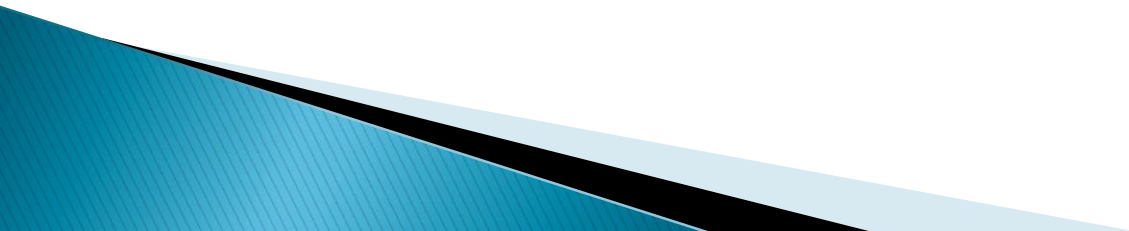


Without commitment you never start but more importantly without consistency you never finish.

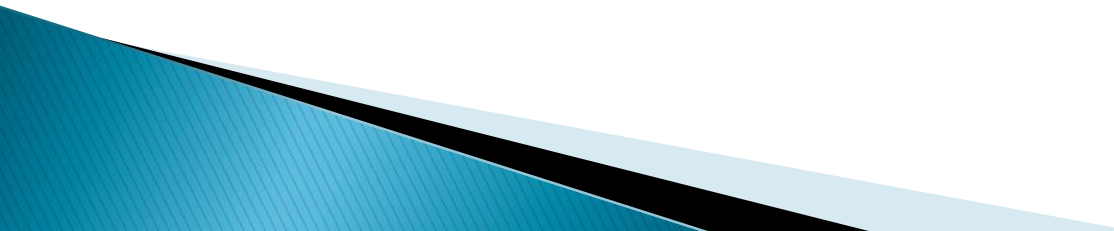


# **How I got CVE-2019-13655 for DOS by Buffer overflow**


# root@kali:~# whoami

Name: Dipak Prajapati

Msc. Cyber security student working passively as a bug bounty hunter on online bug hunting platforms like Bugcrowd.com with rank in top 320, Google VRP with rank in top 280, Microsoft Security Response Center's bounty program, National Critical Information Infrastructure Protection Centre of india's RVDP etc.



# What we are going to discuss:

- 1) Summary of how I came to the vulnerability.
    - how I reached to the specific API.
    - how I get to know potentially there is DOS by buffer over flow via pixel flood.
  - 2) Technical details.
    - Vulnerability, Impact ,Type
    - DOS is not bulk requesting.
    - why server was taking 55000+ millisecond to response for one GET request.
  - 3) How I was able to exploit this.
  - 4) How I registered the CVE ID.
  - 5) Difference between bug bounty hunter and web application penetration tester.
- 

# Summary of how I came to the vulnerability.

## Technical details.

IMGIX is a web image CDN (Content delivery network) provider which processes and optimizes images also.

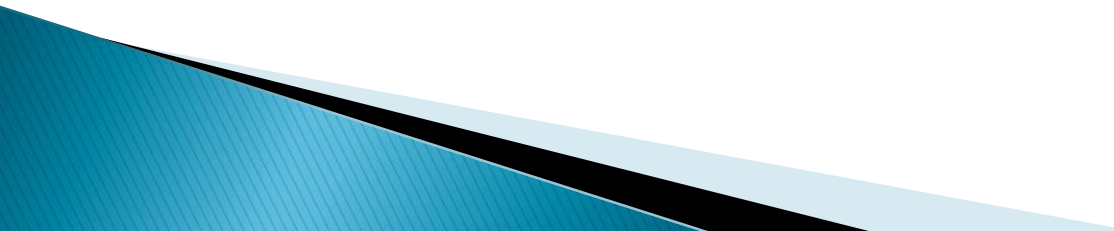
**Vulnerability:** buffer over flow

**Impact:** DOS

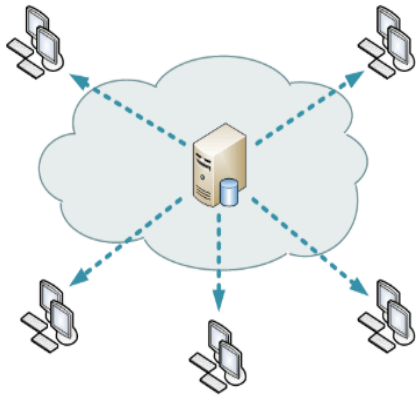
**Type:** Pixel flood

I was invited for testing a web application from [bugcrowd.com](https://bugcrowd.com), during testing the web application I found that the web application is using a 3rd party service **IMGIX** for their CDN (where they save uploaded profile pictures).

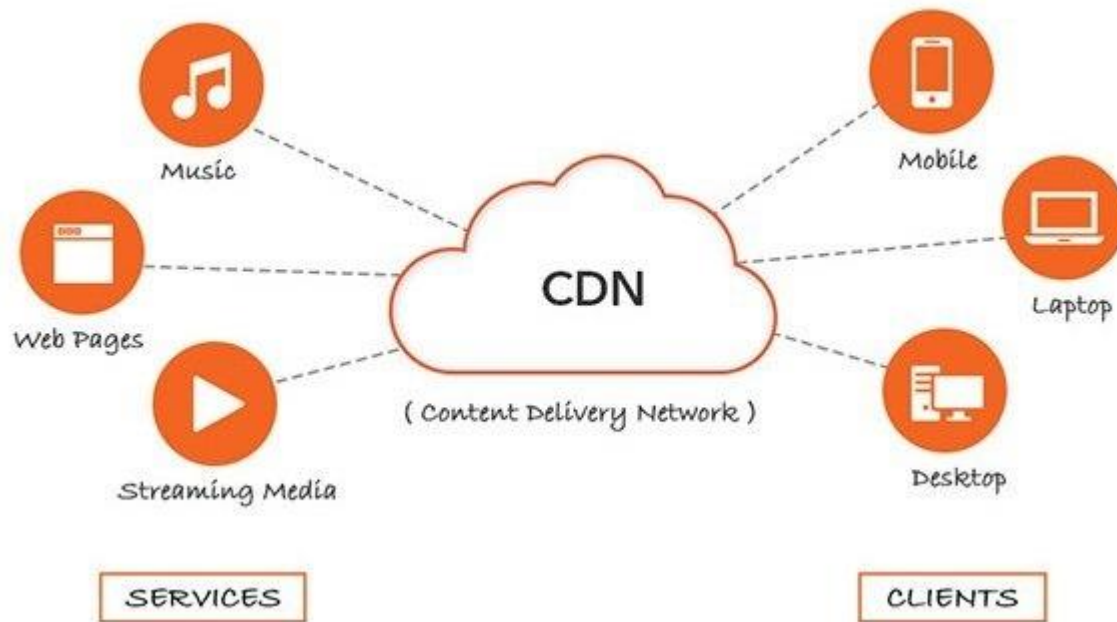
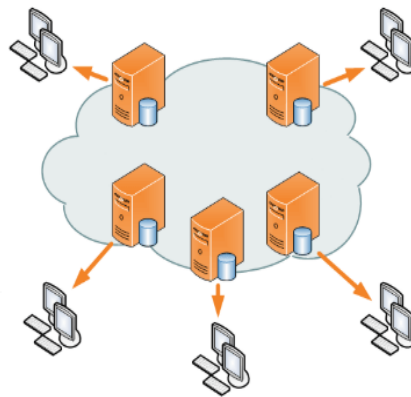
Now for my curiosity I was never heard of this CDN service provider, I start poking it with uploading insecure files and images.



## Before CDN



## After CDN



1) I uploaded a crafted image in the IMGIX CDN.

**description of Image:**

I have an image of 5kb, 260x260 pixels. In the image itself I exchange the 260x260 values with 0xfafa x 0xfafa (so 64250x64250 pixels).

2) IMGIX saved the image uploaded by me successfully.

3) Now while calling the image from CDN this happened

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

15 x 16 x 17 x 18 x 19 x 20 x 21 x 22 x 23 x 24 x 25 x 26 x 27 x 28 x 29 x 30 x ...

Go Cancel < >

Request

Raw Params Headers Hex

GET /b4b3e4e6-b127-4b12-9808-1d80bbca5d2f?fit=crop&w=90&h=90 HTTP/1.1  
Host: [REDACTED]avatar.imgix.net  
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:67.0) Gecko/20100101 Firefox/67.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: close  
Upgrade-Insecure-Requests: 1

? < + > Type a search term 0 matches

Done

Target: https://[REDACTED]avatar.imgix.net

Response

Raw Headers Hex

HTTP/1.1 500 Internal Server Error  
Server: imgix-fe  
Cache-Control: public,max-age=300  
X-Imgix-ID: ba1c9ca8e90da8aaf6f2e014ef2f2de9a0187ddf  
[REDACTED]  
Connection: close  
Accept-Ranges: bytes  
Content-Type: text/plain  
Access-Control-Allow-Origin: \*  
X-Content-Type-Options: nosniff  
X-Served-By: cache-lax8622-LAX, cache-bom18223-BOM  
X-Cache: MISS, MISS  
Content-Length: 0

From here I got to know  
about 3rd party service  
imgix CDN

Response time in  
millis



? < + > Type a search term 0 matches

406 bytes | 55,171 millis




# Why server is giving such big latency:

A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold. A buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers. Writing outside the bounds of a block of allocated memory can corrupt data, crash the program, or cause the execution of malicious code.

–OWASP

## How I got to know potentially there is DOS by buffer over flow via pixel flood:

while calling the image (GET request) IMGIX tried to process the image and loaded the whole image into memory, it tried to allocate 4128062500 pixels into memory which floods the buffer allocated by the process and server was not able to handle the exception so, It gave latency of 55000–66000 milli seconds and this much latency means a lot.



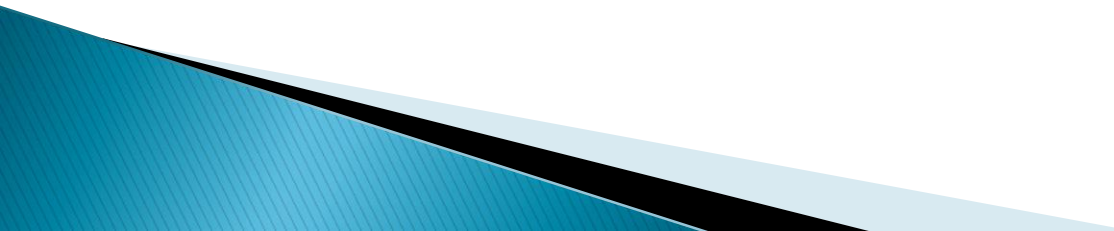
# How I was able to exploit this.

Now giving 55000 milli seconds latency in response will not harm the user of IMGIX or the server resources. So I made this scenario of attack

- 1) Enrol with IMGIX and buy image CDN services.
- 2) upload too many crafted images having 64250x64250 pixels.
- 3) Gather all calling URLs of uploaded images.
- 4) Make a Script or use a tool which call all the URLs simultaneously in multi threads.
- 5) If server gave 55000–66000 milli seconds latency while calling one image then too many calls of different images simultaneously can definitely take the services down for enough time.

# DOS and DDOS does not means sending too much request to a target system:

The Denial of Service (DoS) attack is focused on making a resource (site, application, server) unavailable for the purpose it was designed. There are many ways to make a service unavailable for legitimate users by manipulating network packets, programming, logical, or resources handling vulnerabilities, among others. If a service receives a very large number of requests, it may cease to be available to legitimate users. In the same way, a service may stop if a programming vulnerability is exploited, or the way the service handles resources it uses.



# Types of DOS and how much I like to find DOS

File Edit View History Bookmarks Tools Help

Your Elastic Security Team, beti X



Bugcrowd Inc. (US) | https://bugcrowd.com/payments



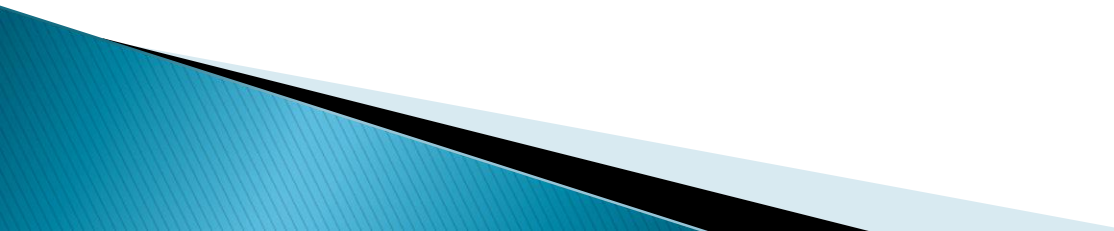
Dashboard Programs Submissions **Payments** Leaderboard

## Remitted payments (\$ [REDACTED])

Download CSV

Program		Rewarded For	Amount	Date Rewarded
[REDACTED]		Add companies to victim's account by IDOR / After malicious act attacker can fake identification of existing account	\$100.00	[REDACTED]
[REDACTED]	★	Bypass of Application-Level Denial-of-Service (DoS) via adding number of user without limitation on server	\$300.00	[REDACTED]
[REDACTED]	★	App level DOS attack by Adding Custom apps	\$500.00	[REDACTED]
[REDACTED]	★	DOS to other users by Adding Attributes	\$500.00	[REDACTED]
[REDACTED]	★	DOS to other users by bricking questions	\$500.00	[REDACTED]
[REDACTED]	★	Application level DOS by overloading action plan's JSON variable in server memory	\$150.00	[REDACTED]
[REDACTED]	★	Application-Level Denial-of-Service (DoS) via company-request	\$900.00	[REDACTED]
Pinterest	★	Pixel flood attack	\$200.00	[REDACTED]
[REDACTED]	★	Application-Level Denial-of-Service (DoS) via creating huge amount of group	\$500.00	[REDACTED]
[REDACTED]	★	Application-Level Denial-of-Service (DoS) via generating huge amount of tokens	\$500.00	[REDACTED]
[REDACTED]	★	Application-Level Denial-of-Service (DoS) via adding a unlimited templates on the API server	\$500.00	[REDACTED]
[REDACTED]	★	Application-Level Denial-of-Service (DoS) via adding a number of folders on the server	\$500.00	[REDACTED]
Researcher Operations Budget (Private)		BountySlayersProgram_Q12019	\$300.00	[REDACTED]
Pinterest		Injecting links, data uri, js files, collecting IPs and many more.	\$200.00	[REDACTED]
[REDACTED]		Captcha Bypass in Account Verification	\$100.00	[REDACTED]
[REDACTED]	★	Application level DOS via Resource Depletion by creation huge amount of sources/destinations	\$200.00	[REDACTED]

# How I registered the CVE ID.

- 1) I tried to find any CVE ID related to this vulnerability in IMGIX.
  - 2) Then I got to know that this vulnerability in this service/server software is not registered.
  - 3) I filled the CVE ID request form and they replied me on 18 Jul 2019 with a CVE ID: CVE-2019-13655.
    - The CVE Assignment Team will examine the given details of vendor and vulnerability found in it, after resolving the vulnerability they give a ID.
  - 4) Then I filled CVE ID Publication form with a blog for public description.
  - 5) CVE Assignment Team Published my CVE ID on 30 Jul 2019.
- 

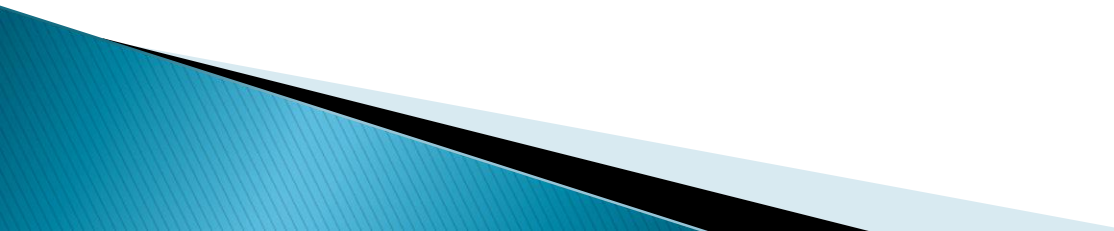
# Difference between bug bounty hunter and Penetration tester:

Bug bounty hunter:

A bug bounty hunter always focus on critical vulnerabilities which can reward him/her good amount of bounties.

Penetration tester:

A Penetration tester focus on Securing the application, He/she has to test low impact vulnerability and high impact vulnerability both.



# THANK YOU

Ease is the biggest threat to progress. Don't wish for easy situations, wish to become better.

