Summary

A path traversal vulnerability in the mass delete handler allows ar
outside the application's root directory by supplying traversal sec

Vulnerability

The mass delete handler iterates over the `file[]` POST array and a
sanitization:

```php
$files = $_POST['file'];
if (is_array($files) && count($files)) {
    foreach ($files as $f) {
        if ($f != '') {
            $new_path = $path . '/' . $f;   // $f is never passed t
            if (!fm_rdelete($new_path)) {
                $errors++;
            }
        }
    }
}
```

`fm_rdelete()` then operates on the fully resolved path with no bou

```php
function fm_rdelete($path) {
    if (is_link($path)) {
        return unlink($path);
    } elseif (is_dir($path)) {
        // recursively deletes everything inside
    } elseif (is_file($path)) {
        return unlink($path);
    }
}
```

Other file operation handlers (rename, single delete, copy) pass th
mass delete handler does not, making it the only unprotected path.

Proof of Concept

Setup — create a file outside the web root:

```bash
echo del > /tmp/test.txt
cat /tmp/test.txt
```