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



**09-Feb-2024**

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

**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. **CSRF – Cros Site Request Forgery CRITICAL**

During a security assessment, it was discovered that the port forwarding page of the web application is vulnerable to Cross-Site Request Forgery (CSRF) attacks. This vulnerability exposes the port forwarding functionality to unauthorized manipulation by attackers, potentially leading to unauthorized access to internal resources or services

Several URLs under the /cgi-bin/luci/admin/network/ endpoint in the CPE network administration interface are vulnerable to Cross-Site Request Forgery (CSRF) attacks. These vulnerabilities expose the network configuration functionalities, including wireless settings, firewall rules, and network interfaces, to unauthorized manipulation by malicious actors.

**Recommendation**

* **Use Anti-CSRF Tokens**: Implement Anti-CSRF tokens as part of your web application's forms and requests. These tokens are unique per session and are included in each request. Upon receiving a request, the server validates the token to ensure it matches the expected value, thereby preventing CSRF attacks.
* **Same-Site Cookies**: Set the SameSite attribute on cookies to "Strict" or "Lax" to prevent cookies from being sent in cross-origin requests. This helps mitigate CSRF attacks by limiting the scope of cookies to the same site origin.
* **Implement Referer Header Checking**: Validate the Referer header in incoming requests to ensure that they originate from the expected domain. While not foolproof due to potential Referer header manipulation, it can provide an additional layer of protection against CSRF attacks.
* **Use CSRF Protection Frameworks**: Utilize CSRF protection frameworks provided by web application frameworks or security libraries. These frameworks often offer built-in mechanisms for generating and validating CSRF tokens, simplifying implementation and reducing the risk of vulnerabilities.
* **Strict Content Security Policy (CSP):** Implement a strict Content Security Policy (CSP) to mitigate the impact of XSS attacks, which can be leveraged to execute CSRF attacks. A strict CSP can help prevent the execution of injected scripts and limit the attack surface.

**CSRF-POC**

* Port forwarding option in router.
* Create port forwarding request and capture the same.

**A screenshot of a computer screen

Description automatically generated**

* Creating CSRF payload for the same with different port and send to user.

A screenshot of a computer program

Description automatically generated

* CSRF payload execution

A screenshot of a computer

Description automatically generated

* In Result we can see multiple Port are open

A screenshot of a computer

Description automatically generated