Code Review Worksheet

CPSC 491/491L/492L

Review Identification Information

Date of Review: 17 Novemeber 2023

Project Review is Part of: Medcurity Network Inventory (10)

Branch/PR Identifier: Commit 6fb76c755621eec773e1410ed48aaa53bda2bbd8, PR #11 Automate Server ....

Code Reviewer(s): Brandon

Code Author or Source of Code: Colleen

These following categories should be addressed, though not all might be applicable, it will depend upon the project and code at hand.

#1 Project, Milestone, Issues, Features Included, and/or Pull Request description
Find IP Addresses with Crawler

#2 Code Structural, Design Considerations, and Style Guide Suggestions
Modularizing, using python-isms, all good

#3 Code Architectural and Efficiency Considerations
Architecture is good

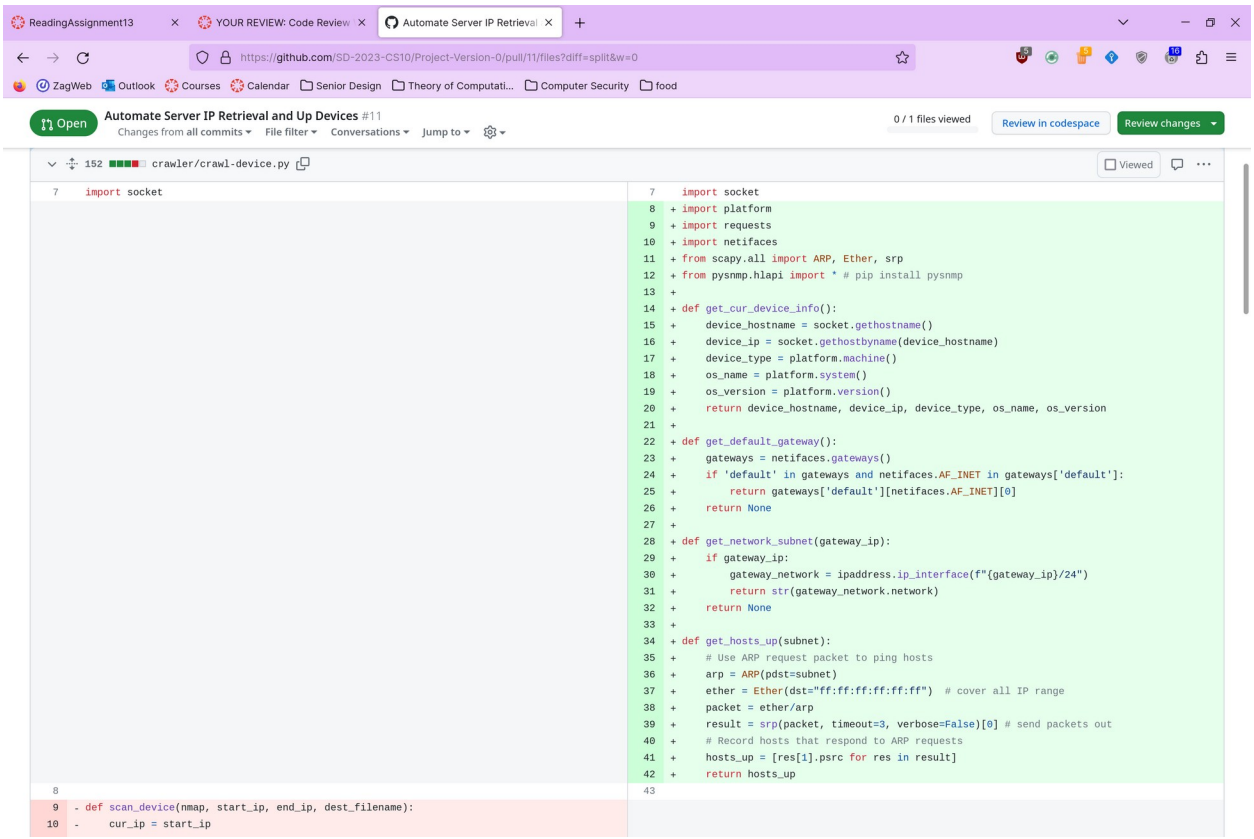
#4 Execution, Runtime, and Bugs
None

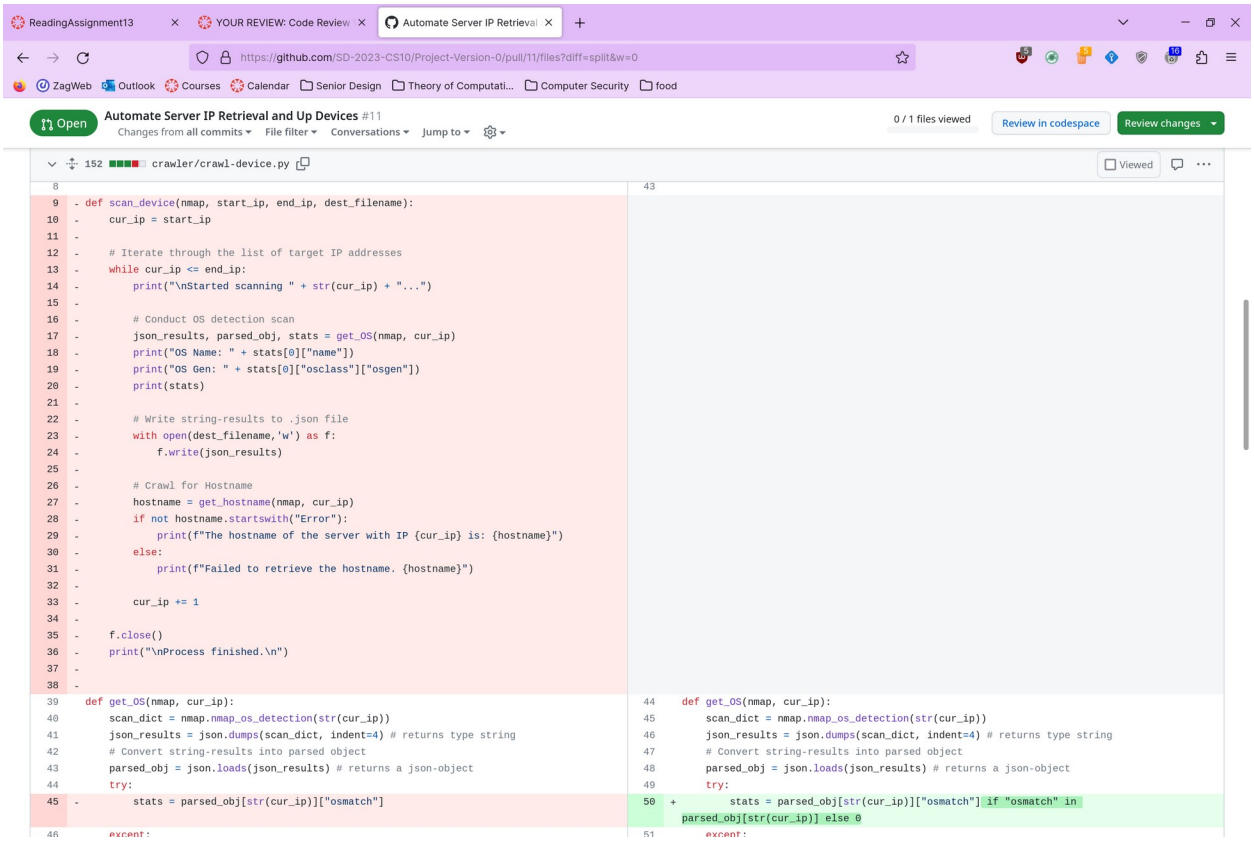
#5 Documentation Quality and Completeness
Minor comments describing section parts, nice.

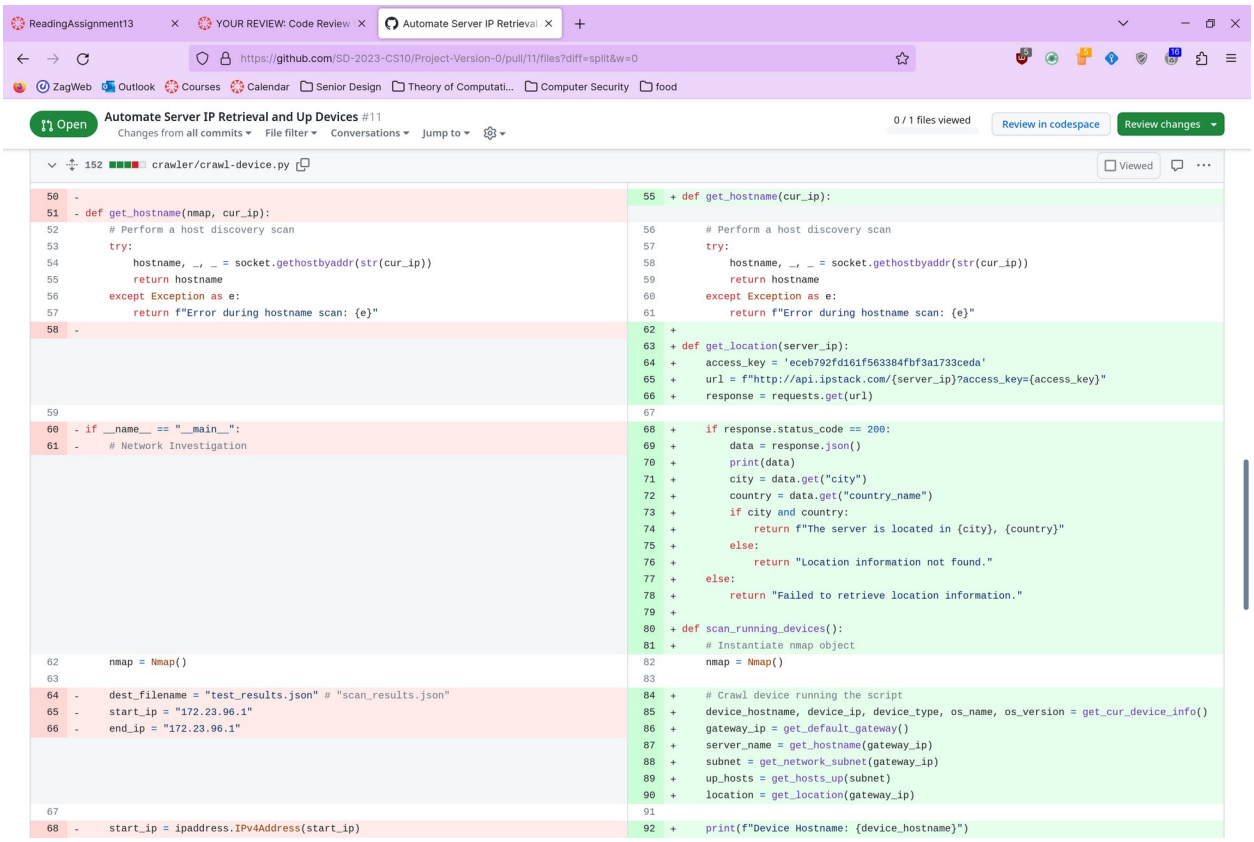
#6 Testing, Tests, Coverage, and suggested testing improvements
Testing in main to use while developing is good.

#7 General Comments, Notes or Other Suggestions for Author
Looks good! Glad you are making solid progress on the crawler!

#8 Appendix - Include up to five pages of the code under review as a diff.

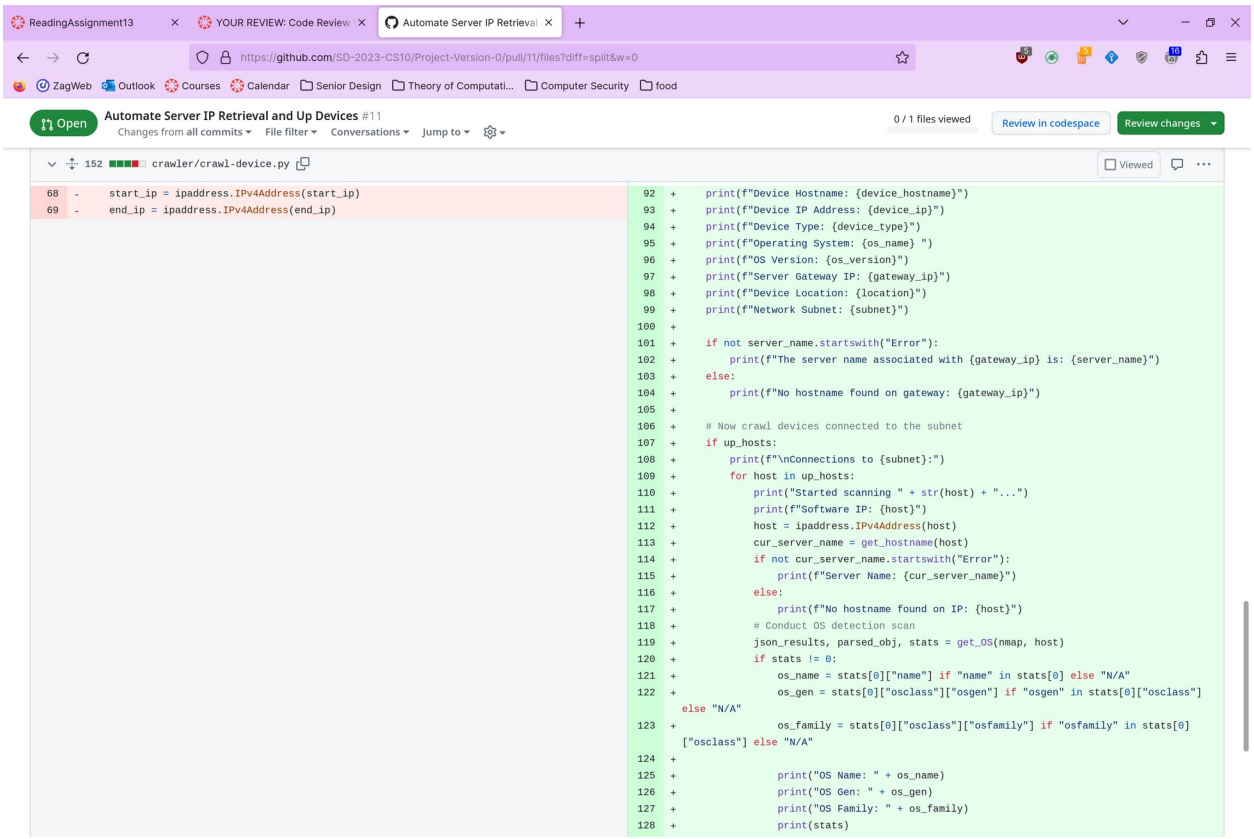

ReadingAssignment13
YOUR REVIEW: Code Review
Automate Server IP Retrieval
https://github.com/SD-2023-CS10/Project-Version-0/pull/11/files?diff=split&w=0
ZagWeb
Outlook
Courses
Calendar
Senior Design
Theory of Computati...
Computer Security
food
Open
Automate Server IP Retrieval and Up Devices #11
Changes from all commits
File filter
Conversations
Jump to
0 / 1 files viewed
Review in codespace
Review changes
152
crawler/crawl-device.py
Viewed
import socket
import platform
import requests
import netifaces
from scapy.all import ARP, Ether, srp
from pysnmp.hlapi import * # pip install pysnmp
def get_cur_device_info():
device_hostname = socket.gethostname()
device_ip = socket.gethostbyname(device_hostname)
device_type = platform.machine()
os_name = platform.system()
os_version = platform.version()
return device_hostname, device_ip, device_type, os_name, os_version
def get_default_gateway():
gateways = netifaces.gateways()
if 'default' in gateways and netifaces.AF_INET in gateways['default']:
return gateways['default'][netifaces.AF_INET][0]
return None
def get_network_subnet(gateway_ip):
if gateway_ip:
gateway_network = ipaddress.ip_interface(f"{gateway_ip}/24")
return str(gateway_network.network)
return None
def get_hosts_up(subnet):
# Use ARP request packet to ping hosts
arp = ARP(pdst=subnet)
ether = Ether(dst="ff:ff:ff:ff:ff:ff") # cover all IP range
packet = ether/arp
result = srp(packet, timeout=3, verbose=False)[0] # send packets out
# Record hosts that respond to ARP requests
hosts_up = [res[1].psrc for res in result]
return hosts_up
def scan_device(nmap, start_ip, end_ip, dest_filename):
cur_ip = start_ip

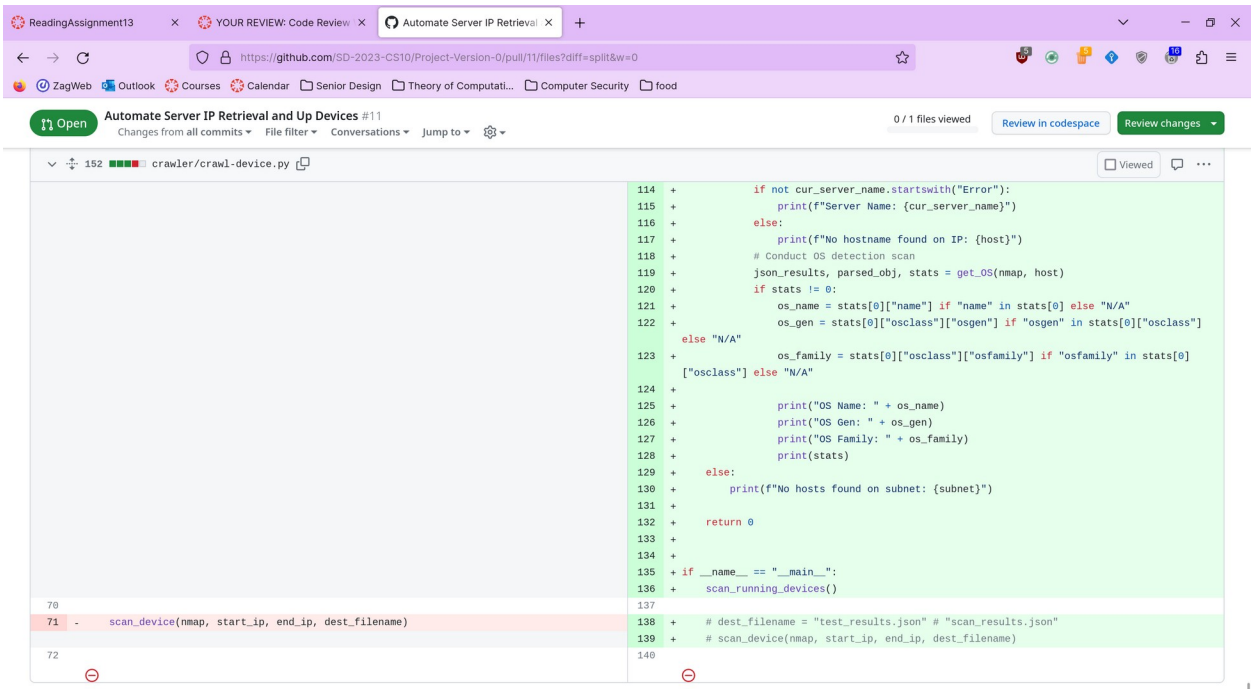

ReadingAssignment13
YOUR REVIEW: Code Review
Automate Server IP Retrieval
https://github.com/SD-2023-CS10/Project-Version-0/pull/11/files?diff=split&w=0
ZagWeb
Outlook
Courses
Calendar
Senior Design
Theory of Computati...
Computer Security
food
Open
Automate Server IP Retrieval and Up Devices #11
Changes from all commits
File filter
Conversations
Jump to
0 / 1 files viewed
Review in codespace
Review changes
152
crawler/crawl-device.py
Viewed
def scan_device(nmap, start_ip, end_ip, dest_filename):
cur_ip = start_ip
# Iterate through the list of target IP addresses
while cur_ip <= end_ip:
print("\nStarted scanning " + str(cur_ip) + "...")
# Conduct OS detection scan
json_results, parsed_obj, stats = get_OS(nmap, cur_ip)
print("OS Name: " + stats[0]["name"])
print("OS Gen: " + stats[0]["osclass"]["osgen"])
print(stats)
# Write string-results to .json file
with open(dest_filename,'w') as f:
f.write(json_results)
# Crawl for Hostname
hostname = get_hostname(nmap, cur_ip)
if not hostname.startswith("Error"):
print(f"The hostname of the server with IP {cur_ip} is: {hostname}")
else:
print(f"Failed to retrieve the hostname. {hostname}")
cur_ip += 1
f.close()
print("\nProcess finished.\n")
def get_OS(nmap, cur_ip):
scan_dict = nmap.nmap_os_detection(str(cur_ip))
json_results = json.dumps(scan_dict, indent=4) # returns type string
# Convert string-results into parsed object
parsed_obj = json.loads(json_results) # returns a json-object
try:
stats = parsed_obj[str(cur_ip)]["osmatch"]
except:
def get_OS(nmap, cur_ip):
scan_dict = nmap.nmap_os_detection(str(cur_ip))
json_results = json.dumps(scan_dict, indent=4) # returns type string
# Convert string-results into parsed object
parsed_obj = json.loads(json_results) # returns a json-object
try:
stats = parsed_obj[str(cur_ip)]["osmatch"] if "osmatch" in parsed_obj[str(cur_ip)] else 0
except:




```
50  -
51  - def get_hostname(nmap, cur_ip):
52        # Perform a host discovery scan
53        try:
54            hostname, _, _ = socket.gethostbyaddr(str(cur_ip))
55            return hostname
56        except Exception as e:
57            return f"Error during hostname scan: {e}"
58  -
59
60  - if __name__ == "__main__":
61  -     # Network Investigation
62        nmap = Nmap()
63
64  -     dest_filename = "test_results.json" # "scan_results.json"
65  -     start_ip = "172.23.96.1"
66  -     end_ip = "172.23.96.1"
67
68  -     start_ip = ipaddress.IPv4Address(start_ip)
```

```
55  + def get_hostname(cur_ip):
56        # Perform a host discovery scan
57        try:
58            hostname, _, _ = socket.gethostbyaddr(str(cur_ip))
59            return hostname
60        except Exception as e:
61            return f"Error during hostname scan: {e}"
62  +
63  + def get_location(server_ip):
64  +     access_key = 'eceb792fd161f563384fbf3a1733ceda'
65  +     url = f"http://api.ipstack.com/{server_ip}?access_key={access_key}"
66  +     response = requests.get(url)
67
68  +     if response.status_code == 200:
69  +         data = response.json()
70  +         print(data)
71  +         city = data.get("city")
72  +         country = data.get("country_name")
73  +         if city and country:
74  +             return f"The server is located in {city}, {country}"
75  +         else:
76  +             return "Location information not found."
77  +     else:
78  +         return "Failed to retrieve location information."
79  +
80  + def scan_running_devices():
81  +     # Instantiate nmap object
82        nmap = Nmap()
83
84  +     # Crawl device running the script
85  +     device_hostname, device_ip, device_type, os_name, os_version = get_cur_device_info()
86  +     gateway_ip = get_default_gateway()
87  +     server_name = get_hostname(gateway_ip)
88  +     subnet = get_network_subnet(gateway_ip)
89  +     up_hosts = get_hosts_up(subnet)
90  +     location = get_location(gateway_ip)
91
92  +     print(f"Device Hostname: {device_hostname}")
```


ReadingAssignment13
YOUR REVIEW: Code Review
Automate Server IP Retrieval
https://github.com/SD-2023-CS10/Project-Version-0/pull/11/files?diff=split&w=0
ZagWeb
Outlook
Courses
Calendar
Senior Design
Theory of Computati...
Computer Security
food
Open
Automate Server IP Retrieval and Up Devices #11
Changes from all commits
File filter
Conversations
Jump to
0 / 1 files viewed
Review in codespace
Review changes
152 crawler/crawl-device.py
Viewed
68 - start_ip = ipaddress.IPv4Address(start_ip)
69 - end_ip = ipaddress.IPv4Address(end_ip)
92 + print(f"Device Hostname: {device_hostname}")
93 + print(f"Device IP Address: {device_ip}")
94 + print(f"Device Type: {device_type}")
95 + print(f"Operating System: {os_name} ")
96 + print(f"OS Version: {os_version}")
97 + print(f"Server Gateway IP: {gateway_ip}")
98 + print(f"Device Location: {location}")
99 + print(f"Network Subnet: {subnet}")
100 +
101 + if not server_name.startswith("Error"):
102 + print(f"The server name associated with {gateway_ip} is: {server_name}")
103 + else:
104 + print(f"No hostname found on gateway: {gateway_ip}")
105 +
106 + # Now crawl devices connected to the subnet
107 + if up_hosts:
108 + print(f"\nConnections to {subnet}:")
109 + for host in up_hosts:
110 + print("Started scanning " + str(host) + "...")
111 + print(f"Software IP: {host}")
112 + host = ipaddress.IPv4Address(host)
113 + cur_server_name = get_hostname(host)
114 + if not cur_server_name.startswith("Error"):
115 + print(f"Server Name: {cur_server_name}")
116 + else:
117 + print(f"No hostname found on IP: {host}")
118 + # Conduct OS detection scan
119 + json_results, parsed_obj, stats = get_OS(nmap, host)
120 + if stats != 0:
121 + os_name = stats[0]["name"] if "name" in stats[0] else "N/A"
122 + os_gen = stats[0]["osclass"]["osgen"] if "osgen" in stats[0]["osclass"] else "N/A"
123 + os_family = stats[0]["osclass"]["osfamily"] if "osfamily" in stats[0]["osclass"] else "N/A"
124 +
125 + print("OS Name: " + os_name)
126 + print("OS Gen: " + os_gen)
127 + print("OS Family: " + os_family)
128 + print(stats)


ReadingAssignment13
YOUR REVIEW: Code Review
Automate Server IP Retrieval
https://github.com/SD-2023-CS10/Project-Version-0/pull/11/files?diff=split&w=0
ZagWeb
Outlook
Courses
Calendar
Senior Design
Theory of Computati...
Computer Security
food
Open
Automate Server IP Retrieval and Up Devices #11
Changes from all commits
File filter
Conversations
Jump to
0 / 1 files viewed
Review in codespace
Review changes
152
crawler/crawl-device.py
Viewed
114 + if not cur_server_name.startswith("Error"):
115 + print(f"Server Name: {cur_server_name}")
116 + else:
117 + print(f"No hostname found on IP: {host}")
118 + # Conduct OS detection scan
119 + json_results, parsed_obj, stats = get_OS(nmap, host)
120 + if stats != 0:
121 + os_name = stats[0]["name"] if "name" in stats[0] else "N/A"
122 + os_gen = stats[0]["osclass"]["osgen"] if "osgen" in stats[0]["osclass"] else "N/A"
123 + os_family = stats[0]["osclass"]["osfamily"] if "osfamily" in stats[0]["osclass"] else "N/A"
124 +
125 + print("OS Name: " + os_name)
126 + print("OS Gen: " + os_gen)
127 + print("OS Family: " + os_family)
128 + print(stats)
129 + else:
130 + print(f"No hosts found on subnet: {subnet}")
131 +
132 + return 0
133 +
134 +
135 + if __name__ == "__main__":
136 + scan_running_devices()
70
71 - scan_device(nmap, start_ip, end_ip, dest_filename)
72
137
138 + # dest_filename = "test_results.json" # "scan_results.json"
139 + # scan_device(nmap, start_ip, end_ip, dest_filename)
140
© 2023 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About