Abstract:

Cyber-attacks are becoming more sophisticated and thereby presenting increasing challenges in accurately detecting intrusions. Failure to prevent the intrusions could degrade the credibility of security services, e.g. data confidentiality, integrity, and availability. Meanwhile, Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) are one of the most important defense tools against the sophisticated and ever-growing network attacks. Most research in the field of network intrusion detection heavily relies on datasets. Datasets in this field, however, are scarce and difficult to reproduce. To compare, evaluate, and test related work, researchers usually need the same datasets or at least datasets with similar characteristics as the ones used in related work. In this work, I have used the Intrusion Detection Dataset Toolkit (ID2T) to alleviate the problem of reproducing datasets with desired characteristics to enable an accurate replication of scientific results.

1. About the internship:

Summer Research Fellowship Programme was introduced by Indian Academy of Sciences. I was selected for fellowship for two months. They have allocated me a professor in Indian Institute of Technology Hyderabad. I have created datasets which are very much essential for various machine learning based intrusion detection systems under the guidance of Dr. Tamma Bhemmarjuna Reddy and Dr. A. Suresh Kumar.

They are working to built a intrusion detection system based on adaptive machine learning methods and they are also planning to deploy a IDS for IITH. I have given a small contribution to it by helping with the datasets creation and evaluating the datasets. I also injected various synthetic attacks to some databases using some available tools and made a balanced dataset.

The faculty of IITH are very supportive and given all the facilities to me. The facilities of IITH are very good and advanced.

My college faculty also made very much help to me regarding some doubts. The satisfaction and euphoria that accompany the successful completion of task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

2. System Requirements:

2.1. Software Requirements:

1. Operating system : Ubuntu or Kali
2. Programming Language: Python 3
3. Tools: ID2T, Wireshark
4. Environment: Jupiter Notebook or Google Colab

2.2. Hardware Requirements:

1.Processor: Octacore

2. Graphic card: NVIDIA Genforce 1050 Ti

3.RAM: >=12GB

4.Harddisk: >=1 TB

3. INTRODUCTION:

The future of the Internet is to provide more secure applications and services , which is the key motivation to create Intrusion Detection Systems (IDS). These systems aim to monitor and analyze the network traffic and automatically discriminate possible threats from normal user behavior. One of the most common approaches for the creation of Intrusion Detection Systems is to classify the threats through the application of machine learning models. In turn, to train a machine learning model, it is necessary to input a dataset composed of simulated traffic or a dataset composed of attack and normal-proven traffic.

Evaluating the detection capabilities of Network Intrusion Detection Systems (NIDSs) is a crucial task in today's Internet age. NIDSs have become an almost mandatory line of defense against attacks due to our dependency on the Internet and the Internet's threat landscape. The need to develop NIDSs that can keep up with evolving attacks and motivated adversaries has yielded much research in the direction of identifying old and new, previously unobserved, threats. Evaluating NIDSs have intrinsic complexities and challenges that need to be addressed irrespective of which intrusion detection method they use. The evaluation of NIDSs relies on quality datasets to assess their capabilities. Quality datasets enable accurate comparison of different intrusion detection methods. Reliable datasets, however, are not readily available.

Reliable datasets useful for the evaluation of NIDSs are hard to obtain. Widely available datasets tend to be outdated, often lack labeled attacks and usually contain overlooked defects. Further- more, these datasets are often not publicly available or difficult to obtain . Most datasets are distributed as Packet Capture (PCAP) files. These files, being in essence just an ordered collection of network packets, can be thought as of having packets originating from one of two sources: a benign or a malicious one. Packets originating from benign sources compose the normal or back-ground traffic of the dataset. Packets created by malicious activities on a network compose the attack traffic of the dataset.

4. REVIEW OF DATASETS:

Many researchers have published papers based on generated IDS datasets, such as ISCX, UNSW-NB15, and UGR'16. In this section, we introduce several IDS datasets with their characteristics. We highlight several important requirements from their perspective.

4.1. KDD99

The KDD99 dataset was created in 1999, using tcpdump, and was built based on the data captured by the DARPA 98 IDS evaluation program. The training data are around four gigabytes of compressed TCP data from seven weeks of network traffic. The network traffic contains attack traffic and normal traffic. The capture of the network traffic was done in a simulated environment. The dataset contains a total of 24 attack types, which fall into four main categories: Denial of Service (DOS), Remote to Local (R2L), User to Root (U2R), and probing. KDD99 has been used extensively in IDS research. The report showed that during 2010–2015, 125 published papers performed IDS evaluation using KDD99. While this dataset is considered inadequate for evaluation, such as a lot of redundant instances recorded, the main problem is that the dataset is not up to date with the recent situation and attack vectors. Although many researchers were already convinced with this information, studies from another group of researchers argued that this dataset is the most widely used for benchmarking or to limit their study only for KDD99 .

4.2. MAWILab

MAWI was built in 2001 and consists of a set of labels locating traffic anomalies in the MAWI archive. This dataset contains tcpdump packet traces captured from an operational testbed network in a link between Japan and the United States. The archive contains 15 min of daily traces. This dataset is huge with a very long period. The labeled MAWI archive is known as MAWILab, obtained from a graph-based methodology that combines different and independent anomaly detectors. MAWI archives labeling is based on inferences that results in no ground-truth traffic that can be used for evaluation. The label has three classes: anomalous for a true anomaly, suspicious indicates that the traffic is likely to be anomalous, and notice is assigned as an anomaly but it does not reach a consensus from all anomaly detectors. Several researchers used MAWILab for anomaly detection and generating labeled flow. The limitation of this dataset is that the packet traces are captured for 15 min each day. The header information is available in the packet traces but the payload is removed.

4.3. CAIDA (Center of Applied Internet Data Analysis)

CAIDA has several different types of datasets, categorized as ongoing, one-time snapshot, and complete. CAIDA collects the data from different locations, and each of the datasets has different characteristics, such as Distributed Denial of Services (DDoS) attack, UDP probing, BGP monitoring, IPv4 census with passive traffic traces captured from a darknet, an academic ISP, and a residential BGP with active measurements of ICMP ping, HTTP GET and traceroutes. Most of the datasets are anonymized with IP addresses and the payload, which severely reduces their usefulness. Based on their catalog, during 2017–2020, most of the papers related to IDS and security focused on Denial of Service (DoS), Distributed Denial of Service (DDoS), DNS security, Network Telescope Daily Randomly, and Uniformly Spoofed Denial-of-Service (RSDoS) Attack Metadata. Each record contains the IP address of the attack victim, the number of distinct attacker IPs in the attack, the number of distinct attacker ports and target ports, the cumulative number of packets observed in the attack, the cumulative number of bytes seen for the attack, the maximum packet rate seen in the attack as the average per minute, the timestamp of the first and the last observed packet of the attack, the autonomous system number of target_IP at the time of the attack, and the country and continent geolocation of target_IP at the time of the attack. This dataset is updated every day.

4.4. SimpleWeb

SimpleWeb is a dataset collected from the network of the University of Twente . This dataset contains packet headers of all packets that are transmitted over the uplink of access to the internet. The packets are captured with tcpdump and Netflow version 5. The payload from the packets is removed because it contains sensitive information, such as HTTP requests or conversations of instant messaging applications. The labeled dataset for suspicious traffic is collected by using a honeypot server. Despite no ground-truth data being available, researchers still use it to compare with the different real-world environment (e.g., campus network, backbone link), while others employ it for background traffic for botnet detection, and to evaluate publicly available dataset for similarity searches to detect network threats.

4.5. NSL-KDD

NSL-KDD is an updated dataset that tries to solve some of the inherent problems in the KDD99 dataset. The NSL-KDD dataset contains features and labels indicating either normal or an attack, with specific types of attacks. Every instance in the training set contains a single connection session, which is divided into four groups,

such as basic features from the network connection, content-related features, time-related features, and host-based traffic features. Each instance is labeled either as normal or attack. These attacks are categorized into four groups: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probing. Many researchers use it as a benchmark to help them to compare their intrusion detection systems performance. Several groups of researchers used different scopes, such as IoT-based networks and Vehicular Ad Hoc Network (VANET). The former is for SYN flood, UDP flood, and Ping of Death (PoD) detection, while the latter is mostly for DDoS detection. Other researchers used different methods and switched from conventional machine learning to deep learning based methods.

4.6. IMPACT

IMPACT is a marketplace of cyber-risk data. The data distribution and tool repository are provided by multiple providers and stored and accessed from multiple hosting sites. The datasets related to cyber-attacks, such as the daily feed of network flow data produced by Georgia Tech Information Security Center's malware analysis system, updates once a year. These datasets are only open for specific countries based on approval by the Department of Homeland Security (DHS).

4.7. UMass

UMass is a trace repository provided by the University of Massachusetts Amherst . The network-attack-relevant data are provided with various type of data, such as traffic flow from the TOR network, a trace of attack simulation on peer-to-peer data sharing network, passive localization attack simulation with reality mining dataset containing sensor data (proximity, location, location labels, etc.), and survey data (personal attributes, research group, position, neighborhood of apartment, and lifestyle).

4.8. Kyoto

This dataset was created between 2006 and 2015 by Kyoto University through honeypot servers. This dataset was created using Bro 2.4 (the former name of Zeek) with 24 statistical features consisting of 14 features extracted based on the KDD99 dataset and an additional 10 features, such as IDS_detection, Malware_detection, Ashula_detection, Label, Source_IP_Address, Source_Port_Number, Destination_IP_Address, Destination_Port_Number, Start_Time, and Protocol . The information is limited to the attack information targeting the honeypot server. There are no packet traces or information about the payload. Furthermore, the information on how to label the dataset is not found. Several published papers using the Kyoto

dataset focused on anomaly detection, especially on the feature analysis, feature dimensionality reduction and ensemble classifier.

4.9. IRSC

This dataset was created by Indian River State College and consists of network flows and full packet capture. The dataset represents a real-world environment in which the attack traffic has two different types: the controlled version, which is synthetically created by the team, and the uncontrolled version, which are the real attacks. The flow based traffic created with the Silk and the full packet capture created with the Snort IDS. The additional source of flow data was produced from the Cisco firewall using NetFlow version 9. While the authors stated that the dataset is a complete capture with payload and flow data, unfortunately, it is not publicly available.

4.10. UNSW-NB15

UNSW-NB15 was created using a commercial penetration tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). This tool can generate hybrid synthetically modern normal activities and contemporary attack behaviors from network traffic. They collected tcpdump traces for a total duration of 31 h. From these network traces, they extracted 49 features categorized into five groups: flow features, basic features, content features, time features, and additional generated features. Feature and statistical analyses are the most common methods used in several published papers employing UNSW-NB15. While it could obtain 97% accuracy by using 23 features, incorporated the XGBoost algorithm for feature reduction, using several traditional machine learning algorithms for evaluation, such as Artificial Neural Network (ANN), Logistic Regression (LR), k-Nearest Neighbor (kNN), Support Vector Machine (SVM) and Decision Tree (DT).

4.11. CICIDS-2017

This dataset was created by the Canadian Institute for Cybersecurity at University of Brunswick in 2017. The purpose of CICIDS-2017 was intrusion detection, and it consisted of several attack scenarios. In this dataset, the attack profiles were produced using publicly available tools and codes. Six attack profiles were implemented, such as brute force, heartbleed, botnet, DoS, DDoS, web attack, and infiltration attack. The realistic background traffic was generated, using a B-Profile system . The B-Profile system extracted 25 behaviors of users based on several protocols, such as HTTP, HTTPS, FTP, SSH, and SMTP. The network traffic features were captured with CICFlowMeter, which extracted 80 features from the pcap file. The flow label

included SourceIP, SourcePort, DestinationIP, DestinationPort, and Protocol. Mixed methods are used, incorporating CICIDS-2017 to detect specific attacks such as DoS attack by using feature reduction, web-attack detection, and anomaly web traffic with ensemble architecture and feature reduction. Others are improving the AdaBoost-based method to counter the imbalance of the training data, and combining feature selection and information gain to find relevant and significant features and to improve accuracy and execution time.

5. INTRUSION DETECTION SYSTEMS :

Intrusion can be defined as any kind of unauthorised activities that cause damage to an information system. This means any attack that could pose a possible threat to the information confidentiality, integrity or availability will be considered an intrusion. For example, activities that would make the computer services unresponsive to legitimate users are considered an intrusion. An IDS is a software or hardware system that identifies malicious actions on computer systems in order to allow for system security to be maintained. The goal of an IDS is to identify different kinds of malicious network traffic and computer usage, which cannot be identified by a traditional firewall. This is vital to achieving high protection against actions that compromise the availability, integrity, or confidentiality of computer systems. IDS systems can be broadly categorized into two groups: Signature-based Intrusion Detection System (SIDS) and Anomaly-based Intrusion Detection System (AIDS).

Signature-based intrusion detection systems (SIDS)

Signature intrusion detection systems (SIDS) are based on pattern matching techniques to find a known attack; these are also known as Knowledge-based Detection or Misuse Detection (Khraisat et al., 2018). In SIDS, matching methods are used to find a previous intrusion. In other words, when an intrusion signature matches with the signature of a previous intrusion that already exists in the signature database, an alarm signal is triggered. For SIDS, host's logs are inspected to find sequences of commands or actions which have previously been identified as malware. SIDS have also been labelled in the literature as Knowledge-Based Detection or Misuse Detection (Modi et al., 2013). Figure 1 demonstrates the conceptual working of SIDS approaches. The main idea is to build a database of intrusion signatures and to compare the current set of activities against the existing signatures and raise an alarm if a match is found. For example, a rule in the form of "if: antecedent -then: consequent" may lead to "if (source IP address=destination IP address) then label as

an attack ". SIDS usually gives an excellent detection accuracy for previously known intrusions (Kreibich & Crowcroft, 2004). However, SIDS has difficulty in detecting zeroday attacks for the reason that no matching signature exists in the database until the signature of the new attack is extracted and stored. SIDS are employed in numerous common tools, for instance, Snort (Roesch, 1999) and NetSTAT (Vigna & Kemmerer, 1999).

Traditional approaches to SIDS examine network packets and try matching against a database of signatures. But these techniques are unable to identify attacks that span several packets. As modern malware is more sophisticated it may be necessary to extract signature information over multiple packets. This requires the IDS to recall the contents of earlier packets. With regards to creating a signature for SIDS, generally, there have been a number of methods where signatures are created as state machines, formal language string patterns or semantic conditions. The increasing rate of zero-day attacks has rendered SIDS techniques progressively less effective because no prior signature exists for any such attacks. Polymorphic variants of the malware and the rising amount of targeted attacks can further undermine the adequacy of this traditional paradigm. A potential solution to this problem would be to use AIDS techniques, which operate by profiling what is an acceptable behavior rather than what is anomalous, as described in the next section.

Anomaly-based intrusion detection system (AIDS)

AIDS has drawn interest from a lot of scholars due to its capacity to overcome the limitation of SIDS. In AIDS, a normal model of the behavior of a computer system is created using machine learning, statistical-based or knowledge-based methods. Any significant deviation between the observed behavior and the model is regarded as an anomaly, which can be interpreted as an intrusion. The assumption for this group of techniques is that malicious behavior differs from typical user behavior. The behaviors of abnormal users which are dissimilar to standard behaviors are classified as intrusions. Development of AIDS comprises two phases: the training phase and the testing phase. In the training phase, the normal traffic profile is used to learn a model of normal behavior, and then in the testing phase, a new data set is used to establish the system's capacity to generalise to previously unseen intrusions. AIDS can be classified into a number of categories based on the method used for training, for instance, statistical based, knowledge-based and machine learning based. The main advantage of AIDS is the ability to identify zero-day attacks due to the fact that recognizing the abnormal user activity does not rely on a signature database. AIDS triggers a danger signal when the examined behavior differs from the usual behavior.

Furthermore, AIDS has various benefits. First, they have the capability to discover internal malicious activities. If an intruder starts making transactions in a stolen account that are unidentified in the typical user activity, it creates an alarm. Second, it is very difficult for a cybercriminal to recognize what is a normal user behavior without producing an alert as the system is constructed from customized profiles.

6. TOOLS USED :

THE INTRUSION DETECTION DATASET TOOLKIT (ID2T):

ID2T is a public and open source command-line toolkit that injects synthetic attacks into (network) traffic captures. To achieve this, the toolkit first analyzes and collects statistics from a traffic capture. The statistics are then used by attack scripts to set internal parameters, and to create attack packets that generally replicate the statistical properties of background traffic. Some attack scripts are meant to generate packets that distinctively alter background traffic statistics, and ID2T facilities this. For example, ID2T's DDoS attack script may significantly change the packet size distribution of a network (among other properties) while retaining many other statistical properties (such as the packet's TTL distribution). Finally, ID2T merges the attack packets with the packets of the input. The outputs of ID2T are a PCAP file with injected attacks and a file of labels that clearly indicate when attacks start and end. If needed, then ID2T can watermark or taint packets to associate each packet to an attack.

CICFLOWMETER:

CICFlowMeter is a network traffic flow generator and analyser.

It can be used to generate bidirectional flows, where the first packet determines the forward (source to destination) and backward (destination to source) directions, hence more than 80 statistical network traffic features such as Duration, Number of packets, Number of bytes, Length of packets, etc. can be calculated separately in the forward and backward directions.

Additional functionalities include, selecting features from the list of existing features, adding new features, and controlling the duration of flow timeout. The output of the application is the CSV format file that has six columns labeled for each flow (FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort, and Protocol) with more than 80 network traffic analysis features.

Note that TCP flows are usually terminated upon connection teardown (by FIN packet) while UDP flows are terminated by a flow timeout. The flow timeout value can be assigned arbitrarily by the individual scheme e.g., 600 seconds for both TCP and UDP.

IOT-23 Dataset:

IoT-23 is a new dataset of network traffic from Internet of Things (IoT) devices. It has 20 malware captures executed in IoT devices, and 3 captures for benign IoT devices traffic. It was first published in January 2020, with captures ranging from 2018 to 2019. This IoT network traffic was captured in the Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic. Its goal is to offer a large dataset of real and labeled IoT malware infections and IoT benign traffic for researchers to develop machine learning algorithms. This dataset and its research is funded by Avast Software, Prague.

The IoT-23 dataset consists of twenty three captures (called *scenarios*) of different IoT network traffic. These scenarios are divided into twenty network captures (pcap files) from infected IoT devices (which will have the name of the malware sample executed on each scenario) and three network captures of real IoT devices network traffic (that have the name of the devices where the traffic was captured). On each malicious scenario we executed a specific malware sample in a Raspberry Pi, that used several protocols and performed different actions. The network traffic captured for the benign scenarios was obtained by capturing the network traffic of three different IoT devices: a Philips HUE smart LED lamp, an Amazon Echo home intelligent personal assistant and a Somfy smart doorlock. It is important to mention that these three IoT devices are real hardware and not simulated . This allows us to capture and analyse real network behaviour. Both malicious and benign scenarios run in a controlled network environment with unrestrained internet connection like any other real IoT device.

7. MY WORK:

I had experimented with the Intrusion Detection Dataset Toolkit(ID2T) tool for the generation of the new dataset. It is created by injecting the attack packets using the statistical analysis of background network data. The inputs for ID2T are a PCAP(Packet Capture) file containing background data, and user supplied parameters. Outputs are a PCAP file with injected synthetic attacks into it, a labels (xml) file specifying the time and location of the injected attacks, and a report on the statistical analysis of the background traffic (see Figure 1).

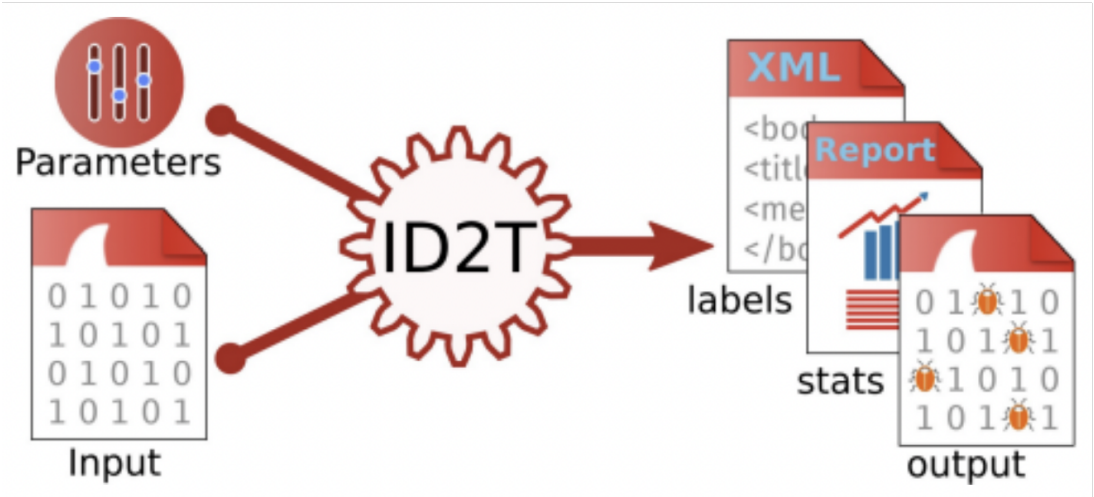


Figure 1

Attacks are manually programmed and use the statistics collection mechanism of ID2T to replicate the network properties of any input PCAP file. Many attacks are readily available which ranges from Denial of Service(DoS) to malware, and botnet infections. [1] propose that one can inject Probing, surveillance, Resource Exhaustion, Exploitation, and Botnet infection attacks. The input PCAP has no restrictions on it's size, and Internet Protocol Version 6(IPV6) traffic is not considered.

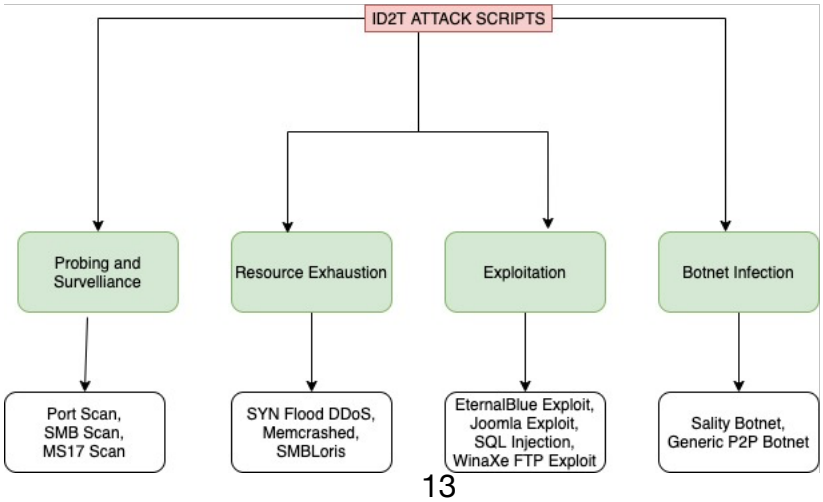


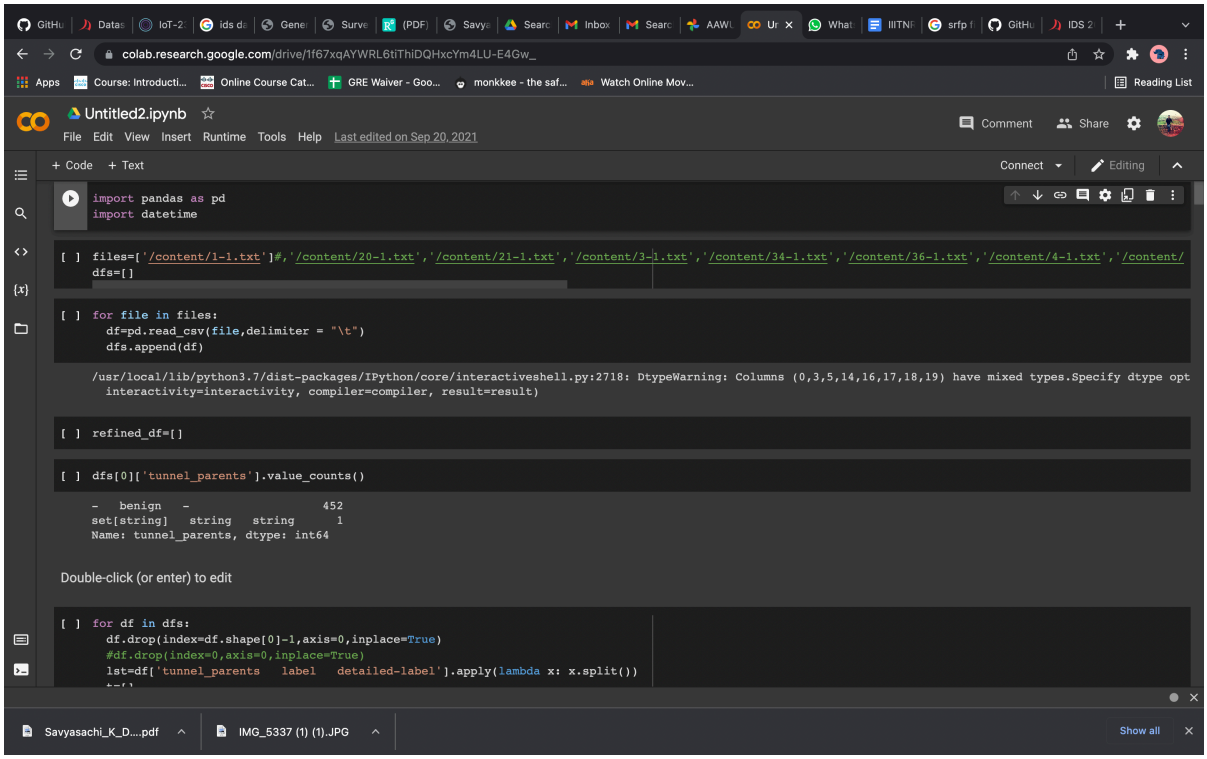
Figure 2: Classification of the synthetic attacks ID2T can inject.ID2T can inject 12 different attacks.We classify the attacks into four different classes.

ID2T also helps to removes the unused fields (IP address, checksum, etc.) from a packet using the module Testing Intrusion Detection Dataset (TIDED). ID2T uses reliability tests to ensures that network traffic is adhered to certain properties(e.g., the distribution of IP addresses conforming to supposedly back- bone network traffic), TIDED uses the results of these reliability tests for attack scripts to produce a better replication of the properties of background traffic.

I had taken a CICDS 2017 dataset and injected some attack packets of Distributed Denial of Service (DDoS), PortScan, and SQL injection into it and performed a statistical analysis on it.

By the help of cicflowmeter, I have also extracted the features from IOT23 dataset and added attack label to each packet.

8.IMPLEMENTATION SCREENSHOTS:




```
    df.drop(index=df.shape[0]-1,axis=0,inplace=True)
    #df.drop(index=0,axis=0,inplace=True)
    lst=df['tunnel_parents   label   detailed-label'].apply(lambda x: x.split())
    t=[]
    l=[]
    d=[]
    for ele in range(0,lst.shape[0]):
      t.append(lst[ele][0])
      l.append(lst[ele][1])
      d.append(lst[ele][2])
    df['tunnel_parents']=t
    df['label']=l
    df['detailed-label']=d
    df.drop(columns="tunnel_parents   label   detailed-label",axis=1,inplace=True)
    df.drop(index=0,axis=0,inplace=True)
    df.reset_index(inplace=True)
    t=[]
    for i in range(df.shape[0]):
      t.append(datetime.datetime.fromtimestamp(float(df['ts'][i])).strftime('%Y-%m-%d %H:%M:%S'))
    df['time_stamp']=t
    df.drop(columns='ts',axis=1,inplace=True)
    labels=df
    df3=pd.DataFrame()
    df3['src_ip']=labels['id.orig_h']
    df3['dst_ip']=labels['id.resp_h']
    df3['src_port']=labels['id.orig_p']
    df3['dst_port']=labels['id.resp_p']
    df3['protocol']=labels['proto']
    df3['label']=labels['label']
    df3['detailed_label']=labels['detailed-label']
    df3['src_port']= df3["src_port"].astype(int)
```

```
refined_df[0]
```

	src_ip	dst_ip	src_port	dst_port	protocol	label	detailed_label
0	192.168.100.103	65.127.233.163	51524	23	6	Malicious	PartOfAHorizontalPortScan
1	192.168.100.103	63.150.16.171	56305	23	6	Malicious	PartOfAHorizontalPortScan
2	192.168.100.103	111.40.23.49	41101	23	6	Malicious	PartOfAHorizontalPortScan
3	192.168.100.103	131.174.215.147	60905	23	6	Malicious	PartOfAHorizontalPortScan
4	192.168.100.103	91.42.47.63	44301	23	6	Malicious	PartOfAHorizontalPortScan
...	...	...	...	...	...	...	...
56802	192.168.100.103	168.81.120.214	55138	9527	6	Malicious	PartOfAHorizontalPortScan
56803	192.168.100.103	202.73.232.156	43763	48478	17	Benign	-
56804	192.168.100.103	130.162.175.105	36796	23	6	Malicious	PartOfAHorizontalPortScan
56805	192.168.100.103	148.98.157.252	43763	3083	17	Benign	-
56806	192.168.100.103	2.246.156.236	60920	8080	6	Malicious	PartOfAHorizontalPortScan

56807 rows × 7 columns

```
[ ] refined_df[0]['label'].value_counts()
```

```
benign    1374
Name: label, dtype: int64
```

```
[ ] dfsc=pd.read_csv('/content/8-1.csv')
```

```
dfsc
```

	src_ip	dst_ip	src_port	dst_port	protocol	timestamp	flow_duration	flow_byts_s	flow_pkts_s	fwd_pkts_s	bwd_pkts_s	tot_fwd_pkts	tot_b
0	192.168.100.113	81.2.254.224	123	123	17	2018-07-31 18:45:11	5490.0	32786.885246	364.298725	182.149362	182.149362	1	
1	192.168.100.113	192.168.100.102	22	64923	6	2018-07-31 18:45:13	0.0	0.000000	0.000000	0.000000	0.000000	1	
2	192.168.100.113	147.231.100.5	123	123	17	2018-07-31 18:45:21	1741.0	103388.856979	1148.765078	574.382539	574.382539	1	
3	192.168.100.113	31.31.74.35	123	123	17	2018-07-31 18:45:44	4495.0	40044.493882	444.938821	222.469410	222.469410	1	
4	192.168.100.113	147.251.48.140	123	123	17	2018-07-31 18:46:16	6988.0	25758.443045	286.204923	143.102461	143.102461	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4219	192.168.100.113	128.185.250.50	38248	50	6	2018-08-01 18:43:39	31950801.0	13.896365	0.187789	0.187789	0.000000	6	
4220	192.168.100.113	147.231.100.5	123	123	17	2018-08-01 18:44:09	1988.0	90543.259557	1006.036217	503.018109	503.018109	1	
4221	192.168.100.113	147.251.48.140	123	123	17	2018-08-01 18:44:13	5233.0	34397.095356	382.189948	191.094974	191.094974	1	

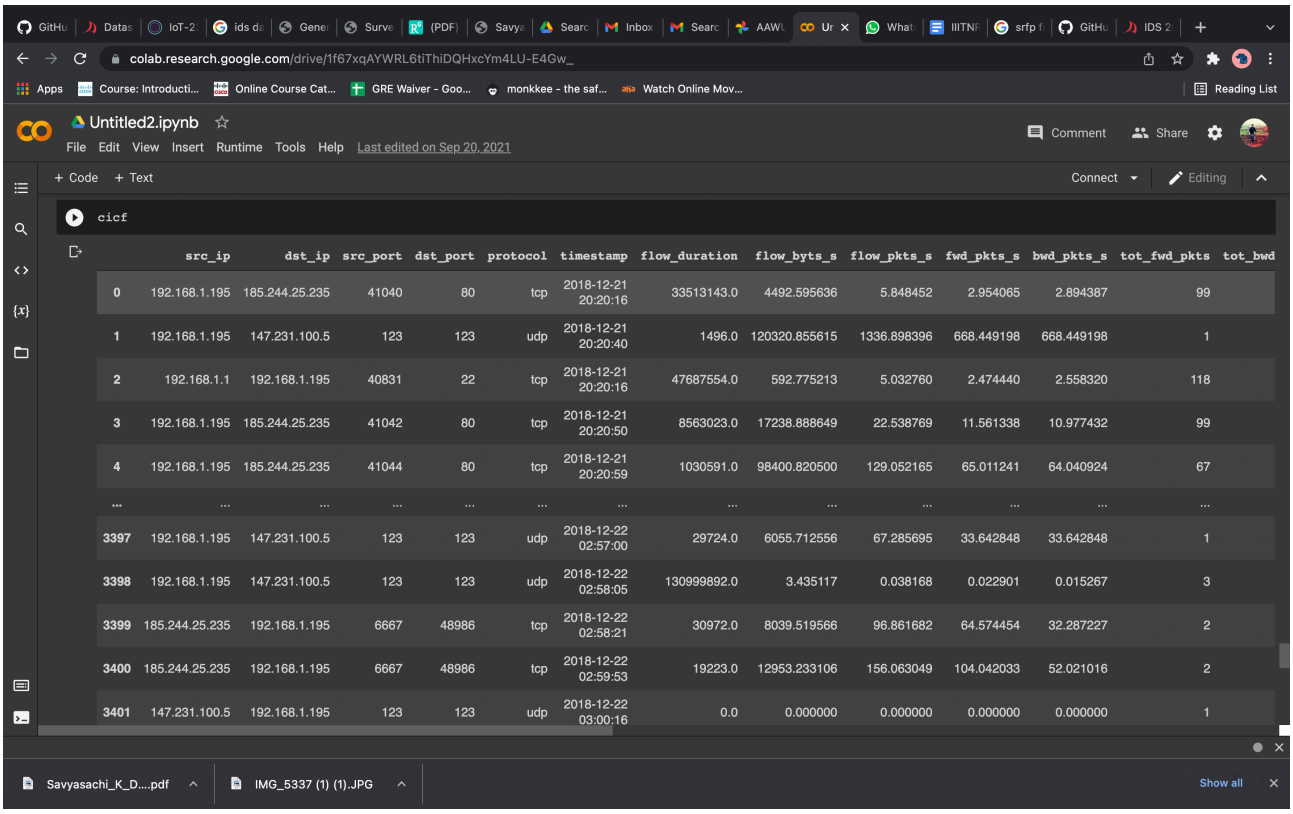
```
#df3['dst_port']= df3["dst_port"].astype(str).astype(int)
```

```
[ ]  df3
```

	src_ip	dst_ip	src_port	dst_port	protocol	label	detailed_label
0	192.168.100.103	65.127.233.163	51524	23	tcp	Malicious	PartOfAHorizontalPortScan
1	192.168.100.103	63.150.16.171	56305	23	tcp	Malicious	PartOfAHorizontalPortScan
2	192.168.100.103	111.40.23.49	41101	23	tcp	Malicious	PartOfAHorizontalPortScan
3	192.168.100.103	131.174.215.147	60905	23	tcp	Malicious	PartOfAHorizontalPortScan
4	192.168.100.103	91.42.47.63	44301	23	tcp	Malicious	PartOfAHorizontalPortScan
...	...	...	...	...	...	...	...
56802	192.168.100.103	168.81.120.214	55138	9527	tcp	Malicious	PartOfAHorizontalPortScan
56803	192.168.100.103	202.73.232.156	43763	48478	udp	Benign	-
56804	192.168.100.103	130.162.175.105	36796	23	tcp	Malicious	PartOfAHorizontalPortScan
56805	192.168.100.103	148.98.157.252	43763	3083	udp	Benign	-
56806	192.168.100.103	2.246.156.236	60920	8080	tcp	Malicious	PartOfAHorizontalPortScan

56807 rows × 7 columns

```
df3['protocol'].value_counts()
```



	src_ip	dst_ip	src_port	dst_port	protocol	timestamp	flow_duration	flow_byts_s	flow_pkts_s	fwd_pkts_s	bwd_pkts_s	tot_fwd_pkts	tot_bwd
0	192.168.1.195	185.244.25.235	41040	80	tcp	2018-12-21 20:20:16	33513143.0	4492.595636	5.848452	2.954065	2.894387	99	
1	192.168.1.195	147.231.100.5	123	123	udp	2018-12-21 20:20:40	1496.0	120320.855615	1336.898396	668.449198	668.449198	1	
2	192.168.1.1	192.168.1.195	40831	22	tcp	2018-12-21 20:20:16	47687554.0	592.775213	5.032760	2.474440	2.558320	118	
3	192.168.1.195	185.244.25.235	41042	80	tcp	2018-12-21 20:20:50	8563023.0	17238.888649	22.538769	11.561338	10.977432	99	
4	192.168.1.195	185.244.25.235	41044	80	tcp	2018-12-21 20:20:59	1030591.0	98400.820500	129.052165	65.011241	64.040924	67	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3397	192.168.1.195	147.231.100.5	123	123	udp	2018-12-22 02:57:00	29724.0	6055.712556	67.285695	33.642848	33.642848	1	
3398	192.168.1.195	147.231.100.5	123	123	udp	2018-12-22 02:58:05	130999892.0	3.435117	0.038168	0.022901	0.015267	3	
3399	185.244.25.235	192.168.1.195	6667	48986	tcp	2018-12-22 02:58:21	30972.0	8039.519566	96.861682	64.574454	32.287227	2	
3400	185.244.25.235	192.168.1.195	6667	48986	tcp	2018-12-22 02:59:53	19223.0	12953.233106	156.063049	104.042033	52.021016	2	
3401	147.231.100.5	192.168.1.195	123	123	udp	2018-12-22 03:00:16	0.0	0.000000	0.000000	0.000000	0.000000	1	

9. DISCUSSION AND CONCLUSION:

Cybercriminals are targeting computer users by using sophisticated techniques as well as social engineering strategies. Some cybercriminals are becoming increasingly sophisticated and motivated. Cybercriminals have shown their capability to obscure their identities, hide their communication, distance their identities from illegal profits, and use infrastructure that is resistant to compromise. Therefore, it becomes increasingly important for computer systems to be protected using advanced intrusion detection systems which are capable of detecting modern malware. In order to design and build such IDS systems, it is necessary to have a complete overview of the strengths and limitations of contemporary IDS research. We summarized the results of recent research and explored the contemporary models on the performance improvement of AIDS as a solution to overcome on IDS issues. However, We have discovered major problem was imbalance datasets. As normal activities are frequently changing and may not remain effective over time, there exists a need for newer and more comprehensive datasets that contain wide-spectrum of malware activities. A new malware dataset is needed, as most of the existing machine learning techniques are trained and evaluated on the knowledge provided by the old dataset such as DARPA/ KDD99, which do not include newer malware activities. Therefore, testing is done using these dataset collected in 1999 only, because they are publicly available and no other alternative and acceptable datasets are available. While widely accepted as benchmarks, these datasets no longer represent contemporary zero-day attacks. Though CICDS-2017 dataset contains many new attacks, it is not adequate. For that reason, testing of AIDS using these datasets does not offer a real evaluation and could result in inaccurate claims for their effectiveness. This study also examines four common evasion techniques to determine their ability to evade the recent IDSs. An effective IDS should be able to detect different kinds of attacks accurately including intrusions that incorporate evasion techniques. Developing IDSs capable of overcoming the evasion techniques remains a major challenge for this area of research.

10. REFERENCES:

1. Garcia Cordero et al. (2015) ID2T: a DIY Dataset Creation Toolkit for Intrusion Detection System
2. Tavallaee, M.; Stakhanova, N.; Ghorbani, A.A. Toward credible evaluation of anomaly-based intrusion-detection methods. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 2010, 40, 516–524. [CrossRef]
3. Aviv, A.J.; Haeberlen, A. Challenges in Experimenting with Botnet Detection Systems. In Proceedings of the 4th Workshop on Cyber Security Experimentation and Test (CSET 11), San Francisco, CA, USA, 8 August 2011.
4. Velan, P.; Cermák, M.; ˇ Celeda, P.; Drašar, M. A survey of methods for encrypted traffic classification and analysis. ˇ Int. J. Netw. Manag. 2015, 25, 355–374. [CrossRef]
5. De Lucia, M.J.; Cotton, C. Identifying and detecting applications within TLS traffic. In Proceedings of the Cyber Sensing 2018, Orlando, FL, USA, 15–19 April 2018; Volume 10630. [CrossRef]
6. Kaur, S.; Singh, M. Automatic attack signature generation systems: A review. IEEE Secur. Priv. 2013, 11, 54–61. [CrossRef]
7. Ahmed, M.; Naser Mahmood, A.; Hu, J. A survey of network anomaly detection techniques. J. Netw. Comput. Appl. 2016, 60, 19–31. [CrossRef]
8. Zeek IDS. 2021. Available online: https://zeek.org (accessed on 10 May 2021)
9. On Generating Network Traffic Datasets with Synthetic Attacks for Intrusion Detection
10. Sebastian Abt and Harald Baier. 2013. Are we missing labels? A study of the availability of ground-truth in net- work security research. In Proceedings of the Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS'14).
11. United States Military Academy. 2009. CDX 2009 Network. Retrieved from https://www.westpoint.edu/centers-and- research/cyber- research- center/datasets.

12. Akamai. 2018. The state of the internet / security report. Retrieved from https:// www.akamai.com/uk/en/multimedia/ documents/case- study/spring- 2018- state- of- the- internet- security- report.pdf

13. Rafael Ramos Regis Barbosa, Ramin Sadre, Aiko Pras, and Remco Meent. 2010. Simpleweb/University of Twente Traffic Traces Data Repository. Technical Report. Centre for Telematics and Information Technology, University of Twente.

14. Steven M. Bellovin. 1992. Packets found on an internet 1 introduction 2 address space oddities. Comput. Commun. 23, 3 (1992), 1–8.

15. Monowar H. Bhuyan, Dhruba K. Bhattacharyya, and Jugal K. Kalita. 2015. Towards generating real-life datasets for network intrusion detection. Int. J. Netw. Secur. 17, 6 (2015), 683–701.

16. Daniela Brauckhoff, Arno Wagner, and May Martin. 2008. FLAME: A flow-level anomaly modeling engine. In Proceedings of the Conference on Cyber Security (CSET'08).

17. CAIDA. 2017. CAIDA Data—Overview of Datasets, Monitors, and Reports. Retrieved from http://www.caida.org/ data/overview/.

18. National CyberWatch Center. 2017. Mid-Atlantic Collegiate Cyber Defense Competition. Retrieved from https://maccdc.org/.

19. Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set.
20. Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset
21. A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection
22. A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems
23. https://www.unb.ca/cic/datasets/ids-2017.html