Excel@FIT 2023
http://excel.fit.vutbr.cz

NetLoiter: A Tool for Automated Testing of Network Applications using Fault-Injection

Michal Rozsíval

supervisor: Aleš Smrčka

Abstract
Safety and security are crucial for applications communicating over the network. However, the development of these applications is performed in an ideal environment, unlike their usage in a real-world environment, which contains faults, like packet loss, delay, or corruption caused by poor transmission lines or cyber-attacks are present. This paper introduces the NetLoiter, which aims to automate the simulation of these faults, thus allowing the developers to handle them correctly. The Netloiter is deployed in hardware (MITM), visible software (proxy), and hidden software (impacts packets directly in the system kernel using NFTables, TC, or WFP) solutions. The main advantage of the NetLoiter is automating its workflow using its dynamic reconfiguration in combination with its RestAPI; this, for example, allows us to find the communicating parties' reliability limits automatically.

michal.rozsival@vut.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

The technologies communicating over the network are developed mainly in a single computer or small business network environment. This development environment is ideal; it has almost no delay, a reliable connection, and packets are mostly not lost or corrupted. However, the real-world (production) environment has non-negligible delays caused by the characteristics of the transmission link and processing nodes; packets are more often discarded [1], which may lead to disconnection or may not be established at all; and data corruption occurs due to both SW (bit-noise) or HW (malfunction of the communication node) errors and cyber attacks (man-in-the-middle attack). These situations occur in real-world communications, and not treating them can lead to safety and security risks. For example:

- Remotely controlled vehicle – acceleration and subsequent loss of communication.
- Client-Server communication with an unpredictably behaving client (e.g., poor internet signal).
- Cyber attacks (e.g., the MITM, impacting of selected packets or data modification).

Several tools allow the simulation of these faults by emulating the network, such as the NetEm that inspired this tool. NetEm can be applied to network interfaces on Linux systems and allows the definition of various conditions, such as packet dropping, packet delay, or packet modification. Its shortcomings include subjective complexity of use, the impossibility of dynamic reconfiguration, and the absence of contextual and external information when processing communications. [2][3]

I have developed the NetLoiter, a tool that is simple to use but has extensive functionality, inspired mainly by real-world network scenarios. It provides configurable ways to modify network parameters to make it faulty. The main advantages of this tool are its easy extensibility and automated usage via its RestAPI, which allows us to find the limits of the reliable behavior of the communicating parties based on a defined scenario, see fig. 1.

NetLoiter has been successfully integrated and used in CAMEA[1] (camera systems) and Roboauto[2] (teleoperation) companies.

2. Deployment

NetLoiter is designed in three variants, one hardware, and two software. The hardware solution is an exter-

[1] https://www.camea.cz/cz/
[2] https://roboauto.tech/

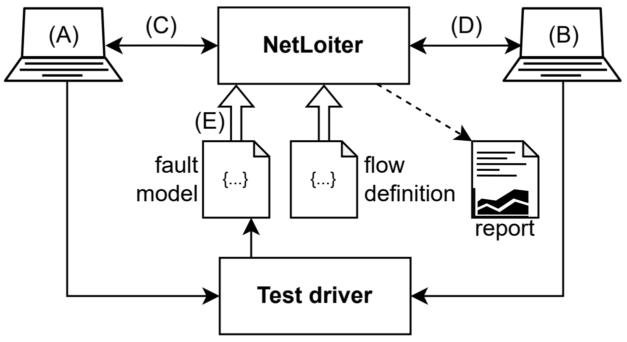


Figure 1. Overall NetLoiter architecture.

nal device connected between the communicating stations, representing a bridge/switch (OSI/ISO Layer 2) or an IP network router (OSI/ISO Layer 3). On the other hand, the software solutions are run directly on one of the communicating stations, either in a hidden form, in which it intercepts their communication without the knowledge of the communicating parties, or in a visible form, in which TCP or UDP streams must be redirected to the NetLoiter. NetLoiter intercepts communication between two or more nodes. All the traffic specified by the flow definition is sent to the NetLoiter, where the traffic processing depends on the rules defined by the fault model. NetLoiter can report all the events and actions during run-time for a more thorough analysis.

The choice of solution depends on the specific requirements:

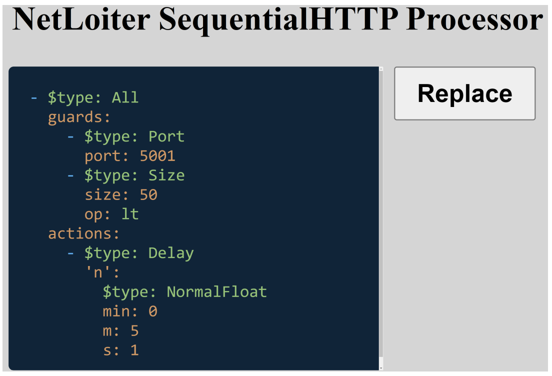
- Visible software solution – uses TCP and UDP sockets proxy. Requires changes in communicating applications; they must redirect their communication to NetLoiter.
- Hidden software solution – uses NFTables or Traffic Control on the Linux system and Windows Filtering Platform on the Windows system. Silently captures communication directly in the system kernel. This solution requires elevated user permissions to kernel access. Requires changes in communicating applications (redirection).
- Hardware solution – uses man-in-the-middle device placed in the communication, e.g., Raspberry Pi.

3. Usage

The NetLoiter targets test automation. After initialization, it can be dynamically reconfigured using its RestAPI, see fig 2.

Communication is affected by rules, which are compound of the:

- **Guards** – additional packet filtering: EveryN, ICMP, IP, Port, Prob, Protocol, Time, Count, TimePeriod, CountPeriod, Size, ...
- **Actions** – executed on selected packets: Delay, Drop, Reorder, Replicate, Restart, Finish, Throttle, BitNoise, SocketTCP, ...
- **ValueGenerators** – provides non-constant values to guards/actions: Uniform, Normal, Sequence, ...



NetLoiter SequentialHTTP Processor

```
- $type: All
  guards:
    - $type: Port
      port: 5001
    - $type: Size
      size: 50
      op: lt
  actions:
    - $type: Delay
      'n':
        $type: NormalFloat
        min: 0
        m: 5
        s: 1
```

Figure 2. The test driver can control the NetLoiter via its RestAPI.

4. Conclusion

The paper introduced the NetLoiter tool for testing the robustness of network applications and how they perform in different network conditions. The tool implements fault-injection on a network layer, simulating a number of faults and attacks which unreliable or even insecure networks may cause. NetLoiter is modular, dynamically reconfigurable and easy to use.

Acknowledgements

I would like to thank my supervisor Ing. Ales Smrčka, Ph.D., for his professional guidance, helpfulness, and valuable advice.

References

[1] Carlos Alexandre Gouvea Da Silva and Carlos Marcelo Pedroso. Mac-layer packet loss models for wi-fi networks: A survey. *IEEE Access*, 7:180512–180531, 2019.

[2] Linux Manual Page. NetEm - Network Emulator, 2011.

[3] Hemminger Stephen. Network emulation with netem. In *Proceedings of the 6th Australia's National Linux Conference*, pages 1–8, 2005.