Creation of User Profiles and Finding Fraud Patterns

[1] Chanakya Pulikanti (M.Tech Student)

[2] A. Shiva Kumar (Assistant Professor)

Dept of Computer Science and Engineering

Mahaveer Institute of Science and Technology,
Jawaharlal Nehru Technological University, Hyderabad.

Keywords: **Model User, Fuzzy Logic, Rule Based Fuzzy, Evolution of the System Based Fuzzy, UNIX, Support Machine Based Vector.**

ABSTRACT

This is a system to detect the fraud user in a network. Generally, it is suggested that gaining knowledge about the computer user helps administrator in predicting user actions and find the Modus operandi of the user. It is the responsibility of the administrator to maintain and protect the network from various attacks and other malicious activities in and out of the network. Here, we use a system that captures the user activities in the form of LINUX commands i.e. commands typed by the Linux user on the terminal while at work. Sequence of LINUX commands typed by the user is captured and considered as a user profiles/behavior. Here the similarity function is used to compare the profiles and see how close they are. If the commands given by the current logged in user is a close match with the ones in the database then we can say that the probability of the current user thinking and activities are on similar lines. If the match is with suspicious users in the database then the user is considered as fraud user and is blocked.

1. Introduction:

Behavior of a Person can actually be known by his deeds. We can categories computer users into two, one the legitimate user and the other is the suspicious one or fraud user per say. To analyze the user in LINUX environment, we need to analyze the commands which user types on the command terminal. If the commands given by the user are tending towards some suspicious activities and if those commands project adverse effect on the servers then the user is considered as a fraud user.

In this paper, we use an adaptive approach for creating profiles of users and recognize them; this approach is called Evolving Agent (User) Behavior Classification Based on Distributions of relevant events (EVABCD). This is all about behavior of the user i.e. sequence of commands typed by the user. The major jobs which are proposed by the EVABCD is to create the profiles, update them if any relevant new sequences are found with respect to the corresponding profiles .The need of updating the fraud profiles is because behavior of the user changes, and may even use different strategies or commands to attack a server.

2. Methodology:

The focus is on

2.1 Profile construction.

2.2 Tree Traversal

2.3 Similarity function

2.4 Modus Operandi

2.5 Action taken.

2.1 Profile Construction:

Profile is constructed by the sequence of commands given by the computer user. This basically involves three steps. Segmentation of the sequences, storing all the subsequences into a trie. If the dependencies of the commands are relevant, the subsequence suffixes are also inserted into a trie. After inserting all the subsequences and their corresponding suffixes, the completed trie is obtained which is a user profile.

2.2 Tree Traversal:

After the trie is created, the subsequences that characterize the user profile and its relevance are calculated by traversing the trie. Traversal always starts from the root node.

As an example, Let us consider the general Ping flooding sequence entered by a LINUX user for attacking a server. We store all the subsequences into a trie, so that we can access them easily and at a faster rate.

Number of subsequences in the sequence is given by the formula:

[N-Length+1]

Where, N is the number of commands in a sequence.

Length is the subsequence length which will be provided by the user.

Number of commands per subsequence is given by:

2N (N =subsequence length)

Total number of different Patterns for the sequence is given by:

Number of subsequences * subsequence length

Sequence:

Ping –V, Ping 192.168.0.1, ping -I 5 192.168.0.1, ping –c 5 xyz.com, ping –s 100 192.168.0.1, ping –f 192.168.0.1

Considering Subsequence length as 3 so we get

=6-3+1 =3+1 =4

So we have 4 subsequences for the above sequence of equal length.

Subsequence 1 and its suffixes

Ping –V, Ping 192.168.0.1, ping -I 5 192.168.0.1

Ping 192.168.0.1, ping -I 5 192.168.0.1

Ping -I 5 192.168.0.1

Subsequence 2 and its suffixes

Ping 192.168.0.1, ping -I 5 192.168.0.1, ping –c 5 xyz.com

Ping -I 5 192.168.0.1, ping –c 5 xyz.com

Ping –c 5 xyz.com

Subsequence 3 and its suffixes

Ping -I 5 192.168.0.1, ping –c 5 xyz.com, ping –s 100 192.168.0.1

Ping –c 5 xyz.com, ping –s 100 192.168.0.1

Ping –s 100 192.1680.1

Subsequence 4 and its suffixes

Ping –c 5 xyz.com, ping –s 100 192.168.0.1, ping –f 192.168.0.1

Ping –s 100 192.168.0.1, ping –f 192.168.0.1

Ping –f 192.168.0.1

A trie is a tree-based date structure for storing strings in order to make pattern matching faster. All the subsequences are stored in trie data structure, so that the information can be retrieved faster. The search traversal for the right node for the right pattern will always start from the root.

Finally once the profiles are created it is updated into the database, which holds all the fraud profiles.

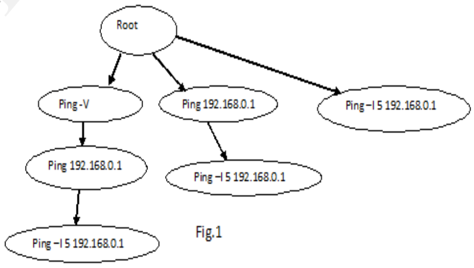

Fig.1

Fig1 Trie shows the subsequences and suffixes of a sequence

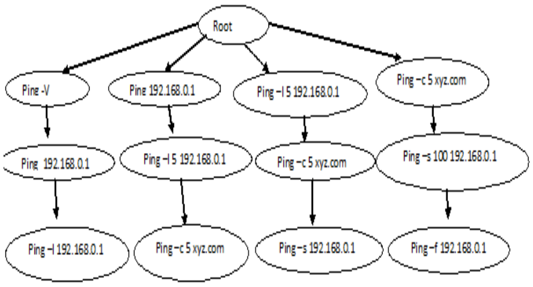

Fig. 2

Fig2 below Trie shows all the subsequences of the sequence.

2.3 Similarity function:

Database holds all the profiles. Two profiles are said to be similar only when the cosine distance between them is less. The current logged in users profile is compared with all the profiles in the database, and the least distance one is considered to be the best match.

2.4 Finding fraud MO:

Modus operandi in our context is nothing but method of operation by a fraud user. The term is used to describe the pattern followed by a fraudster. In this case, if a Person uses similar commands or same commands to achieve his targets all the time, like, coming from the same geographic location, login-in from the same system, same IPdomain is a pattern followed by fraudster.

A modus operandi can assist administrator in identification, apprehension and can also be used to determine link between frauds. Here we take, similarity function, user information along with some network parameters into consideration to catch the Modus operandi of the user. If Current logged-in User profile matches with fraud profiles in the database and matches with the IP address or the IPdomain, hailing from the same department as well as the designation are the strong parameters to confirm the current user as a fraud user.

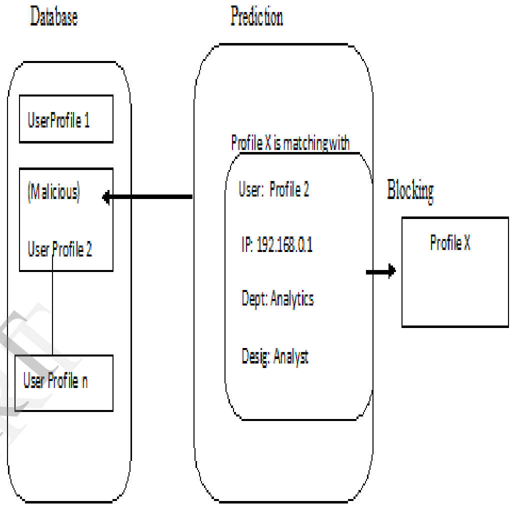

Fig 3

Fig.3 Block diagram of the system, which Shows profile matching and other parameters to confirm fraud.

Fig.4 Below, Depecting how profiles are directly matched and how profiles are logically interlinked with other profiles, which are in the Database.

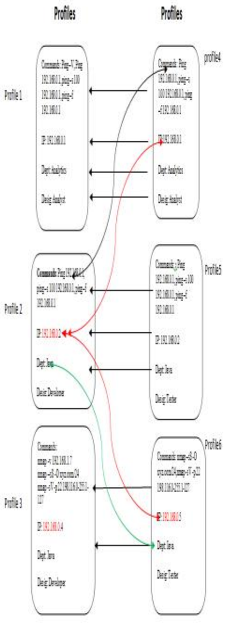
Profiles
Profiles
Profile 1
Commands: Ping -V, Ping 192.168.0.1, ping -c 100 192.168.0.1, ping -f 192.168.0.1
IP: 192.168.0.1
Dept: Analytics
Desig: Analyst
profile4
Commands: Ping 192.168.0.1, ping -c 100 192.168.0.1, ping -f 192.168.0.1
IP: 192.168.0.1
Dept: Analytics
Desig: Analyst
Profile 2
Commands: Ping 192.168.0.1, ping -c 100 192.168.0.1, ping -f 192.168.0.1
IP: 192.168.0.2
Dept: Java
Desig: Developer
Profile5
Commands: Ping 192.168.0.1, ping -c 100 192.168.0.1, ping -f 192.168.0.1
IP: 192.168.0.2
Dept: Java
Desig: Tester
Profile 3
Commands: nmap -v 192.168.1.7 nmap -sS -O xyz.com/24 nmap -sV -p22 198.116.0-255.1-127
IP: 192.168.0.4
Dept: Java
Desig: Developer
Profile6
Commands: nmap -sS -O xyz.com/24 nmap -sV -p22 198.116.0-255.1-127
IP: 192.168.0.5
Dept: Java
Desig: Tester

Fig.4

Fig.4 Depecting how profiles are directly matched and how profiles are logically interlinked with other profiles, which are in the Database.

Fig 4

2.5 Action Taken:

After confirming the user as fraud, the user is blocked and is not allowed to login into the network further.

3. Results:

Lot of effort has been put in to design this system using the existing system to eliminate the fraudster from the network

Effects on Systems due to fraud activities

1) Slow network performance.
2) Consumption of resources, such as bandwidth, disk space, or processor time.
3) Continuous utilization of resources.
4) Disruption of state information, such as unsolicited resetting of TCP sessions.
5) Disruption of physical network components.
6) Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

REFERENCES

[1] J.H. Morra, Z. Tu, L.G. Apostolova, A. Green, A.W. Toga, and P.M. Thompson, "Comparison of Adaboost and Support Vector Machines for Detecting Alzheimer's Disease through Automated Hippocampal Segmentation," IEEE Trans. Medical Imaging, vol. 29, no. 1, pp. 30-43, Jan. 2010.

[2] P. Angelov, X. Zhou, and F.lawonn, "Evolving Fuzzy Rule-Based Classifiers," Proc. IEEE Symp. Computational Intelligence in Image and Signal Processing (CIISP '07), pp. 220-225, 2007.

[3] J.A. Iglesias, A. Ledezma, and A. Sanchis, “Sequence Classification Using Statistical Pattern Recognition,” Proc. Int’l Conf. Intelligent Data Analysis (IDA), pp. 207-218, 2007.

[4] G.A. Kaminka, M. Fidanboylu, A. Chang, and M.M. Veloso, “Learning the Sequential Coordinated Behavior of Teams from Observations,” Proc. RoboCup Symp., pp. 111-125, 2002.

[5] S. Greenberg, “Using Unix: Collected Traces of 168 Users,” master’s thesis, Dept. of Computer Science, Univ. of Calgary, Alberta, Canada, 1988.

[6] P. Angelov and D. Filev, “Simplets: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy Models,” Proc. IEEE Int’lConf. Fuzzy Systems (IEEE-FUZZ), pp.1068-1073, 2005.

[7] G. John and P. Langley, “Estimating Continuous Distributions in Bayesian Classifiers,” Proc. Conf. Uncertainty in Artificial Intelligence, pp. 338-345, 1995.

[8] www.google.com

[9] Creating Evolving User Behavior. Profiles Automatically. Jose Antonio Iglesias

[10] Various LINUX books.