Red Sentry

Modern Pentests to Fight Modern Hackers.

Dunder Mifflin

Web Application Penetration Test

COMPLETE REPORT

January 1st, 2022

Red Sentry

Contact Information
+1 (888) 337- 0467

TABLE OF CONTENTS

Summary

Project Overview

Dunder Mifflin engaged Red Sentry to assess the security of their web application. The following report details the findings identified during the course of the engagement, which started on January 1st, 2022

Goals

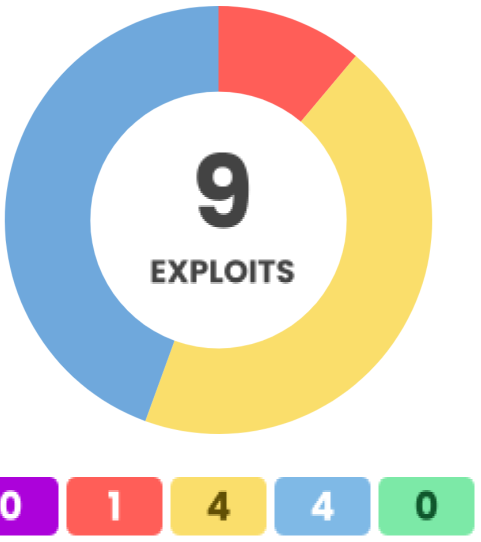
Identify vulnerabilities in the company's web application following the OWASP Top 10 methodology.

Dates

January 1st, 2022	January 1st - 6th, 2022	January 7th, 2022
KICKOFF	**TESTING PERIOD**	**DELIVERY**

Findings

GRADE



Scope

- stage-order.dundermifflin.com
- Mobile files

Executive Summary

The assessment team conducted a penetration test on Dunder Mifflin's web application and mobile app files. Overall, the application was insecure as 9 issues were detected.

Regarding the severity of the findings, one high vulnerability was spotted. It consists of broken access control that allows an attacker to access information from other users.

Next, two types of medium vulnerabilities were identified. These are related to some improper access controls and the possibility to inject NoSQL statements in some requests.

Additionally, three types of low vulnerabilities were detected. An email flooding issue, some pieces of sensitive information exposed and a security misconfiguration found in the mobile application files.

In summary, to improve the organization's security posture, the assessment team recommends improving the authorization mechanism within the application, sanitizing users input to avoid NoSQL injection, implementing rate-limiting mechanisms, avoiding the exposure of sensitive information, and changing potentially insecure setups if possible.

Impact

Based on the findings, the assessment team identified the following risks:

- Overall, the reported issues affect directly the confidentiality risk to a high extent
- Additionally, the email flooding issue affects availability risk to a low extent

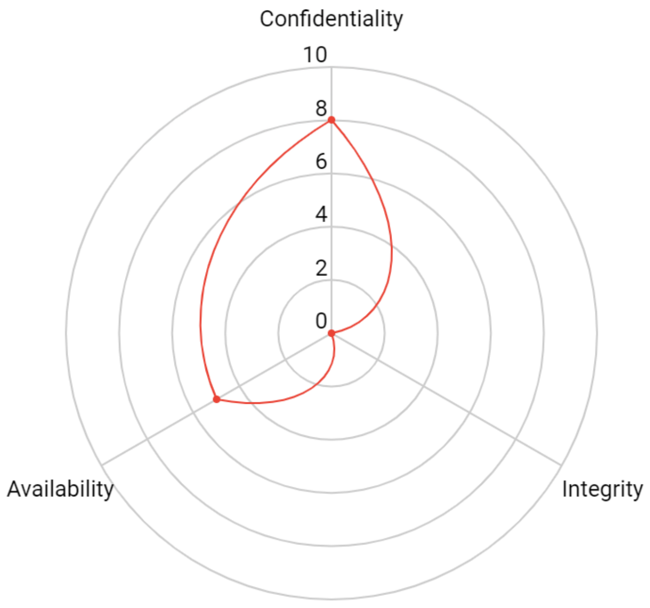


Figure 1: Security risk chart

As the image above shows, there's still an overall medium risk as the average risk is above 4. Only one of the risk components is in the high range and availability is within the medium range.

As a general rule, an area below 4 shows low business impact, between 4 and 7 medium impact and above 7 high.

External Penetration Findings

Broken Access Controls

HIGH

In order to access certain features of an application, some validations are needed to confirm either if the user is authorized to do so, or if an input is valid to be processed by the application server or if the data given to a user belongs only to that user. If any of those validations are incomplete, it could lead to a sensitive data exposure issue.

Details

While performing different tests on the “place order” feature, a broken access control instance was found.

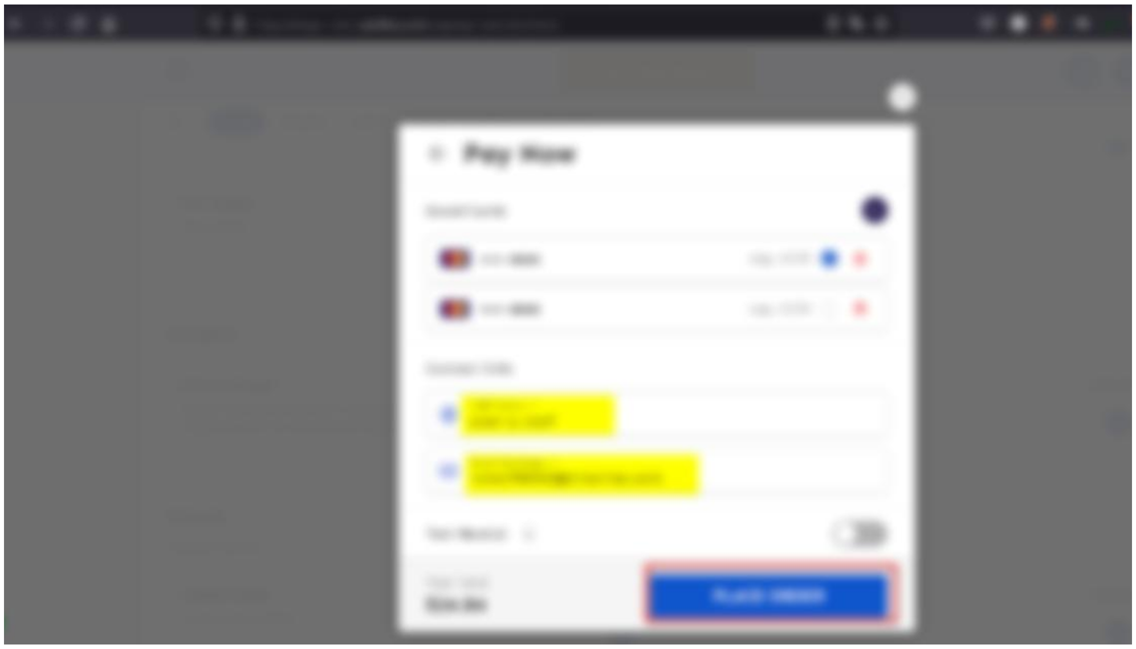


Figure 2: Pay modal

First, in order to reproduce this issue, an order was placed using the email "grant@sample.com". This generated a backend request to /dundermifflin-api/v1/account/me/order/4624/submit with the email and other information as body parameters.

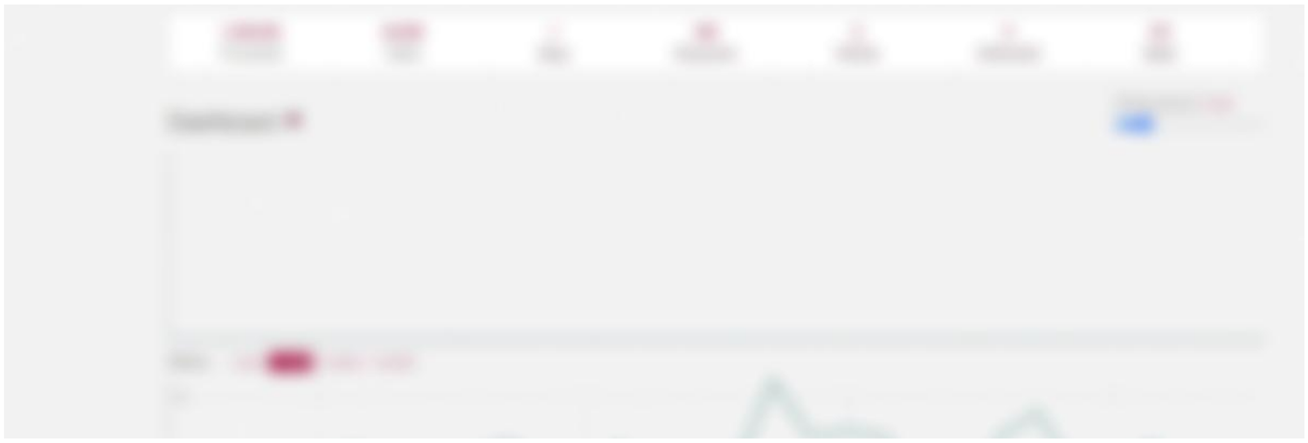


Figure 3: POST request modified

Then, we change the request by modifying the email to that of our target, in this case, "grant@gmail.com". Additionally, we will iterate over the order number "4624" to see if we're able to perform multiple requests simultaneously.

Figure 4: Target data disclosed

As can be seen above, at least one of our iterated requests was successful, revealing sensitive information from the target, like its name and other business-related data.

Business impact

The information retrieved by this issue could be used by an attacker to steal the victim's identity or make fraudulent purchases.

Recommendations

To mitigate the risk of this issue, the assessment team recommends the following steps:

- Set more consistent access controls according to each existing type of user, role, and permissions

Locations & Occurrences (1)

- stage-order.dundermifflin.com/dundermifflin-api/v1/account/me/order/4624

Resources

About Broken Access controls

cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html

NoSQL Injection

MEDIUM

Generally speaking, a SQL injection is a type of attack that allows an attacker to execute specially crafted queries using SQL language on a relational DB with a purpose different than the business one intended. Nevertheless, unlike relational databases, NoSQL databases do not use a common query language. NoSQL query syntax is product-specific and queries may be written in the programming language of the application.

Having said that, MongoDB is one of the most popular NoSQL databases nowadays, it uses a format called BSON (Binary Javascript Object Notation) and if it’s not correctly sanitized, malicious users can make special queries, called NoSQL injections, to either steal sensitive data, add corrupted data, or even erase critical business information.

Details

After testing some endpoints that contained a JSON body message which contained all of the data needed to perform a MongoDB query, the assessment team found a NoSQL injection issue on the api/v1/combined path.

Figure 2: NoSQL injection showing information about available sites and companies

As it can be seen from the image above, a malicious user might have access to the names and subdomains of the organizations contained in the database.

Business impact

Besides the obvious confidentiality risk this issue poses, it can also be leveraged to perform brute-forcing or phishing attacks to gain access to one of Dunder Mifflin's dedicated portals.

Recommendations

To mitigate the risk of this issue, the assessment team recommends the following steps:

- Apply validations to inputs coming either from GET parameters or request body
- Improve your firewall control policies to block requests different from the intended

Locations & Occurrences (2)

- pentest1.dundermifflin.com/api/v1/combined
- pentest2.dundermifflin.com/api/v1/combined

Resources

NoSQL Injection

ghostlulz.com/nosql-injection/

What is NoSQL injection?

www.invicti.com/blog/web-security/what-is-nosql-injection/

Improper Access Controls

MEDIUM

In order to access certain features of an application, usually some validations are needed to confirm either if the user is authorized to do so, or if an input is valid to be processed by the application server or if the data given to a user belongs only to that user. If any of those validations is incomplete, it could lead to a sensitive data exposure issue.

Details

While the assessment team was reviewing the /api/v1/organizations/search path, an improper access control instance was discovered.

Figure 3: Successful request obtaining organizations data

From the image above it can be seen that sensitive information might be retrieved by a malicious user, information that is in fact similar to the one displayed on the previous vulnerability.

Business impact

The impact is similar to the previous vulnerability, so it can be leveraged to perform brute-forcing or phishing attacks to gain access to one of Dunder Mifflin's dedicated portals.

Recommendations

To mitigate the risk of this issue, the assessment team recommends the following steps:

- Setting more consistent access controls according to each existing type of user, role, and permissions

Locations & Occurrences (2)

- pentest1.dundermifflin.com/api/v1/organizations/search
- pentest2.dundermifflin.com/api/v1/organizations/search

Resources

Access control vulnerabilities and privilege escalation

portswigger.net/web-security/access-control

Email Flooding

LOW

Email Flooding is a form of a denial-of-service (DoS) attack designed to overwhelm an inbox or inhibit a server by sending a massive number of emails to a specific person or system. The aim is to fill up the recipient's disk space on the server or overload a server to stop it from functioning. The feature to send mail to users or businesses has no rate limiting or no captcha implemented. Therefore, a malicious user can use this to mail bomb any email's inbox with emails.

Details

The assessment team discovered two different features that allow an attacker to perform email flooding.

Forgot password

After clicking on the "Forgot your password" option located on the login page, and entering a valid email address, the corresponding backend request was intercepted.

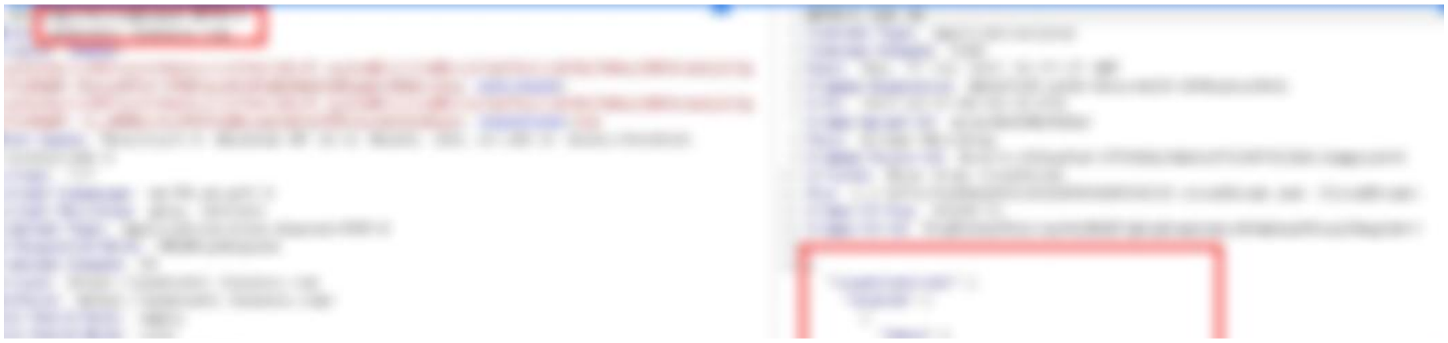
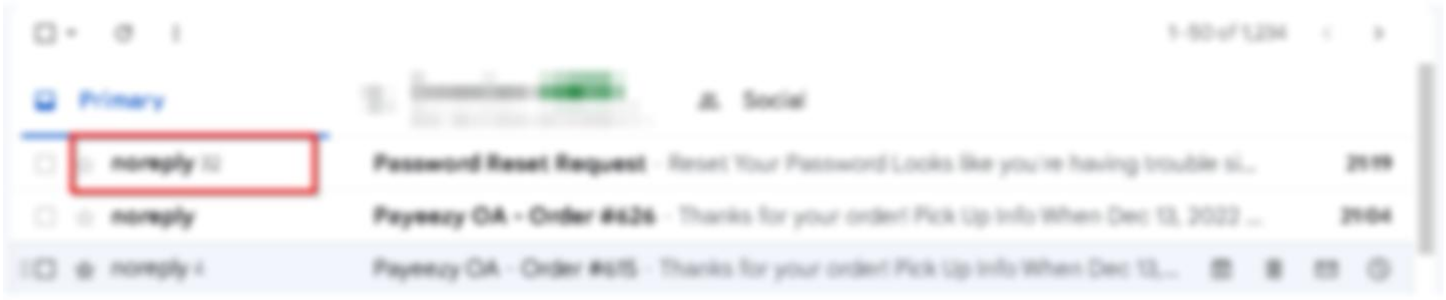


Figure 5: Intercepted request

Then, this request is sent multiple times simultaneously.

Figure 6: Response from multiple requests

As seen above, successful responses were obtained from those requests. And, after checking the email inbox of the email target, it was flooded by reset password emails.



noreply 32 Password Reset Request - Reset Your Password Looks like you're having trouble si... 21:19
noreply Payeezy OA - Order #626 - Thanks for your order! Pick Up Info When Dec 13, 2022 ... 21:04
noreply 4 Payeezy OA - Order #615 - Thanks for your order! Pick Up Info When Dec 13,...

Figure 7: Email inbox

Send receipt

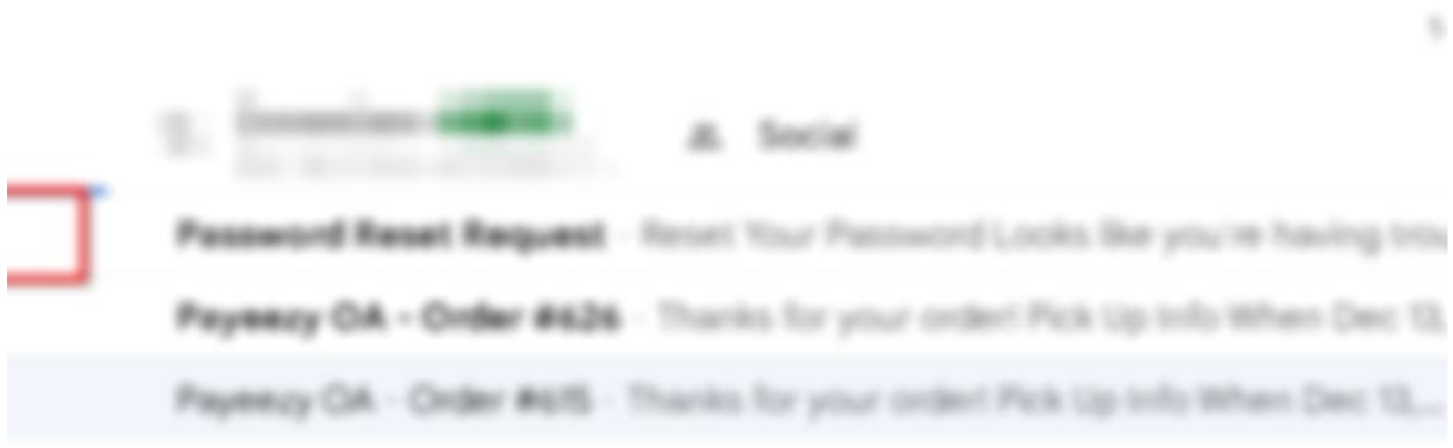
Relatedly, a similar issue occurs when repeating the same steps over the "send receipt" feature.

Figure 8: Payment info view

Now, after intercepting the corresponding request, multiple requests are performed simultaneously.

Figure 9: Responses from multiple requests

As can be seen above, this results in successful responses.



Password Reset Request - Reset Your Password Looks like you're having trou

Payeezy OA - Order #626 - Thanks for your order! Pick Up Info When Dec 13,

Payeezy OA - Order #615 - Thanks for your order! Pick Up Info When Dec 13,...

Figure 10: Email inbox

Leading to an email flood as multiple emails start taking space from the inbox.

Business impact

This type of attack might potentially cause financial loss as well as slow down services as this can consume a large amount of storage. However, a reputational cost is related to affected users if an email flood is caused by these features.

Recommendations

To mitigate the risk of this issue, the assessment team recommends the following steps:

- Set limiting rates on those requests to restrict the amount of allowed simultaneous attempts
- Implement a CAPTCHA mechanism to increase the difficulty to perform such requests in an automated manner

Locations & Occurrences (2)

- stage-order.dundermifflin.com/dundermifflin-api/v1/account
- stage-order.dundermifflin.com/dundermifflin-api/receipt/send/<id>

Resources

About Email Flooding

https://www.acunetix.com/support/docs/faqs/how-can-i-prevent-a-scan-from-causing-an-email-flood/

About rate limiting

https://developer.zendesk.com/documentation/ticketing/using-the-zendesk-api/best-practices-for-avoiding-rate-limiting/

Sensitive information exposed

LOW

Sensitive Data Exposure occurs when an organization unknowingly exposes sensitive data or when a security incident leads to the accidental or unlawful destruction, loss, alteration, or unauthorized disclosure of, or access to sensitive data. Such exposure may occur as a result of inadequate protection of a database, misconfigurations when bringing up new instances of data stores, and inappropriate usage of data systems, among others.

Details

After decompiling the IPA file, the assessment team found different pieces of sensitive information exposed.

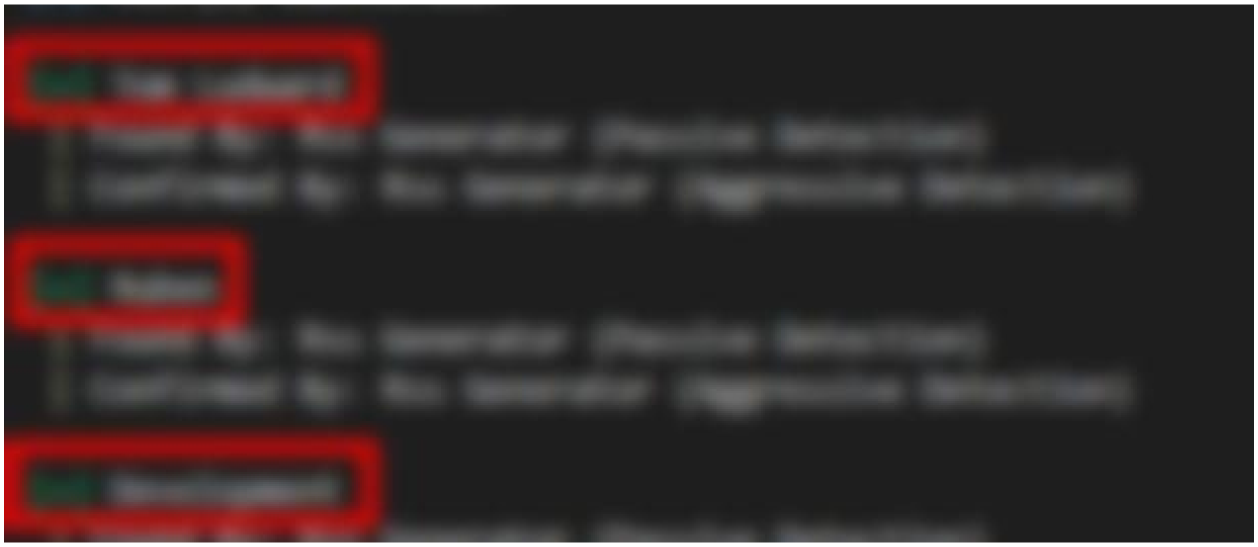


Figure 11: Exposed tokens

As can be seen above, information like API keys, database URLs, and storage bucket URLs are disclosed.

This issue can be found in the DEMO, DEV, PRD, STG, and UAT files.

Business impact

Such a situation put the in-scope assets at risk for two reasons: Disclosing an API key in a file within the app can allow an attacker to gain access to sensitive data and/or functionality as this could be used to perform unauthorized actions, such as accessing and modifying the app's data or making unauthorized purchases. In some cases, this vulnerability could also be used to steal the app's users' personal information or to perform other malicious actions on their behalf.

Recommendations

To mitigate the risk of this issue, the assessment team recommends the following steps:

- Store sensitive information in another service like a database
- Use encrypted copies of that information where the decryption key is securely stored as an environment variable out of reach of regular users

Locations & Occurrences (1)

- IPA file (iOS mobile app)

Resources

Sensitive data exposure

owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Security misconfiguration

LOW

Security misconfiguration occurs when security settings are not adequately defined in the configuration process or maintained and deployed with default settings. This might impact any layer of the application stack, cloud, or network. Misconfigured clouds are a central cause of data breaches, costing organizations millions of dollars.

Details

While analyzing the mobile application files the assessment team found two instances of security misconfigurations.

Weak permissions

The APK file is decompiled in order to analyze the component's behavior.

Figure 12: WebViewAppLinkResolver file

Here it can be seen that JavaScript is enabled within a webview component, this might allow an attacker to inject malicious code into the app, which could be used to steal sensitive information from the app or the device. In some cases, this vulnerability could even be used to install malicious apps on the device without the user's knowledge.

Export flag

While analyzing the AndroidManifest file a misconfiguration was identified.

Figure 13: AndroidManifest file

Multiple entities with the android:exported parameter set to true were found, such as activity, receiver, and service.

Business impact

This might potentially allow any app within the device, with access to the internet, to access the in-scope app's exposed components, such as its activities, services, and broadcast receivers. By doing so, an attacker could steal sensitive information or perform actions on behalf of the user without their knowledge.

Recommendations

To mitigate the risk of this issue, the assessment team recommends the following steps:

- Consider disabling JavaScript in the webview component, unless it is absolutely necessary for the app's functionality.
- If JavaScript is required, carefully control which web pages and scripts are allowed to run in the webview, and regularly review the app's code to ensure that it is secure.
- Set the android:exported flag to false for all components that do not need to be publicly accessible.
- Use a permission system to carefully control which other apps are allowed to access the app's components.

Locations & Occurrences (2)

- APK (WebViewAppLinkResolver file)
- APK (AndroidManifest file)

Resources

Improper Export of Android Application Components

cwe.mitre.org/data/definitions/926.html

About Android Manifest

https://developer.android.com/guide/topics/manifest/manifest-intro

Appendix (A) Severity Description

The assessment team used the following criteria to rate the findings in this report. Red Sentry derived these risk ratings from the industry and organizations such as OWASP.

Severity Descriptions

The severity of each finding in this report is independent. Finding severity ratings combine direct technical and business impact with the worst-case scenario in an attack chain. The more significant the impact, and the fewer vulnerabilities that must be exploited to achieve that impact, the higher the severity.

Critical

CRITICAL

Vulnerability is an otherwise high-severity issue with additional security implications that could lead to exceptional business impact. Examples: trivial exploit difficulty, business-critical data compromised, bypass of security controls, direct violation of communicated security objectives, and large- scale vulnerability exposure.

High

HIGH

Vulnerability may result in direct exposure including, but not limited to: the loss of application control, execution of malicious code, or compromise of underlying host systems. The issue may also create a breach in the confidentiality or integrity of sensitive business data, customer information, and administrative and user accounts. In some instances, this exposure may extend farther in the infrastructure beyond the data and systems associated with the application.

Medium

MEDIUM

Vulnerability does not lead directly to the exposure of critical application functionality,sensitive business and customer data,or application credentials. However, it can be executed multiple times or leveraged in conjunction with another issue to cause direct exposure. Examples include brute-forcing and client- side input validation.

Low

LOW

Vulnerability may result in limited exposure of application control, sensitive business and customer data, or system information. This type of issue provides value only when combined with one or more issues of a higher risk classification. Examples include overly detailed error messages, the disclosure of system versioning information, and minor reliability issues.

Informational

INFO

Finding does not have a direct security impact but represents an opportunity for additional layers of security, is considered a best practice, or has the possibility of turning into an issue over time. Finding is a security-relevant observation that has no direct business impact or exploitability, but may lead to exploitable vulnerabilities. Examples include poor communication between organizations, documentation encouraging poor security practices, or lack of security training.