INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 162-1671

e-ISSN : 2583-1062

Impact Factor: 5.725

A SECURE FEDERATED INTRUSION DETECTION MODEL WITH BLOCKCHAIN AND DEEPLEARNING

V. Swetha[1], Macherla Sreeya[2], Shaik Talha[3], Gandi Nikshith Kumar[4]

[1]Assisstant Professor, Information Technology, ACE Engineering College, India.

[2,3,4] Information Technology, ACE Engineering College, India.

ABSTRACT

Cloud computing is a technology that efficiently uses computing infrastructures, but it also poses threats to intruders who exploit the complex and distributed nature of these infrastructures. Cyber-attacks are becoming more complex, making it difficult to detect intrusions effectively. Despite the development of various Deep Learning approaches, security issues remain, especially in federated cloud computing domains. This work proposes a Secure Federated Intrusion Detection Model Version 1 (SecFedIDM-V1) using blockchain technology and Bidirectional Long Short-Term Memory (BiLSTM) Recurrent Neural Network (RNN). The model uses the Cobourg Intrusion Detection Dataset (CIDDS) for training, testing, and validation. The SecFedIDM-V1 is deployed as a Python-based web application that captures network packets for classifying attacks into normal or attack types. The 80:10:10 BiLSTM network outperforms the GRU in the evaluation results.

Keywords: Block Chain, Deep Learning, Intrusion Detection, Cyber attacks, BiLSTM, CIDDS

1. INTRODUCTION

Cloud computing offers IT-based services to geographically dispersed locations via the Internet, allowing users to pay only for what they consume, converting capital goods into other operational expenses. Virtually shared resources in cloud computing include computation and storage resources, software applications, operating systems, and network infrastructures. Over $1 trillion has been devoted to cloud-based computing, offering benefits such as extensibility, resource allocation based on need, less management overhead, an adjustable pricing system, and easier application development and delivery. Cloud federation, which consists of services from several providers aggregated in a single pool, supports resource migration, resource redundancy, and the combination of complementary resources or services. Primary cloud service models include Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Softwareas-a-Service (SaaS). As technology advances, the number of hacking incidents is on the rise, and security issues have become a major challenge in a complicated technical world. To prevent and secure IT cloud infrastructure and operations, active Intrusion Detection Systems or Models (IDS/M) are essential. Blockchain technology has permeated Information and Communication Technology (ICT) and its usage has increased due to the value of cryptocurrencies and venture capital investments in blockchain start-up companies. Blockchain uses a ledger to record and share transactions among participants, enabling parties to do business without a trusted third party.

2. OBJECTIVES

Various experts have proposed using Machine Learning (ML) algorithms to develop IDMs to improve the security of computation and network resources. Machine Learning emulates human cognitive abilities through modelling deduction, inference, extrapolation, and synthesis. It could be leveraged to build IDMs thereby enabling precise identification of malicious traffic, leading to fewer false positive alerts. Common ML algorithms that have been used for developing IDMs include Support Vector Machines (SVM), Decision Trees (DT), Bayesian Networks, Naïve Bayes etc.. However, the foregoing traditional ML methods focus on the manual engineering of features and are less efficient in the presence of enormous data that needs classification in a production environment. Multiple classification tasks reduce accuracy as dataset size grows. In order to overcome the high data analysis challenges of traditional ML algorithms, Deep Learning(DL) algorithms are employed to improve IDMs performance, especially in multi-class scenarios

3. METHODOLOGY

The Secure Federated Intrusion Detection Model (SecFedIDM-V1) uses Bidirectional Long Short-Term Memory (LSTM) to detect and classify malicious activities on networks. The model stores malicious packet details in a private blockchain ledger, accessible only to certified nodes. Blockchain addresses privacy and safety concerns by allowing peer data sharing over a decentralized ledger network. Validators, like miners, replace third parties for decentralized transaction validation. Distributed consensus allows individuals to agree on verifying digital transactions without the need for a reliable third party. This method analyzes packet vectors to obtain fine-grained features for intrusion traffic detection. The attention mechanism generates features that are imported into a fully connected layer for feature fusion, obtaining key features that accurately characterize network traffic behaviors.

4. LITERATURE SURVEY

Talaei Khoei and Kaabouch developed a novel outlier-based method for detecting zero-day cyberattacks, achieving detection accuracy of 90.01% for DoS (GoldenEye), 98.43% for DoS (Hulk), 90.01% for port scanning, and 99.67% for DDoS attacks. Sharafaldin et al. proposed the Hierarchical Deep Learning System based on Big Data (BDHDLS), which employs behavioral and content features to understand network traffic characteristics and payload information. The model was trained on the Information Centre of Excellence for Tech Innovation (ISCX) Intrusion Detection Assessment dataset (ISCXIDS2012) and achieved an accuracy rate of 99.5%. Lee et al. developed a paradigm for adaptive ensemble learning, using a multi-tree algorithm and multiple basic classifiers such as Decision Trees, Random Forests, K-Nearest Neighbors (KNN), and Deep Neural Networks (DNN).

Ring proposed a CNN–IDS-based network intrusion detection model, which achieved an accuracy of 99.23%. Chattopadhyay et al. proposed an IDS using Core Vector Machines (CVM), a data mining-based classifier with a lower false positive rate and computational overhead. The classifier was trained and tested on the KDDCup’99 dataset, achieving a detection rate of 99% and a false positive rate of 27%. Shen et al. presented a Genetic Algorithm (GA) and SVM-based alarm intrusion detection algorithm for use in a human-controlled IDS, improving population search efficiency and information sharing.

In summary, various researchers have developed innovative methods for detecting zero-day cyberattacks, including Khoei and Kaabouch's outlier-based method, Sharafaldin et al.'s BDHDLS, Lee et al.'s adaptive ensemble learning, Ring's CNN–IDS-based model, Chattopadhyay et al.'s CVM-based IDS, and Shen et al.'s GA-based alarm intrusion detection algorithm.

5. PROPOSED SYSYTEM

This project proposes a Secure Federated Intrusion Detection Model (SecFedIDM-V1). The model uses Bidirectional Long Short-Term Memory (LSTM), a deep learning algorithm, to detect and classify malicious activities on the network by monitoring incoming network packets. The details of the packets classified as malicious are stored in a blockchain ledger in a private network where only certified nodes can access it, and the ledger serves as a signature database.

6. HARDWARE AND SOFTWARE REQUIREMENTS

6.1 HARDWARE REQUIREMENTS:

System	: Intel(R) Core™ i3-7020U CPU@ 2.30GHz
Hard Disk	: 1 TB.
Input Devices	: Keyboard, Mouse
Ram	: 4 GB.

6.2 SOFTWARE REQUIREMENTS:

Operating system	:	Windows XP/7/10.
Coding Language	:	Python
Tool	:	Anaconda
Interface	:	OPENCV

7. PACKAGES USED

Here's a concise list of packages used in the code, suitable for the project:

Built-in:

- os
- datetime
- hashlib
- random
- base64
- sys

Third-Party:

- Crypto.Cipher (likely PyCryptodome)

- Flask
- werkzeug.utils
- ecdsa
- sendmail (custom or third-party)
- RSA (custom)
- main (custom)
- eval (custom, use with caution)
- NumPy

8. TECHNOLOGY DESCRIPTION

SecFedIDM-V1 leverages a combination of Python and various technologies to achieve its secure federated intrusion detection functionality. Python, known for its readability and extensive libraries, serves as the foundation for development. Libraries like NumPy can be used for data analysis, while functionalities like web development (through Flask, if used) might be relevant for user interaction. Cryptographic libraries like Crypto.Cipher and potentially custom implementations (ECDSA, RSA) are crucial for ensuring data integrity and secure communication within the blockchain network, a core component of SecFedIDM-V1. This blockchain, potentially implemented through the `generateblockchain` function, facilitates secure and distributed storage of intrusion detection data. Additionally, the project might utilize a Deep Bidirectional LSTM network (potentially within the `eval` module) for advanced intrusion detection analysis using machine learning. However, caution is advised when using `eval` due to security risks. While the exact purpose of the `cloud` module remains unclear, the overall combination of Python and these technologies empowers SecFedIDM-V1 to be a secure, scalable, and potentially machine learning-driven solution for intrusion detection.

9. SOURCE CODE

```
import os
import datetime
import hashlib
import Crypto.Cipher
from flask import Flask, session, url_for, redirect, render_template, request, abort, flash
from werkzeug.utils import secure_filename
import random
import base64
from hashlib import sha256, md5
import ecdsa
import os
import sys
from sendmail import sendmail
from ecdsa import SigningKey, VerifyingKey
from RSA import encrypt,decrypt,generate
from main import generateblockchain
#from cloud import uploadFile,downloadFile,close
from eval import main
import NumPy as np
app = Flask(__name__)
app.secret_key = os.urandom(24)
@app.route("/")
def FUN_root():
return render_template("index.html")
@app.route("/owner")
def FUN_admin():
```

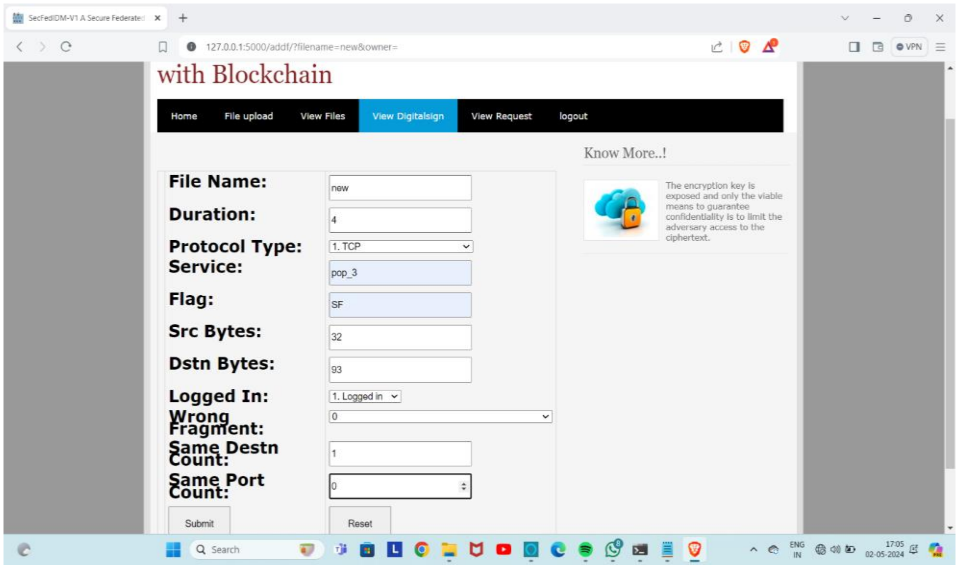
```
return render_template("owner.html")
@app.route("/bhome")
def bhome():
return render_template("bhome.html")
@app.route("/ownerlogact",methods = ['GET','POST'])
def owner_logact():
if request.method == 'POST':
status=owner_login(request.form['username'],request.form['password'])
if status:
session['username'] = request.form['username']
return render_template("ownerhome.html",m1="sucess")
else:
return render_template("owner.html",m1="sucess")
@app.route("/serverlogact",methods = ['GET','POST'])
def serverlogact():
if request.method == 'POST':
status=server_logact(request.form['username'],request.form['password'])
if status:
session['username'] = request.form['username']
return render_template("shome.html",m1="sucess")
else:
return render_template("server.html",m1="sucess")
@app.route("/bchainlogact",methods = ['GET','POST'])
# Globals begin.
hmsg=''
ds =''
pvk = ''
pbk = ''
# Globals end.
@app.route("/Gensig/", methods = ['GET' , 'POST'])
def owner_split():
fname = request.args.get('fname')
owner = request.args.get('owner')
data = request.args.get('data')
#RSA ORIGHINAL
key_pair = generate(8)
print('Generated Key pairs')
public_key = key_pair["public"]
private_key = key_pair["private"]
print(key_pair["public"])
print(key_pair["private"])
datas,key=generateblockchain('firstblock',data)
print("blockchain")
@app.route("/bviewencfiles")
def bviewencfiles():
digitalsigndata = bserver_viewdata()
return render_template("bviewencfiles.html",spliinfo = digitalsigndata)
```

```
@app.route("/vuserreq")
def owner_vuserreq():
userrequest = user_request()
return render_template("vuserreq.html",userreqdata = userrequest)
@app.route("/logout")
def FUN_logout():
return render_template("index.html")
@app.route("/response", methods = ['GET','POST'])
def owner_response():
fname1 = request.args.get('filename')
data1 = request.args.get('data')
owner1 = request.args.get('owner')
email1 = request.args.get('email')
result = owner_request(fname1,owner1,email1)
rdata = result
status = owner_update(rdata,fname1,owner1,email1)
if status == True:
for hsmsg,pvk,blockkey in rdata:
print(hsmsg,pvk)
sendmail(email1,pvk,blockkey)
return render_template("vuserreq.html")
@app.route("/ownerhome")
def FUN_ownerhome():
return render_template("ownerhome.html")
@app.route("/Upload", methods = ['GET','POST'])
def owner_upload():
if request.method == 'POST':
file = request.files['inputfile']
check = upload_file(request.form['fname'],file,session['username'])
print(check)
if check:
return render_template("fileupload.html",m1="success")
else:
return render_template("fileupload.html",m1="Failed")
@app.route("/fileupload")
def FUN_fileupload():
return render_template("fileupload.html”)
@app.route("/ownerviewfiles")
def FUN_ownerviewfiles():
viewdata = owner_viewfiles(session['username'])
if viewdata:
return render_template("ownerviewfiles.html",showdata = viewdata)
else:
return render_template("vuserreq.html",m1="false")
@app.route("/verify1", methods = ['GET','POST'])
def user_verfiy()
filename =request.form['filename']
```

```
hsmsg = request.form['hashmsg']
publickey = request.form['publickey']
BlockHash = request.form['BlockHash']
status,datas = algo(filename,hsmsg,publickey,BlockHash)
if status == True:
for data in datas:
return render_template('successfulV.html',finaldata=data) .
else:
return render_template('failed.html') # A redirect added.
@app.route("/logout")
def admin_logout():
return render_template("index.html")
if __name__ == "__main__":
app.run(debug=True,host='127.0.0.1', port=5000)
```

10. OUTPUT

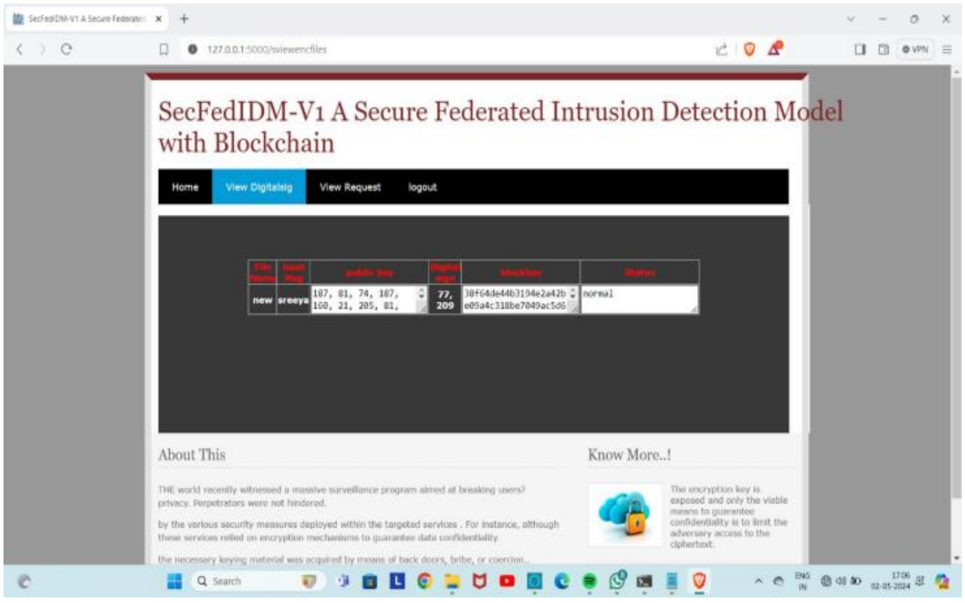
The final results of the project:



File Name:	new
Duration:	4
Protocol Type:	1. TCP
Service:	pop_3
Flag:	SF
Src Bytes:	32
Dstn Bytes:	93
Logged In:	1. Logged in
Wrong Fragment:	0
Same Destn Count:	1
Same Port Count:	0

The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

Fig 10.1 Values Of Normal Attack


SecFedIDM-V1 A Secure Federated Intrusion Detection Model with Blockchain
Home
View Digitalsig
View Request
logout
About This
Know More..!

File Name	hash Msg	public key	Digital sign	blockhas	Status
new	sreeya	187, 81, 74, 187, 160, 21, 205, 81,	77, 209	38f64de44b3194e2a42b e09a4c318be7049ac5d6	normal

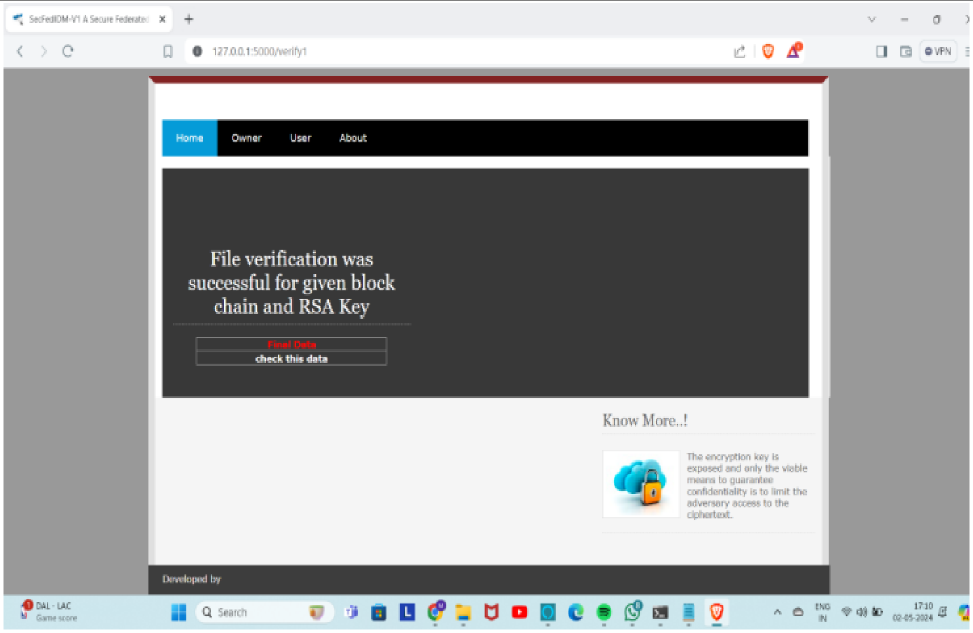
THE world recently witnessed a massive surveillance program aimed at breaking users' privacy. Perpetrators were not hindered.

by the various security measures deployed within the targeted services . For instance, although these services relied on encryption mechanisms to guarantee data confidentiality

the necessary keying material was acquired by means of back doors, bribe, or coercion.

The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

Fig:10.2 Normal Attack


Home Owner User About

File verification was successful for given block chain and RSA Key

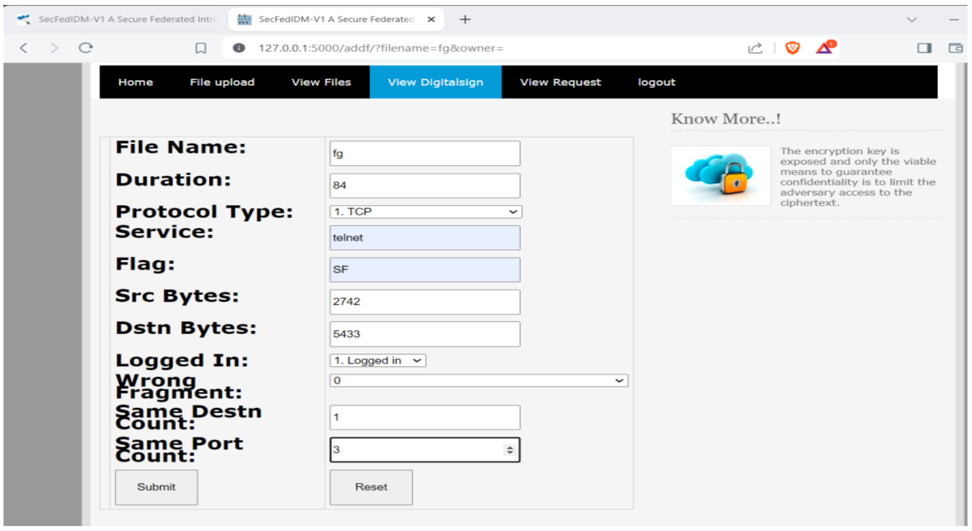
Final Data
check this data

Know More..!

The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

Developed by

Fig:10.3 Actual Data In File


Home File upload View Files View Digitalsign View Request logout

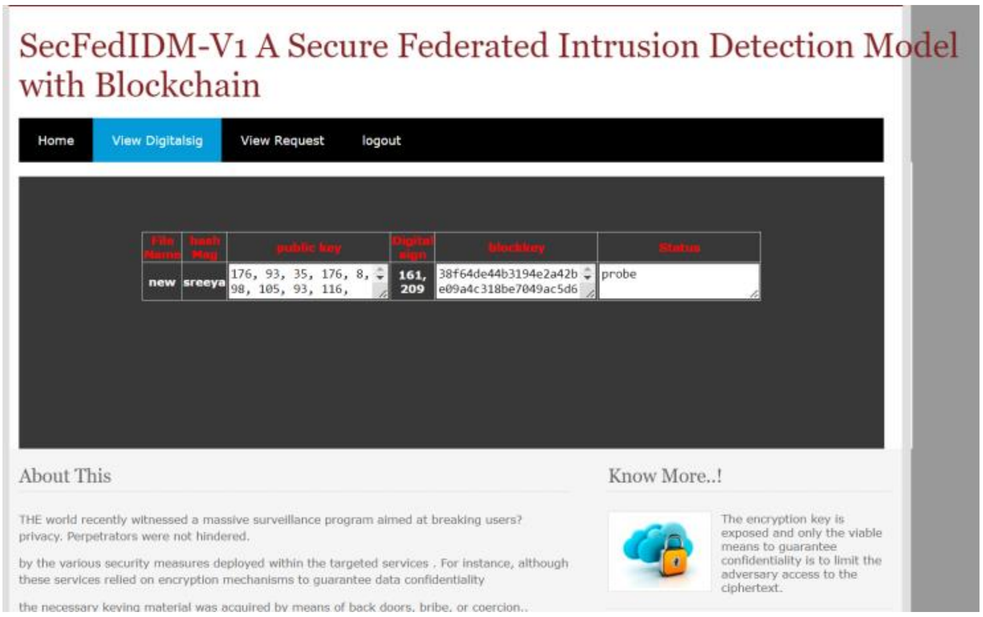
File Name:	fg
Duration:	84
Protocol Type:	1. TCP
Service:	telnet
Flag:	SF
Src Bytes:	2742
Dstn Bytes:	5433
Logged In:	1. Logged in
Wrong Fragment:	0
Same Destn Count:	1
Same Port Count:	3

Submit Reset

Know More..!

The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

Fig:10.4 Values of Probe Attack


SecFedIDM-V1 A Secure Federated Intrusion Detection Model with Blockchain

Home View Digitalsig View Request logout

File Name	hash Msg	public key	Digital sign	blockkey	Status
new	sreeya	176, 93, 35, 176, 8, 98, 105, 93, 116,	161, 209	38f64de44b3194e2a42b e09a4c318be7049ac5d6	probe

About This

THE world recently witnessed a massive surveillance program aimed at breaking users? privacy. Perpetrators were not hindered.

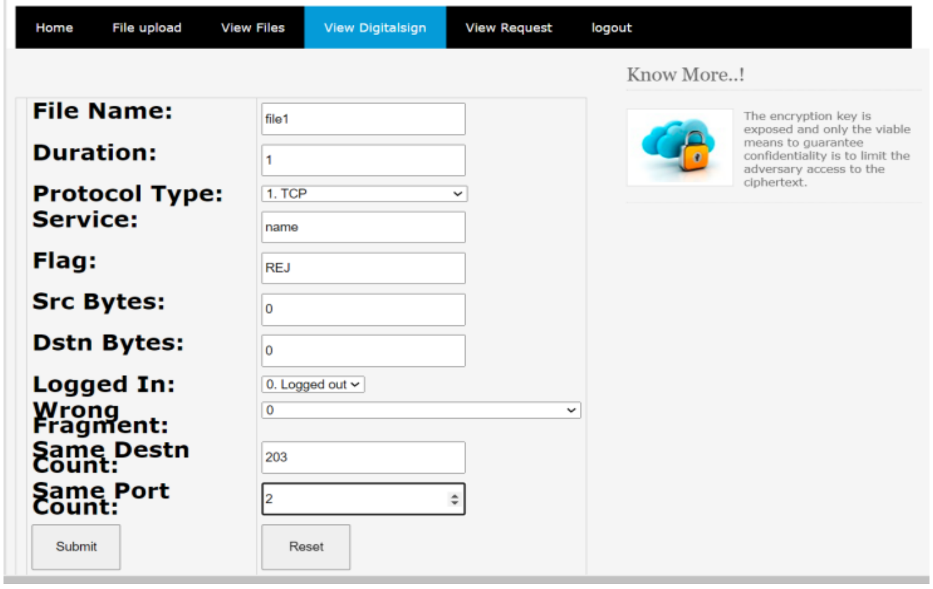
by the various security measures deployed within the targeted services . For instance, although these services relied on encryption mechanisms to guarantee data confidentiality

the necessary keying material was acquired by means of back doors, bribe, or coercion..

Know More..!

The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

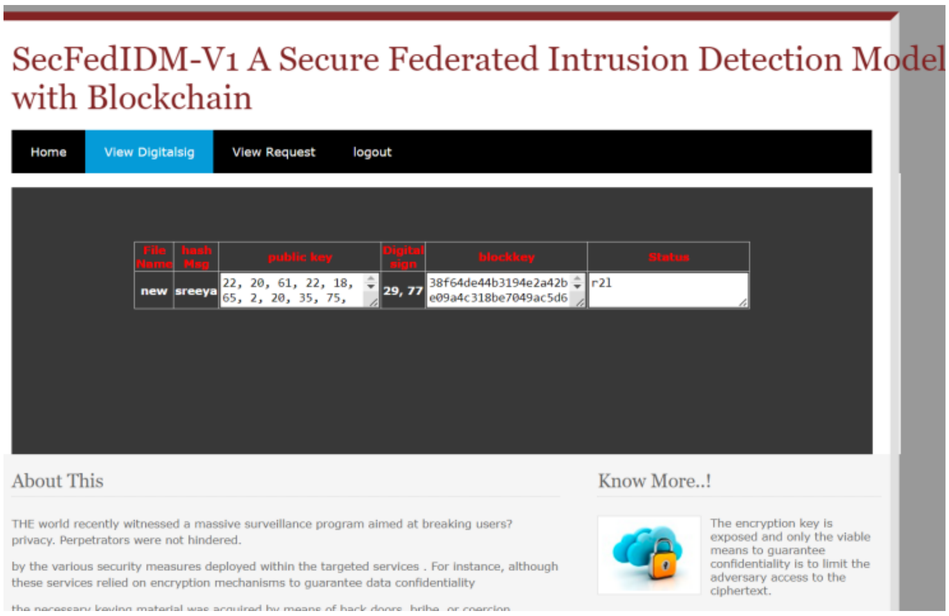
Fig:10.5 Probe Attack



File Name:	file1
Duration:	1
Protocol Type:	1. TCP
Service:	name
Flag:	REJ
Src Bytes:	0
Dstn Bytes:	0
Logged In:	0. Logged out
Wrong Fragment:	0
Same Destn Count:	203
Same Port Count:	2

The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

Fig:10.6 Values Of r21 Attack



SecFedIDM-V1 A Secure Federated Intrusion Detection Model with Blockchain

File Name	hash Msg	public key	Digital sign	blockkey	Status
new	sreeya	22, 20, 61, 22, 18, 65, 2, 20, 35, 75,	29, 77	38f64de44b3194e2a42b e09a4c318be7049ac5d6	r21

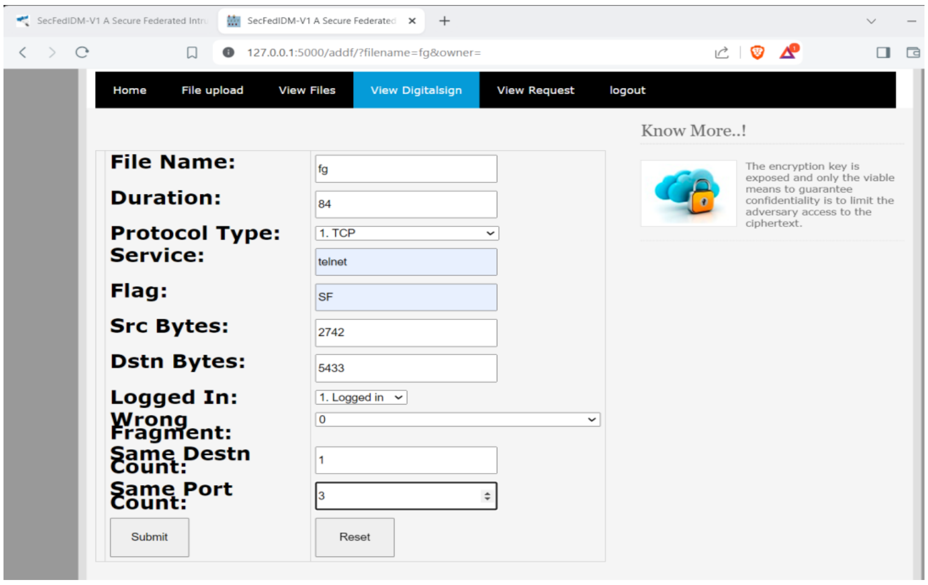
THE world recently witnessed a massive surveillance program aimed at breaking users? privacy. Perpetrators were not hindered.

by the various security measures deployed within the targeted services . For instance, although these services relied on encryption mechanisms to guarantee data confidentiality

the necessary keying material was acquired by means of back doors, bribe, or coercion

The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

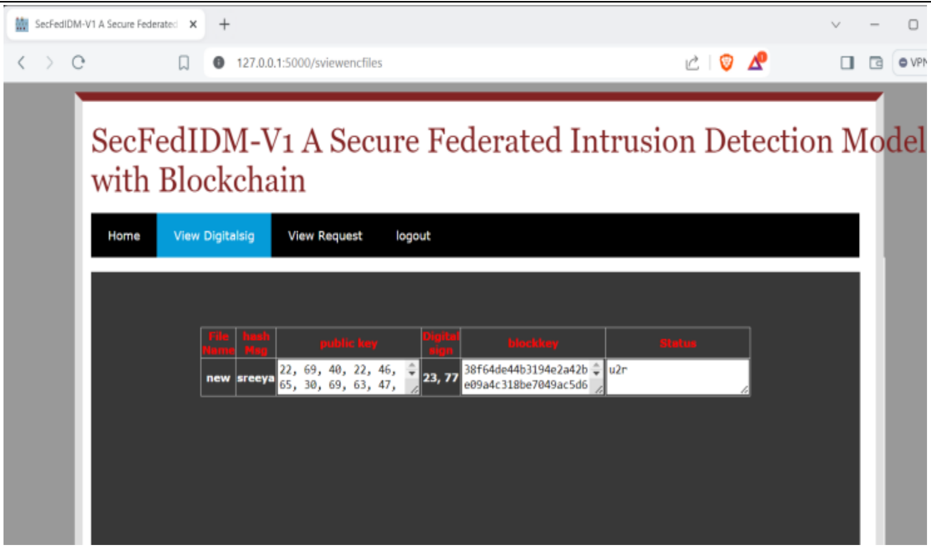
Fig:10.7 r21Attack


SecFedIDM-V1 A Secure Federated Intr
SecFedIDM-V1 A Secure Federated
127.0.0.1:5000/addf/?filename=fg&owner=
Home
File upload
View Files
View Digitalsign
View Request
logout
Know More..!
Submit
Reset

File Name:	fg
Duration:	84
Protocol Type:	1. TCP
Service:	telnet
Flag:	SF
Src Bytes:	2742
Dstn Bytes:	5433
Logged In:	1. Logged in
Wrong Fragment:	0
Same Destn Count:	1
Same Port Count:	3

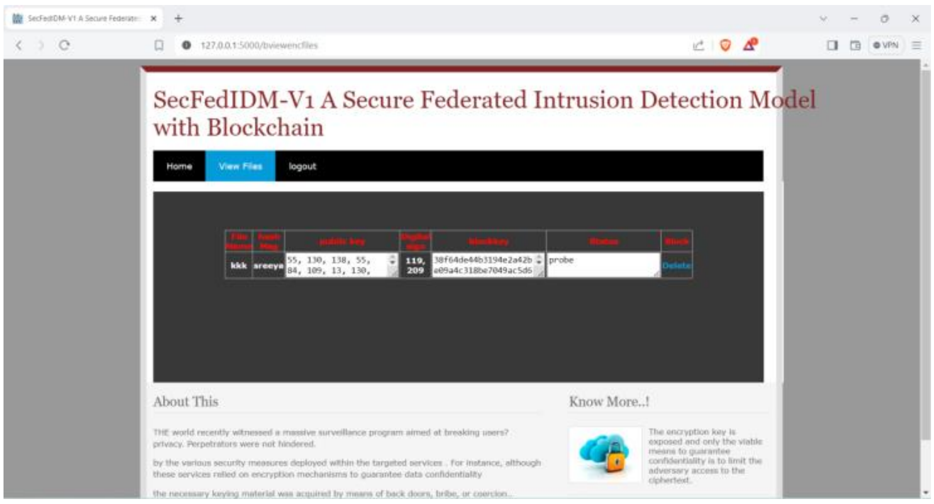
The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

Fig:10.8 Values Of u2r Attack


SecFedIDM-V1 A Secure Federated
127.0.0.1:5000/sviewencfiles
SecFedIDM-V1 A Secure Federated Intrusion Detection Model with Blockchain
Home
View Digitalsig
View Request
logout

File Name	hash Msg	public key	Digital sign	blockkey	Status
new	sreeya	22, 69, 40, 22, 46, 65, 30, 69, 63, 47,	23, 77	38f64de44b3194e2a42b e09a4c318be7049ac5d6	u2r

Fig:10.9 u2r Attack


SecFedIDM-V1 A Secure Federate
127.0.0.1:5000/bviewencfiles
SecFedIDM-V1 A Secure Federated Intrusion Detection Model with Blockchain
Home
View Files
logout

File Name	hash Msg	public key	Digital sign	blockkey	Status	Block
kkk	sreeya	55, 130, 138, 55, 84, 109, 13, 130,	119, 209	38f64de44b3194e2a42b e09a4c318be7049ac5d6	probe	Delete

About This

THE world recently witnessed a massive surveillance program aimed at breaking users? privacy. Perpetrators were not hindered.

by the various security measures deployed within the targeted services . For instance, although these services relied on encryption mechanisms to guarantee data confidentiality

the necessary keying material was acquired by means of back doors, bribe, or coercion..

Know More..!

The encryption key is exposed and only the viable means to guarantee confidentiality is to limit the adversary access to the ciphertext.

Fig:10.10 Deleting Attack File

11. CONCLUSION

In modern times, cloud federation has gained usage among Cloud Service Providers (CSP) as a means of sharing computational resources for better service delivery.However, with the increase in malicious activities and cyber-attacks, the need for secure transactions and communication between federated entities becomes critical. This study presents a BiLSTM-based IDS model for a federated cloud platform named SecFedIDM-V1. Firstly, we developed a cloud federation testbed using open stack. Secondly, we experimentally evolved an IDS model using CIDDS dataset with BiLSTM RNN giving the best performance metrics for intrusion traffic detection on an OpenStack-based federated cloud testbed. Thirdly, in order to enhance the security of the proposed architecture, we integrated a blockchain to serve as a secure datastore for intrusion signatures. In order to make the model usable for cloud system administrators, we developed a cloud native web application that incorporates the proposed BiLSTM IDS model. As a proof of concept, we illustrated how different classes of malicious traffic could be detected with high precision from the cloud application. The web app could be deployed on other experimental or production-federated cloud platforms to detect malicious intrusions.

12. FUTURE SCOPE

In the future, we hope to extend the architecture as well as the web application by covering novel and higher number of network attack classes. Investigate methods to optimize the scalability and performance of the IDS model and the overall architecture to handle large-scale federated Foster

collaboration with academic institutions, industry partners, and cybersecurity communities to leverage collective expertise and resources for advancing the state-of-the-art in federated cloud security. This could involve participating in research consortia, open-source initiatives, or collaborative projects aimed at addressing cybersecurity challenges in cloud computing.loud environments efficiently.

13. REFERENCES

[1] Shavi Spinzi, “The evolution of industry 4.0,through the eyes of the pcb manufacturer,” 2017,https://www.electronicdesign.com/technologies/test-measurement/article/21207506/the evolution-of-industry-40-through-the-eyes-of-the-pcb-manufacturer, Last accessed on 2022-12- 23.

[2] D. B. Anitha and M. Rao, “A survey on defect detection in bare pcband assembled pcb using image processing techniques,” in 2017 Interna-tional Conference on Wireless Communications, Signal Processing andNetworking (WiSPNET), 2017, pp. 39–43.

[3] M. Moganti, F. Erçal, C. H. Dagli, and S. Tsunekawa, “Automatic pcbinspection algorithms: A survey,” Comput. Vis. Image Underst., vol. 63,pp. 287–313, 1996.

[4] T. Evgeniou and M. Pontil, “Support vector machines: Theory andapplications,” in Machine Learning and Its Applications, vol. 2049, 092001, pp. 249–257.

[5] E. Grossi and M. Buscema, “Introduction to artificial neural networks,”European journal of gastroenterology and hepatology, vol. 19, pp. 1046–54, 01 2008.[11] C. R. Reeves, “Genetic algorithms,” in Encyclopedia of Database Sys-tems, 1993.

[6] L. Rokach and O. Maimon, “Decision trees,” in The Data Mining andKnowledge Discovery Handbook, 2005.

[7] M. Seul, L. O’Gorman, and M. J. Sammon, “Template matching,” inPractical Algorithms for Image Analysis: Description, Examples.