Bitwarden Network Security Assessment Report

ISSUE SUMMARIES, IMPACT ANALYSIS, AND RESOLUTION

BITWARDEN, INC

Table of Contents

Summary

In May 2023, Bitwarden engaged with cybersecurity firm Cure53 to perform penetration testing and assessment against Bitwarden IPs, servers,and web applications. A total of 9 days were invested to reach total coverage needed for Bitwarden.

The overall impression from Cure53 is that Bitwarden, including the network infrastructure and web applications that power the product, exhibits a strong security foundation with zero exploitable vulnerabilities found. Only one issue was discovered, which was promptly resolved by the Bitwarden team.

This report was prepared by the Bitwarden team to cover the scope and impact of the issue found by the Cure53 team and expected resolution steps. For completeness and transparency, a copy of the summary delivered by Cure53 has also been attached to this report.

Issue

BWN-07-001 WP2: Potential SSRF via DNS rebinding

It was discovered that the icons-service utilized as part of Bitwarden's vault feature is abusable for Server-side request forgery (SSRF). There have been stringent measures incorporated by Bitwarden to protect against SSRF exploits that attempt to reach private IP address spaces and cloud metadata services, but it can be bypassed by DNS rebinding technique, leading to blind SSRF.

Resolution

Status: Issue has been fixed post-assessment.

Icon fetching service has been rewritten, by resolving the IP once, validating it, then using that resolved IP to pull data directly instead of looking up the domain name again, to prevent the DNS timing attack.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Miscellaneous Issues

This section covers any and all noteworthy findings that did not incur an exploit but may assist an attacker in successfully achieving malicious objectives in the future. Most of these results are vulnerable code snippets that did not provide an easy method by which to be called. Conclusively, whilst a vulnerability is present, an exploit may not always be possible.

BWN-07-001 WP2: Potential SSRF via DNS rebinding *(Medium)*

Cure53 noted that the *icons*-service utilized as part of Bitwarden's vault feature is abusable for SSRF. This service is intended to fetch favicons from newly-created vault items that contain a URL. This approach, however, evokes SSRF susceptibility by design, since arbitrary URLs are fetchable.

Ample evidence attests to the stringent measures incorporated by Bitwarden to protect against SSRF exploits that attempt to reach private IP address spaces and cloud metadata services. However, studies performed against the underlying source code verified the complete absence of security shielding against DNS rebinding attacks, observable via the following lines of the *icons*-service's source code.

Affected file:
server/src/Icons/Services/IconFetchingService.cs

Affected code:

```
private async Task<HttpResponseMessage> GetAsync(Uri uri)
{ [...]
    // Resolve host to make sure it is not an internal/private IP address try
    {
        var hostEntry = Dns.GetHostEntry(uri.Host);
        if (hostEntry?.AddressList.Any(ip => IsInternal(ip)) ?? true)
        {
                                                return null;
                                            }
  [...]

      using (var message = new HttpRequestMessage())
      {
          message.RequestUri = uri;
  [...]
```

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

The first highlighted line in the code snippet above resolves the provided URL. Subsequently, the *IsInternal()* check is run to ensure this does not point to any private or otherwise blacklisted address spaces. Next, the actual HTTP request is initiated during the third and subsequent highlighted lines. The issue here pertains to the fact that the original DNS entry for the provided URL between the highlighted lines can expire and return a new, private IP that points to *localhost*, for instance. As such, the check described above is bypassable.

To reproduce this erroneous behavior via the *icons*-service, the following request chain can be performed:

Steps to reproduce:

1. Fetch the *icons*-service with a provided URL:

   Example payload:
   https://icons.bitwarden.net/attacker1.mmap.space/icon.png

2. On *http://attacker1.mmap.space/favicon.ico*, the service responds with an HTTP redirect such as the following:

   Example response:
   ```
   HTTP/1.1 302 Found
   Server: nginx/1.14.0 (Ubuntu)
   Date: Thu, 25 May 2023 07:55:09 GMT
   Content-Type: text/html; charset=UTF-8
   Transfer-Encoding: chunked
   Connection: keep-alive
   Location: http://178.62.204.220-1-127.0.0.1-1-a.rebindc.dd.h4x.tv/xyz
   ```

3. Note that the service under *178.62.204.220-1-127.0.0.1-1-a.rebindc.dd.h4x.tv* now resolves to *178.62.204.220* for the first DNS request (to satisfy the *IsInternal()* check) and to *127.0.0.1* on the second DNS request, when the final HTTP query is performed. Thus, this process succeeds in bypassing the implemented SSRF protection.

As corroborated by a number of publicly reported cases, this type of TOCTOU vulnerability has proven challenging to resolve via the code. Typically, stricter networking rules on the egress layer are implemented to prevent requests to unexpected resources. However, since Bitwarden provides a self-hosted configuration in addition, the developer team should consider resolving this issue in the codebase directly.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

An effective solution here would be to utilize HTTP transport dial callbacks to verify the final destination address before sending the request, provided the underlying framework or HTTP client supports callbacks of this nature.

In fact, Go provides optimal mitigation for this fault via the HTTP transport *Dial func*[4], which is utilized in packages that are specifically designed to neutralize this threat vector, such as *safeurl*[5] .