Digital Policy Office
The Government of the Hong Kong Special Administrative Region
of the People's Republic of China

HKCERT Hong Kong Computer Emergency Response Team Coordination Centre

政府資助計劃 Government-funded programme

THE IoT SECURITY STUDY REPORT ON DIGITAL SIGNAGE

Disclaimer

The Hong Kong Computer Emergency Response Team Coordination Centre (HKCERT) of the Hong Kong Productivity Council (HKPC) reserves the right to amend the document from time to time without prior notice.

While every attempt has been made to ensure that the information contained in this document is obtained from reliable sources, HKCERT is not responsible for any errors or omissions, or for the results obtained from the use of this information. All information in this document is provided "as is", with no guarantee of completeness, accuracy, timeliness or of the results obtained from the use of this information, and without warranty of any kind, express or implied, including, but not limited to warranties of performance, merchantability and fitness for a particular purpose.

The information contained in this document is intended to provide general information and for reference only. Reliance or use of this information shall be at the reader's own risk. Nothing herein shall to any extent substitute for the independent investigations and the sound technical and business judgment of the reader. In no event will HKCERT, HKPC or its partners, employees or agents, be liable to you or anyone else for any decision made or action taken in reliance on the information in this document, or for any consequential, special or similar damages, even if advised of the possibility of such damages.

License

Table of Contents

1 Introduction

Digital signage is popular among industries for promoting products and displaying information to customers. As an IoT device, it can be targeted by hackers for cyber attacks. Therefore, HKCERT has conducted security study on eight digital signages. The results and observations are published along with security recommendations for the general public and digital signage users.

The goal of the study is to identify potential vulnerabilities associated with common digital signage systems. The security study was conducted in October 2024. The details of the vulnerability findings, and recommendations are documented in this report.

The objectives of this security study are:

- Conduct security tests on the selected digital signages and its client-side application and web management portal application
- Identify security risks in the selected digital signages and web management portals
- Recommend safeguards to mitigate the identified risks

2 Security Test Methodologies and Findings Summary

2.1 Security Test Methodologies

Digital signage systems are usually set up in wireless networks, with signage devices often running Android or Windows operating systems. Users can update the display content by accessing the signage or content management system (CMS) to upload media, adjust screen settings, and setup schedule etc. After the content is saved in the management system, it is sent to the signage device for display. A diagram typically illustrates this content update process from the user to the signage device.

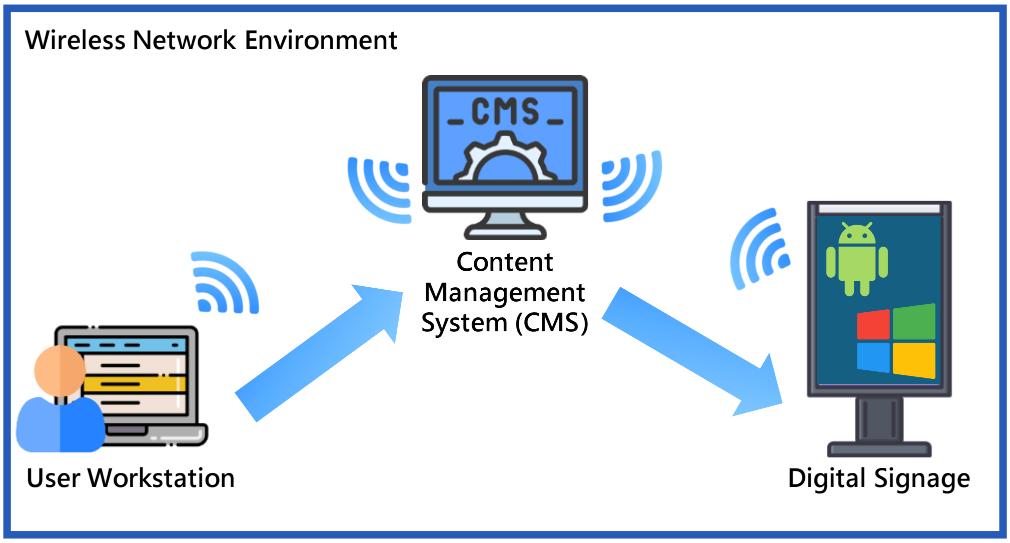


Figure 1 - Content to be deployed from user workstation to a signage device

This study contains testing results on four different brands of digital signage with both Windows and Android Operating System (OS), in total eight devices and the corresponding web management portal respectively. Security tests had been carried out on the selected digital signages shown in the table below:

Brand	**Operating System**	**Digital Signage Device Test ID**	**Signage Web Management Portal***	**Signage Web Management Portal Test ID**
A	Windows	**A1**	N/A	N/A
	Android	**A2**		
B	Windows	**B1**	Yes	**B1P**
	Android	**B2**	Yes	**B2P**

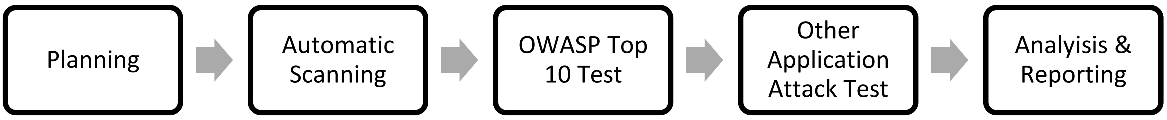
C	Windows	**C1**	Yes	**CP**
	Android	**C2**		
D	Windows	**D1**	Yes	**DP**
	Android	**D2**		

Table 2-1. Selected Digital Signages for Security Test

* *Brand A digital signage solution does not include a web management portal for security test;* Brand B has different web management portal for Windows and Android OS signages respectively.

A grey box approach was used in this security test. The security test was provided with the network environments and IP addresses of the digital signages, as well as the credentials to authenticate to the web management portal.

The security test methodology process is illustrated as follows:



2.2 Findings Summary

This section summarised the findings identified in the security test. The following table summarise them according to their risk level. The findings have been identified with OWASP Top 10 [1] and OWASP IoT Top 10 [2].

Risk Level	**Total Number of Findings**	Number of Findings on Signage Web Management Portals	Number of Findings on Digital Signage Devices
High	**10**	5	5
Medium	**6**	4	2
Low	**4**	2	2
Total	**20**	11	9

Table 2-2. Finding Summary

A total of 20 findings were found in the security test. 11 of them are risk findings identified in the signage web management portals and 9 risk findings identified in the digital signage devices.

3 Security Test Risk Ratings and Definitions

The risk items identified during the security test were analysed in terms of their impact and likelihood. They will be assigned a risk level as illustrated in the following risk rating table:

Risk		Likelihood		
		High	Medium	Low
Impact	High	High	High	Medium
	Medium	High	Medium	Low
	Low	Medium	Low	Low

Table 3-1. Risk Rating Table

The following tables summarise the definitions of the risk impact and likelihood levels:

Impact	Descriptions
High	The host can be compromised by exploiting the vulnerability or the data/service/user may be serious affected.
Medium	The vulnerability alone may not lead to direct compromise of the host. However, when used in combination with other vulnerabilities or with certain prerequisites met, it is possible to directly/indirectly lead to fully/partially compromise of the system/data/user's security.
Low	The service/data/user may be affected by the vulnerability but it's not fatal nor significant.

Table 3-2. Risk Impact Definitions

Likelihood	Descriptions
High	Easy access to the attack surface or exploit codes/tools are readily available.
Medium	Limited access to attack surface or require in-depth knowledge, specialised skills or knowledge to exploit.
Low	Limited access to the attack surface. Exploitation is only feasible when certain prerequisites are met or mainly theoretical.

Table 3-3. Risk Likelihood Definitions

4 Findings on Signage Web Management Portals

4.1 Summary of Findings

The following tables summarise the number of risk issues identified in the security test.

Finding ID	Description	Risk
IoT-WEB-01	Sensitive Information Disclosure	High
IoT-WEB-02	Insecure Password Hash	High
IoT-WEB-03	Outdated Software Libraries	High
IoT-WEB-04	SQL Injection	High
IoT-WEB-05	Broken Access Control	High
IoT-WEB-06	Client-Side Validation Bypass	Medium
IoT-WEB-07	Cross-Site Scripting	Medium
IoT-WEB-08	Session Fixation	Medium
IoT-WEB-09	Files Accessible Without Authentication	Medium
IoT-WEB-10	Changing Password does not Require Re-authentication	Low
IoT-WEB-11	Insecure HTTP Usage	Low

Table 4-1. Finding List – Signage Web Management Portals

	*Signage Web Management Portal Test ID **			
Finding ID	***B1P***	***B2P***	***CP***	***DP***
IoT-WEB-01	-	-	-	Affected
IoT-WEB-02	-	Affected	-	Affected
IoT-WEB-03	Affected	Affected	Affected	Affected
IoT-WEB-04	-	-	-	Affected
IoT-WEB-05	-	-	-	Affected
IoT-WEB-06	-	-	-	Affected
IoT-WEB-07	-	Affected	-	-
IoT-WEB-08	Affected	-	Affected	-
IoT-WEB-09	Affected	-	Affected	-
IoT-WEB-10	-	Affected	-	Affected
IoT-WEB-11	Affected	Affected	Affected	Affected

Table 4-2. Vulnerability Matrix – Signage Web Management Portals

* “-“ means not affected.

4.2 Detailed Findings on Signage Web Management Portals

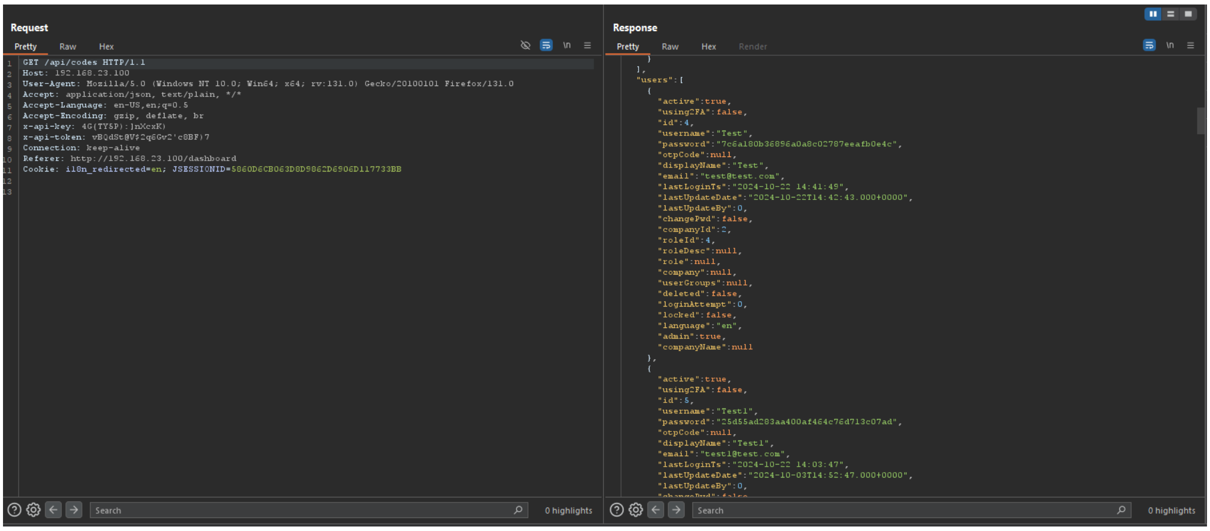
4.2.1 High Risk Findings

4.2.1.1 IoT-WEB-01: Sensitive Information Disclosure

Risk level	High
Affected Test ID	DP
OWASP Top 10	A01:2021 - Broken Access Control A05:2021 - Security Misconfiguration

Details

This finding reveals insufficient access controls, allowing any user to access the endpoint and retrieve sensitive information. The exposed endpoint lets users view confidential data, including user lists, passwords, roles, and other sensitive details. This vulnerability can result in serious security issues, such as user impersonation, account takeovers, and wider system compromises.



```
GET /api/codes HTTP/1.1
Host: 192.168.23.100
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
x-api-key: 4G(TY5P):]nXcxK)
x-api-token: vBQdSt@V$2q6Gv2'c8BF)7
Connection: keep-alive
Referer: http://192.168.23.100/dashboard
Cookie: i18n_redirected=en; JSESSIONID=5860D6CB063D8D9862D6906D117733BB
```

```
    }
  ],
  "users":[
    {
      "active":true,
      "using2FA":false,
      "id":4,
      "username":"Test",
      "password":"7c6a180b36896a0a8c02787eeafb0e4c",
      "otpCode":null,
      "displayName":"Test",
      "email":"test@test.com",
      "lastLoginTs":"2024-10-22 14:41:49",
      "lastUpdateDate":"2024-10-22T14:42:43.000+0000",
      "lastUpdateBy":0,
      "changePwd":false,
      "companyId":2,
      "roleId":4,
      "roleDesc":null,
      "role":null,
      "company":null,
      "userGroups":null,
      "deleted":false,
      "loginAttempt":0,
      "locked":false,
      "language":"en",
      "admin":true,
      "companyName":null
    },
    {
      "active":true,
      "using2FA":false,
      "id":5,
      "username":"Test1",
      "password":"25d55ad283aa400af464c76d713c07ad",
      "otpCode":null,
      "displayName":"Test1",
      "email":"test1@test.com",
      "lastLoginTs":"2024-10-22 14:03:47",
      "lastUpdateDate":"2024-10-03T14:52:47.000+0000",
      "lastUpdateBy":0,
      "changePwd":false
```

Figure 2 - List all users, their password hashes, and other sensitive data

4.2.1.2 IoT-WEB-02: Insecure Password Hash

Risk level	High
Affected Test ID	B2P, DP
OWASP Top 10	A02:2021 - Cryptographic Failures

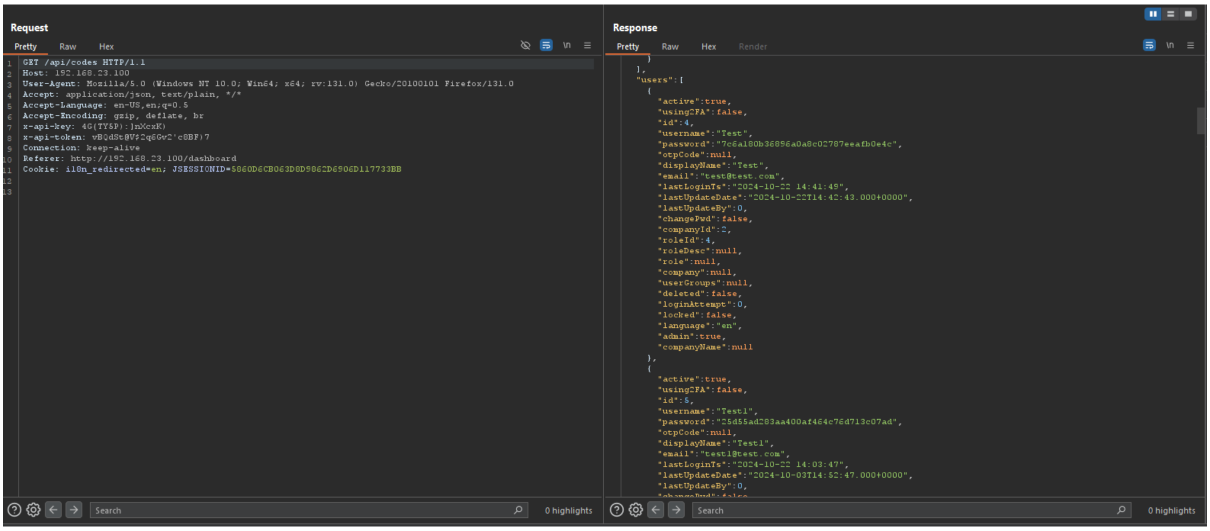
Details

The portals use a MD5 as the password hashing algorithm, which is not a suitable hashing algorithm for passwords by modern standards. Moreover, the affected portals did not use a password salt as part of the input to the hash function. This makes it easier for attackers to crack the hash value.

For example, the password hash for user "Test" was: "25d55ad283aa400af464c76d713c07ad". As cryptographic salts were not used in the MD5 hashes and the password was weak, attacker could recover the password easily by performing lookup on public hashes databases. The password for the "Test" account was "12345678".

This indicated that the password hashes were simple MD5 hashes without proper cryptographical salts (Password hash=MD5(password)). Unsalted MD5 hashes are known to be vulnerable to various types of attacks (e.g. precomputation password attacks such as rainbow table, which generate a list of known passwords and corresponding MD5 hash and store in a database for future lookup). Attackers who have access to the password hashes could recover the plaintext password from hashes with little effort.

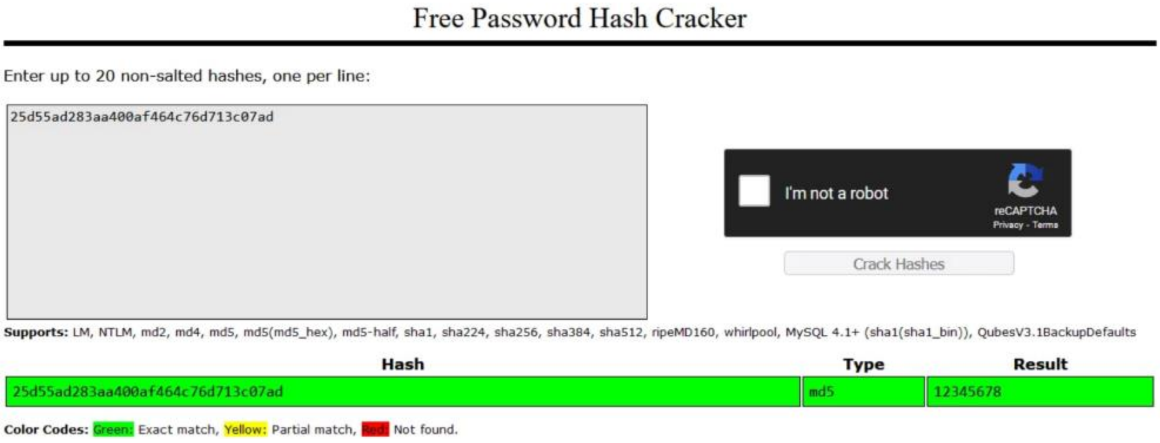
Even being used with a proper salt, MD5 is no longer considered as a strong hashing algorithm for passwords. For example, the “IT Security Guideline [G3]” of HKSAR government require that at least SHA-2 should be used for password hashing purposes for user passwords. MD5 is considered weaker than SHA-2.



```
GET /api/codes HTTP/1.1
Host: 192.168.23.100
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
x-api-key: 4G{TY5P):]nXcxK)
x-api-token: vBQdSt@V$2q6Gv2'c8BF)7
Connection: keep-alive
Referer: http://192.168.23.100/dashboard
Cookie: i18n_redirected=en; JSESSIONID=5860D6CB063D8D9862D6906D117733BB
```

```
      }
    ],
    "users":[
      {
        "active":true,
        "using2FA":false,
        "id":4,
        "username":"Test",
        "password":"7c6a180b36896a0a8c02787eeafb0e4c",
        "otpCode":null,
        "displayName":"Test",
        "email":"test@test.com",
        "lastLoginTs":"2024-10-22 14:41:49",
        "lastUpdateDate":"2024-10-22T14:42:43.000+0000",
        "lastUpdateBy":0,
        "changePwd":false,
        "companyId":2,
        "roleId":4,
        "roleDesc":null,
        "role":null,
        "company":null,
        "userGroups":null,
        "deleted":false,
        "loginAttempt":0,
        "locked":false,
        "language":"en",
        "admin":true,
        "companyName":null
      },
      {
        "active":true,
        "using2FA":false,
        "id":5,
        "username":"Test1",
        "password":"25d55ad283aa400af464c76d713c07ad",
        "otpCode":null,
        "displayName":"Test1",
        "email":"test1@test.com",
        "lastLoginTs":"2024-10-22 14:03:47",
        "lastUpdateDate":"2024-10-03T14:52:47.000+0000",
        "lastUpdateBy":0,
```

Figure 3 - An Unsalted MD5 hash has been detected



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
25d55ad283aa400af464c76d713c07ad
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
25d55ad283aa400af464c76d713c07ad	md5	12345678

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Figure 4 - 25d55ad283aa400af464c76d713c07ad > 12345678

4.2.1.3 *IoT-WEB-03: Outdated Software Libraries*

Risk level	High
Affected Test ID	B1P, B2P, CP, DP
OWASP Top 10	A06:2021 - Vulnerable and Outdated Components

Details

B1P, B2P, CP and DP portals were detected using outdated software libraries with known vulnerabilities. This poses some security risk as these vulnerabilities can be exploited by attackers.

The vulnerable software libraries are as follows:

Affected Portal ID	**Version**	**Known Vulnerabilities**	**Highest Risk Level (CVSS Score)**
DP	JavaScript: bootstrap 4.4.1	CVE-2024-6531 CVE-2024-6484	Medium (5.9)
B2P	JavaScript: jquery 1.8.3	CVE-2020-7656 CVE-2020-11022 CVE-2020-11023 CVE-2019-11358 CVE-2015-9251 CVE-2012-6708	Medium (6.5)
B1P CP	JavaScript: ExtJS 4.1.1.1	CVE-2007-2285 CVE-2018-9046	High (7.8)

Table 4-3. Vulnerable Libraries

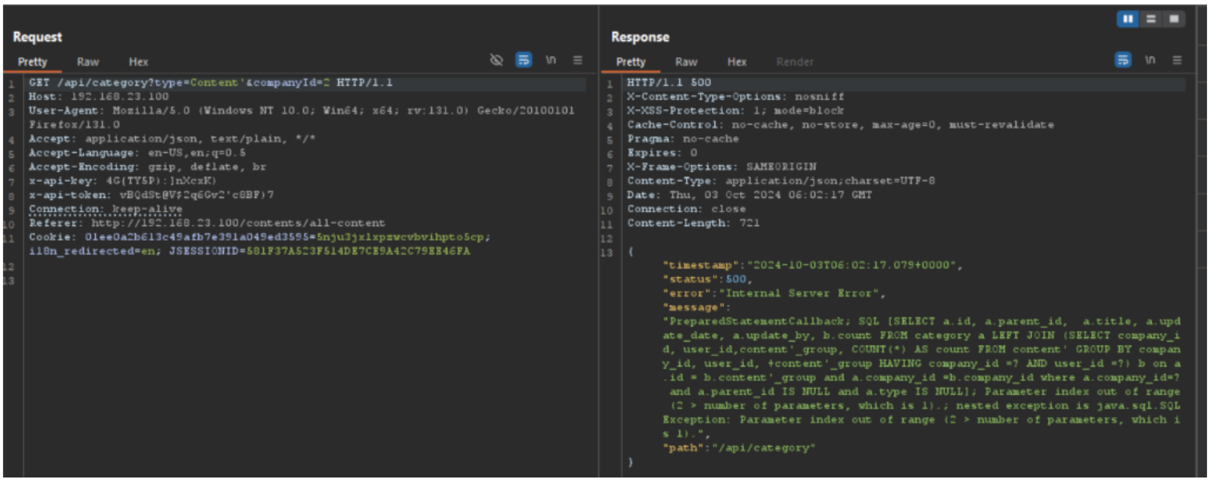
4.2.1.4 IoT-WEB-04: SQL Injection

Risk level	High
Affected Test ID	DP
OWASP Top 10	A03:2021 - Injection

Details

DP portal fails to properly sanitise user input before concatenating it into SQL queries. When a single quote is entered in the input fields, it triggers a SQL error, suggesting that the input is directly concatenated in the SQL query without adequate escaping or parameterization, leading to a broken SQL syntax.

Although the input resulted in a SQL error, no full proof-of-concept (PoC) exploit that would allow for data extraction or command execution was successful during the security test. This may be due to the use of "PreparedStatement" call-backs in the database interaction, which generally mitigates SQL Injection risks by securely handling some input parameters. However, some values were still concatenated to the SQL statement without proper input sanitization and validation, making the portal potentially vulnerable to attacks. The user-controlled input values were used multiple times in the SQL statement, making it harder to construct a valid statement to be executed.



```
GET /api/category?type=Content'&companyId=2 HTTP/1.1
Host: 192.168.23.100
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101
Firefox/131.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
x-api-key: 4G(TY5P):]nXcxK)
x-api-token: vBQdSt@V$2q6Gv2'c8BF)7
Connection: keep-alive
Referer: http://192.168.23.100/contents/all-content
Cookie: 01ee0a2b613c49afb7e391a049ed3595=5nju3jxlxpzwcvbvihpto5cp;
i18n_redirected=en; JSESSIONID=581F37A523F514DE7CE9A42C79EE46FA
```

```
HTTP/1.1 500
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: SAMEORIGIN
Content-Type: application/json;charset=UTF-8
Date: Thu, 03 Oct 2024 06:02:17 GMT
Connection: close
Content-Length: 721

{
    "timestamp":"2024-10-03T06:02:17.079+0000",
    "status":500,
    "error":"Internal Server Error",
    "message":
    "PreparedStatementCallback; SQL [SELECT a.id, a.parent_id,  a.title, a.upd
    ate_date, a.update_by, b.count FROM category a LEFT JOIN (SELECT company_i
    d, user_id,content'_group, COUNT(*) AS count FROM content' GROUP BY compan
    y_id, user_id, +content'_group HAVING company_id =? AND user_id =?) b on a
    .id = b.content'_group and a.company_id =b.company_id where a.company_id=?
     and a.parent_id IS NULL and a.type IS NULL]; Parameter index out of range
     (2 > number of parameters, which is 1).; nested exception is java.sql.SQL
    Exception: Parameter index out of range (2 > number of parameters, which i
    s 1).",
    "path":"/api/category"
}
```

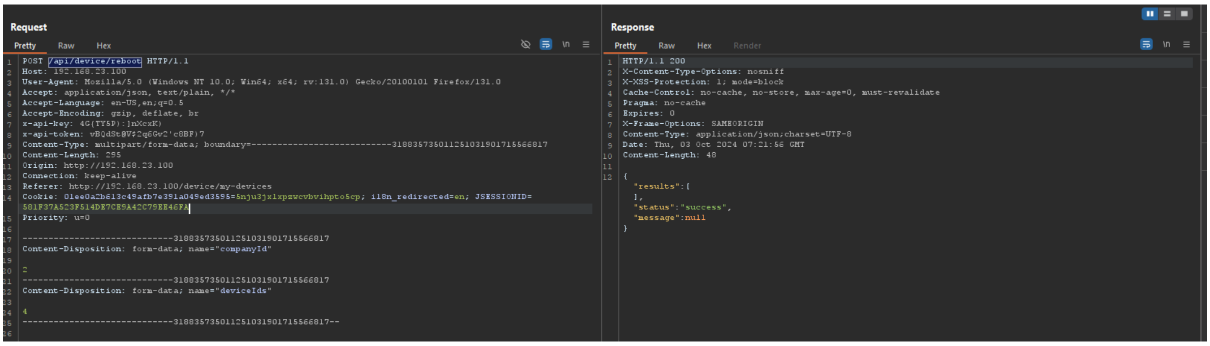
Figure 5 – SQL error

4.2.1.5 IoT-WEB-05: Broken Access Control

Risk level	High
Affected Test ID	DP
OWASP Top 10	A01:2021 - Broken Access Control

Details

In the portal, it was observed that a regular user account, which should not have privileged access, is able to reboot a device. This indicates a failure in enforcing role-based access control, allowing users with insufficient privileges to perform critical system operations. This can result in service downtime, disrupting operations and affecting other users.



```
POST /api/device/reboot HTTP/1.1
Host: 192.168.23.100
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
x-api-key: 4G(TY5P):]nXcxK)
x-api-token: vBQdSt@V$2q6Gv2'c8BF)7
Content-Type: multipart/form-data; boundary=---------------------------318835735011251031901715566817
Content-Length: 295
Origin: http://192.168.23.100
Connection: keep-alive
Referer: http://192.168.23.100/device/my-devices
Cookie: 01ee0a2b613c49afb7e391a049ed3595=5nju3jxlxpzwcvbvihpto5cp; i18n_redirected=en; JSESSIONID=
581F37A523F514DE7CE9A42C79EE46FA
Priority: u=0

-----------------------------318835735011251031901715566817
Content-Disposition: form-data; name="companyId"

2
-----------------------------318835735011251031901715566817
Content-Disposition: form-data; name="deviceIds"

4
-----------------------------318835735011251031901715566817--
```

```
HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: SAMEORIGIN
Content-Type: application/json;charset=UTF-8
Date: Thu, 03 Oct 2024 07:21:56 GMT
Content-Length: 48

{
  "results":[
  ],
  "status":"success",
  "message":null
}
```

Figure 6 - User should not have privileged access to reboot

Moreover, when logged in as a regular user, the left sidebar menu did not have the "Schedule" button. However, it was observed that a regular user could access the schedule by directly visiting the URL link to the schedule page.

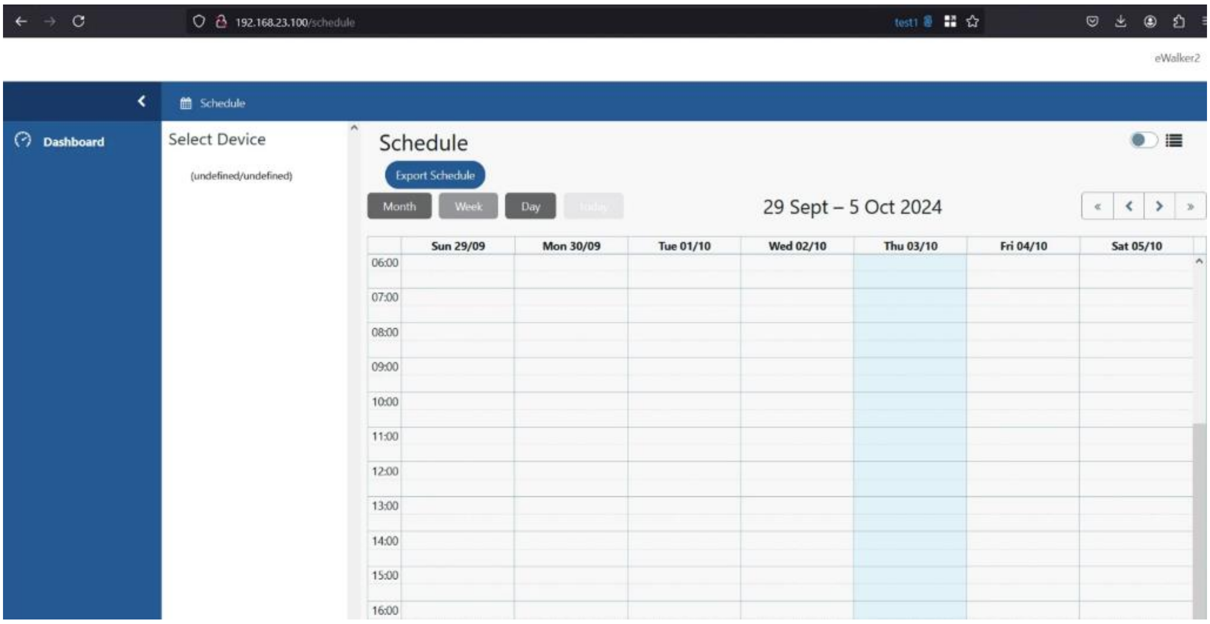


Figure 7 - The left sidebar menu did not have the "Schedule" button

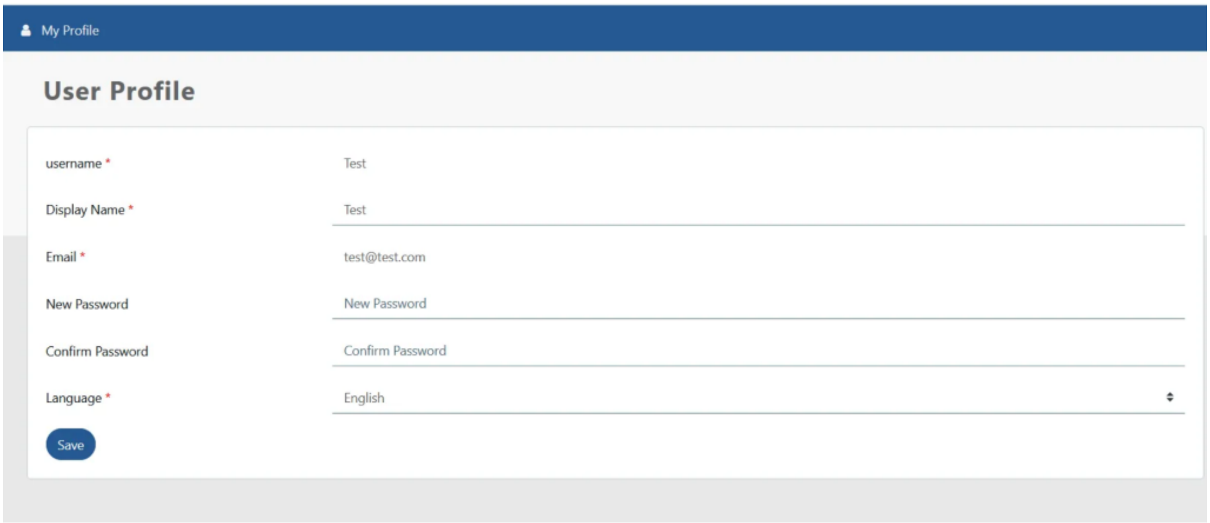
4.2.2 Medium Risk Findings

4.2.2.1 IoT-WEB-06: Client-Side Validation Bypass

Risk level	Medium
Affected Test ID	DP
OWASP Top 10	A05:2021 - Security Misconfiguration

Details

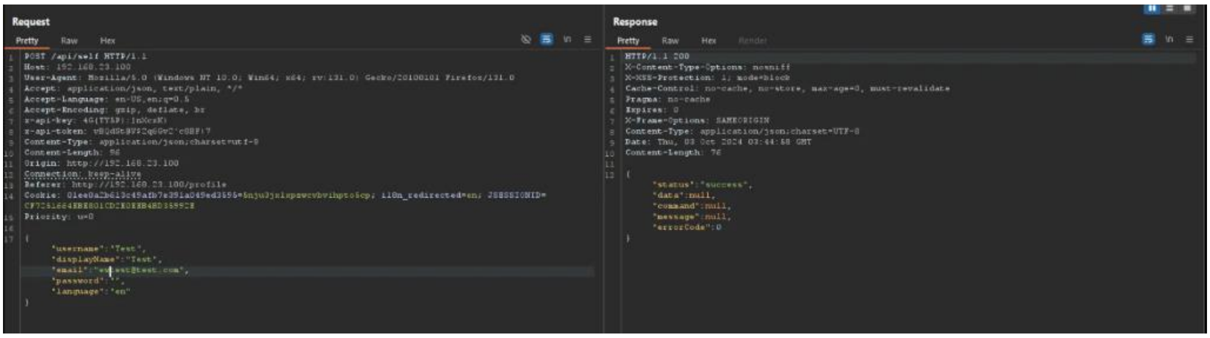
The portal enforces validation of the “email” parameter on the client side, preventing users from modifying it through the user interface. However, it was discovered that the email parameter can still be modified. By altering the email in the HTTP request, the change is accepted by the server, and the login username is also modified.



User Profile

username *	Test
Display Name *	Test
Email *	test@test.com
New Password	New Password
Confirm Password	Confirm Password
Language *	English

Figure 8 - Cannot Modify the “email” parameter in Client Side



```
POST /api/self HTTP/1.1
Host: 192.168.23.100
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
x-api-key: 4G(TY&P):InXcsK)
x-api-token: vBQdSt$V$2q6Gv2'cSBF!7
Content-Type: application/json;charset=utf-8
Content-Length: 96
Origin: http://192.168.23.100
Connection: keep-alive
Referer: http://192.168.23.100/profile
Cookie: 01ee0a2b613c49afb7e391a049ed3555=5nju3jxlxpswcvbvihpto5cp; i18n_redirected=en; JSESSIONID=
CF7251664BBE801CD2E0EBB4BD35992E
Priority: u=0

{
    "username":"Test",
    "displayName":"Test",
    "email":"ewtest@test.com",
    "password":"",
    "language":"en"
}
```

```
HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: SAMEORIGIN
Content-Type: application/json;charset=UTF-8
Date: Thu, 03 Oct 2024 03:44:58 GMT
Content-Length: 76

{
    "status":"success",
    "data":null,
    "command":null,
    "message":null,
    "errorCode":0
}
```

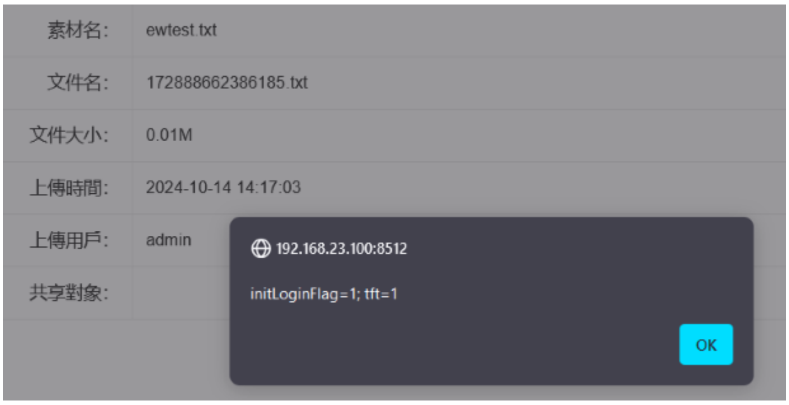
Figure 9 - Successful to Modify the “email” parameter in Server Side

4.2.2.2 IoT-WEB-07: Cross-Site Scripting

Risk level	Medium
Affected Test ID	B2P
OWASP Top 10	A05:2021 - Security Misconfiguration

Details

The vulnerability was identified when a text file was uploaded to the portal containing a malicious XSS payload. The input, "<img onerror="alert(document.cookie)" src=a>", was not properly sanitised by the server before being rendered on the page. As a result, the browser executes the script upon loading the page that displays the uploaded file, leading to the execution of alert(document.cookie), which demonstrates access to sensitive user data like cookies. The attack required an account with upload privilege.



素材名: ewtest.txt
文件名: 172888662386185.txt
文件大小: 0.01M
上傳時間: 2024-10-14 14:17:03
上傳用戶: admin
共享對象:

192.168.23.100:8512

initLoginFlag=1; tft=1

Figure 10 - Successful to implement the XSS Payload

4.2.2.3 IoT-WEB-08: Session Fixation

Risk level	Medium
Affected Test ID	B1P, CP
OWASP Top 10	A07:2021 - Identification and Authentication Failures

Details

The session cookie, which stores the session token, remains unchanged after user login. It leads to session fixation vulnerabilities that allow an attacker to impersonate a legitimate user by reading or manipulating their session token before login.

The session token does not change even after a successful login on the systems.

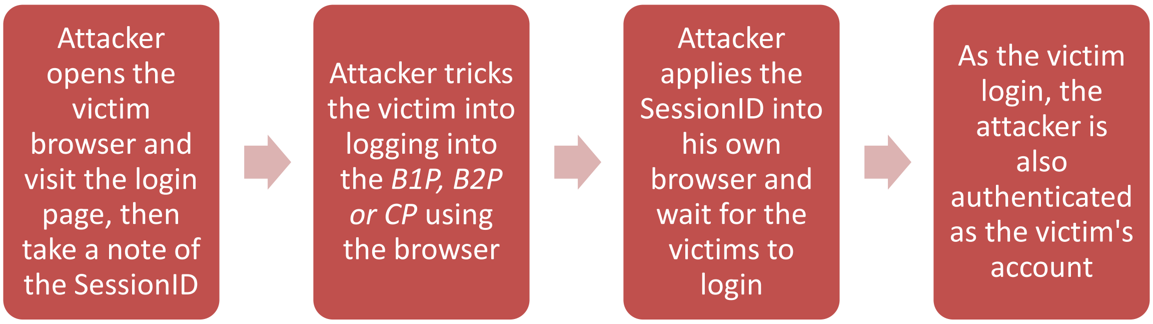

Attacker opens the victim browser and visit the login page, then take a note of the SessionID
Attacker tricks the victim into logging into the *B1P, B2P or CP* using the browser
Attacker applies the SessionID into his own browser and wait for the victims to login
As the victim login, the attacker is also authenticated as the victim's account

Figure 11 - Attack Flow of Session Fixation

The attacker will be authenticated after the victim is authenticated as they share the same session token, which is used by the system to identify users' authentication state.

4.2.2.4 IoT-WEB-09: Files Accessible Without Authentication

Risk level	Medium
Affected Test ID	B1P, CP
OWASP Top 10	A01:2021 - Broken Access Control

Details

In the portal, it was observed that anyone with the file's URL can access the files, even without authentication or permissions. This lack of access control allows unauthorised users to retrieve files by simply navigating to the URL.

4.2.3 Low Risk Findings

4.2.3.1 IoT-WEB-10: Changing Password Does not require Re-authentication

Risk level	Low
Affected Test ID	B2P, DP
OWASP Top 10	A04:2021 - Insecure Design

Details

It was found that the current password is not required when changing password. It would allow attackers who own a valid session to change the password without credentials. A valid session could be obtained through CSRF, XSS, from Event Logs or attacker gain access to a logged in portal through the browser.

4.2.3.2 IoT-WEB-11: Insecure HTTP Usage

Risk level	Low
Affected Test ID	B1P, B2P, CP, DP
OWASP Top 10	A04:2021 - Insecure Design

Details

B1P, B2P, CP and DP portals were transmitting sensitive data, such as login credentials or personal information, over HTTP instead of HTTPS. HTTP does not provide encryption, which means all data is sent in plaintext and can be easily intercepted by attackers on the network. This lack of encryption exposes sensitive information, compromising both user privacy and the integrity of the portal’s data.

5 Findings on Digital Signage Devices

5.1 Summary of Findings

The following table summarises the number of risk issues identified in the security test.

Finding ID	Description	Risk
IoT-DEV-01	Unauthorised Control via Infrared	High
IoT-DEV-02	Unauthorised Command Sending to the Signage	High
IoT-DEV-03	Exposed External Interface Ports	High
IoT-DEV-04	Enabled Touch Screen Allow Breakout	High
IoT-DEV-05	Display Malicious Programs using USB Device	High
IoT-DEV-06	Unencrypted Data Traffic	Medium
IoT-DEV-07	Disabled Windows Firewall / Windows Defender	Medium
IoT-DEV-08	Denial of Service (DoS)	Low
IoT-DEV-09	Unnecessary Network Services Exposed	Low

Table 5-1. Finding List – Digital Signage Devices

	Digital Signage Device Test ID *							
Finding ID	***A1***	***A2***	***B1***	***B2***	***C1***	***C2***	***D1***	***D2***
IoT-DEV-01a	Affected	Affected	-	Affected	-	Affected	-	Affected
IoT-DEV-01b	-	Affected	-	Affected	-	-	-	Affected
IoT-DEV-02	-		Affected	-	Affected	Affected	-	-
IoT-DEV-03	Affected	Affected	Affected	Affected	Affected	Affected	Affected	Affected
IoT-DEV-04	Affected	Affected	Affected	Affected	Affected	Affected	Affected	Affected
IoT-DEV-05	-	-	Affected	-	Affected	-	-	-
IoT-DEV-06	Affected	Affected	Affected	Affected	Affected	Affected	Affected	Affected
IoT-DEV-07	Affected	-	-	-	-	-	Affected	-
IoT-DEV-08	Affected	Affected	Affected	Affected	Affected	Affected	Affected	Affected
IoT-DEV-09	Affected	-	Affected	-	Affected	-	Affected	-

Table 5-2. Vulnerability Matrix – Digital Signage Devices

* “-“ means not affected.

5.2 *Detailed Findings on Digital Signage Devices*

5.2.1 *High Risk Findings*

5.2.1.1 IoT-DEV-01: Unauthorised Control via Infrared

Risk level	High
Affected Test ID	(a) Using Penetration Test Tool - *A1, B1, B2, C1, C2, D1, D2* (b) Using Universal Remotes - A2, B2, C2, D2
OWASP IoT Top 10	I10:2018 - Lack of Physical Hardening

Details

Infrared (IR) sensors were found in the affected signages. An attacker can control the signage using an infrared remote controller, enabling actions such as returning to the main menu, opening the browser to visit other websites, or even turning off the signage.

(a) Using Penetration Test Tool

A list of IR signal addresses and commands can be discovered during command brute-forcing. These can be used to control the monitor/system of the signage using a penetration test tool's built-in infrared module or an NEC infrared transmitter.

(b) Using Universal Remotes

Universal remotes may also be capable of executing some of the commands. Universal remotes typically come with a database of codes for various TV brands and models. An attacker could potentially control the signage by finding the correct code set using the search function on the universal remote.

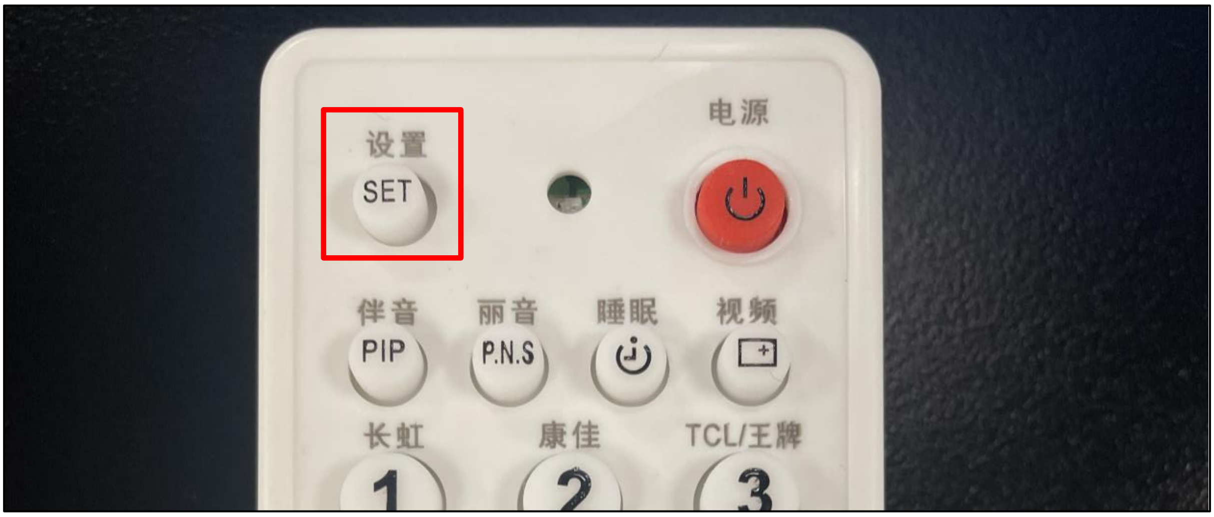


Figure 12 - The universal remote includes a function for searching for the correct commands

Using the universal remote, we can control the systems inside A2, B2 and C2 signages. We can also turn off the system of D2 using the search function of the remote. In some situations, turning off the displaying screen cannot be reflected in the management portal. Therefore, some of the attacks cannot be detected without physical inspection.

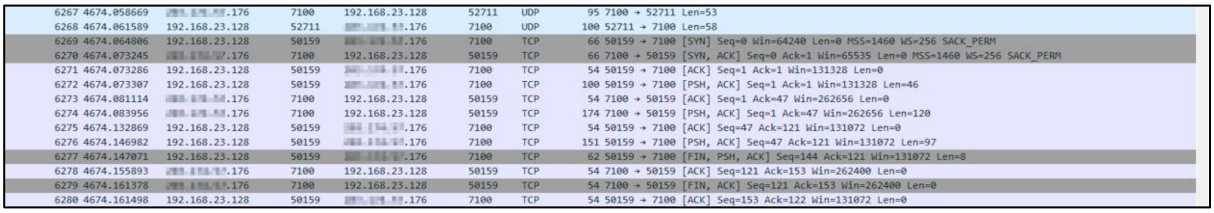
5.2.1.2 IoT-DEV-02: Unauthorised Command Sending to the Signage

Risk level	High
Affected Test ID	*B1, C1, C2*
OWASP IoT Top 10	I02:2018 - Insecure Network Services I03:2018 - Insecure Ecosystem Interfaces I07:2018 - Insecure Data Transfer and Storage

Details

It was observed that an attacker can impersonate the server to send and receive commands to and from the signage if they are able to send packets to it. This vulnerability potentially allows attackers to close the player or even shut down the machine remotely without permission.

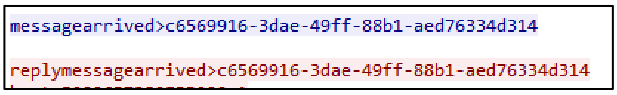
The following figure shows the captured traffic when the server attempts to send an “Open Player” command to B1 signage.



```
6267 4674.058669  [redacted].176    7100   192.168.23.128   52711  UDP    95 7100 → 52711 Len=53
6268 4674.061589  192.168.23.128    52711  [redacted].176   7100   UDP   100 52711 → 7100 Len=58
6269 4674.064806  192.168.23.128    50159  [redacted].176   7100   TCP    66 50159 → 7100 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
6270 4674.073245  [redacted].176    7100   192.168.23.128   50159  TCP    66 7100 → 50159 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
6271 4674.073286  192.168.23.128    50159  [redacted].176   7100   TCP    54 50159 → 7100 [ACK] Seq=1 Ack=1 Win=131328 Len=0
6272 4674.073307  192.168.23.128    50159  [redacted].176   7100   TCP   100 50159 → 7100 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=46
6273 4674.081114  [redacted].176    7100   192.168.23.128   50159  TCP    54 7100 → 50159 [ACK] Seq=1 Ack=47 Win=262656 Len=0
6274 4674.083956  [redacted].176    7100   192.168.23.128   50159  TCP   174 7100 → 50159 [PSH, ACK] Seq=1 Ack=47 Win=262656 Len=120
6275 4674.132869  192.168.23.128    50159  [redacted].176   7100   TCP    54 50159 → 7100 [ACK] Seq=47 Ack=121 Win=131072 Len=0
6276 4674.146982  192.168.23.128    50159  [redacted].176   7100   TCP   151 50159 → 7100 [PSH, ACK] Seq=47 Ack=121 Win=131072 Len=97
6277 4674.147071  192.168.23.128    50159  [redacted].176   7100   TCP    62 50159 → 7100 [FIN, PSH, ACK] Seq=144 Ack=121 Win=131072 Len=8
6278 4674.155893  [redacted].176    7100   192.168.23.128   50159  TCP    54 7100 → 50159 [ACK] Seq=121 Ack=153 Win=262400 Len=0
6279 4674.161378  [redacted].176    7100   192.168.23.128   50159  TCP    54 7100 → 50159 [FIN, ACK] Seq=121 Ack=153 Win=262400 Len=0
6280 4674.161498  192.168.23.128    50159  [redacted].176   7100   TCP    54 50159 → 7100 [ACK] Seq=153 Ack=122 Win=131072 Len=0
```

Figure 13 - The captured traffics when the server tries to send an “Open Player” command to the signage

Before the server sends the command to the signage, it will send a UDP packet containing “messagearrived>{UUID}” to notify the signage. The signage will reply to the server with a “replymessagearrived” UDP packet.

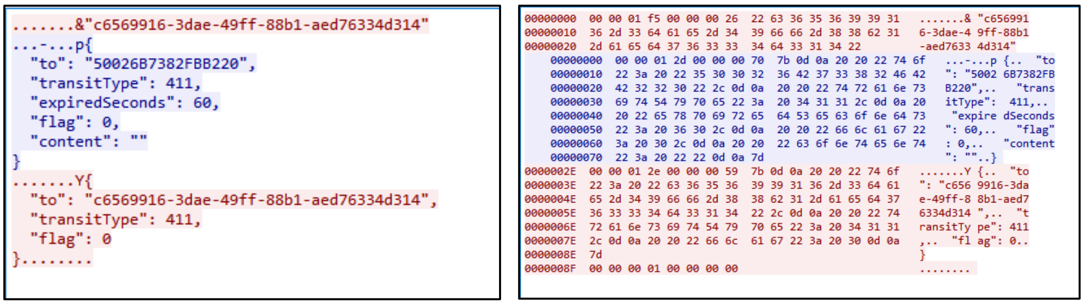


```
messagearrived>c6569916-3dae-49ff-88b1-aed76334d314

replymessagearrived>c6569916-3dae-49ff-88b1-aed76334d314
```

Figure 14 - The content of the UDP packets A

The signage will establish a TCP connection with the IP address that sent the UDP packet (In Figure 13, signage server’s has the IP of ‘X.X.X.176`). The signage will then send a TCP packet containing the UUID it received earlier. The server will send the command to the signage, with “transitType” specifying the command the signage needs to perform. After receiving the command, the signage will send a packet back to the server to confirm receipt before executing the command.



```
.......&"c6569916-3dae-49ff-88b1-aed76334d314"
...-...p{
  "to": "50026B7382FBB220",
  "transitType": 411,
  "expiredSeconds": 60,
  "flag": 0,
  "content": ""
}
.......Y{
  "to": "c6569916-3dae-49ff-88b1-aed76334d314",
  "transitType": 411,
  "flag": 0
}........
```

```
00000000  00 00 01 f5 00 00 00 26  22 63 36 35 36 39 39 31   .......& "c656991
00000010  36 2d 33 64 61 65 2d 34  39 66 66 2d 38 38 62 31   6-3dae-4 9ff-88b1
00000020  2d 61 65 64 37 36 33 33  34 64 33 31 34 22         -aed7633 4d314"
    00000000  00 00 01 2d 00 00 00 70  7b 0d 0a 20 20 22 74 6f   ...-...p {..  "to
    00000010  22 3a 20 22 35 30 30 32  36 42 37 33 38 32 46 42   ": "5002 6B7382FB
    00000020  42 32 32 30 22 2c 0d 0a  20 20 22 74 72 61 6e 73   B220",..   "trans
    00000030  69 74 54 79 70 65 22 3a  20 34 31 31 2c 0d 0a 20   itType":  411,..
    00000040  20 22 65 78 70 69 72 65  64 53 65 63 6f 6e 64 73    "expire dSeconds
    00000050  22 3a 20 36 30 2c 0d 0a  20 20 22 66 6c 61 67 22   ": 60,..   "flag"
    00000060  3a 20 30 2c 0d 0a 20 20  22 63 6f 6e 74 65 6e 74   : 0,..   "content
    00000070  22 3a 20 22 22 0d 0a 7d                            ": ""..}
0000002E  00 00 01 2e 00 00 00 59  7b 0d 0a 20 20 22 74 6f   .......Y {..  "to
0000003E  22 3a 20 22 63 36 35 36  39 39 31 36 2d 33 64 61   ": "c656 9916-3da
0000004E  65 2d 34 39 66 66 2d 38  38 62 31 2d 61 65 64 37   e-49ff-8 8b1-aed7
0000005E  36 33 33 34 64 33 31 34  22 2c 0d 0a 20 20 22 74   6334d314 ",..  "t
0000006E  72 61 6e 73 69 74 54 79  70 65 22 3a 20 34 31 31   ransitTy pe": 411
0000007E  2c 0d 0a 20 20 22 66 6c  61 67 22 3a 20 30 0d 0a   ,..  "fl ag": 0..
0000008E  7d                                                 }
0000008F  00 00 00 01 00 00 00 00                            ........
```

Figure 15 - The content of the TCP packets B

The attacker can impersonate the server to send commands to the signage by first sending a UDP packet, establishing a TCP connection, and then requesting actions by sending different values for “transitType”.

Since B1, C1 and C2 signage are managed by the web management portals which manufactured by the same software producer, the same vulnerability exists on B1, C1 and C2.

5.2.1.3 IoT-DEV-03: Exposed External Interface Ports

Risk level	High
Affected Test ID	A1, A2, B1, B2, C1, C2, D1, D2
OWASP IoT Top 10	I10:2018 - Lack of Physical Hardening

Details

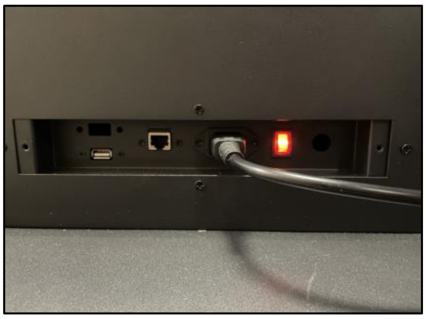
Several external interface ports exist on the back of the signage, including USB ports, a LAN port, HDMI ports, and more. An attacker could exploit these ports to perform various attacks, such as injecting a malicious USB flash drive to display harmful content or turning off the machine.

The number of external interface ports on each signage is as follows:

External Interface Ports	***Digital Signage Device Test ID***							
	A1	***A2***	***B1***	***B2***	***C1***	***C2***	***D1***	***D2***
Power Switch	1	1	1	1	1	1	1	1
System On/Off Button	1	0	1	0	1	0	1	0
USB Port	1	1	2	2	4	2	2	1
LAN Port	1	1	1	1	1	1	1	1
HDMI OUT Port	0	0	0	0	1	0	0	0
HDMI IN Port	0	0	0	0	0	*1**	0	0
VGA Port	0	0	0	0	1	0	0	0
Audio OUT Port	0	0	0	0	1	0	0	0

Table 5-3. External Interface Ports on Signage Devices

* The HDMI IN Port is working properly as the connected system can recognise the monitor of the signage. However, it was unable to change the monitor's channel to display the connected system.

Digital Signage Device Test ID	**Photo(s)**
A1	
A2	
B1	
B2	
C1	

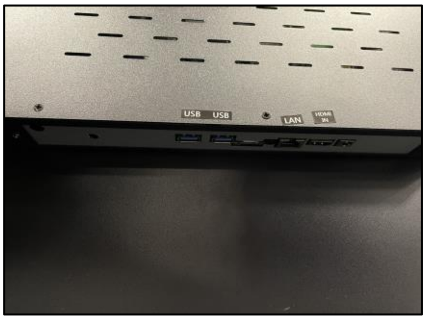
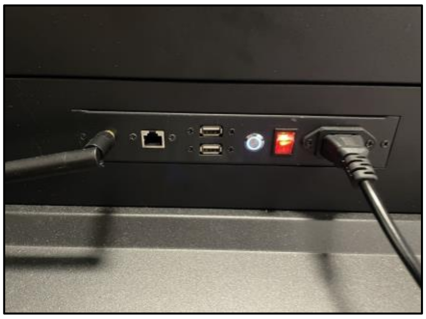
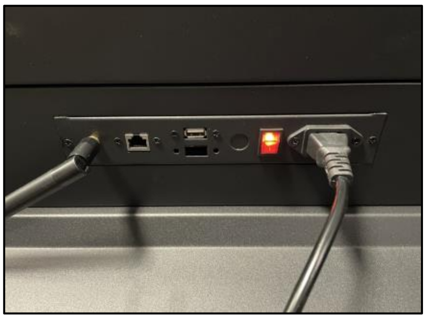
C2	
D1	
D2	

Table 5-4. Photos of External Interface Ports on Signage Devices

5.2.1.4 IoT-DEV-04: Enabled Touch Screen Allow Breakout

Risk level	High
Affected Test ID	A1, A2, B1, B2, C1, C2
OWASP IoT Top 10	I10:2018 - Lack of Physical Hardening

Details

All of the signages allow users to interact with them by touch. Some of the signages could open the media player settings menu or even exit from the media player by performing specific touch gestures.

The following table shows the touch gestures needed to exit from the media player / perform other actions:

Digital Signage Device Test ID	Gestures
A1	Method 1: Swipe with finger from the left edge of the screen to open the widgets -> Click “x” to close the player Method 2: Swipe with finger from the right edge of the screen to open the notification centre, which allows attacker to open applications like Setting
A2	Swipe with one finger from the bottom edge of the screen to show the navigation bar -> Tap Home ○
B1	Clicking the upper corner of the screen 5 times to open the player menu -> Exit
B2	Press the screen 5 seconds -> The prompt “Please continue” appears -> click 5 times to open the player menu -> Exit
C1	Clicking the upper corner of the screen 5 times to open the player menu -> Exit
C2	Clicking the upper corner of the screen 5 times to open the player menu -> Exit
D1	Unable to breakout using touch gestures
D2	Unable to breakout using touch gestures

Table 5-5. Gestures to Exit from the Media Player / Perform Other Actions

5.2.1.5 IoT-DEV-05: Display Malicious Programs using USB Device

Risk level	High
Affected Test ID	*B1, C1*
OWASP IoT Top 10	I03:2018 - Insecure Ecosystem Interfaces

Details

The affected signages have a function that allows them to pull a program from a USB flash drive to the media player. Combined with the vulnerability IoT-DEV-03, this enables an attacker to run a malicious program if they can create a valid program on the USB flash drive.

The program on the USB flash drive must follow a specific folder structure, which an attacker could determine by capturing and analysing the traffic of the signage as it downloads the program from the server.

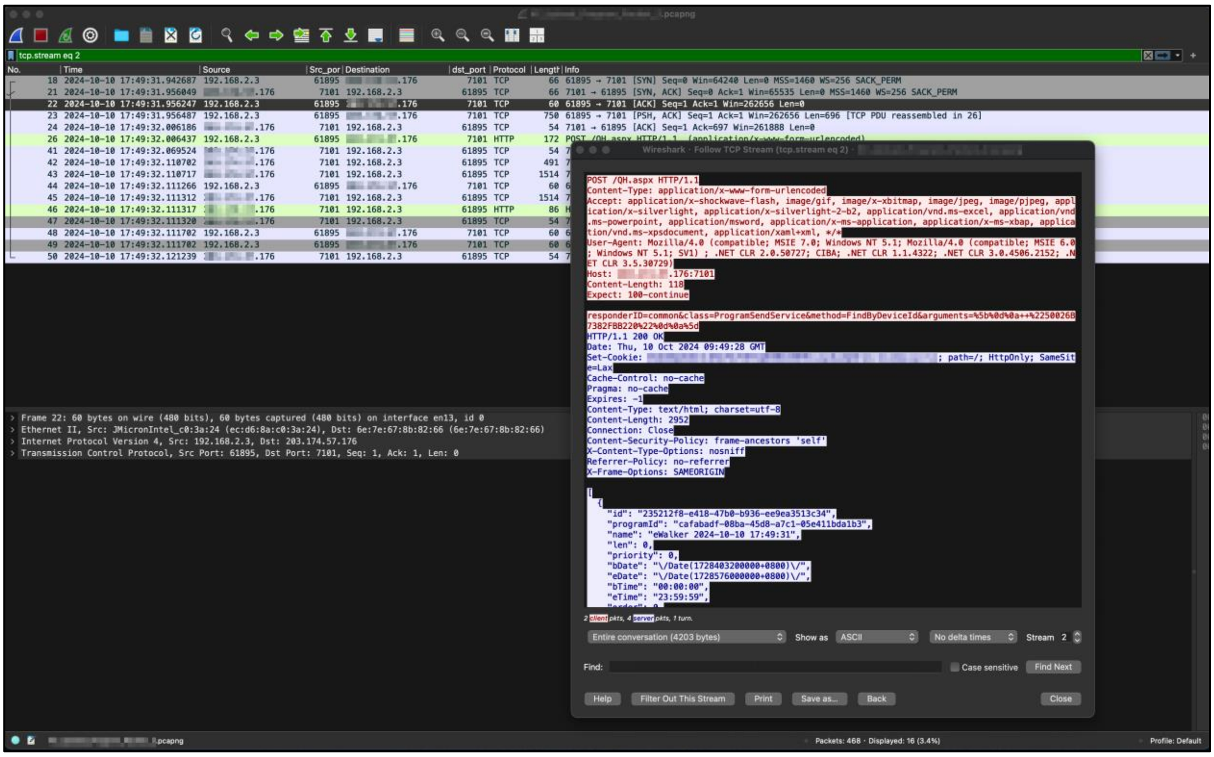
5.2.2 Medium Risk Findings

5.2.2.1 IoT-DEV-06: Unencrypted Data Traffic

Risk level	Medium
Affected Test ID	A1, A2, B1, B2, C1, C2, D1, D2
OWASP IoT Top 10	I03:2018 - Insecure Ecosystem Interfaces I07:2018 - Insecure Data Transfer and Storage

Details

We discovered that all the signage did not encrypt their data traffic. This allows the attacker to perform man-in-the-middle (MitM) attack and sniff some information, e.g. images and videos from the signage, or even interfere with traffic flow to perform other attacks, like IOT-DEV-02.



```
tcp.stream eq 2
No.  Time                        Source              Src_por Destination         dst_port Protocol Length Info
  18 2024-10-10 17:49:31.942687 192.168.2.3         61895   [redacted].176      7101  TCP     66 61895 → 7101 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
  21 2024-10-10 17:49:31.956049 [redacted].176      7101    192.168.2.3         61895 TCP     66 7101 → 61895 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  22 2024-10-10 17:49:31.956247 192.168.2.3         61895   [redacted].176      7101  TCP     60 61895 → 7101 [ACK] Seq=1 Ack=1 Win=262656 Len=0
  23 2024-10-10 17:49:31.956487 192.168.2.3         61895   [redacted].176      7101  TCP    750 61895 → 7101 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=696 [TCP PDU reassembled in 26]
  24 2024-10-10 17:49:32.006186 [redacted].176      7101    192.168.2.3         61895 TCP     54 7101 → 61895 [ACK] Seq=1 Ack=697 Win=261888 Len=0
  26 2024-10-10 17:49:32.006437 192.168.2.3         61895   [redacted].176      7101  HTTP   172 POST /QH.aspx HTTP/1.1  (application/x-www-form-urlencoded)
  41 2024-10-10 17:49:32.069524 [redacted].176      7101    192.168.2.3         61895 TCP     54 7
  42 2024-10-10 17:49:32.110702 [redacted].176      7101    192.168.2.3         61895 TCP    491 7
  43 2024-10-10 17:49:32.110717 [redacted].176      7101    192.168.2.3         61895 TCP   1514 7
  44 2024-10-10 17:49:32.111266 192.168.2.3         61895   [redacted].176      7101  TCP     60 6
  45 2024-10-10 17:49:32.111312 [redacted].176      7101    192.168.2.3         61895 TCP   1514 7
  46 2024-10-10 17:49:32.111317 [redacted].176      7101    192.168.2.3         61895 HTTP    86 H
  47 2024-10-10 17:49:32.111320 [redacted].176      7101    192.168.2.3         61895 TCP     54 7
  48 2024-10-10 17:49:32.111702 192.168.2.3         61895   [redacted].176      7101  TCP     60 6
  49 2024-10-10 17:49:32.111702 192.168.2.3         61895   [redacted].176      7101  TCP     60 6
  50 2024-10-10 17:49:32.121239 [redacted].176      7101    192.168.2.3         61895 TCP     54 7

> Frame 22: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface en13, id 0
> Ethernet II, Src: JMicronIntel_c0:3a:24 (ec:d6:8a:c0:3a:24), Dst: 6e:7e:67:8b:82:66 (6e:7e:67:8b:82:66)
> Internet Protocol Version 4, Src: 192.168.2.3, Dst: 203.174.57.176
> Transmission Control Protocol, Src Port: 61895, Dst Port: 7101, Seq: 1, Ack: 1, Len: 0
```

Wireshark · Follow TCP Stream (tcp.stream eq 2) · [redacted]

```
POST /QH.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Accept: application/x-shockwave-flash, image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, appl
ication/x-silverlight, application/x-silverlight-2-b2, application/vnd.ms-excel, application/vnd
.ms-powerpoint, application/msword, application/x-ms-application, application/x-ms-xbap, applica
tion/vnd.ms-xpsdocument, application/xaml+xml, */*
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Mozilla/4.0 (compatible; MSIE 6.0
; Windows NT 5.1; SV1) ; .NET CLR 2.0.50727; CIBA; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .N
ET CLR 3.5.30729)
Host: [redacted].176:7101
Content-Length: 118
Expect: 100-continue

responderID=common&class=ProgramSendService&method=FindByDeviceId&arguments=%5b%0d%0a++%2250026B
7382FBB220%22%0d%0a%5d
HTTP/1.1 200 OK
Date: Thu, 10 Oct 2024 09:49:28 GMT
Set-Cookie: [redacted]; path=/; HttpOnly; SameSit
e=Lax
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=utf-8
Content-Length: 2952
Connection: Close
Content-Security-Policy: frame-ancestors 'self'
X-Content-Type-Options: nosniff
Referrer-Policy: no-referrer
X-Frame-Options: SAMEORIGIN

[
  {
    "id": "235212f8-e418-47b0-b936-ee9ea3513c34",
    "programId": "cafabadf-08ba-45d8-a7c1-05e411bda1b3",
    "name": "eWalker 2024-10-10 17:49:31",
    "len": 0,
    "priority": 0,
    "bDate": "\/Date(1728403200000+0800)\/",
    "eDate": "\/Date(1728576000000+0800)\/",
    "bTime": "00:00:00",
    "eTime": "23:59:59",
```

2 client pkts, 4 server pkts, 1 turn.

Entire conversation (4203 bytes) · Show as: ASCII · No delta times · Stream 2

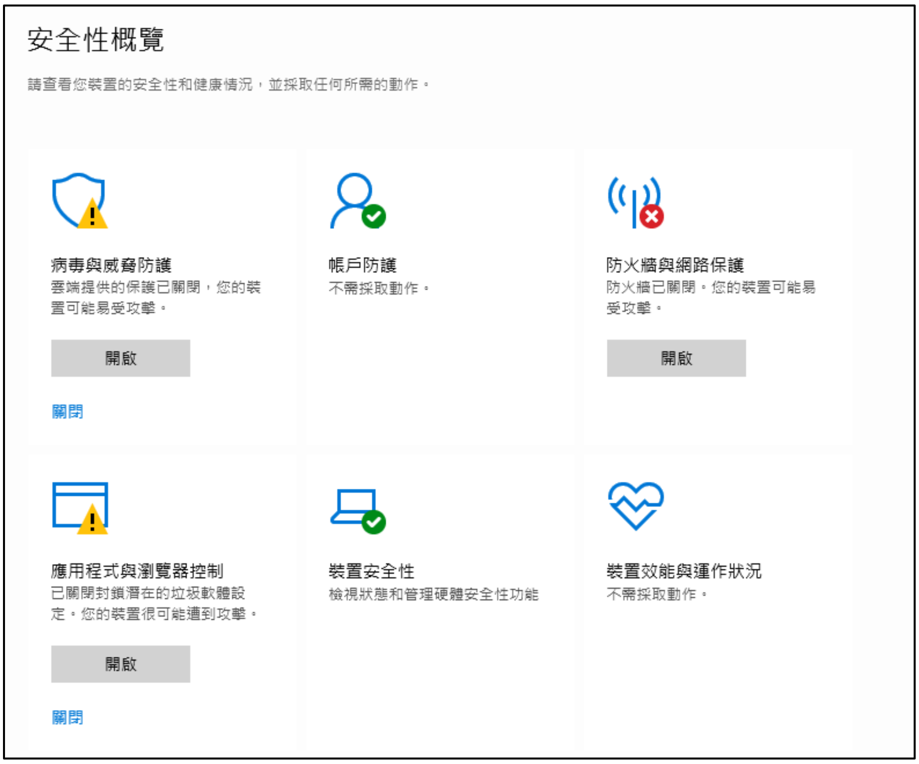
Figure 16 - A TCP / HTTP packet that contains the content of the program

5.2.2.2 IoT-DEV-07: Disabled Windows Firewall or Windows Defender

Risk level	Medium
Affected Test ID	*A1, D1*
OWASP IoT Top 10	I03:2018 - Insecure Ecosystem Interfaces I07:2018 - Insecure Data Transfer and Storage

Details

We discovered that *A1* signage by default disabled Windows Firewall and *D1* signage by default did not contain Windows Defender. With the Windows Firewall disabled, the signage is more susceptible to unauthorised access. With the Windows Defender disabled, the signage is more vulnerable to viruses, ransomware, and other malicious software.



安全性概覽

請查看您裝置的安全性和健康情況，並採取任何所需的動作。

病毒與威脅防護
雲端提供的保護已關閉，您的裝置可能易受攻擊。

帳戶防護
不需採取動作。

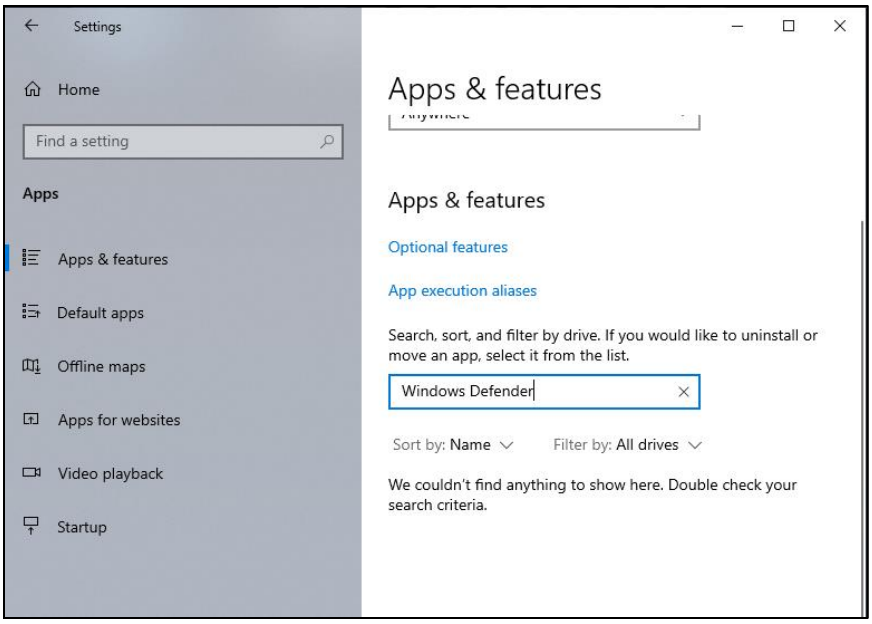
防火牆與網路保護
防火牆已關閉。您的裝置可能易受攻擊。

應用程式與瀏覽器控制
已關閉封鎖潛在的垃圾軟體設定。您的裝置很可能遭到攻擊。

裝置安全性
檢視狀態和管理硬體安全性功能

裝置效能與運作狀況
不需採取動作。

Figure 17 - Windows Firewall is disabled in *A1* Signage



Apps & features

Search, sort, and filter by drive. If you would like to uninstall or move an app, select it from the list.

Windows Defender

Sort by: Name

Filter by: All drives

We couldn't find anything to show here. Double check your search criteria.

Figure 18 - Windows Defender is not installed in *D1* Signage

5.2.3 Low Risk Findings

5.2.3.1 IoT-DEV-08: Denial of Services (DoS)

Risk level	Low
Affected Test ID	A1, A2, B1, B2, C1, C2
OWASP IoT Top 10	I02:2018 - Insecure Network Services

Details

When performed port scanning via a LAN connection / Wi-Fi connection on the target signage, the signage was responding slowly, resulting in user interaction lag. This suggests a potential vulnerability to Denial of Service (DoS) attack, such as TCP SYN Flood, which could exhaust its resources and render it unusable or inaccessible.

5.2.3.2 IoT-DEV-09: Unnecessary Network Services Exposed

Risk level	Low
Affected Test ID	A1, B1, C1, D1
OWASP IoT Top 10	I07:2018 - Insecure Data Transfer and Storage

Details

Some of the signage have Remote Procedure Calls (RPC) service enabled, with Network Ports 135 and 445 open. This may increase the risk of being attacked through these network ports.

6 Security Recommendations

According to the findings that discovered in this security test, recommendations have been provided in this section as to mitigate and minimise the security risks. Furthermore, digital signage users are recommended adopt the security best practices according to HKCERT's "IoT Security Guideline on Digital Signage" [3].

6.1 Security Recommendations on Signage Web Management Portals

IoT-WEB-01 – Sensitive Information Disclosure

- Implement strict access control measures through authentication and authorisation to ensure that only authorised users can access sensitive endpoints.
- Password and password hashes should never be returned in an API response.

IoT-WEB-02 – Insecure Password Hash

- Use established password hashing algorithm, e.g. Argon2id, BCrypt, or PBKDF2 with appropriate parameters.
- Use a unique random password salt in the password hash when performing password hashing to provide stronger brute force resilience. It will significantly decrease the chance of leaking user's plaintext password if an attacker gains access to the underlying database.

IoT-WEB-03 – Outdated Software Libraries

- Regularly check and update for security patches from the software library vendors.

IoT-WEB-04 – SQL Injection

- Do not construct SQL statement by concatenating user inputs.
- Use parameterised query for all variables.
- Implement strict input validation using whitelisting approach to ensure user input meets expected criteria.
- Avoid exposing detailed SQL error messages by displaying generic errors that conceal database and application logic details.

IoT-WEB-05 – Broken Access Control

- Ensure that only authorised users with the necessary permissions can access critical functions such as system reboot and management portal's URLs.
- Regular users should be strictly restricted to their assigned privileges.

IoT-WEB-06 – Client-Side Validation Bypass

- Ensure that all input, including parameters like "email", is validated on the server side.

IoT-WEB-07 – Cross-Site Scripting

- Implement server-side sanitization to erase user-supplied content.
- Validate the content type of uploaded files to ensure they match expected formats
- Enforce a strong Content Security Policy (CSP) to restrict the execution of inline JavaScript and loading of unauthorised external resources.

IoT-WEB-08 – Session Fixation

- Ensure that the session token changes upon every successful login, logout or any security context change.

IoT-WEB-09 – Files Accessible Without Authentication

- Implement proper access control to ensure uploaded files are accessible only to authenticated users with the appropriate permissions.
- Store files in protected directories or outside the public web directory, serving them through authenticated access routes that validate the user's session.

IoT-WEB-10 – Changing Password does not Require Re-authentication

- Validate current password when changing password.

IoT-WEB-11 – Insecure HTTP Usage

- Configure the server to use HTTPS for all communication by implementing SSL/TLS and redirecting HTTP traffic to HTTPS
- Enable HTTP Strict Transport Security (HSTS) to enforce HTTPS for all future communications, automatically redirecting users from HTTP to HTTPS.

6.2 Security Recommendations on Signage Devices

IoT-DEV-01 – Unauthorised Control via Infrared

- Disable the infrared sensor by unplugging the wire, or blocking the infrared sensor using tape or other materials, if the infrared function is not used.

IoT-DEV-02 – Unauthorised Command Sending to the Signage

- Encrypt the communication between the server and the digital signage device.
- Configure the signage to verify the origin of the packet to ensure that the command is coming from the legitimate server.
- Include timestamps and nonces in the command packet to prevent replay attacks.
- Block malicious packets sent from unknown IP addresses and only allow packets from the legitimate signage server by creating firewall rules.

IoT-DEV-03 – Exposed External Interface Ports

- Restrict physical access to external interface ports, such as USB ports, HDMI ports, and network ports, by adding physical locks.

IoT-DEV-04 – Enabled Touch Screen Allow Breakout

- Disable the touch function:

For Windows signages, users can disable the touch function by disabling the "HID-Compliant touchscreen" with the following steps:

1. Press "Windows" key + "X". Select "Device Manager".
2. Under "Human Interface Devices", find "HID-Compliant touchscreen".
3. Right-click the device name and select "Disable".

For Android signages, users can pin an app's screen to keep it in view until the user unpin it with your PIN, pattern, or password. They will have to turn on "app pinning" first with the following steps:

1. Open the signage's Settings app
2. Tap "Security" or "Security & location" > "Advanced" > "App pinning"
3. Turn on "App pinning"

Then the user can pin the application with the following steps:

1. Go to the media player app
2. Open the overview by tapping 'Overview'
3. At the top of the image, tap the app's icon

4. Tap ‘Pin’

If the touch function is necessary, the vendor should lock the program or disable certain operating system gestures to ensure that an attacker cannot exit the media player. Additionally, the vendor can perform OS hardening to restrict which programs are allowed to execute.

IoT-DEV-05 – Display Malicious Programs using USB Device

- Restrict physical access to external interface ports by disabling the access through system or adding physical locks, e.g. USB ports, HDMI ports, network ports etc.

IoT-DEV-06 – Unencrypted Data Traffic

- Encrypt communication between the server and the digital signage device.

IoT-DEV-07 – Disabled Windows Firewall / Windows Defender

- Install and enable both Windows Firewall and Window Defender.

IoT-DEV-08 – Denial of Service (DoS)

- Apply rate limit to a specific signage and detect illegitimate traffic and block it at the routing level by configurating the routers / switches.

IoT-DEV-09 – Unnecessary Network Services Exposed

- Disable the network services if it is unnecessary, or block network access using network firewall.

7 References

[1] https://owasp.org/Top10/

[2] https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10

[3] https://www.hkcert.org/security-guideline/iot-security-guideline-for-digital-signage