hackerhood
BY RED HOT CYBER

“Keep your chin up, someday there will be happiness in Nottingham again. You'll see. (Robin Hood)

Disclosure report

Revisions table

Version	Researcher	Discovery date
1.0	Alessandro Sgreccia	03/12/2024

Who we are

Within the Red Hot Cyber community, the "HackerHood" initiative takes shape. It is a group of ethical hackers passionate about dissemination and sharing, who have agreed to raise awareness of cyber risk, through a series of practical initiatives.

Hackerhood, carries out bug hunting activities on a heterogeneous series of IT products in order to raise awareness of risk and to improve the ecosystem in which we live.

Legal note

Vendor Info

- **Organization Name:** ZYXEL
- **Web Page:** https://www.zyxel.com
- **Email:** security@zyxel.com.tw
- **Vulnerability Disclosure Info Web Page:** https://www.zyxel.com/global/en/support/security-advisories

Credits and Research Team Info

- **First Name**: Alessandro
- **Last Name:** Sgreccia (Member of Hackerhood Hacking Group)
- **Research Firm**: Hackerhood
- **Organization Name**: RedHotCyber https://www.redhotcyber.com

Disclosure Policy

We strongly believe that a coordinated disclosure is the best approach to properly and efficiently address the risk related to security vulnerabilities (i.e. Coordinated Vulnerability Disclosure – CVD).

If everything goes as intended, after your confirmations and, eventually, the CVE ID publication, we will proceed with a full disclosure on our Web page. If you do not agree with a full disclosure for the vulnerabilities, please let us know by responding to this communication. In this case we will just publish the CVE details.

However, if no response is provided or you do not intend to take any action to assess the security issue, we will proceed as follows:

After the first communication with no response within a week, it is resent. If no response is provided at all, we will proceed with a disclosure of the vulnerability on our public Web Site after 90 days.

After the acknowledgement of the security issues, if no status updates are provided within the next month, we will send you a final communication warning that the vulnerability information will be published after 90 days.

As a security research team, we will be glad to support you in the evaluation and remediation processes.

Best regards

Hackerhood hacking team

1. CWE-285: Improper Authorization

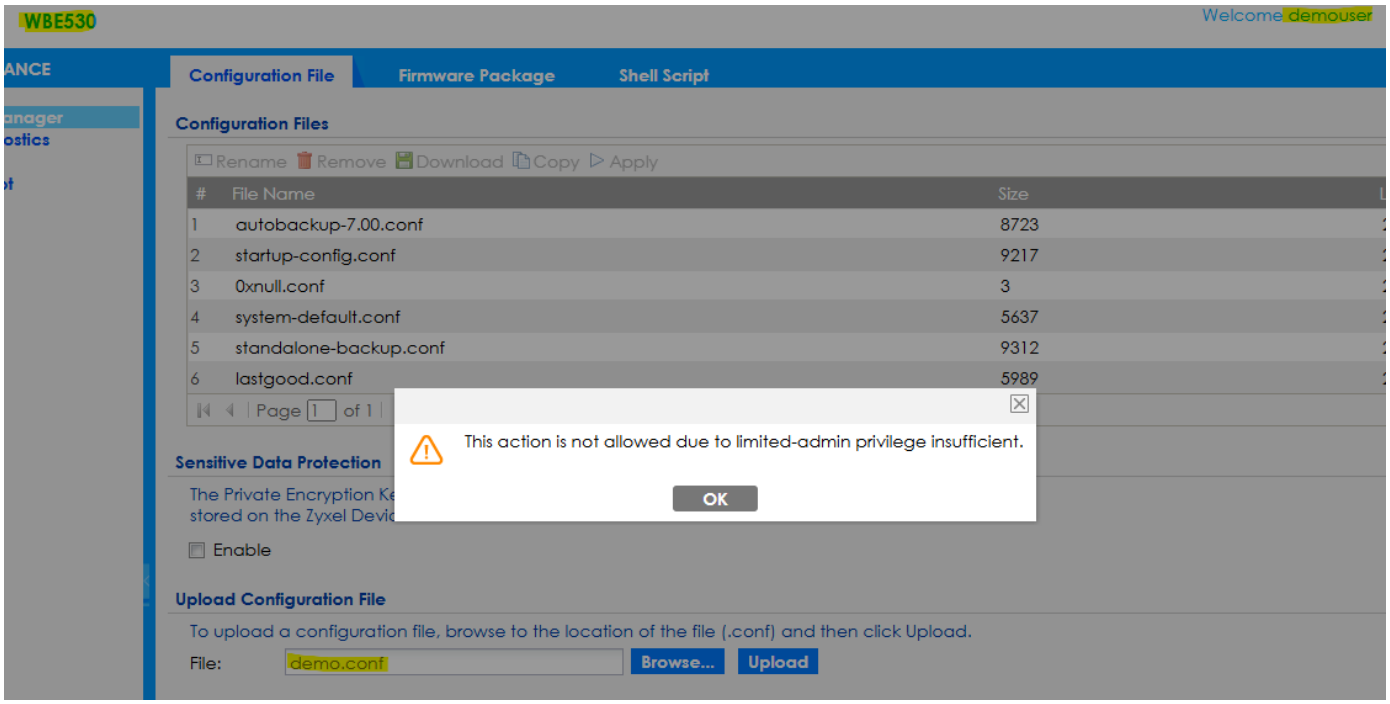
- **Product Line:** Tested on WBE530/WBE660S
- **Vulnerable Version:** V7.00(3) (Tested on **on-premise** mode)
- **Summary:** The File Upload CGI script does not check user privileges.
- **Prerequisites:** Compromised limited-admin user

Step-by-Step Instructions and Proof of Concept (PoC)

1. A user with a "limited-admin" account can log in via the Management Web Interface.
2. Once logged in, the user can attempt to upload a custom configuration file.
3. If successful, this operation can result in the user gaining full administrative access to the device, bypassing the intended privilege restrictions.

Observed Behavior

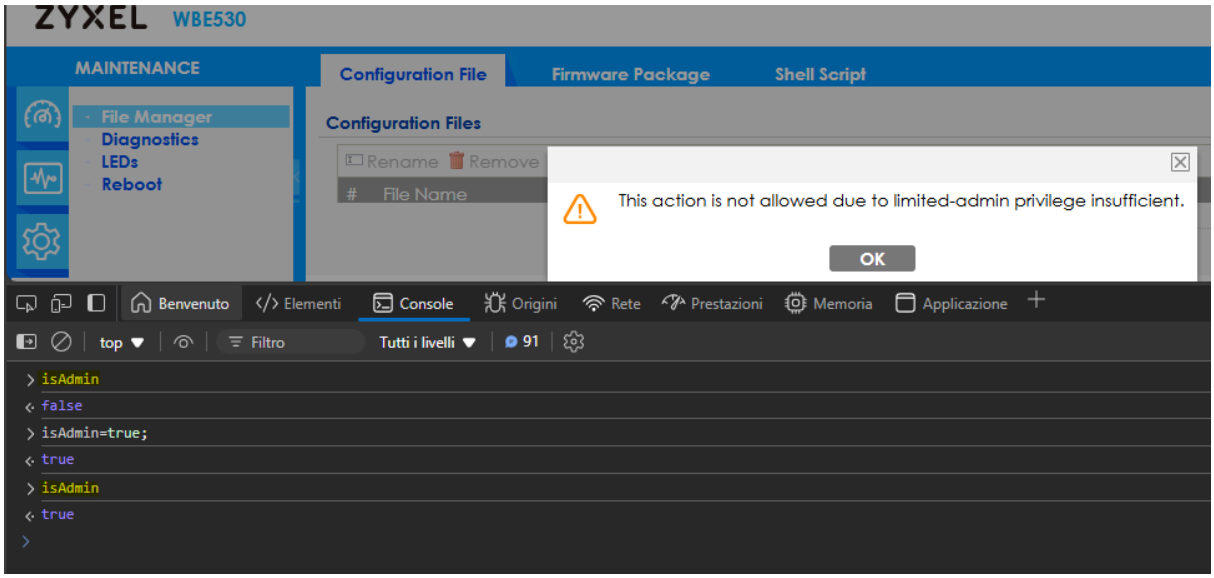
When a user with "limited-admin" privileges attempts to upload a custom configuration file, a popup is displayed, informing the user that the upload is prohibited due to insufficient privileges. However, this restriction can be bypassed under certain conditions, allowing unauthorized administrative access.



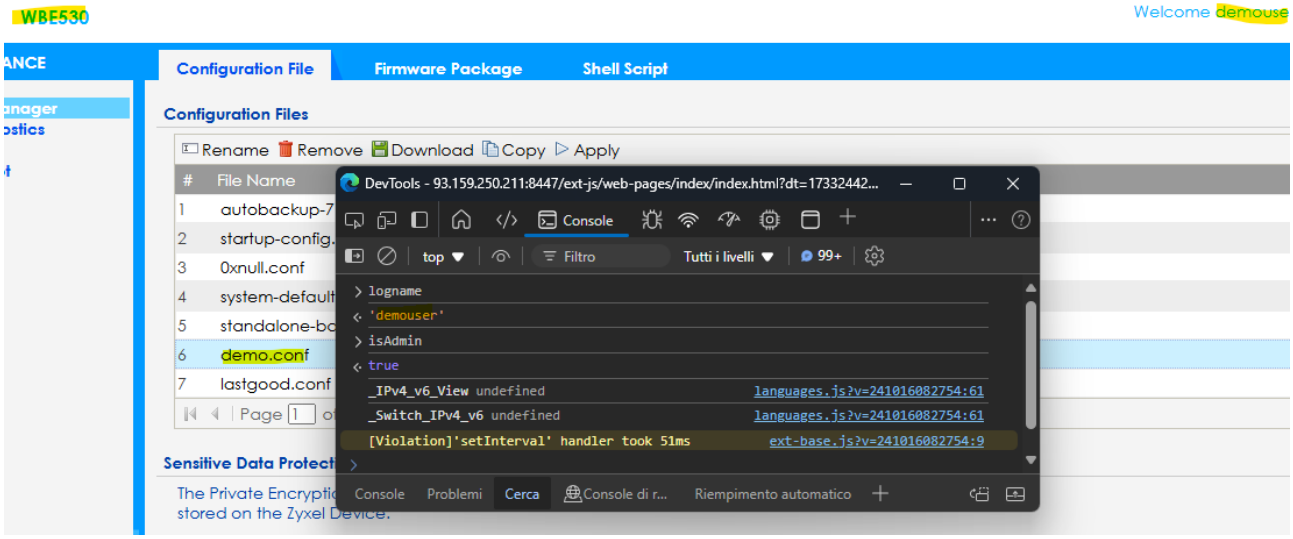
This popup is managed by the script **"ZyFunction.js"**, located in **"/ext-js/common/"**, which checks the local variable **"isAdmin"**. If the variable is set to **"False"**, the warning message is displayed.

```
2061  var dumpMsgIfLimitedAdmin = function() {
2062      if (!isAdmin) {
2063          Ext.MessageBox.show({
2064              msg: "This action is not allowed due to limited-admin privilege insufficient.",
2065              buttons: Ext.MessageBox.OK,
2066              icon: Ext.MessageBox.WARNING
2067          });
2068      }
2069      return !isAdmin;
2070  }
```

However, by manipulating "isAdmin" variable, it is possible to attempt the upload, bypassing the client-side check.



The CGI script does not verify the identity of the requester, allowing a limited user to upload customized configuration files.



If the limited user attempts to upload the **"startup-config.conf"**, the device will reboot and load the provided configuration, thereby allowing the attacker to gain full control of the device.

Recommendations for Mitigating the Vulnerability

1. **Server-Side Validation:**
   Implement strict server-side validation to verify the identity and permissions of the user before processing any configuration file upload requests. Ensure that the server checks the user's role (e.g., isAdmin) independently of client-side controls.
2. **Restrict File Uploads:**
   Limit the ability to upload configuration files to users with verified administrative privileges. Reject all upload attempts from accounts with restricted or limited permissions.
3. **Input Sanitization:**
   Validate and sanitize all uploaded files to ensure they meet expected formats and contain no unauthorized modifications or commands.
4. **Enhanced Logging and Monitoring:**
   Implement detailed logging for file upload attempts, including the source IP, user role, and timestamp. Regularly review logs for unauthorized or suspicious activity.
5. **Client-Side Security:**
   While client-side checks can enhance user experience, they should not be relied upon as the sole mechanism for security. Ensure all critical validation occurs server-side.