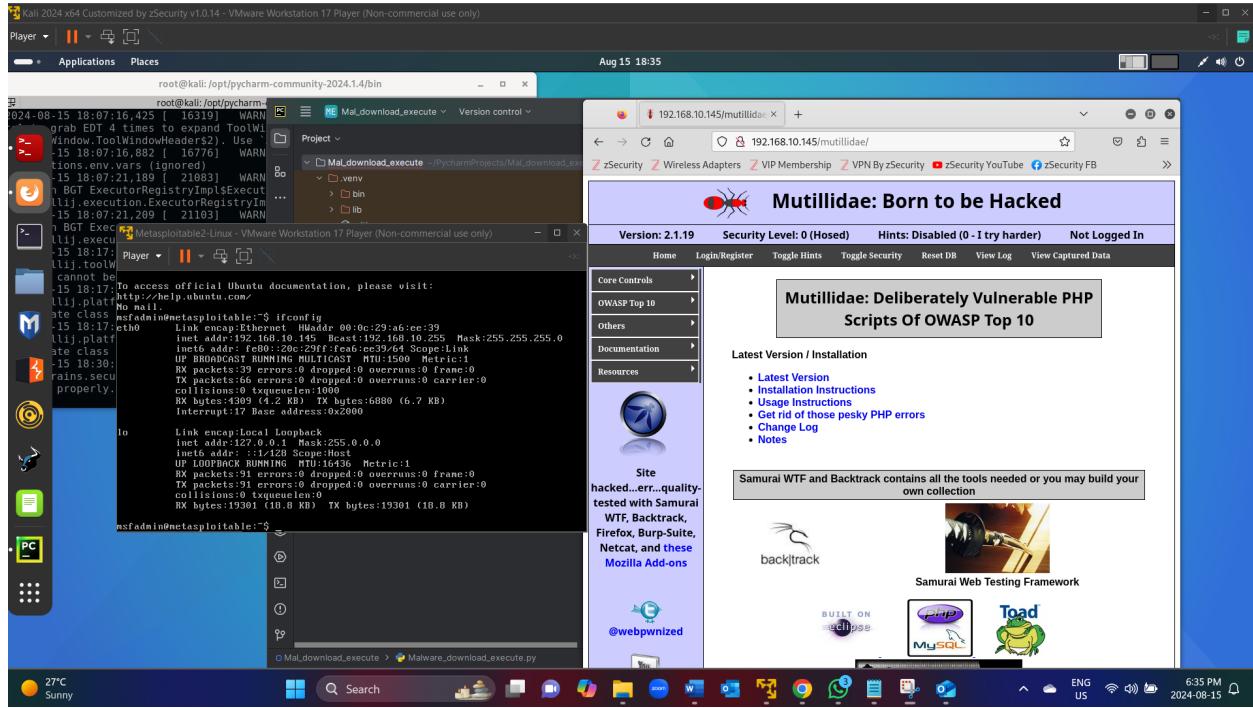


WEBSITE/ WEB APPLICATION HACKING:

Download Metasploitable:



Website Hacking – Writing a Crawler:

Sending GET Requests To Web Servers:

```
#!/usr/bin/env python
```

```
import requests
```

```
url = "google.com"
```

```
get_response = requests.get("http://" + url)
```

```
print(get_response)
```

The screenshot shows a Kali Linux desktop environment. In the center, there's a PyCharm interface with two files open: 'crawler.py' and 'downloads.py'. The 'crawler.py' file contains the Python code shown above. A warning message in the bottom right corner of the PyCharm window says: 'External file changes sync might be slow. PyCharm cannot receive filesystem event notifications for the project. Is it on a network drive?' Below the PyCharm window, a terminal window is running with root privileges. The command 'python3 crawler.py' is being run, and the output shows a successful response from the Google website. The desktop bar at the bottom shows various icons and system status.

```
root@kali:~/PycharmProjects/Crawler# python3 crawler.py
<Response [200]>
root@kali:~/PycharmProjects/Crawler#
```

Kali 2024 x64 Customized by zSecurity v1.0.14 - VMware Workstation 17 Player (Non-commercial use only)

Player Applications Places Aug 15 18:49

root@kali:~/PycharmProjects/Crawler# python3 crawler.py

```
root@kali:~/PycharmProjects/Crawler# python3 crawler.py
[200]
<Response [200]>
root@kali:~/PycharmProjects/Crawler#
```

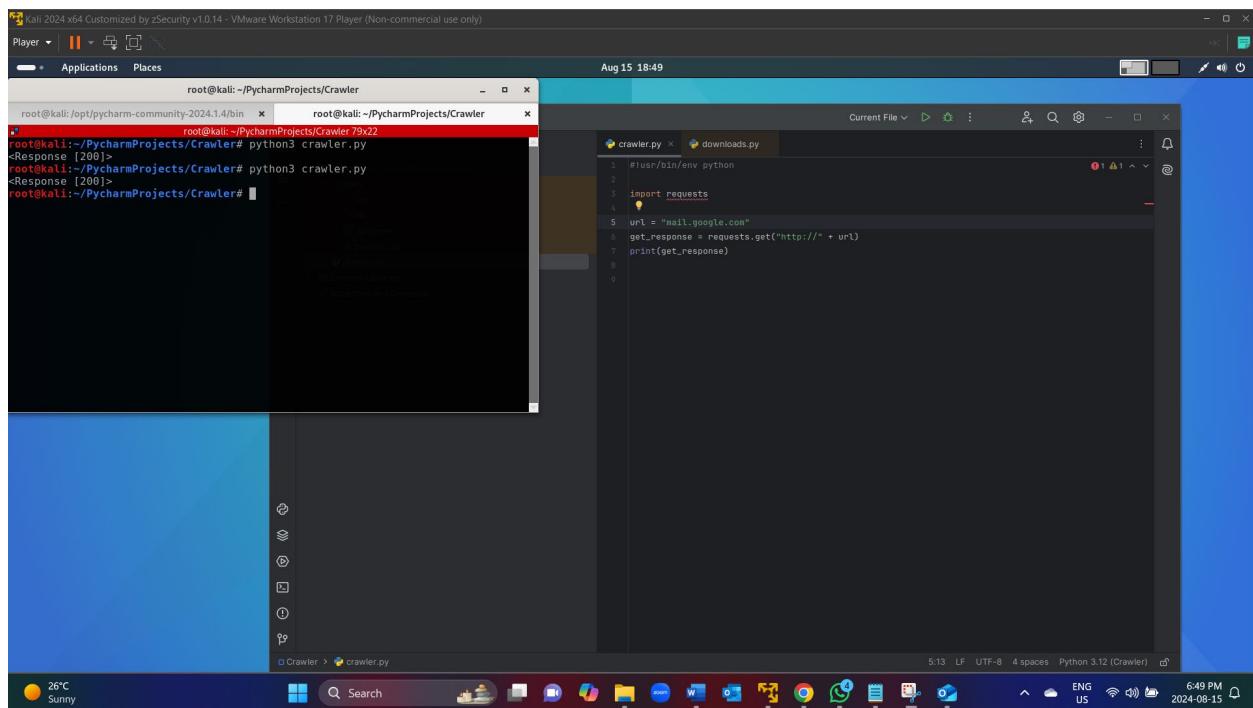
PyCharm Projects Crawler

crawler.py x downloads.py

```
1 #!/usr/bin/env python
2
3 import requests
4
5 url = "mail.google.com"
6 get_response = requests.get("http://" + url)
7 print(get_response)
```

5:13 LF UTF-8 4 spaces Python 3.12 (Crawler) ⚡

26°C Sunny 6:49 PM 2024-08-15



Kali 2024 x64 Customized by zSecurity v1.0.14 - VMware Workstation 17 Player (Non-commercial use only)

Player Applications Places Aug 15 18:50

root@kali:~/PycharmProjects/Crawler# python3 crawler.py

```
root@kali:~/PycharmProjects/Crawler# python3 crawler.py
get_response = requests.get("http://" + url)
File "/usr/lib/python3/dist-packages/requests/api.py", line 73, in get
    return request('get', url, params=params, **kwargs)
File "/usr/lib/python3/dist-packages/requests/api.py", line 59, in request
    return session.request(method=method, url=url, **kwargs)
File "/usr/lib/python3/dist-packages/requests/sessions.py", line 589, in request
    resp = self.send(prep, **send_kwargs)
File "/usr/lib/python3/dist-packages/requests/sessions.py", line 703, in send
    r = adapter.send(request, **kwargs)
File "/usr/lib/python3/dist-packages/requests/adapters.py", line 519, in send
    raise ConnectionError, request=request
requests.exceptions.ConnectionError: HTTPConnectionPool(host='mailnnnffsf.google.com', port=80): Max retries exceeded with url: / (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7fb2a33651d0>: Failed to establish a new connection: [Errno -2] Name or service not known'))
```

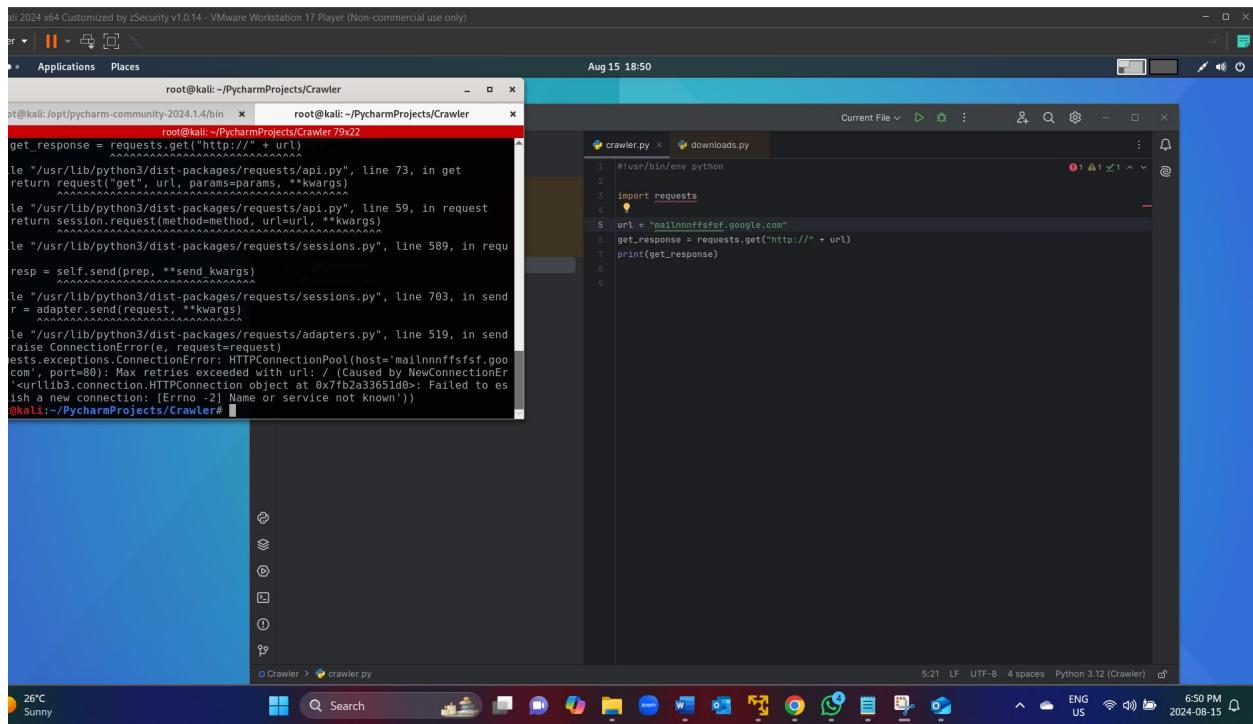
PyCharm Projects Crawler

crawler.py x downloads.py

```
1 #!/usr/bin/env python
2
3 import requests
4
5 url = "mailnnnffsf.google.com"
6 get_response = requests.get("http://" + url)
7 print(get_response)
```

5:21 LF UTF-8 4 spaces Python 3.12 (Crawler) ⚡

26°C Sunny 6:50 PM 2024-08-15

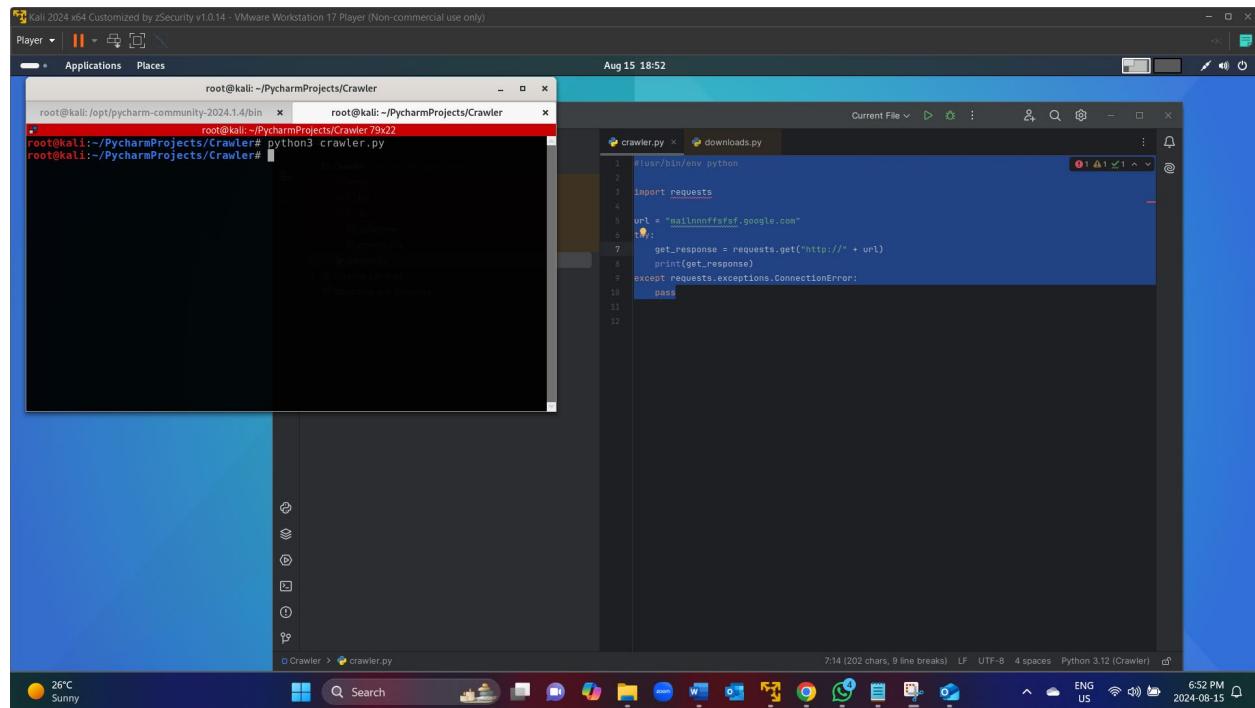


```
#!usr/bin/env python

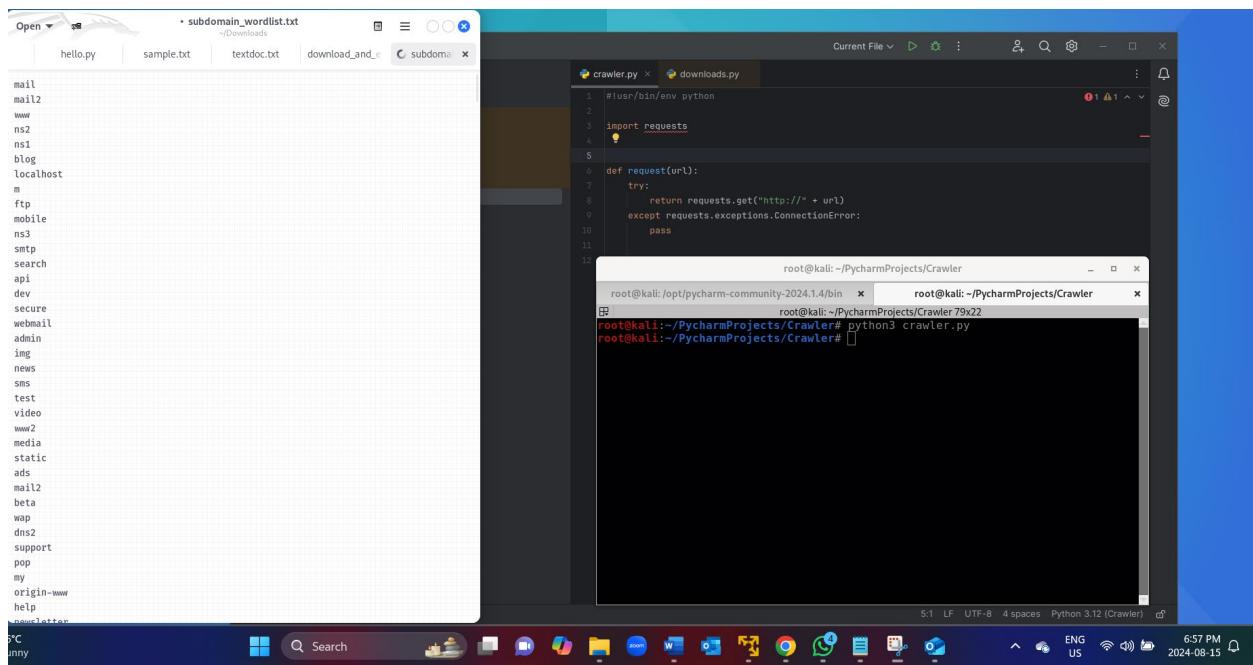
import requests

url = "mailnnnffsf.google.com"

try:
    get_response = requests.get("http://" + url)
    print(get_response)
except requests.exceptions.ConnectionError:
    pass
```



Discovering Website Subdomains Using Python



```
#!/usr/bin/env python
```

```
import requests
```

```
def request(url):
```

```
    try:
```

```
        return requests.get("http://" + url)
```

```
    except requests.exceptions.ConnectionError:
```

```
        pass
```

```
target_url = "google.com"
```

```
with open("/root/Downloads/subdomain_wordlist.txt", "r") as wordlist_file:
```

```
    for line in wordlist_file:
```

```
        word = line.strip()
```

```

test_url = word + "." + target_url

response = request(test_url)

if response:

    print("[+] Discovered subdomain --> " + test_url)

```

The screenshot shows a Kali Linux desktop environment with a PyCharm community edition window open. The terminal window on the left displays the output of the script, showing numerous discovered subdomains of 'google.com'. The code editor window on the right contains the Python script 'crawler.py'.

```

root@kali:~/PycharmProjects/Crawler# python3 crawler.py
[+] Discovered subdomain --> mail.google.com
[+] Discovered subdomain --> www.google.com
[+] Discovered subdomain --> bing.google.com
[+] Discovered subdomain --> img.google.com
[+] Discovered subdomain --> mobile.google.com
[+] Discovered subdomain --> search.google.com
[+] Discovered subdomain --> admin.google.com
[+] Discovered subdomain --> news.google.com
[+] Discovered subdomain --> video.google.com
[+] Discovered subdomain --> ads.google.com
[+] Discovered subdomain --> support.google.com
[+] Discovered subdomain --> help.google.com
[+] Discovered subdomain --> chat.google.com
[+] Discovered subdomain --> image.google.com
[+] Discovered subdomain --> tv.google.com
[+] Discovered subdomain --> sites.google.com
[+] Discovered subdomain --> music.google.com
[+] Discovered subdomain --> images.google.com
[+] Discovered subdomain --> pay.google.com
[+] Discovered subdomain --> games.google.com
[+] Discovered subdomain --> business.google.com
[+] Discovered subdomain --> travel.google.com
[+] Discovered subdomain --> photo.google.com
[+] Discovered subdomain --> tools.google.com
[+] Discovered subdomain --> apps.google.com
[+] Discovered subdomain --> local.google.com
[+] Discovered subdomain --> home.google.com
[+] Discovered subdomain --> files.google.com
[+] Discovered subdomain --> shopping.google.com
[+] Discovered subdomain --> labs.google.com

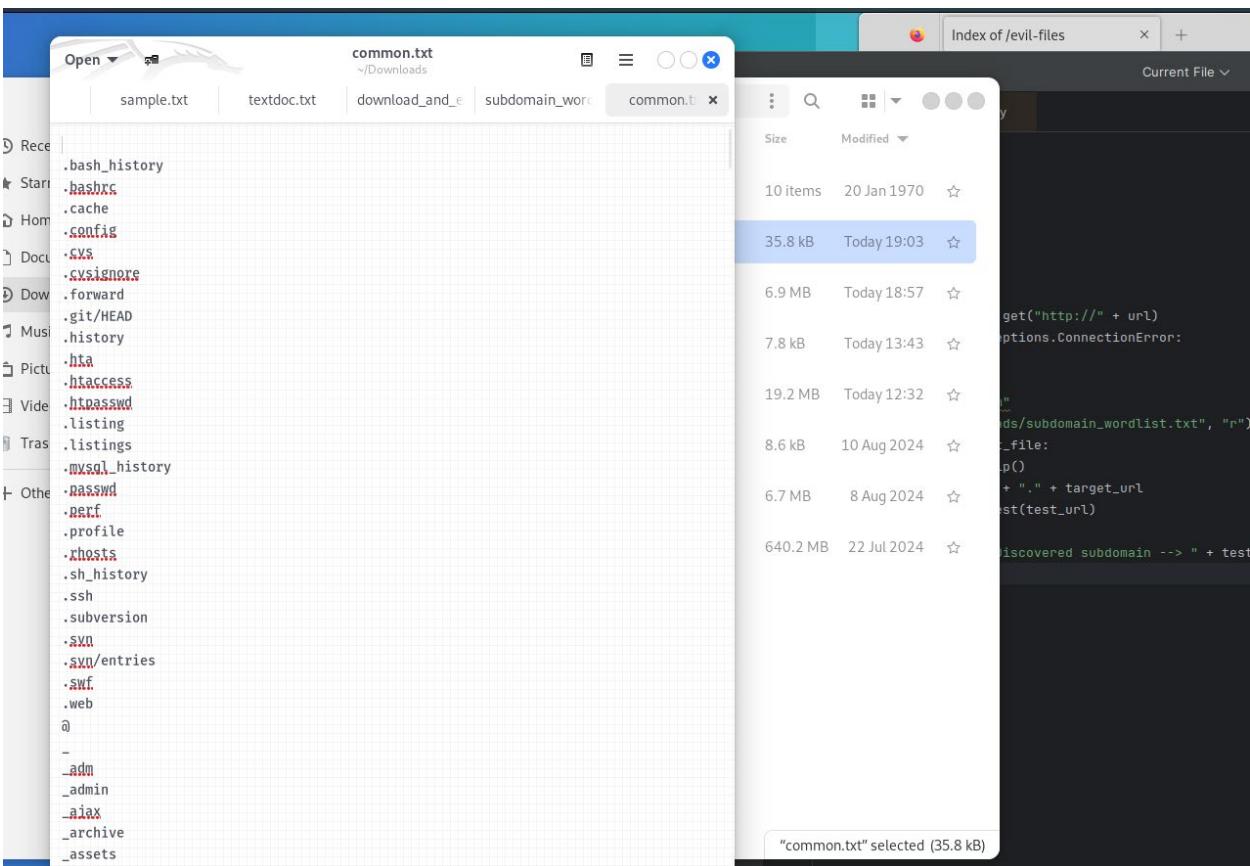
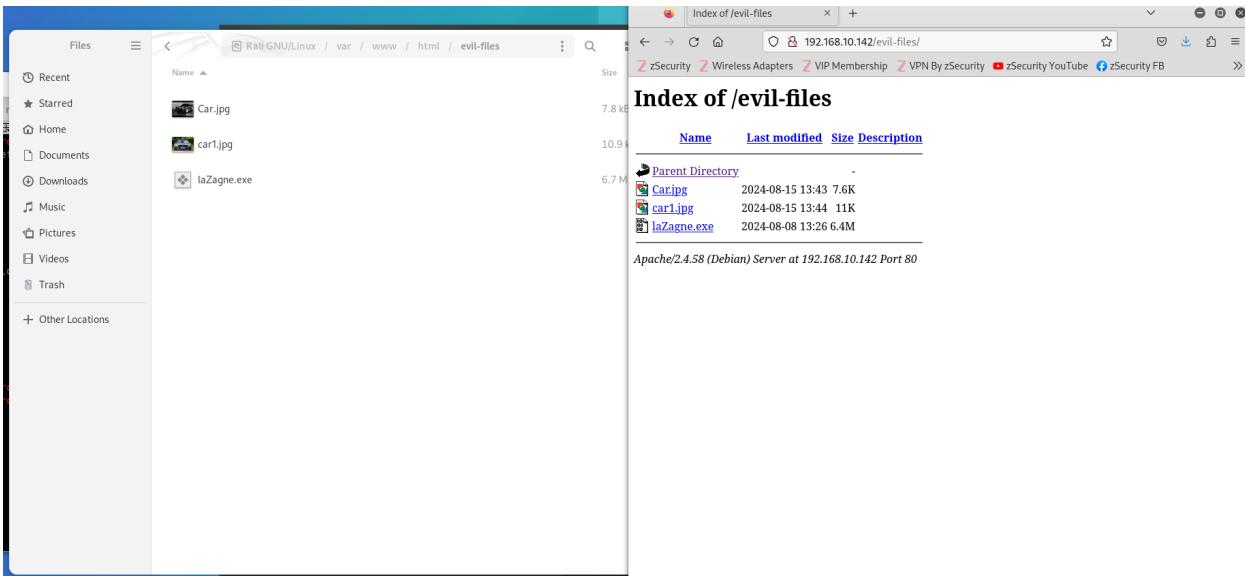
```

```

1 #!/usr/bin/env python
2
3 import requests
4
5
6 Usage
7 def request(url):
8     try:
9         return requests.get("http://" + url)
10    except requests.exceptions.ConnectionError:
11        pass
12
13 target_url = "google.com"
14 with open("/root/Downloads/subdomain_wordlist.txt", "r") as wordlist_file:
15     for line in wordlist_file:
16         word = line.strip()
17         test_url = word + "." + target_url
18         response = request(test_url)
19         if response:
20             print("[+] Discovered subdomain --> " + test_url)

```

Discovering Hidden Paths in Websites:



```
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
No mail.  
msfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet HWaddr 00:0c:29:a6:ee:39  
          inet addr:192.168.10.145 Bcast:192.168.10.255 Mask:255.255.255.0  
          inet6 addr: fe80::20c:29ff:fea6:ee39/64 Scope:Link  
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
             RX packets:39 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:66 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:1000  
             RX bytes:4309 (4.2 KB) TX bytes:6880 (6.7 KB)  
             Interrupt:17 Base address:0x2000  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1 Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
             UP LOOPBACK RUNNING MTU:16436 Metric:1  
             RX packets:91 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:91 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:0  
             RX bytes:19301 (18.8 KB) TX bytes:19301 (18.8 KB)  
  
msfadmin@metasploitable:~$
```

The screenshot shows a Kali Linux desktop environment. On the left, there's a dock with icons for terminal, file manager, browser, and other tools. The main window is a browser displaying the 'Mutilidae' web application, which appears to be a penetration testing tool. To the right of the browser is a code editor showing a Python script named 'crawler.py'. The script uses the 'requests' library to make HTTP requests and handle connection errors. It reads URLs from a file named 'common.txt' and appends words from another file to form test URLs. The desktop taskbar at the bottom shows various open applications and system status icons.

```
#usr/bin/env python  
import requests  
  
usage  
def request(url):  
    try:  
        return requests.get("http://" + url)  
    except requests.exceptions.ConnectionError:  
        pass  
  
target_url = "http://192.168.10.145/mutillidae/"  
with open("/root/Downloads/common.txt", "r") as wordlist_file:  
    for line in wordlist_file:  
        word = line.strip()  
        test_url = target_url + "/" + word  
        response = requests.get(test_url)  
        if response:  
            print("[+] Discovered URL -> " + test_url)
```

Crawler.py()

```
#!usr/bin/env python
```

```
import requests
```

```
def request(url):
```

```
    try:
```

```
        return requests.get("http://" + url)
```

```
    except requests.exceptions.ConnectionError:
```

```
        pass
```

```
target_url = "http://192.168.10.145/mutillidae/"
```

```
with open("/root/Downloads/common.txt", "r") as wordlist_file:
```

```
    for line in wordlist_file:
```

```
        word = line.strip()
```

```
        test_url = target_url + "/" + word
```

```
        response = request(test_url)
```

```
        if response:
```

```
            print("[+] Discovered URL --> " + test_url)
```

2024 x64 Customized by zSecurity v1.0.14 - VMware Workstation 17 Player (Non-commercial use only)

Applications Places Aug 15 19:45

root@kali:~/PycharmProjects/Crawler# python3 crawler.py

```

root@kali:~/PycharmProjects/Crawler# python3 crawler.py
[+] Discovered URL --> 192.168.10.145/mutillidae/
[+] Discovered URL --> 192.168.10.145/mutillidae/classes
[+] Discovered URL --> 192.168.10.145/mutillidae/credits
[+] Discovered URL --> 192.168.10.145/mutillidae/documentation
[+] Discovered URL --> 192.168.10.145/mutillidae/footer
[+] Discovered URL --> 192.168.10.145/mutillidae/header
[+] Discovered URL --> 192.168.10.145/mutillidae/home
[+] Discovered URL --> 192.168.10.145/mutillidae/images
[+] Discovered URL --> 192.168.10.145/mutillidae/inc
[+] Discovered URL --> 192.168.10.145/mutillidae/includes
[+] Discovered URL --> 192.168.10.145/mutillidae/index
[+] Discovered URL --> 192.168.10.145/mutillidae/installation
[+] Discovered URL --> 192.168.10.145/mutillidae/javascript
[+] Discovered URL --> 192.168.10.145/mutillidae/login
[+] Discovered URL --> 192.168.10.145/mutillidae/notes
[+] Discovered URL --> 192.168.10.145/mutillidae/password-not-found
[+] Discovered URL --> 192.168.10.145/mutillidae/passwords
[+] Discovered URL --> 192.168.10.145/mutillidae/phpinfo
[+] Discovered URL --> 192.168.10.145/mutillidae/phpinfo.php
[+] Discovered URL --> 192.168.10.145/mutillidae/phpMyAdmin
[+] Discovered URL --> 192.168.10.145/mutillidae/register
[+] Discovered URL --> 192.168.10.145/mutillidae/robots
[+] Discovered URL --> 192.168.10.145/mutillidae/robots.txt
[+] Discovered URL --> 192.168.10.145/mutillidae/styles
root@kali:~/PycharmProjects/Crawler#

```

controllers
CONTRASEÑAS
controls
converge_local
converge

25°C Sunny 19:56 LF UTF-8 4 spaces Python 3.12 (Crav) 7:45 PM 2024-08-15

Index of /mutillidae/passwords

Name	Last modified	Size	Description
Parent Directory	-		
accounts.txt	11-Apr-2011 20:14	176	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.10.145 Port 80

```

usage
def request(url):
    try:
        return requests.get("http://" + url)
    except requests.exceptions.ConnectionError:
        pass

target_url = "192.168.10.145/mutillidae/"

with open("/root/Downloads/common.txt", "r") as wordlist_file:
    for line in wordlist_file:
        word = line.strip()
        test_url = target_url + "/" + word
        response = request(test_url)
        if response:
            print("[+] Discovered URL --> " + test_url)

```

19:56 LF UTF-8 4 spaces

```
root@kali:~/PycharmProjects/Crawler79x32
root@kali:~/PycharmProjects/Crawler# python3 crawler.py
192.168.10.145/mutillidae/pa <--> 192.168.10.145/mutillidae/passwords/accounts.txt
zSecurity z Wireless Adapters z VIP Membership z VPN By zSecurity zSecurity YouTube zSecurity FB >>
'admin', 'adminpass', 'Monkey!!!'
'adrian', 'somepassword', 'Zombie Films Rock!!!
'john', 'monkey', 'I like the smell of confunk
'ed', 'pentest', 'Commandline KungFu anyone?'
```

```
import requests

usage
def request(url):
    try:
        return requests.get("http://" + url)
    except requests.exceptions.ConnectionError:
        pass

target_url = "192.168.10.145/mutillidae/"

with open("/root/Downloads/common.txt", "r") as wordlist_file:
    for line in wordlist_file:
        word = line.strip()
        test_url = target_url + "/" + word
        response = request(test_url)
        if response:
            print("[+] Discovered URL --> " + test_url)
```

Reading Response Content

```
#!/usr/bin/env python
```

import requests

```
def request(url):
```

try:

```
return requests.get("http://" + url)
```

```
except requests.exceptions.ConnectionError:
```

pass

```
target_url = "zsecurity.org"
```

```
response = request(target_url)
```

```
print(response.content)
```

The screenshot shows a Kali Linux desktop environment with several open windows:

- Terminal 1:** Shows the command `root@kali: ~/PyCharmProjects/Spider# cd ..` followed by a list of Python scripts: arp_spoof, hello, DNS_spoof, arpspoof_detector, keylogger, Hello, code_injector, mac_changer, Mal_download_execute_and_report, network_scanner, Packet_sniffer, downloads, replace_downloads, Spider, execute_command.py, reverse_backdoor.
- Terminal 2:** Shows the command `root@kali: ~/PyCharmProjects/Spider# python3 spider.py` followed by the output: <Response [200]>.
- PyCharm IDE:** A project named "Spider" is open. The "spider.py" file is the active editor, containing Python code for a web crawler. The code uses the requests library to get URLs from a target URL and handles connection errors. It also prints the response content. The "crawler.py" file is shown in the background.

Extracting Useful Data From Response

```
#!/usr/bin/env python

import requests
import re

def request(url):
    try:
        return requests.get("http://" + url)
    except requests.exceptions.ConnectionError:
        pass

target_url = "zsecurity.org"

response = request(target_url)

href_links = re.findall(r'(?>href=")(.*?)"', response.content.decode('utf-8'))

print(href_links)
```

The screenshot shows a Kali Linux terminal window and a PyCharm code editor side-by-side.

Terminal Window:

```
root@kali:~/PycharmProjects/Spider 79:39
root@kali:~/PycharmProjects/Spider
```

Code Editor (PyCharm):

```
#usr/bin/env python
# coding: utf-8
import requests
import re
usage
def request(url):
    try:
        return requests.get("http://" + url)
    except requests.exceptions.ConnectionError:
        pass
target_url = "zsecurity.org"
response = request(target_url)
href_links = re.findall(pattern=r'(?>href=")(.*?)"', response.content.decode('utf-8'))
print(href_links)
```

The code in the editor is a Python script named `crawler.py` that uses the `requests` library to send GET requests to a target URL (`zsecurity.org`) and extract href attributes from the HTML content using a regular expression pattern.

Filtering Results

Spider.py

```
#usr/bin/env python

import requests
import re
import urlparse

target_url = "https://zsecurity.org"

def extract_links_from(url):
    response = requests.get(target_url)
    return re.findall(r'(?>:href=")(.*?)"', response.content.decode('utf-8'))

href_links = extract_links_from(target_url)
for link in href_links:
    link = urlparse.urljoin(target_url, link)

    if target_url in link:
        print(link)
```

The screenshot shows a Kali Linux terminal window and a PyCharm IDE window side-by-side.

Kali Linux Terminal:

```
root@kali:~/PycharmProjects/Spider# curl -s https://zsecurity.org | grep <a href= | sed 's/.*<a href=\(.*\)>.*$/\1/g' | sort | uniq
```

PyCharm IDE:

Current File ▾ spider.py crawler.py

```
1 #!/usr/bin/env python
2
3 import requests
4 import re
5 import urlparse
6
7 target_url = "https://zsecurity.org"
8
9 usage
10 def extract_links_from(url):
11     response = requests.get(target_url)
12     return re.findall(pattern=r'(?:(?:href="#")|(.?))+(?:\s+|$)', response.content.decode('utf-8'))
13
14 href_links = extract_links_from(target_url)
15 for link in href_links:
16     link = urlparse.urljoin(target_url, link)
17
18     if target_url in link:
19         print(link)
```

Extracting Unique Links & Storing Them In a List:

```
#!/usr/bin/env python

import requests
import re
import urlparse

target_url = "https://zsecurity.org"
target_links = []

def extract_links_from(url):
    response = requests.get(target_url)
    return re.findall(r'(?<:href=")(.*?)"', response.content.decode('utf-8'))

href_links = extract_links_from(target_url)
for link in href_links:
    link = urlparse.urljoin(target_url, link)

    if "#" in link:
        link = link.split("#")[0]

    if target_url in link and link not in target_links:
        target_links.append(link)
        print(link)
```

Kali 2024 x64 Customized by zSecurity v1.0.14 - VMware Workstation 17 Player (Non-commercial use only)

Player Applications Places Aug 16 11:33

root@kali:~/PycharmProjects/Spider

root@kali:~/PycharmProjects/Spider 79x38

```
https://zsecurity.org/?_page=3
https://zsecurity.org/?_page=10
https://zsecurity.org/courses/learn-ethical-hacking-using-the-cloud/
https://zsecurity.org/user/ZaldySabih/
https://zsecurity.org/courses/learn-bug-bounty-hunting-web-security-testing-fro-m-scratch/
https://zsecurity.org/courses/the-ultimate-dark-web-anonymity-privacy-security-course/
https://zsecurity.org/courses/learn-ethical-hacking-from-scratch/
https://zsecurity.org/about-us/
https://zsecurity.org/download-custom-kali/
https://zsecurity.org/contact/
https://zsecurity.org/generability-disclosure/
https://zsecurity.org/faq/
https://zsecurity.org/forums/
https://zsecurity.org/privacy/
https://zsecurity.org/shipping/
https://zsecurity.org/refunds/
https://zsecurity.org/terms/
https://zsecurity.org/hacking-and-security/submit-an-article/
https://zsecurity.org/account/?action=lostpassword
https://zsecurity.org/wp-content/uploads/siteorigin-widgets/sow-features-default-eec270d3b90e.css
https://zsecurity.org/wp-content/plugins/so-widgets-bundle/widgets/features/css/style.css
https://zsecurity.org/wp-content/plugins/so-widgets-bundle/icons/icomon/style.css
https://zsecurity.org/wp-content/plugins/so-widgets-bundle/icons/fontawesome/style.css
https://zsecurity.org/wp-content/uploads/siteorigin-widgets/sow-headline-default-40cbb13e7355.css
https://zsecurity.org/wp-content/uploads/siteorigin-widgets/sow-image-default-787d6771435.css
https://zsecurity.org/wp-content/plugins/learnpress-woo-payment/assets/lp_woo_min.css
https://zsecurity.org/wp-content/uploads/siteorigin-widgets/sow-image-default-d6614b6747a.css
```

root@kali:~/PycharmProjects/Spider#

Current File spider.py crawler.py

```
#usr/bin/env python

import requests
import re
import urlparse

target_url = "https://zsecurity.org"
target_links = []

def extract_links_from(url):
    response = requests.get(target_url)
    return re.findall(pattern=r"(?:(?:http|https)://)?([\w\.-]+\.\w+)([\w\.-]+)", response.content.decode('utf-8'))

href_links = extract_links_from(target_url)
for link in href_links:
    link = urlparse.urljoin(target_url, link)

    if "#" in link:
        link = link.split("#")[0]

    if target_url in link and link not in target_links:
        target_links.append(link)
        print(link)

if __name__ == "__main__":
    extract_links_from(target_url)
```

25:1 LF UTF-8 4 spaces Python 3.12 (Spider) ⚡

Spider > spider.py

24°C Sunny

Search

Aug 16 11:33 AM 2024-08-16

Recursively Discovering All Paths On a Target Website

```
#!/usr/bin/env python

import requests
import re
import urlparse

target_url = "https://zsecurity.org"
target_links = []

def extract_links_from(url):
    response = requests.get(target_url)
    return re.findall(r'(?<:href=")(.*?)"', response.content.decode('utf-8'))

def crawl(url):
    href_links = extract_links_from(url)
    for link in href_links:
        link = urlparse.urljoin(url, link)

        if "#" in link:
            link = link.split("#")[0]

        if target_url in link and link not in target_links:
            target_links.append(link)
            print(link)
            crawl(link)

crawl(target_url)
```


Testing code with Python3:

Spider.py ()

```
#!usr/bin/env python

import requests
import re
import urllib.parse as urlparse

target_url = "http://192.168.10.145/mutillidae/"

target_links = []

def extract_links_from(url):
    response = requests.get(target_url)
    return re.findall(r'(?<:href=")(.*?)"', response.content.decode('utf-8'))

def crawl(url):
    href_links = extract_links_from(url)
    for link in href_links:
        link = urlparse.urljoin(url, link)

        if "#" in link:
            link = link.split("#")[0]

    if target_url in link and link not in target_links:
        target_links.append(link)
        print(link)
        crawl(link)
```

crawl(target_url)

Crawl.py ()

```
#!/usr/bin/env python
```

import requests

```
def request(url):
```

try:

```
return requests.get("http://" + url)
```

```
except requests.exceptions.ConnectionError:
```

pass

```
target_url = "192.168.10.145/mutillidae/"
```

```
with open("/root/Downloads/common.txt", "r") as wordlist_file:
```

```
for line in wordlist_file:
```

```

word = line.strip()

test_url = target_url + "/" + word

response = request(test_url)

if response:

    print("[+] Discovered URL --> " + test_url)

```

The screenshot shows a Kali Linux desktop environment. In the center, there is a terminal window titled 'root@kali: /PycharmProjects/Spider' displaying a list of discovered URLs. Below the terminal is a PyCharm project window titled 'Spider' containing a Python script named 'spider.py'. The script uses the requests library to perform a GET request on a target URL (192.168.10.145/mutillidae) and prints discovered URLs to the console. The desktop bar at the bottom shows various application icons and system status.

```

root@kali: /PycharmProjects/Spider
root@kali: /PycharmProjects/Spider 79:23
root@kali: /PycharmProjects/Spider$ python3 spider.py
[+] Discovered URL -> 192.168.10.145/mutillidae/
[+] Discovered URL -> 192.168.10.145/mutillidae//classes
[+] Discovered URL -> 192.168.10.145/mutillidae//credits
[+] Discovered URL -> 192.168.10.145/mutillidae//documentation
[+] Discovered URL -> 192.168.10.145/mutillidae//favicon.ico
[+] Discovered URL -> 192.168.10.145/mutillidae//footer
[+] Discovered URL -> 192.168.10.145/mutillidae//header
[+] Discovered URL -> 192.168.10.145/mutillidae//index
[+] Discovered URL -> 192.168.10.145/mutillidae//images
[+] Discovered URL -> 192.168.10.145/mutillidae//inc
[+] Discovered URL -> 192.168.10.145/mutillidae//includes
[+] Discovered URL -> 192.168.10.145/mutillidae//index
[+] Discovered URL -> 192.168.10.145/mutillidae//index.php
[+] Discovered URL -> 192.168.10.145/mutillidae//installation
[+] Discovered URL -> 192.168.10.145/mutillidae//javascript
[+] Discovered URL -> 192.168.10.145/mutillidae//login
[+] Discovered URL -> 192.168.10.145/mutillidae//logout
[+] Discovered URL -> 192.168.10.145/mutillidae//page-not-found
[+] Discovered URL -> 192.168.10.145/mutillidae//passwords
[+] Discovered URL -> 192.168.10.145/mutillidae//phpinfo
[+] Discovered URL -> 192.168.10.145/mutillidae//phpinfo.php
[+] Discovered URL -> 192.168.10.145/mutillidae//phpMyAdmin
[+] Discovered URL -> 192.168.10.145/mutillidae//register

```