penetration testing report that helps in bridging the gap between the technical and non-technical, while also serving as a roadmap for the development and implementation of security enhancements.

4.1 Web Application Pentest Findings

The AltusCloud Network Guardian underwent penetration testing from a grey-box perspective. Several notable weaknesses were observed within the limited scope. The sections below outline the attacks undertaken by the ALLENDEVAUX assessors.

4.1.1 Footprinting and Intelligence Gathering

Upon reviewing the web application, `http://192.168.10.1` and `http://192.168.11.1` it was observed that a user accessing the Network Guardian application was met by an authentication portal requesting a single

password for access. Assessors reviewed the application source code and fuzzed directories associated with the application.

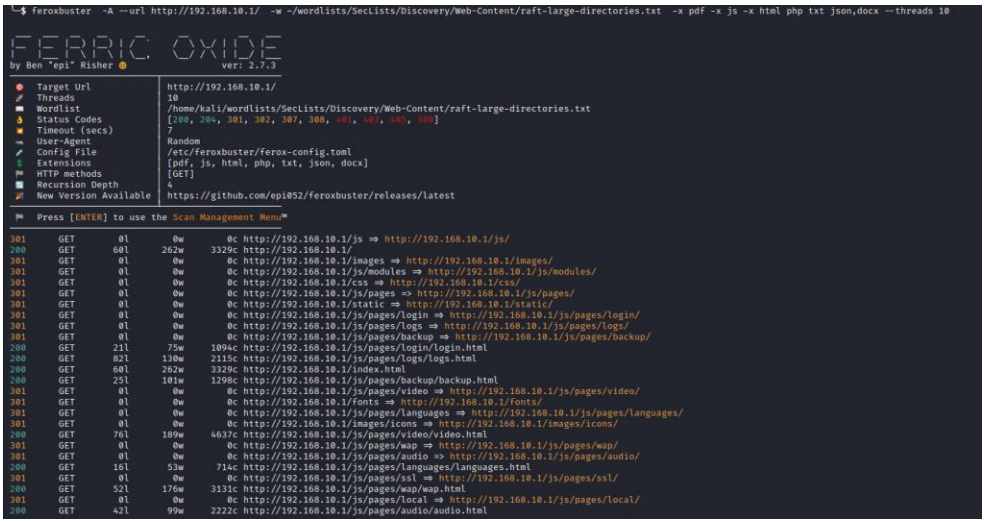


Figure 4.Directory enumeration CLI tool, Feroxbuster, identifying accessible directories and files of http://192.168.10.1.

Assessors endeavored to uncover the extended scope of the appliance web assets, fuzzing subdirectory lists against `http://192.168.11.64`. Many different wordlists were leveraged during these attempts, and several files' addresses were found but none were accessible via unauthenticated access.

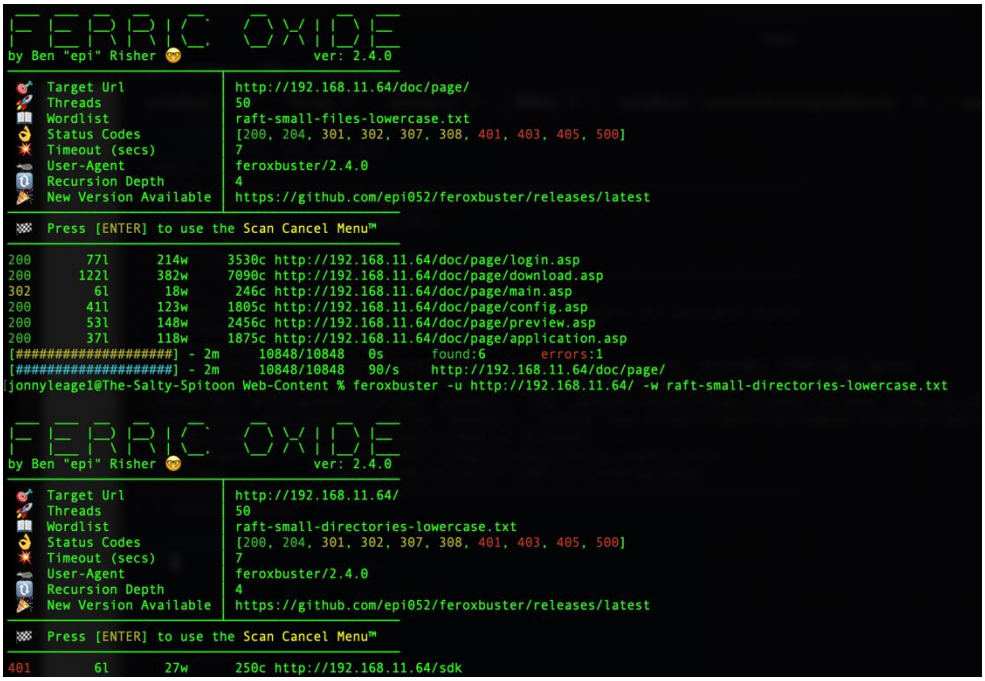


Figure 5. Directory Testing Using CLI Tools for http://192.168.11.64.

During the footprinting phase, it was found that the Network Guardian exposed JavaScript files due to improper use of the 403-error response. The application should have returned a 403 error when attempting to access directories that are

forbidden, but instead displayed various JavaScript directories, even though the data was inaccessible. While no sensitive data was exposed, this finding is concerning as it reveals the application's internal file structure and could potentially provide attackers with useful information for future attacks.

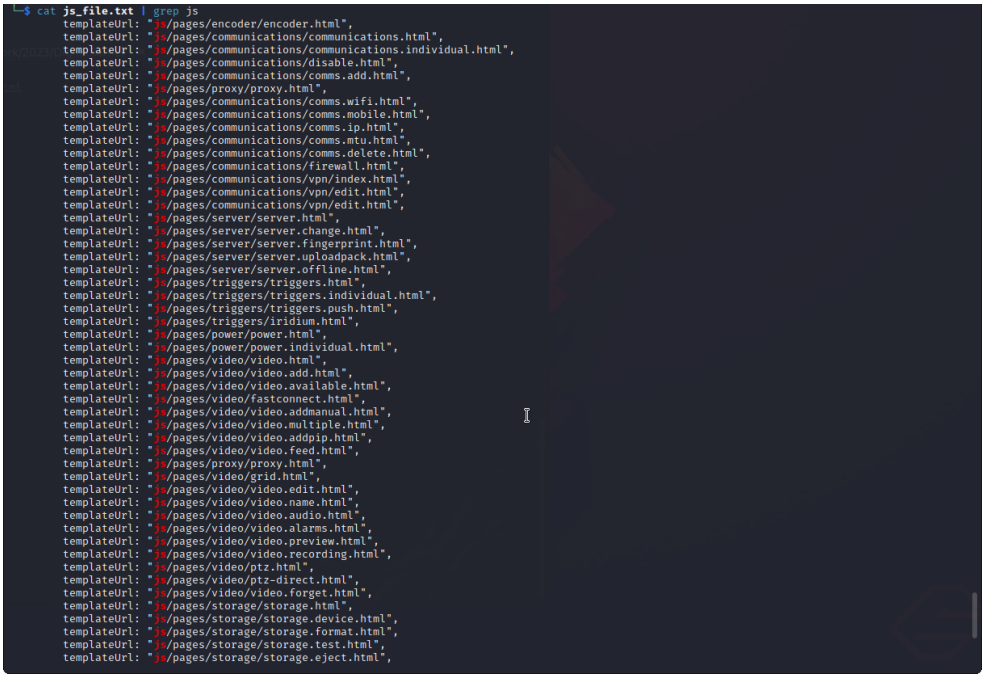


Figure 6. Assessors exploiting the absence of 403 forbidden response to output all JavaScript files.

Assessors discovered a vulnerability in the system where a template page containing potentially sensitive information was accessible. Through reconnaissance and reconnaissance-based attacks, the attacker was able to identify the location of the template page and accessed the page, allowing them to obtain the sensitive information.

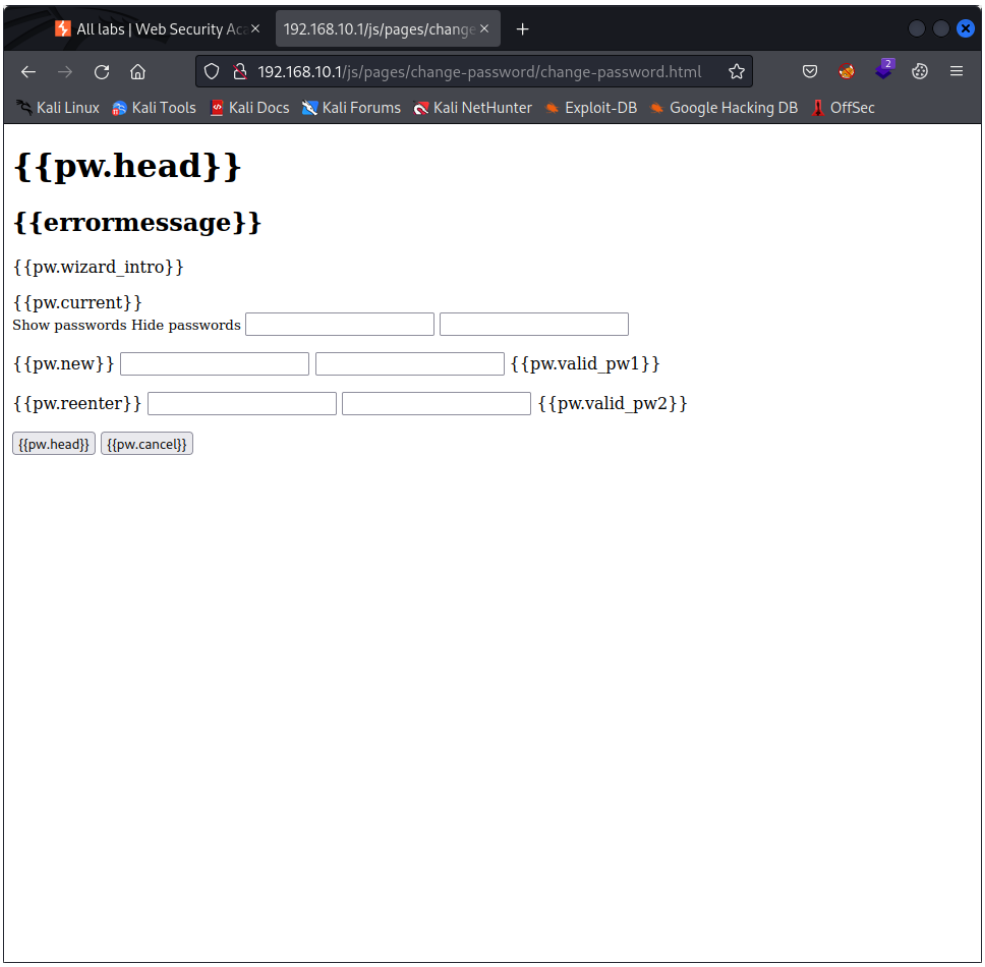


Figure 7. Change Password Template HTML Page with potentially sensitive information.

During the footprinting phase, the assessors attempted to use CLI web vulnerability scanners, such as Nikto, to identify vulnerabilities in the target system. The scanning alerted assessors that the target application utilises WebDAV.

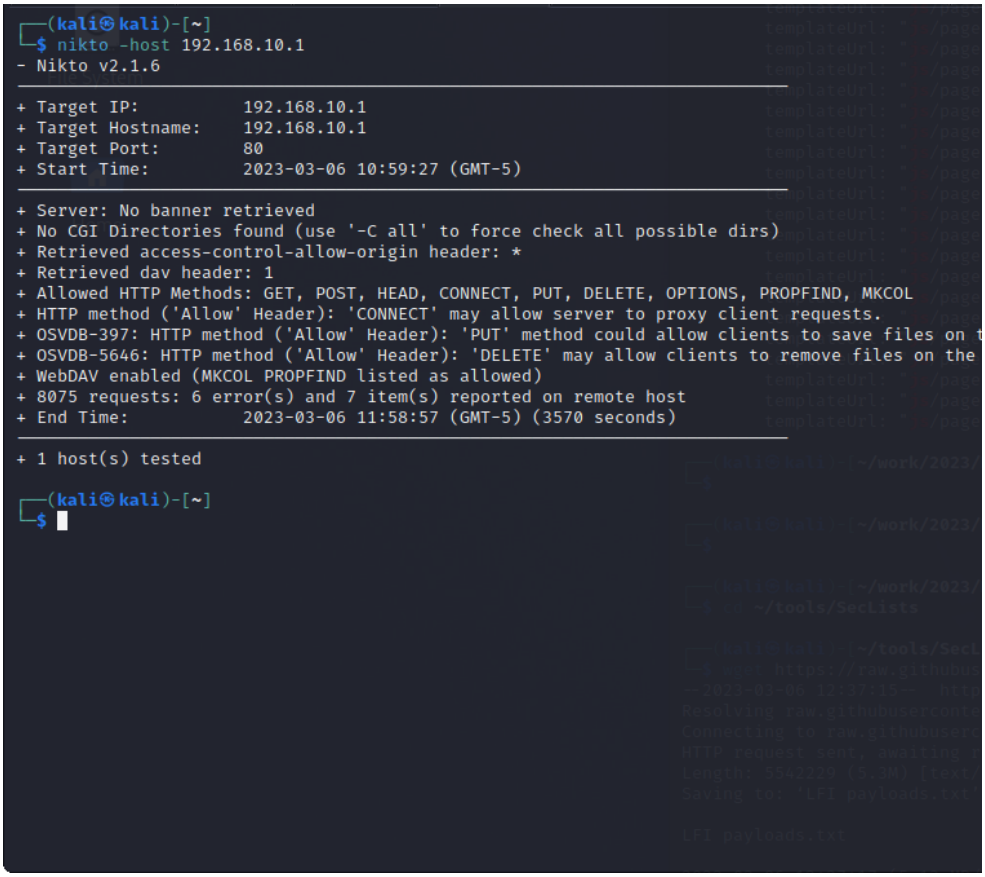


Figure 8. Nikto scan directed to the target host.

WebDAV, or Web Distributed Authoring and Versioning, is an extension of the HTTP protocol that enables users to collaboratively edit and manage files on remote web servers. WebDAV facilitates file sharing and collaboration, making it a crucial component for many organizations. However, it also presents potential security risks as it grants unauthorized users access to sensitive files and folders. Consequently, it is crucial to secure WebDAV to prevent unauthorized access, tampering, or data leaks. To assess the security of the application, the assessors attempted to exploit any vulnerabilities in WebDAV but were unsuccessful in identifying any significant weaknesses.

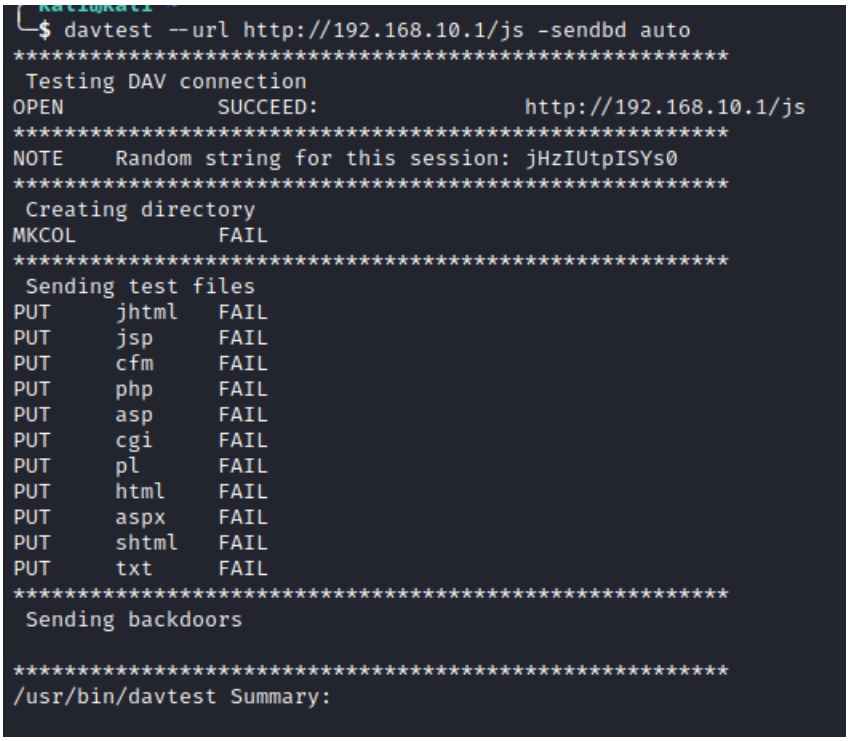


Figure 9. WebDAV testing unsuccessful.

4.1.2 Static Binary Analysis and Command Injection

During the penetration testing engagement, the assessors had access to the application binary and were able to perform a static analysis on the application. This allowed them to analyse the code and identify any potential vulnerabilities or weaknesses in the application's design or implementation. The static analysis process involved examining the code without actually executing it, which helped to identify potential security issues such as buffer overflows, injection vulnerabilities, and other code-related weaknesses. The results of the static analysis were then used to inform further testing and to help prioritize any necessary remediation efforts.

ALLENDEVAUX assessors conducted a white box review of the application by SSH-ing into it and examining the files associated with the application. This approach allowed the assessors to conduct a more comprehensive analysis of the application's security posture by examining the code from an internal perspective. By reviewing the JS files, the assessors were granted full exposure to potential security issues such as authentication bypass, insecure communications, or hardcoded credentials. Additionally, the assessors analyzed the application's logic flow and input validation mechanisms to identify any potential vulnerabilities that could be exploited by attackers.

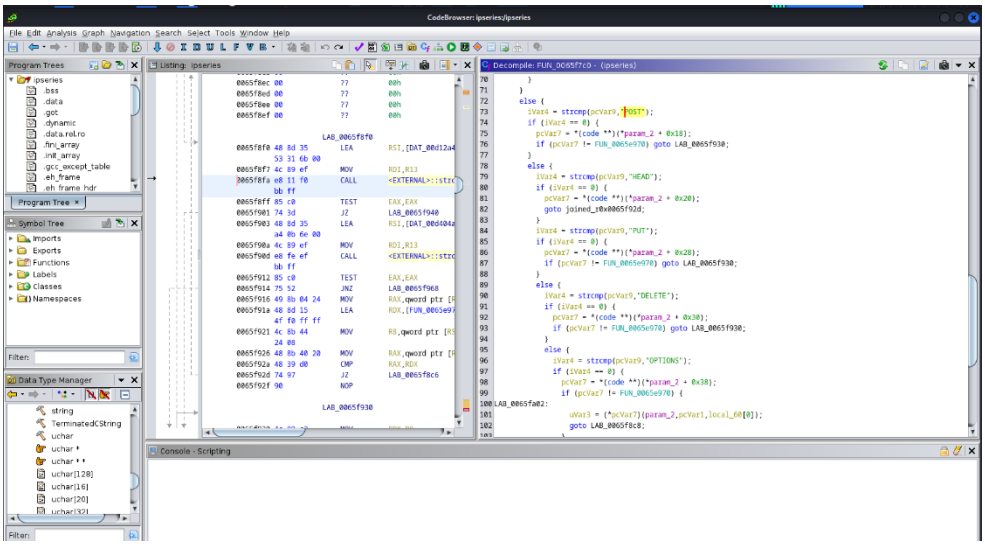


Figure 10. Identification of HTTP requests within a binary decompiler during the static analysis.

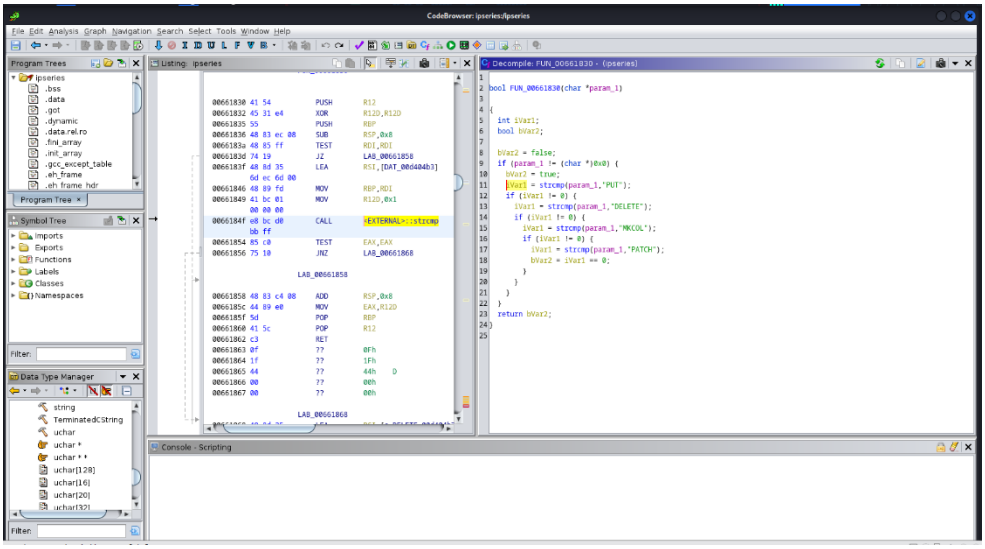


Figure 11. Further static analysis of the HTTP request & response process.

Assessors performed static analysis of the app binary and identified the use of the "`strcpy`" function. Even though no successful exploitation was carried out, the use of this function is considered poor practice as it can lead to buffer overflow vulnerabilities. Moreover, the assessors also detected dynamic entries "`%s`" to the system function which could potentially be misused by attackers. Although no successful exploitation occurred, it is important to flag this finding for further review and potential remediation to enhance the application's security posture.

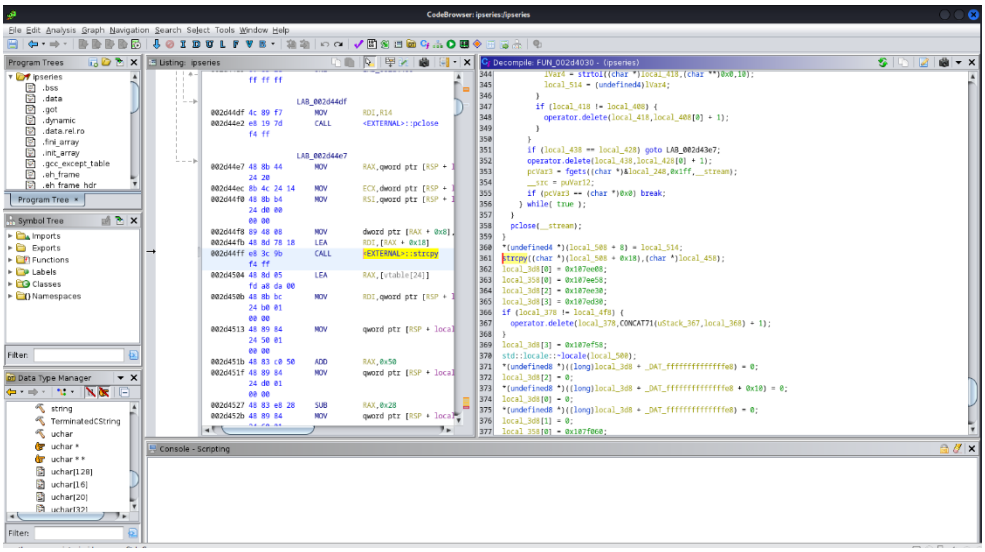


Figure 12. Traditionally unsafe strcpy function being utilised by the binary.

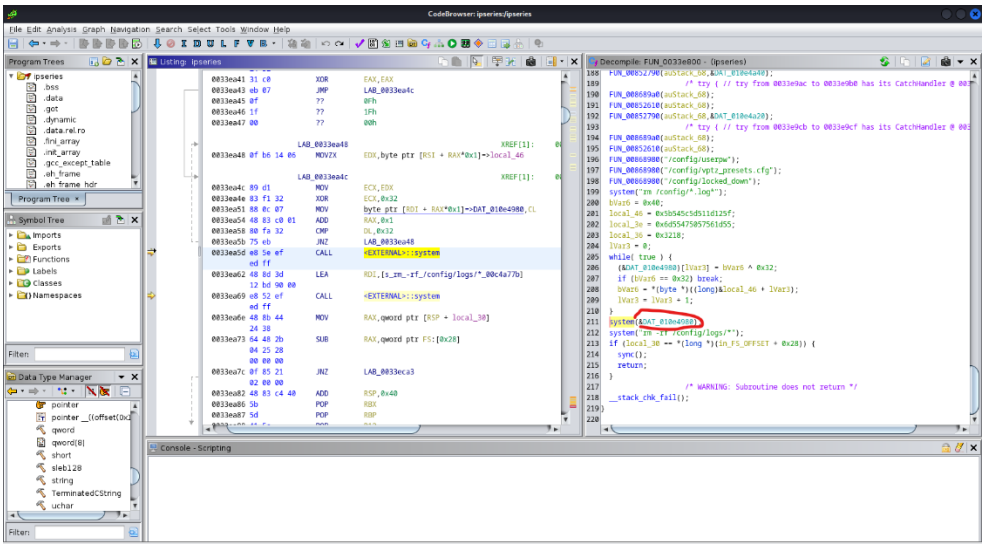


Figure 13. Potential use of dynamic entry into the system function was flagged by assessors as a dangerous practice.

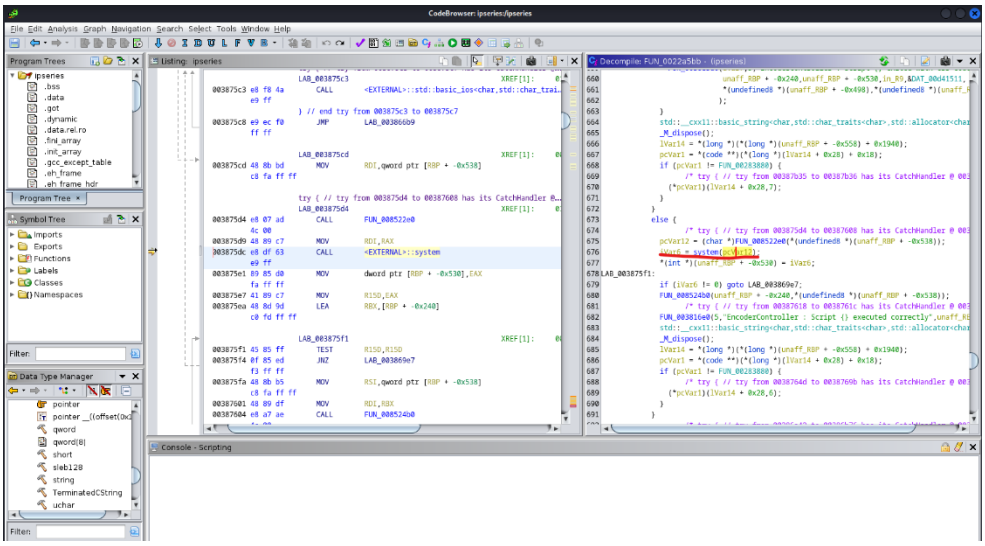


Figure 14. Further potential use of dynamic entry into the system function was flagged by assessors as a dangerous practice.

During the static analysis of the web application binary, ALLENDEVAUX assessors discovered a security vulnerability in the implementation of a system command executed on the backend. The use of the system command was found to be insecure, allowing users to escape the command via the frontend of the application. This vulnerability enabled the assessors to perform command injection attacks, which allowed them to execute arbitrary commands on the backend server. As a result, a threat actor could gain unauthorized access to sensitive data, modify critical files, and take control of the server.

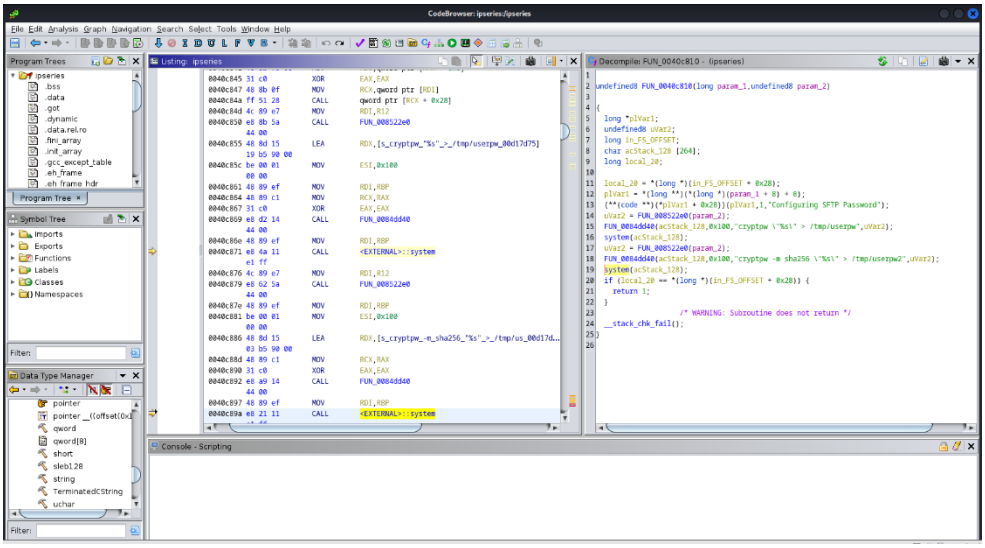


Figure 15. The image above outlines the use of the system function in the app's binary.

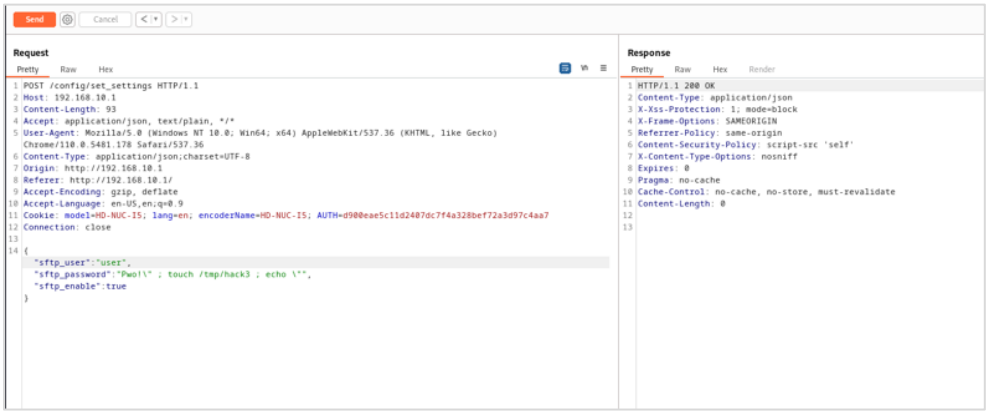


Figure 16. SFTP password update request, with a command escape (;) and POC command.

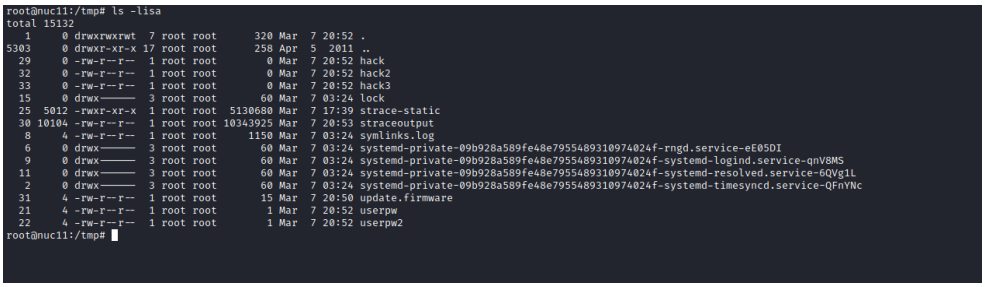


Figure 17. A POC directory, "hack", was created as a result of the command injection finding. This file also runs as root by default.

Binary review continued as assessors attempted to discover any other hard coded vulnerabilities. During this analysis, assessors managed to identify United Kingdom phone provider details that should be reported. This finding has been filed as a point of interest due to the credentials likely being publicly accessible information.

```
var apnObj = [{
        name: "BTMOBILE-UK",
        display: "BT Mobile",
        apn: "btmobile.bt.com",
        country: "uk",
        username: "bt",
        password: "bt",
        comms: "gsm"
    }, {
        name: "EE-UK",
        display: "EE",
        apn: "everywhere",
        country: "uk",
        username: "eesecure",
        password: "secure",
        comms: "gsm"
    }, {
        name: "O2-UK",
        display: "O2",
        apn: "mobile.o2.co.uk",
        country: "uk",
        username: "mobileweb",
        password: "password",
        comms: "gsm"
    }, {
        name: "ORANGE-UK",
        display: "Orange",
        apn: "orangeinternet",
        country: "uk",
        username: "user",
        password: "pass",
        comms: "gsm"
    }, {
        name: "STREAM-UK",
        display: "Stream Communications",
        apn: "stream.co.uk",
        country: "uk",
        username: "streamip",
```

Figure 18. UK phone provider credentials.

Additionally, the static code review prompted assessors to question the unusual HTTP status codes used by AltusCloud.

```
}]), evApp.controller("login", ["$rootScope", "$scope", "$state", "$timeout", "encoder", "$cookies", "$interval", "focus", "capabilities", function (a, o, i, c, r, s
    a.$state = i, a.loggedIn = !1, a.username = "", o.error = !1, o.form = {}, o.login = a.content.login_content, angular.isDefined(a.pageRefresh) && t.cancel(a.pagei
        r.login({}).then(function (e) {
            o.nonce = e.data.nonce;
            var t = {
                password: SHA1(SHA1(o.loginpassword) + o.nonce),
                nonce: o.nonce
            };
            n && (t.overridelogin = "yes"), r.login(t).then(function (e) {
                d.get(), s.put("AUTH", e.data.session), a.username = "Administrator", a.loggedIn = !0, i.go("encoder")
            }, function (e) {
                switch (o.form.form.$setPristine(), e.status) {
                    case 429:
                        o.errormessage = "Too many failed login attempts. Please try again in " + Math.ceil(e.data.timeout / 60) + " minutes.";
                        break;
                    case 462:
                        o.errormessage = "No password supplied.";
                        break;
                    case 464:
                        o.loginForm = !1, o.error = !1, c("force");
                        break;
                    case 500:
                        o.errormessage = "Internal server error.";
                        break;
                    case 461:
                        o.errormessage = "Incorrect password.";
                        break;
                    case 460:
                        o.errormessage = "Unknown request.";
                        break;
                    case -1:
                        o.errormessage = "Connection has been refused, the encoder may be unavailable.";
                        break;
                    case 463:
                        o.errormessage = "Unauthorized."
                }
                464 != e.status && (o.loginForm = !0, o.error = !0)
            })
        }, function (e) {
            a.error(e.status, !0)
```

Figure 19. User login logic resolves to peculiar HTTP Status Codes.

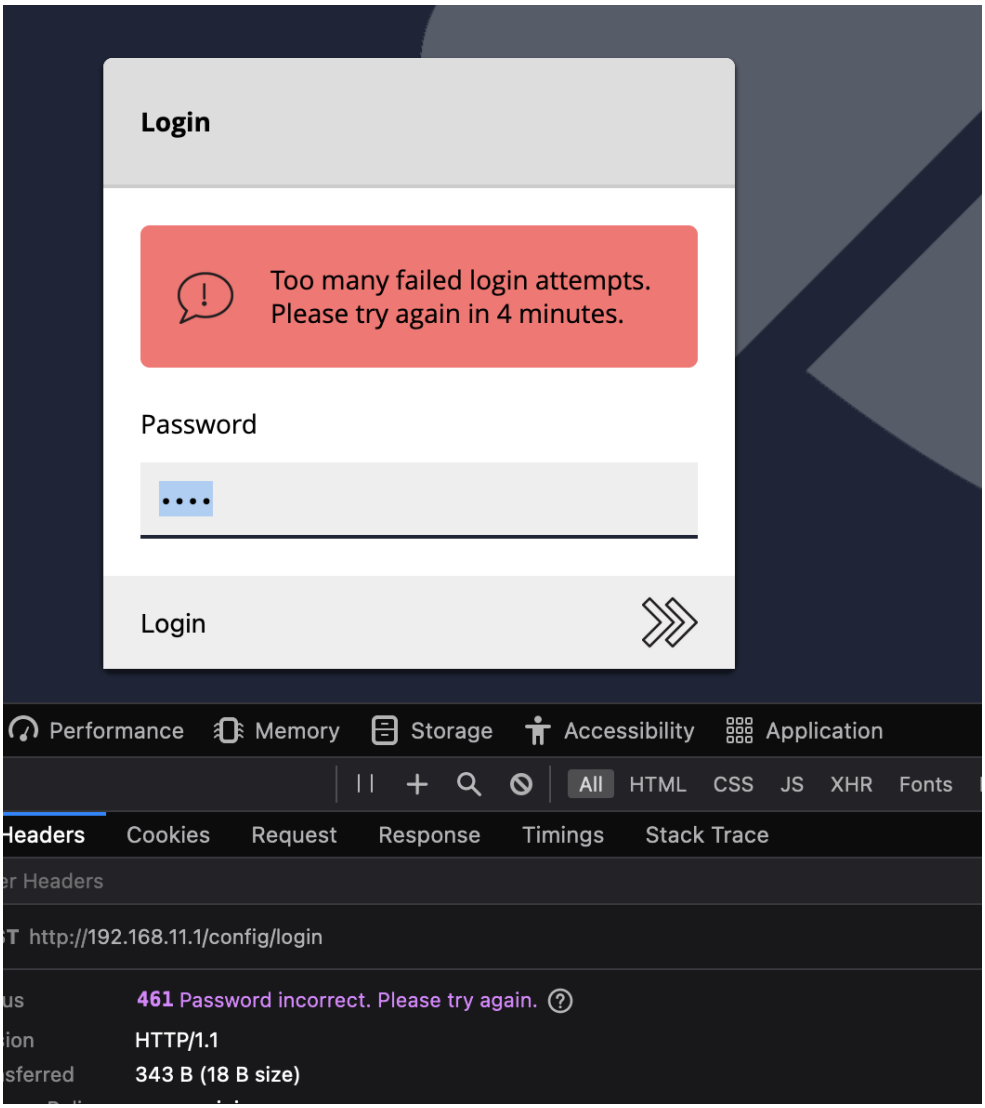


Figure 20. Bruteforce attack thwarted by rate limiting, despite the strange HTTP status code.

4.1.3 Data Transmitted Insecurely

During the assessment of the target web application, it was swiftly identified that the application did not utilise HTTPS protocol for data transmission. As a result, all data sent between the client and the server was transmitted insecurely, including authentication tokens and passwords. This leaves the application vulnerable to a range of security threats, such as eavesdropping, man-in-the-middle attacks, and session hijacking. Instances of this were often observed by assessors as requests would be requested within their BurpSuite tools. "`/config/camera_settings`" was an unprompted request that would routinely render in the background, with the response containing the IP camera's credentials. The response also directs the user to the exact endpoint in which the credentials can be inputted, and the video feed can be viewed:

“rtsp://192.168.11.64:554/Streaming/Channels/101?transportmode=unicast&profile=Profile_1”.

```
GET /config/camera_settings HTTP/1.1
Host: 192.168.10.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15;
rv:103.0) Gecko/20100101 Firefox/103.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.10.1/
Connection: close
Cookie: AUTH=16df5bfaeb39217f7d1f32390e1ccec1006d3e9f;
lang=en
```

Figure 21. /config/camera_settings request captured in burp.

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Xss-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
Content-Security-Policy: script-src 'self'
X-Content-Type-Options: nosniff
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, must-revalidate
Content-Length: 1240

{"layouts":[],"licensed_channels":0,"max_channels":16,"feed
s":[{"details":{"interface_name":"Ethernet: LAN
2","interface_id":"eth1_config","ip_address":"192.168.11.64
","auto_configure":true,"alarm_source_enabled":false,"passw
ord":"REDACTED","make":"HIKVISION","friendly_name":"cam","a
larm_source_port":4444,"firmware_version":"V5.4.6 build
170629","hardware_id":"88","analytics_type":"","auto_config
ure_bandwidth":3000,"index":0,"username":"admin","type":"ON
VIF","ptz":{"enabled":false},"port":80,"add_secureconnect":
true,"transport":"tcp","watchdog_monitor":false},"config":{
"enabled":true,"rtsp_url":"rtsp://192.168.11.64:554/Streami
ng/Channels/101?transportmode=unicast&profile=Profile_1","o
vf_audio_token":"","ptz_url":"","img_url":"http://192.168.1
1.64/onvif/Imaging","ovf_media_token":"Profile_1","media_ur
l":"http://192.168.11.64/onvif/Media","dev_url":"http://192
.168.11.64/onvif/device_service","ovf_audio_source_token":"
","ovf_source_token":"VideoSource_1","ovf_ptz_config_token"
:"","ovf_ptz_presets":"","read_only":false},"stream":{"stat
us":"Streaming","video_resolution":"1920 x
1080","video_avg_fps":"25.0","video_avg_bitrate":2625712,"v
ideo_codec":"H264"}}]}
```

Figure 22. /config/camera_settings response captured in burp.

4.1.4 Code Injection

Unlike the command injection finding referenced earlier, assessors attempted to discover code injection attacks against the target system using various injection

techniques, such as SSTI, JavaScript, and HTML injection. However, most of the injection attempts were unsuccessful as the system was found to be properly sanitised, effectively blocking any malicious code from being executed.

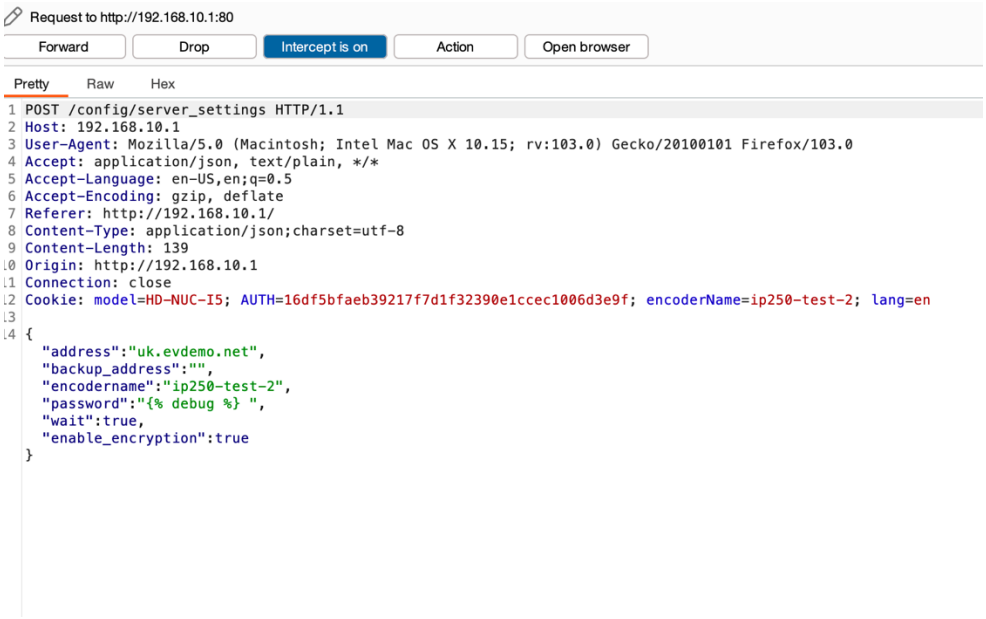


Figure 23. SSTI attempt injected in the password field.

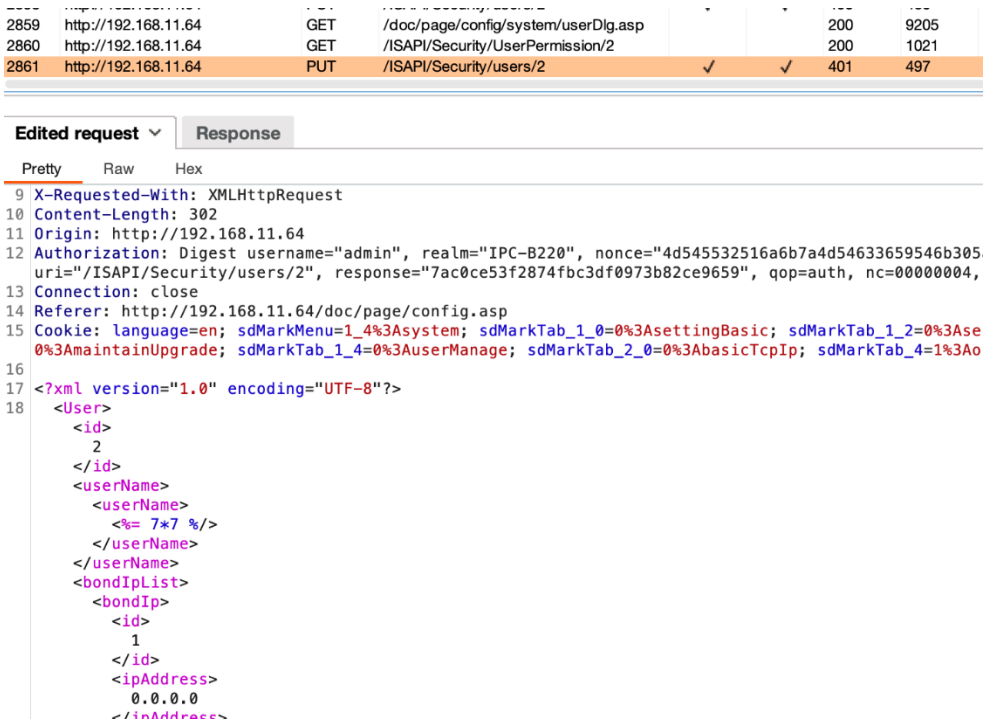


Figure 24. Another SSTI attempt in an alternate coding language -- this attempt failed also.

During this testing, assessors would often come across an input field limitation, preventing user inputs of certain characters. This was an obstacle as testers tried to overcome sanitisation deterrents. ALLENDEVAUX assessors then observed that

only the initial request was preventing certain characters, but if the request was sent and then manipulated in transit, no such restrictions would be compelled. Requests were captured and altered within Burp Suite's repeater to inject dangerous characters in an attempt to leverage a code injection proof of concept; but despite the newly discovered restriction bypass, the application remains appropriately protected from code injection tactics.

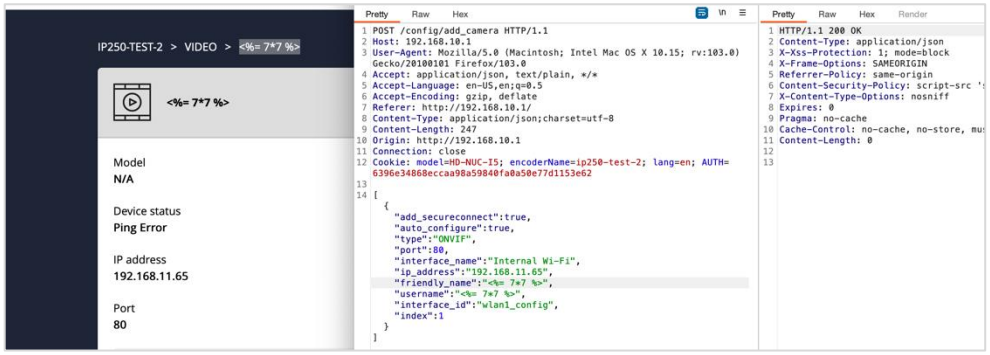


Figure 25. Intercepting a request and altering the username to reflect a template injection POC.

5 CONCLUSION & RECOMMENDATIONS

After performing a grey-box penetration test against the AltusCloud Network Guardian application, ALLENDEVAUX assessors found a few areas that that may be of further interest to the AltusCloud' engineering group. In that context, please see the following recommendations to harden the service further.

5.1 Remove Command Injection Vulnerability (CRITICAL)

ALLENDEVAUX assessors have identified a command injection vulnerability within certain system functions of the AltusCloud's binary, used within the web application. Despite the application having multiple layers of security, it was discovered that these specific system functions could be potentially exploited, given the right circumstances.

Command injection vulnerabilities are a critical risk, posing significant threats to system integrity, confidentiality, and availability. The reason this finding is considered critical lies in the potential damage an attacker can cause if the vulnerability is left unaddressed. An attacker exploiting this vulnerability could, for instance, execute unauthorized commands, create reverse shells, or even gain root access to the system. The severity of these potential consequences demands immediate and comprehensive remediation efforts.

To effectively address the identified command injection vulnerability and protect the application from potential exploitation, the following recommendations are provided:

- Implement measures to sanitize user input;
- Limit the use of system functions and shell invocations;
- Employ least privilege principle;
- Use parameterized APIs; and
- Conduct regular security audits.

By implementing these recommendations, AltusCloud can significantly reduce the risk associated with the command injection vulnerability, safeguarding the system and ensuring its security and resilience. Here is more information across each of these five recommendations:

- Implement measures to sanitize user input: Evaluate all instances where user input is passed to the system function. Implement measures to sanitize these inputs by filtering out hazardous characters or strings that could potentially be used for command injection attacks. This prevents malicious user input from being executed as commands by the system. (OWASP Foundation, 2021)
- Limit the use of system functions and shell invocations: Where possible, replace system functions and shell invocations with safer alternatives. This could involve using specific built-in functions for tasks instead of passing commands to the

system function. By doing so, the application can significantly reduce the possibility of command injection. (OWASP Foundation, 2021)

- Employ the least privilege principle: Ensure that each process within the application runs with the least privileges necessary to perform its function. This could significantly limit the impact of a successful command injection attack, as the malicious commands would be executed with minimal privileges. (NIST SP 800-123, 2008)
- Use parameterized APIs: Use APIs or libraries that support parameterized commands or queries, which can ensure that user input is handled safely, and command injection is prevented. (OWASP Foundation, 2021)
- Conduct regular security audits: Regular security testing, including penetration testing and vulnerability assessments, can help identify and address potential vulnerabilities in the application. This proactive approach can help to uncover and remediate security issues before they can be exploited by attackers. (NIST SP 800-115, 2008)

Addressing the command injection vulnerability within the system functions of AltusCloud's binary is crucial to protecting the integrity and security of the system. By tackling this critical issue, AltusCloud can prevent attackers from exploiting the vulnerability to execute unauthorized commands or gain unauthorized access to the system. Ensuring the security and integrity of the application is essential for maintaining user trust and preserving the company's reputation.

5.2 Review User Input Character Bypass (HIGH)

During the penetration testing of the AltusCloud Network Guardian application, ALLENDEVAUX assessors identified a potential vulnerability related to user input character bypass. Despite the application having robust frontend character restrictions, it was discovered that these restrictions can be bypassed by injecting modifications into the captured HTTP request.

Although a proof-of-concept exploit was not produced for this vulnerability, the potential risks associated with it are significant. If exploited, an attacker could potentially manipulate the application's behaviour or input unexpected values, which could lead to further security issues like Cross-Site Scripting (XSS) or SQL Injection. Hence, this vulnerability is classified as HIGH.

Addressing this vulnerability promptly is crucial due to its potential impact on data integrity, confidentiality, and availability. To effectively mitigate this risk and safeguard the AltusCloud Network Guardian application, the following recommendations are provided:

- Implement a backend sanitisation check;
- Strengthen the frontend input validation;
- Apply output encoding;

- Use parameterized queries; and
- Conduct regular security assessments.

These recommendations, when implemented, can significantly reduce the risk associated with the user input character bypass vulnerability. Here is more detail on each of these five recommendations:

- Implement a backend sanitisation check: Backend sanitisation checks should be implemented in conjunction with frontend JavaScript checks. These checks can prevent the injection of harmful symbols and provide an additional layer of security when frontend validations are bypassed (OWASP Foundation, 2021).
- Strengthen the frontend input validation: Frontend input validation plays a critical role in restricting what users can input. Enhancing these validations can prevent harmful inputs from reaching server-side (Microsoft, 2021).
- Apply output encoding: Output encoding can protect against potential injection attacks by ensuring that any input from users is safe to display. Encoding user inputs can prevent any potentially harmful scripts from executing (CWE, 2019).
- Use parameterized queries: If user input is used in queries, use parameterized queries or prepared statements. This ensures that user input is always treated as literal data and not executable code, reducing the risk of injection attacks (CIS, 2020).
- Conduct regular security assessments: Regular security assessments, including vulnerability assessments and penetration tests, can identify and remediate potential vulnerabilities early. This proactive approach can help mitigate security issues before they can be exploited by attackers (NIST SP 800-115, 2008).

Addressing the User Input Character Bypass vulnerability is essential for maintaining the security of the AltusCloud Network Guardian application. By implementing these recommendations, AltusCloud can protect its systems and data from potential exploits and maintain the integrity, confidentiality, and availability of the application.

5.3 Incorporate TLS for the Application (HIGH)

During the penetration testing of the AltusCloud Network Guardian application, ALLENDEVAUX assessors identified a significant vulnerability related to the lack of secure network communication. The application was found to be using HTTP instead of HTTPS for data transmission, meaning all data exchanged between the application and the client was unencrypted and susceptible to interception.

This HIGH risk vulnerability could potentially allow an attacker with network access to eavesdrop on the data exchange, leading to potential data theft or tampering. This vulnerability poses a significant risk to data confidentiality, integrity, and potentially the availability of the system, depending on the nature of the intercepted data.

Mitigating this vulnerability promptly and effectively is crucial for AltusCloud, especially considering the potential consequences. The following recommendations are provided:

- Implement Transport Layer Security (TLS);
- Enforce HTTPS for all communications;
- Use secure cookies;
- Enable HTTP Strict Transport Security (HSTS); and
- Regularly update and patch your security protocols.

By implementing these measures, AltusCloud can significantly reduce the risk associated with insecure communication. Here are more details on each of these recommendations:

- Implement Transport Layer Security (TLS): TLS is a protocol that ensures encryption and authentication for network connections, protecting all transmitted data from unauthorised access (GlobalSign, 2021).
- Enforce HTTPS for all communications: By enforcing HTTPS for all communications, AltusCloud can ensure that all data transmitted over the network is encrypted, making it much harder for attackers to intercept and read the data (DigiCert, 2020).
- Use secure cookies: Using the 'secure' attribute for cookies ensures that they are only sent over HTTPS, preventing them from being intercepted over an unencrypted HTTP connection (Mozilla, 2021).
- Enable HTTP Strict Transport Security (HSTS): HSTS ensures that the browser only connects to the server via a secure HTTPS connection, even if the user or a link specifies HTTP. This can prevent downgrade attacks where an attacker tries to force a connection to use HTTP instead of HTTPS (Google, 2020).
- Regularly update and patch your security protocols: AltusCloud should continuously monitor for updates and patches for their security protocols to ensure they are utilising the most secure and up-to-date versions (SANS Institute, 2020).

Implementing TLS for the AltusCloud Network Guardian application is not only a best practice but a necessity for maintaining the confidentiality and integrity of the data being transmitted. By addressing this issue, AltusCloud can significantly enhance its security posture and protect its systems and data from potential exploits.

5.4 Update or Remove Vulnerability JavaScript Libraries (MED-HIGH)

During the penetration testing of the AltusCloud Network Guardian application, ALLENDEVAUX assessors identified an issue concerning the usage of outdated JavaScript libraries with known vulnerabilities. The host for Network Guardian, http://192.168.10.1, was found to have incorporated these libraries, which could potentially expose the application to security threats.

This vulnerability is rated as MED-HIGH, indicating the importance of addressing it to maintain the application's overall security. To mitigate this risk, it is recommended that AltusCloud either updates or removes these vulnerable JavaScript libraries from the production environment.

The following table outlines the current JavaScript library version used and the suggested upgrade:

Host	Current JS Library & Version	Recommended Upgrade
http://192.167.10.1/#/login	Angular 1.5.5	≥ Angular 1.8.0

To address this issue effectively and protect the AltusCloud Network Guardian application from potential exploits, the following recommendations are provided:

- Update the outdated JavaScript libraries;
- Regularly monitor for library updates;
- Remove unnecessary libraries;
- Implement security headers; and
- Conduct ongoing security assessments.

Implementing these recommendations can significantly reduce the risk associated with outdated JavaScript libraries. Here are more details on each of these recommendations:

- Update the outdated JavaScript libraries: Upgrade the vulnerable JavaScript libraries to their latest stable versions or, if possible, switch to more secure alternatives (OWASP, 2021).
- Regularly monitor for library updates: Continuously check for updates and patches for JavaScript libraries in use, ensuring that the most recent and secure versions are always implemented (Snyk, 2020).
- Remove unnecessary libraries: Evaluate the application's dependencies and remove any libraries that are not essential to its functionality. This reduces the potential attack surface for an adversary (NPM, 2021).
- Implement security headers: Utilize security headers, such as Content Security Policy (CSP), to restrict the execution of JavaScript from untrusted sources and reduce the risk of Cross-Site Scripting (XSS) attacks (Mozilla, 2021).
- Conduct ongoing security assessments: Perform regular security assessments, including vulnerability assessments and penetration tests, to identify and remediate potential vulnerabilities in the application (NIST SP 800-115, 2008).

By addressing the outdated JavaScript libraries issue, AltusCloud can significantly enhance the security posture of the Network Guardian application, protecting it from potential exploits that could compromise its confidentiality, integrity, and availability.

5.5 Remove Sensitive Information Exposure (MED-HIGH)

During the penetration testing of the AltusCloud Network Guardian application, the assessment team identified two instances of potential information disclosure. These vulnerabilities can provide attackers with unnecessary insight into the application, potentially enabling harmful activities.

The first vulnerability was related to a template page found to contain potentially sensitive information. While this information was not considered critical, its presence contradicts best practice guidelines for secure data management. It's important to ensure that only necessary data is stored and available within the system.

The second vulnerability was tied to inappropriate 403 HTTP responses. Instead of correctly restricting access, these inadequate responses exposed the web application's JavaScript files to the evaluators. Such exposure could potentially allow attackers to access sensitive data or inject malicious code into the system.

This vulnerability is a serious concern because it could potentially allow unauthorized access to sensitive system data or provide an entry point for further attacks. Therefore, it's imperative to address these issues promptly to maintain the application's overall security.

To mitigate the risk associated with these vulnerabilities, the following recommendations are made:

Review and remove the identified template page;

- Correct the HTTP response codes;
- Implement Content Security Policy (CSP);
- Regularly conduct security reviews; and
- Implement secure coding practices.

These recommendations are further explained as follows:

- Review and remove the identified template page: AltusCloud should scrutinize the identified page for any sensitive information and delete or modify it as necessary (OWASP, 2021).
- Correct the HTTP response codes: Proper implementation of HTTP response codes can prevent unauthorized access and exposure of sensitive files (Microsoft, 2021).

- Implement Content Security Policy (CSP): CSP can help to prevent Cross-Site Scripting (XSS) attacks by controlling which resources the browser is allowed to load (Mozilla, 2021).
- Regularly conduct security reviews: Regular security assessments can help identify and rectify potential vulnerabilities before they can be exploited by attackers (NIST SP 800-115, 2008).
- Implement secure coding practices: Adhering to secure coding practices can prevent vulnerabilities from being introduced into the system in the first place (SANS Institute, 2020).

Addressing these information exposure vulnerabilities is crucial for maintaining the security and integrity of the AltusCloud Network Guardian application. Implementing the above recommendations can significantly enhance the application's security posture, protecting it from potential attacks.

5.6 Prevent Clear Text Displays of Data in Transport (MED-HIGH)

During the penetration testing of AltusCloud's Network Guardian application, the testing team observed that the application did not sufficiently protect data during transit. This finding indicates that private information, including usernames, passwords, and other sensitive data, was being sent across networks without the appropriate security measures. This unencrypted transmission of data opens up the possibility for unauthorized interception and access. The presence of such a vulnerability presents a significant security threat to the application and its users and could lead to data breaches and exposure of confidential information.

To mitigate this threat, AltusCloud is strongly recommended to incorporate security mechanisms that deter the visibility of data in plaintext during transit. These measures could include data encoding, encryption, or obfuscation, which can help to ensure that information remains confidential even if intercepted.

The following strategies are suggested for remediation:

- Adopt secure communication protocols, such as TLS or HTTPS;
- Implement application-level data encryption;
- Regularly conduct security audits; and
- Train employees on secure coding practices.

Here's a more detailed explanation for each strategy:

- Adopt secure communication protocols, such as TLS or HTTPS: These protocols ensure that all data transmitted between the client and server is encrypted and authenticated, significantly reducing the risk of data interception (NIST Special Publication 800-52 Rev. 2, 2019).

- Implement application-level data encryption: This involves encrypting data before it is transmitted, adding an extra layer of security and ensuring that even if data is intercepted, it cannot be understood without the decryption key (NIST Special Publication 800-38A, 2001).
- Regularly conduct security audits: Regular audits can help identify potential vulnerabilities and assess the effectiveness of current security measures, allowing for continuous improvement of security (NIST Special Publication 800-53A, 2014).
- Train employees on secure coding practices: Ensuring that those who work on the application are familiar with secure coding practices can prevent the introduction of vulnerabilities into the system (SANS Institute, 2020).

Addressing this data exposure vulnerability is essential for maintaining the security and integrity of the AltusCloud Network Guardian application. By adopting these recommendations, AltusCloud can significantly enhance its security posture and better protect its users' data.

5.7 Review Use of the “strcpy” Function (INFO)

The assessment process revealed that the application's codebase includes the use of the strcpy function. Historically, this function has been associated with buffer overflow vulnerabilities, which could potentially be leveraged by malicious actors to manipulate the application and run arbitrary code.

Although the assessment did not culminate in an actual exploit, it is nonetheless critical that this function be scrutinized. If feasible, it should be substituted with safer alternatives like strncpy or memcpy. Prioritizing secure coding practices is paramount to safeguard against potential security breaches. A comprehensive code review is also advised to detect any other similar functions or coding practices that could potentially expose the system to security risks. By proactively addressing these vulnerabilities, the overall security robustness of the application can be enhanced, and potential exploits can be thwarted.

Below are the detailed recommendations:

- Substitute strcpy with safer alternatives: Replacing strcpy with safer alternatives such as strncpy or memcpy can significantly reduce the risk of buffer overflow vulnerabilities (CWE-120, 2020).
- Prioritize secure coding practices: Adopting secure coding practices can help prevent the introduction of vulnerabilities in the codebase (SANS Institute, 2020).
- Conduct a comprehensive code review: This helps in identifying potential vulnerabilities in the code and fixing them before they can be exploited by attackers (OWASP Code Review Guide, 2021).

By actively addressing these issues, AltusCloud can significantly enhance the security of the Network Guardian application and mitigate potential threats.

6 REFERENCES

Hout, v. d. (2019). *Standardised Penetration Testing? Examining the Usefulness of Current Penetration Testing Methodologies.* London: University Of London.

Infosec Institute. (2019, September 4). The Types of Penetration Testing. *InfoSec Institute*. Retrieved from https://resources.infosecinstitute.com/the-types-of-penetration-testing/

Northcutt, S., Shenk, J., Shackleford, D., Rosenberg, T., Silas, R., & Mancini, S. (2006, June). Penetration Testing: Assessing Your Overall Security Before Attackers Do. *SANS Institute InfoSec Reading Room*. Retrieved from https://www.sans.org/reading-room/whitepapers/analyst/penetration-testing-assessing-security-attackers-34635

PTES. (2014). PTES Technical Guide. Retrieved October 21, 2022, from http://www.pentest-standard.org/index.php/Main_Page

Redscan. (2020, October). *Types of pen testing: white box, black box and everything in between.* Retrieved from Redscan: https://www.redscan.com/news/types-of-pen-testing-white-box-black-box-and-everything-in-between/

7 SUPPLEMENTAL REPORTS

The ALLENDEVAUX pentesting team may have provided supplemental reports within a secure repository which include additional analysis regarding any assessment findings. The details may include an inventory of digital assets followed by specific tests performed, the protocol results returned, any weaknesses found in systems, and guidance to address the weaknesses. This information should not be given to partners or customers as it contains private and public IP addresses, vulnerability information, port information, and other confidential data.

To access the secure repository, please contact your account representative at ALLENDEVAUX & COMPANY, or send an email to infosec@allendevaux.com; state your request in the email and it will be routed to the right individual.

APPENDIX A

Web App	Title	SEV.	OWASP:21	CWE	Context
http://192.168.10.1 Network Guardian	Command Injection	5	A3	CWE-78	REQUEST POST /config/set_settings HTTP/1.1 Host: 192.168.10.1 Content-Length: 93 Accept: application/json, text/plain, */* User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.178 Safari/537.36 Content-Type: application/json;charset=UTF-8 Origin: http://192.168.10.1 Referer: http://192.168.10.1/ Accept-Encoding: gzip, deflate Accept-Language: en-US,en;q=0.9 Cookie: lang=en; AUTH=d900eae5c11d2407dc7f4a328bef72a3d97c4aa7 Connection: close {"sftp_user":"user","sftp_password":"Pwo!\" ; touch /tmp/hack3 ; echo \"","sftp_enable":true} RESPONSE HTTP/1.1 200 OK Content-Type: application/json X-Xss-Protection: 1; mode=block X-Frame-Options: SAMEORIGIN Referrer-Policy: same-origin Content-Security-Policy: script-src 'self' X-Content-Type-Options: nosniff Expires: 0 Pragma: no-cache Cache-Control: no-cache, no-store, must-revalidate Content-Length: 0
http://192.168.10.1 Network Guardian	User Input Restriction Bypass	4	A4	CWE-20	REQUEST POST /config/add_camera HTTP/1.1 Host: 192.168.10.1 User-Agent: Mozilla/ 5.0 (Macintosh: Intel Mac OS X 10.15; rv:103.0) Gecko/20100101 Firefox/ 103.0 Accept: application/json, text/plain, */* Accept-Language: en-US, en;q=0.5 Accept-Encoding: gzip, deflate Referer: http://192.168.10.1/ Content-Type: application/json;charset=utf-8 Content-Length 247 Origin: http://192.168.10.1 Connection: close Cookie: lang=en; AUTH=6396e34868eccaa98a59840fa0a50e77d1153e62 [{ "add_secureconnect":true, “auto_configure”:true, "type": "ONVIF, "port": 80, "interface_name":"Internal Wi-Fi",

					'ip_address": "192.168.11.65", "friendly_name": "<%= 7×7 %>" "username": "<%= 7×7 %>", "interface id":"wlan1 config", "index":1 }] RESPONSE HTTP/1.1 200 OK Content-Type: application/json X-Xss-Protection: 1; mode=block X-Frame-Options: SAMEORIGIN Referrer-Policy: same-origin Content-Security-Policy: script-src 'self' X-Content-Type-Options: nosniff Expires: 0 Pragma: no-cache Cache-Control: no-cache, no-store, must-revalidate Content-Length: 0
http://192.168.10.1 Network Guardian	Absence of TLS	4	A2	CWE-319	REQUEST GET http://192.168.10.1/ Referer: http://192.168.10.1/ Cookie: lang=en; AUTH=cd01fcce9c0fab107491d749a8cf366afa827366; Host: 192.168.10.1 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15 Accept: */* RESPONSE HTTP/1.1 200 OK Date: Sat, 04 Mar 2023 01:57:44 GMT Last-Modified: Tue, 05 Apr 2011 23:00:00 GMT Etag: "4d9b9ef0.3329" Content-Type: text/html Connection: keep-alive Cache-Control: max-age=5 Strict-Transport-Security: max-age=15552000 Content-Security-Policy: script-src 'self' X-Frame-Options: SAMEORIGIN X-Xss-Protection: 1; mode=block X-Content-Type-Options: nosniff Referrer-Policy: same-origin Content-Length: 3329 Accept-Ranges: bytes Set-Cookie: encoderName=ip250-test-2; domain=192.168.10.1; path=/ Set-Cookie: lang=en; domain=192.168.10.1; path=/ Set-Cookie: model=HD-NUC-I5; domain=192.168.10.1; path=/ Set-Cookie: AUTH=cd01fcce9c0fab107491d749a8cf366afa827366; domain=192.168.10.1; path=/ <!DOCTYPE html><!--[if IE 8]> <html class="no-js lt-ie10 lt-ie9"><![endif]--><!--[if IE 9]> <html class="lt-ie10"><![endif]--><!--[if gt IE 9]><!--><html id="html"><!--<![endif]--><head><st...

http://192.168.10.1 Network Guardian	Vulnerable JavaScript Libraries	3	A6	CWE-937	REQUEST GET http://192.168.10.1/#/login Host: 192.168.10.1 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15 Accept: */* RESPONSE Vulnerable javascript library: Angular version: 1.5.5 Details: In angular versions below 1.6.5 both Firefox and Safari are vulnerable to XSS in $sanitise if an inert document creat ---------- In angular versions below 1.7.9, Object prototype can be polluted using a __proto__ payload with merge() function. In angular version 1.7.9, __proto__ is blocked on deep merging to prevent Object prototype from being polluted. Please refer to vendor documentation (https://github.com/angular/angular.js/commit/726f49dcf6c23106ddaf5cfd5e2e592841db743a, https://github.com/angular/angular.js/blob/master/CHANGELOG.md#179-pollution-eradication-2019-11-19) for latest security updates. ---------- angular.js prior to 1.8.0 allows cross site scripting. The regex-based input HTML replacement may turn sanitised code into unsanitised one. Wrapping "<option>" elements in "<select>" ones changes parsing behavior, leading to p...
http://192.168.10.1 Network Guardian	Remove Sensitive Information Exposure	3	A5	CWE-200	-
http://192.168.10.1 Network Guardian	Prevent Clear Text Data in Transit	3	A2	CWE-319	REQUEST GET /config/camera_settings HTTP/1.1 Host: 192.168.10.1 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:103.0) Gecko/20100101 Firefox/103.0 Accept: application/json, text/plain, */* Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Referer: http://192.168.10.1/ Connection: close Cookie: AUTH=16df5bfaeb39217f7d1f32390e1ccec1006d3e9f; lang=en

					RESPONSE HTTP/1.1 200 OK Content-Type: application/json X-Xss-Protection: 1; mode=block X-Frame-Options: SAMEORIGIN Referrer-Policy: same-origin Content-Security-Policy: script-src 'self' X-Content-Type-Options: nosniff Expires: 0 Pragma: no-cache Cache-Control: no-cache, no-store, must-revalidate Content-Length: 1240 {"layouts":[],"licensed_channels":0,"max_channels":16,"feeds":[{"details":{"interface_name":"Ethernet: LAN 2”,"ip_address":"192.168.11.64","auto_configure":true,"alarm_source_enabled":false,"password":"REDACTED","make":"HIKVISION","friendly_name":"cam","alarm_source_port":4444, hardware_id":"88","analytics_type":"","auto_configure_bandwidth":3000,"index":0,"username":"admin","type":"ONVIF","ptz":{"enabled":false},"port":80,"add_secureconnect":true,"transport":"tcp","watchdog_monitor":false},"config":{"enabled":true,"rtsp_url":"rtsp://192.168.11.64:554/Streaming/Channels/101?transportmode=unicast&profile=Profile_1","ovf_audio_token":"","ptz_url":"","img_url":"http://192.168.11.64/onvif/Imaging","ovf_media_token":"Profile_1","media_url":"http://192.168.11.64/onvif/Media","dev_url":"http://192.168.11.64/onvif/device_service","ovf_audio_source_token":"","ovf_source_token":"VideoSource_1","ovf_ptz_config_token":"","ovf_ptz_presets":"","read_only":false},"stream":{"status":"Streaming","video_resolution":"1920 x 1080","video_avg_fps":"25.0","video_avg_bitrate":2625712,"video_codec":"H264"}}]}

Thank You!

Contact Information

E: pentesting@allendevaux.com

W: www.allendevaux.com

ALLENDEVAUX
& COMPANY