# Converting Network Attacks to Standard Semantic Web Form in Cloud Computing Infrastructure

Afshin Rezakhani Roozbahani
Department of Computer Engineering
Ayatollah Boroujerdi University
Boroujerd, Iran

Leila Rikhtechi
Department of Computer Engineering
Islamic Azad University
Boroujerd Branch, Iran

Nasibe Mohammadi
Department of Computer Engineering
Islamic Azad University
Arak Branch, Iran

## ABSTRACT

Nowadays security has an important role in communications. The major weakness in detection/prevention systems is that the power of them is restricted only to the network on which algorithms are applied. This paper presents a new method to solve the problem of their localities. We propose place snorts with the capability to convert detected attacks properties to semantic web forms (SWFs) in several verified servers in cloud computing infrastructure. The major advantage of this approach is that all intrusion detection/prevention systems in world can use SWFs to detect/prevent any attack well. We will evaluate this method and show that the resulted traffic is balanced by the time.

**Keywords:** Snort; Semantic Web; Servers

## 1. INTRODUCTION

Network security starts from authenticating the user, commonly with a username and a password. Once authenticated, a firewall enforces access policies such as what services are allowed to be accessed by the network users [1]. Though effective to prevent unauthorized access, this component may fail to check potentially harmful content such as computer worms or Trojans being transmitted over the network. Anti-virus software or an intrusion prevention system helps detect and inhibit the action of such malware [2].

## 2. BACKGROUND

### 2.1. Intrusion Detection/Prevention Systems

In this section we consider a few systems that are used in networks to detect and pursue anomalies.

- **Firewalls**

Firewall is a program or hardware device that controls the flow of information coming through the Internet connection into the private network. When a packet of information reaches the firewall, the filter will examine the content of the packet and the packet is allowed to go through if it does not violate the rules that are set in the firewall implementation [3].

- **IDSs**

The intrusion detection system [4] is a protecting system which detects the anomaly occurrences on the network. The proposed approach uses intrusion detection which includes collecting data, searching ports, achieving the control of computers, and ultimately hacking; it can report and control the intrusions

- **Honeypots**

Honeypots are closely monitored network decoys serving several purposes: they can distract adversaries from more valuable machines on a network, they can provide early warning about new attack and exploitation trends and they allow in-depth examination of adversaries during and after exploitation of a honeypot [5].

- **Snorts**

Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol and anomaly-based inspection, Snort is the most widely deployed IDS/IPS technology worldwide. With millions of downloads and over 250,000 registered users, Snort has become the de facto standard for IPS [6].

### 2.2. Cloud Computing Infrastructure

Cloud computing is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like a public utility. It is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service, Platform-as-a-Service and Software-as-a-Service. The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams [7].

### 2.3. Concept of Semantic Web

The Semantic Web is an evolving development of the World Wide Web in which the meaning (semantics) of information and services on the web is defined, making it possible for the web to "understand" and satisfy the requests of people and machines to use the web content [8, 9].

## 3. OUR PROPOSED APPROACH

There are many intrusion detection/prevention systems in world with different detector algorithms in them. Traditional algorithms did not able to detect attacks well. We try to able all intrusion detection/prevention systems such as Firewalls, IDSs, Honeypots and Snorts for recognizing any attack behavior well and we effort to create an understandable platform between all of them. So we propose place several servers in a cloud computing infrastructure with modern and intelligence algorithms [16] for detecting the attacks by Snort systems continuously. When an attack is detected in any server, snort that exist in it, convert attack to a semantic web form (SWF) and then save it in own knowledgebase and at last send to other servers. These servers collaborate together and update their knowledgebase continually. There are knowledgebase and converters in all IDSs/IPSs in the world. When they receive any doubtful connection in the network, convert this behavior to semantic web form and compare with their knowledgebase to understand that this connection is safe or anomaly. In the other hand, all intrusion detection/prevention systems use of servers to update their knowledgebase and detect attacks well. This approach is shown in the figure1.

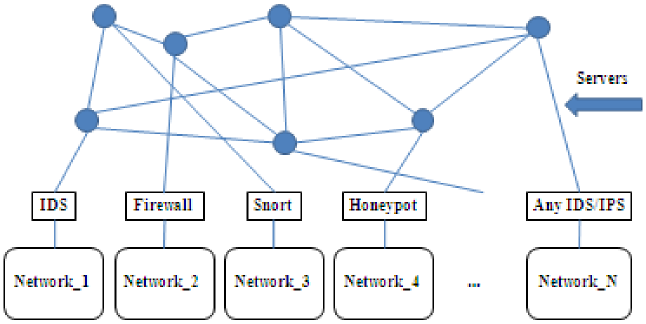


Figure1. Placing servers to give attacks properties to all IDSs/IPSs

Now we consider this approach accurately in bellow sections.

## 3.1 Servers Topology in Cloud Computing Infrastructure

In our proposed approach the connections between servers are not changed in Internet and they can be existed in any local network or wide network physically. But we suggest that mobile agents create connections between servers [17] for making a logical model of them. We assume the servers are nodes in a graph model in cloud computing infrastructure. The follow Figure show the servers in cloud computing infrastructure in graph model.

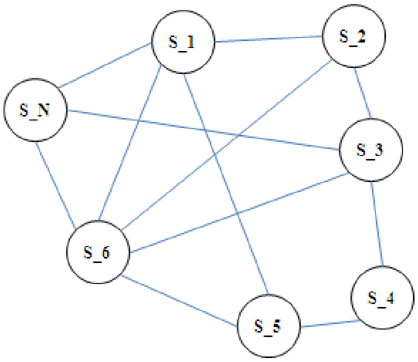


Figure2. Servers in Graph model

These nodes can have a cost in edges that is computed via traffics rate between servers, because exist several systems among two servers.

## 3.2 Convert the Detected Attacks to Semantic Web Model

The semantic web comprises the standards and tools of XML, XML Schema, RDF, RDF Schema and OWL that are organized in the Semantic Web Stack. The ontology section has an important role to convert any concept in networks into a standard forms. It was explained by Jeffrey Undercoffer *et al* paper copiously [10]. For example they showed ontology in Syn Flood Attack is similar to Figure3.

```
<Intrusion:Host rdf:about='&IntrOnt;00035"
         Intrusion:IP_Address="130.85.112.231"
         rdfs:label='00035'>
        <Intrusion:resulting_in rdf:resource=
                   '&IntrOnt;00038"/>
</Intrusion:Host>

<Intrusion:Syn_Flood rdf:about="&IntrOnt;00038"
         Intrusion:Exceed_T='true"
         Intrusion:time='15:43:12"
         Intrusion:date='02/22/2003"
         rdfs:label='00038'/>
```

Figure3. Notation for an Instance of a Syn Flood Attack

We propose to use of fuzzy logic [18] for converting detected attacks to RDF schema in order to improve the power of semantic web forms that will be created in intrusion detection/ prevention systems. So we convert any detected attack to subsets of properties according figure4. Of course this form is referred for other aims in another paper [11].

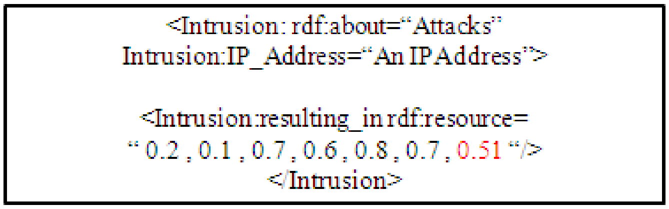
| Attack ID | 1 | 2 | 3 | 4 | 5 | . | . | . | M |
|---|---|---|---|---|---|---|---|---|---|

Figure4. Details of converting the attacks

The description of above fields is explained in bellow.

Field 1: A fuzzy value between 0 and 1, which shows the number of packets in terms of time.
Field 2: A fuzzy value between 0 and 1, which shows the achievement rate to abnormal ports.
Field 3: A fuzzy value between 0 and 1, which shows the threat rate of a connection (For this reason the connections with high rates of external connectivity can be considered as threats).
Field 4: A fuzzy value between 0 and 1, which shows the demand rate to achieving to system files in computers inside the network.
Field 5: A fuzzy value between 0 and 1, which shows the threat rate from outside of network for running malicious script.
Field M: To have a comprehensive evaluation metric, we calculate the average value of the above factors.
Attack ID: This field determines a unique name for any detected attack.

When an attack is detected for sending to other servers, all of above fields can be used in novel semantic web form in cloud computing infrastructure. This explanation is shown in figure5.



```
<Intrusion: rdf:about="Attacks"
Intrusion:IP_Address="An IP Address">

<Intrusion:resulting_in rdf:resource=
" 0.2 , 0.1 , 0.7 , 0.6 , 0.8 , 0.7 , 0.51 "/>
</Intrusion>
```

Figure5. The Semantic Web Form of a detected Attack

### 3.3 Saving Detected Attacks Properties in Knowledgebase

We propose to save intrusion detection systems (Snort) knowledgebase in servers that exist in cloud computing infrastructure. These servers are showed in Figure.6. Because snorts are open source software and we can add the newest methods in them quickly, it is suitable to use of it in our servers.

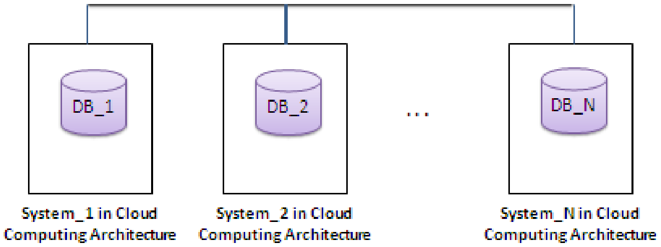
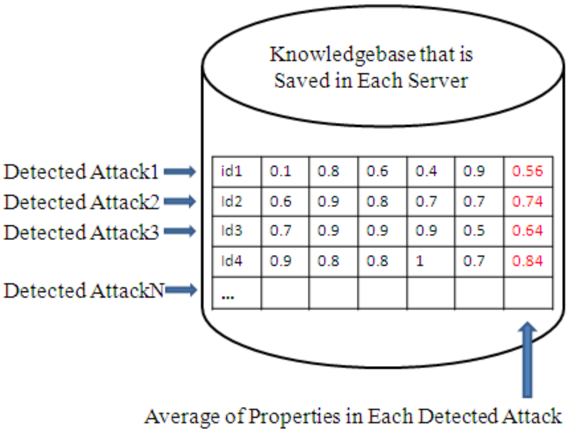

DB_1
DB_2
…
DB_N
System_1 in Cloud Computing Architecture
System_2 in Cloud Computing Architecture
System_N in Cloud Computing Architecture

Figure6. Placing Knowledgebase of IDSs/IPSs in Cloud Computing Infrastructure

In this paper we do not discuss about algorithms which exist in IDSs and IPSs instead of this study, we focus to standard detected attacks in all intrusion detection/prevention systems. We suggest placing intrusion detection/prevention systems (snort) with intelligence and new methods such as neural network [16] approaches to detect attacks in each server. These servers can be placed in each topology such as Star and use of knowledgebase that exist in other servers. When one of them detects any attack behavior, four follow steps are done:

1. The server that detects an attack converts it to semantic web form and searches in its knowledgebase.
2. If this behavior exists in knowledgebase, the server does not do any work.
3. If this behavior does not exist in knowledgebase, the server stores SWF in its knowledgebase.
4. This attributes send to other servers in cloud computing infrastructure.

The accurate schema of server's knowledgebase is shown in figure7.


Knowledgebase that is Saved in Each Server
Detected Attack1
Detected Attack2
Detected Attack3
Detected AttackN
Average of Properties in Each Detected Attack

| | | | | | | |
|---|---|---|---|---|---|---|
| id1 | 0.1 | 0.8 | 0.6 | 0.4 | 0.9 | 0.56 |
| id2 | 0.6 | 0.9 | 0.8 | 0.7 | 0.7 | 0.74 |
| id3 | 0.7 | 0.9 | 0.9 | 0.9 | 0.5 | 0.64 |
| id4 | 0.9 | 0.8 | 0.8 | 1 | 0.7 | 0.84 |
| ... | | | | | | |

Figure.7 Saving Semantic Web form of detected attacks in Servers Knowledgebase

### 3.4 Sending Detected Attacks to Servers with Mobile Agents

In this section we explain the way of sending detected attacks (Web Semantic form) to other servers in above graph. Relations between nodes can be created by mobile agents. Because we place intelligent and powerful algorithms in Snorts of servers, they can detect any attack well. When an attack is detected in one of them, snort converts detected attack property to web semantic form. Then, this semantic web form is placed in a mobile agent and sends to other servers. Accurately, follow steps are done:

1. A detected attack is converted to semantic web form (SWF) with Snort.
2. Snort searches in own knowledgebase for finding SWF that is created from last step.
3. If SWF is existed in snorts knowledgebase, any thing happens.
4. Else, a mobile agent is created for each existent connection in detector server. (Each connection has an especial mobile agent)
5. SWF is sent with mobile agents to all connected servers.
6. Each server that receive SWFs searches own knowledgebase. If they are not existed in knowledgebase, do all above steps again.

By doing these steps, server's knowledgebase is improved continually and servers can use of from other servers abilities.

### 3.5 Considering the security of Mobile Agents

The security of mobile agents is an important point for designing them. We propose to make a digest message from them and encrypt with RSA [12] before sending. Then these encrypted digests and mobile agents are sent to other servers. In destination servers the digest messages are made again from received mobile agents and compared with decrypted digested message. If the result of this comparison is dissimilar, correlative mobile agents are not accepted. This approach shows in figure8.

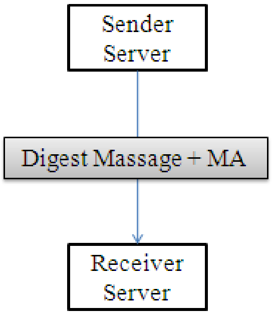


Figure8. Sending Digest Massage with Mobile Agents between servers

## 4. EVALUATION THE PROPOSED APPROACH

Proposed approach will create a standard platform for all intrusion detection/prevention systems and all of them can use of semantic web forms that are placed in servers to detect attacks in cloud computing infrastructure. But our attention focuses in the traffic rate that is entered over the network. In this section, first we consider the probability of identical attacks accordance and then compute the overhead traffic rate that enter in the network.

### 4.1 Considering the Rate of Identical Attacks

We consider in this section the rate of identical attacks that happened in several scenarios. Japan accounted for 30% of all attack traffic. The United States had the second-highest percentage of attack traffic for the second quarter, at 21.5%, while China came in third at 16.8%. But the rate of attack traffic in worldwide is only 3.6% [13]. So there are about 360 attacks between 10000 connections in the world. All IDSs/IPSs try to increase their abilities for detecting the attacks. For this reason the modern systems can detect over 99% of attacks [14]. Therefore, 99% * 360 or 356 of total connections in world are the recognizable attacks in 10000 connections. Now we compute the rate of identical attacks in several scenarios with poisson distribution [15] method.

- Senario1: If 99% of attacks are dissimilar, the average of identical attacks in 10000 connections is:

λ = 0.99 * 356 = 352; // It means that the 352 in 10000 connections are dissimilar detected attacks.
So the probability of identical attack occurrence is:

$$A1 = P(X <= 352) = \sum_{i=0}^{352} \frac{e^{-\lambda_{*}}\,\lambda^{i}}{i!} = 0.5447$$

- Senario2: If 98% of attacks are not similar, the average rate of identical attacks in 10000 connections is:

λ = 0.98 * 356 = 348; // It shows that the 348 in 10000 connection are dissimilar detected attacks.
So the probability of identical attack occurrence is:

$$A2 = P(X <= 348) = \sum_{i=0}^{348} \frac{e^{-\lambda_{*}}\,\lambda^{i}}{i!} \quad = \quad 0.5454$$

…

- Senario10: If 90% of attacks are dissimilar, the average rate of identical attacks in 10000 connections is:

λ = 0.90 * 356 = 320; // It describes that the 320 in 10000 connection are dissimilar detected attacks.
So the probability of identical attack occurrence is:

$$A100 = P(X = 0) = \sum_{i=0}^{0} \frac{e^{-\lambda_{*}}\,\lambda^{i}}{i!} \quad = \quad 1$$

The result of all above scenarios is shown in figure9.

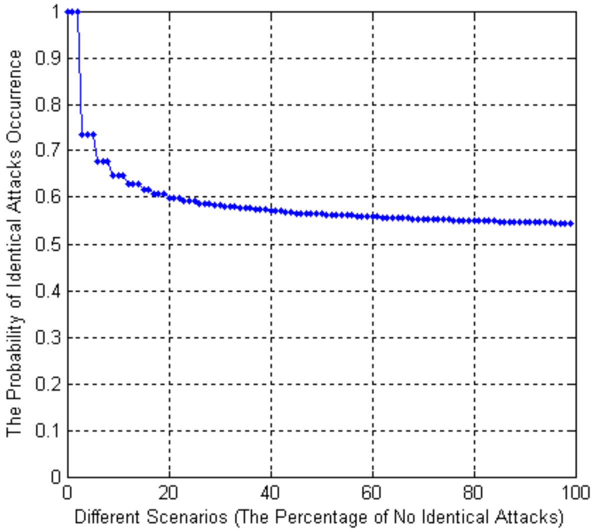


Figure9. Considering the probability of Identical Attacks in Hundred Scenarios

### 4.2 Considering the Average of all probabilities of Identical Attacks

The average of probabilities in all above scenarios is computed according under formula:

$$AVG = \frac{(A1+A2+\cdots+A99+A100)}{100} = 0.5910$$

The values of A1 to A100 are computed from last section. It means that near the 59% of attacks are the identical detected attacks.

### 4.3 Considering the Entered Traffic on the Network

In this section we compute the traffic value that is entered in network. We show the traffic value with "R" and the basic value of traffics between servers with "A" that can be variable from –α to +α . When the semantic web form of detected attacks sends to other servers the traffic value shows by "B". The value of "R" computes as bellow:

$$R = \begin{cases} A +/- \alpha & \text{if Random\_Number <= 59} \\ (A +/- \alpha) + B & \text{if Random\_Number > 59} \end{cases}$$

For example if A = 100 KB and α = -10KB … +10KB and B = 20KB, the traffic values that is entered over the network is shown in figure.10.

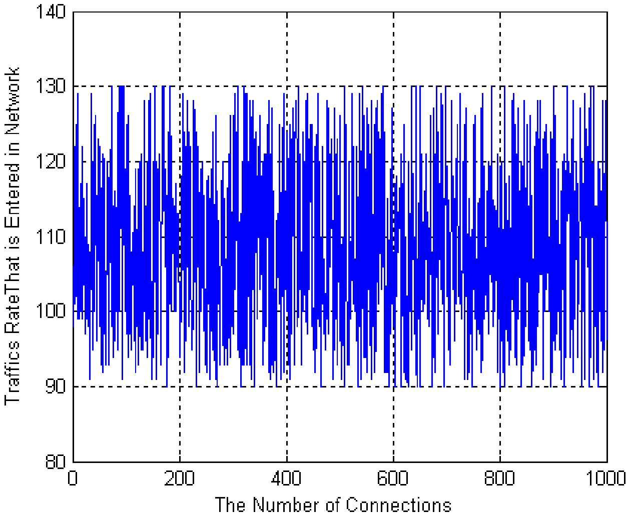


Figure10. Traffic Rate which is entered in Network

This figure show the traffic between servers does not create many extra overhead in the networks and will be balanced. The attacks will be identical and it is not necessary to send them to other servers. So the other important result that obtains from two last figures show the reduction of the traffic rates between servers by the time. The traffic rate for above example shows in figure11. The most important advantage of this evaluation is its ability for showing the real results and it is possible when we enter real values to A, B and α.

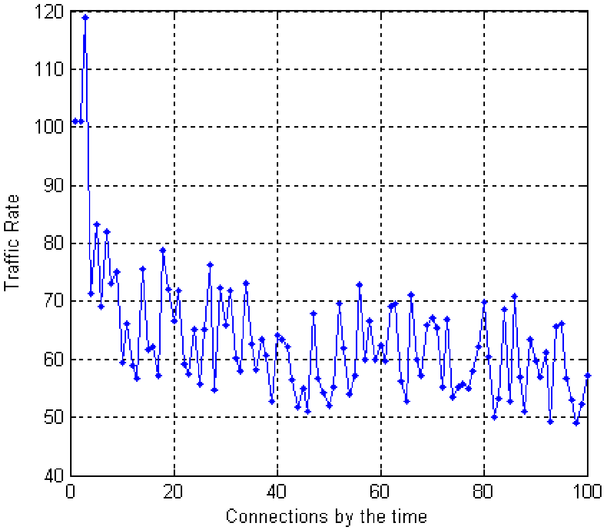


Figure11. The Traffic Rate between Servers by the Time

## 5. CONCLUSION

In this article, a new method is explained to improve the power of all intrusion detection systems such as IDSs, Firewalls and Snorts. The presented method is prepared a flexible infrastructure to take advantage of modern algorithms that is used in detecting the attacks. The proposed approach is able to make a standard platform for all of them. We proposed to create servers in cloud computing infrastructure with modern and intelligence methods. When an attack is detected in a server, snort convert detected attack to a semantic web form and save it in knowledgebase then send to other servers. These servers update their knowledgebase by having relation together. So all intrusion detection/prevention systems use from servers knowledgebase and detect attacks well. On the other hand, the crashes of these servers do not make catastrophic failures on the systems. Also the entered traffics do not create any huge traffic in the network and there will be balanced between them by the time.

## 6. REFERENCES

[1] A Role-Based Trusted Network Provides Pervasive Security and Compliance - interview with Jayshree Ullal, senior VP of Cisco

[2] Dave Dittrich, “Network monitoring/Intrusion Detection Systems (IDS)”, University of Washington.

[3] Kevin Law, “An Introduction of Firewall: Architectures and Functions”, SE 4C03 Computer Networks and Computer Security, 2005.

[4] Steven E Smaha. Haystack: An Intrusion Detection System. In Fourth Aerospace Computer Security Applications Conference, pages 37-44, Tracor Applied Science Inc., Austin, Texas, December 1988.

[5] http://www.honeypots.net/

[6] http://www.snort.org/

[7] Sun microsystems, "Introduction to Cloud Computing architecture" White Paper 1st Edition, June 2009.

[8] Berners-Lee, Tim, James Hendler and Ora Lassila, "The Semantic Web", Scientific American Magazine, Retrieved March 26, 2008.

[9] http://www.w3.org/2001/sw/SW-FAQ. Retrieved March 13, 2008.

[10] Jeffrey Undercoffer, Anupam Joshi, and John Pinkston, "Modeling Computer Attacks: An Ontology for Intrusion Detection", Springer, LNCS 2820, pp. 113–135, 2003.

[11] Afshin Rezakhani Roozbahani, Ramin Nassiri and GolamReza Latif-Shabgahi, "Service Oriented Approach to Improve the Power of Snorts", ICCEE 2009, UAE, December 2009.

[12] R.L.Rivest, A.Shamir and L.Adleman "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM 21 (2): 120–126, 1978.

[13] URL:http://www.networkworld.com/news/2008/090908-japan-attacktraffic.html, Last visited in March 2010.

[14] S.A.Thorat, A.K.Khandelwal, B.Bruhadeshwar and K.Kishore, ”Payload Content based Network Anomaly Detection”, Centre for Security Theory and Algorithmic Research (CSTAR) IIIT-Hyderabad, India, 2007.

[15] Joachim H. Ahrens and Ulrich Dieter, "Computer Methods for Sampling from Gamma, Beta, Poisson and Binomial Distributions", Computing 12 (3) 223–246, 1974.

[16] Shun, J. Malki, H.A., "Network Intrusion Detection System Using Neural Networks", ICNC 08, Jinan , November 2008.

[17] S.A.Onashoga, Adebayo D.Akinde and A. S.Sodiya, “A Strategic Review of Existing Mobile Agent-Based Intrusion Detection Systems” Department of Computer Science, University of Agriculture,Abeokuta, Nigeria, Volume 6, 2009.

[18] http://www.seattlerobotics.org/Encoder/mar98/fuz/flindex.html, Last visited in March 2010.