cryptography

MDPI

Article

Anomalous Traffic Detection and Self-Similarity Analysis in the Environment of ATMSim †

Hae-Duck J. Jeong [1,‡,§], WonHwi Ahn [1,§], Hyeonggeun Kim [1,§] and Jong-Suk R. Lee [2,*]

1 Department of Computer Software, Korean Bible University, Seoul 139-791, Korea; joshua@bible.ac.kr (H.-D.J.J.); myroom9@naver.com (W.A.); gudrmsglgl@naver.com (H.K.)

2 Department of Advanced Application Environment Development, National Institute of Supercomputing and Networking, Korea Institute of Science and Technology Information, Daejeon 34141, Korea

* Correspondence: jsruthlee@kisti.re.kr; Tel.: +82-42-869-0521

† This paper is an extended version of our paper published in 2016 10th IEEE International Conference, Innovative Mobile and Internet Services in Ubiquitous Computing.

‡ Current address: 32 Dongil-ro(st) 214-gil, Nowon-gu, Seoul, Korea.

§ These authors contributed equally to this work.

Received: 29 October 2017; Accepted: 6 December 2017; Published: 12 December 2017

Abstract: Internet utilisation has steadily increased, predominantly due to the rapid recent development of information and communication networks and the widespread distribution of smartphones. As a result of this increase in Internet consumption, various types of services, including web services, social networking services (SNS), Internet banking, and remote processing systems have been created. These services have significantly enhanced global quality of life. However, as a negative side-effect of this rapid development, serious information security problems have also surfaced, which has led to serious to Internet privacy invasions and network attacks. In an attempt to contribute to the process of addressing these problems, this paper proposes a process to detect anomalous traffic using self-similarity analysis in the Anomaly Teletraffic detection Measurement analysis Simulator (ATMSim) environment as a research method. Simulations were performed to measure normal and anomalous traffic. First, normal traffic for each attack, including the Address Resolution Protocol (ARP) and distributed denial-of-service (DDoS) was measured for 48 h over 10 iterations. Hadoop was used to facilitate processing of the large amount of collected data, after which MapReduce was utilised after storing the data in the Hadoop Distributed File System (HDFS). A new platform on Hadoop, the detection system ATMSim, was used to identify anomalous traffic after which a comparative analysis of the normal and anomalous traffic was performed through a self-similarity analysis. There were four categories of collected traffic that were divided according to the attack methods used: normal local area network (LAN) traffic, DDoS attack, and ARP spoofing, as well as DDoS and ARP attack. ATMSim, the anomaly traffic detection system, was used to determine if real attacks could be identified effectively. To achieve this, the ATMSim was used in simulations for each scenario to test its ability to distinguish between normal and anomalous traffic. The graphic and quantitative analyses in this study, based on the self-similarity estimation for the four different traffic types, showed a burstiness phenomenon when anomalous traffic occurred and self-similarity values were high. This differed significantly from the results obtained when normal traffic, such as LAN traffic, occurred. In further studies, this anomaly detection approach can be utilised with biologically inspired techniques that can predict behaviour, such as the artificial neural network (ANN) or fuzzy approach.

Keywords: anomalous traffic detection; stochastic self-similar process; hurst parameter; self-similar estimation method; ATMSim; communication network; cryptography

1. Introduction

In recent years, Internet utilisation has increased rapidly. This is predominantly due to the development of information and communication networks and the widespread distribution of smartphones. Statistics Korea shows that Internet accessibility in 2000 was 44.7%, increasing to 78.4% in 2012. This increase in utilisation has spread across various sectors including government, finance, industry, medicine, and private homes [1]. As a result of these useful developments, the Internet has become an invaluable asset to living in modern society.

As the Internet expanded, various services were developed, including web services, social networking services (SNS), Internet banking, and remote processing systems. These developments have significantly enhanced global quality of life. Unfortunately, alongside these positive developments, serious information security problems have also increased and this has led to Internet privacy invasions and various types of network attacks [1,2].

Network attack methods have been increasing in frequency and severity. These attacks can largely be categorised into three types: sniffing, spoofing, and denial-of-service (DoS). Among these three main types of attacks, distributed DoS (DDoS) and spoofing are representative attacks with the highest frequency and impact on contemporary networks [1,3,4]. The attacker's objective usually determines the choice of attack method, but the most damaging attack method used by malicious users is a combination of attacks. First, the system is paralysed through DDoS attack after which a backdoor is installed through spoofing to enable the attacker to extract internal data through secondary attacks [5,6].

Systems typically incorporate protective elements in an attempt to block these illegal Internet attacks. These protective elements include firewalls, intrusion detection systems (IDSs), and intrusion prevention systems (IPSs) [5,7]. Firewalls are utilized to prevent external invasions as much as possible. Unfortunately, not all intrusions can be blocked by firewalls. Approximately one-third of attacks are blocked, but the remaining attacks are able to successfully breach firewalls [5,6]. Consequently, the importance of IPS continues to increase in an attempt to identify attacks through the network after primary prevention methods have failed [8]. However, the existing IPS is limited with regard to processing a diverse range and vast amounts of data.

This paper presents a possible solution to these problems through an IDS model, the Anomaly Teletraffic detection Measurement analysis Simulator (ATMSim), that is able to process diverse and vast amounts of data effectively [9–11]. ATMSim can be described as a data analysis system using Hadoop-based [11], real-time network traffic monitoring and self-similarity [12,13].

This paper is organised as follows. Section 2 describes the concepts of stochastic self-similar processes as well as ARP spoofing and DDoS, and Section 3 presents ATMSim architecture and how the simulator works. Section 4 describes the scenario design and simulation modelling of this study, and Section 5 provides details of the simulation and an analysis of the results. Finally, Section 6 contains the concluding remarks of this paper.

2. Theoretical Background

2.1. Stochastic Self-Similar Processes

This section briefly goes over the mathematical definition of self-similarity, which can be largely divided into deterministic self-similarity and stochastic self-similarity. The former is also known as fractal self-similarity, and includes self-similarity and circulation. The stochastic self-similarity related to this study refers to the auto-correlation function (ACF) property of the new process created by varying the time unit scale, such as seconds, minutes, hours, days, and months, equivalent to the original process from a stochastic perspective [14,15].

Self-similar long-range dependent stochastic processes occur in many natural and man-made systems [16,17]. In particular, since such processes were discovered on the Internet and other multimedia telecommunication networks two decades ago [14,18], they have become the subject of numerous research investigations with respect to their nature and consequences.

Theoretically, one can distinguish two types of stochastic self-similarity. A continuous-time stochastic process $\mathbf{Y}_t = \{Y_{t_1}, Y_{t_2}, \ldots\}$ is strictly self-similar, with a Hurst parameter, $H, 0.5 < H < 1$, if $\mathbf{Y}_{ct}$ and $c^H \mathbf{Y}_t$ (the rescaled process with time scale ct) have an identical finite-dimensional distribution for any stretching factor c, $c > 0$ [15,19,20]. This means that, for any sequence of time points $t_1, t_2 \ldots t_n$,

$$\{Y_{ct_1}, Y_{ct_2}, \ldots, Y_{ct_n}\} \overset{d}{=} \{c^H Y_{t_1}, c^H Y_{t_2}, \ldots, c^H Y_{t_n}\}, \tag{1}$$

where $\overset{d}{=}$ indicates equivalence with regard to distribution. For this reason, this self-similarity, which can be described as "self-similar in a narrow sense", pertains to probability distribution.

When one restricts analysis of stochastic processes to only their initial two moments or their means, variances, and co-variances, weak self-similarity of stochastic processes (or "self-similarity in a broad sense") becomes relevant. For this reason, it is also described as second-order self-similarity. To give an example, let a sequence $\{X_1, X_2, \ldots\}$ be a time-stationary stochastic process, defined at discrete time intervals $i = 1, 2, 3 \ldots$ Let $E[X_i] = EX$, $Var[X_i] = VarX$, and $\rho_k = E[(X_i - EX)(X_{i+k} - EX)]/VarX$ denote the mean, variance, and autocorrelation coefficient of lag k, respectively. When grouping this sequence of random variables into batches of size m, $m \geq 1$, the aggregated process of $\mathbf{X}^{(m)} = \{X_1^{(m)}, X_2^{(m)}, \cdots\}$, at a given aggregation level m, where $X_i^{(m)} = \frac{1}{m}(X_{im-m+1} + \cdots + X_{im}), i \geq 1$. If $\{X_1, X_2, \ldots\}$ can be defined as a self-similar time-stationary process that is weak, with $0.5 < H < 1$, then

$$Var[X_i^{(m)}] = m^{2H-2} VarX, \tag{2}$$

and

$$\rho_k^{(m)} = \rho_k, k \geq 0, \tag{3}$$

where $\rho_k^{(m)}$ is the autocorrelation coefficient of lag k of the aggregated process $\mathbf{X}^{(m)}$. Equation (3) results in the conclusion that the original process as well as its aggregated version correlate identically with regard to structure. Furthermore, it can also be proven that in any weakly self-similar process with $0.5 < H < 1$,

$$\rho_k \sim H(2H-1)k^{(2H-2)}, \tag{4}$$

as $k \to \infty$, and

$$\sum_{k=-\infty}^{k=+\infty} \rho_k = \infty. \tag{5}$$

See for example [15]. Thus, the autocorrelation function decays very slowly (hyperbolically) with k. Because of these properties, these processes are also described as long-range dependent, or strongly correlated.

The self-similarity parameter (also referred to as the Hurst parameter) is the simplest numerical characteristic of stochastic self-similarity and long-range dependence. The purpose of determining the Hurst parameter of stochastic self-similar processes is to analyse the self-similarity property, which, when the Hurst parameter value is at its closest value to 1, shows the highest correlation.

Sequences of real local area network (LAN) traffic as well as anomalous traffic, R/S-statistic plot, and variance–time plot to identify normal and anomalous traffic are the methods used to determine graphic analysis and self-similarity estimation values. The authors of [17,21] provide more detail on self-similarity and relevant estimation methods.

2.2. ARP Spoofing and DDoS

2.2.1. ARP Spoofing

The Address Resolution Protocol (ARP) is used to relate the IP address in the local network to the physical address and to test for IP address (unnecessary ARP process) redundancy.

The Layer2 class communication process between Host A and Host B is as follows:

(1) Host A: Host B IP/MAC address mapping identification check in the local ARP cache
(2) Host A → ARP request: Broadcasting of the Host B IP MAC address
(3) Host B → ARP reply: Transfer of ARP reply including the Host B IP and MAC address to Host A
(4) Host A: ARP cache update

In Figure 1, when Host C interferes between Host A and Host B with malicious intent and takes on the IP address of 192.168.52.2 while disguised as the MAC address of Host C and continuously sends the ARP reply packet to Host A, Host A is unable to check for manipulation, and thus incorrect data is applied to the ARP table.

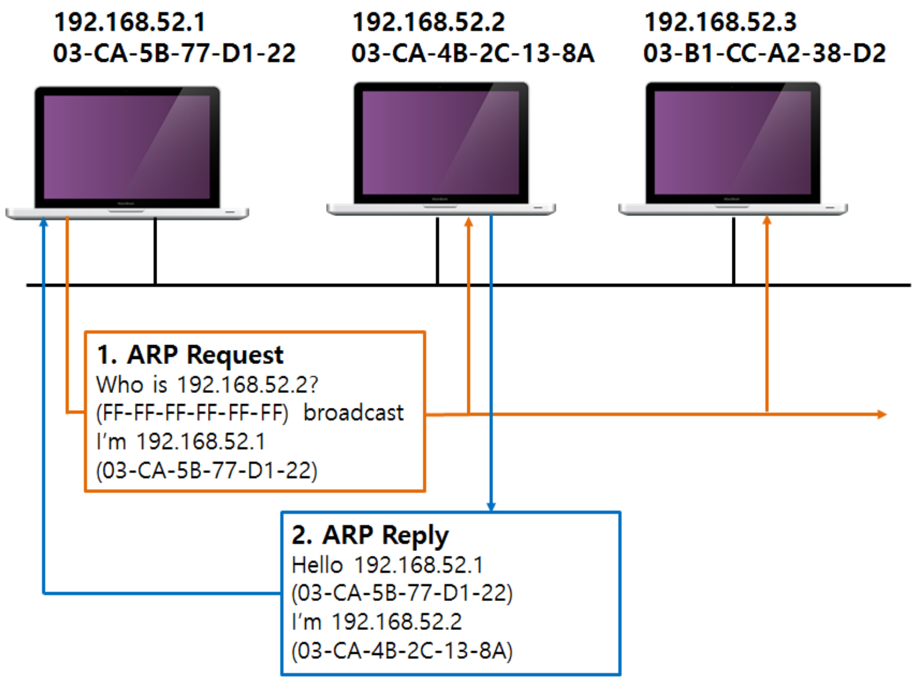


Figure 1. Address Resolution Protocol (ARP) attack.

As seen here, ARP spoofing takes advantage of the ARP protocol vulnerability to modify the ARP cache table data, diverting traffic to the attacker computer in a man-in-the-middle (MITM) attack form [22,23]. Figure 2 shows the ARP spoofing attack process.

(1) IP address .18 ARP spoofing infection
(2) IP address .10 ARP spoofing infection
(3) All communications between the two nodes(.10, .18) are exposed to the hacker

As observed in Figure 2, when infected by ARP spoofing, not only is data leaked to the hacker, but the key point of an ARP spoofing attack is that the infection is unknown.

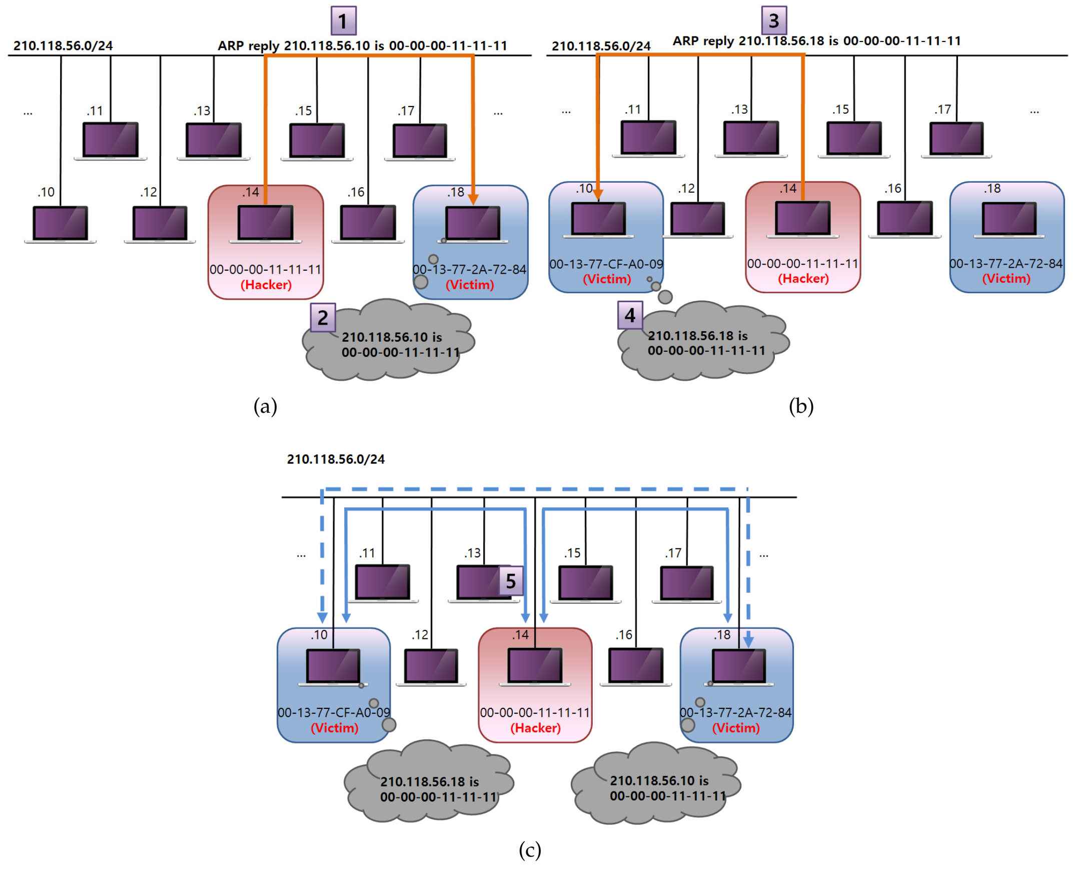

210.118.56.0/24
1
ARP reply 210.118.56.10 is 00-00-00-11-11-11
.10
.11
.12
.13
.14
.15
.16
.17
.18
00-00-00-11-11-11
(Hacker)
00-13-77-2A-72-84
(Victim)
2
210.118.56.10 is 00-00-00-11-11-11
(a)
3
ARP reply 210.118.56.18 is 00-00-00-11-11-11
00-13-77-CF-A0-09
(Victim)
4
210.118.56.18 is 00-00-00-11-11-11
(b)
5
210.118.56.18 is 00-00-00-11-11-11
210.118.56.10 is 00-00-00-11-11-11
(c)

Figure 2. ARP spoofing attack. (**a**) IP Address .18 ARP spoofing infection; (**b**) IP Address .10 ARP spoofing infection; and (**c**) Communication between the two nodes (.10 and .18) was exposed to all hackers.

2.2.2. Distributed Denial-of-Service (DDoS)

There are two forms of denial of service (DoS) attacks: attack from the inside and from the outside. Attacks from the inside are possible with a simple C code or a few lines of script, but the problem with this attack is that the root account has to be accessible. Since most intruders who attack systems are very much interested in first acquiring root access, attacks from the inside are usually the result of mistakes by users rather than being intentional. Unlike attacks from the inside, external attacks are carried out with the intention of attacking the system. Therefore, external attacks are more complex and require a higher level skill. External attack approaches can be largely categorised in the following manner [12,13,24]:

- Application level: They prevent the normal execution of application programs like sendmail, talkd, inetd, and httpd to paralyse the network function of the system.
- Packet level: Can be an attack method at a lower level than the application program level, for example, through packets roaming in the Transmission Control Protocol/Internet Protocol (TCP/IP), or the Ethernet layer, randomly manipulated to paralyse the network function of the target system. A representative example is an SYN flooding attack.
- Network traffic increase: Network traffic is increased to a limited bandwidth to paralyse the network. DoS attacks are approached in the above manner and have the following attack objectives:
- Destructive attack: Damage of disks, data, or systems
- Exhaustion of system resources: Excessive loads on CPU, memory, and disk usage
- Exhaustion of network resources: Depletion of the network bandwidth with garbage data

In accordance with the above objectives of DoS attacks, these attacks seek to hinder the normal operation of the system and a representative attack is DDoS. DDoS attacks utilise an automated tool from distributed locations simultaneously [25,26]. The tools used for DDoS attacks have different names and structures but the basic structure is the same, as shown in Figure 3.

- Attacker: The computer of the hacker leading the attack
- Master: The system that directly receives commands from the attacker and manages the numerous agents
- Handler: The program that performs the above tasks on the program master system
- Agent: The system that directly attacks the attack target
- Demon program: The program that performs the above tasks on the agent system

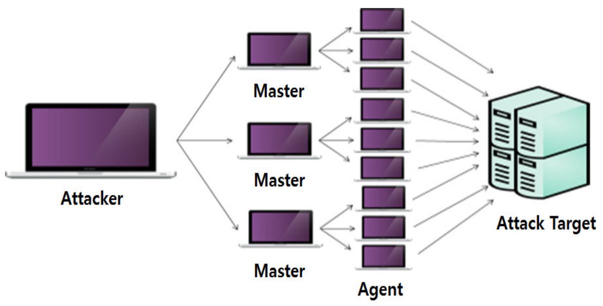


Figure 3. Distributed denial-of-service (DDoS) attack.

When the attacker directly commands the master in the middle, the low-ranking agent directly attacks the target. The detailed attack scenario of DDoS is as follows:

(1) Sniffing or buffer overflow are employed in populous and large-bandwidth networks that cannot be managed in detail to acquire root access.
(2) The network is scanned in sections to determine attack targets and then servers that provide vulnerable services in remote locations are identified.
(3) After identifying the list of vulnerable systems, an exploit list for actual attack is constructed.
(4) The exploit list is compiled and installed on the system with access.
(5) Attack is initiated with the installed exploit.

Since DDoS attack uses an automated tool to move intelligently, as designed by the programming (including irregular variation of the attack time and variable movement of the agent), tracking the attack is very difficult and the scale of damage is significant. Various studies are being pursued to find effective methods to detect DDoS attacks including a DNS sinkhole, network IDS, and SDN. The DDoS infects a zombie PC through the command and control (C & C) server and commences the attack with bots. The zombie PC is constantly communicating with the hacker's PC (C & C server). However, if the relationship between the hacker and the zombie PC is disconnected, the hacker will be unable to control the zombie PC and this prevention method is a DNS sinkhole [27].

Figure 4 shows the already-infected zombie PC requesting the domain and the DNS server responding to the DNS right away with no inspection. The zombie PC receives the domain reply and is connected to the hacker PC to receive commands. However, if there is a sinkhole server, as shown in Figure 4, the sinkhole server is passed through before the request domain and the malicious DNS list previously stored in the sinkhole server is received so that when the zombie PC makes an enquiry to the C & C server, the C & C server IP is converted to the sinkhole IP and damage from the hacker command can be prevented through an IP bypass. However, although the malicious act of the hacker can be blocked using the sinkhole server, there is the disadvantage that already infected zombie PCs cannot be treated. Such zombie PCs are vulnerable with regard to security and are susceptible to other attacks.

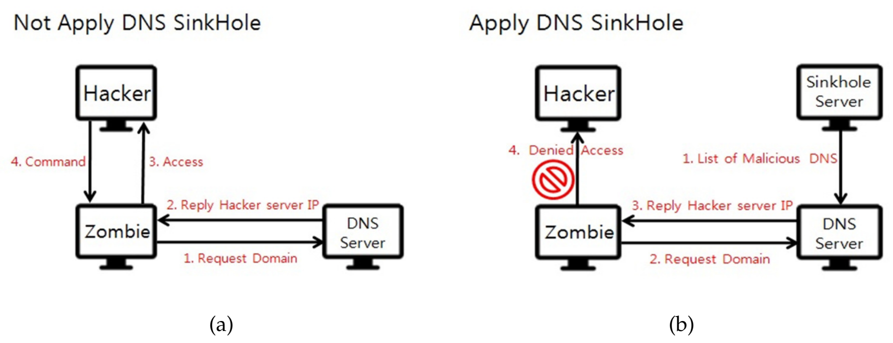


Figure 4. Before and after applying a DNS sinkhole. (**a**) Before applying a DNS sinkhole; (**b**) After applying a DNS sinkhole.

Another DDoS prevention method is the network IDS. Methods include one that detects intrusions based on intrusion pattern data, a method that detects abnormal behaviour detracting from regular behaviour, and a method that simultaneously detects patterns and abnormal behaviour. IDSs can be divided into misuse detection, anomaly detection, and hybrid detection methods. The following are the characteristics, advantages, and disadvantages of each type of IDS.

- Misuse detection: Specific signals are extracted from known attack patterns for intrusion detection, and hence the rate of false positives is low. However, because known attack patterns are used, the rate of false negatives is high for attacks of unknown attack patterns.
- Anomaly detection: A normal range is established and a deviation from that range is determined to be an intrusion. In contrast to the misuse detection method, anomaly detection is able to detect unknown attacks so this method has a low rate of false negatives, but there is the disadvantage that the rate of false positives is high, as attacks within the established normal range are considered normal. Generally, anomaly detection is performed based on stochastic approaches. A variety of techniques is available in addition to the stochastic method including integration of the feature extraction and abnormal behaviour measurement methods [28–31].
- Hybrid detection: This is a network intrusion detection method that combines the advantages of the misuse detection and anomaly detection methods. The pattern storage method of the misuse detection method is used to collect the pattern data, which is then utilised to determine a range that the system uses to detect abnormal phenomena using the anomaly detection method.

The prevention method using an automated IDS by applying the network detection method experiences the phenomenon where the network service is disconnected when the normal bandwidth is exceeded, even for normal traffic, and the phenomenon where the bandwidth is exhausted or service is disconnected because of a time delay after the attack has taken place. In order to resolve these issues, research is being conducted by implementing the buffer concept. The number of anomalous traffic occurrences during a set time interval is stored in the buffer and then expressed as a slope so that detection, prediction, and corresponding prevention are made possible according to the slope. However, this method is limited in response to a specific port and has the disadvantage of not being able to actively respond to multilateral DDoS attacks, and therefore this method requires further improvements.

The following section details the research pursued for effective detection methods of ARP spoofing attacks that can sniff the data of target users by taking advantage of the vulnerabilities of ARP communications, unlike DDoS attacks that create numerous packets. The biggest vulnerability of ARP communications is reliability. The IP address of the desired host is already known and the communication is a static protocol, so MAC addresses are exchanged with no verification process.

Thus, it is inevitably vulnerable in terms of security. Although there is the option of modifying the ARP protocol or employing costly equipment to prevent ARP spoofing attacks, it is realistically difficult to replace expensive equipment, and therefore this method is not practical.

To overcome this disadvantage, various studies are conducted and solutions are presented. Recent studies have focused on decreased network speed, an attack detection method according to the ARP packet reception, regular inspection of the ARP cache table, and ARP spoofing detection identification through routing trace. The ARP cache table real-time inspection and ARP attack detection using routing trace are able to accurately detect ARP spoofing attacks. They overcome the complexity of many proposed algorithms by using the principle of increasing the hop number each time the router is passed through, and no load is experienced by the system and network. However, these methods do not completely resolve all the fundamental security vulnerabilities of the ARP protocol and hence they are still susceptible to new attack methods.

In this paper, ATMSim, which can collect and analyse data for various protocols (unlike that of conventional detection systems for a specific protocol), is used in conjunction with Hadoop, which means that a large amount of data can be processed.

3. ATMSim Architecture and Graphic User Interface

The Anomaly Teletraffic detection Measurement analysis Simulator, abbreviated as ATMSim, is a simulator of anomalous traffic analysis [11]. Java is the programming language that was used for the development environment. This tool is intended for use on a 64-bit Windows 7 operating system as well as Linux. ATMSim can be described as an intrusion detection system which utilizes the high-capacity storage and processing system of Hadoop combined with the self-similarity concept. This combination of Hadoop and the self-similarity concept enables the tool to overcome the limitations of the IDS, specifically that is able to detect only known attacks and has an inability to store various types of data for use in qualitative identification of anomalous traffic. This system was also designed for the purpose of autonomous simulation testing. This was achieved with the incorporation of an attack component to enable self-study of intrusion detection.

3.1. ATMSim Architecture

The anomalous traffic detection method of ATMSim can be largely divided into four parts: collection, storage, analysis, and graphic user interface (GUI).

- The collector module acquires data from web page data, SNS data, and the system log (e.g., Facebook, Twitter) and distributes the data to file collection tools, data collection robots, and data aggregators.
- The storage module is stored and managed by the file storage, and real-time analysis and filtering are performed through data storage and structure data storages.
- The analysis and cluster of the analyser module classify the parallel and distributed data. The analyser module is responsible for the important roles of content analysis, explanation analysis, prediction analysis, natural language processing, and text mining. Also, anomaly-based teletraffic intrusion detection systems (AT-IDSs) detect anomalous traffic using the MapReduce framework.
- Real-time statistical data is acquired in this step and the statistical analysis results are automatically provided.
- Figure 5 shows the ATMSim system architecture.

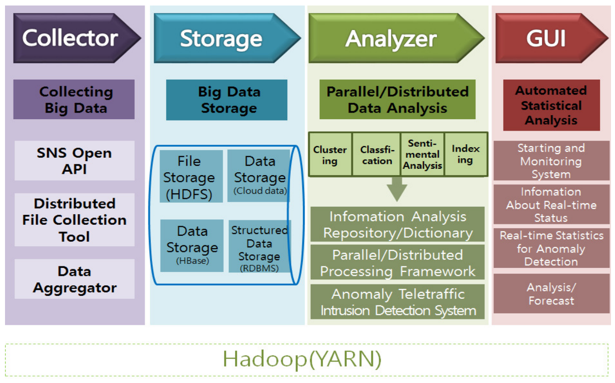

Collector
Collecting Big Data
SNS Open API
Distributed File Collection Tool
Data Aggregator
Storage
Big Data Storage
File Storage (HDFS)
Data Storage (Cloud data)
Data Storage (HBase)
Structured Data Storage (RDBMS)
Analyzer
Parallel/Distributed Data Analysis
Clustering
Classification
Sentimental Analysis
Indexing
Infomation Analysis Repository/Dictionary
Parallel/Distributed Processing Framework
Anomaly Teletraffic Intrusion Detection System
GUI
Automated Statistical Analysis
Starting and Monitoring System
Infomation About Real-time Status
Real-time Statistics for Anomaly Detection
Analysis/ Forecast
Hadoop(YARN)

Figure 5. Anomaly Teletraffic detection Measurement analysis Simulator (ATMSim) architecture. SNS: social networking services; GUI: graphic user interface; HDFS: Hadoop Distributed File System.

ATMSim is a simulator tool for extraction, measurement, and analysis of the abnormal incoming communications traffic from the real-time network communications traffic. ATMSim includes a network attack tool for additional functionality to create pseudorandom numbers in abnormal communications traffic for network security simulations. In order to use ATMSim, installation of the Jpcap library is necessary for packet analysis. Once the Jpcap library appropriate for the operating system is installed, ATMSim can be executed with no platform restrictions if JVM is installed as it is a Java-based program [11,32].

3.2. ATMSim Operation

ATMSim shows real-time visualizations of the detected traffic it has analysed. Please refer to the detailed system structure depicted in Figure 6 to facilitate understanding of the traffic detection method. Because ATMSim is a package that has been developed with Java, Jpcap is used to detect the packets. Furthermore, the JavaFx Library is also used to depict graphs of the real-time and total-time traffic. In the real-time graph the data of packets with source and destination addresses can be seen in graph form using JavaFx for seconds or even smaller time units. ATMSim was also designed with long-term detection analysis capability. In the total-time graph the total analysis time can be seen.

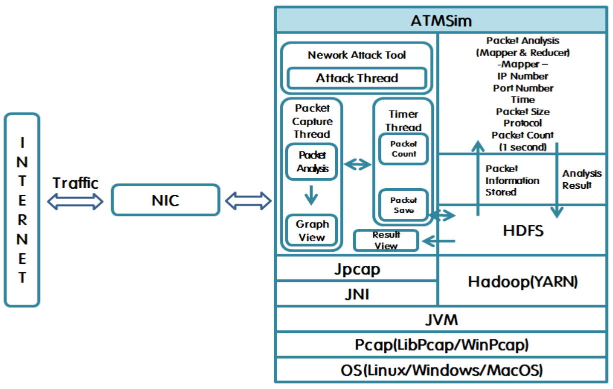


Figure 6. Detailed ATMSim architecture. NIC: network interface card.

3.3. Apache Hadoop YARN

Hadoop is a Java-based open source framework that allows for distributed processing of large data sets. Hadoop is the result of Doug Cutting in 2005 implementing the Google File System (GFS) and MapReduce as published in the paper by Google Labs. Hadoop stores data in the Hadoop Distributed File System (HDFS) and uses the distributed processing system MapReduce to process the data. For Hadoop 1.X, the Job Tracker, the key to the Map Reduce (MR) processing, manages all the resources of the cluster. Unfortunately, it has the problem of slow processing because both management and monitoring are conducted until the successful completion of all the numerous jobs being performed. In Hadoop 2.X, the key element is employing the Apache Hadoop YARN resource management platform to focus on the resource allocation and monitoring necessary for each application of the Hadoop cluster so that various applications could share the Hadoop cluster resources.

For the Hadoop storage system HDFS, the storage can be constructed using low-specification servers. Unlike the conventional high capacity file systems, HDFS allows for tens of hundreds of web server class servers to be used as single storage. Here, the data stored in the HDFS are physically stored in the local disks of the distributed server, but controls like file reading and writing are processed using the APIs provided by HDFS. The structure of HDFS is shown in Figure 7.

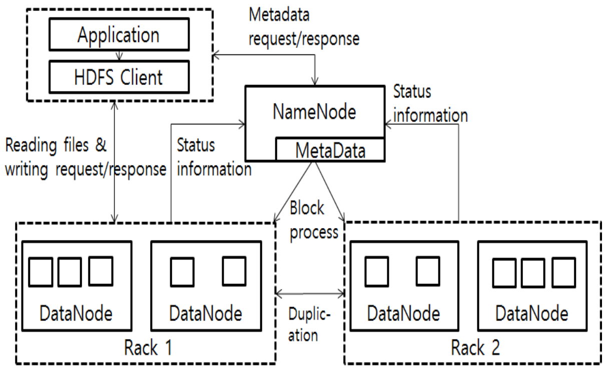


Figure 7. HDFS architecture.

HDFS is composed of a master–slave architecture. Hence, there is one NameNode server that is the master and multiple DataNode servers that are the slaves. The NameNode manages all the metadata of the HDFS and allows the client to access all the files stored in the HDFS. When storing in HDFS, the data divided into blocks are distributed in multiple DataNodes.

For Hadoop 2.X YARN, the MapReduce (MR) structure was newly designed to supplement its disadvantages, resulting in a different structure. MR of Hadoop 2.X is called YARN or MRv2 and the main functions of the job tracker, resource management and job life-cycle management, are separated into new components [27]. The Resource Management component is responsible for the management and allocation of resources for the entire cluster and the resources are reclaimed when tasks are completed. The new component Application Master is responsible for the task scheduling and modifications for each application. Applications are distributed and executed through multiple nodes and a node manager exists for each node. Figure 8 shows the Hadoop YARN structure, which is the new MapReduce.

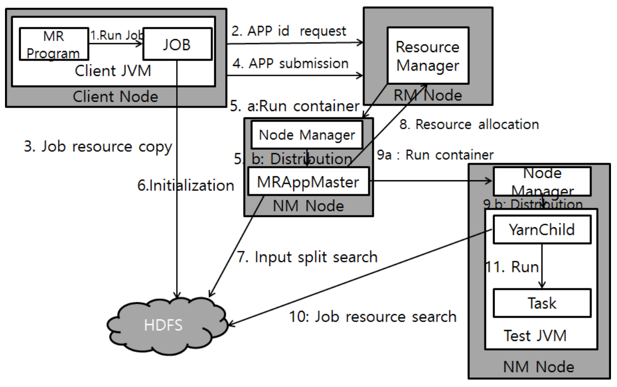


Figure 8. MapReduce Architecture.

3.4. The AT-IDS on Hadoop

An intrusion detection system (IDS) is one security method that aims to strengthen security through overall monitoring of data and internal systems. Analysis approach methods can be categorised into signature-based teletraffic intrusion detection systems (ST-IDSs) and anomaly-based teletraffic intrusion detection systems (AT-IDSs). ST-IDSs require previous attacks to be stored in a database beforehand, with the result that the method is not effective against unknown or new attacks. On the other hand, AT-IDSs look for cases where the deviation between the occurred event and normal behaviour is greater than the defined threshold to detect an attack. This can effectively detect unknown or new attacks. Therefore, the AT-IDS system, which can effectively detect anomaly traffic, was applied.

AT-IDSs can be categorised according to the approach method: stochastic-based, knowledge-based, and machine learning-based methods. Table 1 shows the advantages and disadvantages of these methods. The disadvantages of AT-IDSs, as shown in Table 1, lead to numerous difficulties in detecting normal and anomalous traffic, and largely there are three problems:

- Storage: A large amount of data cannot be stored in the current system.
- Speed: When considering a large amount of data and the processing speed of the computer, the processing speeds of at least a supercomputer are necessary.
- Variety: A diverse range of data including images, text, and video need to be processable. Taking into consideration the above three problems, this study proposes a new platform detection system applying Hadoop to the AT-IDS, overcoming the problems through the high-capacity data storage of the Hadoop HDFS and the processing system MR.

Table 1. Advantages and disadvantages of anomaly-based teletraffic intrusion detection systems (AT-IDS techniques).

Approaches	**Techniques**	**Advantages and Disadvantages**
Statistical-based	- Univariate models - Multivariate models - Time series models - Self-similar models	- Prior knowledge about normal activity not required - Accurate notification of malicious activities - Susceptible to be trained by attackers - Difficult setting for parameters and metrics - Unrealistic quasi-stationary process assumption
Knowledge-based	- Finite state machines - Description languages - Expert systems	- Robustness - Flexibility and scalability - Difficult and time-consuming availability for high-quality knowledge/data
Machine learning-based	- Markov models - Genetic algorithms - Neural networks - Bayesian networks - Fuzzy logic - Clustering and outlier detection	- Flexibility and adaptability - Capture of interdependencies - High dependency on the assumption about the behaviour accepted for the system - Highly resource-consuming

4. Scenario Design and Simulation Modeling

Each scenario of this study was designed to reproduce an external attacker's intrusion on an internal server with ATMSim installed at the Korean Bible University. The network attacks were DDoS and spoofing, and the normal and anomalous traffic was collected. Taking into consideration that the attack occurs at an unspecified time, the traffic collection time for each scenario was limited to two days.

4.1. Scenario Design

The aim of this study was to compare and analyse the normal and anomalous traffic using self-similarity. For this, a total of four scenarios was considered according to the attack method and whether the tool was used or not. The tool was added to the scenarios in order to observe how similar the attack provided through ATMSim is compared to a real attack. All four scenarios were limited to two days for data collection and the collection environment was designed based on the incoming traffic from an external LAN connection to the Korean Bible University. The four scenarios considered in this study were normal traffic, DDoS attack, ARP spoofing, and DDoS and ARP attack.

For the normal LAN traffic scenario, the incoming traffic from the external LAN connection to the Korean Bible University internal LAN was assumed to be normal traffic and ATMSim was used for detection, analysis, and determination of the self-similarity estimation value for the normal traffic. For the DDoS attack scenario, the self-similarity estimation value was determined by collecting the User Datagram Protocol (UDP) flooding attack traffic by an external attacker to the Korean Bible University's internal server. Although it was predicted that the traffic amount would be proportional to the number of agents, it may be similar due to the amount of loss and drop packets.

In the case of the ARP spoofing scenario, the self-similarity estimation value was calculated by collecting the ARP packet that modified the server Mac table after being transmitted to the server. Visual detection of ARP spoofing is possible but it was predicted that the self-similarity estimation value would be similar to that of the normal traffic case.

For the DDoS and ARP spoofing scenario, the self-similarity estimation value was determined by sending the ARP spoofing packet during a DDoS (UDP flooding) attack. It is easily conceived that detecting the ARP spoofing packet during DDoS may not be possible as the server and system would go down due to the DDoS. It was predicted that detection would be possible for non-destructive attacks since the packets entering through the network interface card (NIC) can be collected using Jpcap and then transferred to the ATMSim.

4.2. Simulation Modeling

The simulation implemation tool utilized for this study was the ATMSim for the DDoS aspect [24–26]. NetBotAttacker 2.3 was used to send commands to the agent (in this case a zombie PC). The reason for using a zombie PC was that the master–slave structural characteristic is ideal for recreating this type of scenario. NetBot Attacker 2.3 was used from the external attacker PC to scan the zombie PC. After that, four zombie PCs were also selected at random, to complete the scenario.

With regard to the Address Resolution Protocol (ARP) spoofing attack scenario [22,23,33], the attacker installed in the ATMSim first sent a broadcasting packet. Then, it identified the IP and Mac address mapping values after which it modified the attacker Mac address of the Korean Bible University internal server MAC table where ATMSim was installed using an ARP spoofing attack. This aspect is one of ATMSim's NetworkAttack functionalities.

The method to simulate the DDoS and ARP spoofing attacks scenario first involved the use of the DDoS attack on the internal server where the ATMSim was installed. This is one of the NetworkAttack functionalities of the ATMSim. This was followed by the simultaneous sending of the ARP spoofing packet for detection. Refer to Figures 9–11 to see the design diagram relevant to these scenarios.

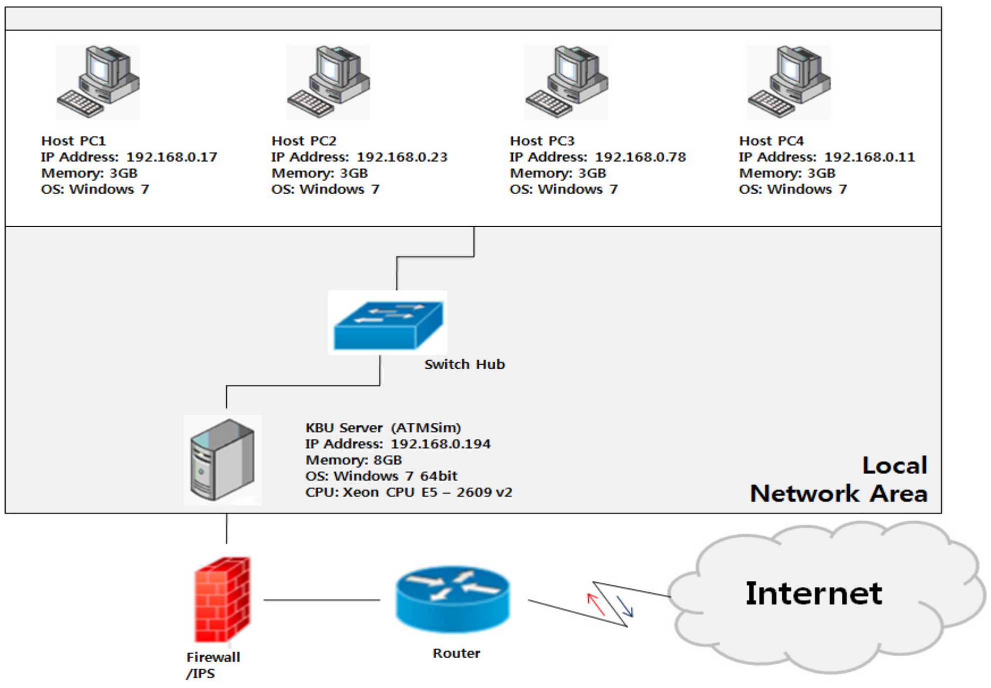


Figure 9. Simulation for collecting real local area network (LAN) traffic.

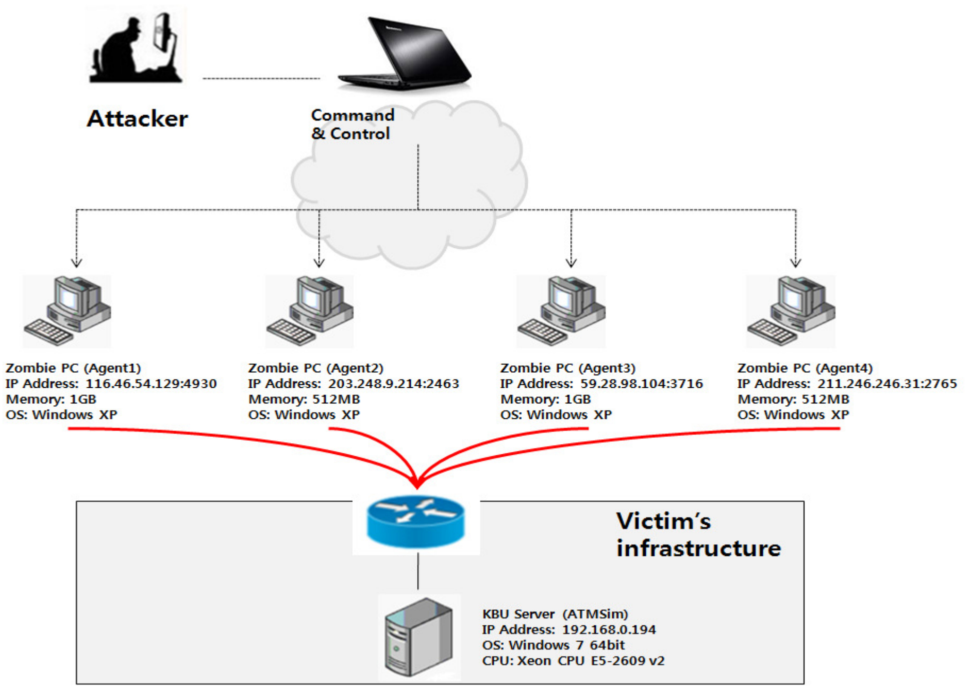


Figure 10. Simulation for collecting real LAN traffic with a DDoS attack [10].

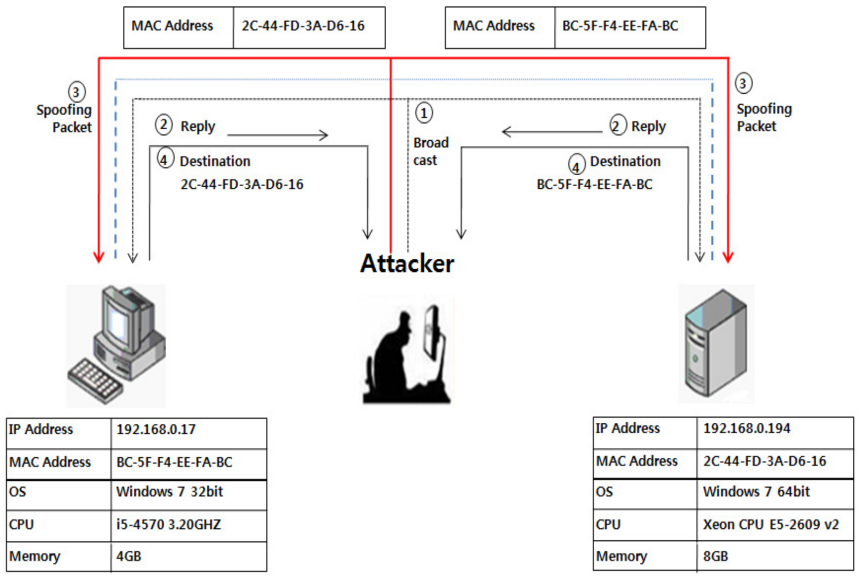


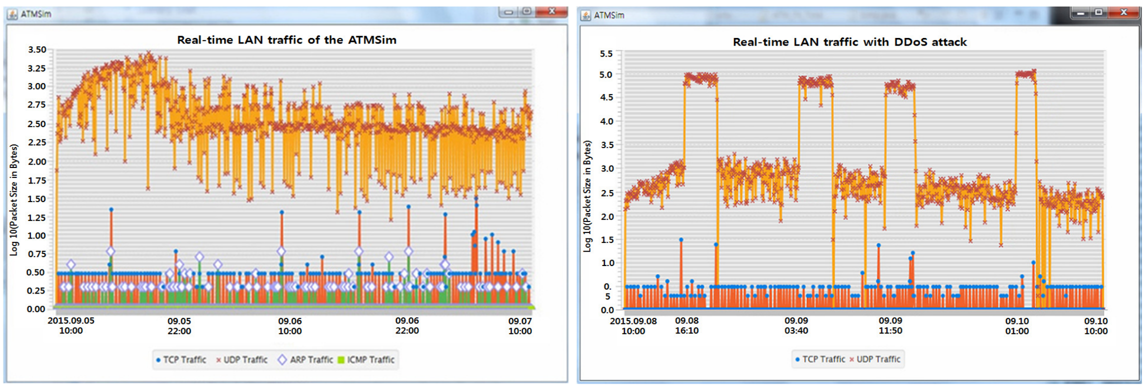
Figure 11. Simulation for collecting real LAN traffic with an ARP spoofing attack [10].

5. Simulation and Result Analysis

This study involved four designed scenarios. A simulation was performed for each scenario and each simulation was conducted for 48 h. The Transmission Control Protocol (TCP) and UDP were commonly included in order to evaluate the detection results of the ATMSim system for each scenario. Furthermore, ARP was included for scenarios in which it was required for comparative purposes, which were set by the packet number.

The normal LAN traffic scenario was conducted first. The Korean Bible University server was used and the normal traffic from the external LAN connection with ATMSim installed was detected. The result of this scenario can be seen in Figure 12a.

The traffic for this scenario was measured for the period from 5–7 September 2015. The UDP packet numbers in log scale were 2.50 for the UDP packet and 0.50 for the TCP packet. The self-similarity estimation values were 0.5970 from the R/S-statistic estimation method, and 0.6045 from the variance–time estimation method. Figures 13a and 14a depict this aspect of the simulation.



(a) (b)

Figure 12. Sequence of real LAN traffic and LAN traffic with DDoS attack. (**a**) Real LAN traffic obtained from ATMSim; (**b**) Real LAN traffic with DDoS attack [10].

The next phase was the DDoS attack scenario which can be seen in Figure 12b. Traffic was measured for the period 8–10 September 2015. The first attack occurred on 8 September, with a 100-thread UDP flooding packet size(log) of 5.0. The second attack occurred on 9 September, with a changed 20-thread UDP flooding packet size of around 4.9. There was no significant difference between the two attacks. The third attack differed from the first two attacks in that it was started by the ATMSim autonomously on 9 September at 11:50 a.m. The UDP flooding packet size value was between 4.7 and 4.8. One zombie PC was involved. The fourth attack, similar to the third, was also started by the ATMSim autonomously on 10 September. In this case, the UDP flooding value was between 5.1 and 5.2 and four zombie PCs were involved. The average packet number for the DDoS attack was measured as 5.0 throughout the attack. The autonomous attack performance of the ATMSim was observed to be in effect equal to the attack tool. With regard to the self-similarity estimation values of the DDoS scenario, the R/S-statistic estimation method was determined to be 0.7737, and the value obtained from the variance-time estimation method was 0.8342. Refer to Figures 13b and 14b in this regard.

Figure 15a depicts the results of the ARP spoofing scenario that included the ARP packet to make detection possible. Traffic was measured from 12–14 September 2015 The first attack occurred on 13 September and the second attack occurred on 14 September. The ARP packet size of each attack was 2.0. The self-similarity estimation value for the R/S-statistic estimation was 0.8001 and the variance–time estimation value was 0.7700. Figures 16a and 17a show that the difference between the traffic in this simulation and normal traffic was significant.

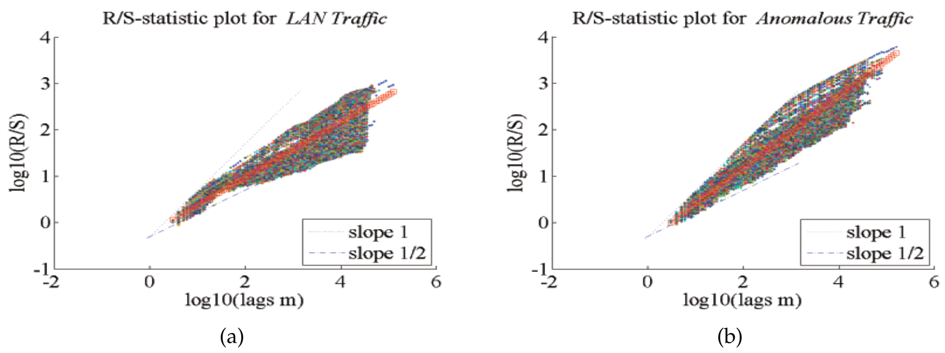


Figure 13. R/S-statistic plot(1/2). (**a**) Real-time LAN traffic; (**b**) Real-time LAN traffic with a DDoS attacks [10].

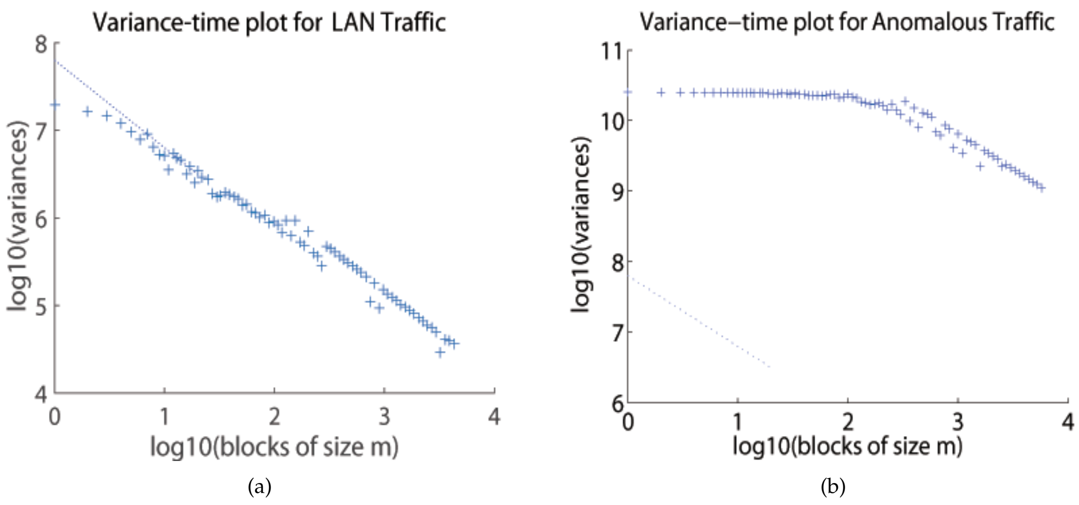
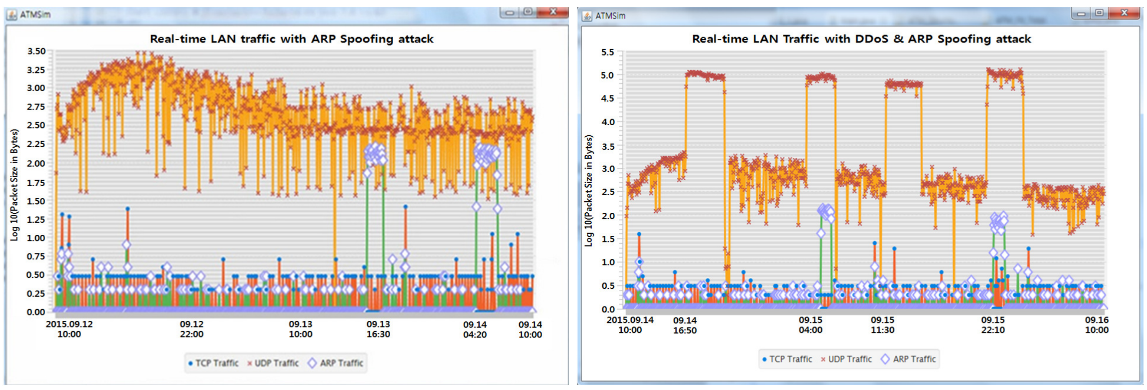
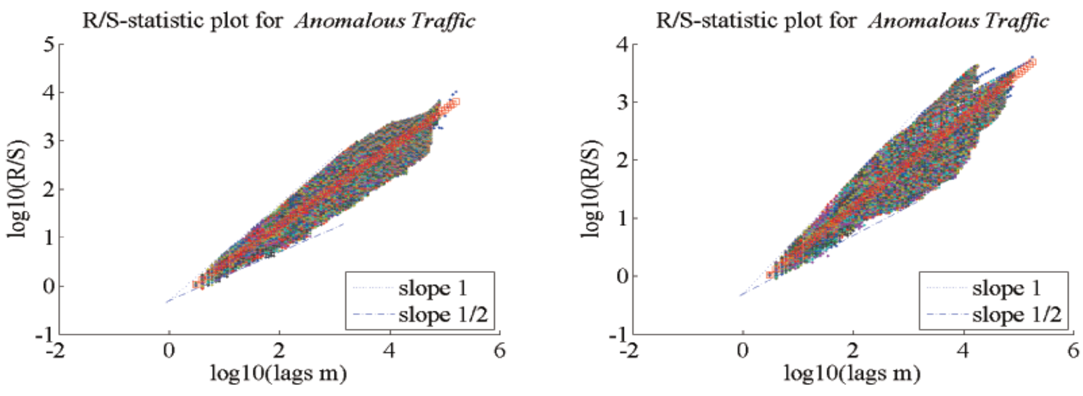


Figure 14. Variance–time plot(1/2). (**a**) Real-time LAN traffic; (**b**) Real-time LAN traffic with a DDoS attacks [10].



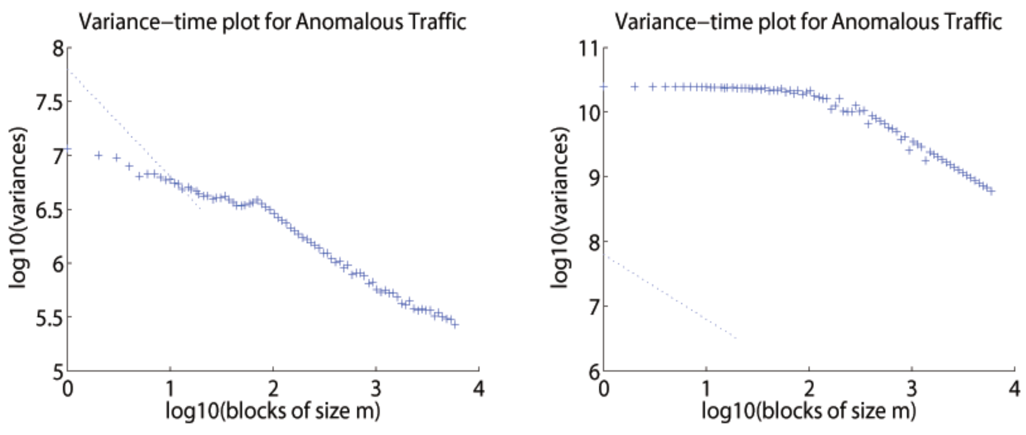
(a) (b)

Figure 15. Sequence of real LAN traffic with ARP spoofing attack; and real LAN traffic with DDoS and ARP spoofing attacks. (**a**) Real LAN traffic with an ARP spoofing attack; (**b**) Real LAN traffic with DDoS and ARP spoofing attacks [10].



(a) (b)

Figure 16. R/S-statistic plot(2/2). (**a**) Real-time LAN traffic with ARP spoofing attack; (**b**) Real-time LAN traffic with DDoS and ARP spoofing attacks [10].



(a) (b)

Figure 17. Variance-time plot(2/2). (**a**) Real-time LAN traffic with an ARP spoofing attack; (**b**) Real-time LAN traffic with DDoS and ARP spoofing attacks [10].

For the final scenario four DDoS attacks were made and alternated with ARP spoofing with a time difference. Figure 15b shows the results of this final scenario. Traffic was measured for the period from

14 September 2015 to 16 September 2016. The first attack occurred on 14 September and involved only a UDP flooding attack. The second attack involved UDP flooding and ARP spoofing simultaneously and occurred on 15 September. In this scenario, the server went down. Even so, the ARP packet measurement was successful and the rest of the scenario progressed as it did before the server went down. The UDP flooding packet size value was 5.0 and the ARP spoofing packet size value was 2.0. These values were the in size as the results in the previous scenario. The self-similarity estimation values were obtained from the R/S-statistic estimation method. The self-similarity estimation value was 0.7664, and the value of the variance–time estimation was 0.7901. Refer to Figures 16b and 17b in this regard.

6. Conclusions

This paper described a possible solution in the form of a research method for the problem of detecting anomalous traffic by using the self-similarity analysis in the ATMSim environment. The ATMSim is a new platform that utilizes Hadoop. With this tool, anomalous traffic was identified and a comparative analysis of anomalous and normal traffic was performed using the process of self-similarity analysis. The methodology described in this paper involved the measurement of anomalous traffic as well as normal traffic. ARP spoofing and DDoS attacks were measured for a period of 48 h each. Four different types of traffic were analysed quantitatively using self-similarity estimation and the results showed the presence of a burstiness phenomenon when anomalous traffic occurred and the self-similarity estimation values were elevated. This is a deviation from what can be observed in normal traffic, such as LAN traffic. Future studies using this anomaly detection approach can utilize techniques of a biological nature to predict behaviour such as artificial neural networks (ANN) or the fuzzy approach [34–37].

Acknowledgments: The authors would like to give thanks to the funding agencies for providing financial support. Parts of this work were supported by a research grant from Korean Bible University and Korea Institute of Science and Technology Information. The authors also thank Susan Elizabet Nel, Myeong-Un Ryu, Sang-Hyun Lim, and Sang-Kug Ye for their constructive remarks and valuable comments.

Author Contributions: Hae-Duck J. Jeong and Jong-Suk R. Lee conceived and designed the experiments; WonHwi Ahn and Hyeonggeun Kim performed the experiments; Hae-Duck J. Jeong and Jong-Suk R. Lee analysed the data; WonHwi Ahn and Hyeonggeun Kim contributed reagents/materials/analysis tools; Hae-Duck J. Jeong and Jong-Suk R. Lee wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
ARP	Address Resolution Protocol
AT-IDS	Anomaly-based Teletraffic Intrusion Detection System
ATMSim	Anomaly Teletraffic detection Measurement analysis Simulator
DoS	Denial-of-Service
DDoS	Distributed Denial-of-Service
GFS	Google File System
HDFS	Hadoop Distributed File System
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
LAN	Local Area Network
MITM	Man-In-The-Middle Attack
MR	Map Reduce
NIC	Network Interface Card

SNS	Social Networking Services
ST-IDS	Signature-based Teletraffic Intrusion Detection System
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol

References

1. Kim, J. A Study on Mitigation Mechanism of DDoS Attacks Utilizing SDN. Master's Thesis, Department of Information Security, Dongguk University, Seoul, Korea, 2015.
2. Kim, J. *Influence and Response of the Upsurge of Internet Information*; Technical Report; Samsung Economic Research Institute: Seoul, Korea, 2007.
3. Hong, S. An Efficient Prevention Technique Using the Reliable ARP Table for ARP Spoofing Attacks. Master's Thesis, Department of Computing, Soongsil University, Seoul, Korea, 2011.
4. Cho, J. A Study of Security Authentication Mechanism for DDoS Attack Prevention in the Internet Environment. Master's Thesis, Department of Infomation & Communications, Gwangju University, Gwangju, Korea, 2012.
5. Lee, H. Design of Effective Corresponding Model as a Security Policy Against Attacks of IP Spoofing in Enterprise Networks. Ph.D. Thesis, Department of Computer Science, Gyeongsang National University, Jinju, Korea, 2013.
6. Choi, J. Education System Design and Implementation of Network Attacks. Ph.D. Thesis, Ewha Womans University Graduate School of Education, Seoul, Korea, 2006.
7. Ha, D.J. Network-Based Intrusion Detection Firewall. Master's Thesis, Yeungnam University Graduate School, Gyeongsan, Korea, 2001.
8. Lee, S. Detection of Bandwidth Exhausting Attacks Using an Improved Prediction Method and its Responding Model. Ph.D. Thesis, Department of Computer Science, Gyeongsang National University, Jinju, Korea, 2015.
9. Callegari, C.; Coluccia, A.; D'Alconzo, A.; Ellens, W.; Giordano, S.; Mandjes, M.; Pagano, M.; Pepe, T.; Ricciato, F.; Zuraniewski, P. Chapter A Methodological Overview on Anomaly Detection. In *Data Traffic Monitoring and Analysis*; Lecture Notes in Computer Science; Biersack, E., Callegari, C., Matijasevic, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7754, pp. 148–183.
10. Jeong, H.D.; Kim, H.; Ahn, W.; Oh, J.; Lee, D.; Ye, S.K.; Lee, J.R. Analysis and Detection of Anomalous Network Traffic. In Proceedings of the 2016 10th IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Fukuoka, Japan, 6–8 July 2016; pp. 403–408.
11. Lee, J.; Ye, S.K.; Jeong, H.D. ATMSim: An Anomaly Teletraffic Detection Measurement Analysis Simulator. *Simul. Model. Pract. Theory* **2014**, *49*, 98–109.
12. Li, M. An Approach to Reliably Identifying Signs of DDoS Flood Attacks Based on LRD Traffic Pattern Recognition. *Comput. Secur.* **2004**, *23*, 549–558.
13. Li, M. Change Trend of Averaged Hurst Parameter of Traffic under DDoS Flood Attacks. *Comput. Secur.* **2006**, *25*, 213–220.
14. Leland, W.; Taqqu, M.; Willinger, W.; Wilson, D. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE ACM Trans. Netw.* **1994**, *2*, 1–15.
15. Park, K.; Willinger, W. Chapter Self-Similar Network Traffic: An Overview. In *Self-Similar Network Traffic and Performance Evaluation*; Park, K., Willinger, W., Eds.; John Wiley & Sons, Inc.: New York, NY, USA, 2000; pp. 1–38.
16. Jeong, H.D.; Lee, J.S.; McNickle, D.; Pawlikowski, K. Self-Similar Properties of Malicious Teletraffic. *Int. J. Comput. Syst. Sci. Eng.* **2012**, *28*, 1–7.
17. Jeong, H.D.; Lee, J.S.; McNickle, D.; Pawlikowski, K. Comparison of Various Estimators in Simulated FGN. *Simul. Model. Pract. Theory* **2007**, *15*, 1173–1191.
18. Leland, W.; Taqqu, M.; Willinger, W.; Wilson, D. On the Self-Similar Nature of Ethernet Traffic. In Proceedings of the ACM SIGCOM'93, San Francisco, CA, USA, 13–17 September 1993; pp. 183–193.
19. Beran, J. *Statistics for Long-Memory Processes*; Chapman and Hall: New York, NY, USA, 1994.
20. Taqqu, M. Self-Similar Processes. In *Encyclopedia of Statistical Sciences*; Kotz, S., Johnson, N., Eds.; John Wiley and Sons, Inc.: New York, NY, USA, 1988; Volume 8.

21. Taqqu, M.; Teverovsky, V.; Willinger, W. Estimators for Long-Range Dependence: An Empirical Study. *Fractals* **1995**, *3*, 785–788.
22. Bae, J. Network Access Control and Management Using ARP Spoofing in Various Windows Environment. Master's Thesis, Department of Electrical and Computer Engineering, Sungkyunkwan University Graduate School, Suwon, Korea, 2011.
23. Wagner, R. *Address Resolution Protocol Spoofing and Man in-the-Middle Attacks*; SANS Institute InfoSec Reading Room: Singapore, 2001.
24. Lee, J.Y.; Yoon, M.S.; Lee, H. Monitoring and Investigation of DoS Attack. *KNOM Rev.* **2004**, *6*, 21–32.
25. Kaspersky. Lab. *Kaspersky DDoS Intelligence Report Q2 2015*; Kaspersky: Moscow, Russia, 2015.
26. Seo, S. Detection and Protection of DoS/DDoS Attacks in WiBro System. Master's Thesis, School of Engineering, Information and Communications University, Daejeon, Korea, 2009.
27. Huang, J.; Nicol, D.M.; Campbell, R. Denial-of-Service Threat to Hadoop/YARN Clusters with Multi-tenancy. In Proceedings of the IEEE International Congress on Big Data, Anchorage, AK, USA, 27 June–2 July 2014; pp. 48–55.
28. Coluccia, A.; D'Alconzo, A.; Ricciato, F. Distribution-based anomaly detection via generalized likelihood ratio test: A general Maximum Entropy approach. *Comput. Netw.* **2013**, *57*, 3446–3462.
29. Fiadino, P.; D'Alconzo, A.; Schiavone, M.; Casas, P. Challenging Entropy-based Anomaly Detection and Diagnosis in Cellular Networks. In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15), London, UK, 17–21 August 2015; pp. 87–88.
30. Karatepe, I.; Zeydan, E. Anomaly Detection in Cellular Network Data Using Big Data Analytics. In Proceedings of the 20th European Wireless Conference, Barcelona, Spain, 14–16 May 2014; pp. 1–5.
31. Sun, B.; Yu, F.; Wu, K.; Leung, V. Mobility Based Anomaly Detection in Cellular Mobile Networks. In Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe'04), Philadelphia, PA, USA, 1 October 2004; pp. 61–69.
32. Jeong, H.D.J.; Ryu, M.U.; Ji, M.J.; Cho, Y.B.; Ye, S.K.; Lee, J.S.R. DDoS Attack Analysis Using the Improved ATMSim. *J. Internet Comput. Serv.* **2016**, *17*, 19–28.
33. Song, M. A study on Detection and Protection for ARP Spoofing Attack based on Routing Trace. Master's Thesis, Department of Computer Engineering, National University of Science and Technology, Seoul, Korea, 2014.
34. Ogiela, M.; Ogiela, U. Towards Cognitive Cryptography. *J. Internet Serv. Inf. Secur.* **2014**, *4*, 58–63.
35. Ogiela, M.; Ogiela, U. Bio-inspired Approaches for Secret Data Sharing Techniques. In Proceedings of the Signal Processing, Computer Networks and Telecommunications (ICIIBMS), Okinawa, Japan, 28–30 November 2015; pp. 75–78.
36. Ogiela, M.; Ogiela, U. Bio-Inspired Cryptographic Techniques in Information Management Applications. In Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications, Crans-Montana, Switzerland, 23–25 March 2016; pp. 1059–1063.
37. Wang, G.; Hao, J.; Ma, J.; Huang, L. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Syst. Appl.* **2010**, *37*, 6225–6232.