EXECUTIVE SUMMARY

Northwave conducted a comprehensive code-based penetration test on CyberDrain's software stack, comprising of an open-source PowerShell backend and a React frontend. The objective was to identify potential vulnerabilities that could be exploited by malicious entities. Four vulnerabilities were identified, consisting of 3 minor issues and one medium-severity finding, which was swiftly addressed by CyberDrain's development team prior to integration into a release.

FINDINGS SUMMARY:

MINOR FINDING 1

Inconsistent implementation of secure communication protocols was noted in the Azure Function PowerShell backend. The function app had an implementation that allowed access via FTP instead of FTPS or sFTP. This would require credentials and users to actively use the FTP settings. After implementation considerations we've found that the default settings for the CIPP platform require CI/CD and this issue is not an issue for anyone running the recommended and default implementation, still we advise to change the settings to use sFTP in case access via this method is ever required.

MINOR FINDING 2

Hardcoded values were identified in the React frontend which allowed users to adapt the frontend system by editing their fork, and as such have the frontend send incorrect results. This is to be expected of an open source application but our recommendation is to add advisement and comments to the code on changes that could lead to dangerous situations, such as editing the Tenants table.

MINOR FINDING 3 (RESOLVED)

Certain sections of the PowerShell backend exhibited inefficient error handling, particularly within the 'Standards' module. Detailed error messages could unintentionally expose sensitive system information.

MEDIUM FINDING (RESOLVED)

Insufficient input validation was detected in the ListTenants module of the PowerShell backend, potentially enabling command injection attacks. Swift action from CyberDrain led to the resolution of this issue before release into the production environment.

OPEN SOURCE APPLICATION NOTE

CyberDrain leverages open source tools within its PowerShell backend, a practice that can provide robust functionality but also comes with the risk of inherited vulnerabilities. It is important to regularly check for updates and patches from open-source maintainers.

RECOMMENDATIONS

- Enforce the consistent use of HTTPS across all backend functions to ensure secure data transmission.
- Substitute hardcoded credentials with secure secret management systems, such as Azure Key Vault or AWS Secrets Manager.
- Employ standardized error handling mechanisms that prevent disclosure of detailed system information.
- Regularly validate all user-generated input to prevent command injection or similar attack vectors.

CONCLUSION

CyberDrain's prompt response to the medium-severity issue illustrates their commitment to maintain a secure codebase. Nevertheless, to solidify their security posture, it is advised to address the minor issues identified during the assessment. Regular penetration testing, along with diligent updating and patching of open-source tools, will help CyberDrain continue to offer a secure and reliable service to their users.