*Journal Computer Science*

Original Research Paper

# Scorpius: sFlow Network Anomaly Simulator

**[1]Marcos V.O. de Assis and [2]Mario Lemes Proença Jr.**

*[1]Department of Engineering and Exact, Federal University of Paraná, Palotina, Brazil*
*[2]Department of Computer Science, State University of Londrina, Londrina, Brazil*

*Article history*
Received: 5-04-2015
Revised: 2-06-2015
Accepted: 16-7-2015

Corresponding Author:
Marcos V. O. de Assis
Department of Engineering and Exact, Federal University of Paraná, Palotina, Brazil

Email: marcos.assis@ufpr.br

**Abstract:** Due to the increasing amount of data traveling computer networks every day, efficient management of this information is required to ensure the quality of the services provided by them. Development of new network management tools and mechanisms is a widely approached area due to its importance, not only to the current technology, but also to next generation network standards and equipments. Several researches have been directed to the use of IP Flows in order to increase the efficiency of these management tools. Although there are several proposed approaches in this area, most of them don't have suitable test scenarios to validate their performance results. In this study, we present Scorpius, a new simulation tool able to help testing network management mechanisms based on IP Flows. Scorpius is capable of simulating different kinds of anomalies, such as Denial of Service (DoS), Distributed Denial of Service (DDoS), Flash Crowd and Port Scan, directly into the flow export files. This characteristic unites the advantages of tests in real network environments without the drawbacks of the occurrence of real anomalies and attacks, even controlled ones. This approach makes the processes of performance analysis of anomaly detection approaches easier, without interfering or hampering the operation of the analyzed network. In order to validate the efficiency of the presented tool, we use real data collected from a large-scale network environment.

**Keywords:** Scorpius, Flows, Anomaly, Simulation, Network Management

## Introduction

One of the most relevant aspects about modern society is the importance of information. With the globalization process, most of the information traffic occurs into computational environments, large-scale networks characterized by their high-speed transmission and huge amount of data transportation. Furthermore, software and solutions typically executed locally, such as text processing tools, are migrating to the cloud environment as web solutions, creating a growing convergence of applications that need to be connected all the time (Ponnuramu and Tamilselvan, 2012; Prakash *et al*., 2014).

Thus, one of the most important and addressed topic in researches nowadays is the management and security of computer networks. It is an essential task to guarantee the quality of the services provided by computer networks, especially in large-scale environments such as large companies, hospital environments and metropolitan area networks (Huang *et al*., 2008; Tsagkaris *et al*., 2012). Attacks are becoming increasingly frequent every day, especially those that aim to hamper the availability of services, such as Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks (Liu, 2009; Hua *et al*., 2007), or to find security breaches in these services, such as Port Scan attacks (Korczynski *et al*., 2011). Thus, the development of effective anomaly detection tools is an extremely important matter. Even though it is a widely addressed area, it is still open due to the complexity of detection and identification processes, mainly in large-scale environments.

Over the years, several different approaches have been proposed to help the network management and the anomaly detection in computer network environments. Most of them are based on the Simple Network Management Protocol (SNMP) (Amaral *et al*., 2012; Zarpelão *et al*., 2007), mainly using counters to measure traffic volume patterns in order to detect the occurrence of attacks or anomalies, such as (Brutlag, 2000). Nowadays the SNMP protocol is still present in several network management solutions, but another approach is gaining momentum among the area researchers. This

approach is called IP Flow analysis, capable of describing not only volume aspects of the network traffic, but also qualitative information about the communication processes that compose it, such as IP addresses and ports of origin and destination and protocols. Many of the latest researches in the area are using IP flows on anomaly detection solutions, achieving promising results, such as (Jadidi *et al*., 2013), (Androulidakis and Papavassiliou, 2007), (Bartos *et al*., 2011), (Hong *et al*., 2008) and (de Assis *et al*., 2013).

Since 2000, our research group "Computer Networks and Data Communication" has been working on the development of anomaly detection solutions based on the analysis of Digital Signatures of Network's Segment (DSNS). The researches began with the usage of a SNMP based analysis, generating several important results, such as (Proença Jr. *et al*., 2004) and (Zarpelao *et al*. 2009). From 2011, our group started to use IP Flow based analysis through the exportation protocols IPFix, NetFlow and sFlow. The solid knowledge raised through SNMP pattern (DSNS) researches were applied to IP flow analysis. This new approach generated several contributions, which are based on Digital Signatures of Network's Segment using Flow analysis (DSNSF), such as (Fernandes *et al*., 2013), (Pena *et al*., 2014b; 2014a).

One of the most difficult steps towards the development of network management solutions and anomaly detection tools is the efficacy test of the proposed approach. These tools must be tested in all kinds of environments, in order to validate its efficiency as thoroughly as possible. Among the possibilities of existing tests environment, are the simulated, real and controlled network environments.

Aiming to unite the advantages of each one of these environments, as well as mitigate their disadvantages, this paper presents an alternative for network anomaly detection approaches testing: an IP Flow anomaly simulation tool called Scorpius (de Assis, 2014). This tool is capable of simulating different kinds of anomalies directly into flow export files, such as Denial of Service (DoS), Distributed Denial of Service (DDoS), Flash Crowd and Port Scan. This approach unites the efficiency of real network test scenarios with the precision of simulated networks and controlled attacks without interfering or hampering the network operation.

This tool was developed due to the testing requirement of several anomaly detection systems and approaches studied by our research group and was already used on several works, such as (de Assis *et al*., 2014), (Carvalho *et al*. 2014; 2013).

To validate the efficiency of the presented tool on simulating network anomalies into large-scale networks, we use real IP flows collected at State University of Londrina-Brazil, an environment composed of more than 7000 different hosts. To test Scorpius' precision, we compare a real anomaly that occurred on the collection environment with a simulated anomaly generated by the presented tool.

The remainder of this paper is composed of the following sections: Section 2 presents the related works, while section 3 describes the presented tool. Section 4 presents the obtained results of this software usage. Finally, section 5 shows the conclusions of this paper.

## *Related Work*

The usage of computational resources to help the solution of complex calculations and problems considerably improved scientific researches of all study areas. These areas usually need to test new theories, methods or approaches in order to validate the overall results of the study and verify whether the initial objective was achieved or not. However, some test environments are difficult to reproduce or to observe. The computational solution to this problem is usually the simulation, an approach where the desired test environment is artificially (virtually) reproduced, which is used by several different researchers. Sinreich and Marmor (2004), the authors highlights the importance of simulation processes on the reduction of costs and productivity improvement of hospital environments. They describe the basis of simulation tools in this environment, which must be intuitive, simple to use and flexible. Mempel *et al*. (2010), the authors propose the usage of transient simulation rather than traditional methods on testing modern protection equipments, presenting the advantages of this change through the analysis of different applications. Canova *et al*. (1999), the authors use simulation processes on the development of a weapon called Predator SRAW, or short range attack weapon. The gains of the simulation usage are visible not only on the cost reduction of the project, but also on the overall quality of the product. Ab-Rahman (2011), the authors test different algorithms for a Fiber-to-the-Home Passive Optical Network Automatic self-restoration scheme through the use of access control system using simulation approaches.

As observed, several different application fields use simulation in order to solve specific problems. This situation is no different in network management and anomaly detection systems, a widely addressed study area with great importance, specially due to the constant increase of traffic in large-scale networks. Kuhl *et al*. (2007), the authors present a simulation modeling approach to represent computer networks and Intrusion Detection Systems (IDS) in order to test the security of these environments. Bhatia *et al*. (2009), the authors perform a simulation to identify the occurrence of zero day silent worm in Local Networks (LAN) or intranets. They highlight the fact that, compared to the time and cost involved in setting up an entire test bed into real networks, the use of simulations are considered fast and inexpensive solutions. Puketza *et al*. (1996), the authors use a methodology based on the simulation of users (both the intruders and normal ones) in order to test Intrusion Detection Systems (IDS).

In order to improve the performance evaluation tests of network anomaly detection approaches, the simulation tool used on the process should provide different anomalous scenarios. These scenarios must be capable of testing the proposed approach on different kinds of situation, such as DoS, DDoS, Port Scans attacks or Flash Crowd events, through the use of modern and current attack techniques and methodologies. Liu (2009), the author presents a research on Denial of Service attacks and detection programming, highlighting the fact that this is the most popular attack in the network security area. Additionally, the author describes some Distributed Denial of Service approaches, techniques based on the conventional DoS methodologies. If the DoS attack is the most popular, Distributed Denial of Service attacks are the most serious security problems in computer networks nowadays, as said by (Hua *et al*., 2007). In their research, the authors state that this kind of attack is particularly difficult to detect due to its distributed nature, where the attackers behavior usually are similar to normal users behavior. This matter also represents the central discussion of the research presented in (Li *et al*., 2009), where the authors propose the usage of probability metrics in order to distinguish DDoS attacks from Flash Crowd events, anomalies characterized by the parallel usage of a network resource by a huge amount of legitimate users. Korczynski *et al*. (2011), the authors highlight that Port-scan attacks are usually a precursor of intrusion attempts and it is an activity of difficult identification due to its low volume of generated data.

Another important aspect on developing a simulation tool for network anomaly detection approaches is the support of network flow analysis. The use of IP flows is a powerful solution not only for the detection of the previously mentioned attacks and anomalies, but also for the identification of the origin of the problem. Unlike the SNMP technology, the use of IP flows provides several qualitative information of the collected communication processes. Several different network anomaly detection approaches are using flow analysis in recent years, such as (Jadidi *et al*., 2013), (Androulidakis and Papavassiliou, 2007), (Bartos *et al*., 2011) and (Hong *et al*., 2008).

Thus, it is important for network anomaly detection researchers to be able to test their proposed approaches in complete test scenarios. The most complete environment available for testing is always the environment in which the solution is proposed to be used, i.e., real operating computer networks. However, anomalies and attacks, even the controlled ones, can impact on the operation of these networks, hampering the experience of its users. Thus, the most effective solution is the usage of a simulation tool capable of injecting the behavior of attacks and anomalies directly into exported network files. This paper presents Scorpius (de Assis, 2014), a JAVA application capable of injecting the behavior of DoS, DDoS, Flash Crowd and Port scan anomalies directly into flow records exported from real networks through the sFlow protocol. This tool helps testing processes of new anomaly detection methods with the advantages of using real network data flows without interfering on its normal operation.

## Presented Simulation Tool

One of the main difficulties on the development of new techniques, models and anomaly detection systems in computer networks is the testing. These tools must be tested in several different environments in order to evaluate its efficiency in a way as completely as possible. Among the existing test possibilities, there are:

Simulated networks: Test environments completely controlled, where it is possible to control every single network variable in order to reproduce the required behavior. Although this method is extremely complete, the assembly complexity of a simulated network is relatively high.

Real networks: Test environments where real data collected from operating networks (production environment) are used, enabling the measurement of the efficiency of the tested tool directly at the environment it is intended to be applied. The real network testing is extremely important to anomaly detection systems evaluation, once they illustrate exactly the scenarios where these systems operate. However, its biggest advantage brings on a disadvantage, because the completeness level of the tests is directly related to the diversity of the anomalies that occur on this network. Thus, if Flash Crowd anomalies are the only one present on the used network, the system will not be tested to other types of anomalies, such as DoS, DDoS or Port Scans, for example.

Controlled networks: Test environments where different anomalies are directed and performed into a specific computer network in an intentional way. In these types of environments, particular hours of the day are usually selected in order to avoid hampering the network's normal operation. Even though this approach is less complex than the development of simulated networks, these anomalies/attacks, even if controlled, can impair the network's normal operation, especially in large scale networks, which are used 24 h a day. Aiming to ease the performance evaluation tests of anomaly detection and network management systems, an IP Flow anomaly simulation tool was developed, uniting the main advantages of the different approaches previously presented. This tool is called Scorpius-sFlow Anomaly Simulator and significantly simplifies the performance study and evaluation of anomaly detection systems based on IP flows analysis. The current version of Scorpius is a freeware and it is available for the community access through the tool's website website: "http://redes.dc.uel.br/scorpius/" (de Assis, 2014).

**DOI: 10.3844/jcssp.2015.662.664**

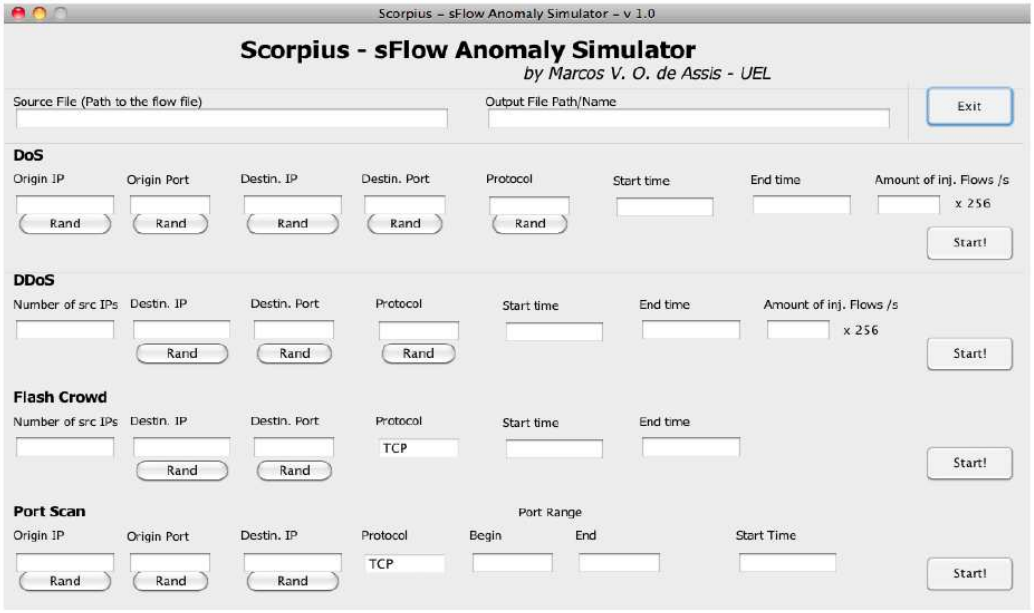


Fig. 1. Scorpius' Interface

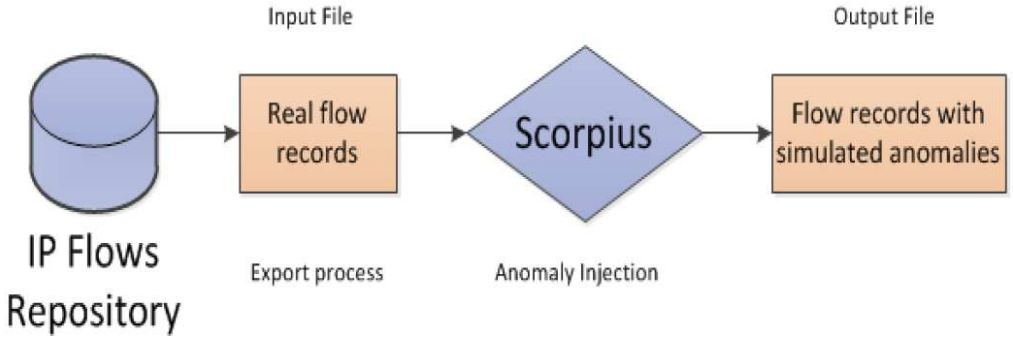


Fig. 2. Diagram of the Scorpius' operation

*Scorpius-sFlow Anomaly Simulator*

Scorpius can basically inject new flow records into an IP flow exportation file, simulating the behavior of specific anomalies into specific time intervals. Its current version supports the DoS, DDoS, Flash Crowd and Port scan anomalies or attacks. The Fig. 1 presents the Scorpius tool in its current version.

This is an application developed using the JAVA programming language, through the use of the production environment Eclipse (Helios version). The Scorpius tool unites the control of simulated network environments, the wealth of information of real networks and the flexibility and diversity of anomalies of controlled environments with the safety of a quick and simplified simulation. Its operations can be observed in the diagram presented by Fig. 2.

As can be verified, Scorpius uses a real data flow file as input, injects anomalous flows with the required characteristics at a specific time interval and generates a new IP flows file as output with the solicited anomalies already incorporated. This file can be used for testing in an identical way as real network environment, once its data are in fact real with the exception of the simulated anomalies. These anomalies are injected as additional behaviors to the network normal behavior at the solicited time interval, not excluding or changing any real flow records. Thus, different kinds of anomalies can be tested in real environments, even if these anomalies did not inflict the utilized network during the time period used for testing. Furthermore, the operation of the network used on the performance evaluation tests of network anomaly detection approaches and tools does not suffer from any influences through this entire process. Figure 3 presents an excerpt from an IP flow file (exported from the sFlow protocol), where the same portion of file can be analyzed before and after the injection (simulation) of an anomaly of the DoS type, beginning at 10 o'clock of October 1st, 2012.

**DOI: 10.3844/jcssp.2015.662.664**

**Before:**

```
2012-10-01 09:59:58.500     0.000 UDP      108.35.22.143:50039 ->     189.90.71.56:55893      256    17920     1
2012-10-01 10:00:01.608     0.000 TCP      74.125.234.162:80   ->   189.90.69.164:61529      256   367104     1
2012-10-01 10:00:01.608     0.000 TCP      209.207.230.25:80   ->     189.90.65.11:4479      256    71680     1
2012-10-01 10:00:01.608     0.000 TCP        201.54.66.11:80   ->    189.90.65.11:32247      256    17408     1
2012-10-01 10:00:01.608     0.000 TCP    177.177.181.110:27566 ->    189.90.74.46:64576      256    17408     1
2012-10-01 10:00:01.608     0.000 TCP      74.125.134.108:465  ->   189.90.72.124:49614      256    18944     1
2012-10-01 10:00:05.607     0.000 TCP        201.54.66.10:80   ->    189.90.65.11:30355      256   389632     1
2012-10-01 10:00:05.607     0.000 TCP        189.14.52.15:80   ->    189.90.65.11:26071      256   389632     1
2012-10-01 10:00:05.607     0.000 UDP      71.45.153.156:6881  ->     189.90.74.46:22271      256    17920     1
```

**After:**

```
2012-10-01 09:59:58.500     0.000 UDP      108.35.22.143:50039 ->     189.90.71.56:55893      256    17920     1
2012-10-01 10:00:01.608     0.000 TCP      74.125.234.162:80   ->   189.90.69.164:61529      256   367104     1
2012-10-01 10:00:01.608     0.000 UDP      11.11.11.11:80      ->     22.22.22.22:80         256     2048     1
2012-10-01 10:00:01.608     0.000 TCP      209.207.230.25:80   ->     189.90.65.11:4479      256    71680     1
2012-10-01 10:00:01.608     0.000 TCP        201.54.66.11:80   ->    189.90.65.11:32247      256    17408     1
2012-10-01 10:00:01.608     0.000 TCP    177.177.181.110:27566 ->    189.90.74.46:64576      256    17408     1
2012-10-01 10:00:01.608     0.000 TCP      74.125.134.108:465  ->   189.90.72.124:49614      256    18944     1
2012-10-01 10:00:05.607     0.000 TCP        201.54.66.10:80   ->    189.90.65.11:30355      256   389632     1
2012-10-01 10:00:05.607     0.000 UDP      11.11.11.11:80      ->     22.22.22.22:80         256     2048     1
```

Fig. 3. Snippet from IP flows file: before and after a simulated DoS attack

It is important to highlight that this is the first version of the tool, which has support only for IP flows of the default exportation format of the sFlow protocol.

### *Scorpius Operation: User View*

The operation of Scorpius, from the user point of view, is simplified: First of all, the user fills the field "Source File" with the complete path of the IP flows file used as basis to the anomaly injection. Then, the field "Output File Path/Name" is also filled with the path and name of the resulting file (if the file path is not specified, Scorpius will generate the file in the same directory where it is located).

Afterwards, an anomaly is chosen by the user, which fills the fields relating to them and then click the "Start!" button to start the process. Some of these fields have a random data generation button (Rand), which simplifies the analysis process in situations where only the anomaly detection is important, not necessarily the identification of its origin. In the case of "Origin IP" and "Destin. IP" (IP Addresses of origin and destination, respectively) fields, the generation of invalid IP addresses in real environments is possible, which does not influence the performance evaluation process of anomaly detection systems.

The Time fields (Hours) must be filled with the HH:MM:SS (hours:minutes:seconds) pattern and the field "Start Time" must be previous to the "End Time".

Finally, the field "Amount of inj. Flows /s" (Number of Injected Flows per second) must be filled with the quantity of flows injected in each second on the file within the specified time interval. It is important to highlight that the sFlow protocol has native support to sampling and, thus, each flow injected per second represents other 256 flows collected by the switch. For these reasons, this value is commonly defined as 1.

### *Scorpius Operation: Low-Level View*

For the low level point of view, each supported anomaly has different behaviors, which were implemented in Scorpius as is following described:

- Denial of Service (DoS): The denial of service attacks is, basically, an attack of a single host to a specific destination host that aims at the congestion of a service or server. Thus, after filling the data and the selection of the "Start!" button by the user, Scorpius injects the selected number of flows present in the "Amount of inj. Flows /s" field to each second from the previously stipulated start and end time. These flows contain the information of IP addresses of origin and destination, ports of origin and destination and protocol defined by the user. Although the protocol used is of free choice (TCP, UDP or ICMP), it is recommended the usage of the UDP protocol, since it is the most commonly addressed approach in this kind of attack, followed by the ICMP. The flow attribute Bytes is calculated accordingly to the minimum size of packet sending from each one of the different protocols
- Distributed Denial of Service (DDoS): In a similar way with the DoS, distributed denial of service attacks are intended to congest a service or server. However, unlike the DoS, this attack uses several different hosts that, intentionally or not (as in the case of BotNets or zombie networks, composed by infected computers remotely controlled by hackers), attacks a single host. Its operation on Scorpius is similar to the DoS attack, with the difference that there is no need to specify the IP addresses of origin for the attack, only how many different hosts are used for it. The IP addresses of these hosts are generated randomly by Scorpius, which generates a flow record

to each origin host, increasing the impact caused on the network in comparison to the DoS attack

- Flash Crowd: These events are not considered attacks, being classified as anomalies. They are events in which a huge amount of legitimate users access a server or service simultaneously, causing and involuntary congestion similar to the previously mentioned attacks. These events occur more frequently in websites or WEB applications, such as the electronic address of universities disclosing the results of selective processes. Thus, the most commonly used protocol in these cases is the TCP and for this reason, this protocol is predefined in this kind of anomaly injection. As the IP addresses of origin are varied, each second new IP addresses of origin are randomly generated by Scorpius and each one of these addresses of origin generate a flow record per second
- Port Scan: This anomaly is basically a procedure used to verify the vulnerabilities of a specific host. Thus, an origin host performs a scanning process in all or in a range of ports of a destination host, verifying which one are open. Several methodologies can be used in this process, most of them using the TCP protocol (defined as standard in this kind of anomaly). The Scorpius tool implements the method called Half-Open, widely used due to its characteristic of not allowing the TCP connection to be established, avoiding the attacker to be identified. This method basically sends a SYN message to a specific port at the destination host and, if it receives a SYN+ACK response, which indicates that the port is currently open, it responds with a RST message to finalize the connection before it is established. Thus, only 3 packets are exchanged for each port scanned

In Scorpius, the user must fill the fields IP addresses of origin and destination, port of origin and start time for the attack. It does not have an end time because it is finalized when all the specified ports are scanned. Also, the user must fill the fields “Start” and “End” relating to the Port Range macro field, which may vary from 1 to 65535. Each second, a new flow relating to a port of the destination host is generated on the file and the next flow injected will be related to the 256th following port, i.e., the first flow injected is relating to the scanning of the port 1, the next injected flow is relating to the scanning of the port 256, then 512 and so on. Thus, if 1024 ports are scanned, only 5 flows are generated, once each analyzed port was selected at the sampling process of the sFlow protocol with rates of 1:256.

The attacks and anomalies here described directly influence the behavior of the affected network. Through the IP flow analysis, it is possible to verify these effects through different features or dimensions. Table 1 illustrates how each one of the attacks/anomalies supported by Scorpius affects traffic behavior.

In Table 1, orgIP and desIP represents the features IP addresses of origin and destination, respectively, as well as orgPort and desPort represents the features Ports of origin and destination, respectively. These features, unlike bits, packets and flows per second, are qualitative information. In order to enable the quantitative analysis of these features, we used the Shannon Entropy (Shannon, 2001) to represent them. This metric have lower outcomes when the concentration of the entries are high and higher outcomes when the dispersion of the entries are elevated. In Table 1, the symbol "++" represents an increase on the behavioral movement, "--" indicates its decrease, "+-" represents that the behavior has been affected by a behavioral increase or decrease and the "N/A" points out that the feature has not being affected by the anomaly.

### *Scorpius Comparison with other Tools*

One of the most used network attack tools in the last years is named LOIC, an acronym for "Low Orbit Ion Cannon" (Praetox Technologies, 2010). This tool was developed in order to provide a simple and easy way to perform a Denial of Service (DoS) attack, where the user can change and personalize several attributes and "flood" the destination IP with the selected packets. Although it is a powerful approach, this tool is limited: It only performs DoS attacks. In order to perform a Distributed DoS attack (DDoS), several different machines simultaneously executing LOIC are necessary, which is an effective approach for hacker community groups (usually composed by a large set of people), but not for the average research teams. Furthermore, this is a real network attack tool, which means that LOIC's usage on performance evaluation of network anomaly detection approaches is based on a controlled network environment. As previously mentioned, even controlled attacks can impair the network's normal operation, especially in large scale networks.

Unlike the LOIC tool, Scorpius can simulate DoS, DDoS, Port Scan and Flash Crowd attacks/anomalies, injecting their behavior directly into flow data in a secure way.

Another widely addressed approach to evaluate the performance and efficiency of network anomaly detection approaches is the simulation process. Xiao *et al*. (2009), the authors propose an anomaly detection scheme based on machine learning for wireless sensor networks, using the Network Simulator 2 (NS2) tool to evaluate their approach. NS2 is a powerful tool able to simulate different network environments and configurations, where it is possible to configure the behavior of both network assets and hosts. Although network simulations can synthetically generate traffic and emulate specific nodes, the limited size of the experiments and the difficulties on background traffic generation, as well as the difficulty of modeling real-like network environments, are drawbacks that may hamper the performance evaluation process of anomaly detection approaches.

Table 1. Attacks/Anomalies signatures

| | Flash Crowd | DoS | DdoS | Port-Scan |
|---|---|---|---|---|
| bits/s | ++ | N/A | N/A | N/A |
| packets/s | ++ | ++ | ++ | ++ |
| flows/s | ++ | ++ | ++ | ++ |
| orgIP | N/A | -- | N/A | -- |
| desIP | -- | -- | -- | -- |
| orgPort | +- | -- | -- | -- |
| desPort | -- | -- | -- | ++ |

Scorpius unites the power of simulation with the reliability of real network environment data, injecting anomalous behavior directly into real IP flows. This is a simple, effective and secure approach for anomaly detection performance measurement.

## Obtained Results

To measure the efficiency of the presented tool on the simulation of anomalies, we used real IP flows exported by the sFlow protocol, collected at the State University of Londrina-Brazil. During the collection process, we identified two real anomaly occurrences. These anomalies are Flash Crowd events and were used to test the precision of the simulations generated by Scorpius.

First of all, Scorpius need to use as base data collected in a normal day, i.e., a day where no anomalies occurred. Then, this normal day behavior is injected with an anomaly, simulating the parallel execution of both behaviors. In other words, the input data of Scorpius are the IP flow data which describes the behavior of a normal day, on which a simulation of a specific anomaly is performed. Figure 4 presents the graphs of seven flow dimensions (or features) of the day used as Scorpius input on the tests performed, collected in 9th October, 2012.

Through the exported sFlow file which describes the base day, Scorpius injects new IP flows into this file, simulating an additional activity that occurs in parallel to the normal one. To measure the efficiency of the presented tool, we analyzed the days 5th and 30th October, 2012, in which we identified the occurrence of anomalous behaviors relating to Flash Crowd events.

Figure 5 presents the real behavior collected on 5th October, 2012 (green), in which Flash Crowd anomalies were identified in three different hours: from 10:00 to 10:20, from 12:00 to 12:30 and from 14:35 to 18:00 hours. The blue line represents the behavior of the Scorpius output, which is composed by the base day (input of the tool) and the injected Flash Crowd anomaly on the mentioned time intervals. The simulations were performed using approximately 1280 hosts (IP addresses) of origin accessing a single IP address of destination on the analyzed network.

Table 2. Correlation Coefficients and NMSE results of the comparison between Scorpius' anomaly simulation and a real anomalous day (October 5th, 2012)

| | Correlation Coefficient | NMSE |
|---|---|---|
| bits/s | 0,94 | 0,05 |
| packets/s | 0,91 | 0,04 |
| flows/s | 0,91 | 0,04 |
| IP Origin | 0,88 | 0,01 |
| IP Destin. | 0,41 | 0,03 |
| Port Origin | 0,65 | 0,04 |
| Port Destin. | 0,85 | 0,01 |

Table 3. Correlation Coefficients and NMSE results of the comparison between Scorpius' anomaly simulation and a real anomalous day (October 30th, 2012)

| | Correlation Coefficient | NMSE |
|---|---|---|
| bits/s | 0,91 | 0,06 |
| packets/s | 0,87 | 0,06 |
| flows/s | 0,87 | 0,06 |
| IP Origin | 0,72 | 0,02 |
| IP Destin. | 0,42 | 0,03 |
| Port Origin | 0,47 | 0,10 |
| Port Destin. | 0,74 | 0,02 |

Table 4. Blend-Altman plot percentage error of the comparison between Scorpius' anomaly simulation and a real anomalous days

| | October 5th, 2012 (%) | October 30th, 2012 (%) |
|---|---|---|
| bits/s | 5.3 | 4.8 |
| packets/s | 4.7 | 6.8 |
| flows/s | 4.7 | 6.8 |
| IP Origin | 3.7 | 5.3 |
| IP Destin. | 6.4 | 4.8 |
| Port Origin | 4.6 | 5.9 |
| Port Destin. | 4.5 | 6.8 |

Figure 6 presents the real behavior collected on 30th October, 2012 (green), which presents Flash Crowd anomalies identified at two different time intervals: From 10:50 to 11:40 and from 20:00 to 22:00 h. The blue line represents the Scorpius output which, as previously mentioned, is composed by the union of the base day and the Flash Crowd anomalies injected by the tool in these time intervals. The simulation was performed with around 1790 to 2560 hosts accessing a single IP address of destination on the network, for the first and second anomalous intervals identified in this day, respectively.

In order to quantitatively measure this efficiency, we used three different metrics. The first is the Correlation Coefficient (CC), which measures the concordance degree between the real movement and the simulation. The second is the Normalized Mean Square Error (NMSE), which measures the error between them, exposing the most striking differences. As mentioned in (Sarin *et al*., 2010), correlation and error metrics are the most used approach on testing simulation outcomes.

**DOI: 10.3844/jcssp.2015.662.664**

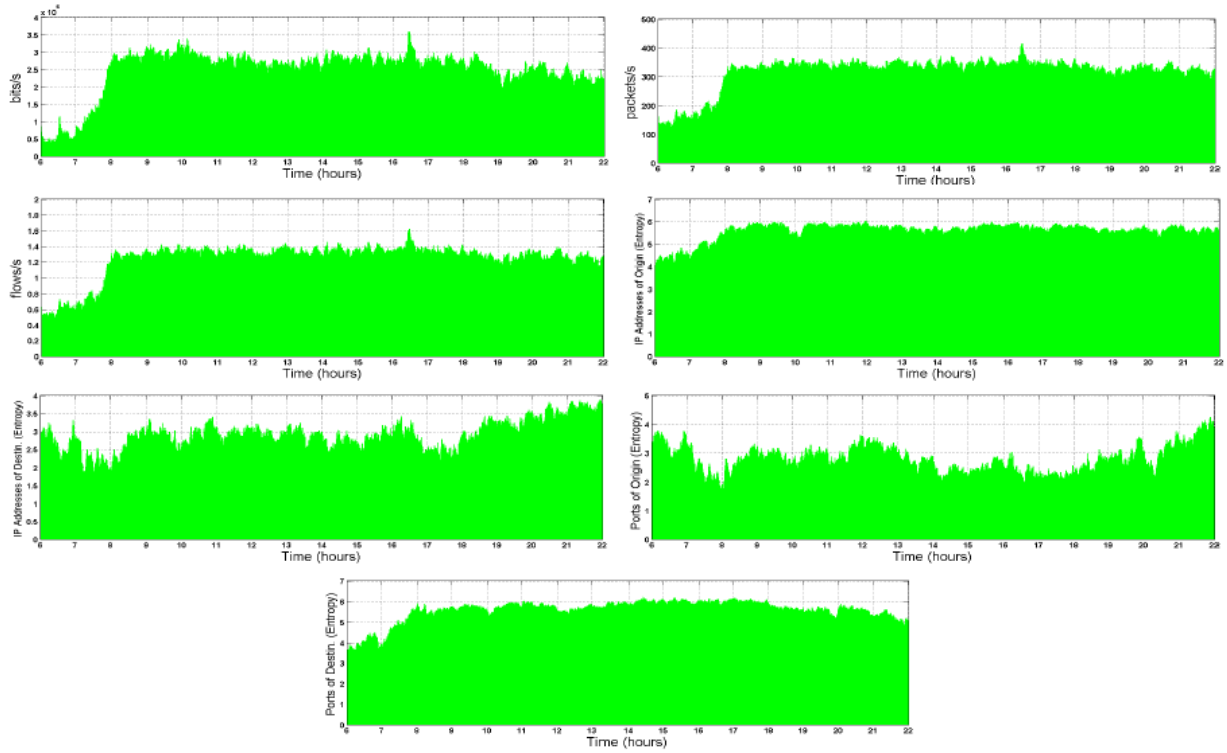


Fig. 4. Base day used by Scorpius (09/10/2012)

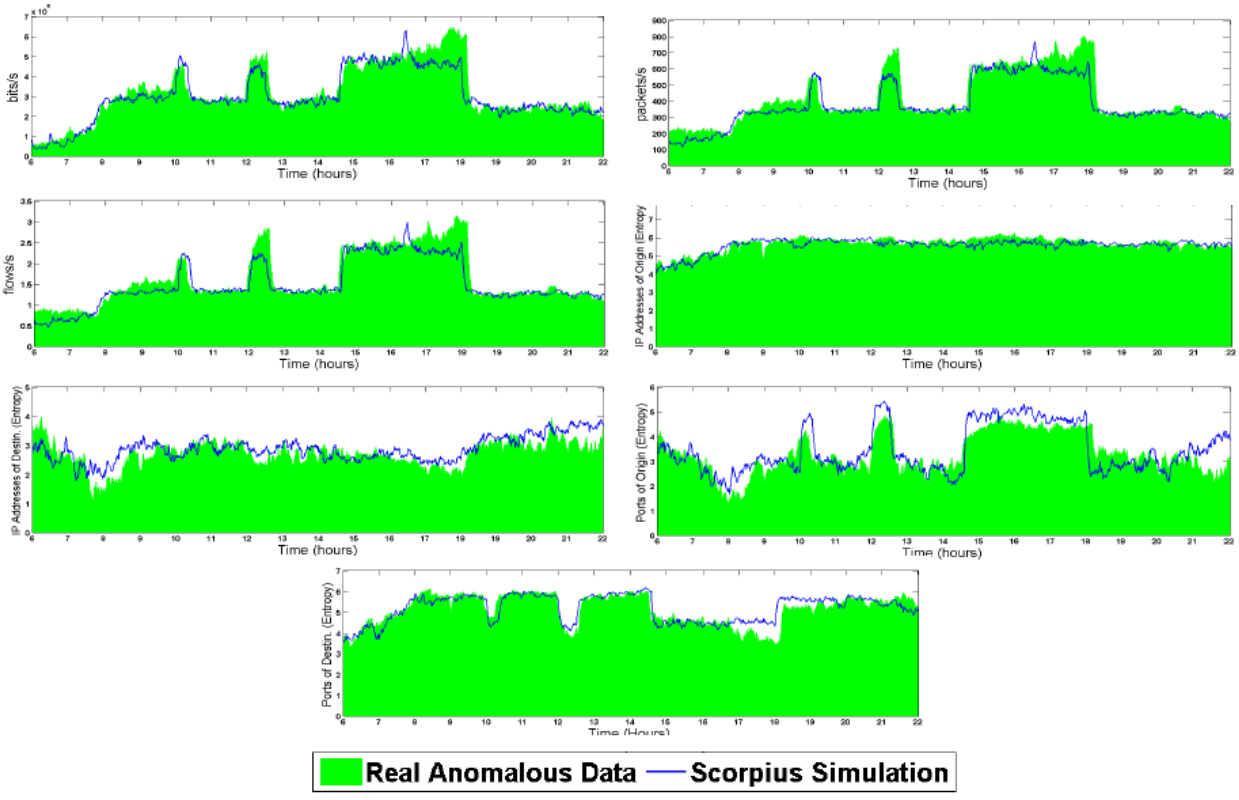


Fig. 5. Anomaly simulation performed by Scorpius compared to a real anomalous day (October 5$^{th}$, 2012)

**DOI: 10.3844/jcssp.2015.662.664**

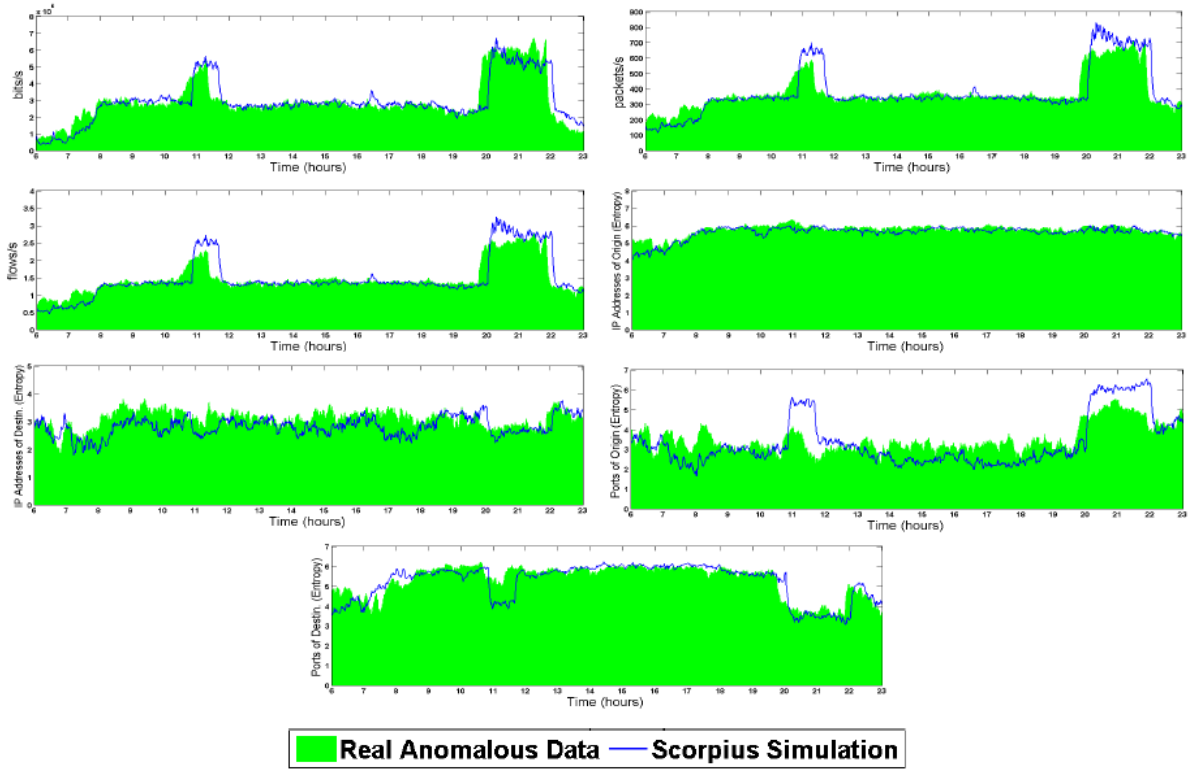


Fig. 6. Anomaly simulation performed by Scorpius compared to a real anomalous day (October 30th, 2012)

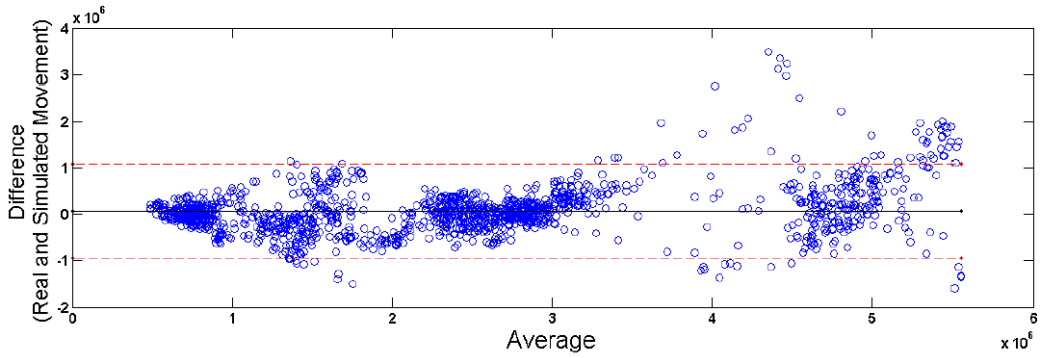


Fig. 7. Bland-Altman plot between real anomalous day and Scorpius' simulation for October 5th, 2012

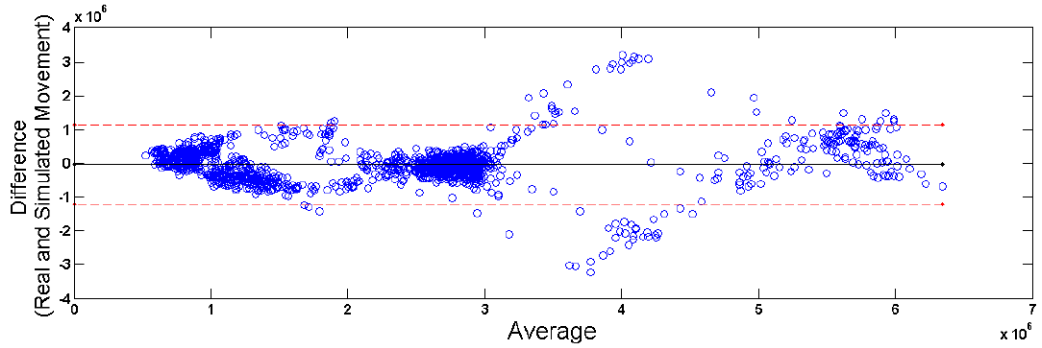


Fig. 8. Bland-Altman plot between real anomalous day and Scorpius' simulation for October 30th, 2012

Finally, the third technique is the Blend-Altman plot (Altman and Bland, 1983), a graphic metric capable of analyzing the concordance between two data sets.

Through the CC test, the result 1 is the optimal outcome and 0 is the worst possible scenario. For the NMSE test, 0 represents the optimal results (data tested are exactly the same). The obtained results through the use of these technique can be observed on Table 2 and 3.

Visually, it is possible to conclude that the simulation performed by Scorpius is extremely efficient. The numerical results presented on Table 2 and 3 highlight several important aspects. First of all, the results of the features IP addresses of destination and Ports of origin fared worse than the other features mainly due to its behavioral characteristic. Those features are less stable than the others, which directly influenced the CC and NMSE results. Furthermore, as can be observed, Fig. 4-6 shows 7 flow features which describe the same traffic. The features IP Addresses and Ports of origin and destination are qualitative dimensions. In order to quantify them, we used the Shannon Entropy (Shannon, 2001) approach, where high values represent a higher diversity of entries (IP addresses or ports) and low values represent a higher concentration of entries, metric calculated in one minute intervals.

Relating to October 5th, 2012, the results presented by Fig. 5 and Table 2 show the great efficiency of Scorpius on simulating the Flash Crowd anomaly. The flow features bits, packets and flows per second were affected by the increase of the traffic movement, due to the huge amount of information generated by the anomaly. The feature IP addresses of origin remains the same, because the hosts responsible for the Flash Crowd event are legitimate users, not interfering in the concentration analysis. The feature IP addresses of destination suffered a slight decrease, due to the fact that several hosts were accessing a single destination IP. The movement of the feature Ports of origin increased during the anomaly because different ports were used to access the destination host. Similarly, the feature Ports of destination presented movement decrease during the anomaly due to the fact that the hosts were accessing the destination host through a single port, increasing the concentration of the flow entries.

Scorpius anomaly simulation successfully described the same behaviors on the analyzed flow features. The CC test highlight a high concordance between the simulation and the real anomalous day, except on the features IP addresses of destination and Ports of origin, which are unstable, as previously mentioned. The NMSE test shows that the difference between the simulation and the real anomalous day is low, close to the optimal metric outcome.

For October 30th, 2012, results, presented by Fig. 6 and Table 3, the behavior of the analyzed anomaly is the same as previously mentioned for October 5$^{th}$, 2012, since both days presented Flash Crowd anomalies. The difference between these days is basically the intensity of the anomalies, which increased for the second test day. Another difference is that for the second test day, the users accessed the destination host through different Ports, decreasing the destination Port entropy impact on the first anomaly occurrence (form 10:50 to 11:40). Furthermore, the Ports of origin used by the users were more recurrent, decreasing the origin Port entropy impact on this same time interval.

Scorpius anomaly simulation fared worse in this test than the previously addressed, mainly because, in this first version of the tool, it does not support many intensity customization features. However, Scorpius successfully described the behavior of the analyzed flow dimensions, achieving CC and NMSE outcomes similar to the previously mentioned tests.

Finally, the results achieved by Scorpius in the Blend-Altman plot can be observed in Fig. 7 and 8 for the flow feature “bits per second”. The solid lines in the center of the graphics represent the average and the dashed lines are thresholds, which are calculated by adding and subtracting (upper and lower limits) the average to the standard deviation times 1.96. The outcome of this technique points out that the analyzed data sets are in concordance if most of the points are located between these thresholds. As observed, most of the points are located between the mentioned thresholds both in Fig. 7 and 8. To translate the visual outcome of the Blend-Altman plot to a quantitative metric, we counted the number of points that are out of the technique's thresholds, measuring the error percentage for each analyzed day and flow feature. The obtained result can be observed on Table 4.

As observed, the percentage errors are always under 7% rate. Furthermore, the percentage error is 5.3% on the average, which highlight the efficiency of the presented simulation tool.

## Conclusion

In this study, we presented a network anomaly simulation tool based on IP Flows, named Scorpius. The tool was described in detail and its operation was analyzed through system and user points of view. Scorpius supports 4 different anomalies, which was also described in order to illustrate how Scorpius performs the simulation process of each one of them.

Scorpius was submitted to quantitative performance metrics to measure its efficiency and simulation precision, achieving high Correlation Coefficient results and low Normalized Mean Square Error outcomes for most of the tested flow features through the analyzed anomalous days. Scorpius was able to successfully describe the behavior of the tested anomaly even without many intensity customization features.

The obtained results point out that Scorpius achieved its main objective: To simulate the behavior of

anomalies in computer networks through IP flows, using as base days with normal behavior, without interfering on the operation of the analyzed network. We can conclude that the presented tool is capable of simulating anomalies efficiently and directly into IP flows, eliminating the need for complex network simulation configuration, controlled attacks or the occurrence of real attacks for complete testing of new anomaly detection systems. Thus, Scorpius represents an important auxiliary tool on the development of new network management and security approaches.

In future works, we intend to analyze the behavior of other anomalies, in order to improve the performance of the presented tool in different scenarios. Furthermore, we intend to implement more intensity customization features in order to improve the accuracy of the simulation process.

## Acknowledgement

This work was supported by Federal University of Paraná, through the Banpesq/2014016797 Project.

## Funding Information

This work was supported by SETI/Fundação Araucária for Betelgeuse/41.939/21/2012 Project's financial support;

## Author's Contributions

**Marcos V. O. de Assis:** Corresponding author of this paper, participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

**Mario Lemes Proença Jr.:** Provided access to the dataset used on testing, coordinated the data-analysis and contributed to the writing of the manuscript.

## References

Ab-Rahman, S.M. and S.R.A. Mahir, 2011. Development of an algorithm for fiber-to-the-home passive optical network automatic self-restoration scheme using access control system. J. Comput. Sci., 7: 1846-1853. DOI: 10.3844/jcssp.2011.1846.1853

Altman, D.G. and J.M. Bland, 1983. Measurement in medicine: The analysis of method comparison studies. J. Royal Statistical Society, 32: 307-317. DOI: 10.2307/2987937

Amaral, A.A., B.B. Zarpelão, L.S. Mendes, J.J.P.C. Rodrigues and M.L. Proença Jr., 2012. Inference of network anomaly propagation using spatio-temporal correlation. J. Netw. Comput. Appli., 35: 1781-1792. DOI: 10.1016/j.jnca.2012.07.003

Androulidakis, G. and S. Papavassiliou, 2007. Intelligent flow-based sampling for effective network anomaly detection. Proceedings of the IEEE Global Telecommunications Conference, Nov. 26-30, IEEE Xplore Press, Washington, pp: 1948-1953. DOI: 10.1109/GLOCOM.2007.374

Bartos, K., M. Rehak and V. Krmicek, 2011. Optimizing flow sampling for network anomaly detection. Proceedings of the 7th International Wireless Communications and Mobile Computing Conference, Jul. 4-8, IEEE Xplore Press, Istanbul, pp: 1304-1309. DOI: 10.1109/IWCMC.2011.5982728

Bhatia, A., P.S. Dhabe and S.G. Pukale, 2009. Java based simulator to detect zero-day silent worms using ACTM. Proceedings of the IEEE International Advance Computing Conference, Mar. 6-7, IEEE Xplore Press, Patiala, pp: 847-852. DOI: 10.1109/IADCC.2009.4809125

Brutlag, J.D., 2000. Aberrant behavior detection in time series for network monitoring. Proceedings of the 14th Systems Administration Conference, (Dec. 3-8), USENIX Association, USA, pp: 139-146.

Canova, B.S., P.H. Christensen, M.D. Lee, B.R. Tripp and M.H. Pack *et al*., 1999. Simulation to support operational testing: A practical application. Proceedings of the Winter Simulation Conference, Dec. 5-8, IEEE Xplore Press, Phoenix, pp: 1071-107 8. DOI: 10.1109/WSC.1999.816823

Carvalho, L.F., G. Fernandes Jr., M.V.O. de Assis, J.J.P.C. Rodrigues and M.L. Proença Jr., 2014. Digital signature of network segment for healthcare environments support. IRBM, 35: 299-309. DOI: 10.1016/j.irbm.2014.09.001

Carvalho, L.F., J.J.P.C. Rodrigues, S. Barbon and M.L. Proenca, 2013. Using ant colony optimization metaheuristic and dynamic time warping for anomaly detection. Proceedings of the 21st International Conference on Software, Telecommunications and Computer Networks, Sep. 18-20, IEEE Xplore Press, Primosten, pp: 1-5. DOI: 10.1109/SoftCOM.2013.6671906

de Assis, M.V.O., 2014. *Scorpius-sFlow Anomaly Simulator* (version 1.0). JAVA. Brazil.

de Assis, M.V.O., J.J.P.C. Rodrigues and M.L. Proença Jr., 2013. A novel anomaly detection system based on seven-dimensional flow analysis. Proceedings of the IEEE Global Communications Conference, Dec. 9-13, IEEE Xplore Press, Atlanta, pp: 735-740. DOI: 10.1109/GLOCOM.2013.6831160

de Assis, M.V.O., J.J.P.C. Rodrigues and M.L. Proença Jr., 2014. A seven-dimensional flow analysis to help autonomous network management. Inform. Sci., 278: 900-913. DOI: 10.1016/j.ins.2014.03.102

Fernandes, G., A.M. Zacaron, J.J.P.C. Rodrigues and M.L. Proenca, 2013. Digital signature to help network management using principal component analysis and k-means clustering. Proceedings of the IEEE International Conference on Communications, Jun. 9-13, IEEE Xplore Press, Budapest, pp: 2519-2523. DOI: 10.1109/ICC.2013.6654912

Hong, W., G. Zhenghu, G. Qing and W. Baosheng, 2008. Detection network anomalies based on packet and flow analysis. Proceedings of the 7th International Conference on Networking, Apr. 13-18, IEEE Xplore Press, Cancun, pp: 497-502. DOI: 10.1109/ICN.2008.83

Hua, L., H. Guang-min and Y. Xing-miao, 2007. Global detection of DDoS attack based on time and frequency analysis. Proceedings of the International Conference on Communications, Circuits and Systems, Jul. 11-13, IEEE Xploer Press, Kokura, pp: 462-466. DOI: 10.1109/ICCCAS.2007.6251606

Huang, X., S. Ganapathy and T. Wolf, 2008. A framework for network state management in the next-generation internet architecture. Proceedings of the IEEE Global Telecommunications Conference, Nov. 30-Dec. 4, IEEE Xplore Press, New Orleans, pp: 1-5. DOI: 10.1109/GLOCOM.2008.ECP.435

Jadidi, Z., V. Muthukkumarasamy, E. Sithirasenan and M. Sheikhan, 2013. Flow-based anomaly detection using neural network optimized with gsa algorithm. Proceedings of the IEEE 33rd International Conference on Distributed Computing Systems Workshops, Jul. 8-11, IEEE Xplore Press, Philadelphia, pp: 76-81. DOI: 10.1109/ICDCSW.2013.40

Korczynski, M., L. Janowski and A. Duda, 2011. An accurate sampling scheme for detecting SYN flooding attacks and portscans. Proceedings of the IEEE International Conference on Communications, Jun. 5-9, IEEE Xplore Press, Kyoto, pp: 1-5. DOI: 10.1109/icc.2011.5962593

Kuhl, M.E., J. Kistner, K. Costantini and M. Sudit, 2007. Cyber attack modeling and simulation for network security analysis. Proceedings of the Winter Simulation Conference, Dec. 9-12, IEEE Xplore Press, Washington, pp: 1180-1188. DOI: 10.1109/WSC.2007.4419720

Li, K., W. Zhou, P. Li, J. Hai and J. Liu, 2009. Distinguishing DDoS attacks from flash crowds using probability metrics." Proceedings of the 3rd International Conference on Network and System Security, Oct. 19-21, IEEE Xplore Press, Gold Coast, pp: 9-17. DOI: 10.1109/NSS.2009.35

Liu, W., 2009. Research on DoS attack and detection programming. Proceedings of the 3rd International Symposium on Intelligent Information Technology Application, Nov. 21-22, IEEE Xplore Press, Nanchang, pp: 207-210. DOI: 10.1109/IITA.2009.165

Mempel, C., B. Cialla and R. Luxenburger, 2010. Optimized testing of modern protection equipment using transient simulation. Proceedings of the 10th IET International Conference on Developments in Power System Protection, Mar. 29-Apr. 1, IEEE Xplore Press, Manchester, pp: 1-5. DOI: 10.1049/cp.2010.0340

Pena, E.H.M., S. Barbon, J.J.P.C. Rodrigues and M.L. Proença Jr., 2014a. Correlational paraconsistent machine for anomaly detection. Proceedings of the IEEE Global Communications Conference, Dec. 8-12, IEEE Xplore Press, Austin, pp: 551-556. DOI: 10.1109/GLOCOM.2014.7036865

Pena, E.H.M., S. Barbon, J.J.P.C. Rodrigues and M.L. Proença Jr., 2014b. Anomaly detection using digital signature of network segment with adaptive arima model and paraconsistent logic." Proceedings of the IEEE Symposium on Computers and Communication, Jun. 23-16, IEEE Xplore Press, Funchal, pp: 1-6. DOI: 10.1109/ISCC.2014.6912503

Ponnuramu, V. and L. Tamilselvan, 2012. Data integrity proof and secure computation in cloud computing. J. Comput. Sci., 8: 1987-1995. DOI: 10.3844/jcssp.2012.1987.1995

Praetox Technologies, 2010. Low Orbit Ion Cannon. Intense School, NFOSEC Institute.

Prakash, P., G. Kousalya, S.K. Vasudevan and K.K. Rangaraju, 2014. Distributive power migration and management algorithm for cloud environment. J. Comput. Sci., 10: 484-491. DOI: 10.3844/jcssp.2014.484.491

Proença Jr., M.L., C. Coppelmans, M. Bottoli, A. Alberti and L.S. Mendes, 2004. The Hurst Parameter for Digital Signature of Network Segment. In: Telecommunications and Networking - ICT 2004, J.N. de Souza, P. Dini and P. Lorenz (Eds.), Springer Berlin Heidelberg, Brazil, ISBN-10: 978-3-540-27824-5, pp: 772-781.

Puketza, N.J., K. Zhang, M. Chung, B. Mukherjee and R.A. Olsson. 1996. A methodology for testing intrusion detection systems. IEEE Trans. Software Eng., 22: 719-729. DOI: 10.1109/32.544350

Sarin, H., M. Kokkolaras, G. Hulbert, P. Papalambros and S. Barbat *et al*., 2010. Comparing time histories for validation of simulation models: Error measures and metrics. J. Dynamic Syst. Measurement Control, 132: 061401-061411. DOI: 10.1115/1.4002478

Shannon, C.E., 2001. A mathematical theory of communication. SIGMOBILE Mobile Comput. Commun. Rev., 5: 3-55. DOI: 10.1145/584091.584093

Sinreich, D. and Y.N. Marmor, 2004. A simple and intuitive simulation tool for analyzing emergency department operations. Proceedings of the Winter Simulation Conference, Dec. 5-8, IEEE Xplore Press, pp: 1994-2002. DOI: 10.1109/WSC.2004.1371561

Tsagkaris, K., A. Galani, P. Demestichas, G. Nguengang and M. Bouet *et al*., 2012. Identifying standardization opportunities of an operator-driven, framework for unifying autonomic network and service management. Proceedings of the IEEE International Conference on Communications, IEEE Xplore Press, Ottawa, pp: 6921-6925. DOI: 10.1109/ICC.2012.6364850

Xiao, Z., C. Liu and C. Chen, 2009. An anomaly detection scheme based on machine learning for WSN. Proceedings of the 1st International Conference on Information Science and Engineering, Dec. 26-28, IEEE Xplore Press, Nanjing, pp: 3959-396 2. DOI: 10.1109/ICISE.2009.235

Zarpelão, B.B., L.D.S. Mendes and M.L. Proença Jr., 2007. Anomaly detection aiming pro-active management of computer network based on digital signature of network segment. J. Netw. Syst. Manage., 15: 267-83. DOI: 10.1007/s10922-007-9064-y

Zarpelao, B.B., L.S. Mendes, M.L. Proenca and J.J.P.C. Rodrigues, 2009. Parameterized anomaly detection system with automatic configuration. Proceedings of the IEEE Global Telecommunications Conference, Nov. 30-Dec. 4, IEEE Xplore Press, Honolulu, pp: 1-6. DOI: 10.1109/GLOCOM.2009.5426189