Journal of Engineering and Applied Sciences 14 (Special Issue 6): 9294-9299, 2019
ISSN: 1816-949X
© Medwell Journals, 2019

Linux Firewall Implementation on Raspberry Pi 3

Lubos Mercl and Josef Horalek
Faculty of Informatics and Management, University of Hradec Kralove,
Hradec Kralove, Czech Republic

Abstract: With the advent of technologies associated with the internet of things solutions and devices it became necessary to solve the security of these infrastructures that could be easily defeated and are often involved in standard network infrastructure which also includes other critical data. This study deals with the firewall creation on Linux distributions on the Raspberry Pi 3 device. Effective internal network protecting consists creating some firewall implementation for allowing or denying some configured connections or separation of a part of the network to the separated network (demilitarized zone). Firewall implementation may also be used for protecting of individual IoT device based on Linux implementation. In this study, there are described solutions for creating and configuring firewall based on Linux distribution Raspbian Jessie Lite which is primarily intended for Raspberry Pi 3 equipment.

Key words: Firewall, Linux, Raspberry Pi, internet of things, Raspbian, critical

INTRODUCTION

Internet of Things (further IoT) is a solution where individual devices are connected to the internet and communicate with other devices or systems and sending data for analysis and usage by expert system (Gubbia *et al*., 2013).

IoT end devices are usually single-purpose devices connected wireless network (exceptionally wired) that through this interface to transmit information according to its intended use (Babar *et al*., 2010; Gubbia *et al*., 2013; Horaek *et al*., 2015).

Because the public networks are dangerous and an IoT idea is relatively new, so, it means a security of this devices are not main subject of solution and devices can be relatively easily challenged or data can be intercepted and vulnerable to attack (Babar *et al*., 2010; Horaek *et al*., 2015).

For secured solution IoT architecture includes a several parts in Wireless Sensor Networks (WSN) which should ensure safety (Gubbia *et al*., 2013):

- Hardware which cares about data collection or ensures its purpose
- Communication stack which is used for network communications between devices
- Middleware which collects data from sensors and prepares for analysis
- Secure data aggregation which provides data aggregations into larger units which are more suitable for analysis.

This study mainly deals with the security of the first two elements-hardware and communication stack. Sensor of IoT devices should be protected due to the fact that the correct data are automatically sent and this security is critical and whole environment is needed to protect it from attackers (Gubbia *et al*., 2013).

Security in IoT solutions is also important due to the fact that IoT devices are vulnerable as well as conventional devices on the network which include full-featured operating system and security. Some devices or solutions may be vulnerable even more.

IoT devices connected to a common internal network can have a great impact on the whole network in case of hacking because small networks are not secured as large corporate networks.

The goal of this study is to describe and propose solutions for network security for Linux distributions and also the possibility of creating a firewall solution built on the device Raspberry Pi 3.

Firewall: Firewalls are used to safeguard internal network from attack from outside and also for protecting sensitive data in internal network. They can block unauthorized attackers, unauthorized network communications or unwanted software (Rash, 2007). Each firewall has implemented the Access Control List (ACL) which includes a list of rules for access to network and evaluates incoming and outgoing packets (Rash, 2007). This set of rules contains several chain that make up individual rules. Packet is processed from the first to the last rule and if it

Corresponding Author: Lubos Mercl, Faculty of Informatics and Management, University of Hradec Kralove, Hradec Kralove, Czech Republic

is found that a rule equivalent to this definition, so, the appropriate action (Purdy, 2004). Among the types of firewalls are (Rash, 2007):

- Packet filter which evaluates traffic based on addresses or ports
- Stateful filter which maintains the state realized through connections
- Application layer filter which evaluates traffic based on known protocols (e.g., HTTP, SMTP or SMB)

However, most firewalls implement a combination of these filters.

Firewall implementation in Linux: In Linux is implemented IPfilter (also called netfilter) contains three functions to one rule and they are set via. IPtables (Purdy, 2004; Rash, 2007):

- Filtering, for packet filtering based on defined rules
- Forwarding, for packet forwarding
- Network Address Translation (NAT) for address translation

Configurations and rules are placed to a few independent tables which are (Purdy, 2004):

- Filter which is used for basic filtering, forwarding and logging
- NAT which is used for network address translation,
- Mangle which is used for packet changing
- Raw which is used for packet tracking

Which specific tables in iptables are present depends on the kernel configuration and setup of the core modules. Table filter contains three chains of rules which are (Purdy, 2004):

- Input for incoming packets to the system
- Forward for forwarding packets through the system
- Output for outgoing packets from the system

Security of Linux distributions: For Linux distributions there can be some security issues which can be included from developers because most of Linux distributions are developed by communities and non-profit reasons. From this reason each commercial implementation of Linux based system should be based on proven version from proven developer and should be tested. The most important thing is to have regular updates of the system.

Raspberry Pi 3: Raspberry Pi (further RPi) is a series of credit card-sized single-board computer developed by Raspberry Pi Foundation in 2012 and can be used for many purposes in school, home and industry area and IoT (Gubbia *et al*., 2013). RPi 3 is the third generation of RPi and has these attributes:

- ARM Cortex-A53-Quad-Core ARM processor with 1, 2 GHz frequency
- 1 GB RAM module and micro SD card slot
- WiFi interface 802.11n and Ethernet port (100 Mbits)
- Bluetooth 4.1a Bluetooth Low Energy (BLE)
- Four USB 2.0 ports, full HDMI port and more

For RPi can be used ARM Linux distributions (e.g., Raspbian which is built on Debian, Ubuntu Mate or Fedora) or Windows 10 IoT Core from Microsoft (Upton and Halfacree, 2014).

Raspbian Jessie Lite: Raspbian is official operating system from company Raspberry Pi Foundation. This operating system exist in two versions-Raspbian Jessie and Raspbian Jessie Lite. Both distributions are built on Debian Jessie. While Raspbian Jessie contains full graphical interface, Raspbian Jessie Lite contains only minimal image, this means there is only text interface. This study deals with Raspbian Jessie Lite Version which was released in March 2016 and contains Kernel Version 4.1.

MATERIALS AND METHODS

Problem definition: As mentioned in introduction for IoT is important to ensure the safety infrastructure and sensors to data that are transmitted from the sensor has been consistent and not interception. Consistency should be dealt with mainly at data and applications and interception on the media level.

Another important safety feature that helps protect the internal network and sensors is firewall which helps us protect the internal network against attacks from outside the network computer network or the internet.

To effectively the internal network protecting, it is important to build and configure firewall rules to allow only communication that is appropriate which ideally means that our internal network only allow the communication that is enabled exactly. The same or similar principle applies to outbound traffic from the internal network out.

In this study, there is proposed and fundamentally described solutions for firewall that secures the network from threats to internal network and the internet and a firewall that creates a Demilitarized Zone (DMZ) within a single network. Creating a demilitarized zone can have a positive impact for a communication separating in this separated zone from conventional communication network.

Solution design: As mentioned in the previous section there will be created two configurations-one for firewall between internal network and the internet and the second for DMZ zone creation but both configuration are similar.

Firewall between internal network and the internet: Figure 1 shows the infrastructure with firewall which is placed between internal network and the internet. This is simple network and connection which has been developed for basic tests of this implementation.

Connection to the internet is created on one internal LAN interface on the RPi 3 firewall device. At this interface (eth0) communication is protected using firewall rules and internal network is protected from outside attacks (from internet or provider network). This zone is dangerous because it may not be properly protected and there are more subjects connected.

On the right side of the firewall is the internal (home) network which should be more protected or at least network traffic on this network should be after the administration of the network administrator. At this interface (eth1) should be largely protected outgoing traffic from the network and should be released only packets that are sent from right reasons from a network so as to ensure a secure network connection and the network will not receive sensitive data.

RPi 3 has only one network LAN interface therefore for connection to an internal network is used external USB LAN interface. Internal network is realized by L2 network switch in which all devices are connected via. a network cable (e.g., computers, servers and others). There is also WiFi access point connected which create wiFi network for other devices (e.g., IoT devices, laptops, mobile phones and others).

Firewall between internal network and DMZ network: Another situation that may occur in the IoT design infrastructure is when it is necessary to separate a part of the infrastructure into a separate secure network.

As necessary there can be needed to split the network into two subnets-internal and DMZ network where will be applied some more restricted rules and security demands and only some network traffic will be allowed.

Figure 2 is shows a firewall which has two network interfaces-eth0 which creates internal network and ethl which creates DMZ network. Alternatively, it would be appropriate to create a third network interface which will create connection to the internet.

Figure 2 also shows devices which are named like end device. These devices can be everything-IoT device, computer, server, WiFi access point and others. Device

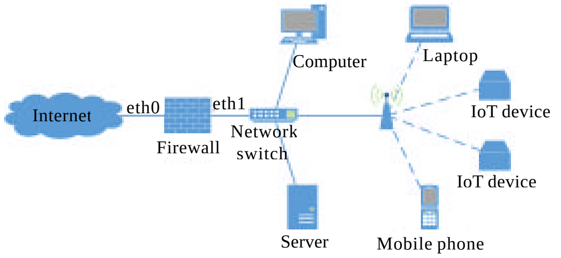


Fig. 1: Firewall between internet and internal network

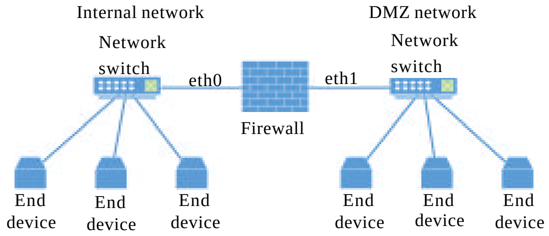


Fig. 2: Firewall between internal and DMZ network

location is given by implementation and requirements mainly on the security of the technology. In general, it can be said that but most should be protected company data and users, therefore, the DMZ networks could be placed servers and protected as less secure components in the network (end user devices or IoT equipment). Another possible implementation would be placed IoT devices to DMZ network.

RESULTS AND DISCUSSION

Implementation: Solution implementation consists of several steps. First was the need to download an image of the Linux Raspbian (specifically Raspbian Jessie Lite) and a tool for creating bootable SD card and extract this image on this card.

The next step was creating and principally discussing solutions for the firewall implementation deployed between the internet and the internal network and the firewall for the creation of a demilitarized zone.

Raspbian Jessie Lite installation and Raspberry Pi 3 preparation: Before the implementation was needed to prepare RPi 3 and above to install the Linux distribution Raspbian Jessie Lite (further Raspbian). Raspbian image (online:https://www.raspberrypi.org/downloads/raspbian/) has been extracted to SD card using the freely available Win32 Disk Imager Tool (online: https://sourceforge.net/projects/win32diskimager/). After image deploying system RPi 3 can be turn on and operating system booted and basic operating system configuration has been implemented and tested which consisted of routine procedures and configuration settings for the device name, time settings, the changes of the default user passwords, settings, network interfaces, DNS service and other common settings. Furthermore, it was necessary to install and enable iptables using this command:

```
apt-get install iptables
```

After installing, we can now proceed to the configuration of firewall rules.

Firewall rules configuration: Each firewall must have defined rules that determine which communication can pass through the firewall and which one cannot. To configure rules should apply the following algorithm:

- Configuration of individual acceptance rules
- Disable all other communications

However, it is important to consider each statement and rule in order to avoid a situation where access to device will be cut off. First, we will need to allow communication to be through the firewall. For example, to enable HTTPS (port 443) is used the following command:

```
iptables-A FORWARD-i eth1-o
eth0-p tcp-dport 443-j ACCEPT
```

In this command, switch -A defines type of rule, therefore forward, switch -I defines input interface, switch -o defines output interface, switch -p defines protocol type, switch -dport defines destination port and switch -j defines action which will be done with a packet, therefore packet will be accepted. According to this syntax, we could then allow other protocols as needed, e.g., HTTP (port 80), DNS (53), FTP (20 and 21) and others.

Configuring return traffic from the internet: Now, it is necessary to create another rule for incoming traffic from the internet which was initialized from internal network. This can be done with this command:

```
iptables-A FORWARD-i eth0-o eth1-m state
state ESTABLISHED, RELATED-j ACCEPT
```

After execution of this command firewall will accept all traffic from the internet (interface eth0) to internal network (eth1) which is established and which was initialized from internal network. For communication to the internet is required to allow masquerade which is basically a rule which allows NAT and remember all through and open connection to the internet. This command is shown below:

```
iptables-t nat-A POSTROUTING-o eth0-j
MASQUERADE
```

Firewall device management security: Also firewall device management is important and should be possible only from specific network addresses, this can be configured by this command:

```
iptables-A INPUT-i eth0-s 192.168.2.2-d
192.168.2.1-p tcp-dport 22-j ACCEPT
```

This command allows incoming SSH connection only from 192.168.2.2 to 192.168.2.1 (firewall IP address). Source IP address can be omitted or replaced with address range with network prefix, for example, 192.168.2.0/24.

Prohibiting all communications: It is also appropriate to prohibit all communication through a firewall that has no defined rule. It can be set by this command:

```
iptables-P INPUT DROP
iptables-P FORWARD DROP
iptables-P OUTPUT DROP
```

These commands will ensure that any communication that does not meet any defined rule is dropped.

Permanent keep of firewall rules: Now, it is necessary to ensure that rules will be configured after shutdown or reboot. There is twofold way to ensure this. The first one is that every time device will be turn on some script with firewall settings will be executed. The second way is configuration with package iptables-persistent which provides automatic setup rules during booting. Installing package can be done by this command:

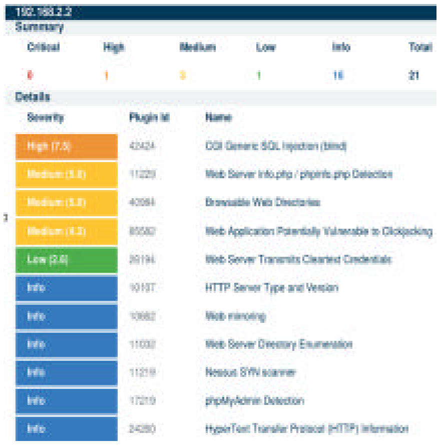
```
apt-get install iptables-persistent
```

After installation all rules for IPv4 communication can be saved by this command:

```
iptables-save>/etc/iptables/rules.v4
```

Current firewall configuration: To view the current firewall configuration can be used the following command:

```
iptables-L-v
```


192.168.2.2

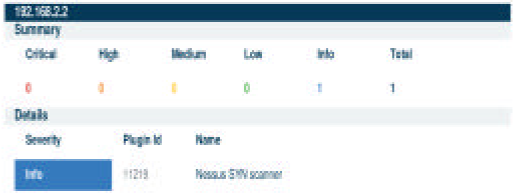
Summary

Critical	High	Medium	Low	Info	Total
0	1	3	1	16	21

Details

Severity	Plugin Id	Name
High (7.5)	42424	CGI Generic SQL Injection (blind)
Medium (5.0)	11229	Web Server info.php / phpinfo.php Detection
Medium (5.0)	40984	Browsable Web Directories
Medium (4.3)	85582	Web Application Potentially Vulnerable to Clickjacking
Low (2.6)	26194	Web Server Transmits Cleartext Credentials
Info	10107	HTTP Server Type and Version
Info	10662	Web mirroring
Info	11032	Web Server Directory Enumeration
Info	11219	Nessus SYN scanner
Info	17219	phpMyAdmin Detection
Info	24260	HyperText Transfer Protocol (HTTP) Information

Fig. 3: Nessus test before firewall activation


192.168.2.2

Summary

Critical	High	Medium	Low	Info	Total
0	0	0	0	1	1

Details

Severity	Plugin Id	Name
Info	11219	Nessus SYN scanner

Fig. 4: Nessus test after firewall activation

In several previous sections there has been described methods for setting firewall rules via. iptables. These commands are only for essential configuration and their syntax can be expanded according to needs. These commands and configurations has been tested during implementation of firewall between internal network and the internet and DMZ network.

Testing: After setting all the rules were tested firewall using Nessus tool which identifies threats and vulnerabilities configurations to prevent network attacks. Figure 3 is shows Nessus test result before firewall rules activation. On this pictures there is shown a lot of threats and vulnerabilities.

Figure 4 shows the Nessus test result after firewall rules activation. This figure shows security was increased. For next firewall test DDoS attack was done via. nap tool which also contains a large number of test scripts to control configuration and network operations and this test provide further information for the other network security.

CONCLUSION

This study objective was describing and proposing solution for creating software firewall on Linux distribution and RPi 3. There were described principles of Linux firewall implementation and basic implementation on the device RPi and the basic configuration of the built-in software firewall solutions. Although, this study deals with RPi 3 and Raspbian Jessie Lite it is possible to use these principles on almost any Linux distributions and implementations of IoT.

In this study two firewall solutions for network separation and filtering network traffic was described- firewall for separating internal network from the internet and firewall for separating DMZ network from internal network. The main difference is that firewall for DMZ network is more restricted. Therefore:

- DMZ network is completely separated and only some connections are allowed which has defined firewall rule.
- Internal network is separated and all connections are allowed but some connections are prohibited.

Unfortunately, the implementation of a firewall will not prevent any attacks and threats in the internal network or on the internet which are primarily:

- Malware
- Attacks from internal network
- Attacks which create alternative paths to network
- Attacks via. allowed protocols (e.g., HTTP or SMTP)
- Unknown threats
- Data interception or modification

For IoT devices should be necessary to take threats that may be related hardware malware from manufacture and it is therefore appropriate to implement a firewall for outgoing data. Furthermore, the tests performed that is evident that to ensure greater security in computer network or the internet, you need to use the firewall.

ACKNOWLEDGEMENT

The support of the specific research project at FIM UHK is gratefully acknowledged.

REFERENCES

Babar, S., P. Mahalle, A. Stango, N. Prasad and R. Prasad, 2010. Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT). In: Recent Trends in Network Security and Applications, Meghanathan, N., S. Boumerdassi, N. Chaki and D. Nagamalai (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-14477-6, pp: 420-429.

Gubbia, J., R. Buyya, S. Marusic and M. Palaniswami, 2013. Internet of things (IoT): A vision, architectural elements and future directions. Future Generat. Comput. Syst., 29: 1645-1660.

Horaek, J., J. Matyska, J. Stepan, M. Vancl and R. Cimler et al., 2015. Lower Layers of a Cloud Driven Smart Home System. In: New Trends in Intelligent Information and Database Systems, Barbucha, D., N.T. Nguyen and J. Batubara (Eds.). Springer, Berlin, Germany, ISBN:978-3-319-16210-2, pp: 219-228.

Purdy, G.N., 2004. Linux Iptables: Pocket Reference. O'Reilly Media, Inc. Sebastopol, USA.,.

Rash, M., 2007. Linux Firewalls: Attack Detection and Response with Iptables, Psad and Fwsnort. No Starch Press, San Francisco, California, ISBN:13:978-59327-141-1, Pages: 291.

Upton, E. and G. Halfacree, 2014. Raspberry Pi User Guide. 3rd Edn., John Wiley and Sons, New York, USA., ISBN: 9781118921661, Pages: 312.