A Review of Automated Intrusion Detection Models

Mala Dutta
Assistant Professor, Department of Computer Engineering,
IET DAVV, Indore

Abstract - *The aim of this paper is to review the working of various automated signature generation models for detection of previously unknown network attacks and polymorphic worms, which is based on anomalous payload of the network packets. We discuss the various types of worms, how they try to evade detection from static signature detection systems and the ability of eight different automated models to detect these worms.*

Keywords : Intrusion detection models, attack signature, polymorphic worms

1. Introduction

Implementation of network security can be categorized into three modes: defense (firewalls), deterrence (cyber laws) and detection. The third one is least commonly implemented on network systems and is carried out by Intrusion Detection Systems [4]. Intrusion Detection Systems are of two types:

i.) Signature-Based
ii.) Anomaly-Based

Signature based intrusion detection methods collect and use information from known types of attacks and find out if someone is trying to attack the network or a particular host in it. The advantage of these systems is that they have low false positive rates but have a serious drawback that they require a prior knowledge of the attack and are hence susceptible to novel attacks whose signatures are not known, making the system prone to attacks until the signature set is updated.

On the other hand, Anomaly-based systems build a normal profile of the network traffic, and mark any behavior that significantly deviates from the normal as an attack. This has the advantage that new attacks can get detected as soon as they take place. The disadvantage is that Anomaly-based systems need a training phase and hence a significant amount of data and time is needed to build an accurate network profile.

2. Analysis of signature generation models

A worm is a piece of code that exploits security flaws in computer systems and replicates itself. Each replicated code then tries to make multiple copies of itself and each of these copies then again tries to exploit the flaws. The various categories of worms are:

Monomorphic: These are the worms whose form remains the same when it replicates. As their signature remains the same, these are the easiest type of worms to be detected.

Oligomorphic: The code of such kind of worms comprise of two parts. The first one is the decryption function followed by the encrypted code. The decryption function of Oligomorphic worms may or may not change on each replication.

Polymorphic: The structure of these types of worms is same as that of Oligomorphic worms. The difference lies in the frequency of change of the decryption function which change every time the worm replicates.

Metamorphic: Unlike Oligomorphic and Polymorphic worms, these worms are not encrypted but recompile themselves with a different coding at each replication [3].

Some of the proposed automatic signature generation models to detect the above types of worms are:

1.) Autograph
2.) Earlybird
3.) Hamsa
4.) Nemean
5.) PADS
6.) Polygraph
7.) Honeycomb
8.) PAYL

The comparison of the above models based on the types of worms they can detect is shown in the figure below. The detailed description of the models can be found in [2].

Model/ Worm	*Monom orphic*	*Oligom orphic*	*Polym orphic*	*Metam orphic*
Autograp h	**Yes**	**No**	**No**	**No**
Earlybird	**Yes**	**Yes**	**No**	**No**
Hamsa	**Yes**	**Yes**	**Yes**	**No**
Nemean	**Yes**	**Yes**	**No**	**No**
PADS	**Yes**	**Yes**	**Yes**	**No**
PAYL	**Yes**	**Yes**	**Yes**	**No**
Polygraph	**Yes**	**Yes**	**Yes**	**No**
Honeycom b	**Yes**	**No**	**No**	**No**

Table 1. Comparison of Signature Generation Models

Existing automated signature generation models detect and generate signatures for monomorphic, oligomorphic and polymorph worms. Currently, Autograph and SweetBait (which uses Honeycomb) detect monomorphic worms. Earlybird and Nemean detect oligomorphic worms. PADS, PAYL, Polygraph and Hamsa detect polymorph worms (see Table 1). All these systems focus on the worm content, except Earlybird, which takes into account the address dispersion by counting the number of connections on different hosts (i.e. different IP addresses). Out of the seven models listed in the above section only three are able to detect polymorphic worms and none of them is able to detect metamorphic worms [5]. In the next paragraph, we will discuss one of those three models (PAYL) in detail.

The PAYL model [1] is based on the principle that zero-day attacks are delivered in packets whose data is unusual and distinct from all prior normal content flowing to or from the victim node. For each node one has to create a normal profile by training the system according to the packets flowing through the node. This involves machine learning concepts. The implementation of PAYL involves two phases:

Training Phase: The training profile for PAYL is built by computing the frequency of each character in the packet. Based on this a normal packet profile is generated for a site. The information (average frequency per packet) is used to find the deviation of a new packet in the generation phase. Classification of the packets is done on the basis of payload length (l) to form clusters. Along with this destination service port (p), destination IP, average byte frequency and standard deviation are also maintained. Hence, the maximum space required is in the order of p*l*k where k is the constant amount of space required to store average byte frequency and standard deviation. To reduce the search time, the Manhattan distance between the clusters is computed and if the distance is below a particular threshold the clusters are combined. The performance of the system heavily depends on the amount of time dedicated to the training phase since the profile created will then be closer to the actual traffic profile of the node.

Generation Phase: For generation of signatures PAYL use ingress-egress packet co- relationship. Firstly, the profile parameter values are computed for incoming packets and then compared to model values: a significant difference from the norm causes the packets to be moved into a suspect pool. For comparison, PAYL uses the Mahalanobis distance, which takes into account not only the average value but also its variance and the covariance of the variables measured (hence better than Euclidian distance). It is also called quadratic distance, which measures the separation of two groups of objects.

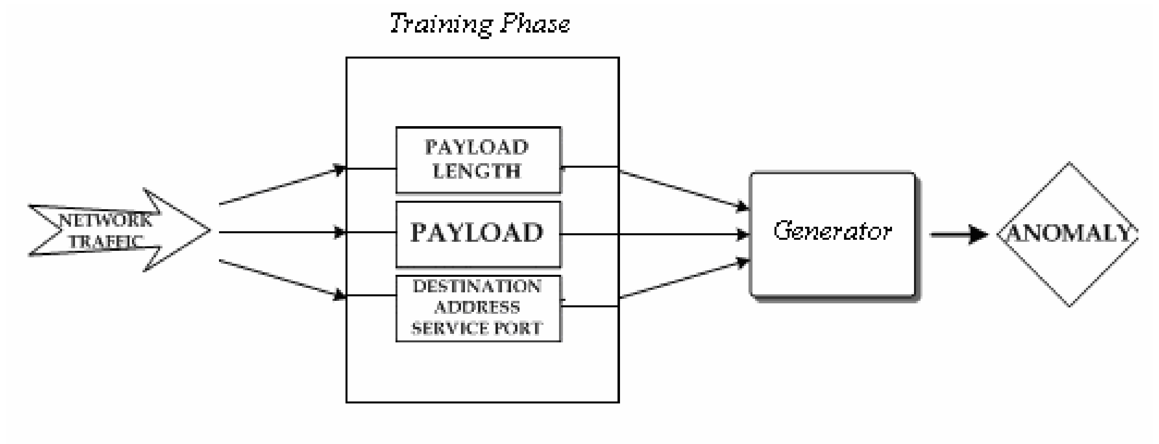

Figure 1. PAYL Architecture

The system makes use of the fact that a worm that has infected a node will try to propagate to other nodes in the network. So it monitors the outgoing traffic to detect any such propagating worm. If any outgoing packet has similar content as well as destination port number as that of a packet in the suspect pool then it is expected to be a propagating worm. For content matching PAYL makes use of longest common subsequence algorithm. LCseq has the advantage of being able to detect polymorphic worms but may introduce more false positives. Once the system detects the propagating worm, its signature generated and is stored in a database. The other nodes in the network can be intimated of the attack and other static Signature Detection Engines can use the signature.

3. Testing signature generation models

Before establishing the test cases, we need to understand how polymorphic worms change their form to evade detection and then try to propagate to other nodes in the network. There are various ways in which a worm tries to change its code and many more are continuously being invented by crackers. Here are some of these methods:

Worms may randomly pad their content. Example:

GET./default.ida?XXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXX XXXX

```
XXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX%u9090%u6858%ucbd3%u7 801%u9090%u6858%ucbd3%u7801%u 9090%u6858%ucbd3%u7801%u9090%
u9090%u8190%u00c3%u0003%u8b00 %u531b%u53ff%u0078%u0000%u0
```

Worms may change the order in which functions appear in the code. Example:

```
GET./default.ida? u9090%u8190%u00c3
%u0003%u8b00 %u531b%u53ff%u0078
%u0000%u0 %u9090%u6858%ucbd3%u7 801%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%
```

Worms may use encryptor and decryptor functions in the code. Example:

```
    Encryptor (){ //encryptor code}
GET./default.ida? XXXXXX%u9090%u6858
%ucbd3%u7 801%u9090%u6858%ucbd3
%u7801%u 9090%u6858%ucbd3%u7801
%u9090%u9090%u8190%u00c3%u0003
%u8b00 %u531b%u53ff%u0078%u0000%u0
    Decryptor (){//decryptor code}
```

4. Conclusion

To establish the success of any model extensive testing needs to be done. One of the ways to do this is to implement the model on your network and open it to the network traffic. However, the model may be improperly trained based on how much the traffic to your network mirrors the actual traffic to other networks in general. An alternative and much more standard way is to use the 1999 DARPA dataset for testing purposes. The DARPA dataset was developed by the Information Systems Technology Group of MIT Lincoln Laboratory. The group has collected and distributed the first standard for evaluation of network intrusion detection systems. It provides both training and test data and the evaluation of most intrusion detection system is done on the basis this dataset. The above models should be analyzed on the parameters of false positives and false negatives based on this standard dataset.

References

[1] K. Wang and S. Stolfo, Anomalous payload- based network intrusion detection, Proceedings of Recent Advance in Intrusion Detection (RAID), Sept. 2004.

[2] Attack Detection and Signature Generation, Sixth Framework Program, European Network of Affined Honeypots, May 2006.

[3] Péter Ször and Peter Ferrie, Hunting For Metamorphic, Symantec, White Papers.

[4] Roberta Bragg, Mark Rhodes-Ousley, Keith Strassberg, Complete Reference-Network Security, Edition 2004, Tata McGraw-Hill.

[5] Sébastien Chainay, Karima Boudaoud , A Model for Automatic generation of behavior based worm signature.