Indian Journal of Science and Technology, *Vol 10(14), DOI: 10.17485/ijst/2017/v10i14/106786, April 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Model for Generating Synthetic Network Flows and Accuracy Index for Evaluation of Anomaly Network Intrusion Detection Systems

C. Madhusudhana Rao[1]* and M. M. Naidu[2]

[1]Department of Computer Science and Engineering, Sri Venkateswara University (SVU) College of Engineering, Tirupati – 517502, Andhra Pradesh, India; masura_c@yahoo.com
[2]School of Computing, Veltech Dr. RR and Dr. SR University, Avadi, Chennai – 600062, Tamil Nadu, India; mmnaidu@yahoo.com

Abstract

Objectives: This study proposes a model for generating synthetic network flows inserting malicious fragments randomly and a new metric for measuring the performance of an Anomaly Network Intrusion Detection System (ANIDS). **Method:** A simulation model is developed for generating synthetic network flows inserting malicious fragments that reflect Denial of Service (DoS) and Probe attacks. An ANIDS shall maximize true positives and true negatives which is equivalent to minimizing Type-I and Type-II errors. The geometric mean of True Positive Rate (TPR) and True Negative Rate (TNR) is proposed as a metric, namely, Geometric Mean Accuracy Index (GMAI) for measuring the performance of any proposed ANIDS. **Findings:** The task of detecting anomalous network flows by inspecting at fragment level boils down to discrete binary classification problem. The Receiver Operating Characteristic (ROC) curve considers False Positive Rates (FPR) and True Positive Rate (TPR) only. It does not reflect the minimization of Type-I and Type-II errors. Maximizing GMAI is the reflection of minimizing 1-GMAI which is equivalent to minimizing Type-I and Type-II errors. Further, the GMAI can be employed as service level for evaluating acceptance sampling based ANIDS. The domain of DoS and Probe attacks, mostly employed by the intruders at fragment level is studied. A conceptual simulation model is developed for generating synthetic network flows incorporating malicious fragments randomly from the domain of DoS and Probe attacks. The conceptual model is translated into operational model (a set computer programs) and synthetic network flows are generated. Using the operational model, the 1000 synthetic network flows are generated for each percentage of anomalous flows varying from 0.1 to 0.9 and employing discrete uniform probability distribution for selecting a fragment for transforming it into malicious. The generated network flows for each percentage of anomalous flows are represented graphically as histogram. It is found that they follow discrete uniform distribution. Hence, the model is validated. **Applications:** The simulation model can be used for generating synthetic networks flows for evaluating ANIDS. The GMAI can be used as service level for evaluating a discrete binary classifier irrespective of domain.

Keywords: Anomalous Flows, Geometric Mean Accuracy Index, Network Intrusion Detection Systems, Synthetic Network Flows, Simulation Model

1. Introduction

Significant, sensitive and proprietary data of a business organization, in storage and transit, are vulnerable for intrusion as internet is used widely nowadays. An intruder undertakes a set of actions that forces to comprise confidentiality, availability and integrity of computer and network resources.

The purpose of deploying an Anomaly Network Intrusion Detection System (ANIDS) is to identify anomalous patterns in network flows that cause damage to computer and network resources. Predominantly,

**Author for correspondence*

Receiver Operating Characteristic (ROC) Curve is employed for performance evaluation of an ANIDS. This study proposes a new index referred to as Geometric Mean Accuracy Index (GMAI) which proves to be more appropriate than ROC curve. Further, the Expected Opportunity Cost (EOC) of a classifier is also formulated and suggested that GMAI can be used as surrogate to service level to select the ANIDS.

The availability of real-life and benchmark data sets for network flows for performance evaluation of ANIDSs is limited. Further, the applicability of such data sets suffers from insufficiency of attack information. Hence, there is a need for synthetic data sets for network flows at fragment level. This study proposes a model for generating such data sets.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 reviews Real-life and Benchmarks datasets. Section 4 proposes a model for generating synthetic network flows; Section 5 proposes a new performance metric, GMAI and an Opportunity Cost model for evaluating ANIDS. Section 6 presents conclusion of the study.

2. Review of Network Security Issues

The Transmission Control Protocol/Internet Protocol (TCP/IP) suite provides data transmission services for applications on internet. An intruder strives to gain unauthorized access to a system violating network security with the intension of interfering with system Confidentiality, data Integrity and Availability (CIA). The state of keeping information, in storage or transit, understandable to intended receivers only is referred to as confidentiality of information. Availability of information refers ensuring legitimate users to access information when needed. The ability of detecting any alteration of information, in storage or transit, is referred to as integrity. The act of ensuring a creator / sender of the information to comply with the same at later time is referred to as non-repudiation. The conformity of identities of sender and receiver with each other and the origin/destination of the information is referred to as authentication.

Cryptography is the science and art of protecting information by *encrypting* it into cipher text. Only with a secret *key* can *decrypt* the message into plain text. The primary objectives of cryptography are to: maintain confidentiality, ensure availability, preserve integrity, enforce non-repudiation and grant authentication.

Cryptography cannot monitor and analyze network traffic data, user activities such as failed login attempts guessing password with invalid / valid User ID, attempts to use privileges that have not been authorized andsystem activities such as system resources utilization, hardware policy violations etc. Further, it cannot protect against transfer of virus.

The use of firewalls to monitor the network traffic data for allowing it subjected to specific rules. However, a firewall fails to provide protection against malicious insiders, new threats and transfer of virus[1]. An Intrusion Detection System (IDS) that is able to identify attacks against vulnerable services and applications, privilege violations, unauthorized logins and access to sensitive files. An IDS is a dynamic monitoring system whereas a firewall is a static monitoring system[2].

An IDS is primarily classified[3] into Host-based Intrusion Detection System (HIDS) and Network-based Intrusion Detection System (NIDS). An HIDS, deployed at host level, collects data from log files and verifies login, logoff, and modification of data. It also verifies access information to system resource like files/ memory/registry. NIDS, deployed at network level, verifies network traffic data by observing packet information. It also collects information from network Management Information Base (MIB).

Based on the detection mechanism, the NIDSs are classified into three types as given hereunder:

1. Misuse-Based NIDS (MNIDS)

It is also known as signature-based NIDS that employ a set of rules for detecting known attacks.

2. Anomaly-Based NIDS (ANIDS)

It attempts to detect unknown attacks, abnormal patterns in network traffic data not matching with expected normal patterns.

3. Hybrid NIDS (HNIDS)

It exploits the merits of the above two approaches for detecting both known and unknown attacks.

As this study considers ANIDS only, its brief description is presented here referring the generic view of its architecture[4] shown in Figure1. An anomaly detection engine comprises pre-processing and matching mechanism as software components. It receives a packet from internet as input through pre-processing module and outputs an alarm on detecting anomalous pattern of the packet. The pre-processing module captures a packet and organizes its data suitable for matching mechanism as input. The classification algorithm of matching mechanism detects the anomalous behavior of a packet making use of the configuration and reference data. The alarm forms the basis for intrusion mitigation, monitoring, and management; and updating configuration and reference data. The information about known intrusion signatures or profiles of normal behavior is stored as reference data whereas the intermediate results are stored as configuration data. The analysis and interpretation of alarm information for diagnosing actual attacks is referred to as post-processing. The post-processing is carried by human analyst and post-processing module. Whenever new intrusions are known as a result of post-processing, the security manager updates intrusion signatures.

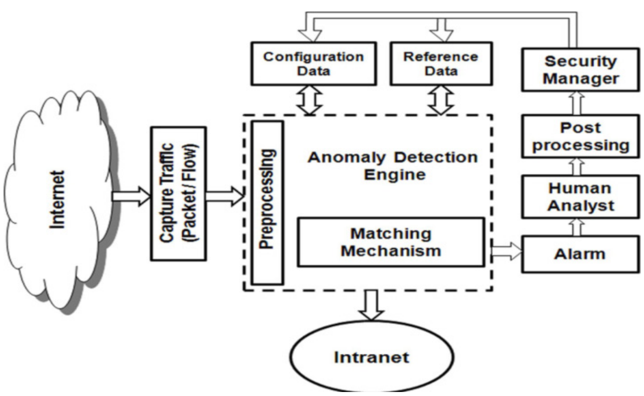

Figure 1. ANIDS architecture.

The network attacks detected by ANIDS are classified[5] as:

1. Remote to Local Attack (R2L)
The intruders send packets to system over a network and make use of vulnerability to gain local access.

2. Probing Attack (PROBE)
The intruder attempt to gather information about systems i.e., how many and what types of systems are on the network and what services the system supports for the purpose of intrusion.

3. Users to Root Attack (U2R)
The intruder access to a normal user accounts on the system exploits weak or mis-configured system security polices, operating systems log files to gain access to root.

4. Denial of Service Attack (DoS)
These attacks prevent legitimate users accessing a host or using network resources / services.[6] Grouped the attacks as shown in the following Table 1.

Table 1. Attack categories

DoS	Probe	U2R	R2L
Land, Teardrop Fraggle, Xmas, Rose, Winnuke, nestea Neptune, Ping Of Death, Smurf,Apache2, Back	Host scan, mscan nmap, saint satan, Port scan	Perl, xterm, eject ffbconfig fdformat ps, loadmodule	dictionary ftp-write guest phf xlock xsnoop

The DoS attacks consume prohibitively high level of resources such as link bandwidth, CPU processing power and memory storage leading to disruption of services. In 2013, it was found[7] that 60 percent of companies were affected by DoS attacks more than once with the probability of 0.87. By 2015, 73 percent of companies are affected by DoS attacks. The tools for detecting and mitigating DoS attacks at host and network level are provided. The demand[7] for such tools has grown up to 70% by 2015 out of which the demand for network level specific tools is around 78 %.

The description and impact of DoS and PROBE attacks is presented in the Table 2.Here, the Pareto principle is applicable as 95% of network traffic is handled by TCP/UDP protocols and the demand for network level specific tools for mitigating DoS attacks is 78%. Hence, it is felt appropriate to develop a model for generating synthetic network flows inserting DoS and PROBE attacks at fragment level. Such synthetic data are immensely useful for evaluating the effectiveness of ANIDS as the real-life data are scarce.

3. Review on Datasets

For evaluating the effectiveness and efficiency of ANIDS, appropriate datasets are required. It is preferable to employ benchmark or real-life datasets.

The data generated from simulated environments of several networks for different attack scenarios is referred to as benchmark data. The data collected over a period of time from real-life network traffic prone to attacks constitute real-life datasets[8].

The benchmark datasets are provided in DEFCON[9], CAIDA[10], LBNL[11], and KDDcup99[12] and NSL-KDD[13]. The DEFCON datasets reflects in limited way real-life network traffic. The CAIDA and LBNL datasets are heavily anonymized. The KDDcup99 and NSL- KDD datasets are not fragment-based datasets.

Table 2. Description and impact of DoS and PROBE attacks

S.No	Attack	Protocol	Attack Category	Attack Description	Impact
1	LAND	TCP	DoS	Source IP is spoofed as destination IP and it sends ACK to itself.	Destination is frozen
2	Xmass	TCP	DoS	URG, PSH and FIN flags are set	Destination is rebooted
3	Nestea	TCP	DoS	Fragment offset is changed	Operating system is crashed
4	Rose	TCP	DoS	Intentionally too small fragment is crafted which would be identified during packet reassembling at destination	Processor resources are exhausted
5	Win nuke	TCP	DoS	Sends fragment to destination port 139 over NetBIOS setting URG flag	Operating system is crashed
6	Tear drop	UDP	DoS	Fragment offset is changed	Operating system is crashed
7	Fraggle	UDP	DoS	Connection request is broadcasted spoofing victim's IP address as source IP so that all hosts which receive the request respond	Network and processor resources are exhausted
8	Port Scan	TCP/ UDP	PROBE	Spoofed source IP sends fragments varying destination ports	Vulnerable ports are found
9	Host Scan	TCP/ UDP	PROBE	Spoofed source IP sends fragments to different destination hosts	Vulnerable host are found

The Kyoto[14], MAWI[15], ISCX-UNB[16], UNIBS[17] are real-life datasets available for researchers. The Kyoto dataset is prepared from network traffic traces collected from a honey pot connected to an internet. By design, the honey pot capture unusual traffic and it does not reflect real-life scenario. For the context, fragment-based datasets are applicable whereas MAWI is a flow-based dataset. The ISCX-UNB and UNIBS are raw traffic traces but not attack data.

It is obvious that the intruders inserts DoS and PROBE attacks referred in Table 2 by changing one or more fragments of a network flow. The above datasets lack of requisite data for evaluating ANIDS in the context of DoS and PROBE attacks. Hence, it is motivated to develop a model for generating synthetic network flow datasets at fragment level that reflects real life traffic

4. A Model for Generating Synthetic Network Flows

This study proposes a model to generate synthetic network flows inserting DoS and PROBE attacks at fragment level that would be useful for evaluating the effectiveness of any proposed ANIDS by researchers.

A network flow is partitioned into a Number of Packets (NoP) and in turn each packet is divided into a number of fragments. Ultimately, a network flow is a set of fragments passing through an observation point in a network. The attributes of each fragment of a network flow are as follows:<Flow Number, Packet Identity, Fragment Identity, Source IP, Destination IP, Source Port , Destination Port, Protocol, Urgent Flag, Acknowledgement Flag, Push Flag, Reset Flag, Synchronous Flag, Final Flag, More Fragment Flag, Length of Fragment and Fragment Offset>.

Table 5. Input parameter assumptions

Parameter Notation	Parameter Description	Assumption
S	Cardinality of Source IP Set	100
D	Cardinality of Destination IP Set	50
MFS	Maximum Network Flow Size	500 KB
MPS	Maximum Packet Size	65536 bytes
MTU	Maximum Transmission Unit	1500 bytes
CSP	Cardinality of Source port Set	65536
CDP	Cardinality of Destination port Set	65536
P	Probability of a flow being anomalous	0.1 to 0.9

An intruder gets access to the fragments of network flow, in transition from a source to a destination, and inserts the malicious data in one or more fragments

Table 3. Normal fragments

Attack	SIP	DIP	SP	DP	Proto	FLAGS							length	FO
						U	A	P	R	S	F	MF		
Normal	X	Y	P	Q	TCP	0	1	0	0	0	0	1	EQ MTU	Multiples of offset value
Normal	X	Y	P	Q	UDP	0	0	0	0	0	0	1	EQ MTU	Multiples of offset value

Table 4. Anomalous fragments affected with DoS / PROBE attacks

Attack	SIP	DIP	SP	DP	Proto	FLAGS							length	FO
						U	A	P	R	S	F	MF		
Land	Y	Y	P	Q	TCP	0	1	0	0	1	0	0	40	0
Xmass	X	Y	P	Q	TCP	1	0	1	0	0	1	0	40	0
Nestea	X	Y	P	Q	TCP	0	1	0	0	0	0	1	EQ MTU	Not Multiples of offset value
Rose	X	Y	P	Q	TCP	0	1	0	0	0	0	1	LT MTU	Multiples of offset value
Winnuke	X	Y	P	Q =139	TCP	1	0	0	0	0	0	1	EQ MTU	Multiples of offset value
Port Scan	X	Y	P	Not Q	TCP	0	1	0	0	1	0	0	40	0
Host Scan	X	Not Y	P	Q	TCP	0	1	0	0	1	0	0	40	0
Tear drop	X	Y	P	Q	UDP	0	0	0	0	0	0	1	EQ MTU	Not Multiples of offset value
Fraggle	X	Broadcast Address	P	Q	UDP	0	0	0	0	0	0	0	40	0
Port Scan	X	Y	P	Not Q	UDP	0	0	0	0	0	0	0	40	0
Host Scan	X	Not Y	P	Q	UDP	0	0	0	0	0	0	0	40	0

transforming the flow as anomalous flow. The formats of normal and anomalous fragments affected with DoS / PROBE attacks of a network flow are presented Tables 3 and Table 4. A model is developed for generating synthetic anomalous flows. Its input parameters and data assumptions are given in Table 5 and Table 6. The overall processing logic of the model for generating synthetic network flows is given in Figure 2.Line 1 is to read the input viz., Maximum Number of Flows (MNF), S, D, CSP, CDP and P. The lines 2-5 initialize MFS, MPS, MTU and Flow Number (FN). The loop (while) between the lines 6 and 24 is for generating the normal / malicious fragments for each network flow. The lines 7 to 13 generate random variates that represent seven attributes of a network flow.

The function TCP-NORMAL-FLOW (F) generates TCP normal flows. Lines 1-2 computes NoP of maximum size and Residual Packet (RP) for the network flow size. The For loop in lines 3-9 generate fragment level network flow records for all packets of a flow and lines 10-16 generate fragment level records for RP. These network flow records are appended to the TCP Normal Flow file **tn** in Figure 3.

The For loop in lines 4-20 of the function TCP-ANOMALOUS-FLOW (F) generate fragment level anomalous network flow record if FrT is malicious else generate Normal network flow record and lines 21-38 generate for residual packet as shown in Figure 4.

The function TCP-ATTACK (T, F) calls corresponding attack function based on T. Each attack function inflicts the attack. The attributes of fragment level flow records for attacks are given in Table 5 and normal flow is given in the Table 6.The anomalous flow records are appended to the TCP anomalous Flow file **ta** in Figure 4.The pseudo code for TCP attacks is given in Figures 4(a) to 4(i).

```
SYNTHETIC-FLOWS ()
    Read MNF,S,D,CSP,CDP,P
    MFS←500
    MPS←65536
    MTU←1500
    FN←1
    while(FN<=MNF)
    SIP ← DURV (1,S)
    DIP ← DURV (1,D)
    SP ← DURV (1,CSP)
    DP ← DURV (1,CDP)
    Proto ← DURV (1,2)
    FS ← DURV (1,MFS)*1024
    FT← DER (P)
    F=< FN, SIP, DIP, SP, DP, Proto, FS,FT> ▷ Generated network flow
    if (Proto = 1)&& (FT=1)
    sf ←TCP-NORMAL-FLOW(F)
    if (Proto = 1)&& (FT=0)
    sf ←TCP-ANOMALOUS-FLOW(F)
    if (Proto = 2)&& (FT=1)
    sf ←UDP-NORMAL-FLOW(F)
    if (Proto = 2)&& (FT=0)
    sf ←UDP-ANOMALOUS-FLOW(F)
    Write sf
    FN←FN+1
```

Figure 2. Generation of synthetic network flows.

```
TCP-NORMAL-FLOW (F)
    NoP ← ⌊FS / MPS⌋ ▷ NoP of maximum packet size
    RP ← FS − (MPS * NoP) ▷ RP in bytes
    for i=1 to NoP
    NoF ← ⌊MPS / (MTU − 20)⌋ ▷ Number of fragments of size MTU
    RF ← (MPS − (MTU − 20) * NoF) ▷ Residual fragment in bytes
    for j=1 to NoF
    tn← <FN,i,j,SIP,DIP,SP,DP,Proto,0,1,0,0,0,0,1,1500,(j-1)*185>
    if (RF≠0)
    tn← <FN,i,j,SIP,DIP,SP,DP,Proto,0,1,0,0,0,0,0,(RF+20),(j-2)*185+RF/8)>
    if (RP≠0)
    NoRF ← ⌊RP / (MTU − 20)⌋ ▷ Number of fragments of RP
    RPRF ← (RP − (MTU − 20) * NoRF) ▷ Residual fragment of RP in bytes
    for k=1 to NoRF
    tn← <FN,i,k,SIP,DIP,SP,DP,Proto,0,1,0,0,0,0,1,1500,(k-1)*185>
    if (RPRF≠0)
    tn← <FN,i,k,SIP,DIP,SP,DP,Proto,0,1,0,0,0,0,0,(RPRF+20),(k-2)*185 +(RPRF/8)>
    Return tn
```

Figure 3. Generation of TCP normal flows.

Table 6. Data assumptions

Data Notation	Data Description	Assumption
FS	Network Flow Size	A random variate from discrete uniform distribution [1,MFS]
SIP	Source IP Address	A random variate from discrete uniform [1,S]
SP	Source Port	A random variate from discrete uniform [1,CSP]
DP	Destination Port	A random variate from discrete uniform [1,CDP]
DIP	Destination IP Address	A random variate from discrete uniform [1,D]
Proto	Protocol	A random variate from discrete uniform [1,2] 1 indicates TCP and 2 indicates UDP
FT	Flow type	A random variate from discrete empirical [0,1] 0 indicates anomalous flow and1 indicates Normal flow
FrT	Fragment type	A random variate from discrete empirical [0,1] 0 indicates malicious fragment and1 indicates Normal fragment

The function UDP-NORMAL-FLOW (F) generates UDP normal flows. Lines 1-2 computes NoP of maximum size and RP for the network flow size. The For loop in lines 3-9 generate fragment level network flow records for all packets of a flow and lines 10-16 generate fragment level network flow records for RP. These network flow records are appended to the UDP Normal Flow file **un** in Figure 5.

The function UDP-ANOMALOUS -FLOW (F) generates anomalous network flows. The For loop in lines 4-20 generate fragment level anomalous network flow record if FrT is malicious else generate Normal flow record and lines 21-38 generate for RP.

The function UDP-ATTACK (T, F) calls corresponding attack function based on T. Each attack function inflicts the attack. The attributes of fragment level flow records for attacks are given in Table: 5 and normal flow is given in the Table: 6 .The anomalous flow records are appended to the UDP anomalous Flow file ua as shown in Figure 6. The pseudocode for UDP attacks is given in Figures 6 (a) to 6(f).The sample output is given in the Table 7.

5. Accuracy Index for Evaluation of ANIDS

Anomaly intrusion detection methods classify a network flow into either anomalous flow or normal flow. Hence, it is a binary classification problem. The detection of oil spills in satellite radar images[18], fraud detection in mobile communications[19] or credit cards[20], diagnosis of rare diseases[21], and text classification in information retrieval[22] are some of typical examples of binary classification

TCP- ANOMALOUS-FLOW (F)

1 $\mathrm{NoP} \leftarrow \left\lfloor \frac{\mathrm{FS}}{\mathrm{MPS}} \right\rfloor \triangleright$ NoP of maximum packet size

2 $\mathrm{RP} \leftarrow \mathrm{FS} - (\mathrm{MPS} * \mathrm{NoP}) \triangleright$ RP in bytes

3 NoA←7 $\triangleright$ Number of attacks

4 **for** i=1 to NoP

5 $NoF \leftarrow \left\lfloor \frac{MPS}{MTU - 20} \right\rfloor \triangleright$ Number of fragments of size MTU

6 $\mathrm{RF} \leftarrow (\mathrm{MPS} - (\mathrm{MTU} - 20) * \mathrm{NoF} \triangleright$ Residual fragment in bytes

7 **for** j=1 to NoF

8 FrT← DER(P) $\triangleright$ Fragment type

9 If (FrT==1)

10 T= DURV (1,NoA)

11 ta ← TCP-ATTACK (T,F)

```
else
ta ← NORMAL(F)
if (RF≠0)
FrT← DER(P)
If (FrT==1)
T= DURV (1,NoA)
ta ← TCP-ATTACK (T,F)
else
ta ← NORMAL(F)
if (RP≠0)
NoRF ← ⌊RP / (MTU − 20)⌋ ▷ Number of fragments of RP
RPRF ← (RP − (MTU − 20) * NoRF) ▷ Residual fragment of RP in bytes
for k=1 to NoRF
FrT← DER(P)
If (FrT==1)
T= DURV (1, NoA)
ta ← TCP-ATTACK (T,F)
else
ta ← NORMAL(F)
if (RPRF≠0)
FrT← DER(P)
If (FrT==1)
T= DURV (1, NoA)
ta ← TCP-ATTACK (T,F)
else
ta ← NORMAL(F)
Return ta
```

Figure 4. Generation of TCP anomalous flows.

```
TCP-ATTACK (T,F)
CASE T OF
    1   LAND-ATTACK (F)
    2   XMASS-ATTACK (F)
    3   NESTEA-ATTACK (F)
    4   ROSE-ATTACK (F)
    5   WINNUKE-ATTACK (F)
    6   PORTSCAN-ATTACK (F)
    7   HOSTSCAN-ATTACK (F)
```

```
END CASE
Return t_ta
```

Figure 4(a). Selection of TCP attacks.

```
NORMAL(F)
t_ta←<FN,i,j,SIP,DIP,SP,DP,Proto,0,1,0,0,0,0,1,1500,(j-1)*185>
Return t_ta
```

Figure 4 (b). TCP fragment level normal flow.

```
LAND-ATTACK (F)
SIP ←DIP
t_ta←<FN,i,j,SIP,DIP,SP,DP,Proto,0,1,0,0,1,0,0, 40,0 >
Return t_ta
```

Figure 4(c). Land attack.

```
XMASS-ATTACK (F)
t_ta←<FN,i,j,SIP,DIP,SP,DP,Proto,1,0,1,0,0,1,0, 40,0 >
Returnt_ta
```

Figure 4(d). X mass attack.

```
NESTEA-ATTACK (F)
FO←FO − k*185 // 0<k<1
t_ta←<FN,i,j,SIP,DIP,SP,DP,Proto,0,1,0,0,0,0,1,1500,FO>
Returnt_ta
```

Figure 4(e). Nestea attack.

```
ROSE-ATTACK (F)
MF←1
t_ta←<FN,i,j,SIP,DIP,SP,DP,Proto,0,1,0,0,0,0,1,400, (j-1)*185>
Returnt_ta
```

Figure 4(f). Rose attack.

```
WINNUKE-ATTACK (F)
DP ←139
t_ta←<FN,i,j,SIP,DIP,SP,DP,Proto,1,1,0,0,0,0,1,1500, (j-1)*185>
Returnt_ta
```

Figure 4(g). Win nuke attack.

```
PORTSCAN-ATTACK (F)
DP ←DURV(1,65536)
t_ta←<FN i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,1,0,0,40,0 >
Return t_ta
```

Figure 4(h). Port scan attack.

```
HOSTSCAN-ATTACK (F)
DIP← DURV(1,50)
```

```
t_ta ← <FN,i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,1,0,0, 40,0 >
Return t_ta
```

Figure 4(i). Host scan attack.

```
UDP-NORMAL-FLOW (F)
NoP ← ⌊FS / MPS⌋ ▷ NoP of maximum packet size
RP ← FS − (MPS * NoP) ▷ RP in bytes
for i=1 to NoP
NoF ← ⌊MPS / (MTU − 20)⌋ ▷ Number of fragments of size MTU
RF ← (MPS − (MTU − 20) * NoF) ▷ Residual fragment in bytes
for j=1 to NoF
un← <FN,i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,1,1500,(j-1)*185
if (RF≠0)
un← <FN,i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,0,(RF+20),((j- 2)*185+RF/8)
if (RP≠0)
NoRF ← ⌊RP / (MTU − 20)⌋ ▷ Number of fragments of RP
RPRF ← (RP − (MTU − 20) * NoRF) ▷ Residual fragment of RP in bytes
for k=1 to NoRF
un← <FN,i,k,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,1,1500,(k-1)*185>
if (RPRF≠0)
un← <FN,i,k,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,0,(RPLF+20),(k-2)*185 +(RPRF/8)>
Return un
```

Figure 5. Generation of UDP normal flows.

problems. Normally, confusion matrix is employed for representing the measures of binary classification problem. The confusion matrix for representing the measures of an anomalous flow detection method is shown in Figure 7.

An anomalous flow is specified as “Positive” whereas the normal flow is specified as “Negative”. The term “True” indicates correct detection whereas “False” indicates incorrect detection. The number of anomalous flows correctly detected is categorized under “True Positive” whereas the number of incorrectly detected anomalous flows aka Type-II Errors is categorized under “False Negative”. The number of normal flows correctly detected is categorized under “True Negative” whereas the number of incorrectly detected normal flows aka Type-I errors is categorized under “False Positive”.

Foster Provost et al[23] proposed a set of metrics based on the elements of confusion matrix. The same are described below briefly.

5.1 True Positive Rate (TPR)

TPR is the ratio between the number of true positive flows and anomalous flows refer Equation (1). It is also known as sensitivity, hit rate or recall.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

5.2 True Negative Rate (TNR)

TNR is the ratio between the number of true negative flows and normal flows refer Equation (2). It is also known as specificity.

Table 7. Sample output of normal and anomalous network flows

FN	i	*j*	*SIP*	*DIP*	*SP*	*DP*	*Proto*	*FLAGS*						*length*	*FO*	**Attack**	
								U	A	P	R	S	F	MF			
1	1	1	83	13	440	224	TCP	0	1	0	0	0	0	1	1500	7585	Normal
5	1	16	44	44	906	906	TCP	0	1	0	0	1	0	0	40	0	Land
10	4	42	16	45	536	424	TCP	1	0	1	0	0	1	0	40	0	Xmass
12	3	14	71	4	237	520	TCP	0	1	0	0	0	0	1	1500	2316	Nestea
21	2	1	42	16	536	71	TCP	0	1	0	0	0	0	1	400	0	Rose
49	3	4	81	44	965	139	TCP	1	0	0	0	0	0	1	1500	555	Winnuke
58	1	21	78	50	730	803	TCP	0	1	0	0	1	0	0	40	0	Port Scan
61	1	29	8	1	678	221	TCP	0	1	0	0	1	0	0	40	0	Host Scan
4	1	6	83	13	440	224	UDP	0	0	0	0	0	0	1	1500	925	Normal
35	5	42	56	1	904	389	UDP	0	0	0	0	0	0	1	1500	3078	Teardrop
42	3	21	55	255	708	209	UDP	0	0	0	0	0	0	0	40	0	Fraggle
51	5	24	69	9	586	67	UDP	0	0	0	0	0	0	0	40	0	Port Scan
89	2	32	83	49	231	1013	UDP	0	0	0	0	0	0	0	40	0	Host Scan

UDP- ANOMALOUS -FLOW (F)

$\mathbf{NoP} \leftarrow \left\lfloor \frac{\mathbf{FS}}{\mathbf{MPS}} \right\rfloor$ ⊳ NoP of maximum packet size

$\mathbf{RP} \leftarrow \mathbf{FS} - (\mathbf{MPS} * \mathbf{NoP})$ ⊳ RP in bytes

$\mathbf{NoA} \leftarrow \mathbf{4}$ ⊳ Number of attacks

for i=1 to NoP

$NoF \leftarrow \left\lfloor \frac{MPS}{MTU - 20} \right\rfloor$ ⊳ Number of fragments of size MTU

$\mathbf{RF} \leftarrow (\mathbf{MPS} - (\mathbf{MTU} - \mathbf{20}) * \mathbf{NoF}$ ⊳ Residual fragment in bytes

for j=1 to NoF

FrT← DER(P) ⊳ Fragment type

If (FrT==1)

T= DURV (1,NoA)

ua← UDP-ATTACK (T,F)

else

ua ← UNORMAL(F)

if (RF≠0)

FrT← DER(P) ⊳ Fragment type

If (FrT==1)

T= DURV (1,NoA)

ua← UDP-ATTACK (T,F)

else

ua ← UNORMAL(F)

if (RP≠0)

```
NoRF ← ⌊RP / (MTU − 20)⌋
RPRF ← (RP − (MTU − 20) * NoRF)
for k=1 to NoRF
FrT← DER(P)          ⊳ Fragment type
If (FrT==1)
T= DURV (1, NoA)
ua← UDP-ATTACK (T,F)
else
ua ← UNORMAL(F)
if (RPRF≠0)
FrT← DER(P)          ⊳ Fragment type
If (FrT==1)
T= DURV (1, NoA)
ua← UDP-ATTACK (T,F)
else
ua ← UNORMAL(F)
Return ua
```

Figure 6. Generation of UDP anomalous flows.

```
UDP-ATTACK (T,F)
CASE T OF
    1    TEARDROP- ATTACK (F)
    2    FRAGGLE- ATTACK (F)
    3    UPORT SCAN- ATTACK (F)
    4    UHOST SCAN- ATTACK (F)
END CASE
Return t_ua
```

Figure 6(a). Selection of UDP attacks.

```
UNORMAL (F)
t_ua ← <FN,i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,1,1500,(j-1)*185>
Return t_ua
```

Figure 6 (b). UDP fragment level normal flow.

```
TEARDROP- ATTACK (F)
FO←FO − x*185 // 0<x<1
t_ua ← <FN,i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,1,1500,FO>
Return t_ua
```

Figure 6 (c). Tear drop attack.

```
FRAGGLE- ATTACK (F)
```

```
DIP ←X.X.X.255
t_ua ← <FN,i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,0,40,0 >
Return t_ua
```

Figure 6 (d). Fraggle attack.

```
UPORT SCAN- ATTACK (F)
DP ←DURV(1, 65536)
t_ua ← <FN,i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,0,40,0 >
Return t_ua
```

Figure 6 (e). Port scan attack.

```
UHOST SCAN- ATTACK (F)
DIP ← DURV(1, 50)
t_ua ← <FN,i,j,SIP,DIP,SP,DP,Proto,0,0,0,0,0,0,0,40,0 >
Return t_ua
```

Figure 6 (f). Host scan attack.

		Detected Flows	
		Anomalous	Normal
Actual Flows	Anomalous	True Positive(TP)	False Negative(FN) Type-II Error
	Normal	False Positive(FP) Type –I Error	True Negative(TN)

Figure 7. Confusion matrix.

$$TNR = \frac{TN}{TN + FP} \quad (2)$$

5.3 False Positive Rate (FPR)

FPR is the ratio between the number of false positive flows and normal flows refer Equation (3). It is also known as Type-I Error rate.

$$FPR = \frac{FP}{FP + TN} \quad (3)$$

5.4 False Negative Rate (FNR)

FNR is the ratio between the number of false negative flows and anomalous flows refer Equation (4). It is also known as Type-II Error rate.

$$FNR = \frac{FN}{FN + TP} \quad (4)$$

5.5 Positive Predictive Value (PPV)

PPV is the ratio between the number of true positive flows and the sum of true positive and false positive flows refer Equation (5). This is also called as precision.

$$PPV = \frac{TP}{TP + FP} \quad (5)$$

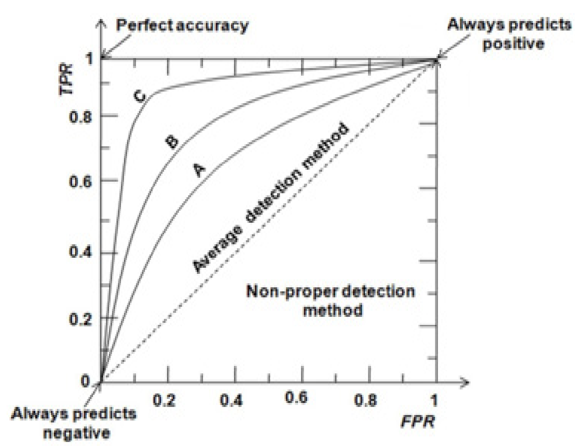

Figure 8. ROC curve.

5.6 Negative Predictive Value (NPV)

NPV is the ratio between the number of true negative flows and the sum of true negative and false negative flows refer Equation (6).

$$NPV = \frac{TN}{TN + FN} \tag{6}$$

5.7 Misclassification Rate (MCR)

MCR is the ratio between the sum of number of false negatives and false positives and total numbers of flows considered for evaluating an AINDS refer Equation (7).

$$MCR = \frac{FP + FN}{TP + FP + FN + TN} \tag{7}$$

5.8 Accuracy

Accuracy is the ratio between the sum of number of true positives and true negatives and total numbers of flows considered for evaluating an AINDS refer Equation (8).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{8}$$

The ROC curve, TPR versus FPR is used widely for evaluating the performance of classifiers[24]. A single point in ROC space is produced when a discrete classifier is applied on a test set. When a continuous classifier is applied on a test set, one point for each of the threshold values is produced in ROC space. An ROC curve is fitted through those points. The ROC curves for three different classifiers are shown in Figure 8.

The point (0, 0) represents entirely misclassification of anomalous and correct classification of normal flows. The point (1, 1) represents entirely correct classification of anomalous flows but entirely misclassification of normal flows. The point (0, 1) represents entirely correct classification of anomalous as well as normal flows. The point (1, 0) represents entirely misclassification of anomalous flows as well as normal flows. The Area under ROC Curve (AUC) is taken as a single value which ranges from 0 to 1 for presenting the performance of a classifier. Generally, it is deduced that higher the AUC better the performance of a classifier. However, it is not so as AUC 1 indicates that TPR is 1 for $0 \leq FPR \leq 1$. It means that the anomalous flows are correctly classified irrespective of misclassification of normal flows. Hence, higher the AUC better the performance of classifier is not appropriate as it does not consider to minimize Type-I error which is also necessary.

Bo Tang et al[25] in text categorization in information retrieval applied Geometric mean of precision (PPV) and recall (TPR) as metric to evaluate performance of classifier for multiple classes. The ANIDS is a binary classifier in which TPR and TNR is to be maximized. Hence, the geometric mean of TPR and TNR is proposed as a single measure which represents the combined accuracy of TPR and TNR. It is referred to as GMAI of a classifier. Mathematically formulation of it is given in Equation (9).

$$Maximize GMAI = f(TPR, TNR) = \sqrt{TPR * TNR} \tag{9}$$

It is obvious that maximizing GMAI is equivalent to minimizing 1- $\sqrt{TPR * TNR}$.

Higher the GMAI better the performance of a classifier. The relation between TPR and TNR is shown in Figure 9. Equivalently, lower the 1-GMAI better the performance of a classifier. Further, the analyst can choose GMAI based on his perception for a given situation and consider it as threshold value. A classifier whose GMAI is greater than or equal to threshold value is chosen. If it is required to choose a classifier from a set of classifiers yielding GMAIs greater than or equal to threshold value, the classifier associated with the maximum GMAI is chosen.

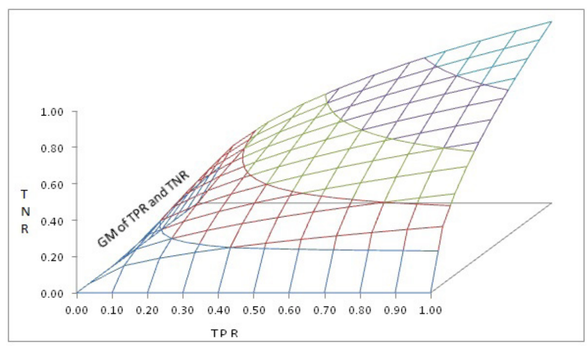

Figure 9. The relationship between TPR and TNR.

5.9 The Cost Model

Obviously, the sum of the elements of confusion matrix resulted as a consequence of applying a classifier on a given test data set, is the cardinality of that test data set. Then the empirical probabilities of the elements of confusion matrix are defined Equation (10) to Equation (13).

The probability of detecting anomalous flow as anomalous is defined as

$$p_{TP} = \frac{TP}{NoF} \tag{10}$$

The probability of detecting anomalous flow as normal is defined as

$$p_{FN} = \frac{FN}{NoF} \tag{11}$$

The probability of detecting normal flow as anomalous is defined as

$$p_{FP} = \frac{FP}{NoF} \quad (12)$$

The probability of detecting normal flow as normal is defined as

$$p_{TN} = \frac{TN}{NoF} \quad (13)$$

Where

TNoF = Total Number of flows = TP + FN+FP+TN

The opportunity cost is the cost of misclassification. The EOC of a classifier is formulated as shown in Equation (14).

$$MinimizeEOC = C_{TP} * p_{TP} + C_{FN} * p_{FN} + C_{FP} * p_{FP} + C_{TN} * p_{TN} \quad (14)$$

Where

$C_{TP} = Oppotunitycostclassifyinganomolusflowasanomolous$

$C_{FN} = Oppotunitycostclassifyinganomolousflowasnormal$

$C_{FP} = Oppotunitycostclassifyingnormalflowasanomolous$

$C_{TN} = Oppotunitycostclassifyingnormalflowasnoraml$

Obviously, the opportunity costs, C_{TP} and C_{TN} are zero as the classification done correctly. Hence, the above equation boils down to:

$$MinimizeEOC = C_{FN} * p_{FN} + C_{FP} * p_{FP}$$

Generally, it is difficult to estimate the opportunity costs, C_{FN} and C_{FP}. Whenever it is difficult to estimates the costs, service level is taken as surrogate. Here, the GMAI can be taken as surrogate for EOC for selection of a classifier or ranking of classifiers.

6. Conclusions

The real-life and benchmark datasets of network flows accessible to researchers lack the requisite data for evaluating ANIDS in the context of DoS and PROBE attacks. The present study proposes a model for generating synthetic network flows useful for effective evaluation of ANIDS in the context of DoS and PROBE attacks.

The study also proposes a performance metric, GMAI in place of widely used ROC which proves to be a better metric. An expected cost model based on the opportunity cost concept which proves to be equivalent to GMAI is formulated. Hence, the GMAI can be treated as surrogate to service level for comparing alternative ANIDS.

7. References

1. Mogul JC, Rashid RF, Accetta MJ. The packet filter: an efficient mechanism for user-level network code. Technical report, Western Research Lab, Digital Equipment Corporation: California, USA; 1987 Nov. p. 1–26.
2. Denning DE. An intrusion-detection model. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Software Engineering. 1987 Feb; 13(2):222–32.
3. Debar H, Dacier M, Wespi A. Towards a taxonomy of intrusion detection systems. The International Journal of Computer and Telecommunications Networking. 1999 Apr; 31(8):805–22.
4. Monowar HB, Bhattacharyya DK, Kalita JK. Network anomaly detection: methods, systems and tools. Institute of Electrical and Electronics Engineers (IEEE) Communications Surveys and Tutorials. 2014 Jan; 16(1):303–36. https://doi.org/10.1109/SURV.2013.052213.00046.
5. Weber D. Taxonomy of computer intrusions [Master thesis]. Cambridge, MA, Massachusetts Institute of Technology; 1998.
6. Kendall K. A database of computer attacks for the evaluation of intrusion detection systems [Master thesis]. Cambridge, MA, Massachusetts Institute of Technology; 1999.
7. Neustar. The Threat scape widens: DDoS aggression and the evolution of IOT risks [Internet]. 2016 [cited 2016 Sep 27]. Available from: https://ns-cdn.neustar.biz/creative_services/biz/neustar/www/resources/whitepapers/it-security/ddos/2016-apr-ddos-report.pdf.
8. Bhuyan MH, Bhattacharyya DK, Kalita JK. Towards generating real-life datasets for network intrusion detection. International Journal of Network Security. 2015 Nov; 17(6):675–93.
9. The SHMOO Group. Defcon data set [Internet]. 2015 [cited 2015 Jun 7]. Available from: http://cctf. Shmoo.com.
10. Centre for Applied Internet Data Analysis (CAIDA). Anonymized internet traces [Internet]. 2015 [cited 2015 Sep 11]. Available from: http://www.caida.org/data/overview.
11. Lawrence Berkeley National Laboratory/International Computer Science Institute Enterprise Tracing Project [Internet]. 2013 [cited 2013 Jul 30]. Available from: http://www.icir.org/enterprise-tracing/Overview.html.
12. UCI Machine Learning Repository. KDD Cup 1999 data data set [Internet]. 2015 [cited 2015 Dec 9]. Available from: https://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data.
13. University of New Brunswick. NSL-KDD data set for network-based intrusion detection systems [Internet]. 2016 [cited 2016 Mar 11]. Available from: http://nsl.cs.unb.ca/NSL-KDD/.
14. Song J, Takakura H, Okabe Y. Description of Kyoto university benchmark data [Internet]. 2016 [cited 2016 Jun

7]. Available from: http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf.
15. MAWI working group traffic archive [Internet]. 2016 [cited 2016 Jun 7]. Available from: http://mawi.wide.ad.jp/mawi.
16. Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Towards developing a systematic approach to generate benchmark datasets for intrusion detection. Computers and Security. 2012 May; 31(3):357–74. Crossref.
17. University of Brescia (UNIBS). UNIBS: Data sharing [Internet]. 2011 [cited 2011 Jul 12]. Available from: http://netweb.ing.unibs.it/~ntw/tools/traces/index.php.
18. Kubat M, Matwin S. Addressing the curse of Imbalanced training sets: one side selection. In the Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, Morgan Kaufmann; 1997. p. 179–86.
19. Fawcett T, Provost F. Combining data mining and host learning for effective user profiling. In the Proceedings of the second international conference on Knowledge Discovery and Data Mining (KDD), Portland; 1996. p. 8–13.
20. Chan PK, Prodromidis A, Stolfo SJ. Distributed data mining in credit card fraud detection. Institute of Electrical and Electronics Engineers (IEEE) Journal on Intelligent Systems. 1999 Nov; 14(6):67–74.
21. Swets JA. Measuring the accuracy of diagnostic systems. Journal of Store, New Series. 1988 Jun; 240(48):1285–93.
22. Lewis DD, Gale WA. A sequential algorithm for training text classifiers. In the Proceedings of seventeenth annual international conference on Research and Development in Informational Retrieval, London; 1994. p. 3–12. Crossref.
23. Provost FJ, Fawcett T. Robust classification for imprecise environments. Machine Learning. 2001 Mar; 42(3):203–31.
24. Maxion RA, Roberts RR. Proper use of ROC curves in intrusion anomaly detection. Technical Report, School of Computing Science, University of Newcastle: Australia; 2004.
25. Tang B, He H, Baggenstoss PM, Kay S. A bayesian classification approach using class-specific features for text categorization. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Knowledge and Data Engineering. 2016 Jun; 28(6):1602–6. Crossref.