A7010E Homework 3

Nico Ferrari (nicfer-0@student.ltu.se)

October 18, 2020

1. Please download the two PDF files! Please discuss and reflect on these two cases in light of thestudied concepts in the applied computer security course?

Observing the emails we can notice that the sender looks sospicious. In fact, in the first email the sender is not an email address connected to IKEA. Since they want you to insert personal details for discounts, this could lead to attacks like **Phishing**.

In the second email, the domain of the email is reported in some websites and, after some researches, it is possible to notices that there are not markets online with that name. The attachment in the email is an Excel file. Excel files could execute macros and install Trojan viruses. This could lead to an attack called **Malspam**. An example comes from an hacking group known as Evil Corp or TA505 has presented a new risk by targeting businesses through the use of malicious Microsoft Excel documents. the victim was opening an excel file known as Grace Wire or Flawed Grace (which could be shared by email) leading to the malware attempting to put a remote access trojan (RAT) on their system.

In order to react to these spam emails, the first thing would be to inform the colleagues in the company about the threat. In fact, information and education is vital for preventing malware resulting from phishing. Enabling the Microsoft Office protections and enabling an antivirus will all help to block the threat and prevent the download of malicious files. Moreover, all the attachments must be checked from malwares before being opened. Policies must be applied in order to not share and ask for credentials through email. some other policies could incourage to accept emails just from addresses present in some sort of white list database, where only the trusted emails are registred.

Some anti-spam software has been implemented during the years. Some techniques helps to reduce the phenomena where an attacker sends malicious emails to several victims by appling some sort of proof-of-work [1]. Other techniques enable a database with a black list, but these do not help to protect from zero day phishing attacks [2]. New techniques try to use dynamic evolving Neural Networks in order to help to find zero-day phishing attacks [3].

2. What is your own reflection on the entire week of the course?

3. References

[1] Adam Back. *Hashcash-A Denial of Service Counter-Measure*. Tech. rep. 2002.

[2] Steve Sheng et al. "An empirical analysis of phishing blacklists". In: *6th Conference on Email and Anti-Spam, CEAS 2009* (2009).

[3] Sami Smadi, Nauman Aslam, and Li Zhang. "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning". In: *Decision Support Systems* 107 (2018), pp. 88–102. ISSN: 01679236. DOI: 10.1016/j.dss.2018.01.001. URL: https://doi.org/10.1016/j.dss.2018.01.001.

Blockchain Technology

Nico Ferrari

I. Blockchain Technology

Blockchain is a dristributed database of records, or public register (called Distributed Ledger Technology) of all transactions and events shared and executed among the participating parties, based on P2P protocol. Each entry on this ledger is verified by the majority of participants and cannot be deleted once inserted [1]. Blockchain is a technology which combines cryptography with distributed computing, created by the creator of Bitcoin which uses the alias Satoshi Nakamoto. He combined them generating a model where a network of computers collaborate in order to maintain a secure and shared database, consisting in a string of blocks.

Each block contains a set of transaction and a unique identifier of the data generated using the hash of the data. The block contains also a timestamp, a nonce used to verify the hash and the hash of the previous block, ensuring the integrity of the entire blockchain through to the first block, called genesis block, as shown in Figure 1.

The computers on the network will have the role to validate the transactions, add them to the block that they are building and then broadcast the block to the network in order to have the same copy among all the computers in the network.

Since there is no central component which validates the database alterations, blockchain depends on a consensus algorithm where all the computers agree on the state of the database. This consensus mechanism is the process in which a majority (or in some cases all) of network validators come to agreement on the state of a ledger maintaining coherent set of facts between the participating nodes.

This consensus algorithm ensures that the transaction are stored in a block before being inserted in the public ledger [2]. When the block is verified, it is added to the blockchain which will result in a chronologically ordered, linear and un-mutable chain of blocks. In fact, each hash value of the block depends on the hash of the previous one, and in order to modify a block, all the other blocks must be altered.

In some blockchain's applications such as Bitcoin, the validatiion of a block is performed by the miners. Miners have the role of validate the blocks and keep their order. In order to do so, a proof-of-work concept has been implemented. The proof-of-work is based on [3] and involves scanning for a value that, when hashed, the hash begins with a prefixed number of zero bits. The work required is exponential the number of zero bits required and can be verified by executing a single hash. After effort has been spent in order to satisfy the proof-of-work, the block cannot be changed without redoing the work [4].

Each transaction is protected through a digital signature, signing it digitally with the private key of the sender and sent to the "public key" of the receiver. The owner of the cryptocurrency needs to prove his ownership of the "private key" in order to spend money. The entity receiving the digital currency then verifies the digital signature by using the public key of the sender on the respective transaction.

In this way, each entity can be referred with his public key, without sharing personal data which may be easily linked to a real-world identity.

In order to keep the blockchain alive, an incetive is given to the miners. This incentive can be obtained by a special transaction in the mined block or by transaction fees.

II. Security Isues and Challenges

A. The Majority Attack (51% Attacks)

With Proof of Work, the probability of mining a block depends on the work done by the miner. Due to reward obtained by mining blocks, more entities take part in *mining pools*, where they share the power in order to try to mining more blocks. When one entity or a mining pool holds the 51% of the computing power, it can control the blockchain.

Having 51% of the computing power, means that the entity can find Nonce value quicker than others and having authority to decide which block is permissible, leading to the following risks:

- Modify the transaction data, it may cause double-spending attack
- To stop the block verifying transaction
- To stop miner mining any available block.

Some solutions to this type of attack have been studied in [5] [6].

B. Scalability of the Blockchain

With the growth of the blockchain, data will become bigger and it will require more and more computing power, as described in [7]. A mitigation to this problem is Simplified Payment Verification (SPV) , a payment verification technology which doesn't maintain the full blockchain information but onlyneeds block header message.

C. Energy Consumption

Due to the Proof-of-Work, blockchain is consuming enormous amounts of power today, with big mining farms running continuously to maintain the networks of the various blockchains. New alternative and innovative technology rises, which focuses on a sustainable solution, such as IOTA, a cryptocurrency based on an innovative technology called the Tangle [8] [9].

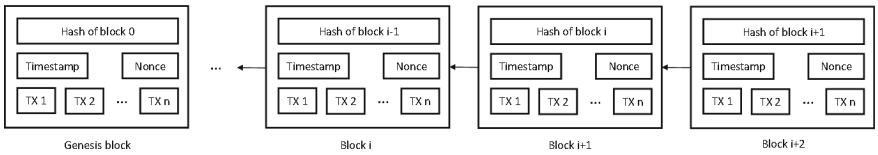


Fig. 1. blockchain example [2]

REFERENCES

[1] M. Crosby Nachiappan Pradan Pattanayak Sanjeev Verma and V. Kalyanaraman, "BlockChain Technology: Beyond Bitcoin," Tech. Rep., 2016.

[2] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business and Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 6 2017.

[3] A. Back, "Hashcash-A Denial of Service Counter-Measure," Tech. Rep., 2002.

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Tech. Rep. [Online]. Available: www.bitcoin.org

[5] I. Eyal and E. G. Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable *," Tech. Rep.

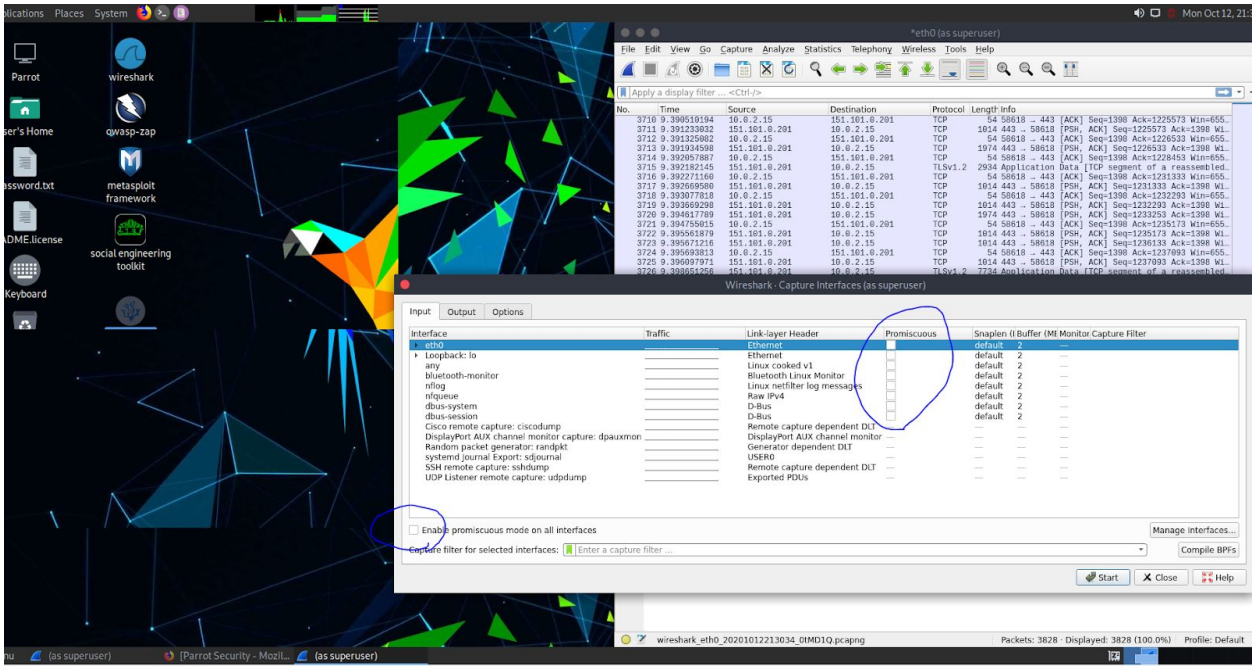
[6] X. Yang, Y. Chen, and X. Chen, "Effective scheme against 51% attack on proof-of-work blockchain with history weighted information," in *Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019*. Institute of Electrical and Electronics Engineers Inc., 7 2019, pp. 261–265.

[7] G. O. Karame, "On the security and scalability of Bitcoin's blockchain," in *Proceedings of the ACM Conference on Computer and Communications Security*, vol. 24-28-October-2016. Association for Computing Machinery, 10 2016, pp. 1861–1862.

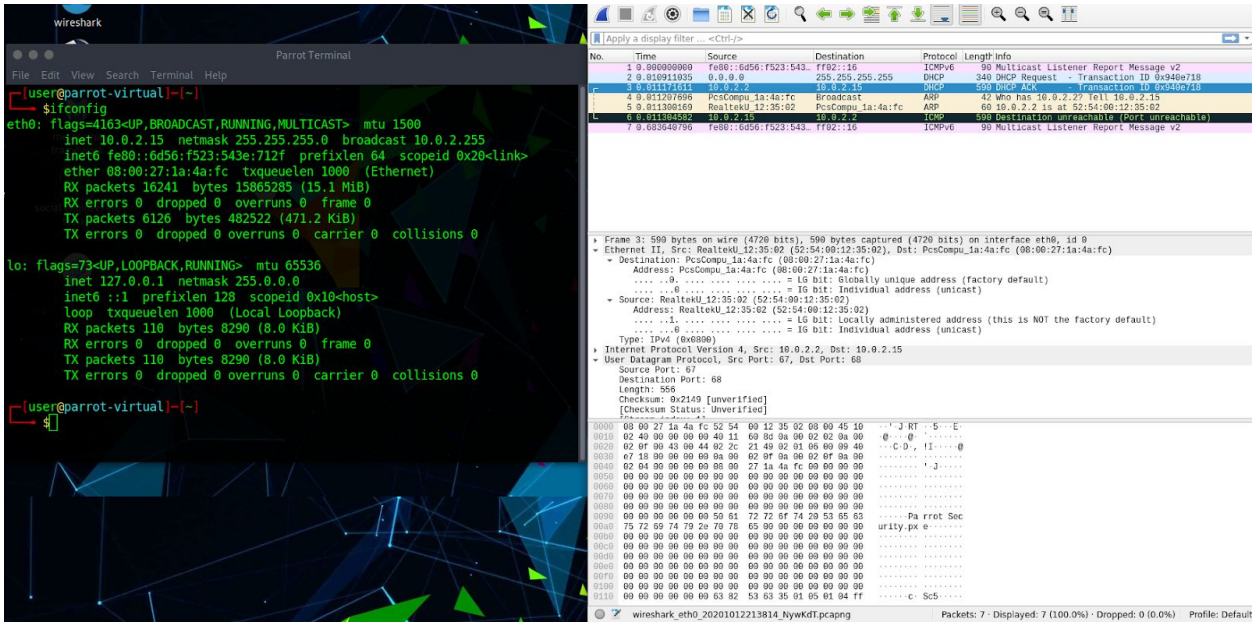
[8] S. Popov, "The Tangle," Tech. Rep., 2018.

[9] S. Popov, H. Moog, D. Camargo, A. Capossele, V. Dimitrov, A. Gal, A. Greve, B. Kusmierz, S. Mueller, A. Penzkofer, O. Saa, W. Sanders, L. Vigneri, W. Welz, and V. Attias, "The Coordicide," Tech. Rep., 2020.

After Setting up wireshark in order to not receive packets sent to other machines, we can start to capture the packets when the network on the VM is enabled.

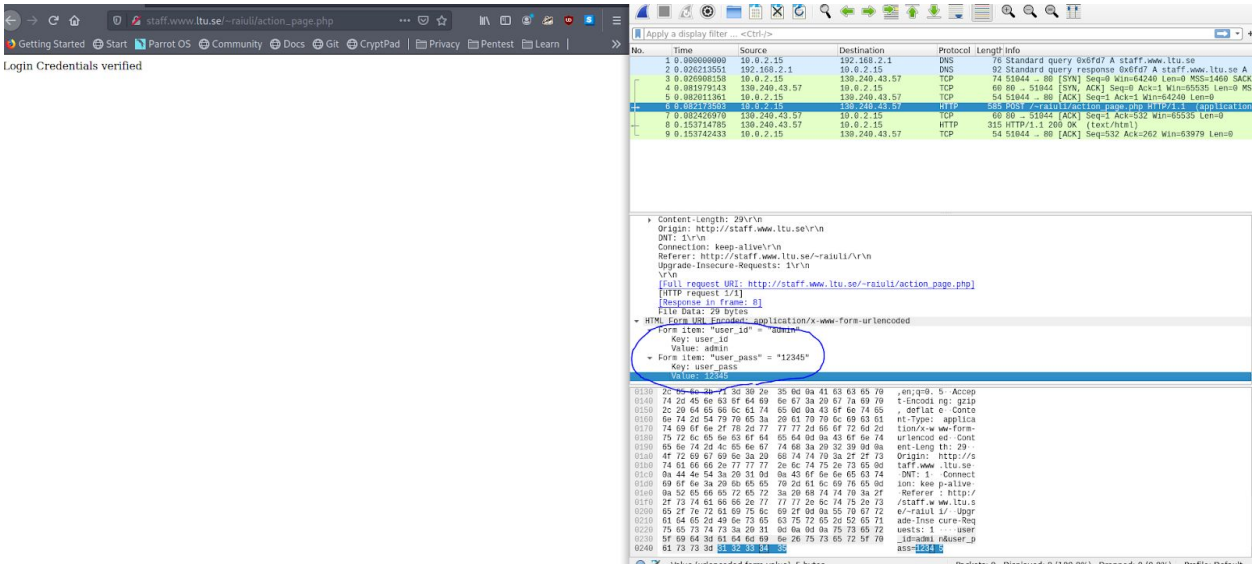


Since Virtual Box has an internal DHCP server enabled when NAT is activated, the VM will get its own ip address with the DHCP protocol as shown in the picture.

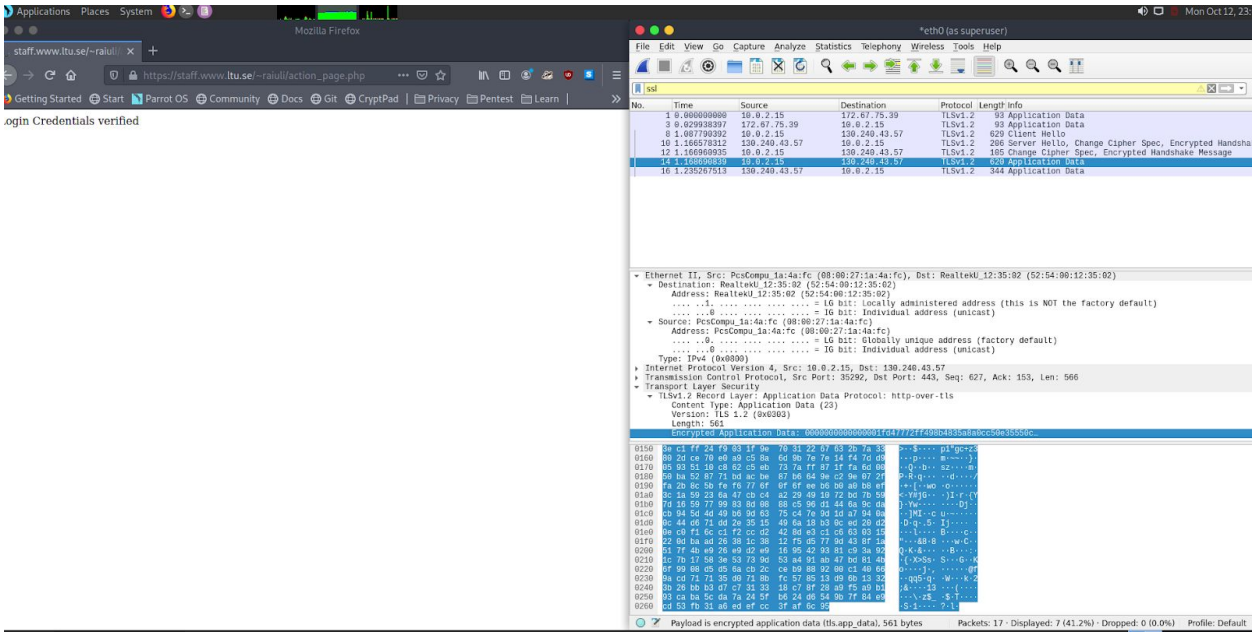


In order to analyze some unencrypted traffic, the http://staff.www.ltu.se/~raiuli/ website has been used. Since it uses the http protocol, the traffic will be unencrypted, so eventual credentials can be stolen. In this case we had a login form where the right credentials were:
username= admin
password=12345

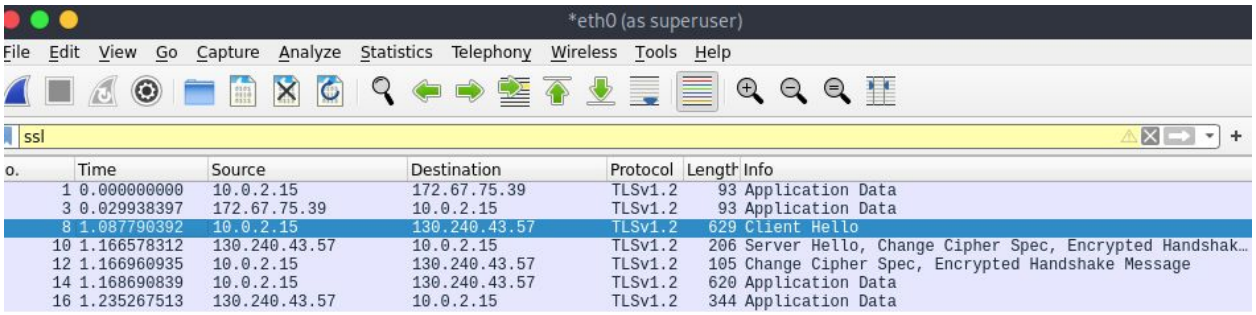
Analysing the traffic we can capture the packet and those credentials are shown in clear, together with all the other info regarding the packet.

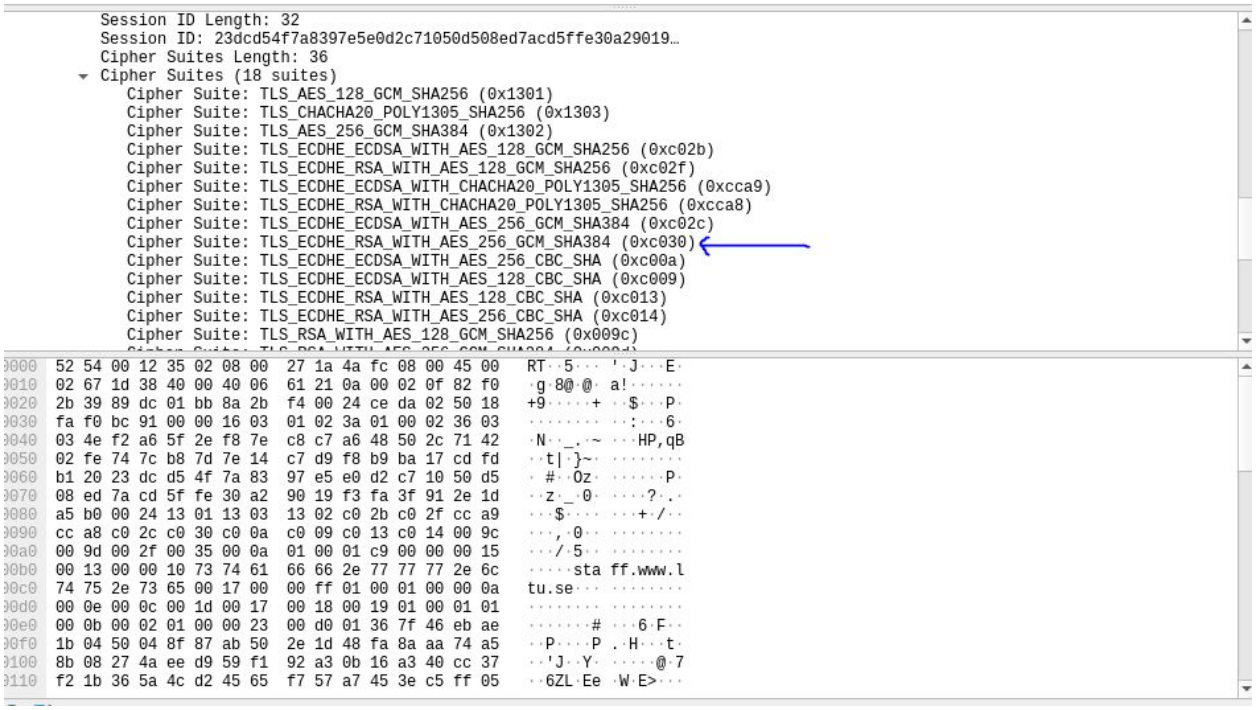


In order to encrypt the traffic, the https protocol has been used and as we see in the following screenshot the traffic is encrypted.

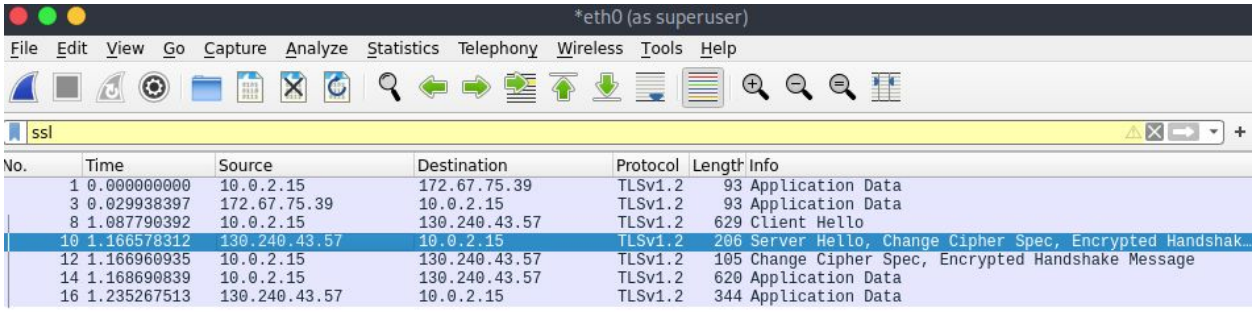


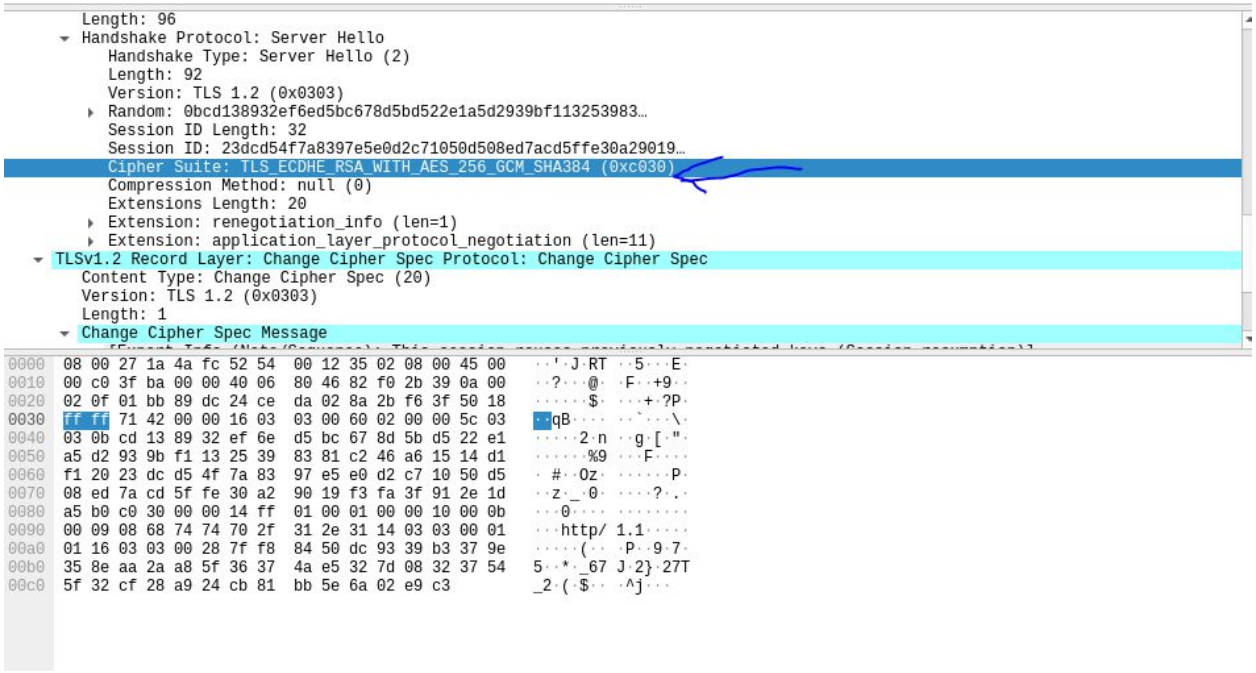
Analysing the transmission, we see that the VM sent a ‘Client Hello’ packet stating the available cipher suites (18 in total).



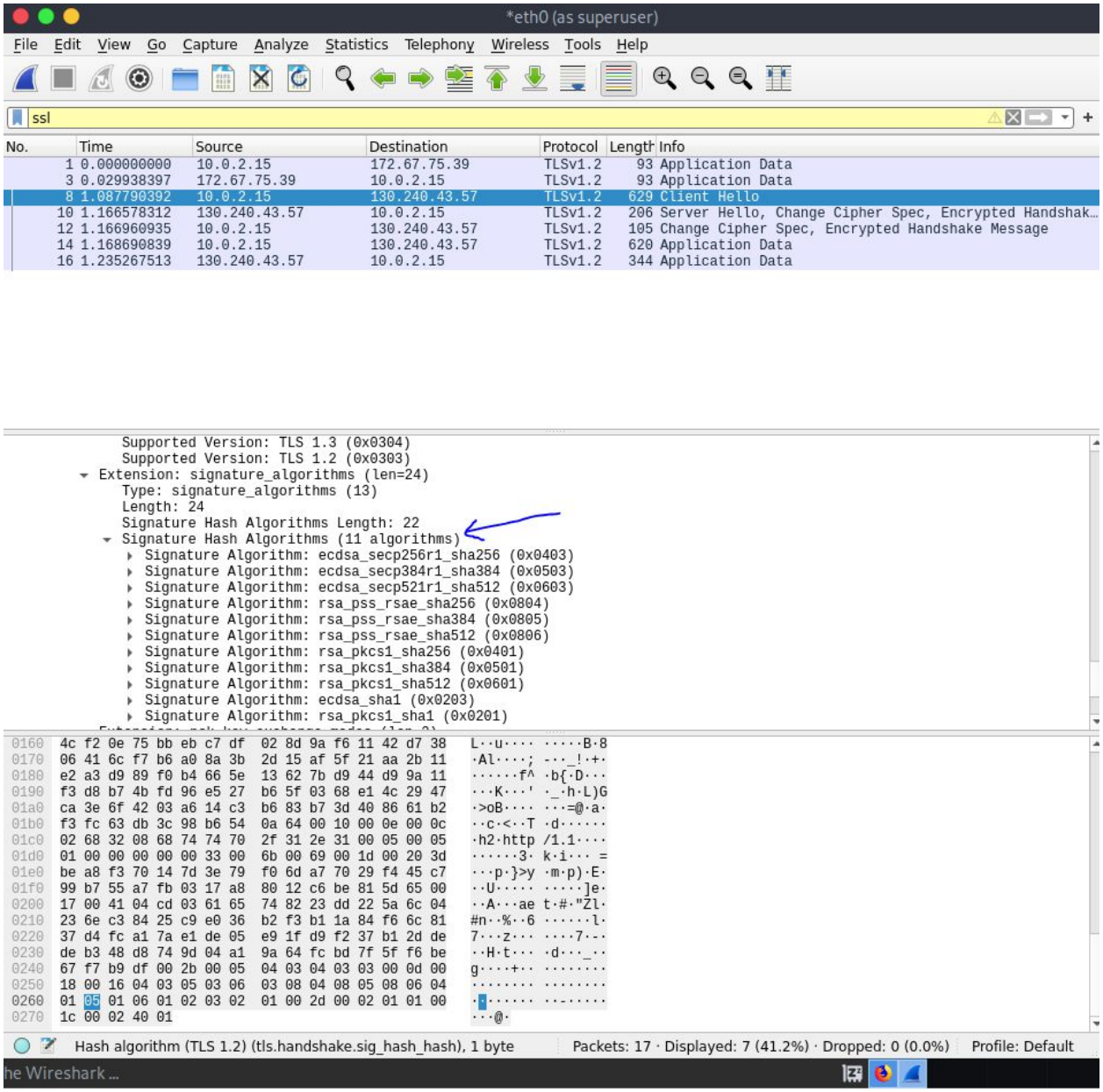


The server then replied to that packet choosing the cipher suite (in this case TCL_ECDHE_RSA_WITH_AES_256_GCM_SHA384)





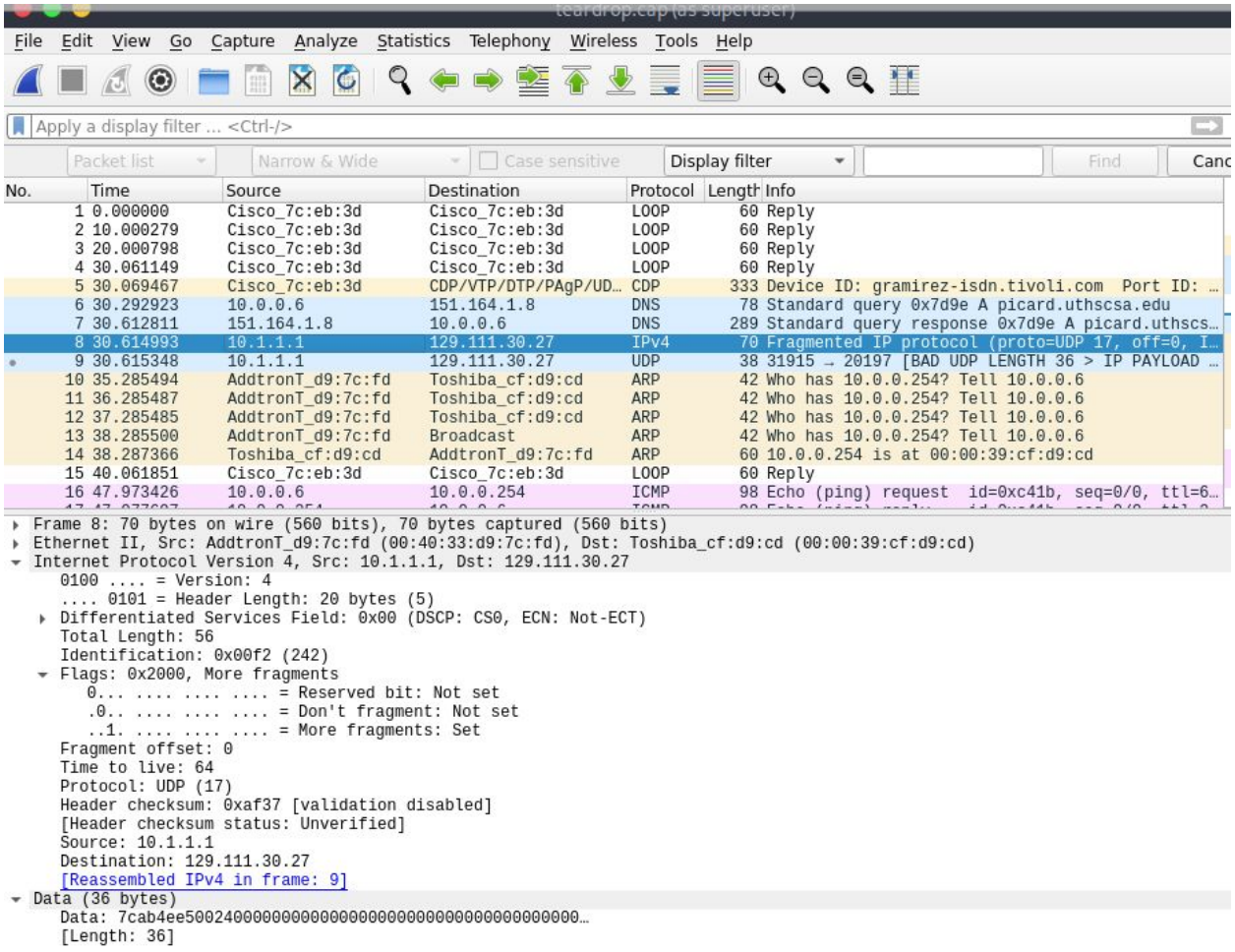
.From the ‘ Client Help’packet it is possible also to se the hash algorithms available. In this case 4: SHA246,SHA384, SHA512,SHA1

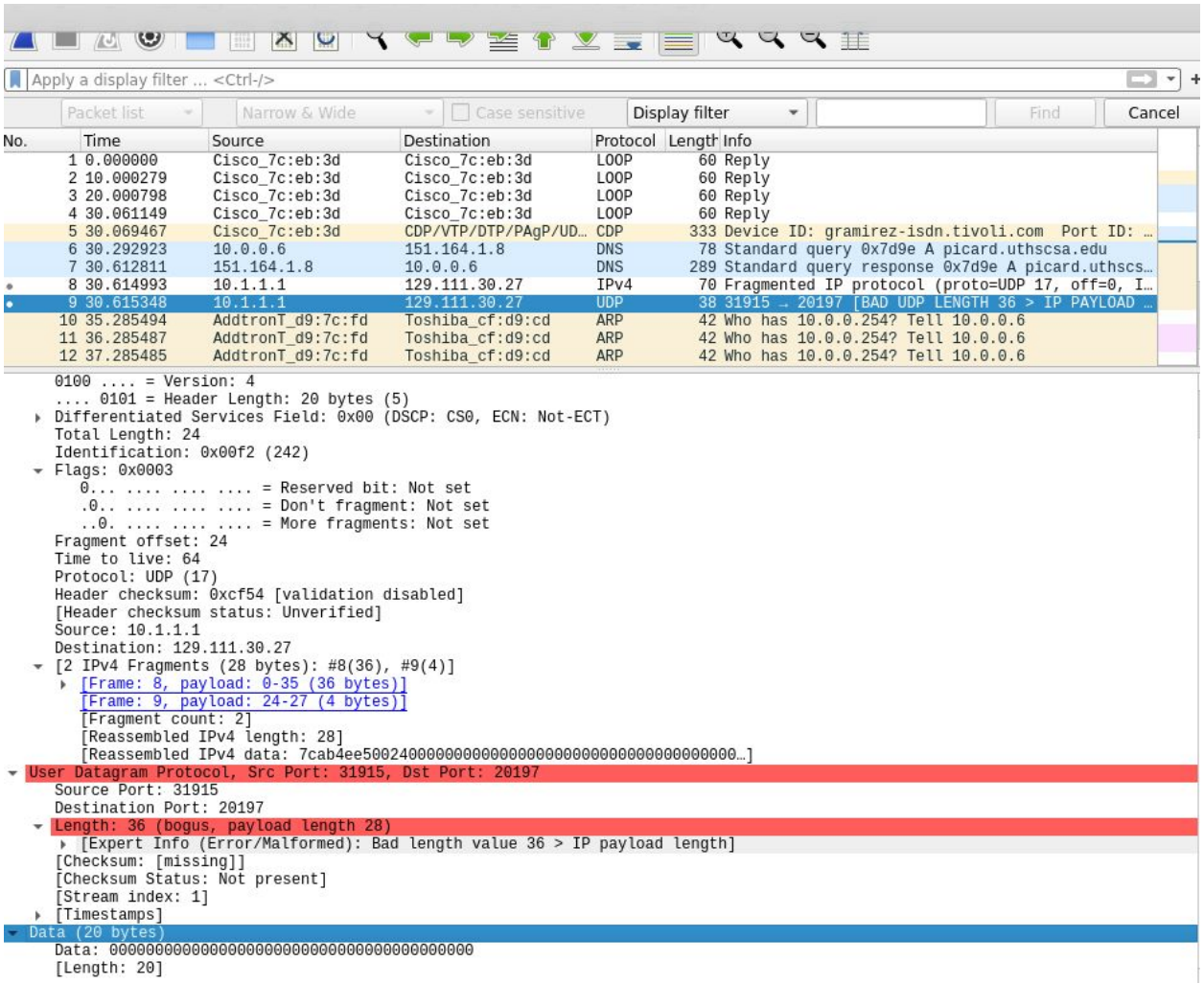


From the wireshar wiki, i have analyzed a teardrop attack. the traffic is recorder in the teadrop.cap file downloaded from the wiki and recorded the traffic during an IP fragmentation attack (aka. Teardrop). During this attack, the victim receives a fragmented packet where the offset field, which indicates the starting position or offset of the data relative to the data of the original unfragmented packet, is not consistent. In fact, when the sum of the offset and size of one fragmented packet differs from that of the next fragmented packet, the packets overlap and the server attempting to reassemble the packet might crash. This was a vulnerability of TCP/IP which affected several OSs such as Windows95, Windows NT and versions of the Linux kernel prior to 2.1.63.

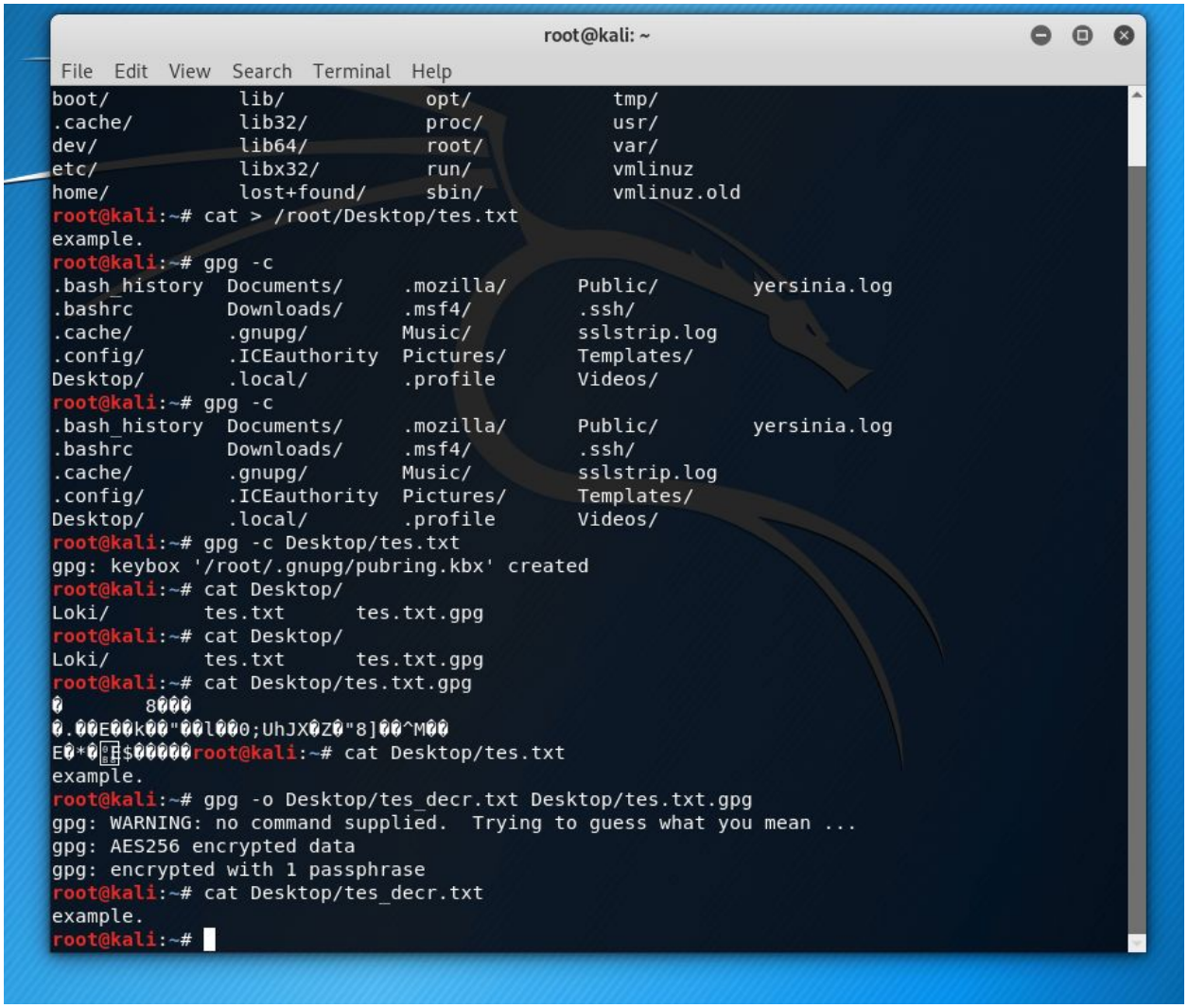
As we can notice fro the following 2 screenshots, the packet n. 8 is the first fragment of a fragmented packet (offset= 0 and More fragments= set). It contains 36 bytes of data but when packet n. 9 is sent (second and last fragment because the flag More fragments= not set), we

can see that the offset is 24 bytes. This will result in an incapacity of the server to reassemble the packet and will overlap the data.

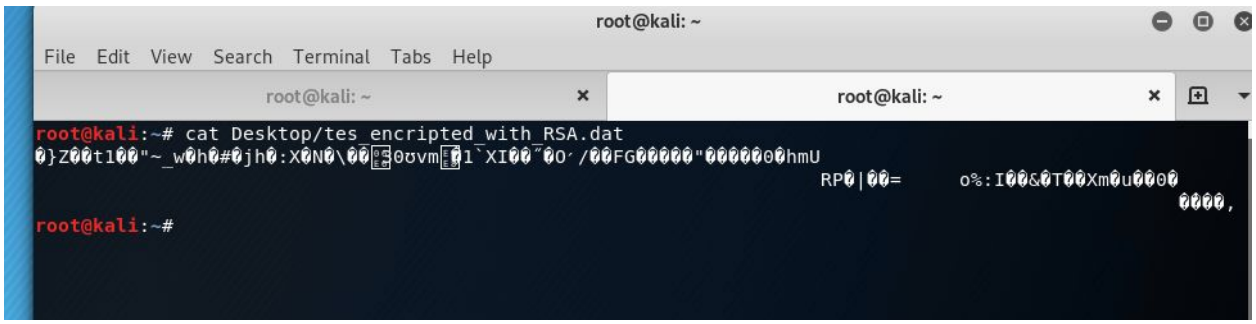


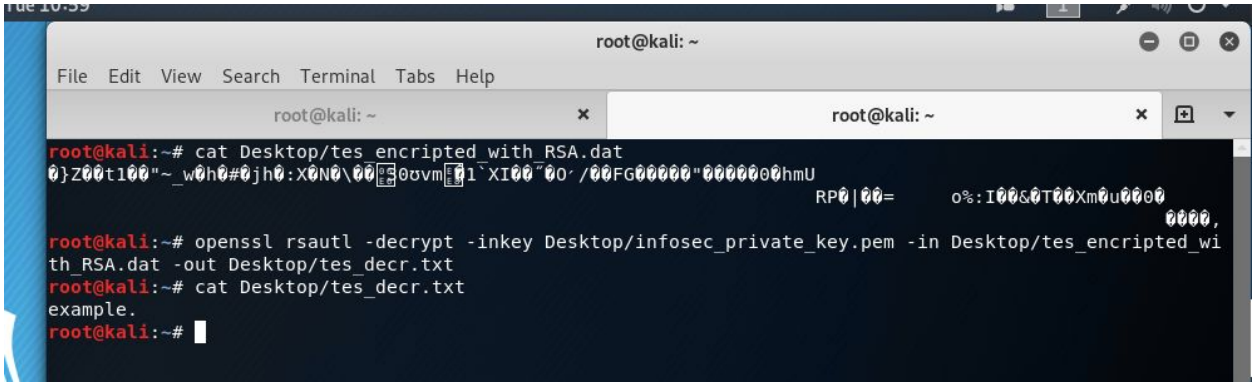

Apply a display filter ... <Ctrl-/>
Packet list
Narrow & Wide
Case sensitive
Display filter
Find
Cancel
No. Time Source Destination Protocol Length Info
1 0.000000 Cisco_7c:eb:3d Cisco_7c:eb:3d LOOP 60 Reply
2 10.000279 Cisco_7c:eb:3d Cisco_7c:eb:3d LOOP 60 Reply
3 20.000798 Cisco_7c:eb:3d Cisco_7c:eb:3d LOOP 60 Reply
4 30.061149 Cisco_7c:eb:3d Cisco_7c:eb:3d LOOP 60 Reply
5 30.069467 Cisco_7c:eb:3d CDP/VTP/DTP/PAgP/UD… CDP 333 Device ID: gramirez-isdn.tivoli.com Port ID: …
6 30.292923 10.0.0.6 151.164.1.8 DNS 78 Standard query 0x7d9e A picard.uthscsa.edu
7 30.612811 151.164.1.8 10.0.0.6 DNS 289 Standard query response 0x7d9e A picard.uthscs…
8 30.614993 10.1.1.1 129.111.30.27 IPv4 70 Fragmented IP protocol (proto=UDP 17, off=0, I…
9 30.615348 10.1.1.1 129.111.30.27 UDP 38 31915 → 20197 [BAD UDP LENGTH 36 > IP PAYLOAD …
10 35.285494 AddtronT_d9:7c:fd Toshiba_cf:d9:cd ARP 42 Who has 10.0.0.254? Tell 10.0.0.6
11 36.285487 AddtronT_d9:7c:fd Toshiba_cf:d9:cd ARP 42 Who has 10.0.0.254? Tell 10.0.0.6
12 37.285485 AddtronT_d9:7c:fd Toshiba_cf:d9:cd ARP 42 Who has 10.0.0.254? Tell 10.0.0.6
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 24
Identification: 0x00f2 (242)
Flags: 0x0003
0... .... = Reserved bit: Not set
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set
Fragment offset: 24
Time to live: 64
Protocol: UDP (17)
Header checksum: 0xcf54 [validation disabled]
[Header checksum status: Unverified]
Source: 10.1.1.1
Destination: 129.111.30.27
[2 IPv4 Fragments (28 bytes): #8(36), #9(4)]
[Frame: 8, payload: 0-35 (36 bytes)]
[Frame: 9, payload: 24-27 (4 bytes)]
[Fragment count: 2]
[Reassembled IPv4 length: 28]
[Reassembled IPv4 data: 7cab4ee50024000000000000000000000000000000000…]
User Datagram Protocol, Src Port: 31915, Dst Port: 20197
Source Port: 31915
Destination Port: 20197
Length: 36 (bogus, payload length 28)
[Expert Info (Error/Malformed): Bad length value 36 > IP payload length]
[Checksum: [missing]]
[Checksum Status: Not present]
[Stream index: 1]
[Timestamps]
Data (20 bytes)
Data: 0000000000000000000000000000000000000000
[Length: 20]

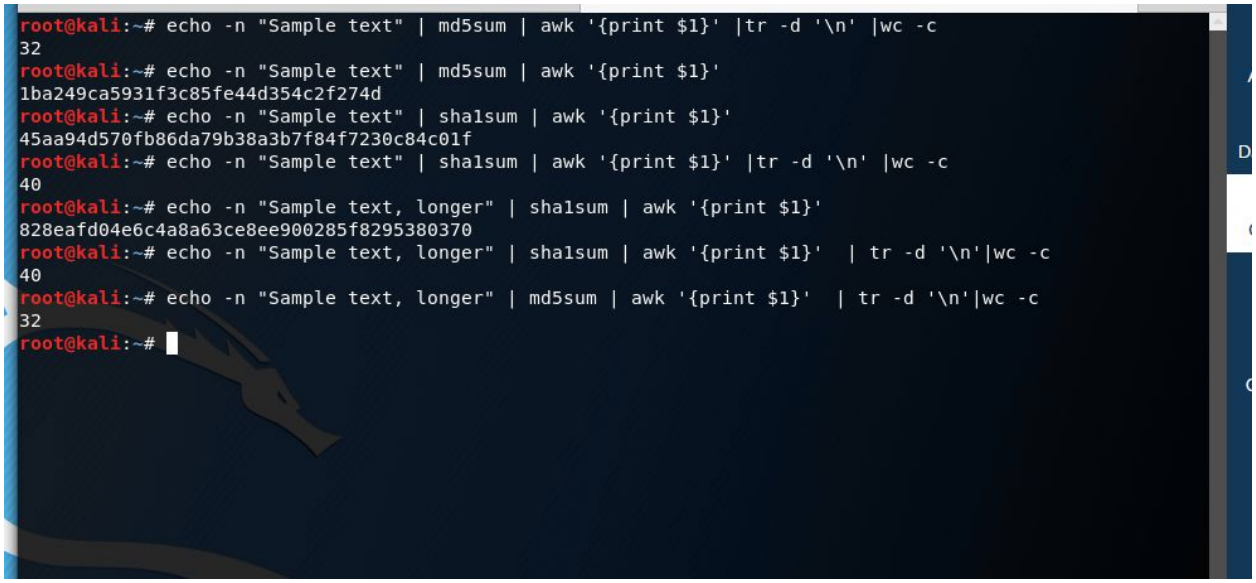
PART 3

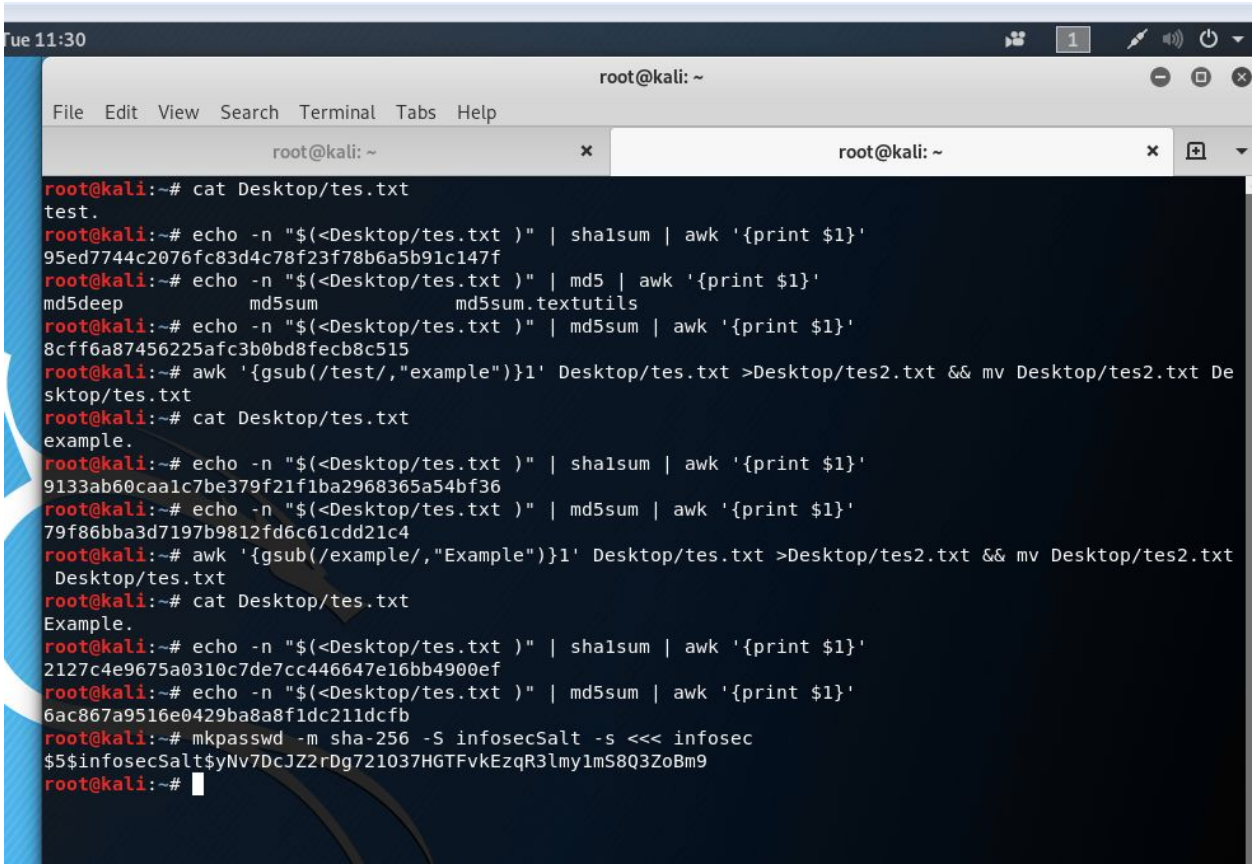

root@kali: ~
File Edit View Search Terminal Help
boot/ lib/ opt/ tmp/
.cache/ lib32/ proc/ usr/
dev/ lib64/ root/ var/
etc/ libx32/ run/ vmlinuz
home/ lost+found/ sbin/ vmlinuz.old
root@kali:~# cat > /root/Desktop/tes.txt
example.
root@kali:~# gpg -c
.bash_history Documents/ .mozilla/ Public/ yersinia.log
.bashrc Downloads/ .msf4/ .ssh/
.cache/ .gnupg/ Music/ sslstrip.log
.config/ .ICEauthority Pictures/ Templates/
Desktop/ .local/ .profile Videos/
root@kali:~# gpg -c
.bash_history Documents/ .mozilla/ Public/ yersinia.log
.bashrc Downloads/ .msf4/ .ssh/
.cache/ .gnupg/ Music/ sslstrip.log
.config/ .ICEauthority Pictures/ Templates/
Desktop/ .local/ .profile Videos/
root@kali:~# gpg -c Desktop/tes.txt
gpg: keybox '/root/.gnupg/pubring.kbx' created
root@kali:~# cat Desktop/
Loki/ tes.txt tes.txt.gpg
root@kali:~# cat Desktop/
Loki/ tes.txt tes.txt.gpg
root@kali:~# cat Desktop/tes.txt.gpg
8
E$
root@kali:~# cat Desktop/tes.txt
example.
root@kali:~# gpg -o Desktop/tes_decr.txt Desktop/tes.txt.gpg
gpg: WARNING: no command supplied. Trying to guess what you mean ...
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
root@kali:~# cat Desktop/tes_decr.txt
example.
root@kali:~#

```
root@kali:~# openssl genrsa -out Desktop/infosec_private_key.pem 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
.........+++++
......+++++
e is 65537 (0x010001)
root@kali:~# openssl rsa
rsa     rsautl
root@kali:~# openssl rsa -
-check    -des3     -idea     -inform   -noout    -outform  -passout  -pubout   -text
-des      -engine   -in       -modulus  -out      -passin   -pubin    -sgckey
root@kali:~# openssl rsa -i
-idea    -in      -inform
root@kali:~# openssl rsa -i
-idea    -in      -inform
root@kali:~# openssl rsa -in Desktop/
infosec_private_key.pem  tes_decr.txt             tes.txt.gpg
Loki/                    tes.txt
root@kali:~# openssl rsa -in Desktop/
infosec_private_key.pem  tes_decr.txt             tes.txt.gpg
Loki/                    tes.txt
root@kali:~# openssl rsa -in Desktop/infosec_private_key.pem  -out Desktop/infosec_public_key.pem -out
-out      -outform
root@kali:~# openssl rsa -in Desktop/infosec_private_key.pem  -out Desktop/infosec_public_key.pem -outfo
rm PEM -p
-passin   -passout  -pubin    -pubout
root@kali:~# openssl rsa -in Desktop/infosec_private_key.pem  -out Desktop/infosec_public_key.pem -outfo
rm PEM -pub
-pubin   -pubout
root@kali:~# openssl rsa -in Desktop/infosec_private_key.pem  -out Desktop/infosec_public_key.pem -outfo
rm PEM -pub
-pubin   -pubout
root@kali:~# openssl rsa -in Desktop/infosec_private_key.pem  -out Desktop/infosec_public_key.pem -outfo
rm PEM -pub
-pubin   -pubout
root@kali:~# openssl rsa -in Desktop/infosec_private_key.pem  -out Desktop/infosec_public_key.pem -outfo
rm PEM -pubout
writing RSA key
root@kali:~# openssl rsautl -encrypt -inkey Desktop/infosec_public_key.pem -pubin -in Desktop/tes
tes_decr.txt  tes.txt       tes.txt.gpg
root@kali:~# openssl rsautl -encrypt -inkey Desktop/infosec_public_key.pem -pubin -in Desktop/tes
tes_decr.txt  tes.txt       tes.txt.gpg
root@kali:~# openssl rsautl -encrypt -inkey Desktop/infosec_public_key.pem -pubin -in Desktop/tes.txt -o
ut Desktop/tes_encripted_with_RSA.dat
root@kali:~#
```




root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
root@kali: ~
root@kali:~# cat Desktop/tes_encripted_with_RSA.dat
root@kali:~# openssl rsautl -decrypt -inkey Desktop/infosec_private_key.pem -in Desktop/tes_encripted_with_RSA.dat -out Desktop/tes_decr.txt
root@kali:~# cat Desktop/tes_decr.txt
example.
root@kali:~#


root@kali:~# echo -n "Sample text" | md5sum | awk '{print $1}' |tr -d '\n' |wc -c
32
root@kali:~# echo -n "Sample text" | md5sum | awk '{print $1}'
1ba249ca5931f3c85fe44d354c2f274d
root@kali:~# echo -n "Sample text" | sha1sum | awk '{print $1}'
45aa94d570fb86da79b38a3b7f84f7230c84c01f
root@kali:~# echo -n "Sample text" | sha1sum | awk '{print $1}' |tr -d '\n' |wc -c
40
root@kali:~# echo -n "Sample text, longer" | sha1sum | awk '{print $1}'
828eafd04e6c4a8a63ce8ee900285f8295380370
root@kali:~# echo -n "Sample text, longer" | sha1sum | awk '{print $1}' | tr -d '\n'|wc -c
40
root@kali:~# echo -n "Sample text, longer" | md5sum | awk '{print $1}' | tr -d '\n'|wc -c
32
root@kali:~#



Wireshark Alternatives

Omnipeek : Omnipeek offers the analytical capabilities superior to those of Wireshark. In fact Omnipeek can scan packets for signs of trouble or detect changes in transfer speeds and then trigger alerts. Omnipeek is not opensource and can run only on WIndows systems.
Ettercap: Ettercap can detect other hacker activities and intrusion, so it is very useful for system defense and, moreover, can generate by default several attacks. It can also identify malicious users and isolate them from the network resulting then in a more powerful tool than wireshark.