ISSN : 1978-6603

JURNAL Ilmiah SAINTIKOM Sains dan Komputer

OPTIMIZATION OF LEVENSHTEIN TECHNIQUE AND INTRUSION DETECTION SYSTEM METHOD TO OVERCOME IN THE MIDDLE ATTACK FROM INTRUDER ON BASED NETWORK TPC/IP

***Dicky Nofriansyah[#1], Saiful Nur Arief[#2] , Badrul Anwar[#3]**
[#1] Student of Padang State University
[#1,2,3,] Information Technology Of STMIK Triguna Dharma, Indonesia
E-Mail: [#1]dickynofriansyah@ymail.com

Abstract

The presence of firewalls has helped in securing the network, but as technology evolved today only with security firewalls that can't be fully guaranteed data security from theft of parties who are not responsible. Because it has developed technology IDS (Intrusion Detection System) is a method of data security on a computer network. With the IDS then these attacks can be prevented or eliminated over. In developing the IDS method is used as a prototype model of system development methods. By using the Levenshtein Technique IDS method of attack intruders either from outside or within the computer network can be seen, making it easier for an admin in handling, so creating a more optimal network security.

***Keywords**:* Levenshtein Technique, Intrusion Detection System, Middle Attack

Abstrak

Kehadiran firewall telah membantu mengamankan jaringan, namun seiring perkembangan teknologi saat ini hanya dengan firewall keamanan yang tidak dapat sepenuhnya dijamin keamanan data dari pencurian pihak-pihak yang tidak bertanggung jawab. Dalam perkembangan teknologi salah satunya menggunakan IDS (Intrusion Detection System). IDS (Intrusion Detection System) adalah metode keamanan data pada jaringan komputer. Dengan IDS maka serangan ini bisa dicegah atau dihilangkan. Dalam mengembangkan metode IDS digunakan sebagai model prototype dari metode pengembangan sistem. Dengan menggunakan metode IDS dengan Levenshtein Teknik serangan penyusup baik dari luar maupun dalam jaringan komputer dapat dilihat, sehingga memudahkan admin dalam menangani, sehingga menciptakan keamanan jaringan yang lebih optimal.

***Kata Kunci**: Teknik Levenshtein, Intrusion Detection System, Middle Attack*

I. INTRODUCTION

1. Background of the Study

The development of technology, especially the Internet (interconnection networking) is very rapid and significant in the life of this present moment. That is because the Internet is easy to access and use, and speed delivery of data to other parties. In addition to the speed and convenience that exist on the internet, the internet is also a lot that can be done by the user, such as browsing and chatting. So, do not be surprised if the internet is so fast developing and broad public interest.

But the development of the Internet that there are still many users who do not realize or do not even know that, in making its activities on the internet, such as chatting or sending e-mail to other parties, this activity is extremely vulnerable to cybercrime, or in a language called cybercrime, Cybercrime in question is to steal data or important information from the victim without the knowledge or permission, of course this is extremely dangerous or harmful to users who experience it. All that can happen is usually caused by a lack of understanding or ignorance of users use the Internet safely from cybercrime.

One crime of cybercrime that is phishing. Phishing is a form of fraudulent activities undertaken to obtain important information such as usernames, passwords, PINs and so on. Phishing activity generally occurs in the TCP / LP-based wireless or wireless networks, where the position of the attacker or attackers are in the midst of victims who were conducting a conversation with a specific purpose. Attacker in this case modifying the Address Resolution Protocol (ARP) cache so that data packets will turn toward the attacker before reaching the destination computer, and the term is called man in the middle attack. The attack in the middle attack is very dangerous, not only the attacker can steal passwords and even control the data among the victims who were conducting the conversation.

Distance Levehenstein created by Vladimir Levehenstein in 1965. The calculation of the edit distance matrix used is calculated by counting the number of differences unuk string between two strings. Distance Levenshtein distance between two strings is defined as the minimum number of edits required to transform one string to another, with the allowable edit operations, namely the insertion, deletion or substitution of a single character

The calculation of the distance between the two strings specified minimum number of operations of two changes to create a string A string becoming B. This algorithm runs starting from the top left corner of a two-dimensional array that has filled a number of the initial string and the string of targets and given the value of cost. The value of cost at the bottom right corner of edit distance into a value

that describes the amount of difference of two string.

2. PROBLEM STATEMENT

This research is focused to identifying and detection intruder on network TCP/IP using Levensthein Technique and Intruder Detection System.

II. ANALYSIS AND PROCEDURE

In doing in the middle attack on a local network using a WI-FI access points can be done by using ARP (Adress Resolution Protocol) already existing on TCP / IP. ARP is a form of protocol for data transfer layer that works on OSI layers to 2. In this case the ARP will connect between the transfer layer from the hardware interface that simultaneously run and serve the higher layers (network layer). Usually the activity in the middle attack on a TCP / IP network utilizing IP scanning to determine whether or not the device is connected on the network.

Snort is a software that basically created as monitoring traffic passing data on the network. To use Snort as an intrusion detection system (IDS) then be made a rule that serves to detect in the middle attack on the network.

The algorithms of Intrusion Detection System (IDS) is:

1. Determine the protected network on snort.
2. Make the snort IDS rule.
3. Running Snort.
4. Placement IDS
5. Search String of Intruder using LT

In determining the protected network, first of all you need to know the IP network, and then enter the IP network into a configuration called snort conf. Given a wireless network with IP 192.168.43.1 server is a network-based protocol TCP / IP. To protect the network with IDS then created a configuration in confidence snort as follows:

- ipvar HOME_NET 192.168.43.1
- ipvar DNS_SERVERS $HOME_NET
- ipvar SMTP_SERVERS $HOME_NET
- ipvar HTTP_SERVERS $HOME_NET
- ipvar SQL_SERVERS $HOME_NET
- ipvar TELNET_SERVERS $HOME_NET
- ipvar SSH_SERVERS $HOME_NET
- ipvar FTP_SERVERS $HOME_NET
- ipvar SIP_SERVERS $HOME_NET
- portvar SHELLCODE_PORTS !80

To create a Snort intrusion detection system, the rule is a very important thing. With the existence of a rule that could serve as a snort IDS.

IDS Rule is a rule made on snort to detect attempted intrusions in packets through the network. If the packet is detected as an attempt to infiltrate, then snort will show a warning. Here is a rule in Snort to detect an IP scanning experiments on a network:

Alert tcp any any -> any any (msg: ”Peringatan !!! scan IP terdeteksi”; flags : FPU;sid:9000003;)

The purpose of the above rules are warning the entire IP that led to all the IP with the message "Warning!!! IP Scan detected "and show proficiency level IP. So, that with the establishment of the

rule can detect attempted intrusions on the network.

Scanning IP is an action taken to map the entire IP for the network. With IP scanning an intruder can determine how many computers are connected to a network and can choose the right type of attack in infiltrating into the desired computer. Therefore, IP Scanning can be detected from the increasing amount of traffic data sent from one computer to all networks.

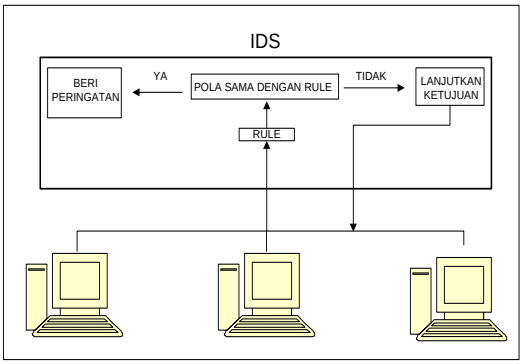

Figure.1 Scheme of IDS

Snort has functioned as IDS will display a warning when there is a data packet which contains a threat. To run snort as IDS snort -c command by typing c:\ snort\etc\snort.conf -A -K -i2 ascii console at the command prompt on windows operating system.

Intrusion Detection System on a network whether it will work well, depending on placing. In principle, the understanding of the placement of IDS components (networks, sensor systems, security agent, and deception system) will produce IDS is really easy to control so that safeguards networks from attacks become more efficient. The sensor is an important component of IDS therefore placement should be considered. Sensor network for intrusion detection system is usually installed in the following locations:

1. Between the router and firewall
2. In the demilitarized zone
3. Behind the firewall
4. Close the remote access server
5. In the network backbone
6. With the key internal network segment
7. On the remote office

To protect the network from intruders, the function of the sensor network is very important. The first to do is to install the sensor network between the router with a firewall. These sensors provide access to control all network traffic, including at the demilitarized zone. The following illustration IDS laying on the image

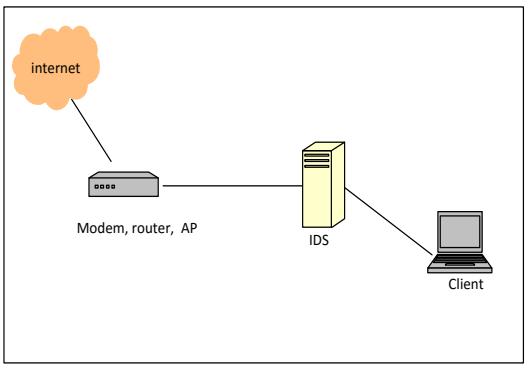

Figure.2 Placing of IDS

From Figure 2 can be explained that each data packet that passes through a network that is installed IDS always processed in order to prove that the package does not contain elements across the network attacks, malicious

code, bad traffic, and other things that could harm the network system itself. In this similarity detection system has a way of working that is by making string comparisons and entering data.

To search for similarities between these data, this method is used initialization (String source), b (String target), i (the value of every character in the source string), and j (the value of every character in the string target), Then do input data a and data b to find a comparison between the data with the data b by calculating the value of i and j to obtain the value of distance (distance). Similarity value data will be obtained using the "Similarity" or resemblance. As it is known that the distance required levenshtein engineering data that will be used to create a string

In this process using a string name as an example of the calculation. For working early in the calculation of method here is to make a comparison matrix between the data and the string value "MUHAMMAD HASAN" and the comparison is "MUHAMMAD IHSAN". To find the value range (distance) that his later used to search for similarity value data, the following is a comparison matrix table. The next step that needs to be done is to determine the value of the distance (distance) from the string. In this process use the following formula to find the value of distance (distance).

Lev a,b = max (i,j)

If min(I,j)=0

Min(i – 1 , j) + 1 → *Deletion*

min(i , j – 1) + 1 → *Insertion*

Min(i – 1 , j – 1) + 1 (a ≠ b) → *Subtitution*

In this case, There are 3 surgery to look for the string spacing, namely:

1. Removal Operations (deletion)
   Removal operation is done by removing the character at index.
   Lev a, b (i-1, j) +1
2. Operation Insertion (insertion)
   Insertion operation is done by inserting a character on the index.
   Lev a, b (i, j-1) +1
3. Operation exchange (substitution)
   Exchange operations conducted by swapping characters on the index.
   Lev a, b (i-1, j-1) +1 if (a ≠ b)

phase 1

- Lev M, M (MIN (i - 1, j - 1) + 0)
  MIN (1-1, 1-1) + 0
  MIN (0, 0) + 0 = 0
- Lev M, U (MIN (i, j - 1) + 1)
  MIN (1, 2-1) + 1
  MIN (1, 1) + 1 = 1
- Lev M, H (MIN (i, j - 1) + 1)
  MIN (1, 3-1) + 1
  MIN (1, 2) + 1 = 2
- Lev M, A (MIN (i, j - 1) + 1)
  MIN (1, 4-1) + 1
  MIN (1, 3) + 1 = 3
- Lev M, M (MIN (i - 1, j - 1) + 0)
  MIN (1-1, 5-1) + 0
  MIN (0, 4) + 0 = 4
- Lev M, M (MIN (i - 1, j - 1) + 0)
  MIN (1-1, 6-1) + 0
  MIN (0, 5) + 0 = 5
- Lev M, A (MIN (i, j - 1) + 1)
  MIN (1, 7-1) + 1
  MIN (1, 6) + 1 = 6
- Lev M, D (MIN (i, j - 1) + 1)

MIN (1, 8-1) + 1
MIN (1, 7) + 1 = 7

- Lev M, I (MIN (i, j - 1) + 1)
  MIN (1, 9-1) + 1
  MIN (1, 8) + 1 = 8
- Lev M, H (MIN (i, j - 1) + 1)
  MIN (1, 10-1) + 1
  MIN (1, 9) + 1 = 9
- Lev M, S (MIN (i, j - 1) + 1)
  MIN (1, 11-1) + 1
  MIN (1, 10) + 1 = 10
- Lev M, A (MIN (i, j - 1) + 1)
  MIN (1, 12-1) + 1
  MIN (1, 11) + 1 = 11
- Lev M, N (MIN (i, j - 13) + 1)
  MIN (1, 13-1) + 1
  MIN (1, 12) + 1 = 12

Phase 2

- Lev U, M (MIN (i - 1, j) + 1)
  MIN (2-1, 1) + 1
  MIN (1, 1) + 1 = 1
- Lev U, U (MIN (i - 1, j - 1) + 0)
  MIN (1-1, 1-1) + 0
  MIN (0, 0) + 0 = 0
- Lev U, H (MIN (i, j - 1) + 1)
  MIN (1, 2-1) + 1
  MIN (1, 1) + 1 = 1
- Lev U, A (MIN (i, j - 1) + 1)
  MIN (1, 3-1) + 1
  MIN (1, 2) + 1 = 2
- Lev U, M (MIN (i, j - 1) + 1)
  MIN (1, 4-1) + 1
  MIN (1, 3) + 1 = 3
- Lev U, M (MIN (i, j - 1) + 1)
  MIN (1, 5-1) + 1
  MIN (1, 4) + 1 = 4
- Lev U, A (MIN (i, j - 1) + 1)
  MIN (1, 6-1) + 1
  MIN (1, 5) + 1 = 5
- Lev U, D (MIN (i, j - 1) + 1)
  MIN (1, 7-1) + 1
  MIN (1, 6) + 1 = 6
- Lev U, I (MIN (i, j - 1) + 1)
  MIN (1, 8-1) + 1
  MIN (1, 7) + 1 = 7
- Lev U, H (MIN (i, j - 1) + 1)
  MIN (1, 9-1) + 1
- MIN (1, 8) + 1 = 8
- Lev U, S (MIN (i, j - 1) + 1)
- MIN (1, 10-1) + 1
- MIN (1, 9) + 1 = 9
- Lev U, A (MIN (i, j - 1) + 1)
- MIN (1, 11-1) + 1
- MIN (1, 10) + 1 = 10
- Lev U, N (MIN (i, j - 1) + 1)
- MIN (1, 12-1) + 1
- MIN (1, 11) + 1 = 11

Table 1: Matrix Table

	0	M	U	H	A	M	M	A	D	I	H	S	A	N
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13
M	1	0	1	2	3	4	5	6	7	8	9	10	11	12
U	2	1	0	1	2	3	4	5	6	7	8	9	10	11
H	3	2	1	0	1	2	3	4	5	6	7	8	9	10
A	4	3	2	1	0	1	2	3	4	5	6	7	8	9
M	5	4	3	2	1	0	1	2	3	4	5	6	7	8
M	6	5	4	3	2	1	0	1	2	3	4	5	6	7
A	7	6	5	4	3	2	1	0	1	2	3	4	5	6
D	8	7	6	5	4	3	2	1	0	1	2	3	4	5
H	9	8	7	6	5	4	3	2	1	1	1	2	3	4
A	10	9	8	7	6	5	4	3	2	2	2	2	2	3
S	11	10	9	8	7	6	5	4	3	3	3	2	3	3
A	12	11	10	9	8	7	6	5	4	4	4	3	2	3
N	13	12	11	10	9	8	7	6	5	5	5	4	3	2

The process continues until stage 13 and get the value of the distance (distance) as follows:

Once this process is complete, then obtained within a string with a string comparison is 2 or edit distance = 2. The results were compared to the distance value is as follows:

Table 2. Result

1st String	**2nd String**
Muhammad Ahsan	Muhammad Ihsan
Distance	
1. Name: 2	
2. Age: 2	
3. Gender: 0	
4. Address: 10	
5. Case: 8	
Totally: 22	

Determining the value of similarity (similarity). In this process use the following formula to find the value similarity (similarity).

The formula used to find the value similarity (similarity) is as follows:

Similarity = (100 edit distance) x 100%

Similarity = (100 - *edit distance*) x 100

= 100 – 22 x 100

= 78 x 100

= 7800

= 78%

III. ACKNOLEDGEMENTS

In this research, researchers wanted to thank several parties. Especially to God who has given the outpouring of knowledge so as to complete the research. In addition, researchers say thank you to both parents, his beloved wife, children and all family members as well as to all parties which has made its contribution to the completion of the study.

REFERENCES

The following is the references to support this research namely as follows:

[1] Srinivasan, SS, Anderson, R., & Ponnavolu, K. 2002. Customer loyalty in e-commerce: an exploration of its antecedents and consequences. Journal of Retailing, 78 (1), 41-50.

[2] Hendriana, Y., Umar, R., Pranolo, A. 2015. Modelling and Design E-Commerce SMI Sector Using Zachman Framework. International Journal of Computer Science and Information Security, 13(8), 9-14.

[3] Fansyuri, Ahmad. 2012. Aplikasi E-Commerce Penjualan Parfum Secara Online. Skripsi. Yogyakarta: Program Studi Teknik Informatika Universitas Ahmad Dahlan.

[4] Yanti, Nur Fitri. 2011. Implementation of E-Commerce Sales Book on Ombak Publisher Based Framework. Skripsi. Yogyakarta: Informatics Department Universitas Ahmad Dahlan.

[5] Beijering, K., Gooskens, C., & Heeringa, W. 2008. Predicting intelligibility and perceived linguistic distances by means of the Levenshtein algorithm.Linguistics in the Netherlands, 15, 13-24.

[6] Yujian, L., & Bo, L. 2007. A normalized Levenshtein distance metric. Pattern Analysis and

Machine Intelligence, IEEE Transactions on, 29(6), 1091-1095.

[7] Hyyrö, H. 2003. A bit-vector algorithm for computing and Damerau Levenshtein edit distances. Nord. J. Comput., 10 (1), 29-39.

[8] Soukoreff, RW, & MacKenzie, IS. 2001. Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic. In CHI'01 extended abstracts on Human factors in computing systems (pp. 319-320). ACM.

[9] Basuki Achmad, 2011, Software Process Model, Jakarta, PT. Mizan Publika.

[10] Grandon, E. E., & Pearson, J. M. 2004. Electronic commerce adoption: an empirical study of small and medium US businesses. Information & management, 42(1), 197-216.

[11] E.W.T. Ngai, F.K.T. Wat,. 2002. A literature review and classification of electronic commerce research, Information and Management 39, pp. 415-429.

[12] G.P. Schneider, J.T. Perry. 2000. Electronic Commerce, Course Technology, Cambridge, MA.

[13] Hendriana, Y., Hardi, R., & Pranolo, A. 2015. Design and Implementation of Online Fashion Store "Demi Outfits" Based on Android. International Journal of Computer Applications Technology and Research, Volume 4–Issue 6, 438 – 443

[14] Denning D."An Intrusion-Detection Model." IEEE Transactions on Software Engineering, Vol. SE-13, No 2, 1987.

[15] Peter Lichodzijewski A.Nur Zincir-Heywood, Malcolm I. Heywood "Host-based Intrusion Detection using Self Organizing maps" *IEEE Communications* 2002.