Digital Forensics

Chandrima Chakraborty
Concordia University
Montreal, Quebec, Canada
c_chakra@encs.concordia.ca

Supervisor: Serguei A. Mokhov
Concordia University
Montreal, Quebec, Canada
Mokhov@cs.concordia.ca

Abstract

In this paper, we'll be pitching in few ideas of how we can use Blockchain Technology in the Field of Digital Forensics. Our main focus in this paper will be on how digital forensic investigators can take advantage of this technology to secure the integrity of digital evidences seized. This could be a huge step forward in the field of digital forensics since the credibility of digital evidences has always been challenged in the court every now and then. I have also discussed about docker framework which we can use in digital Forensics.

1 Introduction

Blockchain is more than cryptocurrencies. Blockchain is actually a distributed ledger that is completely open to anyone and this ledger stores information across a network of personal computers making them not only decentralized but also distributed which means no one person or a company owns the system. It has a very interesting property where if a data is once recorded in the blockchain it is almost impossible to change it and this is what we want to take advantage of. We can use Geth to join Ethereum network, transfer ether between accounts or even mine ethers. Smart contract is an agreement between two people in the form of computer code. They run on the blockchain, so they are stored on a public database and cannot be changed. The transactions that happen in a smart contract are processed by the blockchain, which means they can be sent automatically without a third party.

2 Process

After installing the geth we can check below details

```
root@chand-VirtualBox:/home/chand/Contracts# geth version
 Geth
Version: 1.9.25-stable
Git Commit: e7872729012a4871397307b12cc3fc4772ffcbec6
Architecture:amd64
Protocol Versions: [65 64 63]
Go Version: go1.15.6
Operating System: linux
GOPATH=
GOROOT=go
```

We can also check the balance using getBalance.

```
>eth.getBalance(\0x7BobA211d16c92cD28A372AF43a41a099aed0789")
  0
```

2.1 Solidity Installation

Solidity is one of the main compilers for Ethereum smart contracts. In Ubuntu you can install the latest solc compiler. it is used for designing and implementing smart contracts within the Ethereum Virtual Machine and other blockchain development platforms.

Installed using apt-get install solc.

```
root@chand-VirtualBox:/home/chand# apt-get install solc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcvc4-5
The following NEW packages will be installed:
  libcvc4-5 solc
```

Figure 1: Solidity installation

2.2 Docker installation and process

Install docker using below command.

```
#apt-get install docker.io -y Pull docker from ethereum/solc:0.6.6
There is a docker from where we pull the request.Received the below output after running command.

#docker pull ethereum/solc:0.6.6
   0.6.6: Pulling from ethereum/solc
   b72c78073b08: Pull complete
   Digest: sha256: *********

Status: Downloaded newer image for ethereum/solc:0.6.6
docker.io/ethereum/solc:0.6.6

[NOTE:Checked from docker hub]

  https://hub.docker.com/r/ethereum/solc/tags?page=1&ordering=last_updated


 #apt-get install docker.io -y
```

```
root@chand-VirtualBox:/home/chand# apt-get install docker.io -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd pigz runc ubuntu-fan
```

Figure 2: Docker installation

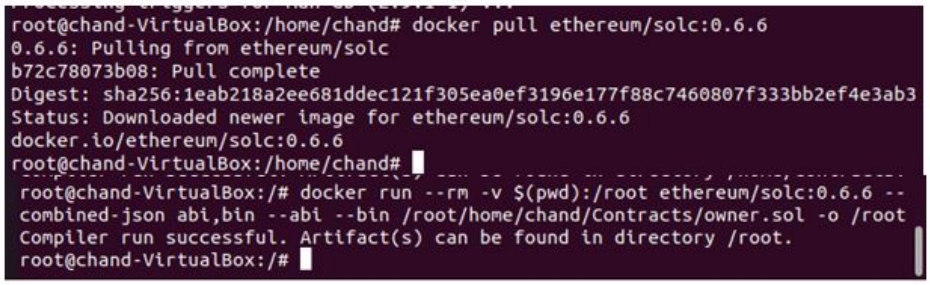

Figure 3: Pull Ethereum

2.3 Digital Forensics and Incident Response using Docker

Docker exists to store and distribute illegal information as a carrier for initiating attacks like traditional cloud services. Containers are typically employed within a microservices-based architecture. While great for speed and scalability, this means that suspicious activity will likely involve multiple containers pertaining to multiple services, across various hosts and with access to different data stores.

While investigating a container breach, make note of any data stores or services that this container accessed. Container acquisition is much faster than VM acquisition.

There are few tools or standard applications that all investigators and incident handlers should know.

- PEScanner -

PEScanner is a tool to perform static analysis of Microsoft portable executable files.

- JSDetox -

JSDetox is a javascript malware analysis tool. With the growing number of .js files spread by phishing and ransomware campaign, this is a must at the moment.

```
$  docker run - -rm {p 3000 : 3000 remnux/jsdetox
```

- SpiderMonkey -

It is also an javascript analyser developed by Mozilla, it helps to analyze malicious scripts.

```
$ docker run - -rm {it {v  /files:/source nacyot/javascript-spidermonkey: latest js <malicious.js>
```

- Malcom -

It is a tool which analysis network communications using graphical representation of netwok traffic.

```
$ docker run {p 8080:8080   -d --name  malcom  tomchop/malcom-automatic
```

- FIR (Fast Incident Response) -

In this a docker file is available on the FIR GitHub repository. It developed by Societe Generale CERT.

```
$ docker build {t  fir
$ docker run  -it   -p  8000:8000  fir
```

- ClamAV -

It quickly scans a suspicious system.

```
$ docker restart clamav
$ docker exec  -it  clamav  /malware/<suspicious_file>
```

- Volatility -

It is a memory forensic framework , it has a docker image in it which is created and maintained by SANS Remnux team.

2.4 Initialize files

The genesis block is created using the genesis state file or genesis.json in Geth. This file contains all the data that will be needed to generate block 0. The mainnet chainID is 1. "chainID" was introduced in EIP155.

The intention in adding it was to make transactions on the Ethereum network look different from those on the Ethereum classic network. Transactions are signed differently depending on the chainID used.

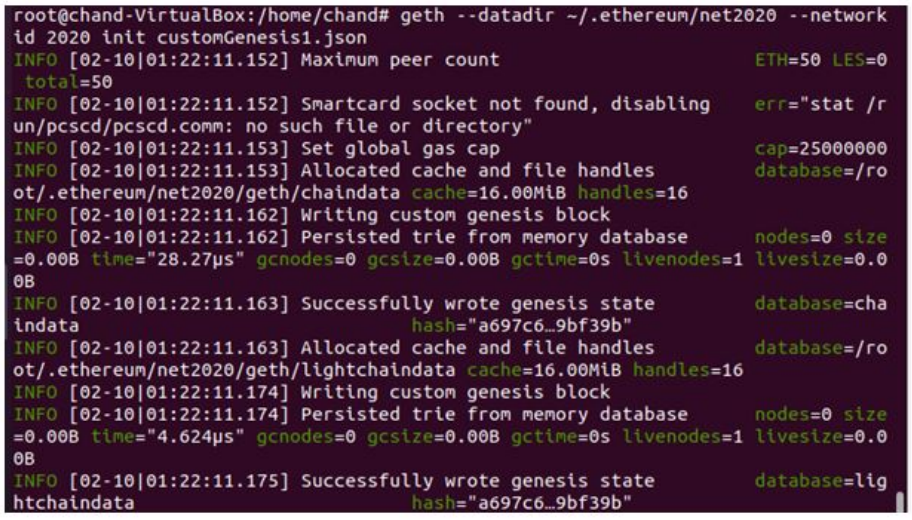

Figure 4: Initialization

2.5 Account creation

It is easier to create account outside geth, which we can access inside Geth console. Accounts can be user-controlled or deployed as smart contracts.

```
>./2021-create-account.sh  (Created two accounts).

(base) chand@chand-VirtualBox:~$ ./2021-create-account.sh
Password:
Repeat password:
**New key was generated**
```

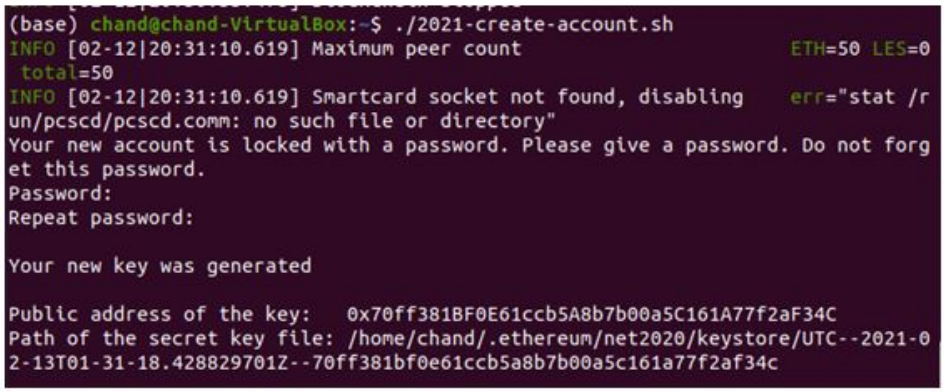

Figure 5: Create first account

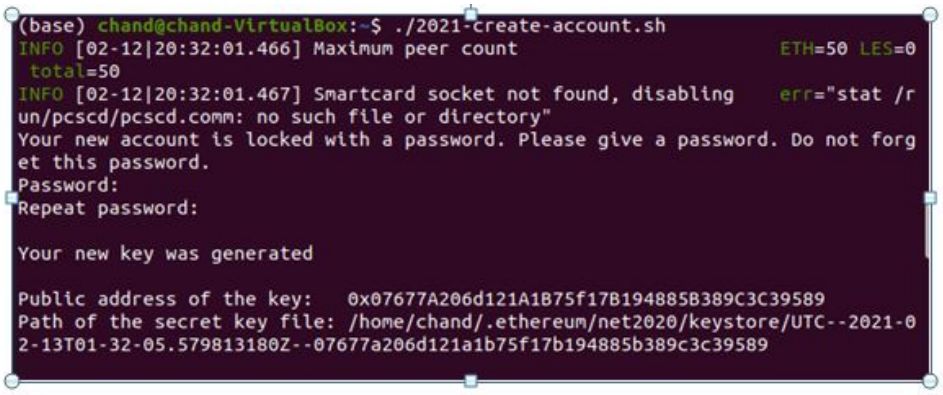

Figure 6: Create second account

2.6 Genesis state creation

In this section we will see the genesis state has successfully created. We have allocated accounts and balance.

```
{
 "alloc": {
    "0xca843569e3427144cead5e4d5999a3d0ccf92b8e": {
      "balance": "1000000000000000000000000000"
    },
    "0x0fbdc686b912d7722dc86510934589e0aaf3b55a": {
      "balance": "1000000000000000000000000000"
    }
```

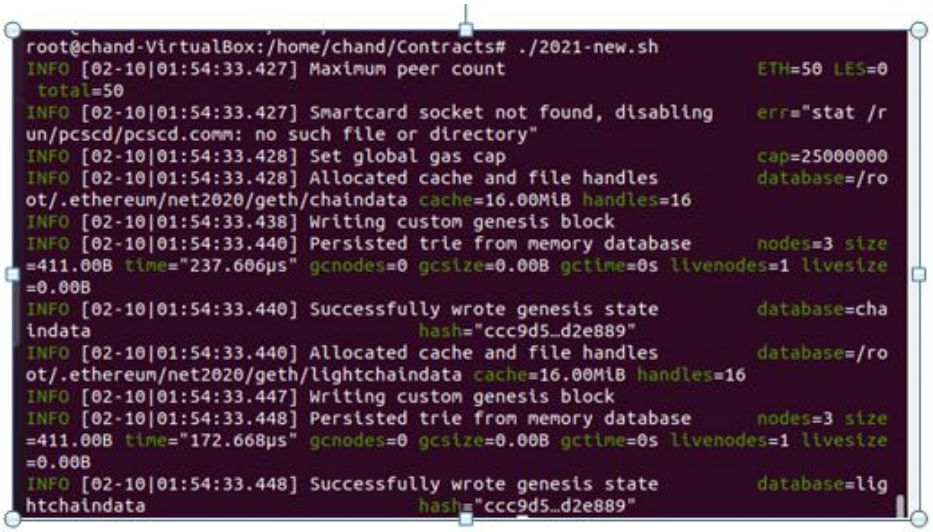

Figure 7: Create second account

```
root@chand-VirtualBox:/home/chand/Contracts# ./2021-new.sh
```

We can also verify if key has been stored inside keystore file. “ls -l” command lists all the file and directories in the current directory or the path supplied on the command line in a “long” listing format.

```
Keys stored inside ~/.ethereum/net2020/keystore
```

3 Implement

3.1 Geth console

In the “2021-start.sh” script file I have mentioned and allocated rpcport, domain and network ID.

–http://localhost:8545 is the RPC port of locally running Ethereum node software.

```
(base) chand@chand-VirtualBox:~$ ./2021-start.sh
```

Keys stored inside ~/.ethereum/net2020/keystore

```
(base) chand@chand-VirtualBox:~$ cd ~/.ethereum/net2020/keystore
(base) chand@chand-VirtualBox:~/.ethereum/net2020/keystore$ ls -lart
total 24
-rw------- 1 chand chand  491 Feb 12 20:29 UTC--2021-02-13T01-29-37.560314280Z-
-a008a226467a526291f67f29536f25015a582556
-rw------- 1 chand chand  491 Feb 12 20:29 UTC--2021-02-13T01-29-58.578713832Z-
-8943b7ab4a9fa30ac9828f93fbd0c136b4981021
drwx------ 4 chand chand 4096 Feb 12 20:30 ..
-rw------- 1 chand chand  491 Feb 12 20:31 UTC--2021-02-13T01-31-18.428829701Z-
-70ff381bf0e61ccb5a8b7b00a5c161a77f2af34c
-rw------- 1 chand chand  491 Feb 12 20:32 UTC--2021-02-13T01-32-05.579813180Z-
-07677a206d121a1b75f17b194885b389c3c39589
drwx------ 2 chand chand 4096 Feb 12 20:32 .
```

Figure 8: Create second account

```
(base) chand@chand-VirtualBox:~$ ./2021-start.sh
```

Figure 9: Geth login

```
>eth.accounts
```

```
//b4fb72800694dc40c20f41bd48fca18585a0d3ca9790f1bf0ea431b8f272c7a2f3482750a6ca6
8e6bce3331c742b06c30c42f6d62840a522e5b8006ec5e291a9@127.0.0.1:30303?discport=0"
INFO [02-10|01:59:23.603] IPC endpoint opened                      url=/root/.e
thereum/net2020/geth.ipc
INFO [02-10|01:59:23.603] HTTP server started                      endpoint=[::
]:8545 cors=* vhosts=localhost
WARN [02-10|01:59:23.645] Served eth_coinbase                      reqid=3 t="3
8.062µs" err="etherbase must be explicitly specified"
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.25-stable-e7872729/linux-amd64/go1.15.6
at block: 0 (Wed Dec 31 1969 19:00:00 GMT-0500 (EST))
 datadir: /root/.ethereum/net2020
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0
 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
>
```

Figure 10: Geth Console

```
Command used inside blockchain
================================
>eth.accounts
>eth.getBalance ()
>web3.fromWei(1e+23 ,"ether").
```

3.2 Create personal account

We can create personal account within geth using personal.newAccount command.

```
> personal.newAccount()
 Passphrase:
**Remember password
0a8cabb56cdff429567cf15a28556cf4
```

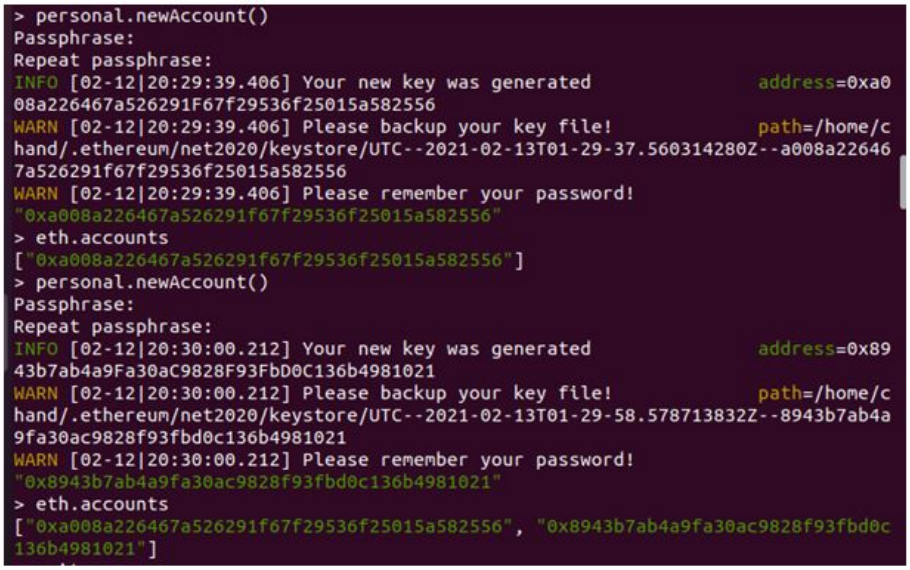

Figure 11: Geth Console

We can check the account details inside geth using eth.accounts command. Here we can see 4 accounts. Two accounts created inside geth console and other two are personal account.

```
    >eth.accounts
[0a8cabb56cdff429567cf15a28556cf4,0a8cabb56cdff429567cf15a28556cf6,0a8cabb56cdff429567cf15a28556cf3,0a8cabb56
```

Account details

```
>eth.accounts
>eth.blockNumber
>eth.coinbase
>eth.getBalance(account0)
>eth.getBalance(account1)
>web3.personal.listAccounts
```

>eth.accounts

```
> eth.accounts
["0xa008a226467a526291f67f29536f25015a582556", "0x8943b7ab4a9fa30ac9828f93fbd0c
136b4981021", "0x70ff381bf0e61ccb5a8b7b00a5c161a77f2af34c", "0x07677a206d121a1b
75f17b194885b389c3c39589"]
```

Figure 12: Accounts

```
> eth.accounts
["0xa008a226467a526291f67f29536f25015a582556", "0x8943b7ab4a9fa30ac9828f93fbd0c
136b4981021", "0x70ff381bf0e61ccb5a8b7b00a5c161a77f2af34c", "0x07677a206d121a1b
75f17b194885b389c3c39589"]
> eth.blockNumber
0
> eth.coinbase
"0xa008a226467a526291f67f29536f25015a582556"
> var account0 = eth.accounts[0];
undefined
> var account1 = eth.accounts[1];
undefined
> eth.getBalance(account0);
0
> web3.fromWei(1e+23, "ether")
"100000"
> eth.getBalance(account0);
0
> eth.getBalance(account1);
0
> web3.personal.listAccounts
["0xa008a226467a526291f67f29536f25015a582556", "0x8943b7ab4a9fa30ac9828f93fbd0c
136b4981021", "0x70ff381bf0e61ccb5a8b7b00a5c161a77f2af34c", "0x07677a206d121a1b
75f17b194885b389c3c39589"]
```

Figure 13: Account details

3.3 Transaction Generation Process

We can send ether from one account to another using sendTransaction function however we will get error “authentication needed “ because by default accounts inside Geth are blocked for security purposes.

Given permission by providing password.

```
> eth.sendTransaction({from:eth.accounts[0], to:eth.accounts[1], value: web3.to
Wei(1, "ether")})
WARN [02-17|00:07:58.590] Served eth_sendTransaction              reqid=17 t=5
.58825ms err="authentication needed: password or unlock"
Error: authentication needed: password or unlock
        at web3.js:6347:37(47)
        at web3.js:5081:62(37)
        at <eval>:1:20(21)
```

Figure 14: Error while transaction

Transaction- We can see the transaction generation process.

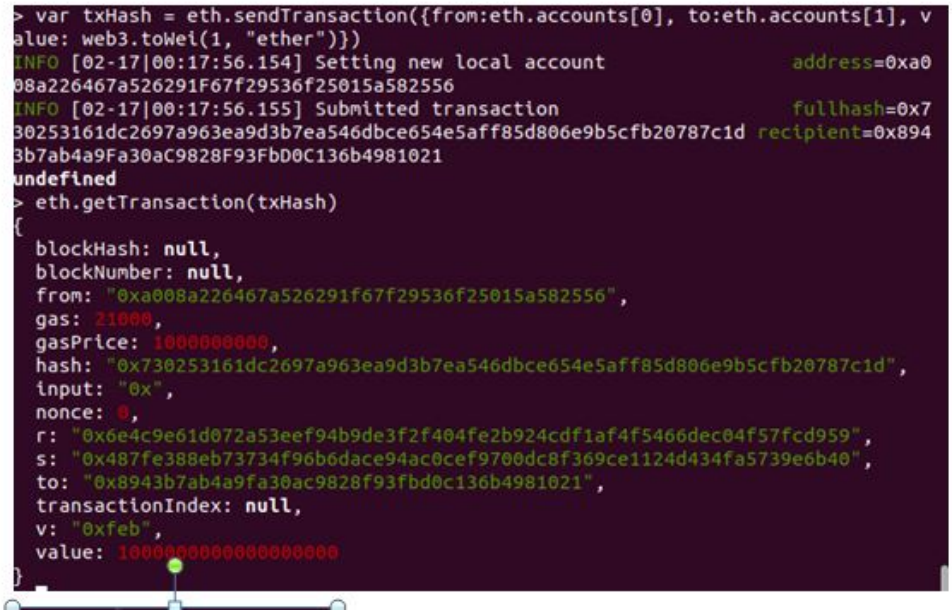

Figure 15: Transaction

Transaction successfully submitted and now we can see and verify receipt.

```
>eth.mining
   true (mining should start before generating transaction)

>eth.coinbase (Own address will be shown).

To create a new transaction object :
    var ownerContractInstance = ownerContract.new(deployTransationObject)
```

3.4 Generate receipt

We can view the transaction receipt using the getTransationReceipt command. etTransaction-Receipt(hash [, callback]). Returns the receipt of a transaction by transaction hash. The receipt is not available for pending transactions.

```
>eth.getTransactionReceipt();
```

After transaction we can check the amount using below command:

```
eth.getBalance (account number)


      We can run below command:
  >miner.start() : It will show the current block number.
```

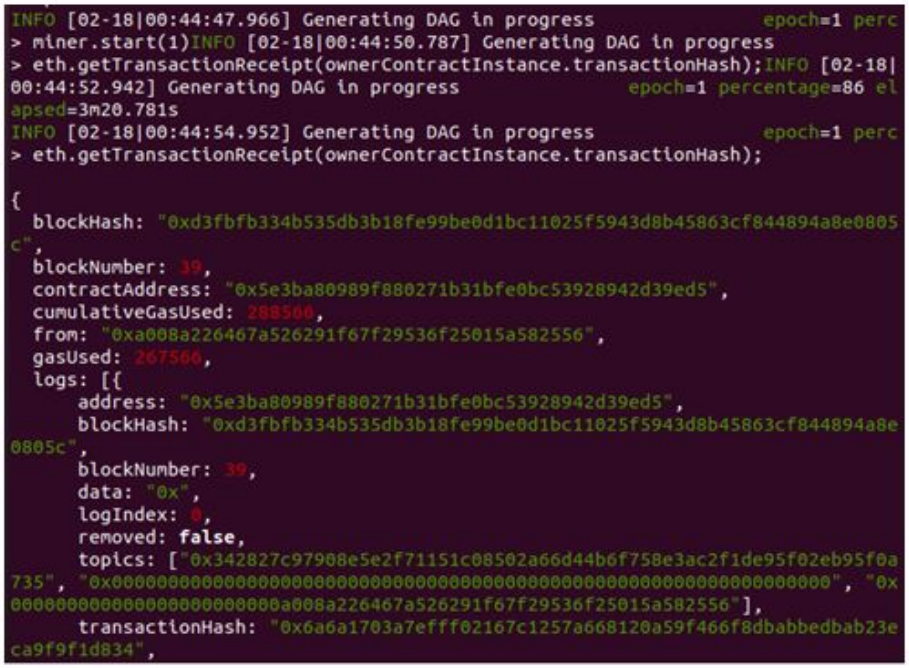

Figure 16: Receipt details

Figure 17: Coinbase after transaction

4 Forensic Investigation

In this section I have used an online tool named Forensically Beta to check if there is any error in an image (has taken a fingerprint) file. It is used for digital image forensics and also includes clone detection, error level analysis, meta data extraction. We can load images (JPEG,AVIF,PNG) to check if there is any kind of hidden characters or pixels. It displays the modifications in percentage in terms of the changes being made on the edges, textures, and surfaces of the original photo.

Image detection

5 Use of Wireshark in Digital Forensics

Wireshark is an Open source Forensic tool.It is used to inspect network traffic and even individual packets. It detects port scanning. It can capture username, password, email address, pictures, videos,personal information.

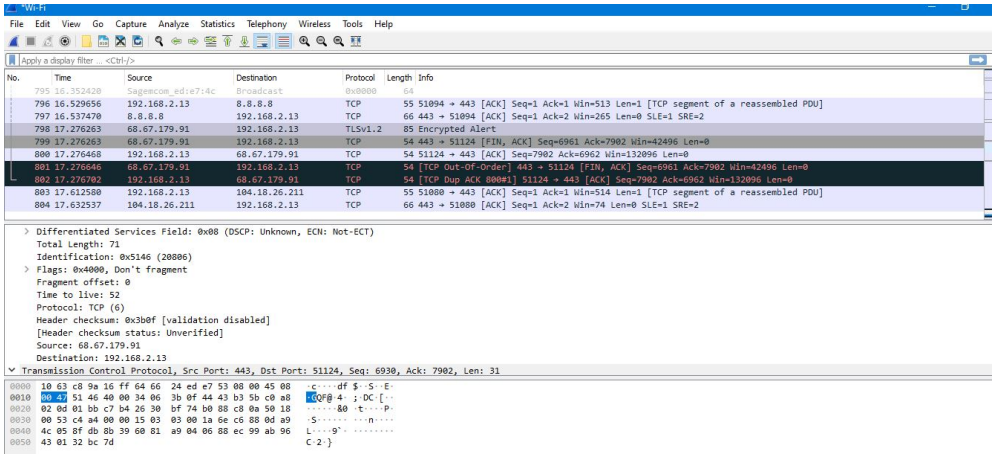

Figure 18: Wireshark detection

I have tried to login to a bank account, extract images from a digital forensic tool and captured those inside wireshark framework so that anyone can identify if any malicious activity is happening within forensic investigation. It is showing encryption failure message as I have entered wrong password.

6 Conclusion

We have discussed about digital forensics and few tools to inspect network traffic and packets.Docker allows applications to use the same Linux kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer.Using Docker we can develop software components, these components can then be combined to create a service offering (SaaS), web or mobile application (app), or distributed blockchain application (dApp).

7 Future Work

We can create our own docker to further strengthen the project and promote it for electronic publishing ,conferences and workshops.

References

[1] BRANDON ARVANAGHI. Explaining-the-genesis-block-in-ethereum. [online], April 2020. https://arvanaghi.com/blog/explaining-the-genesis-block-in-ethereum/.

[2] Ihor D. digital-forensics-and-incident-response. [online], April 2020. https://www.deepshankaryadav.net/digital-forensics-and-incident-response-dfir-using-docker/.

[3] Ihor D. Smart contract for a blockchain. [online], April 2020. https://rubygarage.org/blog/ethereum-smart-contract-tutorial.

[4] Taras Filtov. Setting up your private Ethereum blockchain. [online], April 2020. https://www.dappros.com/202004/setting-up-your-private-ethereum-blockchain/.

[5] Brian Hogan. set-up-docker. [online], April 2020. https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04.

[6] Michael Whittle. Ethereum smart contract using geth and web3.js. [online], April 2020. https://medium.com/coinmonks/deploy-your-first-private-ethereum-smart-contract-using-geth-and-web3-js-2e495c1aadf4.

[7] Nathan Williams. set-up-ethereum-blockchain. [online], April 2020. https://arctouch.com/blog/how-to-set-up-ethereum-blockchain/.