**DWR 2000M 5G (CPE) Multiple Security Vulnerability**



**09-Feb-2024**

|  |  |
| --- | --- |
| **Submitted By** |  |
| **Vinod Kumar Shrimalii (mrnmap)**  **Nayab Keshodwala (NK)** |  |

**DWR 2000M 5G (CPE) Multiple Security Vulnerability**



**09-Feb-2024**

|  |  |
| --- | --- |
| **Submitted By** |  |
| **Vinod Kumar Shrimalii (mrnmap)**  **Nayab Keshodwala (NK)** |  |

**Findings**

* Command Injection Vulnerability in Dlink 5G CPE Diagnostics Utility
* Remote Code execution with admin privilege
* Bypassing Weak Brute Force Protection Mechanism
* Stored Cross Site Script
* Reflected Cross Site Scripting
* CSRF Cross Site Request forgery
* Insecure Telnet connection Allow Direct connection to device as Root.

**NAME OF AFFECTED PRODUCT(S) AND VERSION(S)]**

**Model:** DLink DWR 2000M 5G CPEWith Wifi 6 Ax1800

**Version:** DWR-2000M\_1.34ME

**Date: 09 Feb-2024**

**Description**

The new DWR-2000M 5G customer premise equipment (CPE) serves as a powerful networking focal point for homes and offices. The DWR-2000M CPE integrates LTE Advanced standard, which combines with leading-edge 5G to dramatically elevate connection speeds. Wi-Fi 6 innovation provides users with simultaneous and seamless access to voice, data and video content. Additional product specifications include.

1. **Stored Cross Site Scripting XSS HIGH**

A critical security vulnerability has been identified in the Dlink DWR 2000M router's handling of WiFi SSID names, resulting in the injection of malicious XSS payloads. Due to the lack of input validation on WiFi SSID fields, attackers were able to inject payloads such as "<img src=x onerror=print()>" which executed arbitrary JavaScript code upon rendering the router's homepage. This vulnerability poses significant risks as it allows attackers to execute malicious scripts in the context of other users' web browsers whenever they access the compromised page.

Cross-Site Scripting (XSS) is a common web security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can be used to steal sensitive information, such as session cookies or login credentials, manipulate page content, redirect users to malicious websites, or perform other unauthorized actions. XSS vulnerabilities typically arise when web applications fail to properly validate or sanitize user input, allowing attackers to inject code that is then executed by other users' browsers. There are three main types of XSS: stored XSS, reflected XSS, and DOM-based XSS, each with its own characteristics and attack vectors. Preventing XSS requires robust input validation, output encoding, and adherence to security best practices throughout the development lifecycle of web applications.

**Recommendation**

* Implement robust input validation and sanitization mechanisms for all user-provided data, including WiFi SSID names. Validate input against a whitelist of allowed characters and enforce strict length limits to prevent injection of malicious payloads.
* Ensure proper output encoding of user-generated content before rendering it in web pages. Encode special characters such as "<", ">", and "&" to prevent them from being interpreted as HTML
* Implement a Content Security Policy (CSP) to mitigate the impact of XSS attacks by specifying trusted sources for loading scripts, stylesheets, and other resources. Enforce restrictions on inline script execution to prevent the execution of injected payloads.
* Set HTTP security headers, such as X-XSS-Protection and X-Content-Type-Options, to enhance the security of web applications and prevent common attack vectors, including XSS attacks.

**Stored Cross Site Scripting XSS -POC**

* WiFI SSID field putting XSS payload <img src =x onerror=print ()>

A screenshot of a computer

Description automatically generated

* Same payloads execute at home page whenever user login.

A screenshot of a computer

Description automatically generated

**NAME OF AFFECTED PRODUCT(S) AND VERSION(S)]**

**Model:** DLink DWR 2000M 5G CPEWith Wifi 6 Ax1800

**Version:** DWR-2000M\_1.34ME

**Date: 09 Feb-2024**

**Description**

The new DWR-2000M 5G customer premise equipment (CPE) serves as a powerful networking focal point for homes and offices. The DWR-2000M CPE integrates LTE Advanced standard, which combines with leading-edge 5G to dramatically elevate connection speeds. Wi-Fi 6 innovation provides users with simultaneous and seamless access to voice, data and video content. Additional product specifications include.

1. **Stored Cross Site Scripting XSS HIGH**

A critical security vulnerability has been identified in the Dlink DWR 2000M router's handling of WiFi SSID names, resulting in the injection of malicious XSS payloads. Due to the lack of input validation on WiFi SSID fields, attackers were able to inject payloads such as "<img src=x onerror=print()>" which executed arbitrary JavaScript code upon rendering the router's homepage. This vulnerability poses significant risks as it allows attackers to execute malicious scripts in the context of other users' web browsers whenever they access the compromised page.

Cross-Site Scripting (XSS) is a common web security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can be used to steal sensitive information, such as session cookies or login credentials, manipulate page content, redirect users to malicious websites, or perform other unauthorized actions. XSS vulnerabilities typically arise when web applications fail to properly validate or sanitize user input, allowing attackers to inject code that is then executed by other users' browsers. There are three main types of XSS: stored XSS, reflected XSS, and DOM-based XSS, each with its own characteristics and attack vectors. Preventing XSS requires robust input validation, output encoding, and adherence to security best practices throughout the development lifecycle of web applications.

**Recommendation**

* Implement robust input validation and sanitization mechanisms for all user-provided data, including WiFi SSID names. Validate input against a whitelist of allowed characters and enforce strict length limits to prevent injection of malicious payloads.
* Ensure proper output encoding of user-generated content before rendering it in web pages. Encode special characters such as "<", ">", and "&" to prevent them from being interpreted as HTML
* Implement a Content Security Policy (CSP) to mitigate the impact of XSS attacks by specifying trusted sources for loading scripts, stylesheets, and other resources. Enforce restrictions on inline script execution to prevent the execution of injected payloads.
* Set HTTP security headers, such as X-XSS-Protection and X-Content-Type-Options, to enhance the security of web applications and prevent common attack vectors, including XSS attacks.

**Stored Cross Site Scripting XSS -POC**

* WiFI SSID field putting XSS payload <img src =x onerror=print ()>

A screenshot of a computer

Description automatically generated

* Same payloads execute at home page whenever user login.

A screenshot of a computer

Description automatically generated