# A study on the Analysis of the refused logs by Internet Firewall

Hiroyuki MINAMI*, Masahiro MIZUTA†

*Abstract* — **Due to unwilling access from the Internet, we have to set various filtering programs and they bring us the related log files. Many Internet security vendors have developed their unique solutions and most are based on their collected data and reputations. It is really useful, however, is not directly applicable for some specific small servers.**

**In the study, we introduce our own dataset by the filter program and spams, and offer empirical results through their analyses.**

**Keyword:** *Cyber security; EDA.*

## 1 Introduction

The word "Incident" is used in the field of information infrastructure and stands for bad matters including information leak, virus affection, phishing and spam.

To protect our own information environment, we introduce some kinds of filtering techniques including spam filter (mainly based on Bayesian approach), firewall. Most filters put their logs to record their activity and we sometimes use them to verify what kind of the incidents it happened and where an attack would come from.

The authors usually run some servers in charge of ourselves (called on-premises style). They are being faced to the attacks from all over the world but we cannot refuse them uniformly since we need reachability for valid access.

Many studies are reported on the matter, mainly as anomaly detection, in the field of statistical approach (and machine learning). Usually they would require massive data to cover the whole attacks, however, just to avoid the frequent ones, we just need to set some filtering rules with our own logs since they would reflect the original trend.

In the study, we try to approach so-called concise analysis of our own data to give a solution to set effective filters.

## 2 Background and Firewall logs

We have analyzed Internet traffic data in lower layer (ICMP echo reply) (Minami, 2010). It gives us some useful interpretation on the current network condition, but doesn't tell us the status on each device since the data consist of the response time between two devices.

Typically, a server in the global Internet have own firewall to protect itself from the unwilling access. A kind of 'UNIX-like' operating system is popular to run a server. Linux is most popular one and we use FreeBSD and it has an original firewall program called `ipfw`.

Figure 1 shows example filtering rules in `ipfw`. The first 2 lines means that all traffics in 192.168.0.0/24 are accepted. The next one means that any access from 10.0.0.0/8 is rejected. The last line is so-called stopping rule that all traffics are prohibited since `ipfw` adopts 'First match' policy.

`ipfw` puts a log in Figure 2.

The first 3 items are date and time and a pair of the items [IP-Address]:[port] is our main target. The former is source IP-Address and port and the latter is destination ones. The destination port stands for a network service. For example, 80 is for HTTP (HyperText Transfer Protocol) known as Web, 22

*Information Initiative Center, Hokkaido University, N11W5, Kita-ku, Sapporo, 060-0811, Japan, E-mail: `min@iic.hokudai.ac.jp`, Tel: +81-11-706-3756

†Information Initiative Center, Hokkaido University, N11W5, Kita-ku, Sapporo, 060-0811, Japan, E-mail: `mizuta@iic.hokudai.ac.jp`, Tel: +81-11-706-3755

```
add pass all from me to 192.168.0.0/24
add pass all from 192.168.0.0/24 to me

add deny log ip from 10.0.0.0/8 to any

add 65000 deny log all from any to any
```

Figure 1: Sample rules on ipfw

```
Sep  9 11:30:34 server kernel: ipfw: 1000 Deny TCP 10.218.200.137:9090 192.168.1.1:22 in via ed0
```

Figure 2: ipfw sample log

means SSH (Secure Shell, for remote terminal access). Then, the example line means that the server rejects the traffic from 10.218.200.137 with the portnumber 9090 to the server for SSH (22/TCP) service.

Figure 3 offers the number of the rejected traffic summarized by destination port and the source ports. In the analysis, we adopt Python packages (Pandas, Numpy, Scipy; McKinney, 2013) since they are applicable, useful and effective in both Unix-like OS and Windows OS. It is straightforward that the

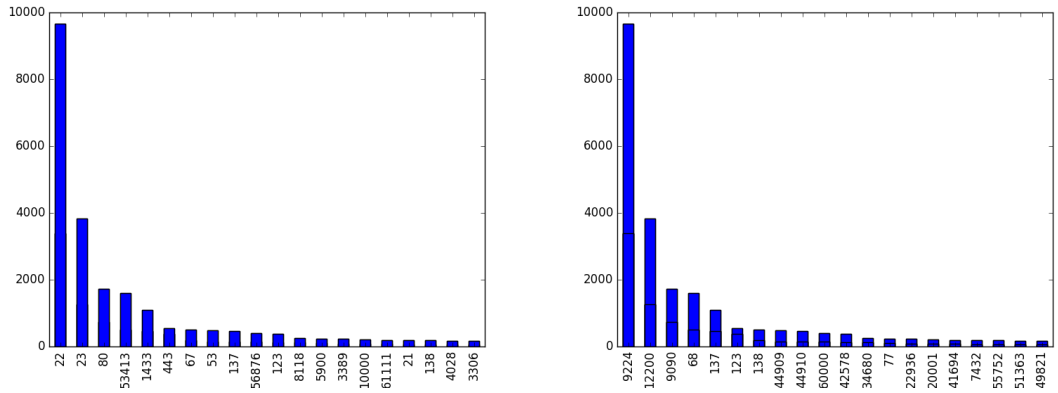


Figure 3: Destination port and Source port

most rejected destination port is 22 since the attacker would like to log-in the server and try to carry out vandals and install their useful applications, and SSH would give him a solution. However, we cannot understand why the most rejected source port is 9090. Usually, it is known that a source port is randomly designated by an operating system, then all source ports would appear randomly. The right plot in Figure 3 offers that the attacker might use a tool whose source port is fixed.

According to our consideration, we refrain from describing the real IP-Address here, but will reveal and give the result with a pair of IP-Address and source port in the presentation.

When we prepare this draft, the number of the log lines is over 30,000 and it is still being accumulated, in spite that the number of the account is just 1. We are going to find a new feature and give additional interpretations.

**References**


[1] M. Collins (2014). Network Security Through Data Analysis. O'Reilly.

[2] W. McKinney (2013). Data Wrangling with pandas, NumPy, and IPython. O'Reilly.

[3] H. Minami (2010). Empirical Studies on the Analysis of a Vast Amount of the Internet Traffic Data in Japan. Abstracts of the 3rd German-Japanese Workshop, University of Karlsruhe, 22