Path Traversal in Upload Functionality in AgeVolt Portal prior to version 0.1

Short Description

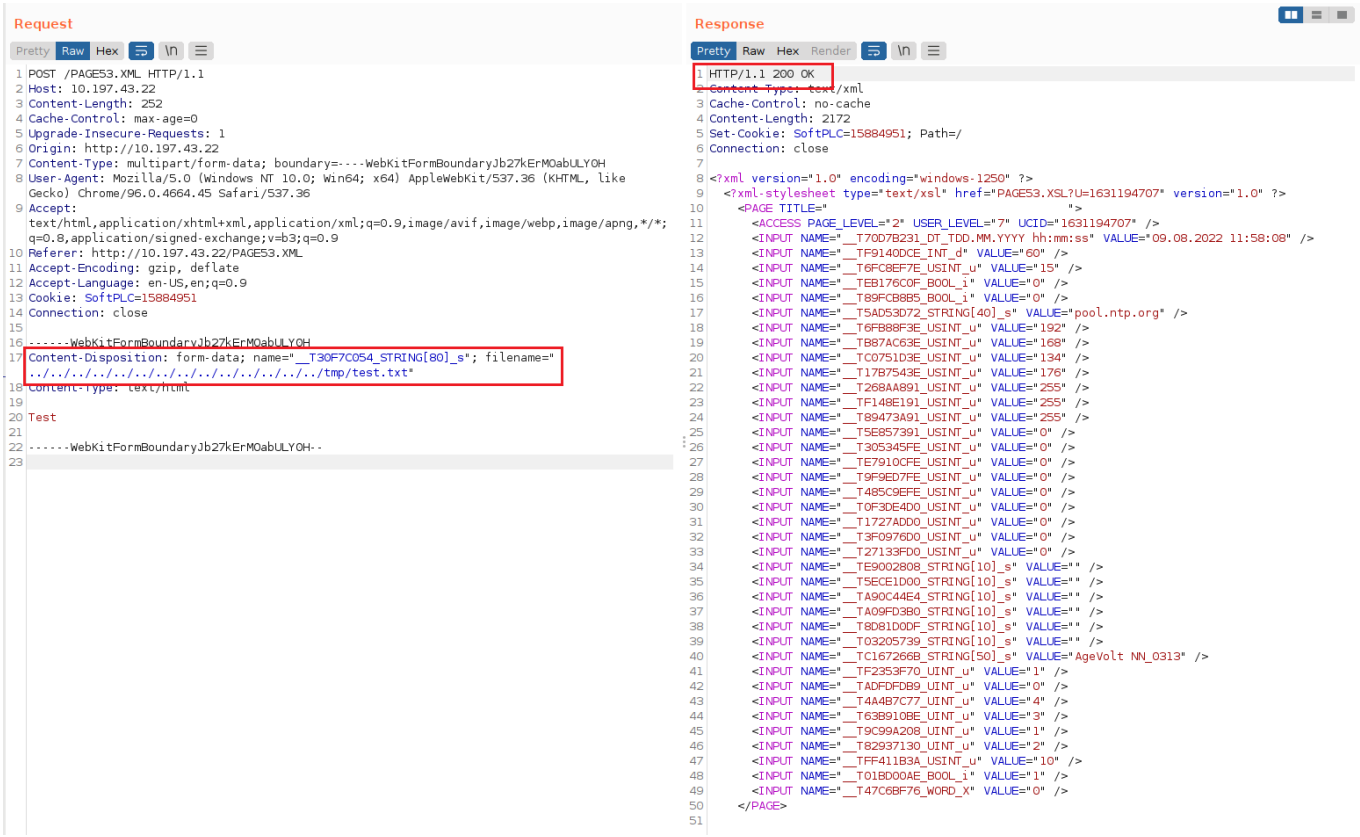
An arbitrary file upload and directory traversal vulnerability (CVE-2022-38484) exist in the file upload functionality of the System Setup menu in AgeVolt Portal prior to version 0.1. A remote authenticated attacker could leverage this vulnerability to upload files to any location on the target operating system with web server privileges.

CVSS:3.1 score = 9.1 (Critical)

Details

Path traversal, also known as directory traversal or directory climbing, is a security vulnerability that occurs when an attacker can manipulate file paths to gain unauthorized access to files or directories outside the intended scope of an application. In the context of an upload functionality, path traversal would allow an attacker to upload a file and then manipulate the file path to access or overwrite sensitive system files or directories, potentially leading to unauthorized access, data disclosure, or even code execution. This vulnerability is a significant security risk.

Below is a request that is vulnerable to a path traversal attack. Specifically, the problem is in the filename parameter, where the full path to the filesystem can be inserted.



In the screenshot above you can see the successful upload of the text file to the path `/tmp/test.txt`

If the web server is running under root privileges (which was the case in our example), this vulnerability can lead to a complete compromise of the entire server (RCE = Remote code execution) by, for example, adding a public key to ssh `authorized_keys` and creating a new user in `/etc/passwd`. The attacker can then simply log in via ssh to the compromised server.

Mitigation

- **Input Validation and Sanitization**:
  - Implement strict input validation to ensure that uploaded file names or paths only contain allowed characters and conform to a predefined format.
  - Sanitize user input to remove or escape characters that could be used for path traversal, such as ".." or "../../."
- **Use Whitelists**:
  - Maintain a whitelist of allowed file types and extensions for uploads. Only permit files that match the whitelist.
  - Apply strict file type checks to ensure that the content of uploaded files matches their declared type.
- **File Upload Directory Restrictions**:

- Store uploaded files in a dedicated, restricted directory that is separate from sensitive system files. This reduces the risk of unauthorized access to critical files.