Traffic and Attack Pattern Analysis for Multiagent Distributed Intrusion Detection System

Grzegorz Kołaczek Krzysztof Juszczyszyn

Institute of Information Science and Engineering, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław , Poland

Abstract

The paper proposes an attack pattern ontology and formal framework for network traffic anomalies detection within a distributed multiagent Intrusion Detection System (MUDIDS) architecture. The role of traffic anomalies detection is discussed, then it has been clarified how some specific values characterizing network communication can be used to detect network anomalies caused by security incidents (worm attack, virus spreading). Finally, it has been defined how to use the proposed techniques in distributed IDS using attack pattern ontology.

Keywords: Ontology, Intrusion detection

1. Introduction

In order to process intrinsically distributed information, most of modern IDS systems are organized in a hierarchical architecture [4], consisting of low level nodes which collect information and management nodes which aim to detect large-scale phenomena. The task of management nodes is to reduce the amount of the data processed, identify attack situations as well as make decisions about responses [10].

In our approach it is assumed that the network system consists of the set of nodes. There are also two types of agents in our multiagent system: monitoring agents (MoA) and managing agents (MA)[7]-[9]. Monitoring agents observe the nodes, process captured information and draw conclusions that are necessary to evaluate the current state of system security within their areas of responsibility. Managing agents are responsible for gathering information from MoA agents and generating reports about global threats and ongoing attacks. Each agent MoA monitors its own area of responsibility consisting of the set of network nodes.

It is commonly known that in the case of worm attack there occur at least two kinds of anomalies: in observed traffic characteristics and in communication scheme which tends to be constant under normal conditions. In this context the system properties observed by the agent MoA in the proposed architecture will fall into two basic (and physically different) categories traffic measurement, communication pattern measurement. The attack recognition is being made on the basis of them.

The MoA agent's algorithm for decision making process is invoked periodically and uses observed values as input data. MoA also stores acquired values thus creating the history of system behavior.

2. Network traffic anomalies and intrusion detection

Intrusion detection systems (IDS) have been proposed as an approach to cope with current security problems. The aim of the intrusion detection is discovering of all abnormal states of the system in relation to the network traffic, users activity and system configuration that may indicate violation of security policy [1], [2]. But although the IDS idea is very simple, implementation of such systems has to deal with a lot of practical and theoretical problems. Difficulties with building intrusion detection systems arise from a complexity of the structure of attacks symptoms, distributed nature of the network systems and dynamics of the source of threats especially the problems of encoding new intrusions scenarios. The security assessment of a network system requires application of complex and flexible mechanisms for monitoring values of system attributes that have an influence on the security level of all network system. Another important element is an effective computational mechanism for evaluating the states of system security on the basis of incomplete, uncertain and inconsistent resources. Finally, the algorithms of machine learning to detect new intrusions pattern scenarios and recognize new symptoms of security system breach in order to update the security system knowledge base must be defined.

3. Evaluation of network traffic anomalies

Traffic attributes that are especially important (because their rapid change during typical attacks) and used during process of anomaly detection are [1]:

- source and destination IP address,
- source and destination port,
- number of bytes and packets sent to the remote hosts,
- number of bytes packets received by the local host,
- TCP flags, especially SYN, RST and FIN flags
- duration of the connection

The values of variables describing these attributes are collected and processed by intrusion detection system in a purpose to identify any anomalous behavior. The simplest decision mechanism applied in intrusion detection system uses threshold test to find out if the observed value is typical or it can be classified as anomalous. This preliminary observations will be then used in metadata-based detection environment in order to reason about network attacks.

3.1. Network traffic related variables used in evaluation

In our approach we observe: source/destination IP and port number, number of bytes sent/received and ration of number of SYN packets to FIN packets. These attributes were selected because a significant number of security incidents like denial of service attacks (DoS and Distributeid DoS), worm attacks, scanning cause changes in their values and so it could be recognized as an anomalous state. For example intrinsic nature of DoS/DDoS or intrusive system scan attacks makes that existing in the normal state of the system communication patterns must be effected by these events [8]. Communication patterns are related to the attributes like IP address of source/destination host or port number of the required network service. Similarly, other attacks like worm, alpha or flash crowd will also have an effect on different traffic related attributes like average duration of the connection or average number of bytes sent by a host [12].

Raw data obtained as a result of above mentioned network traffic parameters observation must be transformed to get some useful information that can be used to identify the deviation between the current system’s state and another state that is supposed to characterize the normal system behavior. In the following sections we describe our approach to transformation of traffic related attributes values.

3.2. Source/destination IP address and port number

To measure changes in IP address and port number space we will observe a value of Shannon entropy related to these attributes [13]. Entropy values are calculated for separate time periods. The length of the period can be a subject of more detailed discussion [1], however we assume that it is possible that different monitoring agents (MoA) use various periods length.

This means that we will evaluate, collect and investigate the following network variables:

- S_IP (ti) - entropy of source IP address in the period t_i,
- D_IP(ti) - entropy of destination IP address in the period t_i,
- D_Port(ti) - entropy of destination port number in the period t_i ,
- S_Port(ti) - entropy of source port number in the period t_i .

Entropy value is evaluated from standard formula:

$$e = -\sum_{i=0}^{N} p_i \log p_i , \quad p_i = \frac{n_i}{\sum_{i=0}^{N} n_i} , 0 \leq i \leq N , \qquad (1)$$

where:

N - cardinal number of IP address/port number set,

n_I - number of packets with a particular source/destination IP address/port number observed in the period $t_{I,}$

$\sum_{i=0}^{N} n_i$ - total number of packets observed in the period t_I

As for some $t_{i\,,}$ the value of $\sum_{i=0}^{N} n_i$ can be equal to zero (no traffic observed in t_i period), we assume that in these periods entropy value is also zero.

Any untypical changes of variables values related to IP address or port number entropy can be treated as a sign of anomalous behavior of the monitored system. Especially we can assign some threshold value which will indicate the state of anomalous entropy level. E.g. AS_IP will be a constant describing the value of acceptable S_IP level

3.3. Number of bytes and packets

Changes of entropy values are strictly related to changes of communication patterns. Using this measure of traffic parameters, some sort of anomalies caused by intrusive actions like DoS or system scan can be detected. However, other types of intrusions do not have to disturb communication patters. For example so called topological worms using internally generated target lists tries to infect only well known by the infected host remote targets. Well known, means that instead of performing random scan to find vulnerable hosts, the worm tries to discover the local communication topology and infect only hosts which sent or received data to or form infected host [14].

Like it has been shown in section 3.2 the values describing number of bytes and packets exchanged by a host will be obtained as a result of observation of incoming and outgoing traffic in each of constant size period while it is observed by MoA.

`TRAFFIC_B_R(ti)` - bytes received by a host in period t_i

`TRAFFIC_B_S(ti)` - bytes sent by a host in period t_i

`TRAFFIC_P_R(ti)` - packets received by a host in period t_i

`TRAFFIC_P_S(ti)` - packets sent by a host in period t_i

Also a traffic threshold value can be assigned and described by e.g. `ATRAFFI_B_R, ATRAFFI_B_S, etc.`

3.4. TCP flags

The TCP flags are important source of information about host's connections state. Typical TCP connection have three phases: connection establishment, data transfer, connection termination. Each phase uses packets with some standard sequences of TCP flags, especially TCP flags brings information about current connection state. However, this information may be incorrect while an intruder can manipulate the packet's content to reach some particular aim (e.g. the intruder tries to obtain information about services activated by host by performing system scan or simmilar effect can be observed during DoS/DDoS attacks) [10].

In our approach we measure a difference between number of sent SYN packets and received RST and FIN packets.

TCP_FLAG = $p_{t_i}^{syn} - p_{t_i}^{rst} - p_{t_i}^{fin}$

where:

TCP_FLAG - parameter indicating temporal start/end connection ratio

$p_{t_i}^{syn}$ - number of sent TCP packets with SYN flag set,

$p_{t_i}^{rst}$ - number of received TCP packets with RST flag set,

$p_{t_i}^{fin}$ - number of received TCP packets with FIN flag set

In normal conditions, in long time observation we should get the mean value of TCP_FLAG near zero. Intrusive actions like system scanning, DoS attacks, may cause the temporal distortion of the mean value of TCP_FLAG

3.5. Duration of the connection

Duration of a connection may be another characteristic attribute in anomaly detection process [1]. During various types of attacks, this value will be affected and so an anomaly may be detected. For example worm infection will generate a large number of connections with similar duration. This worm related connections should change also the observed mean values of connection duration that has been observed in a system. We evaluate simple mean value of connections' duration that have been observed in period t_i. c_{t_i} - mean value of duration of connections that have been observed in a period t_i.

4. Traffic statistics

In section 3, a few traffic related variables have been presented. Values of these variables can be used to obtain useful information about system security incidents. Apart from collecting these values, intrusion detection mechanism must preprocess them to reduce the probability of misinterpretation and so called false-positive alarm.

Our approach uses Mark Burgess (MB) technique to find out anomalous behavior. This technique of anomaly detection has been described in [2], [3]. The main assumptions made in his framework are as follows.

MB defines *iterative expectation function*. Let q be an observation, and $<<q_i>>$ be the i-th estimator of the average, with geometric fall-off, then $<<q_i>>$ may be defined by the recurrence relation:

$$<<q>>_{i+1} = (q \mid <<q>>_i),\ <<q>>_0 = 0 \qquad \textbf{(2)}$$

where

(1)

$$(q_1 \mid q_2) = \frac{wq_1 + \overline{w}q_2}{w + \overline{w}} \quad , \ w, \overline{w} \text{ - const} \qquad \textbf{(3)}$$

The other fundamental notion for MB analysis is pseudo-periodic function:

$$q(t) = \sum_{n=0}^{\infty} q(nP + \tau) \equiv \sum_{n=0}^{\infty} \chi_n(\tau), \tag{4}$$

where $0 \leq \tau < P$

Such pseudo-periodic function can be characterized by two kinds of average: average over corresponding times in different periods (topological average $< \chi(\tau) >_T$), and average of neighboring times in a single period (local average $< \chi(\tau) >_P$).Limited memory versions of these deviations are given by the following formulas:

$$\sigma_{<<T>>}(\tau) \equiv \sqrt{<<(\delta_{<<T>>}\chi)^2 >>_T} \tag{5}$$

$$\sigma_{<<P>>}(n) \equiv \sqrt{<< (\delta_{<<P>>}\chi)^2 >>_P} \tag{6}$$

where, for any measure X:

$$(\delta_{<<P>>}X) \equiv X - << X >>_P \tag{7}$$

$$(\delta_{<<T>>}X) \equiv X - << X >>_T \tag{8}$$

These averages are calculated by replacing the evenly weighted sum over the entire history by an iteratively weighted sum that falls off with geometric degradation. The additional positive consequence of this definition is that in order to obtain all information, one only needs to retain and update the mean and the variance.

In contemporary network, traffic congestion is avoided by packet switching. The traffic has been isolated to 'parallel' branches of a network spanning tree. Network nodes or hosts occupy points at the leaves of these branches and therefore experience an individual (subjective) view of the network traffic. The concept of an anomaly is also a very subjective one because what is unusual for one node is a regular occurrence for another. One of the best places in the network where incidents may be tracked down and so anomalies may be reveal are the network nodes.

As stated above, anomalousness is a subjective judgment, made within the context of past experience, and can be codified into a 'policy' about what is sufficiently anomalous to warrant a response. So, we look for a potential anomalous behavior by comparing current observation to learned experience. If the event looks probable, we can consider it as the evidence derived from a supporting semantic model. As in our approach a Monitoring Agent is responsible for interpretation of the data stream arriving to the particular node, an overall situation assessment must be based on a set of communicates concerning the the traffic-related variable (measured in network nodes) coming from monitoring agents (MoA) and gathered by the managing agent (MA).

5. Attack pattern ontology

We postulate the following generic form of communicate about network variables:

$\mathtt{MoA_1(N_1,V_1)=}\ x$, where $x \in [0,1]$

Which should be read: „*Monitoring agent MoA_1 states, that the value of network variable V_1 measured in node N_1 is normal (i.e. characteristic for the absence of attack) with probability x*". It is also assumed, that for any $\mathtt{V_1}$ exists some threshold value $\mathtt{A_i}$, such that any value $\mathtt{MoA_1(N_1,V_1)< A_i}$ means that we experience an abnormal (suggesting that there's an attack) value of $\mathtt{V_1}$.

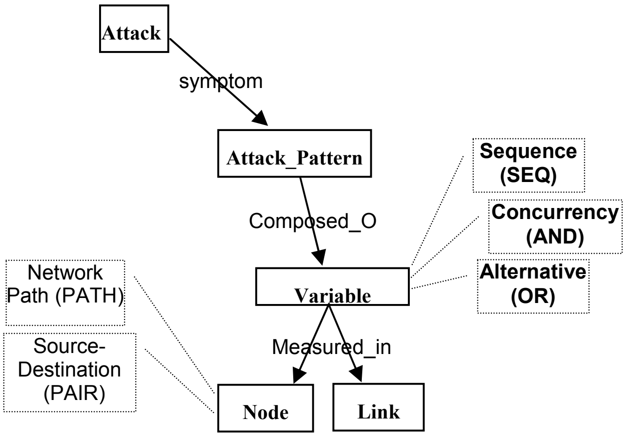

Fig. 1: Core attack pattern ontology.

Moreover, the core ontology containing basic concepts for defining attack patterns is proposed in order to help in defining attacks and to simplify network variable-based computations. The ontology contains basic concepts like *Attack* which is characterized by certain *Attack_Pattern* (Fig.1), which in turn is defined by certain set of observations of the network variables. As mentioned above, the observations are given in form of MoA's communicates about probability of anomalous variable value. Our ontology contains also specific operators which allow to define a *sequence* (SEQ) of communicates, their *concurrency* (AND) or *alternative* (OR). It is also possible to consider *paths* (i.e. sequences of nodes) in network graph (PATH) and origin-destination pairs of network nodes (PAIR).

Now we may define simple network attack, which will help us to illustrate how to use the attack pattern ontology.

6. Attack pattern definition

Let us consider so-called *Reflector Attack* which takes place according to the following scheme:

1. An attacker prepares the attack by compromising several vulnerable hosts which create a network of so called *"zombie" hosts*.
2. The attacker initiates the attack and orders all *"zombie"* hosts to send spoofed SYN packets with the source address set to the victim's IP address to an agent ("reflector*"*) host.
3. The agent (*"reflector"*) host responds to this SYN packet by sending a SYN|ACK
   or a RST packet to the source address, which is actually the victim's IP address.
4. The victim replies with RST packets to reflector's SYN|ACK packets and with no packet to reflector's RST packets
5. The *"zombies"* send a continual storm of theses packets, thus causing the victim host to be flooded by innocent agent host (*"reflector" host*).

With big number of Reflectors, the Target is down in a short time.

All this activity has obvious influence on network variables being measured. But now we may define Reflector Attack using our ontology:

```
DEF_ATTACK (Reflector_Attack)
N1, N2, N3: Node;
//where N1-zombie, N2- victim, N3-
reflector
EXISTS PATH(N1,N2) SUCH THAT:
(
SEQ (
// the sequence of the attack
(                                   //
attack symptoms
MoA(N1, TCP_FLAG)<AFLAG_SYN
AND
MoA(N1, D_IP)<AD_IP
AND
MoA(N1, S_IP)<AS_IP
AND
MoA(N1, D_PORT)<AD_PORT
),
FOR ANY N3 in PATH(N1,N2)
// symptoms at "reflector" nodes
(
MoA(N3, TCP_FLAG)<ATCP_FLAG
AND
MoA(N3, D_PORT)<AD_PORT
),
(
// symptoms at "victim"
MoA(N2, TCP_FLAG  )<ATCP_FLAG
AND at
MoA(N2, TRAFFIC_B_R)<ATRAFFI_B_R
)
)
```

In the above definition we use earlier defined the following variables:

```
D_IP, S_IP
D_PORT
TRAFFIC_B_R
TCP_FLAG
```

Note, that any single observation (like: `MoA(N1, TCP_FLAG)<TCP_FLAG)` doesn't have to (and typically does not) imply that we are experiencing an attack. But taking them together we see that given pattern of observations clearly suggests known type of the attack.

7. Reasoning about attacks

Our distributed intrusion detection system recognizes and alarms about security events according to MoA observations and MA decisions. The accurateness of final IDS decision depends on MoAs evaluation of observed values and MA ability to correctly recognize attack patterns using data delivered by MoA and attack pattern ontology.

Reasoning about attack that is performed by IDS can be described by the following procedure.

1. Each MoA observes and evaluates a set of variables described in sections 3.2 – 3.5. During this step MoA updates variables values so they represents the current system state. Variable list obtained by MoA$_1$ may be similar to the following example:
   - TCP_FLAG= 143
   - D_IP= 3,21
   - S_IP= 4,81
   - D_PORT= 1,98
   - TRAFFIC_B_R= 3962
2. During next step MoA estimates the abnormality level of collected values. After this step MoA will be able to present its opinions about nodes states in a form of attack probabilities presented at the beginning of the section 5.
   - MoA$_1$(N$_1$,TCP_FLAG)= 0,143
   - MoA$_1$(N$_1$,D_IP)= 0,21
   - MoA$_1$(N$_1$,S_IP)= 0,81
   - MoA$_1$(N$_1$,D_PORT)= 0,28

   where N$_1$ – the node observed by MoA$_1$

   The probability values related to MoA observations are estimated with application of statistics presented in section 4. In general, the attack probability is greater, the current observation is more far from the historical records.
3. Third step performed by MoA is a comparison of current probability attack value with corresponding threshold value. For example:

- MoA$_1$(N$_1$,TCP_FLAG)<0,05
- MoA$_1$(N$_1$,D_IP)<0,01
- MoA$_1$(N$_1$,S_IP)<0,5
- MoA$_1$(N$_1$,D_PORT)<0,3

As the result, MoA gets some binary vector:

TRAFFIC B R	FLAG SYN	D IP	S IP	D PORT
0	0	0	0	1

Where ‘0’ value in a vector means ‘normal state’ and ‘1’ stands for “annomaly”.

4. Next step is performed by a MA. The MA collects and processes binary vectors obtained from MoAs. The MA compares vectors to the known attack patterns.

   For example the MA possess the folowing list of MoA binary vectors:

MoA(N$_1$,XXX)

TRAFFIC_B_R	FLAG_SYN	D_IP	S_IP	D_PORT
1	1	0	0	0

MoA(N$_2$,XXX)

TRAFFIC_B_R	FLAG_SYN	D_IP	S_IP	D_PORT
0	1	0	0	1

MoA(N$_3$,XXX)

TRAFFIC B R	FLAG SYN	D IP	S IP	D PORT
0	1	1	1	1

MoA(N$_4$,XXX)

TRAFFIC B R	FLAG SYN	D IP	S IP	D PORT
0	1	0	0	1

MoA(N$_5$,XXX)

TRAFFIC B R	FLAG SYN	D IP	S IP	D PORT
0	0	0	0	1

Comparing the binary vectors to attack patern defined in section 6 MA recognizes the presense of reflector attack where node N$_3$ plays a role of zombie node, node N$_4$ and N$_2$ play roles of reflectors and node N$_1$ is a victim.

We consider only situation where exist exact mapping between MoA binary vectors and attack patern. However, it is possible that some attacks may be unnoticed using such method. First thing is that MoA may misjudge some observation and as a result it doesn’t report about ubnormal state to the MA.

Second possible situation is that MA received a few different values of binary vectors from several indipandant MoAs that describe the same node N in the network. This unconsistency must be solved by MA otherwise it could not generate the final decision about the network security state.

Third scenario is that there may be several parralel attacks or ubnormal sistuations in a corresponding node and this may also produce some unexact or inonsistent final results.

The resoning about security events in theses three scenarios should also be considered and we plan to enhance our proposal with corresponding elements during further steps of our work

8. Conclusions

In the paper we presented a new method for traffic anomalies detection based on Mark Burges statistics and the attack pattern ontology. As Mark Burgess technique has quite good ability to tolerate seasonal changes, do not require regularized data and requires relatively small set of data and utilizes CPU only on low level we hope that all this features will characterize also our proposal. These features are especially interesting in a context of real time identification performed on a single host and within mobile agent environments. Another important outcome of our work is the application of attack pattern ontology within a process of intrusion detection. The attack ontology allows us to efficiently combine observation coming from different sources (MoAs) and to draw final conclusion about current network security level.

Acknowledgement

This work is partially supported by the Polish State Committee for Scientific Research under (Grant No. 3 T11C 029 29).

References

[1] A. Beach, M. Modaff, Y.Chen, Network Traffic Anomaly Detection and Characterization. cs.northwestern.edu/~ajb200/anomaly%20detection%20paper%201.0.pdf.

[2] M. Burgess, An Approach to Understanding Policy Based on Autonomy and Voluntary Cooperation. *DSOM,* pp. 97-108, 2005.

[3] M. Burgess, Two Dimensional Time-Series for Anomaly Detection and Regulation in Adaptive Systems. *DSOM,* pp.169-180, 2002.

[4] V. Gorodetski, O. Karsaev, A. I. Khabalov and Kotenko, et.al., Agentbased model of Computer Network Security System: A Case Study. *Proceedings of International Workshop Mathematical Methods, Models and Architectures for Computer Network Security, Lecture Notes in Computer Science,* pp. 39-50, 2001.

[5] K. Hwang, H. Liu, Y. Chen, Cooperative Anomaly and Intrusion Detection for Alert Correlation in Networked Computing Systems.

Technical Report, USC Internet and Grid Computing Lab (TR 2004-16) , 2004

[6] T.M. Khoshgoftaar, M.E. Abushadi, Resource-sensitive intrusion detection models for network traffic. *Eighth IEEE International Symposium on Publication*, pp. 249- 258, 2004.

[7] K. Juszczyszyn, N.T. Nguyen, G. Kolaczek and A. Grzech, et.al, Agent-based Approach for Distributed Intrusion Detection System Design. *International Conference on Computational Science 2006*, pp.224-231, 2006.

[8] K. Juszczyszyn, G. Kołaczek, Assessing the Uncertainty of Communication Patterns in Distributed Intrusion Detection System. *KES 2006, LNAI 4252,* pp. 243-250. 2006.

[9] G. Kolaczek, A. Kuchtiak-Pieczynska, K. Juszczyszyn and A. Grzech, et.al, A Mobile Agent Approach to Intrusion Detection in Network Systems. *Lecture Notes in Artificial Intelligence,* 3682:514-519, 2005.

[10] I. Kotenko, et al., Multi-Agent Modeling and Simulation of Distributed Denial-of-Service Attacks on Computer Networks. *Proceedings of Third International Conference Navy and Shipbuilding Nowaday. St. Petersburg*, pp. 38-47, 2003.

[11] M. Thottan, C. Ji, Anomaly detection in IP networks. *IEEE Transactions on Signal Processing,* 51(8): 2191- 2204, 2003.

[12] A. Lakhina, M. Crovella, C. Diot, Characterization of Network-Wide Anomalies in Traffic Flows. *Technical Report BUCS-2004-020, Boston University*, http://citeseer.ist.psu.edu/715839.html, 2004

[13] C.E. Shannon, W. Weaver, The mathematical theory of communication, *University of Illinois Press, Urbana*, 1949.

[14] N.Weaver, V. Paxson, S.Staniford and R. Cunningham, A taxonomy of computer worms. *ACM Workshop on Rapid Malcode - WORM '03, ACM Press, New York, NY,* pp. 11-18, 2003.