Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 9, August 2021: 4805-4815

Evaluate the Detection Model by leveraging theReal Time Dataset from Private Cloud Environment

Pranay Jha[1*], Dr. Ashok Sharma[2], Dr. Mithilesh Kumar Dubey[1]

[1]Lovely Professional University

[2]University of Jammu

*Corresponding author: pranay1988jha@gmail.com (Pranay Jha)

Emails: drashoksharma@hotmail.co.in (Dr. Ashok Sharma), drmithilesh.dwivedi@gmail.com (Dr. Mithilesh Kumar Dubey)

Abstract

Cloud security has risen to prominence as a major concern in recent years. There are numerous security frameworks in use today to protect the environment. Attackers, on the other hand, have always found a way into the organization. In addition to the existing frameworks, the security frameworks must be extended to include the new dataset. Publicly available data sets are no longer effective in detecting new types of assaults and are therefore useless. The UNSW-NB15 data set is used as an offline dataset in this research to train an algorithm to detect malicious network traffic. Furthermore, this work produces its own real-time data collecting by establishing a private cloud environment at the home lab, which serves as a functioning example of the suggested intrusion detection methods. Weused VMware vSphere to generate logs from a real-time environment, as well as malicious and non-malicious traffic on virtual machines in a private cloud environment. This dataset is used as a test data set to assess the performance of the proposed model. The recommended model with UNSW-NB15 and real-time data set has a higher accuracy rate than other existing techniques.

Keywords: *Cloud Security, Distributed IDS, Machine Learning, Intrusion Detection System*

1. Introduction

Cloud security has risen to prominence as a major source of concern for many individuals in recent years. Cybercrime is becoming more prominent as a result of technological improvements over the last generation, as well as the rapid growth of digital mediums such as transportable smartphones and social Media of Things [1]. When people are utterly oblivious of the majority of their daily actions, it becomes stressful. Apart from individuals, a large number of these exploits and assaults are directed at critical infrastructure and organizations. Cybersecurity is a large field of computer science that encompasses study into numerous methods aimed at ensuring data security. Securing the environment requires the use of many frameworks and detection systems [2]. Intrusion detection systems give a method for coping with these attacks by adding another form of security against threats. Along with the increase in the number of commonplace dangers, the data flow on networks can be examined. It is possible to acquire insight into an attacker's behavior by identifying patterns in data from previous attempts. This enables you to distinguish between dangerous and benign packets and hence classify them appropriately [3].

On the other hand, attackers have always been able to infiltrate organizations. Additionally, to the conceptual systems, security frameworks must be extended to include the new dataset. Publicly available data sets are no longer effective at detecting new types of attacks, rendering them useless [4]. This research uses the UNSW-NB15 data set as an offline dataset to develop a classification-based system for detecting malicious network behavior [5]. Additionally, this work collects real-time data by establishing an internal private cloud environment in a home lab, which serves as a functioning demonstration of the suggested intrusion detection algorithms. This study used VMware vSphere to construct a real-time environment, as well as malicious and benign traffic on virtual machines in a private cloud environment. This dataset is used as a test data set to determine the proposed model's performance. It can be downloaded here. The recommended model, which is based on the UNSW-NB15 and real-time data collection, has a greater accuracy rate than other existing techniques. We also collected data from numerous data center endpoints, including Windows servers, Linux servers, ESXi hosts, and other operating systems. We gathered logs from a variety of platforms utilizing vRealize Log Insight, vRealize Network Insight, Syslog Server, System Event Logs, ESXi Host Event Logs, and User Behaviors. We generated the and PCAP files using Kali Linux, an Ubuntu workstation, and a Windows server. We trained the model using the UNSW-NB15 dataset and tested it on real-time data obtained throughout this procedure.

The study defines harmful behavior in five categories: exploit, denial-of-service, probe, generic, and normal [6]. While firewalls are an efficient way to control access to internet resources, attackers have devised a variety of ways to circumvent them. The proposed system is built on a technique for detecting abuse, allowing it to function as an enhanced firewall. As a result, the capabilities of the system are not restricted to those of an IDS [7]. The proposed IDS appears to have the advantage of assisting the administrator in classifying collected data into five categories, one of which is normal, resulting in a lower percentage of false alarms than an anomaly-based IDS.

The proposed paradigm has been successfully implemented in a variety of industrial domains. It is widely utilized in a variety of industries and entails data collection and real-time system monitoring. The proposed IDS is intended to identify harmful attacks automatically. The proposed intrusion detection system (IDS) is capable of collecting and analyzing a variety of factors, such as network traffic and security logs. Additionally, the suggested IDS can determine whether the system has been compromised by evaluating the data and information passing through the system's critical points. Additionally, evidence collection via digital forensics is a crucial area where IDS can be utilized extensively. The updated version of IDS can be used to send an email warning to the administrator or to activate the detection or prevention tool, which records the system's current state. Notably, the obtained system image will comprise the whole contents of the system at the time of the attack. As a result, these photographs can be presented as evidence in court. Thus, it is clear that the suggested IDS may be successfully employed to ensure the security of the information technology

evaluate the detection model by leveraging thereal time dataset from private cloud
environment

environment and in the realm of digital forensics. The proposed technique can be used to defend against several network-based assaults on such systems. Additionally, it applies to all businesses where it will be installed on a computer system to protect the organization’s data. Any suspected behavior will be reported to the security administrator for investigation and possible remediation. Additionally, the proposed methodology applies to securing the Internet of Things (IoT) devices. As a result, our proposed IDS acts as a dog sitter, constantly scanning the internet for potential threats [8].

2. Dataset Preparation

This section discusses the numerous facets of experimentation in detail. This work discusses algorithms for two-class classification. Classification models were developed in four stages: preprocessing, feature selection, model training, and model testing. UNSW NB-15 consists mostly of 47 attributes and two class labels [9]. The dataset contains continuous, discrete, and symbolic features spanning a wide range of values, and so requires pre-processing. Throughout the trial, all nominal characteristics were transformed into integers. Numerical characteristics with a wide range are challenging to manage. As a result, feature scaling was used to reduce their range of possible values. Scaling was not required for Boolean characteristics. To determine the least and greatest values for each attribute in the range [0, 1], min-max normalization was used [10].

2.1. Data Collection

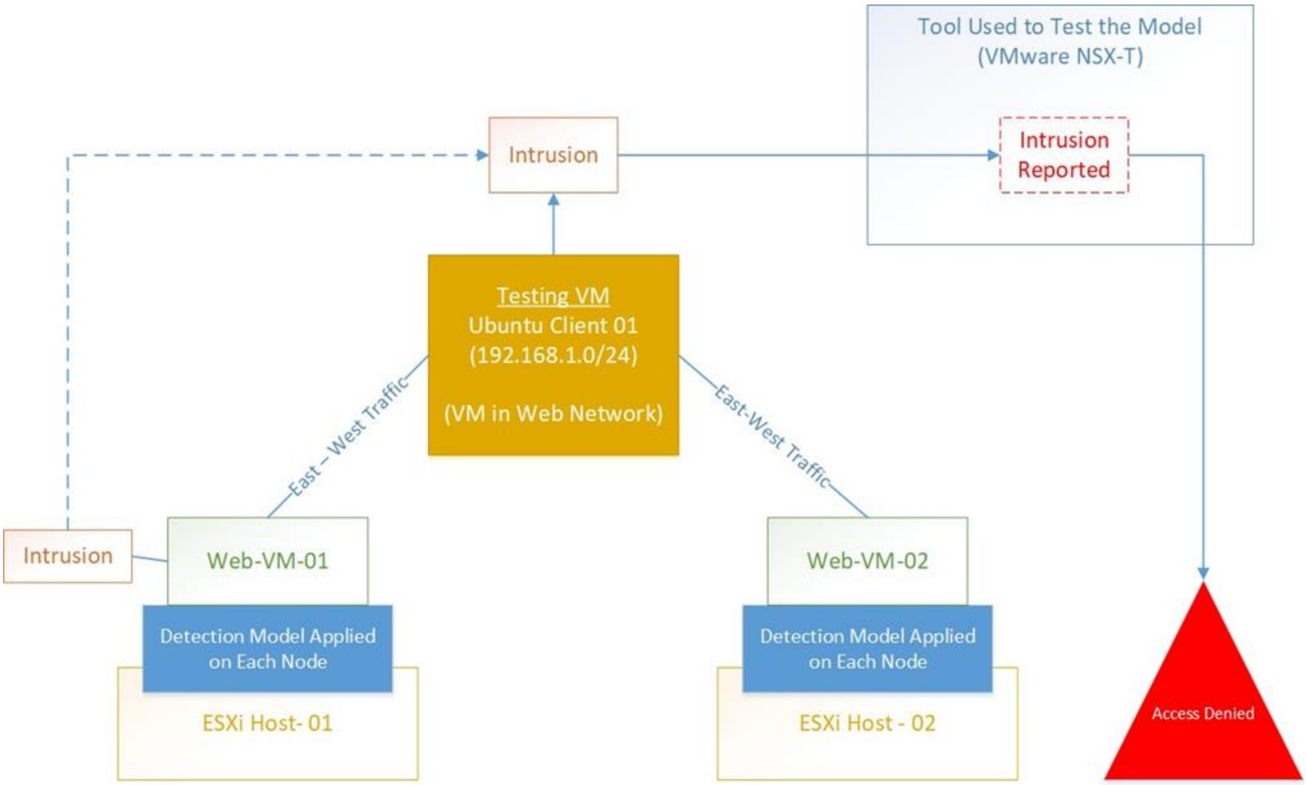
We have collected two types of the dataset for this work. One is from UNSW-NB15, and another is from a real-time environment.

UNSB-NB15 Dataset: The dataset was created with the help of IXIA perfect storm. It is separated into nine categories of modern assault types, and it replicates the flow of traffic regularly. There are 49 features in this data collection, which are classified into different categories. There are numerous datasets available for IDS analyses, including the KDD98, NSLKDD, and others. However, none of these datasets cover the most recent types of attacks. Following the findings of recent IDS research, it has been determined that these datasets do not correctly reflect real-world traffic behavior and current network threats. The characteristics of UNSW-NB15 are divided into several types of flow characteristics, including;

- Fundamental characteristics
- Content characteristics
- Time characteristics
- Additional created features.

Real-Time Dataset: We created our own real-time data collection for this step to test the proposed model. The data set is created by setting up a home lab environment with an Intel NUC machine and virtualization software. This lab is comprised of three ESXi hosts and numerous virtual machines, one of which serves as an attacker and the remainder as a typical user. For 30 days, we monitor the network’s packet flow. On each system, Kali Linux is deployed for the aim of monitoring. Kali Linux is a prominent open-source platform that provides hackers with a suite of security tools. In the lab, we deploy metasploitable operating systems on victim nodes. To generate real-time data, msfconsole (Kali) is employed as an attacker generation on the network, with a metasploitable and purposefully susceptible version of Ubuntu serving as the victim. Here we identified the hardware for ESXi Hosts, vCenter Server, and Distributed IDS [11]. We also ensured that assets are lying in the network and do not have any production impact. We also verified the network ports which are required for packet capturing. The virtual machines which are part of the test should have communication and be reachable to North-South, and East-West traffic. We enabled Wireshark on source and destination IP as per the below figure. We captured the PCAP file for further analysis and documented the detected threats.

Figure 1. Basic architecture of test environment

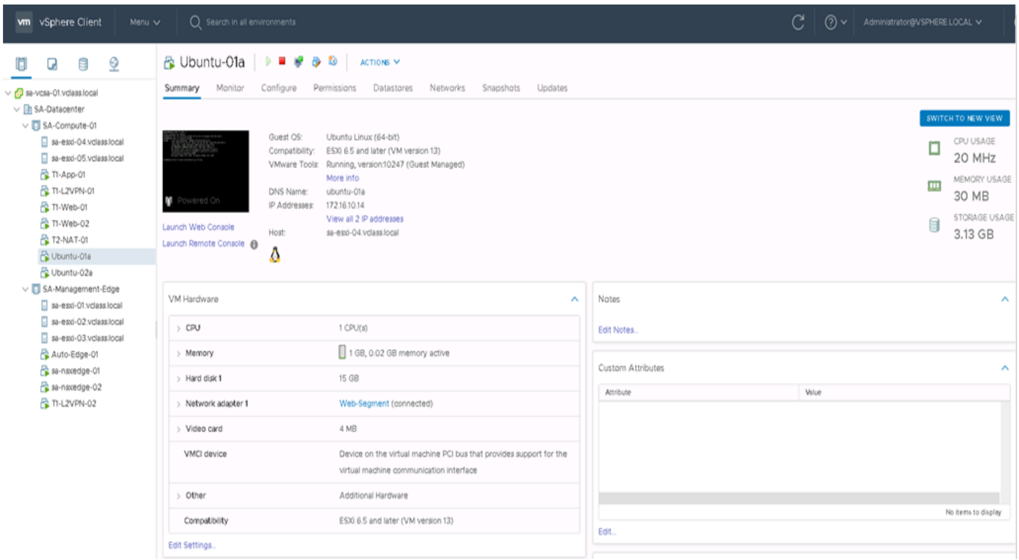


We are using vSphere Client in which we have Two Datacenters as showing in Figure 2. This environment has been used to collect the dataset from a real-time environment.

Figure 3 represents the virtual machines that exist in SA-Datacenter. These virtual machines will be used to generate traffic.

We logged to ubuntu client machines to generate malicious traffic as showing in Figure 4.

evaluate the detection model by leveraging thereal time dataset from private cloud environment

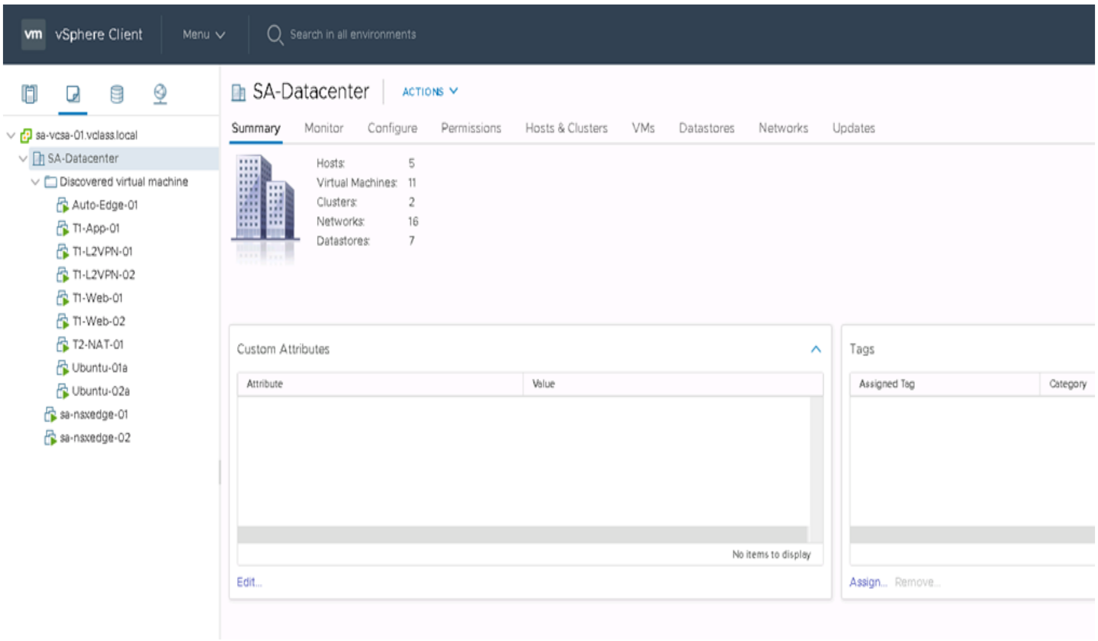


Guest OS: Ubuntu Linux (64-bit)
Compatibility: ESXi 6.5 and later (VM version 13)
VMware Tools: Running, version:10247 (Guest Managed)
DNS Name: ubuntu-01a
IP Addresses: 172.16.10.14
Host: sa-esxi-04.vclass.local

CPU USAGE 20 MHz
MEMORY USAGE 30 MB
STORAGE USAGE 3.13 GB

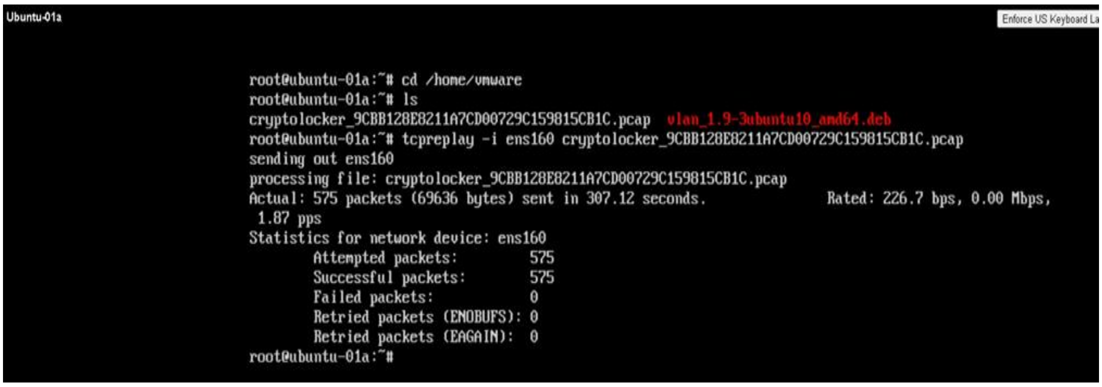
CPU	1 CPU(s)
Memory	1 GB, 0.02 GB memory active
Hard disk 1	15 GB
Network adapter 1	Web-Segment (connected)
Video card	4 MB
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
Other	Additional Hardware
Compatibility	ESXi 6.5 and later (VM version 13)

Figure 2. Private Cloud Infrastructure to collect the dataset



Hosts: 5
Virtual Machines: 11
Clusters: 2
Networks: 16
Datastores: 7

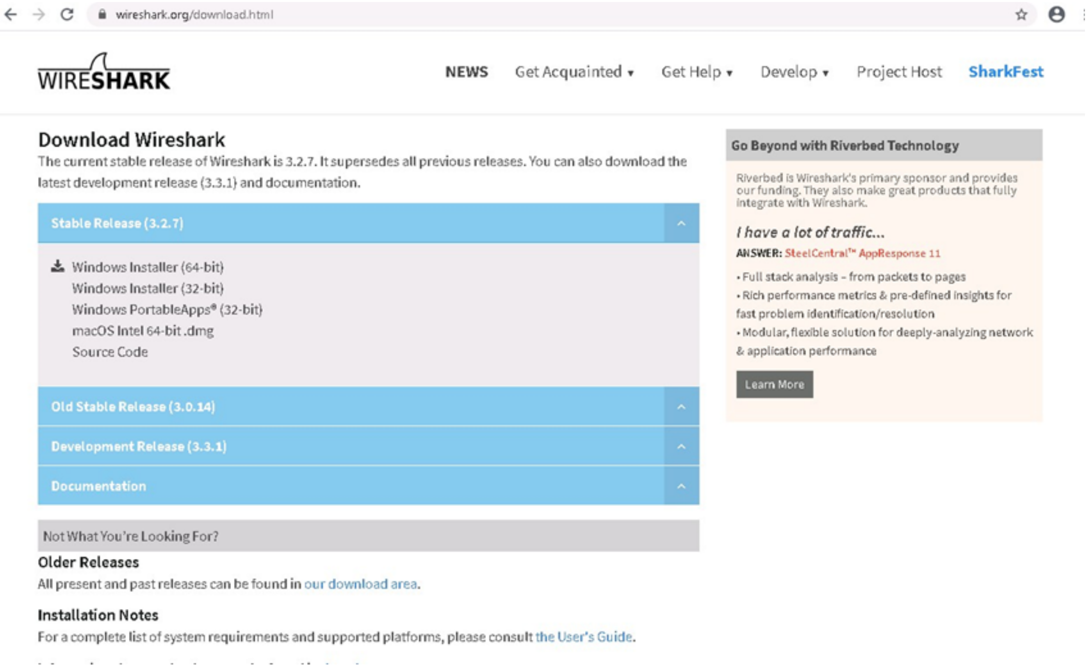
Figure 3. Built Virtual Machines for Client Machines



```
root@ubuntu-01a:~# cd /home/vmware
root@ubuntu-01a:~# ls
cryptolocker_9CBB128E8211A7CD00729C159815CB1C.pcap  vlan_1.9-3ubuntu10_amd64.deb
root@ubuntu-01a:~# tcpreplay -i ens160 cryptolocker_9CBB128E8211A7CD00729C159815CB1C.pcap
sending out ens160
processing file: cryptolocker_9CBB128E8211A7CD00729C159815CB1C.pcap
Actual: 575 packets (69636 bytes) sent in 307.12 seconds.          Rated: 226.7 bps, 0.00 Mbps,
 1.87 pps
Statistics for network device: ens160
        Attempted packets:         575
        Successful packets:        575
        Failed packets:            0
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
root@ubuntu-01a:~#
```

Figure 4. Generate Malicious Traffic from Ubuntu Client Machine

We Installed Wireshark in the test environment to collect the PCAP file, as showing in Figure 5.



Download Wireshark

The current stable release of Wireshark is 3.2.7. It supersedes all previous releases. You can also download the latest development release (3.3.1) and documentation.

Stable Release (3.2.7)

- Windows Installer (64-bit)
- Windows Installer (32-bit)
- Windows PortableApps® (32-bit)
- macOS Intel 64-bit .dmg
- Source Code

Old Stable Release (3.0.14)

Development Release (3.3.1)

Documentation

Not What You're Looking For?

Older Releases

All present and past releases can be found in our download area.

Installation Notes

For a complete list of system requirements and supported platforms, please consult the User's Guide.

Go Beyond with Riverbed Technology

Riverbed is Wireshark's primary sponsor and provides our funding. They also make great products that fully integrate with Wireshark.

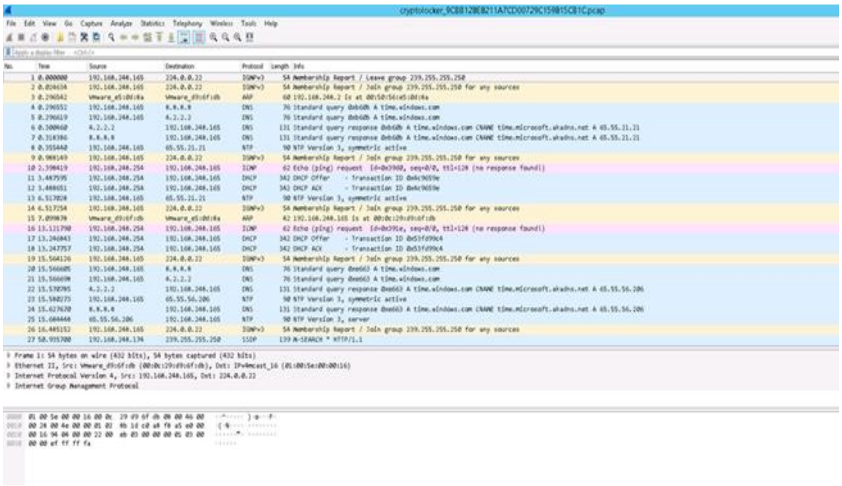
I have a lot of traffic...

ANSWER: SteelCentral™ AppResponse 11

- Full stack analysis – from packets to pages
- Rich performance metrics & pre-defined insights for fast problem identification/resolution
- Modular, flexible solution for deeply-analyzing network & application performance

Figure 5. Installed Wireshark in Test Environment

Figure 6 represents the capture logs using Wireshark.


cryptolocker_9C8B12BEB211A7CD00729C159815C81C.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time        Source            Destination       Protocol Length Info
   1 0.000000    192.168.248.165   224.0.0.22        IGMPv3    54 Membership Report / Leave group 239.255.255.250
   2 0.024634    192.168.248.165   224.0.0.22        IGMPv3    54 Membership Report / Join group 239.255.255.250 for any sources
   3 0.296542    Vmware_e5:0d:4a   Vmware_d9:6f:db   ARP       60 192.168.248.2 is at 00:50:56:e5:0d:4a
   4 0.296552    192.168.248.165   8.8.8.8           DNS       76 Standard query 0xb6db A time.windows.com
   5 0.296619    192.168.248.165   4.2.2.2           DNS       76 Standard query 0xb6db A time.windows.com
   6 0.300460    4.2.2.2           192.168.248.165   DNS      131 Standard query response 0xb6db A time.windows.com CNAME time.microsoft.akadns.net A 65.55.21.21
   7 0.314306    8.8.8.8           192.168.248.165   DNS      131 Standard query response 0xb6db A time.windows.com CNAME time.microsoft.akadns.net A 65.55.21.21
   8 0.355440    192.168.248.165   65.55.21.21       NTP       90 NTP Version 3, symmetric active
   9 0.988149    192.168.248.165   224.0.0.22        IGMPv3    54 Membership Report / Join group 239.255.255.250 for any sources
  10 2.398419    192.168.248.254   192.168.248.165   ICMP      62 Echo (ping) request  id=0x3900, seq=0/0, ttl=128 (no response found!)
  11 3.487595    192.168.248.254   192.168.248.165   DHCP     342 DHCP Offer    - Transaction ID 0x4c9659e
  12 3.488651    192.168.248.254   192.168.248.165   DHCP     342 DHCP ACK      - Transaction ID 0x4c9659e
  13 6.517028    192.168.248.165   65.55.21.21       NTP       90 NTP Version 3, symmetric active
  14 6.517254    192.168.248.165   224.0.0.22        IGMPv3    54 Membership Report / Join group 239.255.255.250 for any sources
  15 7.099870    Vmware_d9:6f:db   Vmware_e5:0d:4a   ARP       42 192.168.248.165 is at 00:0c:29:d9:6f:db
  16 13.121790   192.168.248.254   192.168.248.165   ICMP      62 Echo (ping) request  id=0x391e, seq=0/0, ttl=128 (no response found!)
  17 13.246843   192.168.248.254   192.168.248.165   DHCP     342 DHCP Offer    - Transaction ID 0x53fd99c4
  18 13.247757   192.168.248.254   192.168.248.165   DHCP     342 DHCP ACK      - Transaction ID 0x53fd99c4
  19 15.564126   192.168.248.165   224.0.0.22        IGMPv3    54 Membership Report / Join group 239.255.255.250 for any sources
  20 15.566605   192.168.248.165   8.8.8.8           DNS       76 Standard query 0xe663 A time.windows.com
  21 15.566698   192.168.248.165   4.2.2.2           DNS       76 Standard query 0xe663 A time.windows.com
  22 15.570785   4.2.2.2           192.168.248.165   DNS      131 Standard query response 0xe663 A time.windows.com CNAME time.microsoft.akadns.net A 65.55.56.206
  23 15.580275   192.168.248.165   65.55.56.206      NTP       90 NTP Version 3, symmetric active
  24 15.627670   8.8.8.8           192.168.248.165   DNS      131 Standard query response 0xe663 A time.windows.com CNAME time.microsoft.akadns.net A 65.55.56.206
  25 15.684444   65.55.56.206      192.168.248.165   NTP       90 NTP Version 3, server
  26 16.485152   192.168.248.165   224.0.0.22        IGMPv3    54 Membership Report / Join group 239.255.255.250 for any sources
  27 58.935700   192.168.248.174   239.255.255.250   SSDP     139 M-SEARCH * HTTP/1.1

> Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
> Ethernet II, Src: Vmware_d9:6f:db (00:0c:29:d9:6f:db), Dst: IPv4mcast_16 (01:00:5e:00:00:16)
> Internet Protocol Version 4, Src: 192.168.248.165, Dst: 224.0.0.22
> Internet Group Management Protocol

0000  01 00 5e 00 00 16 00 0c  29 d9 6f db 08 00 46 00
0010  00 28 00 4e 00 00 01 02  4b 1d c0 a8 f8 a5 e0 00
0020  00 16 94 04 00 00 22 00  eb 03 00 00 00 01 03 00
0030  00 00 ef ff ff fa
```

Figure 6. Review of captured logs using Wireshark

The collected dataset has been stored in a Hardrive for further processing which has been discussed in the next sub- sections.

2.2. Data Pre-Processing

Pre-processing is associated with data cleansing. It comprises the removal of redundant characteristics, those which do not contribute to the high IG, and their replacement with derived factors extracted from many other features in the data.

Bearing in mind that some machine learning models require data in a particular format, e.g., the

Random Forest Method does not allow null values. records containing null values must be eliminated or replaced with substitute values [12]. This issue can be remedied by the use of imputation. Additionally, certain machine learning algorithms are incapable of processing data types apart from integers and floats. This interoperability issue can be resolved by typecasting the values or by eliminating the non-compliant functionality. Another critical aspect of data pre-processing is that the data should be consistent with several algorithms to ensure consistency and to reduce computation complexity [13]. The following pre-processing was performed on the UNSW-NB15 data set:

2.2.1. Imbalanced Dataset

Imbalance refers to an erroneous class assignment within the data collection. A data imbalance results in a skewed classification. This issue is evident in the UNSW-NB15 data set. Normal packets account for more than 87 percent of the data set's total flow. To address this issue, we employ an approach comparable to under-sampling an unbalanced data set. We reduced the number of normal packets by 50% while maintaining the number of packets of many other classes at their original level. The remaining statistics now account for 60% of the total.

2.2.2. Missing Value Imputation

We observed that some of the data in the collection had missing values. Missing data can inject significant bias into data administration and interpretation, making it more difficult to handle and evaluate the data and decreasing its accuracy. Without a doubt, the three features with the highest frequency of missing values are the following: the ct flw HTTP mod, the is FTP login and the ct FTP cmd. Records with a high percentage of missing values for one characteristic are more likely than other records to have a high percentage of missing values for one or more other characteristics as well. Currently, we have two options for resolving this problem. We had the option of discarding these samples or doing imputation on them. Since eliminating attributes will affect the accuracy of the solution, we applied imputation.

2.2.3. Feature Extraction & Selection

Feature extraction is a fundamental topic in machine learning that has a significant impact the accuracy for prediction. The data attributes used to train machine learning models to have a significant impact on our results. By utilizing feature extraction, we can increase accuracy and also reduce training time by eliminating over-fitting [14]. In this work, we employ the feature significance method in conjunction with Random Forest. Several features were eliminated during preparing the data type, the last two of which are labels as binary and multi-class which as different types of attacks. It is discovered that a few characteristics contribute the most to classification. This may result in over-fitting. To address this, we eliminated the top most important features, namely sload, sttl, sload, smean, and ct_state_ttl. Together, these five traits store over 70% of the variance. We used imputation techniques and calculated the feature's relevance. It is seen that the top features obtained using each of the three imputation strategies are identical [15].

2.2.4. Splitting the Data

UNSW-NB15 dataset has been divided into two parts. One set is for training purposes and another set is for testing purposes. The ratio of the dataset is 80:20. Data with 80% is used for training the model, and the rest of the 20 percent has been used for testing the model. We have also used real-time data to test the model after the training is done through the UNSW-NB15 dataset.

3. Implementation & Result

3.1. Experimental Setup

This experiment is performed on Intel NUC Box which is Intel i7 CPU, 64 GB of Memory, and 2 TB of Hard drive on each physical server. We utilized 5 NUC Boxes to set up a private cloud environment. Various Vmware tools have been used to create an Infrastructure that includes ESXi, vCenter Server, vRealize Network Insight, and Virtual Machines. Each virtual machine has Kali Linux installed for generating malicious and normal attacks in the environment. After preparing the infrastructure, we proceeded with the data collection phase and performed the experimental part using several classifiers. The evaluation matrix and result section are discussed in the next sub-sections.

3.2. Evaluation Matrix

Several metrics are considered when evaluating the effectiveness of model which is trained on the UNSW-NB 15 dataset. The metrics including TP, TN, FP, FN. TP Stands for True Positive, TN stands for True Negative, FP stands for False Positive, FN stands for False Negative. FP counts the total number of incorrectly classified attacks, and FN denotes the number of incorrectly classified non-attack rows. TP indicates the number of relevant classified attacks, TN indicates the number of relevant classified.

We define memory, accuracy, precision, and F1 score in the following ways:

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

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2\ X\ Precision\ X\ Recall}{Precision + Recall}$$

3.3. PerformanceEvaluatin

evaluate the detection model by leveraging thereal time dataset from private cloud environment

This section summarises the findings from the initial stage of the study, referred to as the preliminary findings. It was suggested that the models work well when data is collected in real-time. When the proposed model is supplied with the real-time data set, its performance can be evaluated. The confusion matrix in Table 1 illustrates the numerous types of assaults that could be launched against the proposed IDS technology. Comparison analysis of different type of attacks are showing in Figure 8. Table 2 shows the precision and recall as well as other metrics values for real-time datasets using the proposed model. Comparison analysis of Model performance is represented in Figure 9. In comparison to DoS and Exploit, Normal and Probe had the highest precision and recall levels. Additionally, DoS has a higher recall than Exploit. For the five categories depicted in the figure, the accuracy is 93.8 percent, and the ADR is 98.29 percent. Based on these findings, it can be concluded that the proposed model performs adequately on the real-time dataset. Random Forest outperformed its competition when used as a binary classification method, obtaining an accuracy of 93.8 percent. We believe it has the potential to dramatically improve accuracy.

Table 1. Confusion Matrix

	Probe	**Exploit**	**Generic**	**DoS**	**Normal**
Probe	1770	70	10	0	70
Exploit	0	1970	180	0	30
Generic	0	100	1430	60	190
DoS	0	370	0	1390	0
Normal	10	50	0	20	1820

4. Conclusion

This work provides a comprehensive classification-based detection system which assesses its efficacy on both publicly available and real-time data sets. The UNSW-NB15 data set is used as an offline dataset in this research to train an algorithm to detect malicious network traffic. Furthermore, this work produces its own real-time data collecting by establishing a

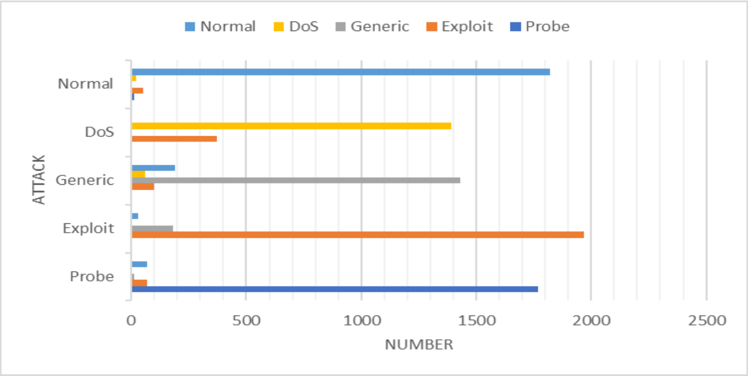


Figure 7. Comparative Analysis of various attacks

Table 2. Performance Evaluation of Model

	Recall	**Precision**	**F1-Score**
DoS	76.93	96.55	88.12
Exploit	92.36	75.94	84.15
Normal	96.89	88.11	92.88
Generic	82.44	89.72	86.13
Probe	94.13	99.54	96.76

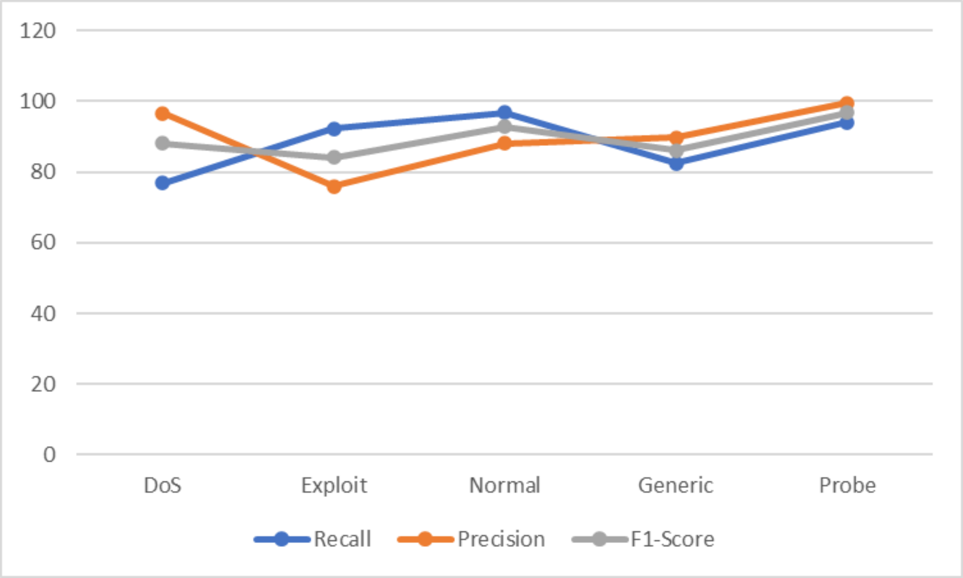


Figure 8. Comparison Analysis of Model Performance

private cloud environment at the home lab, which serves as a functioning example of the suggested intrusion detection methods. We used VMware vSphere to generate logs from a real-time environment, as well as malicious and non-malicious traffic on virtual machines in a private cloud environment. This dataset is used as a test data set to assess the performance of the proposed model. The value of numerous evaluation metrics is discovered to perform better than that of other current classic models. Once the attack is carried out, the proposed detection model has access to the new signatures. Our IDS model has been upgraded to include a signature that will prevent attacks from these categories. The recommended model with UNSW-NB15 and real-time data set has a higher accuracy rate than other existing techniques. The proposed model is 93.8 percent accurate. This proposed integrated model functions as the best algorithm for detecting the malicious Behaviour in the Cloud environment.

References

[1] A Albugmi, M O Alassafi, R Walters, and G Wills. “Data security in cloud computing”, *2016 Fifth International Conference on Future Generation Communication Technologies (FGCT)*, pages 55–59, 2016.

[2] Abdulaziz Aldribi, Issa Traoré, Belaid Moa, and Onyekachi Nwamuo. “Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking”, *Computers & Security*, 88, 101646–101646, 2020.

[3] Douglas Goltz, Michael Attas, Gregory Young, Edward Cloutis, and Maria Bedynski. “Assessing stains on historical documents using hyperspectral imaging”, 2010.

[4] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. “Survey of intrusion detection systems: techniques, datasets and challenges”, *Cybersecurity*, 2(1), 20–20, 2019.

[5] N Moustafa and J Slay. “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set”, *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6, 2015.

[6] Santosh Aditham and Nagarajan Ranganathan. “A System Architecture for the Detection of Insider Attacks in Big Data Systems”, *IEEE Transactions on Dependable and Secure Computing*, 15(6), 974–987, 2018.

[7] Shadi Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein. “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model”, *Journal of Computational Science*, 25, 152–160, 2018.

[8] P Jha and A Sharma. “Framework to Analyze Malicious Behaviour in Cloud Environment using Machine Learning Techniques”, *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–12, 2021.

[9] ‘ ‘ Ids ; | Datasets, | Research | Canadian Institute For Cybersecurity, and | Unb, 2017.

[10] Lee Friedman and Oleg V. Komogortsev. “Assessment of the Effectiveness of Seven Biometric Feature Normalization Techniques”, *IEEE Transactions on Information Forensics and Security*, 14(10), 2528–2536, 2019.

[11] R Perry and B Waldman.

[12] T Ahmad and M N Aziz.

[13] A Behl and K Behl. “Security Paradigms for Cloud Computing”, *2012 Fourth International Conference on Computa- tional Intelligence, Communication Systems and Networks*, pages 200–205, 2012.

[14] V Kantorov and I Laptev, 2014.

[15] X Dong, J Yu, Y Zhu, Y Chen, Y Luo, and M Li. “SECO: Secure and scalable data collaboration services in cloud computing”, *Comput. Secur*, 50, 91–105, 2015.