

OS Telemetry

Attack Documentation

TCP Flood Attack

A TCP (Transmission Control Protocol) flood attack, also known as a TCP flood DDoS (Distributed Denial of Service) attack, is a type of cyberattack that aims to overwhelm a target server or network by inundating it with a massive number of TCP connection requests. This attack can cause the target system to become unresponsive or slow down significantly, ultimately disrupting its normal operation and making it inaccessible to legitimate users.

In a TCP Flood attack, such as a SYN Flood attack, the goal is to overwhelm a target server with a large volume of TCP connection requests. Attackers may use various techniques to make it appear as though these connection requests are coming from different IP addresses. This is known as IP spoofing.

IP spoofing is the practice of manipulating the source IP address in network packets to make it seem like they are originating from a different source. In the context of a TCP Flood attack, IP spoofing can be used to send a massive number of SYN (synchronization) requests with different source IP addresses.

Here's how a TCP flood attack typically works:

1. **TCP Handshake:** When two devices (a client and a server) want to communicate over a TCP connection, they go through a three-way handshake process. The client sends a "SYN" (synchronize) packet to the server, the server responds with a "SYN-ACK" (synchronize-acknowledge) packet, and the client acknowledges with an "ACK" packet. Once this handshake is complete, the connection is established.
2. **TCP Flood Attack:** In a TCP flood attack, the attacker sends a large number of fake or spoofed "SYN" packets to the target server without ever intending to complete the three-way handshake. These "SYN" packets consume server resources as the server allocates memory to track each incomplete connection request.
3. **Resource Exhaustion:** As the server continues to receive an overwhelming number of incomplete "SYN" requests, it quickly exhausts its resources, such as available memory and processing capacity. The legitimate users' requests can't be processed because the server is busy dealing with the flood of fake requests.
4. **Impact:** The impact of a TCP flood attack can range from degrading the target's performance to causing a complete denial of service, where the target becomes entirely inaccessible to legitimate users.
5. **Defence:** To defend against TCP flood attacks, organizations often employ various mitigation techniques, such as rate limiting incoming connection requests, filtering out malicious traffic, and utilizing DDoS mitigation services.

Kill Chain Path:

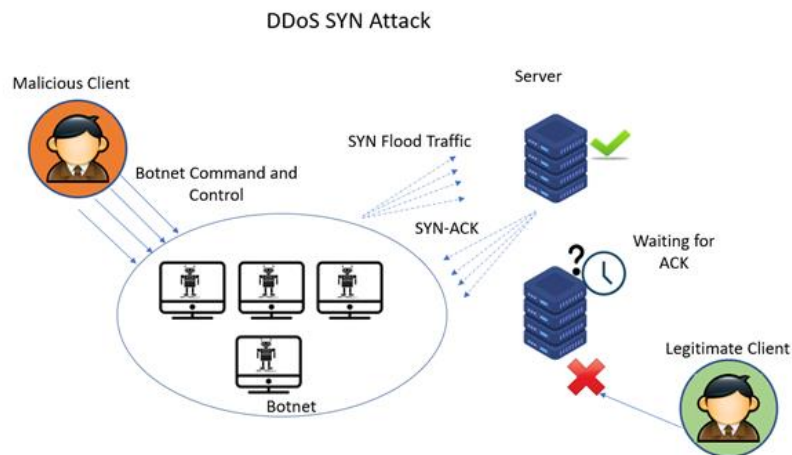


Figure 1 TCP Attack

Reconnaissance: Attackers often perform some level of reconnaissance to identify potential targets or vulnerabilities, although this is typically less relevant in DDoS attacks.

Weaponization: In DDoS attacks, the "weapon" is usually a large number of compromised or botnet-controlled devices capable of sending a high volume of traffic. The attackers don't need to create specific malicious code for this.

Delivery: There's no specific delivery phase in a DDoS attack. Instead, attackers directly launch the attack by sending a flood of traffic to the target's network or service.

Exploitation: DDoS attacks don't typically involve exploiting vulnerabilities in the same way as targeted attacks. Instead, they exploit the fundamental limitations of network infrastructure by overwhelming it with traffic.

Installation: There's no installation phase in a DDoS attack; the attackers don't need to establish persistence on the target's systems.

Command and Control (C2): In DDoS attacks, command and control infrastructure may exist, but it is more likely associated with controlling the botnet of compromised devices than with the attack itself.

Actions on Objectives: The primary objective in a DDoS attack is to disrupt the target's network or service, rather than traditional objectives like data theft or system compromise.

Exfiltration: DDoS attacks typically don't involve data exfiltration; their goal is to render a target's service inaccessible.

Plausible IOCs:

1. Unusual Traffic Patterns:

- A significant increase in network traffic, especially to a single destination or port.
- A high volume of connection attempts in a short time.
- Unexpected or irregular spikes in traffic.

2. Unusual Protocol Behaviour:

- Traffic to a specific port or protocol not typically used.
- Unexpected or unauthorized service or protocol use.

3. Malicious Payloads:

- Detection of known malware signatures in network traffic.
- Anomalous or malicious payload content in packets.

4. Source IP Anomalies:

- Traffic from a source IP address that's not part of network or is on a blacklist.
- IP addresses known to be associated with malicious activity or attackers.

5. Destination IP Anomalies:

- Traffic to a destination IP address that's not part of your network or is associated with malicious hosts or known command and control servers.

6. Port Scanning or Probing:

- Repeated connection attempts to various ports in a short period.
- Scanning behaviour indicative of an attacker looking for open ports or vulnerabilities.

7. Failed Authentication Attempts:

- A series of failed login attempts to a service (e.g., SSH, RDP) that may indicate a brute force attack.

8. Rate Limiting and Connection Limits:

- Reaching connection or rate limits set for specific services, which can indicate an attack.

9. Unusual Traffic Flow:

- Traffic coming from a geographically unusual source for your network.
- Traffic with unusual time-of-day patterns.

10. Known Attack Signatures:

- Signature-based detection of known attack patterns, such as DDoS, port scanning, or application-specific vulnerabilities.

11. DNS Query Anomalies:

- Unusual patterns in DNS queries, which may indicate DNS amplification or reflection attacks.

12. Packet Fragmentation:

- Unusual packet fragmentation patterns that can be indicative of an attack trying to evade detection.

13. Geographical Anomalies:

- Traffic originating from regions or countries not typically associated with legitimate traffic for the organization.

14. High Bandwidth Utilization:

- Abnormally high bandwidth utilization that can indicate a DDoS attack.

15. Unusual Traffic Volumes:

- Sudden or abnormal spikes in traffic volume that don't correspond with normal network activity.

IOCs in the OS Telemetry Logs:

Network Logs:

```
20059,Oct 18, 2023 05:21:06.963537076 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20060,Oct 18, 2023 05:21:06.963634608 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20061,Oct 18, 2023 05:21:06.964513992 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20062,Oct 18, 2023 05:21:06.964573907 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20063,Oct 18, 2023 05:21:06.967910975 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20064,Oct 18, 2023 05:21:06.967970596 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20065,Oct 18, 2023 05:21:06.968959305 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20066,Oct 18, 2023 05:21:06.969013895 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20067,Oct 18, 2023 05:21:06.969215561 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20068,Oct 18, 2023 05:21:06.969263533 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20069,Oct 18, 2023 05:21:06.971934284 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp:tls:tls
20070,Oct 18, 2023 05:21:06.971989000 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20071,Oct 18, 2023 05:21:06.972725148 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20072,Oct 18, 2023 05:21:06.972774485 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20073,Oct 18, 2023 05:21:06.974809871 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20074,Oct 18, 2023 05:21:06.974860484 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20075,Oct 18, 2023 05:21:06.975832361 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20076,Oct 18, 2023 05:21:06.975880524 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20077,Oct 18, 2023 05:21:06.976657129 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp
20078,Oct 18, 2023 05:21:06.976710257 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
20079,Oct 18, 2023 05:21:06.978339204 EDT,151.101.153.55,192.168.126.135,443,47572,,,2886,eth:ethertype:ip:tcp:tls:tls
20080,Oct 18, 2023 05:21:06.978361632 EDT,192.168.126.135,151.101.153.55,47572,443,,,54,eth:ethertype:ip:tcp
```

Figure 2 *Spooled Source IP Addresses* sending SYN requests to the network

EVIL RABBIT (Rootkit)

A **rootkit** is some software that may work independently or cooperatively with some other malicious code to conceal its presence as well as any malicious activities. It is usually intended to conceal the existence of files, directories, logins, processes, remote connections and any malicious activity intended by its payload.

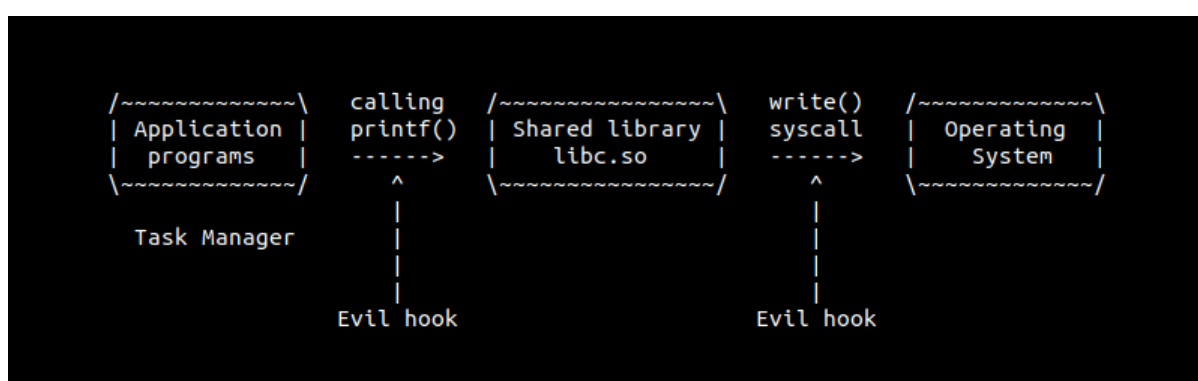


Figure 3 Rootkit interception of sys call services offered by the OS by a program

Imagine we have a simple program that uses the `printf()` function to display "Hello hell" on the console. In a Linux environment, `printf()` is a function provided by the standard C library, `libc.so`.

When this program runs, it calls `printf()`, which internally utilizes a system call called `write()` to send "Hello hell" to the console. System calls are how a program asks the operating system to perform tasks on its behalf.

Now, if a rootkit manages to intercept these system calls, it gains control over the program's behaviour. Rootkits are designed to be stealthy and conceal their malicious actions, giving them extensive control over a program's execution. Although they have the potential for malicious activities, their primary purpose is to hide themselves and their harmful actions to avoid detection.

To successfully intercept system calls, a rootkit needs to be inside the core part of the operating system, known as the kernel, because system calls are functions provided by the operating system.

Similarly, to intercept calls to shared libraries, the rootkit must be inside the memory space of the running program. Shared libraries are like tools that programs use, so being inside the program's memory allows the rootkit to tamper with those tools.

Lifecycle of Rootkits:

1. Delivery:

- **Infection Vector:** Attackers choose a delivery method, such as email attachments, malicious downloads, or exploiting vulnerabilities, to introduce the rootkit into the target system.

2. Installation:

- **Exploitation:** The rootkit is installed on the target system by exploiting vulnerabilities, using techniques like buffer overflows or social engineering to trick users into executing the rootkit.
- **Privilege Escalation:** Rootkits may employ privilege escalation techniques to gain administrative or root-level access, ensuring they have maximum control over the system.
- **Persistence:** The rootkit establishes persistence mechanisms, modifying system configurations, creating hidden files, or adding entries in the start-up process to ensure it survives reboots.

3. Concealment:

- **Stealth:** Rootkits actively hide their presence by intercepting system calls, altering logs, and disguising their files and processes.
- **Anti-Antivirus:** Rootkits employ techniques to evade antivirus and security software, including code obfuscation and polymorphic code.

4. Control and Communication:

- **Command and Control (C2):** The rootkit communicates with external servers controlled by the attacker to receive commands, exfiltrate data, or provide updates.
- **Data Exfiltration:** Rootkits may exfiltrate sensitive data, such as passwords or intellectual property, back to the attacker.

5. Exploitation:

- **Privilege Escalation:** Rootkits continue to exploit vulnerabilities as they are discovered to maintain control over the system and escalate their privileges.
- **Lateral Movement:** If necessary, attackers may use rootkits to move laterally within a network, infecting additional systems.

6. Actions on Objectives:

- **Data Theft or Manipulation:** Attackers may steal, alter, or destroy data based on their objectives. Rootkits provide the means to carry out these actions discreetly.
- **Sabotage or Espionage:** Rootkits can be used to disrupt system operations or engage in corporate espionage.

7. Evasion and Detection Avoidance:

- **Rootkit Updates:** Rootkit developers continually update their code to counter security measures and adapt to changes in the target system.
- **Anti-Forensic Techniques:** Rootkits use anti-forensic methods to remove traces of their presence.

IOCs:

```
(root@kali)-[/home/kali/Desktop/EVIL_RABBIT-master]
# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:19999           0.0.0.0:*               LISTEN
raw6       0      0 [::]:ipv6-icmp         [::]:*                  7
Active UNIX domain sockets (only servers)
```

Figure 4 `netstat -l` reveals that port 19999 is listening to all IP addresses trying to establish connection over it

```
(root@kali)-[/home/kali/Desktop/EVIL_RABBIT-master]
# ldd /usr/bin/python
linux-vdso.so.1 (0x00007ffefef7ec000)
/tmp/evil_rabbit.so (0x00007f57516dc000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f57515e7000)
libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1 (0x00007f57515c8000)
libexpat.so.1 => /lib/x86_64-linux-gnu/libexpat.so.1 (0x00007f575159d000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f57513bb000)
/lib64/ld-linux-x86-64.so.2 (0x00007f57516e4000)
```

Figure 5 All programs infected have the rootkit shared object in its dependencies for stealth

```
6145 Timestamp: 2023-10-19 06:54:38 | Event: CREATE, File: /home/kali/Desktop/EVIL_RABBIT-master/demo/innocent|
6146 Timestamp: 2023-10-19 06:54:38 | Event: CREATE, File: /tmp/evil_rabbit.so
6147 Timestamp: 2023-10-19 06:54:38 | Event: MODIFY, File: /tmp/evil_rabbit.so
6148 Timestamp: 2023-10-19 06:54:38 | Event: CREATE, File: /etc/ld.so.preload
6149 Timestamp: 2023-10-19 06:54:38 | Event: MODIFY, File: /etc/ld.so.preload
6150 Timestamp: 2023-10-19 06:54:38 | Event: CREATE, File: /tmp/.snow_valley
6151 Timestamp: 2023-10-19 06:54:38 | Event: MODIFY, File: /tmp/.snow_valley
```

Figure 6 File Monitoring Log screenshot

Above screenshot represents the file creation of Shared Objects(.so) files in /tmp so that the infection in preload works for all programs.

The *snow_valley* is a file created by the Rootkit whose existence is responsible for the creation of the TCP shell bind on port 19999 listening to all IPs.

```

7129 7128,Oct 19, 2023 06:55:39.966319722 EDT,,,,,,00,eth:ethertype:arp
7130 7129,Oct 19, 2023 06:55:39.966391020 EDT,,,,,,42,eth:ethertype:arp
7131 7130,Oct 19, 2023 06:55:39.967525756 EDT,192.168.126.130,192.168.126.134,60950,19999,,,74,eth:ethertype:ip:tcp
7132 7131,Oct 19, 2023 06:55:39.967576890 EDT,192.168.126.134,192.168.126.130,19999,60950,,,74,eth:ethertype:ip:tcp
7133 7132,Oct 19, 2023 06:55:39.968662106 EDT,192.168.126.130,192.168.126.134,60950,19999,,,66,eth:ethertype:ip:tcp
7134 7133,Oct 19, 2023 06:55:41.355520421 EDT,192.168.126.130,192.168.126.134,60950,19999,,,69,eth:ethertype:ip:tcp:tls
7135 7134,Oct 19, 2023 06:55:41.355565885 EDT,192.168.126.134,192.168.126.130,19999,60950,,,66,eth:ethertype:ip:tcp
7136 7135,Oct 19, 2023 06:55:41.362960369 EDT,192.168.126.134,192.168.126.130,19999,60950,,,230,eth:ethertype:ip:tcp:tls
7137 7136,Oct 19, 2023 06:55:41.363802827 EDT,192.168.126.130,192.168.126.134,60950,19999,,,66,eth:ethertype:ip:tcp
7138 7137,Oct 19, 2023 06:55:49.261972871 EDT,192.168.126.134,185.199.109.154,35376,443,,,93,eth:ethertype:ip:tcp:tls

```

Figure 7 Using the TCP Bind Shell for accessing the infected machine via another machine on the network

Limitations and Detection

- It does not get triggered by *statically compiled programs* (although there are just a few to be seen) as they do not contain dynamic sections and hence doesn't require runtime linking.

The technique requires creating a new process or restarting an existing one, which is a drawback because these actions can be easily detected, raising suspicion among system administrators and security software.

For example, let's say a malware uses this technique to inject itself into a legitimate process, like "notepad.exe." When the malware does this, it needs to either start a new "notepad.exe" process or forcibly restart an existing "notepad.exe" process. These actions, creating a new process or restarting an existing one, are noticeable events that can trigger security alarms, making the malware's presence more easily detectable.

- It is easy to detect, simply listing the SO dependencies via **/usr/bin/ldd** (a BASH script), one can see the malicious shared library getting along with every program.

Benchmark Testing:

The benchmark testing is written in order *to test the basic logging* capabilities of the OS Telemetry tool. It mainly focuses on creating activity in order to verify if such activities are being highlighted in the Telemetry logs.

It is done via 3 different bash files which need to be executed on the target after the OS Telemetry tool has started logging. All the tools have been set on an infinite loop to perform certain functions and get detected by the Telemetry Tool.

Another purpose of the writing these benchmarking tools was to be able to test how the Raspberry Pi performs under stress.

The ***Benchmark_Testing.sh*** consumes considerable amount of CPU resources causing more buffer lined up to be appended to the S3 bucket. However, the *activity of the Telemetry itself was not hindered* even under such conditions where all 3 Benchmarking tools were run simultaneously upon the already running Telemetry tool.

There are 3 Benchmark Tools at the moment targeting different Log tools, which are mentioned as follows:

Benchmark Tool Name	Target Log
Audit.sh	Audit logs & File Monitoring logs, Process.logs
netBen.sh	Networking Logs, File Monitoring Logs, Process.logs
Benchmark_Testing.sh	Process.Logs, sar.logs, iostat, iotop.logs

CODE:

Benchmark_Testing.sh:

```
#!/bin/bash
```

```
while true; do
```

```
    sudo sysbench --test=cpu run &>/dev/null
```

```
    sudo sysbench --test=memory run &>/dev/null
```

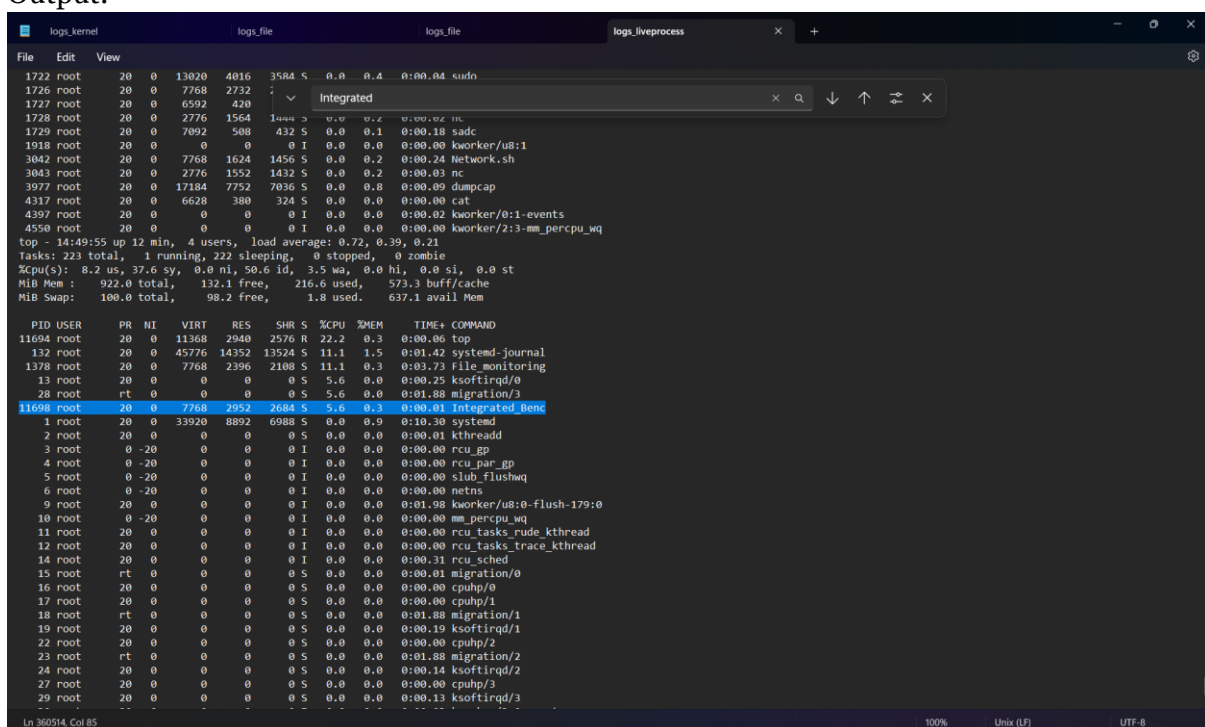
```
    sudo sysbench --test=fileio --file-test-mode=seqwr cleanup &>/dev/null
```

```
    echo "Press Ctrl+C to stop the loop..."
```

```
    sleep 1
```

```
done
```

Output:



```
logs_kernel logs_file logs_file logs_liveprocess
File Edit View
1722 root 20 0 13020 4016 3584 S 0.0 0.4 0:00.04 sudo
1726 root 20 0 7768 2732 1444 S 0.0 0.2 0:00.02 nc
1727 root 20 0 6592 420 0 S 0.0 0.1 0:00.18 sadc
1728 root 20 0 2776 1564 432 S 0.0 0.1 0:00.00 kworker/0:1
1729 root 20 0 7092 508 0 I 0.0 0.0 0:00.00 kworker/0:1
1918 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/0:1
3042 root 20 0 7768 1624 1456 S 0.0 0.2 0:00.02 Network.sh
3043 root 20 0 2776 1552 1432 S 0.0 0.2 0:00.03 nc
3977 root 20 0 17184 7752 7036 S 0.0 0.8 0:00.09 dumpcap
4317 root 20 0 6628 380 324 S 0.0 0.0 0:00.00 cat
4397 root 20 0 0 0 0 I 0.0 0.0 0:00.02 kworker/0:1-events
4550 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/2:3-mm_percpu_wq
top - 14:49:55 up 12 min, 4 users, load average: 0.72, 0.39, 0.21
tasks: 223 total, 1 running, 222 sleeping, 0 stopped, 0 zombie
Cpu(s): 8.2 us, 37.6 sy, 0.0 ni, 50.6 id, 3.5 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 922.0 total, 132.1 free, 216.6 used, 573.3 buff/cache
MiB Swap: 100.0 total, 98.2 free, 1.8 used, 637.1 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
11694 root 20 0 11368 2940 2576 R 22.2 0.3 0:00.06 top
132 root 20 0 45776 14352 13524 S 11.1 1.5 0:01.42 systemd-journal
1378 root 20 0 7768 2396 2108 S 11.1 0.3 0:03.73 file-monitoring
13 root 20 0 0 0 0 S 5.6 0.0 0:00.25 ksoftirqd/0
28 root rt 0 0 0 0 0 S 5.6 0.0 0:01.88 migration/3
11698 root 20 0 7768 2952 2684 S 5.6 0.3 0:00.01 Integrated_Benc
1 root 20 0 33920 8892 6988 S 0.0 0.9 0:10.30 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_pan_gp
5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 slub_flushwq
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
9 root 20 0 0 0 0 I 0.0 0.0 0:01.98 kworker/0:0-flush-179:0
10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
11 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthread
12 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kthread
14 root 20 0 0 0 0 I 0.0 0.0 0:00.01 rcu_sched
15 root rt 0 0 0 0 S 0.0 0.0 0:00.01 migration/0
16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
18 root rt 0 0 0 0 S 0.0 0.0 0:01.88 migration/1
19 root 20 0 0 0 0 S 0.0 0.0 0:00.19 ksoftirqd/1
22 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/2
23 root rt 0 0 0 0 S 0.0 0.0 0:01.88 migration/2
24 root 20 0 0 0 0 S 0.0 0.0 0:00.14 ksoftirqd/2
27 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/3
29 root 20 0 0 0 0 S 0.0 0.0 0:00.13 ksoftirqd/3
Ln 360514, Col 85 100% Unix (LF) UTF-8
```

netBen.sh:

```
# Main loop
while true; do
    clear # Clear the terminal screen for a clean display

    # Display the RPI's IP address
    get_ip_address

    # Ping a specific IP address
    ping_ip

    # Check if a specific port on a remote host is reachable
    check_port

    # Download a file from a given URL and delete it
    download_and_delete

    # Sleep for a few seconds before the next iteration (adjust as needed)
    sleep 10
done
```

Figure 8 Functions of the network testing script

Outputs:

```
Timestamp: 2023-09-18 14:51:07 | User: N/A | Group: N/A | Event: ATTRIB, File: /home/jodd/Desktop/Testing/sample3.txt
Timestamp: 2023-09-18 14:51:07 | User: jodd | Group: tty | Event: MODIFY, File: /dev/pts/2
Timestamp: 2023-09-18 14:51:07 | User: N/A | Group: N/A | Event: DELETE, File: /home/jodd/Desktop/Testing/sample3.txt
```

Figure 9 File Monitoring capturing the downloaded file which is deleted for next iteration

```
196,Oct  5, 2023 14:48:55.560274576 IST,192.168.1.195,192.168.1.100,45254,6996,,,85,eth:ethertype:ip:tcp:data,Blacklisted
197,Oct  5, 2023 14:48:55.582909546 IST,192.168.1.100,192.168.1.235,22,35859,,,98,eth:ethertype:ip:tcp:ssh,Valid IP
198,Oct  5, 2023 14:48:55.610221219 IST,192.168.1.100,192.168.1.195,6996,45254,,,66,eth:ethertype:ip:tcp,Valid IP
199,Oct  5, 2023 14:48:55.676506377 IST,192.168.1.235,192.168.1.100,35859,22,,,54,eth:ethertype:ip:tcp,Valid IP
200,Oct  5, 2023 14:48:56.114308920 IST,192.168.1.100,192.168.1.235,22,35859,,,98,eth:ethertype:ip:tcp:ssh,Valid IP
201,Oct  5, 2023 14:48:56.318096063 IST,192.168.1.235,192.168.1.100,35859,22,,,54,eth:ethertype:ip:tcp,Valid IP
202,Oct  5, 2023 14:48:56.633667973 IST,192.168.1.100,192.168.1.235,22,35859,,,98,eth:ethertype:ip:tcp:ssh,Valid IP
203,Oct  5, 2023 14:48:56.740160304 IST,192.168.1.235,192.168.1.100,35859,22,,,54,eth:ethertype:ip:tcp,Valid IP
204,Oct  5, 2023 14:48:57.152478196 IST,192.168.1.100,192.168.1.235,22,35859,,,98,eth:ethertype:ip:tcp:ssh,Valid IP
205,Oct  5, 2023 14:48:57.222757788 IST,192.168.1.195,192.168.1.100,45254,6996,,,69,eth:ethertype:ip:tcp:data,Blacklisted
206,Oct  5, 2023 14:48:57.222805966 IST,192.168.1.100,192.168.1.195,6996,45254,,,66,eth:ethertype:ip:tcp,Valid IP
207,Oct  5, 2023 14:48:57.226520233 IST,192.168.1.100,192.168.1.195,6996,45254,,,152,eth:ethertype:ip:tcp:data,Valid IP
208,Oct  5, 2023 14:48:57.230933110 IST,192.168.1.195,192.168.1.100,45254,6996,,,66,eth:ethertype:ip:tcp,Blacklisted
209,Oct  5, 2023 14:48:57.300327435 IST,192.168.1.218,192.168.1.255,,,137,137,92,eth:ethertype:ip:udp:nbns,Blacklisted
210,Oct  5, 2023 14:48:57.332413458 IST,192.168.1.235,192.168.1.100,35859,22,,,54,eth:ethertype:ip:tcp,Valid IP
```

Figure 10 Check for Valid IP flags in the logs. It signifies function to ping .235

```
7135 jodd      20      0      7768      2908      2632 S      0.0      0.3      0:00.22 netben.sh
```

Figure 11 Liveprocess Logs screenshot

Audit.sh:

```
# Run scenarios
echo "Running Scenarios"

# Create a new user, change their password, grant superuser privileges
while true;
do

    create_new_user

# Create a text file in the home directory with modified permissions
    create_text_file

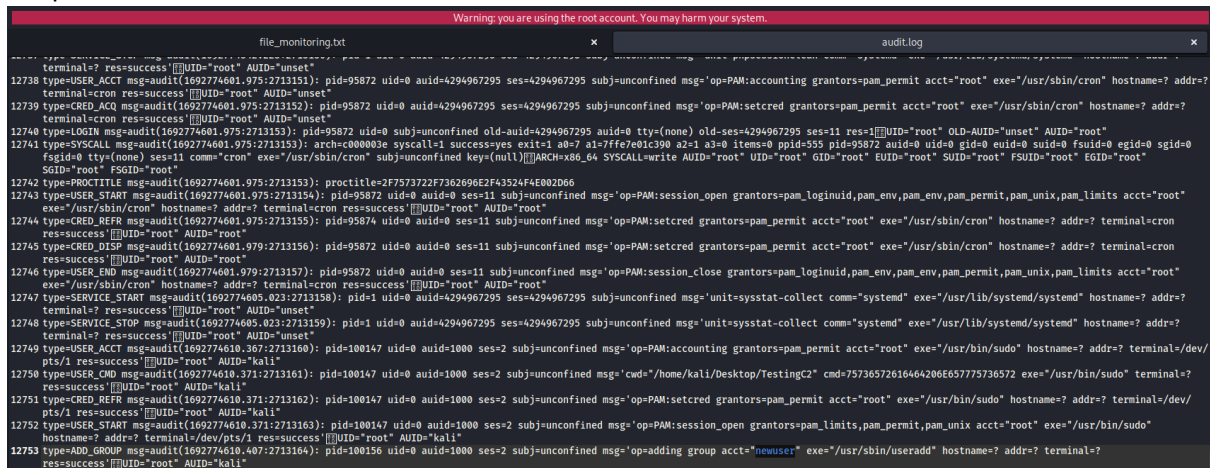
# Simulate unauthorized file access
    simulate_unauthorized_file_access

# Delete the text file after all steps are executed
    delete_text_file

# Delete the user after all steps are executed
    delete_user
```

Figure 12 Functions of the Audit.sh file

Outputs:



```
Warning: you are using the root account. You may harm your system.

file_monitoring.txt x audit.log x

12738 type=USER_ACCT msg=audit(1692774601.975:2713151): pid=95872 uid=0 audit=4294967295 ses=4294967295 subj=unconfined msg='op=PAM:accounting grantors=pam_permit acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=? res=success' [UID="root" AUDID="unset"]
12739 type=CRED_ACG msg=audit(1692774601.975:2713152): pid=95872 uid=0 audit=4294967295 ses=4294967295 subj=unconfined msg='op=PAM:setcred grantors=pam_permit acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=? res=success' [UID="root" AUDID="unset"]
12740 type=LOGIN msg=audit(1692774601.975:2713153): pid=95872 uid=0 subj=unconfined old-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=11 res=1 [UID="root" OLD-AUID="unset" AUDID="root"]
12741 type=SYSCALL msg=audit(1692774601.975:2713153): arch=c000003e syscall=1 success=yes exit=1 a0=7 a1=7ffe7e01c390 a2=1 a3=0 items=0 ppid=555 pid=95872 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=11 comm="cron" exe="/usr/sbin/cron" subj=unconfined key=(null) [ARCH=x86_64 SYSCALL=write AUDID="root" UID="root" GID="root" EUID="root" SUID="root" FSUID="root" EGID="root" Sgid="root" Fsgid="root"]
12742 type=PROCTITLE msg=audit(1692774601.975:2713153): proctitle=2F7573722F7362696E2FA3524FAE002D66
12743 type=USER_START msg=audit(1692774601.975:2713154): pid=95872 uid=0 audit=0 ses=11 subj=unconfined msg='op=PAM:session_open grantors=pam_loginuid,pam_env,pam_env,pam_permit,pam_unix,pam_limits acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cron res=success' [UID="root" AUDID="root"]
12744 type=CRED_REFR msg=audit(1692774601.975:2713155): pid=95874 uid=0 audit=0 ses=11 subj=unconfined msg='op=PAM:setcred grantors=pam_permit acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cron res=success' [UID="root" AUDID="root"]
12745 type=CRED_DISP msg=audit(1692774601.979:2713156): pid=95872 uid=0 audit=0 ses=11 subj=unconfined msg='op=PAM:setcred grantors=pam_permit acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cron res=success' [UID="root" AUDID="root"]
12746 type=USER_END msg=audit(1692774601.979:2713157): pid=95872 uid=0 audit=0 ses=11 subj=unconfined msg='op=PAM:session_close grantors=pam_loginuid,pam_env,pam_env,pam_permit,pam_unix,pam_limits acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cron res=success' [UID="root" AUDID="root"]
12747 type=SERVICE_START msg=audit(1692774605.023:2713158): pid=1 uid=0 audit=4294967295 ses=4294967295 subj=unconfined msg='unit=sysstat-collect comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success' [UID="root" AUDID="unset"]
12748 type=SERVICE_STOP msg=audit(1692774605.023:2713159): pid=1 uid=0 audit=4294967295 ses=4294967295 subj=unconfined msg='unit=sysstat-collect comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success' [UID="root" AUDID="unset"]
12749 type=USER_ACCT msg=audit(1692774610.367:2713160): pid=100147 uid=0 audit=1000 ses=2 subj=unconfined msg='op=PAM:accounting grantors=pam_permit acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/1 res=success' [UID="root" AUDID="kali"]
12750 type=USER_CMD msg=audit(1692774610.371:2713161): pid=100147 uid=0 audit=1000 ses=2 subj=unconfined msg='cmd="/home/kali/Desktop/TestingC2" cmd=7573657261646420655775736572 exe="/usr/bin/sudo" terminal=? res=success' [UID="root" AUDID="kali"]
12751 type=CRED_REFR msg=audit(1692774610.371:2713162): pid=100147 uid=0 audit=1000 ses=2 subj=unconfined msg='op=PAM:setcred grantors=pam_permit acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/1 res=success' [UID="root" AUDID="kali"]
12752 type=USER_START msg=audit(1692774610.371:2713163): pid=100147 uid=0 audit=1000 ses=2 subj=unconfined msg='op=PAM:session_open grantors=pam_limits,pam_permit,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/1 res=success' [UID="root" AUDID="kali"]
12753 type=ADD_GROUP msg=audit(1692774610.407:2713164): pid=100156 uid=0 audit=1000 ses=2 subj=unconfined msg='op=adding group acct="newuser" exe="/usr/sbin/useradd" hostname=? addr=? terminal=? res=success' [UID="root" AUDID="kali"]
```

Figure 13 Audit Logs captured the changes made by the audit.sh

```
Timestamp: 2023-09-18 14:50:19 | User: N/A | Group: N/A | Event: CREATE,ISDIR, File: /home/testUser
Timestamp: 2023-09-18 14:50:19 | User: N/A | Group: N/A | Event: ATTRIB,ISDIR, File: /home/testUser
Timestamp: 2023-09-18 14:50:19 | User: N/A | Group: N/A | Event: ATTRIB,ISDIR, File: /home/testUser/

Timestamp: 2023-09-18 14:50:31 | User: root | Group: adm | Event: MODIFY, File: /var/log/audit/audit.log
Timestamp: 2023-09-18 14:50:31 | User: N/A | Group: N/A | Event: ATTRIB, File: /home/testUser/test.txt
Timestamp: 2023-09-18 14:50:33 | User: root | Group: adm | Event: MODIFY, File: /var/log/audit.log
Timestamp: 2023-09-18 14:50:33 | User: N/A | Group: N/A | Event: CREATE, File: /home/testUser/UnauthorizedFile.txt
Timestamp: 2023-09-18 14:50:33 | User: N/A | Group: N/A | Event: ATTRIB, File: /home/testUser/UnauthorizedFile.txt
```

Logs captured in the File Monitoring Logs

Some other Test scenarios executed:

The tests done:

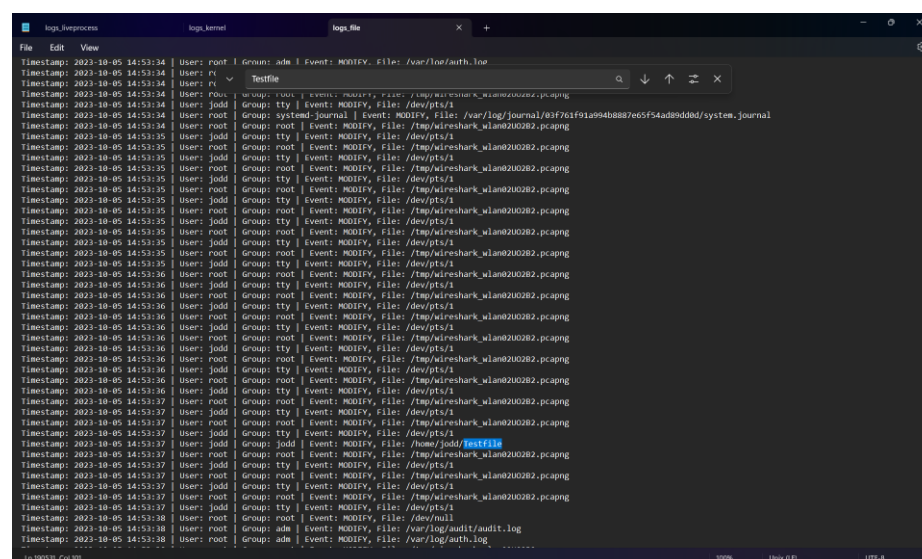
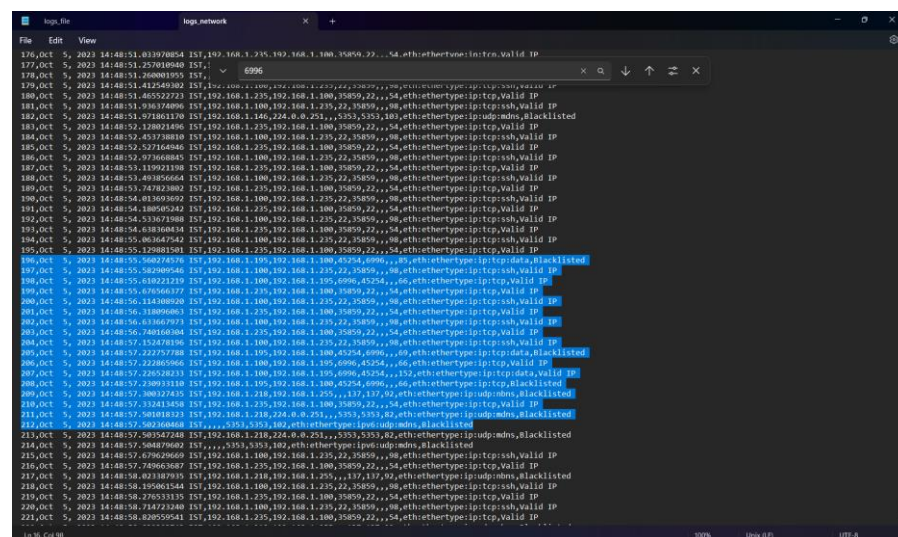
- Send a file to the Rpi from a different Linux Machine
- Execute Commands and run any program or script on the Rpi via different Linux Machine.

The monitoring tool was able to capture the IP Address of the Linux Machine and show the activity between these 2 devices.

File Monitoring Logs also detected a new file creation and modification

Process Info logged the script which was run remotely.

Attaching screenshots of the same.



```
logp_kernel  logp_file  logp_file  logp_livprocess
File Edit View
1722 root 20 0 13020 4016 3584 S 0.0 0.4 0:00.04 sudo
1726 root 20 0 7768 2732 420 S 0.0 0.0 0:00.00
1727 root 20 0 6592 420 S 0.0 0.0 0:00.00
1728 root 20 0 2776 1564 1564 S 0.0 0.0 0:00.00 nc
1729 root 20 0 7092 508 432 S 0.0 0.1 0:00.18 sadc
1918 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/u8:1
3042 root 20 0 7768 1624 1456 S 0.0 0.2 0:00.24 Network.sh
3043 root 20 0 2776 1552 1432 S 0.0 0.2 0:00.00 nc
3977 root 20 0 17184 7752 7036 S 0.0 0.8 0:00.00 dumpcap
4317 root 20 0 6628 380 324 S 0.0 0.0 0:00.00 cat
4397 root 20 0 0 0 0 I 0.0 0.0 0:00.02 kworker/0:1-events
4550 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/2:3-mm_percpu_wq
top - 14:49:55 up 12 min, 4 users, load average: 0.72, 0.39, 0.21
Tasks: 223 total, 1 running, 222 sleeping, 0 stopped, 0 zombie
%Cpu(s): 8.2 us, 37.6 sy, 0.0 ni, 50.6 id, 3.5 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 922.0 total, 132.1 free, 216.6 used, 573.3 buff/cache
MiB Swap: 100.0 total, 98.2 free, 1.8 used, 637.1 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
11694 root 20 0 11368 2940 2576 R 22.2 0.3 0:00.06 top
132 root 20 0 45776 14352 13524 S 11.1 1.5 0:01.42 systemd-journal
1378 root 20 0 7768 2396 2180 S 11.1 0.3 0:01.73 file_monitoring
13 root 20 0 0 0 0 S 5.6 0.0 0:00.25 ksoftirqd/0
28 root rt 0 0 0 0 0 S 5.6 0.0 0:01.88 migration/3
11698 root 20 0 7768 2952 2684 S 5.6 0.3 0:00.01 Integrated_Benc
1 root 20 0 33920 8892 6988 S 0.0 0.9 0:10.30 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 slub_flushwq
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
9 root 20 0 0 0 0 I 0.0 0.0 0:01.00 kworker/u8:0-flush-179:0
10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
11 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthread
12 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kthread
14 root 20 0 0 0 0 I 0.0 0.0 0:00.31 rcu_sched
15 root rt 0 0 0 0 S 0.0 0.0 0:00.01 migration/0
16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
18 root rt 0 0 0 0 S 0.0 0.0 0:01.88 migration/1
19 root 20 0 0 0 0 S 0.0 0.0 0:00.19 ksoftirqd/1
22 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/2
23 root rt 0 0 0 0 S 0.0 0.0 0:01.88 migration/2
24 root 20 0 0 0 0 S 0.0 0.0 0:00.14 ksoftirqd/2
27 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/3
29 root 20 0 0 0 0 S 0.0 0.0 0:00.13 ksoftirqd/3
.. ..
Ln 360314, Col 85 100% Unix (LF) UTF-8
```