Vulnerability Assessment

Prepared for

Federal Aviation Administration

Lynne A. Osmus , Acting Administrator

David M. Bowen, Assistant Administrator for Information Services and Chief Information Officer

Executive Summary

Our team was asked to do a security vulnerability assessment on the FAA's flight control system. During the course of our investigation, we were able to accomplish the following:

- Gain access to and modify the table used to populate the flight tracking system
- View potentially sensitive information regarding your web server configuration
- Gain administrator access to a Windows 2000 SP4 web server

As a result of our investigation, we have compiled a list of recommendations for your organization to consider implementing in the future, which are enclosed in this packet. The recommendations are divided into Critical and Moderate level recommendations. Critical recommendations need to take priority when patching the systems against the vulnerabilities our team was able to identify, as they allow for a remote attacker to do considerable damage to your real-time systems. Moderate recommendations should be easier to implement and could still provide a remote attacker valuable information, but are not time-critical in fashion.

Thank you for allowing us to assess the security of your network. Any questions regarding our recommendations can be forwarded to any team member for further explanation.

Thank you,

Team 2

Critical Actionable Recommendations

We have identified the following vulnerabilities that require immediate attention from a security officer to prevent critical system damage from being inflicted by a remote attacker.

Recommendation #1: Validate User Input from Forms to Prevent SQL Injection

Filename: lookup.php
Host: 129.82.138.40
Vulnerability: Script is Vulnerable to SQL Injection from remote attacker

Problem

The script automatically processes input provided by the user without first checking to see if the data meets a certain criteria. This allows for a remote attacker to take control of the database via a very simple SQL injection. Once the remote attacker has control of the database, they could be able to add invalid flights, modify the tables, delete the entire tables, or insert a script that could redirect users to a malicious site. In essence, an attacker could render the entire flight tracker useless.

Possible Solution

Verification can be enabled to ensure that the flight names entered into the form match criteria that all flight identifiers meet. From our research, we have found that the length of the identifier tends to be 6 characters or less. A good starting point for validation would be to ensure that all input is constrained to 6 characters or less. Also, it would be good to ensure that non-alphanumeric characters are not allowed to be given as input. Restricting the input to a-z, A-Z, and 0-9 would prevent an attacker from using SQL to gain control of the system.

Recommendation #2: Enforce Secure / Strong Password Use Organization-Wide

We were able to gain root access to a machine on the 10.0.0.0/8 network by using a default password for the operating system used. It is vital for your organization to enforce secure strong passwords that are difficult to guess. An attacker gaining root access to an internal machine poses a significant threat to your organization.

Recommendation #3: Update and Patch All Systems, Especially Public Addresses

The Windows 2000 SP4 server at 129.82.138.16 is unpatched. This means that there are open vulnerabilities for at least one of the running services. We were able to gain administrator access to the system by using Metasploit to manipulate the vulnerabilities. These security holes have been disclosed for years and can easily be secured by keeping the system software up to date. Some of these services do not seem to have a use and could even be disabled entirely.

Moderate Actionable Recommendations

Recommendations labeled as moderate should be addressed in order to harden the security of your systems. These vulnerabilities do not provide an attacker with access to your systems, but could assist them in the event of a larger scale attack.

Recommendation #1: Disable Error Reporting in PHP

It is recommended that Error Reporting be disabled in the PHP configuration for all systems that are running in a live environment. Error Reporting should only be enabled for testing purposes prior to deployment. Displaying errors to website viewers allows those viewers to see potentially sensitive configuration information and could give them insight which allows them to more easily execute an attack.

Recommendation #2: Disable Default Apache Error Pages

The default error page in Apache displays the version of Apache being used by the web server. This can provide valuable information to an attacker because outdated versions often have specific vulnerabilities that can be targeted. If the attacker already knows the specific version of Apache being run on a target, the attacker knows the specific vulnerabilities which will be successful.

Recommendation #3: Upgrade Apache to Latest Stable Version

Your web server at 129.82.138.40 is currently running Apache version 2.2.6. There have been security vulnerabilities identified with this version of Apache that could allow a remote attacker to gain access.

Recommendation #4: Create Key-based Access Control List for SSH Connections

Instead of allowing all hosts to connect to the web server via SSH, it would be ideal to authorize specific host keys that have a need to have remote access. This would limit the possibility of a remote intruder connecting to the server via SSH.

Recommendation #5: Use Difficult-to-Guess Table Names in Database

Having table names in a database that are easy to guess provide an attacker with an easier method of gaining access to the information in those tables. In your current situation when an SQL injection is possible, difficult table names make attacks more difficult.