RocketLane.com Vulnerability Report

October 27nd, 2022

Prepared for the benefit of: RocketLane.com's security team
By: Mark Rudnitsky

Notices and disclaimers: This is an independent security evaluation for RocketLane, shared as per industry-standard Responsible Disclosure guidelines. This report is confidential, for distribution only among RocketLane's technical staff. The researcher attests that no exploits were conducted besides proof-of-concept validation; any exploits actually leading to negative impact for RocketLane or its users were not conducted.

Stored Cross-Site Scripting (XSS) via Profile Images

OWASP Category: A03:2021 - Injection

Affected host: https://[customer-name].api.rocketlane.com

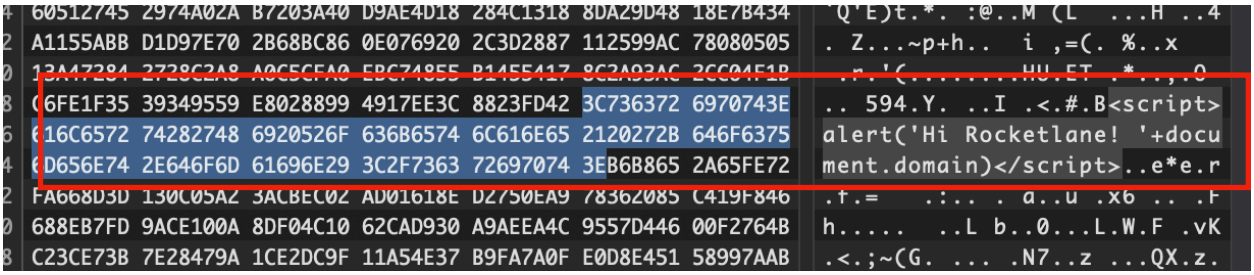
Vulnerable URL: /api/v1/users/photo

Example Payload: `<script>alert('Hi Rocketlane! '+document.domain)</script>`

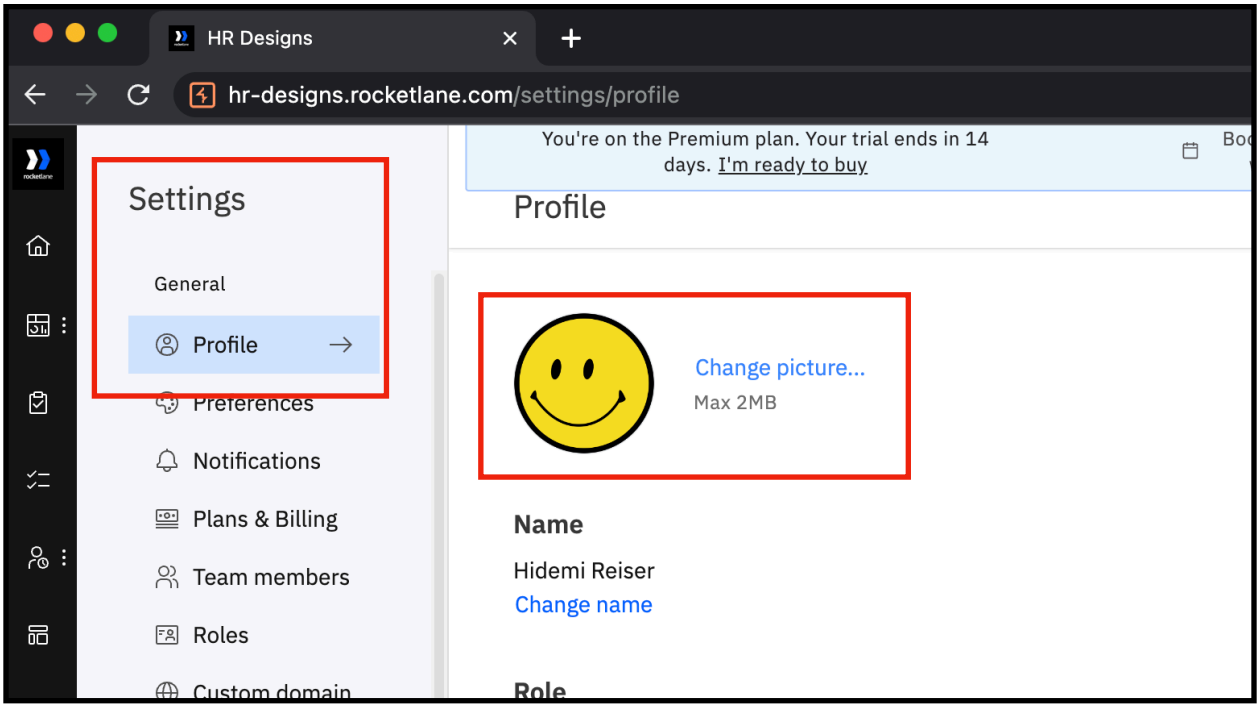
Steps to Replicate:

You will need Burp Suite to replicate this finding.

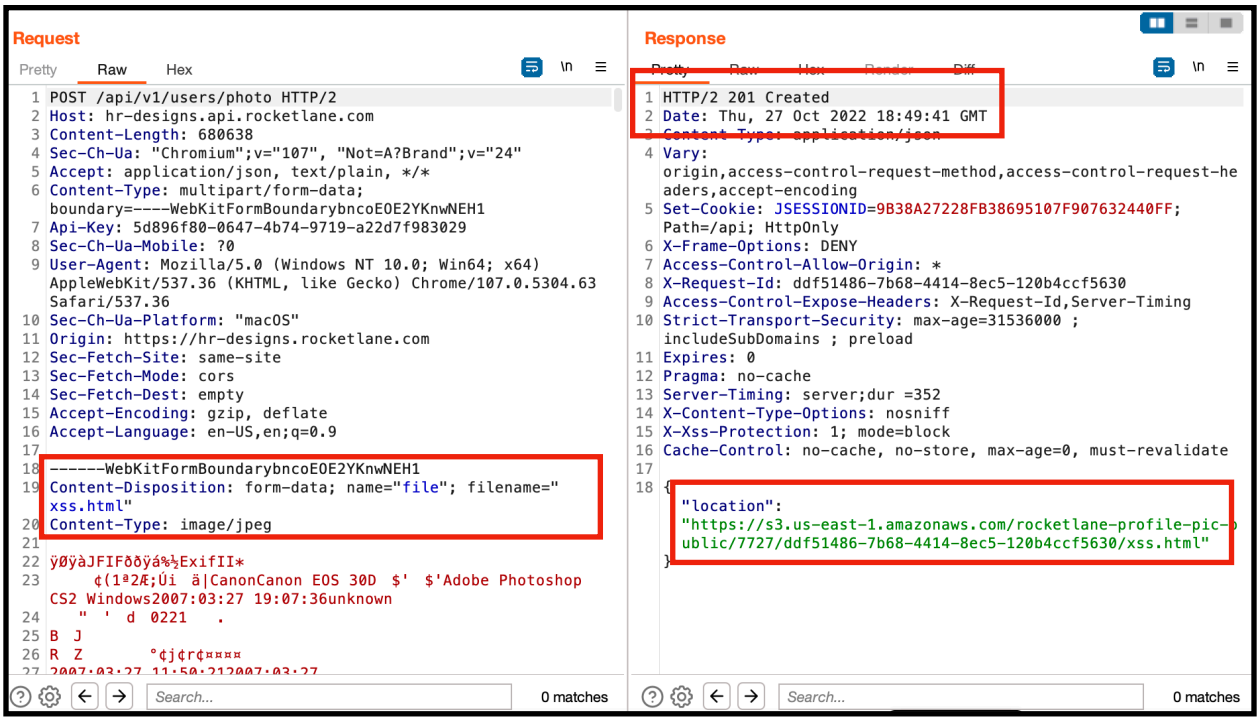
1. Create a RocketLane account as per normal.
    1. For the purposes of this report, the Researcher created the user account "https://hr-designs.rocketlane.com/". The POC is still available there for RocketLane's viewing pleasure.
2. Find an arbitrary JPG image and download it.
    1. **Reference file**: "xss.jpg" of some currency.
3. Using a hex editor such as Hex Fiend, insert the following code snippet into the image.
    1. **Hex:** 3C736372 6970743E 616C6572 74282748 6920526F 636B6574 6C616E65 2120272B 646F6375 6D656E74 2E646F6D 61696E29 3C2F7363 72697074 3E
    2. **ASCII:** <script>alert('Hi Rocketlane! '+document.domain)</script>



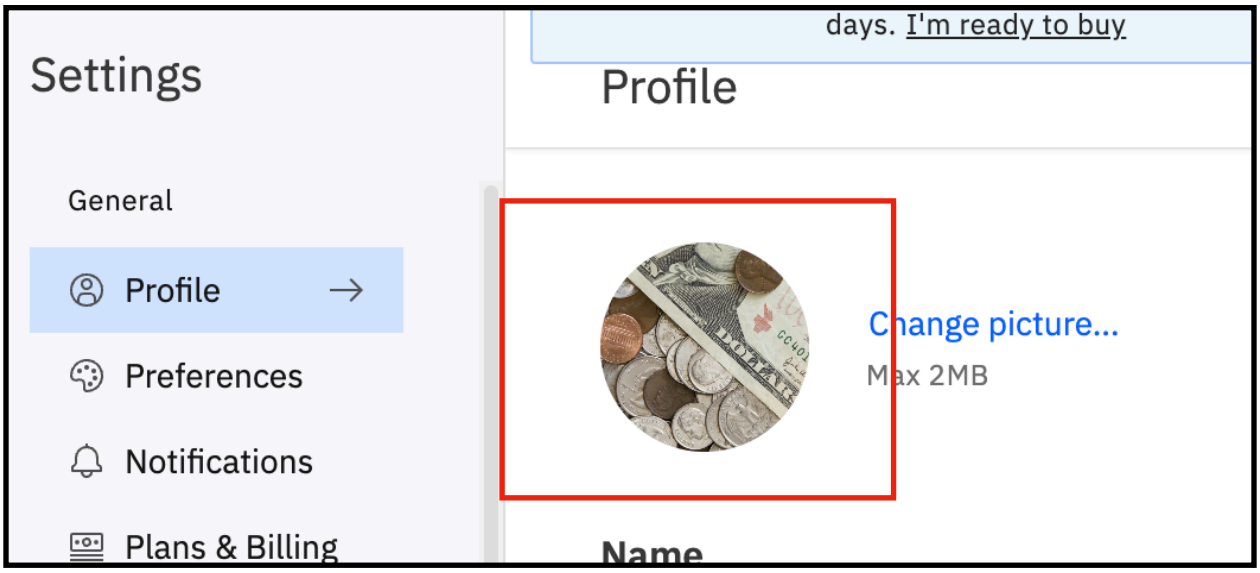
4. Save the image.
5. Navigate to Settings > General > Profile, then click “Change picture …”.



6. Turn on Burp’s Intercept feature.
7. Intercept the POST request to the URL: `/api/v1/users/photo`
8. In Burp Suite’s Intercept field, scroll down to the `Content-Disposition` and `Content-Type` header fields.
9. Modify the `filename` parameter’s extension to “`.html`”. Leave the other fields unchanged.

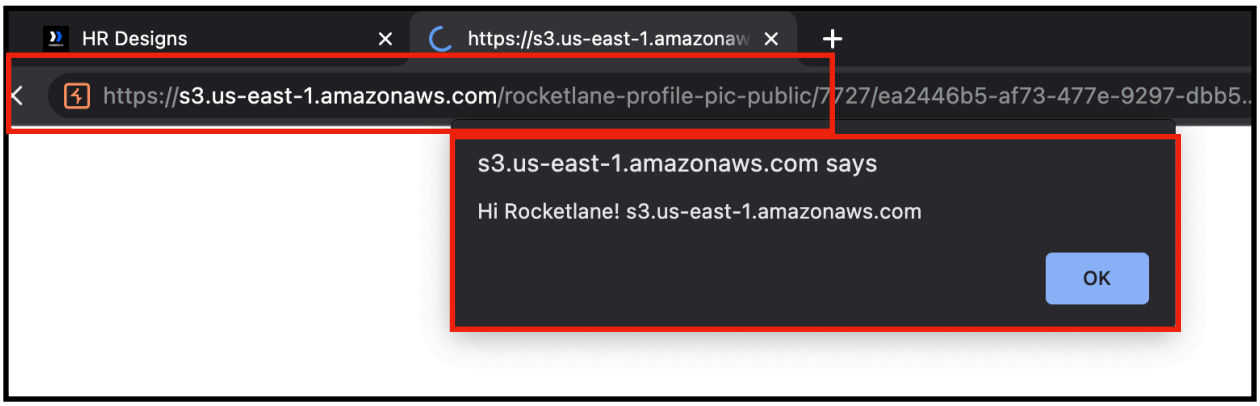


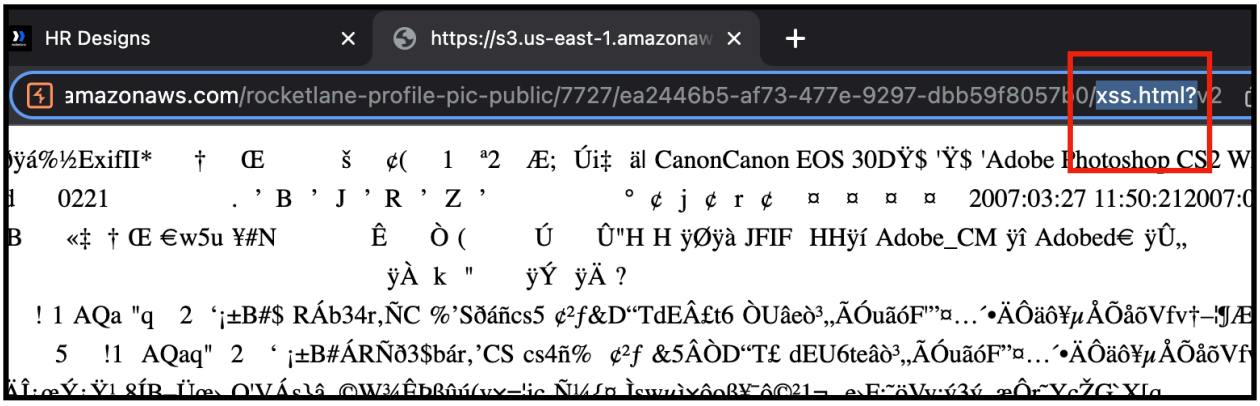
10. Forward the request and turn off Intercept (for convenience).
11. Observe that the crafted image is properly uploaded and rendered after a delay.



12. Right click on the image and select “Open Image in New Tab”.
13. Observe your XSS payload pops.

1. **Researcher's note**: Note in the URL that the profile is "`rocketlane-profile-pic-public`" and the filename is successfully saved as "`xss.html`".





Likelihood: Moderate. There are no protections in place against this attack, but the attacker must have at least basic knowledge of how to conduct hex editing and how to modify requests via Burp Suite or a similar tool.

Impact: Extreme. An attacker can consistently pop this stored XSS on anyone the attacker can convince to view the image as per the process above. This payload can easily be modified to show the results of `document.cookie`, indicating that the victim's cookies can be stolen.

Further, since the website allows requests from arbitrary origins, an attacker who has access to a server on which they can host malicious JavaScript can take this further, setting the `Origin` header to their malicious domain, exfiltrating cookies to the server and taking over accounts via valid cookies at will.

Suggested remediation:
- Analyze all images on the hex level and prohibit the use of any characters that can be used for XSS, such as `0x3c` ("<"), `0x3e` (">"), `0x22` (double quotes), `0x27` (single quote), and `0x3d` ("=").

- Perform server-side validation to ensure that no unwanted extensions can be used and reject all requests that try.
- Do not allow requests from arbitrary origins.

Further reading:
- https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html