DOCUMENTATION OF ICICTA_202

CREATING REAL FILES – 1000

```
msfadmin@metasploitable:~$ ls
generate_files.py  honey_files  real_files  vulnerable
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ cd ~/real_files
msfadmin@metasploitable:~/real_files$ mkdir -p logs configs reports
msfadmin@metasploitable:~/real_files$
msfadmin@metasploitable:~/real_files$ for i in {1..1000}
> do
>  case $((RANDOM % 3)) in
>   0) FOLDER="logs";;
>   1) FOLDER="configs";;
>   2) FOLDER="reports";;
>  esac
>
>  if ((i<=500)); then
>   echo "System log entry $i-normal operation"> "$FOLDER/syslog_$i.txt"
>  else
>   CONF_INDEX=$((i-500))
>   echo "username=user$CONF_INDEX"> "$FOLDER/user_config_$CONF_INDEX.conf"
>  fi
> done
msfadmin@metasploitable:~/real_files$ echo "1000 real files created: 500 .txt +
500 .conf across logs, configs, reports."
1000 real files created: 500 .txt + 500 .conf across logs, configs, reports.
msfadmin@metasploitable:~/real_files$
```

CREATING HONEY FILES – 1000

```
msfadmin@metasploitable:~/honey_files$ mkdir -p finance legal credentials
msfadmin@metasploitable:~/honey_files$ BAIT_NAMES=("password_list" "ssn_backup"
"bank_login" "credentials" "employee_salary" "finance_data" "secret_keys" "accou
nts2025" "tax_returns" "project_funding" "login_info")
msfadmin@metasploitable:~/honey_files$ EXTENSIONS=("docx" "xlsx" "txt")
msfadmin@metasploitable:~/honey_files$ for i in {1..1000}
> do
>  NAME=${BAIT_NAMES[$RANDOM % ${#BAIT_NAMES[@]}]}
>  EXT=${EXTENSIONS[$RANDOM % ${#EXTENSIONS[@]}]}
>  case $((RANDOM % 3)) in
>   0) FOLDER="finance";;
>   1) FOLDER="legal";;
>   2) FOLDER="credentials";;
>  esac
>  FILENAME="${NAME}_bait_$i.${EXT}"
>  FILEPATH="$FOLDER/$FILENAME"
>  echo "This is a honeyfile: $FILENAME"> "$FILEPATH"
>  echo "Access to this file will be monitored and logged.">>"$FILEPATH"
>  echo "DO NOT SHARE - Contains sensitive data such as passwords, finance info,
 or internal records.">>"$FILEPATH"
> done
msfadmin@metasploitable:~/honey_files$ echo "1000 honeyfiles deployed. The bait
is LIVE."
1000 honeyfiles deployed. The bait is LIVE.
msfadmin@metasploitable:~/honey_files$ _
```

COUNT

find ~/real_files ~/honeypot_files -type f | wc -l
2000

```
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ find ~/real_files -type f | wc -l
1000
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ find ~/honey_files -type f | wc -l
1000
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ find ~/real_files ~/honey_files -type f | wc -l
2000
msfadmin@metasploitable:~$
```

ifconfig
IP address of Metasploitable2 = 192.168.222.136
IP address of Kali = 192.168.222.128
Ping from victim: ping -c 4 192.168.122.128
Ping from attacker: ping -c 4 192.168.122.136

NMAP SCAN

<u>To check open ports:</u> nmap -sS -sV 192.168.222.136

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS -sV 192.168.222.136
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-03 14:34 EDT
Nmap scan report for 192.168.222.136
Host is up (0.0029s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:4C:23:80 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.03 seconds
```

PENTESTING

ssh -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedKeyTypes=+ssh-rsa msfadmin@<192.168.222.136>

Password: msfadmin

IN !!

```
┌──(root㉿kali)-[/home/kali]
└─# ssh -oHostKeyAlgorithms=+ssh-rsa -oPubKeyAcceptedKeyTypes=+ssh-rsa msfadmin@192.168.222.136
msfadmin@192.168.222.136's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Tue Jun  3 14:20:47 2025 from 192.168.222.128
msfadmin@metasploitable:~$
```

SCP (SSH File Transfer)

Testing SSH access

ssh msfadmin@<metasploitable-ip>

exit

Download the Folders to Kali

Scp command with legacy algorithms flags included

Downloaded honey files

**scp -oKexAlgorithms=+diffie-hellman-group1-sha1 **
**-oHostKeyAlgorithms=+ssh-rsa **
**-oPubkeyAcceptedKeyTypes=+ssh-rsa **
-r msfadmin@192.168.222.136:/home/msfadmin/honeypot_files ~/Downloads/

```
┌──(root㉿kali)-[/home/kali]
└─# scp -oKexAlgorithms=+diffie-hellman-group1-sha1 \
>  -oHostKeyAlgorithms=+ssh-rsa \
>  -oPubkeyAcceptedKeyTypes=+ssh-rsa \
>  -r msfadmin@192.168.222.136:/home/msfadmin/honey_files ~/Downloads
msfadmin@192.168.222.136's password:
password_list_bait_369.txt                    100%  191   143.0KB/s   00:00
password_list_bait_870.docx                   100%  192   135.1KB/s   00:00
project_funding_bait_777.xlsx                 100%  194   123.5KB/s   00:00
employee_salary_bait_435.txt                  100%  193   160.2KB/s   00:00
employee_salary_bait_356.docx                 100%  194   147.0KB/s   00:00
project_funding_bait_456.txt                  100%  193   138.4KB/s   00:00
password_list_bait_806.xlsx                   100%  192   147.2KB/s   00:00
```

Downloaded real files

**scp -oKexAlgorithms=+diffie-hellman-group1-sha1 **
**-oHostKeyAlgorithms=+ssh-rsa **
**-oPubkeyAcceptedKeyTypes=+ssh-rsa **
-r msfadmin@192.168.222.136:/home/msfadmin/real_files ~/Downloads/

```
┌──(root㉿kali)-[/home/kali]
└─# scp -oKexAlgorithms=+diffie-hellman-group1-sha1 \
 -oHostKeyAlgorithms=+ssh-rsa \
 -oPubkeyAcceptedKeyTypes=+ssh-rsa \
 -r msfadmin@192.168.222.136:/home/msfadmin/real_files ~/Downloads
msfadmin@192.168.222.136's password:
user_config_65.conf                                   100%   16     2.7KB/s   00:00
user_config_253.conf                                  100%   17    14.2KB/s   00:00
user_config_355.conf                                  100%   17    10.7KB/s   00:00
syslog_249.txt                                        100%   38    26.0KB/s   00:00
syslog_233.txt                                        100%   38    24.9KB/s   00:00
user_config_317.conf                                  100%   17    11.0KB/s   00:00
```

Downloaded successfully in Kali Linux

```
┌──(root㉿kali)-[~/Downloads]
└─# ls
honey_files  real_files

┌──(root㉿kali)-[~/Downloads]
└─# ls -l ~/Downloads/honey_files
total 60
drwxr-xr-x 2 kali kali 20480 Jun  3 23:56 credentials
drwxr-xr-x 2 kali kali 20480 Jun  3 23:56 finance
drwxr-xr-x 2 kali kali 20480 Jun  3 23:56 legal

┌──(root㉿kali)-[~/Downloads]
└─# ls -l ~/Downloads/real_files
total 52
drwxr-xr-x 2 kali kali 16384 Jun  3 23:58 configs
drwxr-xr-x 2 kali kali 16384 Jun  3 23:58 logs
drwxr-xr-x 2 kali kali 20480 Jun  3 23:58 reports

┌──(root㉿kali)-[~/Downloads]
└─#
```

LOGGING

nano log_and_zip.sh

```
TIMESTAMP=$(date +%Y%m%d_%H%M%S)
LOGFILE="download_log_$TIMESTAMP.txt"
ZIPFILE="honey_evidence_$TIMESTAMP.zip"

echo "[+] Logging download session ...">$LOGFILE
echo "Date: $(date)">>$LOGFILE
echo "Files Downloaded:">>$LOGFILE

find ~/Downloads/honey_files ~/Downloads/real_files -type f >>$LOGFILE

echo "[+] creating ZIP archive ... "
zip -r $ZIPFILE ~/Downloads/honey_files ~/Downloads/real_files $LOGFILE

echo "Evidence saved in $ZIPFILE successfully."
```

RUN:
chmod +x log_and_zip.sh
./log_and_zip.sh

Logged all files and zipped them into honeypot_evidence_20250604_145812.zip

To check logs inside the zip:

unzip -l honeypot_evidence_*.zip

```
┌──(root㉿kali)-[~kali]
└─# unzip -l honey_evidence_*.zip
Archive:  honey_evidence_20250604_002113.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
        0  2025-06-04 00:09   root/Downloads/honey_files/
        0  2025-06-03 23:56   root/Downloads/honey_files/legal/
      188  2025-06-03 23:56   root/Downloads/honey_files/legal/ssn_backup_bait_42.xlsx
      191  2025-06-03 23:56   root/Downloads/honey_files/legal/accounts2025_bait_581.xlsx
      191  2025-06-03 23:56   root/Downloads/honey_files/legal/finance_data_bait_151.xlsx
      188  2025-06-03 23:56   root/Downloads/honey_files/legal/login_info_bait_866.txt
      190  2025-06-03 23:56   root/Downloads/honey_files/legal/accounts2025_bait_980.txt
      190  2025-06-03 23:56   root/Downloads/honey_files/legal/tax_returns_bait_300.xlsx
      188  2025-06-03 23:56   root/Downloads/honey_files/legal/ssn_backup_bait_197.txt
      191  2025-06-03 23:56   root/Downloads/honey_files/legal/password_list_bait_724.txt
      189  2025-06-03 23:56   root/Downloads/honey_files/legal/ssn_backup_bait_513.xlsx
      188  2025-06-03 23:56   root/Downloads/honey_files/legal/bank_login_bait_125.txt
      192  2025-06-03 23:56   root/Downloads/honey_files/legal/password_list_bait_508.xlsx
      188  2025-06-03 23:56   root/Downloads/honey_files/legal/bank_login_bait_168.txt
      194  2025-06-03 23:56   root/Downloads/honey_files/legal/project_funding_bait_317.xlsx
```

ML INTEGRATION

DATA PREPROCESSING - EXTRACTING FEATURES FROM DATASET

Labelling the data

```
┌──(root㉿kali)-[~]
└─# mkdir -p ~/ml_dataset

┌──(root㉿kali)-[~]
└─# cd ~/ml_dataset

┌──(root㉿kali)-[~/ml_dataset]
└─# mkdir -p data/real data/honey

┌──(root㉿kali)-[~/ml_dataset]
└─# cp -r ~/Downloads/honey_files/* data/honey/

┌──(root㉿kali)-[~/ml_dataset]
└─# cp -r ~/Downloads/real_files/* data/real/
```

Feature Extraction Script

```python
import os
import pandas as pd

def extract_features(folder_path, label):
 data=[]
 for root, dirs, files in os.walk(folder_path):
  for file in files:
   path=os.path.join(root,file)
   try:
    size=os.path.getsize(path)
    with open(path, 'r', errors='ignore') as f:
     content=f.read()
     lines=content.count('\n')
     words=len(content.split())
     chars=len(content)
   except Exception as e:
    size,lines,words,chars=0,0,0,0

   data.append({
    'file_path': path,
    'file_name': file,
    'extension': os.path.splitext(file)[1],
    'size_bytes': size,
    'line_count': lines,
    'word_count': words,
    'char_count': chars,
    'label': label
   })

 return pd.DataFrame(date)

real_df = extract_features("data/real",label=0) #0=real
honey_df = extract_features("data/honey",label=1) #1=honey

full_df=pd.concat([real_df, honey_df], ignore_index=True)
full_df['extension']=full_df['extension'].astype('category').cat.codes
full_df.to_csv("features.csv",index=False)

print("Features extracted. Saved in features.csv")
```

Run it:

python3 extract_features.py

Train Classifiers – Random Forest, SVM, and Autoencoder

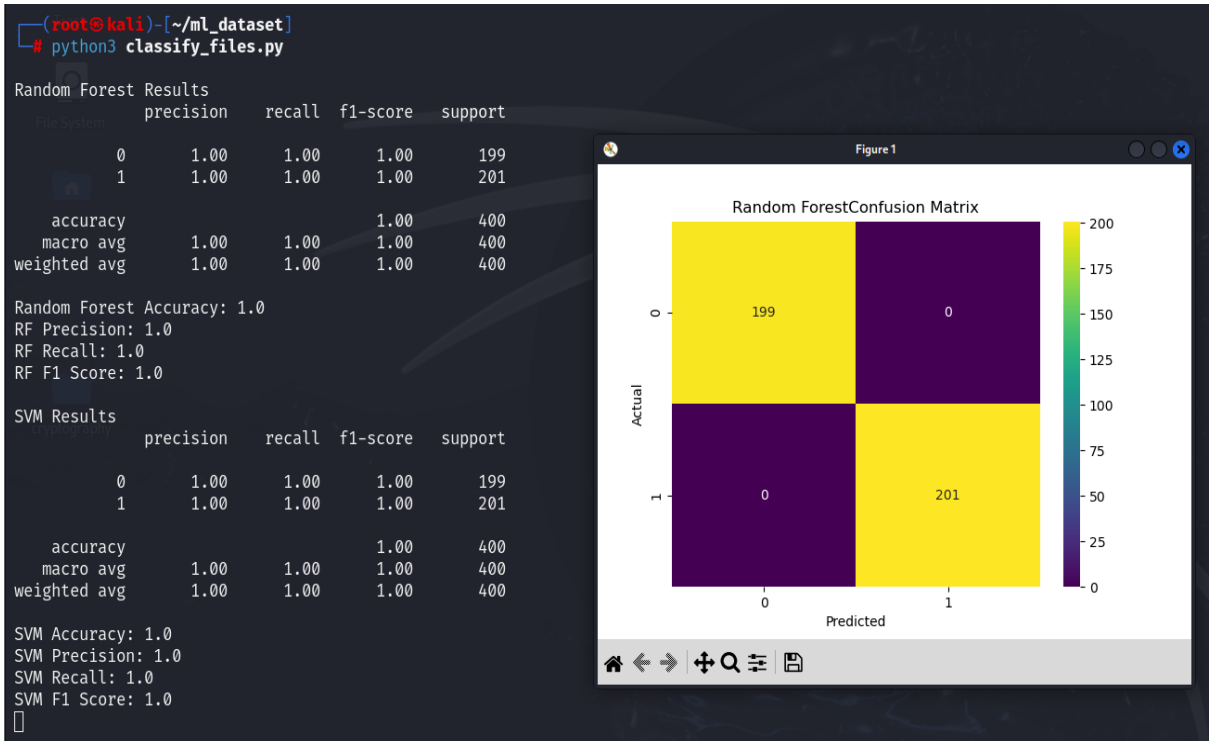
pip install --break-system-packages pandas scikit-learn matplotlib seaborn

nano classify_files.py

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score

#Load dataset
df = pd.read_csv("features.csv")
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['extension'] = le.fit_transform(df['extension'])
#Feature and Label
X=df[['extension','size_bytes','line_count','word_count','char_count']]
y=df['label']
#train_test_split
X_train, X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)

#RANDOM FOREST
rf=RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train,y_train)
rf_preds=rf.predict(X_test)
print("\nRandom Forest Results")
print(classification_report(y_test, rf_preds))
print("Random Forest Accuracy:",accuracy_score(y_test,rf_preds))
print("RF Precision:", precision_score(y_test,rf_preds))
print("RF Recall:", recall_score(y_test,rf_preds))
print("RF F1 Score:",f1_score(y_test,rf_preds))

#SVM
svm=SVC(kernel='linear')
svm.fit(X_train,y_train)
svm_preds=svm.predict(X_test)
print("\nSVM Results")
print(classification_report(y_test, svm_preds))
print("SVM Accuracy:",accuracy_score(y_test,svm_preds))
print("SVM Precision:",precision_score(y_test,svm_preds))
print("SVM Recall:",recall_score(y_test,svm_preds))
print("SVM F1 Score:", f1_score(y_test,svm_preds))

#Confusion Matrix
def plot_cm(model_name, y_true, y_pred):
 cm=confusion_matrix(y_true,y_pred)
 sns.heatmap(cm,annot=True,fmt='d',cmap='viridis')
 plt.title(f'{model_name}Confusion Matrix')
 plt.xlabel('Predicted')
 plt.ylabel('Actual')
 plt.show()

plot_cm("Random Forest",y_test,rf_preds)
plot_cm("SVM",y_test,svm_preds)
```

To run: python3 classify_files.py

Output:



- The script loads features.csv, which contains file metadata (extension, size, line count, word count, character count).
- The 'extension' column is encoded using LabelEncoder to convert text labels into numeric form.
- Features (X) and the target label (y, indicating real or honeyfile) are defined.
- The dataset is split into training and testing sets using train_test_split().
  Two classifiers are trained: Random Forest and Support Vector Machine (SVM).
- Each model predicts labels for the test set, and their performance is measured using:
  Accuracy
  Precision
  Recall
  F1 Score
  Confusion Matrix (via heatmap)

Heatmap:

Top-left: Real files correctly identified as real (True Negatives)
Bottom-right: Honeyfiles correctly identified (True Positives)
Top-right: Real misclassified as honeyfiles (False Positives)
Bottom-left: Honeyfiles misclassified as real (False Negatives)