UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

COMMUNICATIONS AND SYSTEMS ENGINEERING

Richard Adeyinka Adeyeye

DATA SECURITY IN THE CLOUD: Study and Simulations

Master’s thesis for the degree of Master of Science in Technology submitted for inspection, Vaasa, 01.04.2018.

Thesis Supervisor	Professor Mohammed Elmusrati
Thesis Instructor	Bahaa Eltahawy

CONTENTS

ABBREVIATIONS

AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
CPDP	Computers Privacy and Data Protection
DES	Data Encryption Standard
DVWA	Damn Vulnerability Web Application
DCE	Direct Code Execution
DDoS	Distributed Denial of Service
GF	Gallio’s Field
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Acts
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
ICMP	Internet Control Message Protocol
IDEA	International Data Encryption Algorithm
IDS	Intrusion Detection System
IP	Internet Protocol
LLC	Logical Link Control
MAC	Media Access Control
NAS	Network Attached Storage
NIST	National Institute of Standards and Technology
NS3	Network Simulator
NSC	Network Simulation Code
OSI	Open System Interconnection
PaaS	Platform as a Service
PCI	Payment Card Industry
PGP	Pretty Good Privacy
PTCP	Parallel Transport Control protocol
RTCP	Real-Time Transfer Control Protocol
SaaS	Software as a Service

SAN	Storage Area Network
SDN	Software Defined Network
STCP	Sensor Transmission Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XML	Extensible Markup Language

LIST OF FIGURES Page

LIST OF TABLES Page

UNIVERSITY OF VAASA
Faculty of Technology

Author:	Richard Adeyinka Adeyeye
Topic of the thesis:	Data Security in the Cloud: Study and Simulations
Degree:	Master of Science in Technology
Master's Programme:	Communication and Systems Engineering
Supervisor:	Professor Mohammed Salem Elmusrati
Instructor:	Bahaa Eltahawy
Year of entering the University:	2015
Year of completing the thesis:	2018
Number of pages:	80

ABSTRACT

Cloud technology is a nascent technology, thriving in information communication and data storage, and is still under development. Securing the communication links and data has been very paramount to the development of this technology and system. Various techniques, methods and technologies have been implemented in order to secure this system. Security of the cloud has recently witnessed much attention, as there have on-going research and studies towards the development of more potent solutions. Cryptography is one of the feasible and in demand solutions here, as it offers a set of security measures such as confidentiality, integrity and availability.

This thesis work is aimed at understanding the data security in the cloud systems, and the various security threats associated with such technology. To better understand this, a thorough literature review is conducted on a cloud technology, and some of the cloud attacks are simulated. The distributed denial of service is simulated using NS3 and EstiNet, ARP poisoning is simulated using Ettercap, and the SQL injection is simulated using Damn Vulnerable Web Application. At the end of the task, the simulation scenarios were analyzed for better understanding and the observations were concluded. It was found out that these attacks are threat to the communication channels, network bandwidth, and the information being transferred. It was also realized that these attacks could be managed effectively using appropriate cryptographic techniques or technologies to block any unauthorised access to the network.

KEYWORDS: cloud systems, cryptography, information security, cloud attacks, simulations, NS3, Ettercap, EstiNet.

1. INTRODUCTION

Cloud computing is defined as the practice of using computing resources and services deployed on the Internet to manage, store and process data on demand instead of the local server. There are many existing types following this model, such as the platform as a service (PaaS), infrastructure as a service (IaaS), software as a service (SaaS) and any other services that could be moved to the cloud. The term data storage is the act or means of storing data (information) for present or future use. The need for data storage is fast growing, and the demand for the service is also increasing. There are many ways of storing data, for example, stored in an external hard drive, magnetic disk, magnetic tape, memory sticks and/or on the cloud, for big data storage, accessible from anywhere, and for further processing capabilities. In the market level, there are quite a few companies rendering data or information storage systems, among whom are Microsoft offering Azure, Amazon data Storage, Alibaba cloud Virtual Hosting, SugarSync, Carbonite, IDrive, BackBlaze, Dropbox, Tresorit and others for commercial purposes.

Cloud computing is a trending topic in Information Technology (IT), with which much research and innovation are going on currently. The field focuses on the development of IT infrastructures and resources to provide better and new architectures for the storage system. The term cloud computing was defined by Gerald Kaefer of Siemens Technology as the model to allow easy on-demand network access to a shared pool of customizable computing resources (like servers, storage, applications and services, networks) which can be made available quickly and allowed with little management influence or service provider reciprocal actions.

Cloud computing can be classified into four common models; the public, private, community and hybrid clouds. The cloud computing makes use of virtualization technology, which is a new technology aimed at reducing business costs and encourages multi-tenancy. Virtualization is one of the basic technologies to make cloud computing function properly. There are different types of virtualization, among whom are data virtualization (which enables organizations to have processing capabilities to gather data together from different sources and process according to user demand at the appropriate time), desktop virtualization (permits central administrator to carry out multiple configurations, updates and security tests on all virtual desktops), server virtualization (enabling a server to carry out multiple specific functions upon partitioning into different components) and Operating system virtualization (enables the deployment of different operating systems on a single machine) (Red Hat Inc. 2018). It is a technology that dissociates functions from hardware and cloud computing depends on this separation. This new technology is no different than previous ones, as with the benefits it brings, features, and it also comes with a new set of security challenges. This is the aim of this research work.

Due to the significant increase in the amount of data and resources being utilized by consumers from the public and private sectors, there have been questions for tighter and more robust cloud services. We require security to prevent unauthorized access to data being stored either for private or for commercial purposes. Protecting data is crucial, and is a way of protecting its confidentiality, integrity and authenticity (CIA) against attack.

The verification of the access to the storage system is a way of keeping data safe from any user of the services or the system resources. The complexity of the cloud system is based on its architecture, and the importance associated with its use, which requires a strong security and authentication algorithms in order to avoid economic loss. There are several ways of securing a cloud data storage system, among the commonly used security and authentication algorithms are, the cryptographic algorithm, intelligent based authentication, firewall creation and others.

Data security is critical to the development of cloud system and unarguably a major concern in IT, this is because data are located everywhere in various machines and storage facilities. It is therefore important that the cloud service providers protect the essential qualities, the CIA, of data by implementing some security measures such as cryptographic techniques which use algorithms to authenticate data and keeping them secret. This implementation has been effective over the years and it is still in use. The cryptographic technique protects data by encryption and the security technique could be offered either by symmetric, asymmetric or mixture of both (Ankit Jain 2004).

Cryptography, authentication, and authorization techniques can be tested by means of implementing them using simulators. Basically, the simulators are designed using mathematical formulas, algorithms, models to imitate real-life network models in a virtual environment. There are several tools that are being used to simulate the various cloud attacks; these are used for research, academic and industrial purposes. Some of the tools are available as open source and for commercial use. The commercially available simulating tools are mostly used by companies to analyse their various networks; academic institutions for teaching, and research purposes, and for individuals to learn more about security on their own. Choosing a simulator depends on the user’s interest, either to analyse the quality of service, measure the throughput, understand the network performance or to measure effects of various attacks on the computer networks.

This thesis is aimed at getting a better understanding of the security challenges in the cloud system. This will be achieved by studying cloud computing, data security, and by simulating different cloud attacks using different simulating tools like Network Simulator 3 (NS3), EstiNet, Ettercap and Damn Vulnerability Web Application (DVWA). At the end of this thesis, the project would be able to answer the following questions:

- What are the security threats to the cloud system?
- What are the various cloud attacks?
- What are effects of these attacks on communication systems?
- Why and how cryptographic techniques could be implemented in the cloud security?

2. CLOUD SYSTEM

2.1. Cloud Architecture

The cloud storage architecture can be classified into four different actors, which are the data owner, the user, the cloud server and the third-party auditor (Vairagade Sachin et. al 2012). This is represented in Figure 1 below.

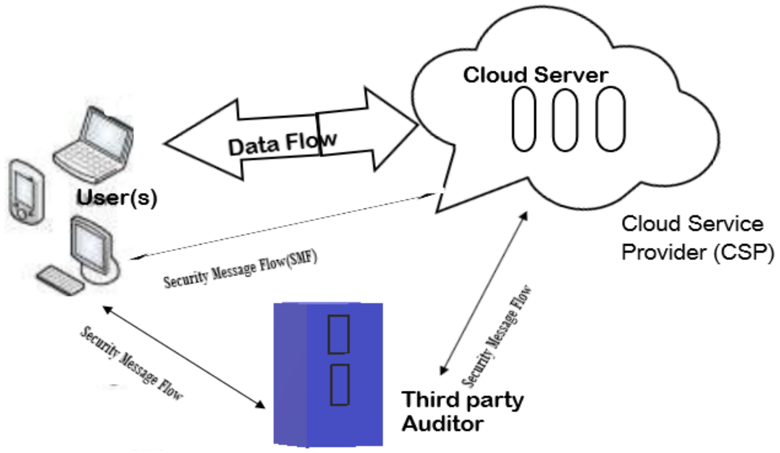


Figure 1. Typical Cloud Storage Architecture (Yassin Ali A. et al 2014).

The User(s) could be individual, private or commercial organisations, who depend on the remote cloud server for storage, maintenance and protection of their data while the Cloud Service Provider is/are responsible for the storage, maintenance and the protection of data being sent by the user. The Third-party Auditor has authorized bodies who have rights to access the cloud storage security, the remote cloud server upon authorization of the data owner (Yassin Ali A. et al 2014).

2.2. Cloud Models

2.2.1. The Public Cloud

It is a cloud infrastructure owned by an organization selling cloud services and made accessible to the public. The resources are rendered on-demand basis and dynamically over the Internet. The model allows both small and medium-sized enterprises to manage their businesses and reduce the creation of more data centres. The model features have the following benefits:

i. helps companies to save costs from spending too much on purchasing, managing and maintaining own On-premises IT resources.
ii. scalable to meet user's needs and workload.
iii. reduces resources wastage because consumers only pay per usage.
iv. can also be set up faster compared to the on-premises infrastructures.

2.2.2. The Private Cloud

It is a cloud infrastructure managed mainly by an organization on their own private network. It could also be managed by a third party either on or off premises. Virtual private cloud happens to be a good solution for maintaining a cost-effective steady task (Online Tech Inc, 2018) and highlighted some other benefits such as:

i. designed to ensure a high level of security by means of preventing other customers in the same data centre from connecting, as the cloud is assigned to a single user.
ii. its IT resources (hardware, network, and storage) performance can be defined and personalized in the private cloud.
iii. offers use on-demand like the public cloud.
iv. complies with (Payment Card Industry) PCI, (Health Insurance Portability and Accountability Act) HIPAA, and Sarbanes Oxley standards and regulations,

which are offered through a virtual private network, or completely private cloud for an assigned user.

v. can be deployed quickly, and highly scalable to meet changing needs.

2.2.3. The Community Cloud

This is also a cloud infrastructure being managed by several organizations and supports certain group sharing a common goal (for example mission, policy, compliance, regulations etc).

2.2.4. The Hybrid Cloud

This cloud infrastructure is the combination of two or more private, or public clouds services joined together by a standard technology for data and application flexibility as a single service. Hybrid cloud allows an organization to use wider varieties of IT services. The benefits it offers are:

i. allows companies to implement on-premises private cloud to host confidential workloads and deploy third-party public cloud to host less sensitive resources.
ii. ability and support for big data processing.
iii. allows companies to use a wider mix of IT services.

2.3. The Cloud Computing Architecture

The cloud computing architecture refers to a written permission, a document that allows communication between stakeholders, earlier high-level design decisions documentations, components, design reuse and patterns between projects in cloud systems (Gerald Kaefer 2010). According to Gerald Kaefer, cloud computing architecture, as shown in Figure 2 below, is also defined as the structure of the system that consists of on-premises and cloud resources, services, software components, geo-location and the relationships between them.

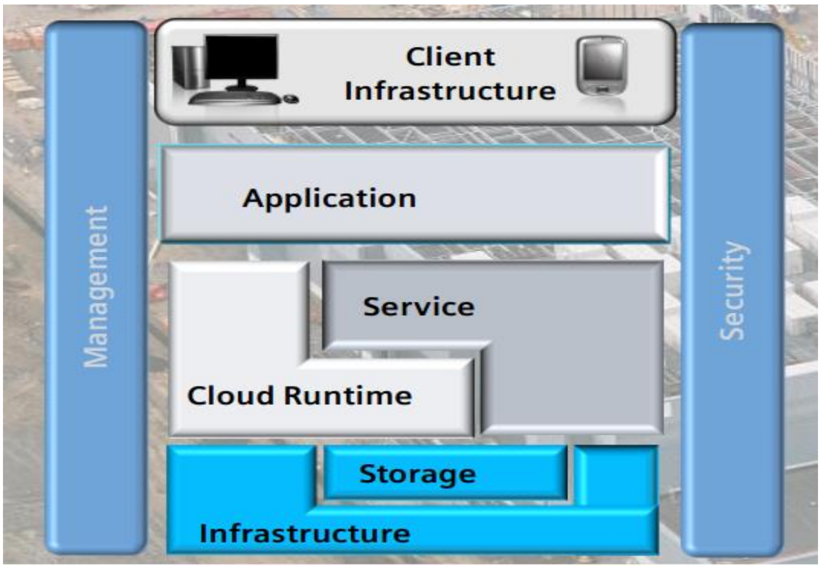


Figure 2. Cloud Computing Architecture (Gerald Kaefer 2010).

2.4. The Cloud Computing Services and Characteristics

These are Information Technology services made available to the end-users upon request. The services can be categorized into Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS). The cloud service models are as described in Figure 3 below. SaaS is an on-demand software, in which applications are presented by the service provider and accessible by the end user over the Internet (Stephen R. et al 2012: p10) e.g. Microsoft office 365, Salesforce and others. PaaS is the rendering of operating systems and other attached services over the Internet without the need to download or install (Stephen R. et al 2012: p10). Given examples are the Microsoft Azure, Elastic Beanstalk and Amazon Web Services. Finally, IaaS is a form of cloud computing for rendering virtualized computing resources provided by the service provider hosting the infrastructural components normally available in an on-premises data centre such as; storage, servers and networking hardware, and made accessible on the Internet (Michelle Boisvert and Stephen J. Bigelow 2013). IBM Bluemix, Microsoft Azure and Amazon Web Services are common examples.

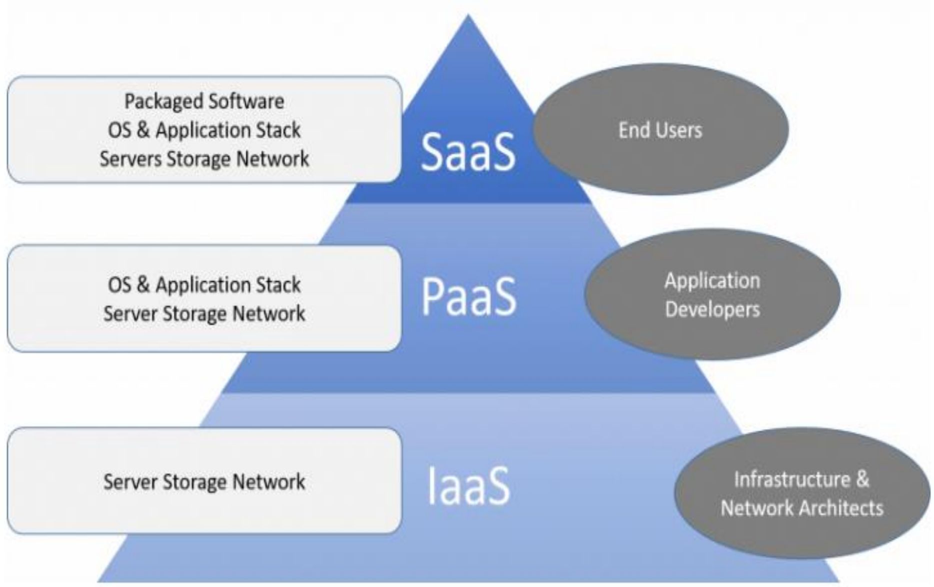


Figure 3. Typical Cloud Service Model (Arron Fu 2017)

The most common characteristics of typical cloud systems are:

i. It provides computing services itself on demand without any influence from human or the service provider.

ii. It has a large network, which can be accessible by any electronic device regardless of place and time.

iii. It has large computing resources like network bandwidth, power and so on, in the physical or virtual pools, that can be assigned automatically on demand by the authorized end-user.

iv. It features unlimited capability for self-growth as the demand grows.

v. The services are readily available without any interruption. They are regularly monitored, controlled and managed.

3. CLOUD SECURITY

The cloud computing security is a rapidly growing service which is offering many of the same purposes as the conventional IT security such as securing important information from unauthorized access, leakage and data obliteration (amazon web services Inc, 2018). Software-based security appliances are being used to oversee and guide information flowing in and out of the cloud resources and it is not compulsory to own physical storage facilities or servers. Cloud services are being governed by different bodies and parastatals according to the law of the country in which the data centres are located and with strict regulatory standards.

3.1. Cloud Computing and Security Challenges

The reasons for the use of cryptographic encryption algorithms are to ensure security and to protect the integrity of data in the cloud storage. The encryption techniques are applied to achieve the data protection standards, which are; confidentiality (blocking unauthorised access to protected data), integrity (protecting data from illegal modification or deletion), availability (data is available to authorised personnel on request), authorization (only authorised user has access to the data), and non-repudiation (that the sender can not deny that the data/information is being sent from him/her) of the secured data. The term encryption is the method of converting a readable text into an unreadable text (meaningless text) using special key and decryption is vice versa. The most commonly used private keys are the Data Encryption Standard (DES), Advanced Encryption Standard (AES) and the International Data Encryption Algorithm (IDEA) (AbdElnapi et al 2016).

The cryptographic algorithms can be categorized into symmetric, asymmetric, and hybrid algorithms. The symmetric algorithm, which is known as private key encryption, is a method of encrypting and decrypting data using one common key. This is illustrated in Figure 4 below

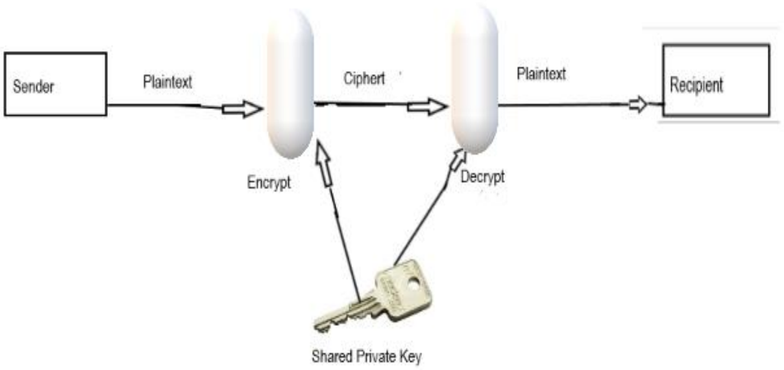


Figure 4. Typical Symmetric Private Key Cryptography (Stallings 2011)

On the other hand, the asymmetric algorithm uses two different keys, the private and public keys, for encryption and decryption respectively. This is illustrated in Figure 5 below.

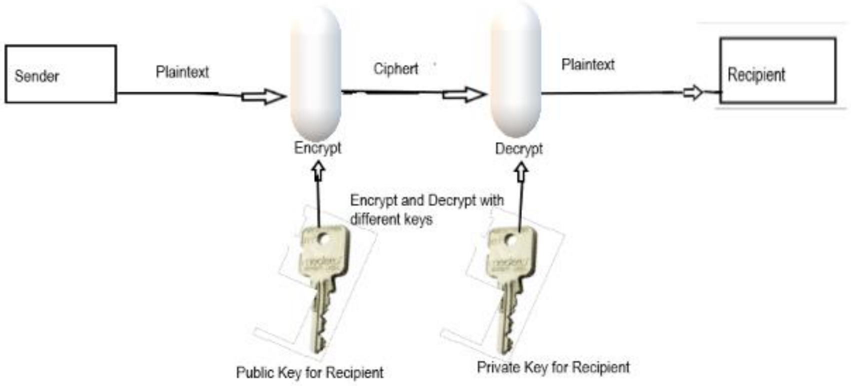


Figure 5. Typical Asymmetric Key Cryptography (Stallings 2011)

In its hybrid model, two or more algorithms are combined and used simultaneously. This model is the most effective encryption algorithm to avoid brute force attack and to apply confidentiality to shared data. It works in such a way that the receiver has the private key and the sender has received the public key, similar to the SSL/TLS protocol in the web browser security; where the web browser has the capability to generate a random symmetric key, encrypt it with the public key and then send it to the server.

In more details, Data Encryption Standard is a commonly used encryption algorithm until now. It is published as FIPS-46 in the Federal Register by National Institute of Standards and Technology (NIST) in January 1977 (Khan Shakeeba et al 2015). It uses a 64- bit block and a 56-bits key size algorithm. The DES block cipher as shown in

Figure 6 below uses a key for encryption and decryption, and the algorithm is administered to a block of a data concurrently instead of serving one bit at a time. DES uses the 64-bit key, but eight of the key are used for parity checks productively reducing the key to 56-bits. Therefore, it would require a maximum of 2^{56} guesses to get the correct key.

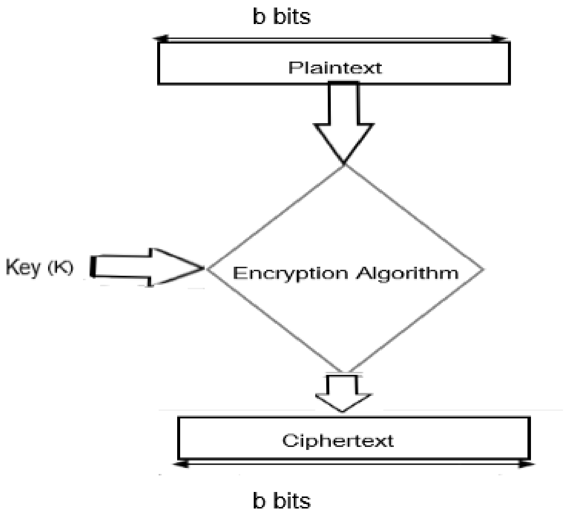
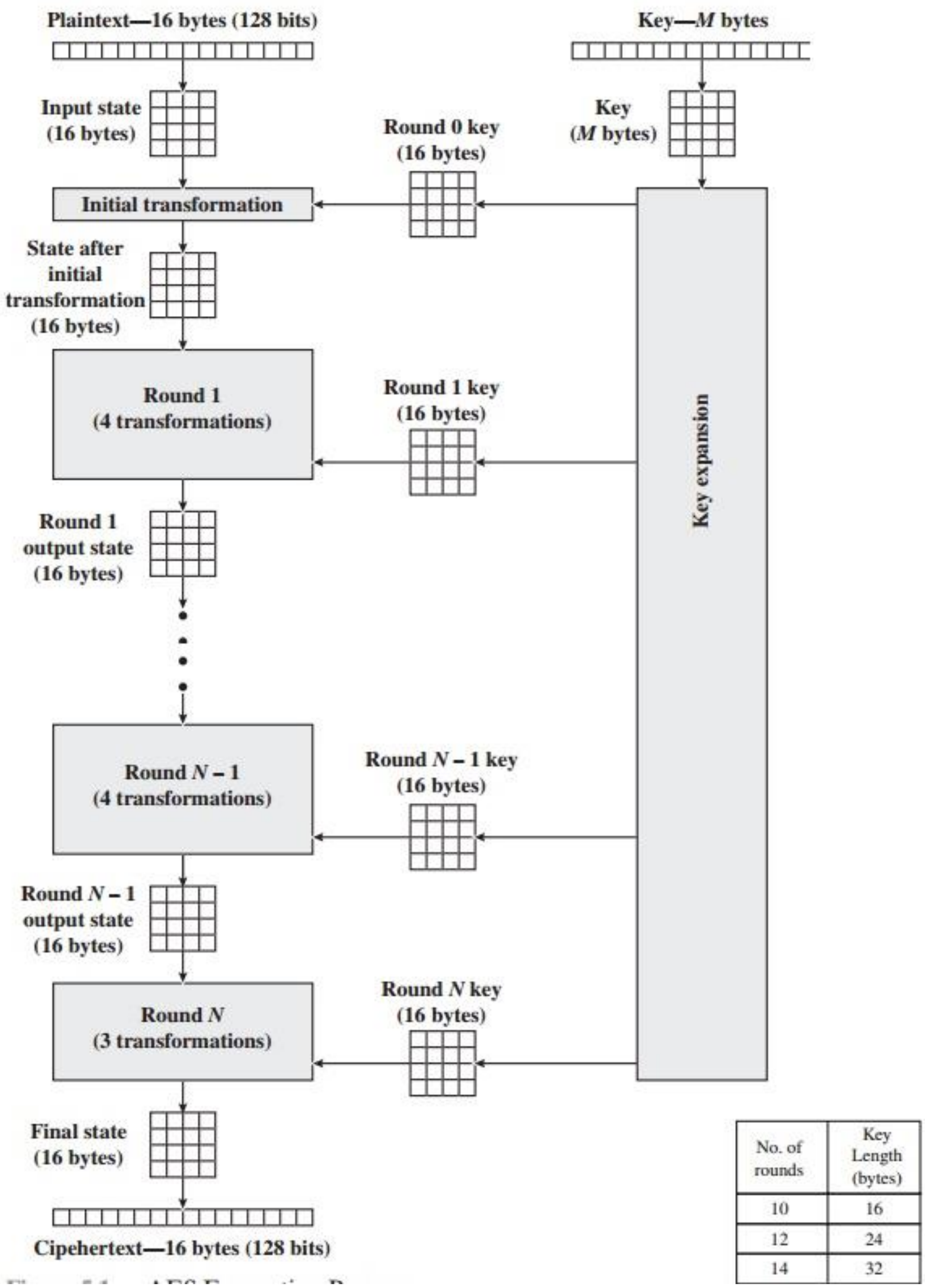


Figure 6. DES Block Cipher (Khan Shakeeba et al 2015)

AES was published in 2001 by the National Institute of Standards and Technology. It is a symmetric block cipher which intends to replace the Data Encryption Standard (DES) because of the latter’s well-known security weaknesses such as brute-force attacks for instance. The operations in AES are performed on byte basis. The algorithm makes use of mathematical operations such as addition, division and multiplication operations in AES are performed over the Galois Field GF(2^8). The key length of an AES can be 16, 24, or 32 bytes as in AES-128, AES-192 and AES-256 block ciphers respectively for encryption and decryption. The AES operational structure is shown in Figure 7 below.


Plaintext—16 bytes (128 bits)
Key—M bytes
Input state (16 bytes)
Key (M bytes)
Round 0 key (16 bytes)
Initial transformation
State after initial transformation (16 bytes)
Round 1 (4 transformations)
Round 1 key (16 bytes)
Key expansion
Round 1 output state (16 bytes)
Round N – 1 (4 transformations)
Round N – 1 key (16 bytes)
Round N – 1 output state (16 bytes)
Round N (3 transformations)
Round N key (16 bytes)
Final state (16 bytes)
Cipehertext—16 bytes (128 bits)

No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 7. AES Structure (Stallings, 2011: p175)

The cipher features N of rounds that depend solely on the length of the key in use, e.g. 10 rounds for a 16-byte key, 12 rounds for a 24-byte key and 14 rounds for a 32-byte key. The round steps in AES are; substitute bytes (replacing each byte by byte indexed by row and column of a 16x16 table), shift rows (byte circular shift to the left. such that, the first row is fixed, second row shifts by 1 byte to the left, third row shifts 2 bytes to the left and fourth row shifts 3 bytes to the left), mix column (matrix multiplication in GF(2^8) using prime polynomial $m(x) = x^8+x^4+x^3+x+1$), and add round keys (XOR state with 128-bits of the round key) (Stallings 2011: p176). The first N-1 rounds consist of four different transformation functions: SubBytes, ShiftRows, MixColumns and AddroundKeys, which are implemented as shown in Figure 8. For the final round, it has only three transformations, with initial single transformation (AddRoundKey) before the first round (Stallings 2011: p177).

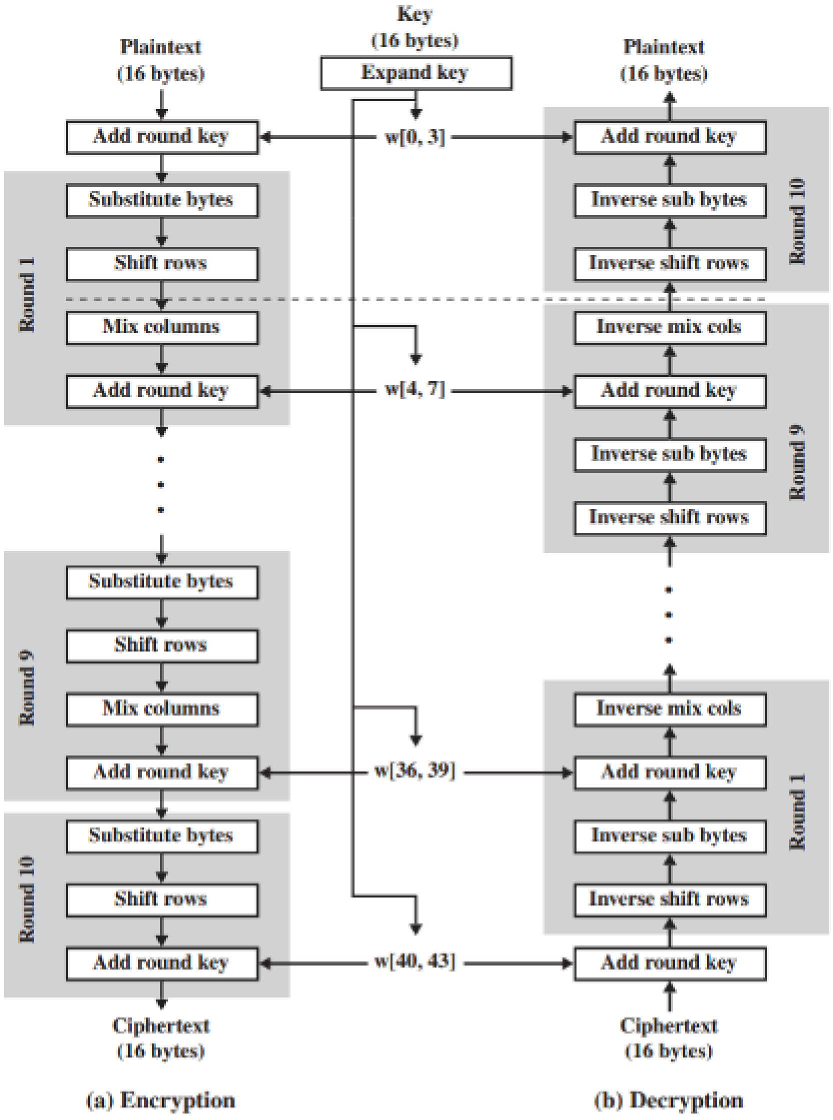


Figure 8. AES Encryption and Decryption (Stallings 2011: p178)

According to Stallings, to decrypt, we use inverse operations in many transformations. The key that is given as the input is then spread into an array of forty-four 32-bit words, w[i]. Four different words (128bits) are used as a round key for each round. The cipher starts with an AddRoundKey stage and then by nine rounds which also contain all the four stages followed by the tenth round of three stages for both encryption and decryption. This can also be seen in the encryption rounds shown in Figure 9 below.

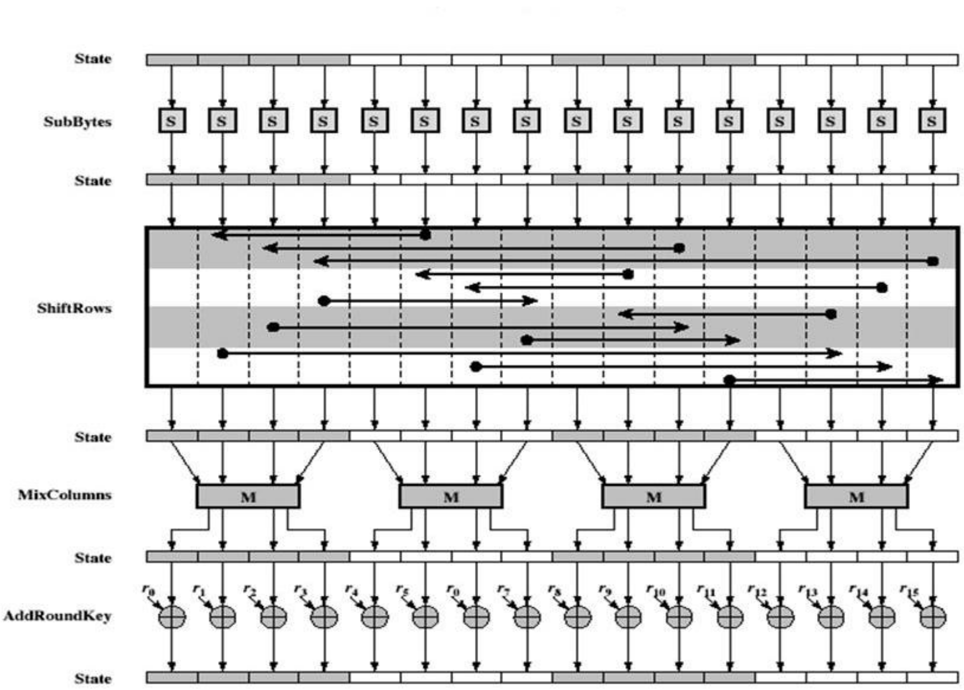


Figure 9. AES Encryption rounds (Stallings 2011: p179)

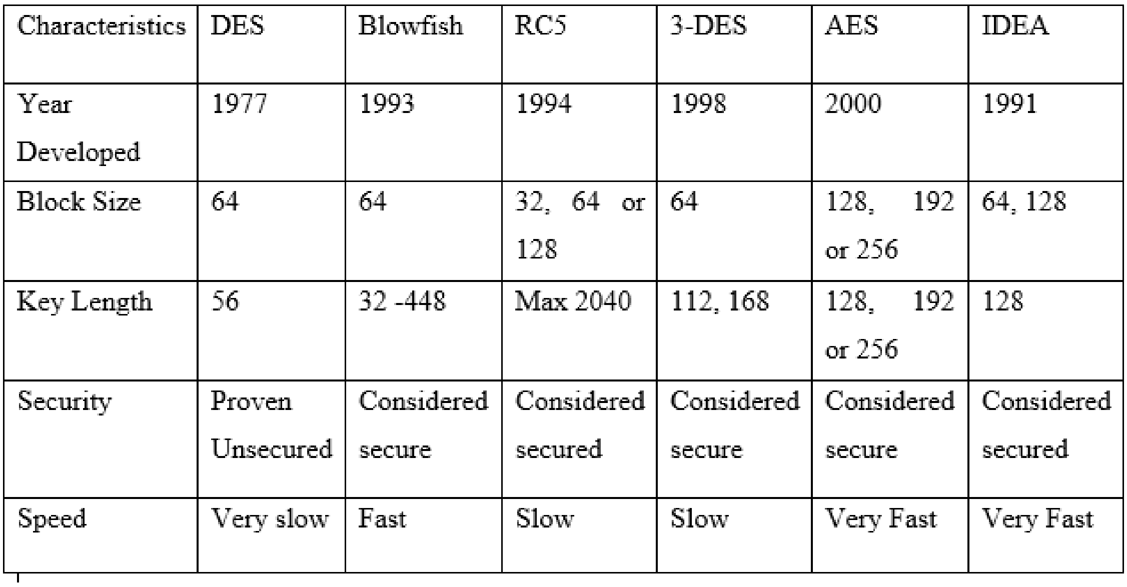
International Data Encryption Algorithm (IDEA) was developed in 1991, and it is now free of use. The algorithm makes use of symmetric block cipher, with 128 bits key and sub-keys are 16 bits. It has a block size of 64 which is divided into four 16-bit sub-blocks. The operations being used in IDEA are XOR, addition modulo and multiplication modulo. The decryption works similarly as the encryption, but rather the round key order is reversed.

Another form of symmetric block cipher designed by Bruce Schneier in 1993 is called Blowfish. It takes a variable-length key from 32 bits to 448 bits and it is a replacement of DES or IDEA and also faster than the two.

Also, Rivest Cipher 5 (RC5) is another symmetric-key block cipher algorithm which is considered secure. It has a 32, 64 or 128 bits block size, a key size of up to 2040 maximum, and 0-255 number of rounds. It was developed in 1994, iterative in structure, suitable for hardware or software, and easy to interpret due to its simple framework.

Additionally, triple DES (3-DES) is another symmetric encryption algorithm. It uses three definite DES keys with a total length of 168 bits. 3-DES works by taking three 56 bits keys (K1, K2 and K3), encrypting first with K1, decrypting next with K2 and encrypting with K3 for the last time (Jon Callas, 2004). 3-DES is weaker and slower than blowfish or AES. In table 1, the comparison among these cryptographic algorithms is given.

Table 1. Cryptography Algorithms Comparison (Randeep Kaur et al 2014).



Characteristics	DES	Blowfish	RC5	3-DES	AES	IDEA
Year Developed	1977	1993	1994	1998	2000	1991
Block Size	64	64	32, 64 or 128	64	128, 192 or 256	64, 128
Key Length	56	32 -448	Max 2040	112, 168	128, 192 or 256	128
Security	Proven Unsecured	Considered secure	Considered secured	Considered secure	Considered secure	Considered secured
Speed	Very slow	Fast	Slow	Slow	Very Fast	Very Fast

3.2. The Cloud Data and its Security

In cloud computing, data can be in three different states, which are data in use, data in transit and data at rest. The likely encryption algorithms which could be implemented to secure data in these states are also discussed in this section.

3.2.1. The Data in Use

Data in use is when the data is in temporary channel or storage facility. At this state, data requires security and should be protected. Data in this state is more susceptible and can be easily attacked because of its availability to different legitimate users in need of it. The way by which to protect data in this state, putting into consideration that it needs to be decrypted at first, is by moving it to a temporal secure location, then by ensuring the availability of a secure channel to the expected destination for the final user.

3.2.2. The Data in Transit

Data in transit is when the data is moving across the network from one point to the other. Data is most liable to attack in this state. However, this can be protected by using Secure Socket Layer (SSL), a protocol being used within the HTTPS, which is an updated, improved version of its former one, Transport Layer Security (TLS).

3.2.3. The Data at Rest

Data at rest is when the data is finally stored in the final storage facility (network attached storage (NAS), storage area network (SAN) and file servers), in the databases and file systems. In this state, to secure data, we may have to consider an effective algorithm that could provide help in encrypting it at rest. Data at this state can be protected using AES, RSA and any other protocols.

The Figure 10 below illustrates and explains data at the three different states further.

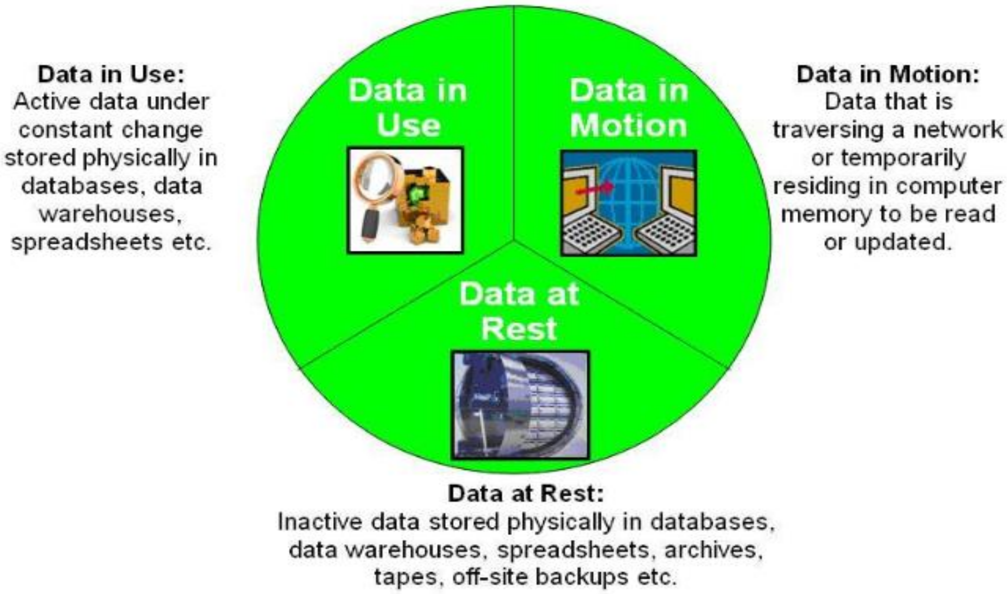

Data in Use:
Active data under constant change stored physically in databases, data warehouses, spreadsheets etc.
Data in Use
Data in Motion
Data in Motion:
Data that is traversing a network or temporarily residing in computer memory to be read or updated.
Data at Rest
Data at Rest:
Inactive data stored physically in databases, data warehouses, spreadsheets, archives, tapes, off-site backups etc.

Figure 10. Data at the Three Different States (ViSolve IT Security Team 2015)

Data at rest are mostly saved in the database. There are three major database vendors that provide database-level protection process in their products with the aim of encrypting the user-defined columns or the entire database files with a secure encryption algorithm so that attackers will not be able to read data in files without discerning the encryption key(s) (Martin Rakhmanov 2010). These three major database vendors are Oracle, Sybase and Microsoft. The database-level encryption functionality and a comparison between their characteristics are as shown in Table 2 below.

Table 2. Encrypting Data at Rest (Martin Rakhmanov 2010)

	Oracle Database 11.2	Sybase 15.5	Microsoft SQL Server 2008
Supported Encryption Algorithm	AES, Triple DES	AES	AES, Triple DES
Supported Key Length	128,192,256 bits for AES, 168 bits for Triple DES	128, 192, 256 bits	128, 192, 256 bits for AES, 192 bits for Triple DES
Encryption Scope	Tablespace, Column	Column	Database (Page level)
Key Assignment Granularity	Key per table	Key per column	Key per database
Encryption Key Storage	Wallet HSM (hardware security module), external PKCS#11 compatible key management system,	Database (not necessarily encrypted one)	Database
Special Key Management Database Roles	No	Yes	No
Special Permission Required to Read Encrypted Data	No	Yes	No
Optional 'salt' Support to Protect against Pattern Analysis	Yes	Yes	Entire files are encrypted: salt per value does not make sense

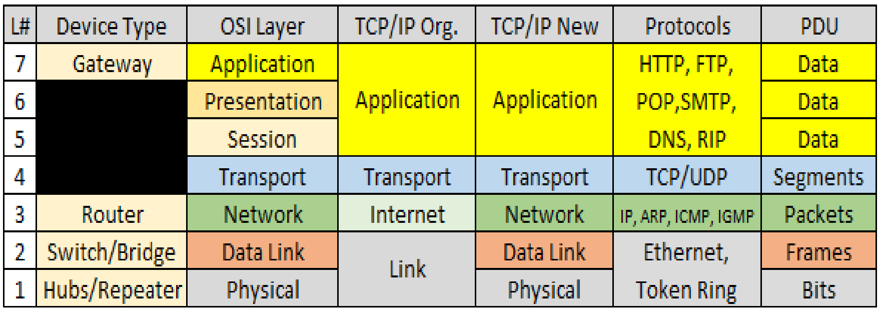
Additionally, another commonly used method for encrypting and decrypting data is PGP (Pretty Good Privacy). The encryption method uses symmetric and asymmetric keys to encrypt data being transferred across networks. The encryption key based approach greatly helps to secure data since the data credentials are not stored in the same place where the secure data is saved. Therefore, upon failure or malicious access to the disk or the storage facility, an unauthorised user or hacker would have no means to decrypt and make use of the data. PGP provides such level of data protection by making use of digital signature. The digital signature is a digitally created and validation public key encryption code securely affixed to the electronically relayed document to certify its contents and the identity of the sender (DocuSign 2018). This offers additional security measure between the sender and the receiver. Most common secure digital signature compared to the less secure ones are shown in Table 3 below.

Table 3. Secure and Unsecured networks (UC Regents 2017)

	Instead of….	Use….
Web Access	HTTP	HTTPS
File transfer	FTP, RCP	FTPS, SFTP, SCP, WebDAV over HTTPS
Remote Shell	telnet	SSH2 terminal
Remote desktop	VNC	radmin, RDP

3.3. The Attack Layers on the Open Systems Interconnection (OSI) Layer

The OSI model is an essential structure of protocols that enables two or more devices to exchange data together across the network system. An individual layer of the model is obliged for accomplishing a certain operation. The TCP/IP Model is shown in Figure 11. The model has seven OSI layers, the protocols in use in each layer respectively, for example, TCP/UDP at the transport layer, Ethernet, token ring at the physical-data-link layer and so also, we have the protocol data unit (PDU) for each layer.



L#	Device Type	OSI Layer	TCP/IP Org.	TCP/IP New	Protocols	PDU
7	Gateway	Application	Application	Application	HTTP, FTP, POP,SMTP, DNS, RIP	Data
6		Presentation				Data
5		Session				Data
4		Transport	Transport	Transport	TCP/UDP	Segments
3	Router	Network	Internet	Network	IP, ARP, ICMP, IGMP	Packets
2	Switch/Bridge	Data Link	Link	Data Link	Ethernet, Token Ring	Frames
1	Hubs/Repeater	Physical		Physical		Bits

Figure 11. Typical TCP/IP Model (Jared Heinrich 2013)

For this work, protocols are viewed from the TCP/IP model perspective taking into consideration the Data Link, Internet, Transport, and Application layers.

3.3.1. The Application Layer

It provides certain functions directly to the user or to some other application programs (web browsers, communication programs and others), identifies communication partners, and ensures required communication interfaces e.g. Ethernet occur on the sender's machine, that is, synchronizes communication (Andrew Froehlich 2006, Teare Diane, 1999). It depends on all other layers to carry out its processes and has protocols that target valid communication over an IP network.

3.3.2. The Transport Layer

This layer manages and creates packets over the network. It accepts data from the session layer and segments the data for transport across the network (Teare Diane, 1999). There are two main protocols in this layer, which are the Transmission Control Protocol (TCP), and the User Datagram Protocol (UDP). TCP allows the recipient to receive packets in order, as they are sent by the web server while on the other hand, UDP does not check error in the packet transmission.

3.3.3. The Network Layer

This is a routing and network addressing layer, and it takes care of the IP addresses of the hosts and routing the data to another host(s) for network communication. Routers use this layer to decide how to forward packets and the layer also defines the network address, which is different from the MAC address, and the logical network layout (Teare Diane 1999).

3.3.4. The Data Link Layer

This layer carries the data in and out over the physical link in a network. The data link works with the non-routable addresses (MAC addresses). This layer is sub-divided into Logical Link Control (LLC) and the Media Access Control (MAC). In the data link layer, data bits are encoded, decoded and organised before being carried as frames between two neighbouring nodes on the same local area network or wide area network. Data link layer takes care of the compliance, that is, the user receives the data accurately.

3.4. The Cloud Security Challenges

Cloud computing is a crucial technological development that makes use of virtualization technology, which is as an upgraded version of hardware resources by means of creating more virtual machines running on the same physical machine. This technological development has come with security challenges due to its structure and design. The most vulnerable areas to attack in cloud computing are the virtual machines and the data itself, in which there are numerous forms of attacks among which the two most common ones are;

i. The Distributed Denial of Service (DDoS) and
ii. The service interruption due to security weaknesses.

DDoS is the situation whereby an attacker targets multiple numbers of computer systems, the server(s), website or any other network services to make online services unavailable by bombarding them with infectious traffic. There are different types of DDoS

attacks tools; the Complex (Agobot, Mstream and Trinoo) and Simple (X-DoS and H-DoS) (Sagar Aman et al 2013). The DDoS attacks could be in the form of extensible markup language (XML) supporting X-DoS and the Hypertext Transfer Protocol (HTTP) supporting H-DoS. In its first form, X-DoS attack opens the XML tags and consumes the system's CPU while the latter attack sends irregular HTTP requests to the target web server and consumes the communication links (Sagar Aman et al 2013).
Other types of attacks in cloud computing according to Oktay et al 2013 are;

i. Attacks on the organization by the current or previous employee(s) holding access privileges to the company information systems.
ii. Credential theft of an authorized user to gain access to the organization's systems, in order to access vital information.
iii. Port scanning in which the attacker looks for an open port and running services on an organization's network connections to capture IP and MAC addresses.
iv. A direct attack on virtual machines by means of using one virtual machine to attack another after one virtual machine's hypervisor and control has been compromised. This is made possible since many virtual machines use the same resource pool.
v. Deliberate alteration of data by an authorized personnel or system administrators for malicious purposes.

The resources available to the attacker determine the nature of possible attacks he can carry out, therefore typical cryptographic attacks are designed to undermine the security of cryptographic algorithms, and are used to decrypt data without the previous knowledge of the decryption (Manikandasaran et al 2016). Also, Manikandasara et al in their journal highlighted some other attacks in cloud computing, which are:

i. Ciphertext-Only Attack: It is presumed that the attacker has the knowledge of the encrypted message/data/text but not the plaintext. Therefore, the attacker depends on some guesses about the plaintext.
ii. Known-Plaintext Attack: This is a situation whereby the cryptanalyst or an attacker has knowledge of the encryption algorithm, plaintext

and the ciphertext. He is therefore obliged to find out the key(s) used to encrypt or decrypt the text.

iii. Chosen- Plaintext Attack: The cryptanalyst selects random plaintext to be encrypted and get the associated ciphertext. The analyst tries to get the secret key for the encryption or create an algorithm for decrypting the ciphertexts using the same key.

iv. Chosen-Ciphertext Attack: The cryptanalyst analyses specific ciphertexts together with the associated plaintexts. The aim is to get the secret key or enough information about the system under attack.

v. Meet-in-the-Middle Attack: It is among the types of the known plaintext attacks, in which the attacker knows some parts of the plaintexts and their corresponding ciphertexts. Here, the ciphertexts with many different secret keys for different encryptions of similar algorithms are easily broken. The scheme of meet-in-the-middle attack is described as shown in Figure 13 below; C = ciphertext, P = plain text, E= encryption algorithm, D = decryption algorithm, K_a and K_b = the two secret keys and $2^{len(ka)}$ - creating the table with all possible values of $E_a(K_a, P)$ (Chris Kowalczyk, 2017).

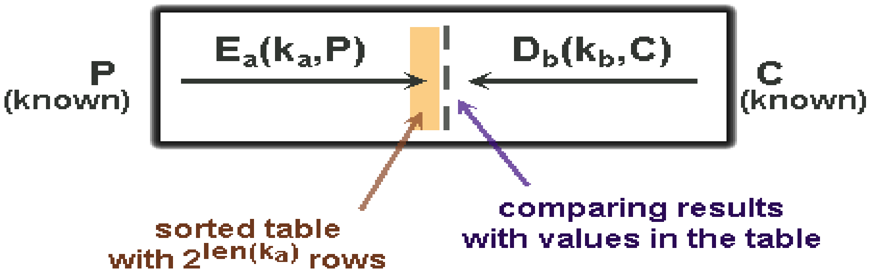


Figure 12. Typical meet-in-the-middle attack (Chris Kowalczyk, 2017)

The attack illustrated in Figure 12 can be carried out in 2 steps as described by Chris Kowalczyk.

Step 1:

i. Form a table with all possible values for the left side of the equation

ii. Compute all possible ciphertexts of the known plaintext formed using the first secret key $E_a(K_a, P)$.

iii. Arrange the final table according to the computed ciphertexts, just to enhance subsequent searching.

Step 2:

i. Compute values of $D_b(K_b, C)$ for the other hand of the equation

ii. Examine them with the values of the first side of the equation calculated earlier and saved in the table

iii. The attacker looks for a pair of secret keys K_a and K_b for which the value $E_a(K_a, P)$ found in the table and the just completed value of $D_b(K_b, C)$ are the same.

vi. Cloud Malware Injection Attack: The attacker creates own infectious service implementation module, SaaS, PaaS or a virtual machine instance IaaS in the form of a decoy, and then injects it into the Cloud system. Once the attack succeeds, it automatically redirects the requests of the valid user to the infectious service, and then the attacker's code begins to be executed. This attack is aimed at manipulating service-to-cloud attack area by either changing the data, causing a blockage or deletion, or disabling some of its functionalities.

vii. SQL Injection Attack: It is an attack on data-driven applications, where dangerous SQL queries are infused for execution in the entry field to gain access to information that was not meant to be shown.

3.5. The Research-based Security Solutions to Cloud Attacks

Many techniques are proposed to solve the different security attacks mentioned earlier as stated in the following list for instance. The different security techniques according to Bhalekar Seema B et al 2014 are:

i. Homomorphic token pre-combination: this is to obtain the integration of storage correctness insurance. Steps involved to attain the security of user’s data are; file distribution preparation, token pre-computation (to identify the precision of the data storage), challenge the token function, and error localisation (to verify the exactness of the data saved on the cloud server)

ii. Mobicloud: a virtual machine for a mobile cloud framework, created for a consumer, with total control of the information kept on its virtual hard drive. The mobile framework for trust management and private data isolation are:

   - Mobile cloud trust management: user identification management, that allows an individual to have total control of identities, meaning that the user has control over data shared over the Internet, as well capabilities for transfer or deletion when required.
   - Multitenancy: user’s data is saved in one sizable database and an uncommon encryption key is used to secure data for each user.

iii. Cryptographic technique: advanced cryptographic techniques such as ID-based public key cryptography (ID-PKC), Certificateless Public Key Cryptography (CL-PKC), key guarantee encryption, threshold decryption, and Information Dispersal Algorithm (IDA)-based scheme for high data distribution.

iv. Data integrity technique, that supplies key to data owner, and produces additional message verification code of the file provided by the data owner.

v. Privacy-Preserving Ciphertext-Policy Attribute Based-Encryption (PP-CP-ABE) scheme, and Attribute-Based Data Storage (ABDS) system. CP-ABE is used to expedite key management and cryptographic access control in an eloquent and effective way. The ABDS system scheme accomplishes expandable and smooth data access control, by means of using public cloud services. ABDS is applicable for mobile computing to ensure that communication and storage overhead are balanced. The technique benefits from the reduced data management operations' costs for both the mobile cloud nodes and storage service providers.

vi. Rivest Cipher 4 (RC4) algorithm, which renders secretiveness over the distinctive network nodes, and enhances the efficiency of processing cryptographic computational systems.

vii. Elliptic curve cryptographic technique, which is key management solution that has been achieved as part of the machine-to-machine local cloud action plan.

viii. The public key cryptographic technique, which offers protection against Internet-based attacks emanating from authentication; encryption, and decryption; and decryption verification among others.

ix. Hybrid textual authentication technique, and Computers Privacy and Data Protection (CPDP) are authentication solutions for users' security in any multi-cloud environment, and they protect against dictionary attacks. The CPDP is used for data security and integrity stored in the cloud.

x. Intrusion Detection System (IDS), and Cloud Intrusion Detection System Service technique are used to detect dangerous code, minimize resources usage on mobile appliances, and minimize software intricacy of mobile appliances. The techniques offer considerable security and flexibility during data exchange in the cloud.

In summary, Table 4 below shows the different security techniques and the different solutions they offer.

Table 4. Security Technique and their Solution (Bhalekar Seema B et al 2014).

Technique	Solution
CPDP Scheme	Provides security and integrity to data stored in the cloud
IDS and Collaborative Intrusion Detections (CIDSs) Scheme	Better detection of malicious code, reduced consumption and reduced software complexity of mobile devices
Advanced Cryptographic Technique	Provides data privacy protection or secure cloud storage
Asymmetric Key Approach	Confidentiality and integrity of the data
RC-4	Confidentiality over the different network
Message Digest 5 (MD5), Unicode Transformation Format 8 (UTF8)	Enhance the security
Pre-computed Token Scheme	Data storage correctness, and error localization

3.6. The Key Exchange Algorithm

The usage of cryptographic algorithm services, incorporates more than one entities, and requires the share of one another's public keys for authentication. This could be seen in the Web server authenticating its identity to the Web browsers. Key exchange is a class of protocols, that allows two entities to share a long-term secret or at least one public key in order to establish a short-term shared secret (Tim Rains 2015). The algorithm makes use of both symmetric and asymmetric protocols. Some examples of key exchange as being used in Windows 10 and Windows Server 2016 protocols are IPSec, Schannel, and Kerberos v5 (Corey Plett et al 2016).

The handshake protocol and the record layer are the two main sub-protocols in the TLS protocol (Nick Sullivan et al 2015). There are two main types of authentication and key establishment mechanisms in handshake protocol, which are RSA-based, and Diffie-Hellman-based (Nick Sullivan et al 2015). According to Nick Sullivan, an ephemeral

Diffie-Hellman handshake is an alternative form of the TLS handshake, and it uses two different mechanisms, one for establishing a shared pre-master secret, and the other one for authenticating the server. The key feature that this relies on, is the Diffie-Hellman key agreement algorithm.

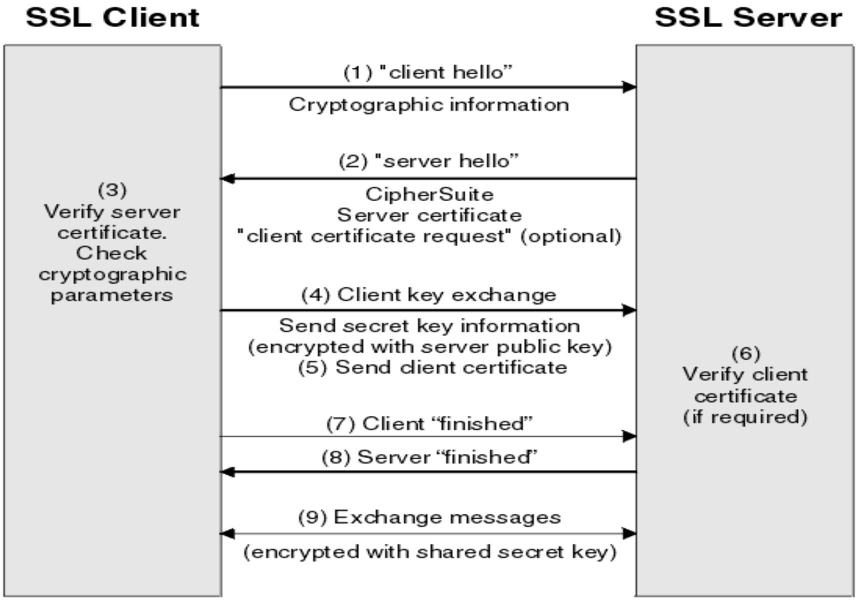


Figure 13: TLS Handshake Overview (IBM Corporation 1999, 2014)

The overview of the TLS handshake in Figure 13 shows the establishment of TLS client and server secret keys that they use for communication. The client TLS authenticates server certificate and checks the cryptographic parameters, while the server TLS verifies the client certificate if needed.

The exchange algorithms between the user (person A), the provider (person B) and the public key g, could work as follows:

i. person A has secret A, sends g^A to person B
ii. person B has secret B, sends g^B to person a
iii. person A computes $(g^B)^A$
iv. person B computes $(g^A)^B$
v. Both A and B end up with g^{AB}, which is their shared secret

The given algorithm is similar to the TLS handshake shown earlier in Figure 13. However, this does not work well with regular numbers because g^{AB} has the tendency to get large, but there are suitable ways to take the n-th root of a number. Nonetheless, there are techniques to overcome this shortage. This is done by using modular arithmetic, which is always dividing the result of the calculation by a large prime number and taking the remainder (Sullivan Nick 2014).

Furthermore, the Diffie Hellman, notably a public key algorithm, has the capability to generate such shared secret key. The asymmetric keys, which are randomly selected from a set of keys with certain properties, conform with the asymmetric algorithm in such a way that two users would probably not generate similar keys, with key sizes commonly ranging from 512 to 4096 bits.

4. THE SIMULATORS OVERVIEW

Simulators are commonly software-defined programmes, that mostly use discrete-event simulation to model the behaviour or performance of a system or a network at different scenarios, under changing state variables at specific cases. There are many types of simulators, that are designed for different purposes, and supported by various technologies. Some of these simulators use GUI, while some are command-line driven, in which, both are available either as open sources, or they require commercial licensing.

4.1. Network Simulator 3 (NS3)

NS3 is a non-commercial software developed in C++, and python scripting interface, to improve the simulation productivity. It is designed for simulating discrete-event network over the Internet, in a virtual environment mainly for companies contributing to the network improvement, academic research, and for personal use.

The NS3 development (ns-3-dev) is hosted in an open-source code management tool available on different platforms (Windows, Linux and others) servers. The most recent version of the software is the ns-3.28, which was released in March 2018. The building tool for NS3 is web application firewall (WAF*)*, which is a build automation tool designed to help in the compilation and installation of computer software, and is written in python scripting language (Nagy Thomas 2018).

4.1.1. NS3 Basic Architecture

The basic NS3 architecture as shown in Figure 14, shows all the common features needed across all classes, protocols, and hardware models. The NS3 network contains the packet, sockets, and the nodes object models. Other components, which are sub-classes of the main component, are the mobility models, high-level wrappers, smart pointers, call-backs and tracing among others. As stated in the documentation, tracing gives users the freedom to select the events to monitor, using complex logic to decide which information to log to the trace file or to carry out inline statistical calculations. Call-back offers dependencies for each module by ensuring pairing between different simulators with the help of the call-back API. The node class is designed as the base class and can also be instantiated.

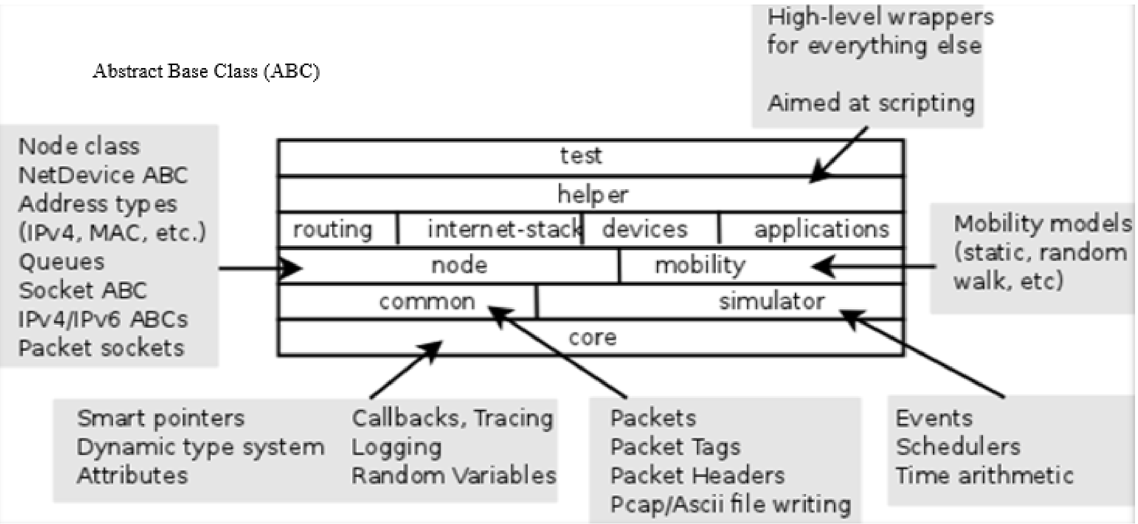


Figure 14. Typical NS3 Architecture (Waleed Ahmed 2015)

The NS3 is among the fastest simulators, cleanly designed, flexible, scalable, and with no graphical user interface to build network topology.

4.1.2. The Technologies

NS3 is built on C++ library to offer a streak of network simulation models. It is carried out as C++ objects, which are abstracted in python. Users are given the privilege to use or communicate with this library by coding in C++ or python to activate series of simulation models in order to simulate the desired situation as described in the documentation. The technologies, as shown in Figure 15, involved comprise of libraries such as Standard Template Library (STL), which is written in C++ and activates some parts of C++ standard library; and the C++ or python applications are the libraries that the users usually communicate with to create instances of various simulation models in order to carry out any simulation design of choice.

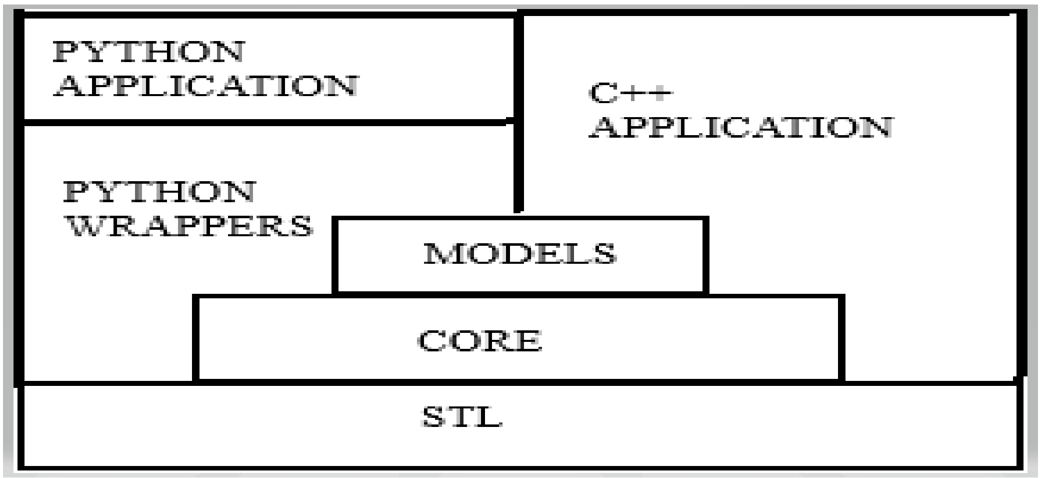


Figure 15. NS3 Key Technologies (NS3, 2017)

4.1.3. The TCP Models

The TCP models are explained in the NS3 documentation, where it was indicated that they are designed to suit the different TCP applications. The applications use specific common header classes from the source/network directory, which allows the users' codes to interchange the applications with the least change to the script. There are three available TCP applications, which are: an inherently harnessed TCP for NS3, the assist for the network simulation cradle (NSC), and the assist for direct code execution (DCE).

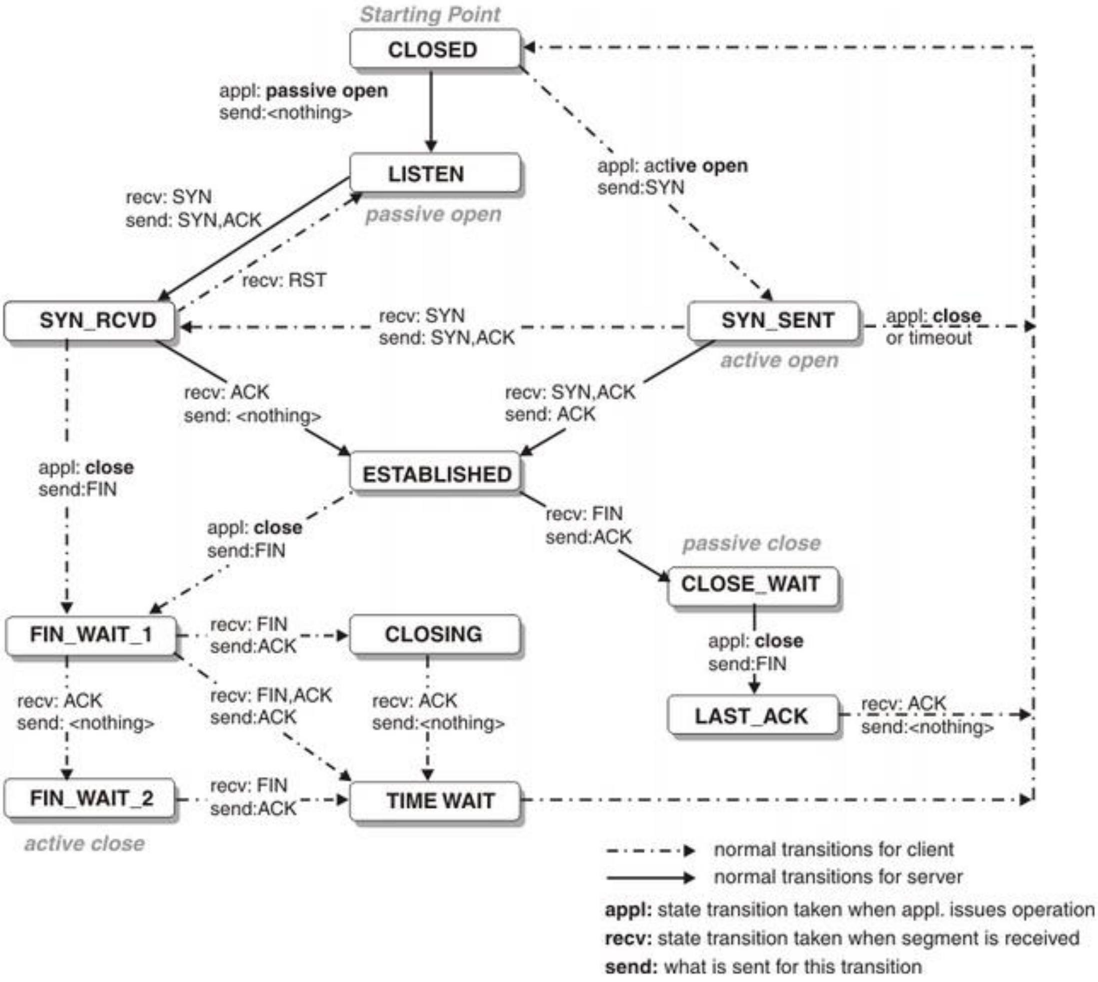


Figure 16. TCP State machine (IBM Corporation, 2013)

The TCP state machine as shown in Figure 16 above can be described as the life stages of a TCP connection, and the alteration of the states upon the acknowledgement of the different events from either the network or local TCP sockets implementation (IBM Corporation, 2013). The three types of messages that control transitions between TCP states are the synchronize (SYN), Finish (FIN) and Acknowledgment (ACK) messages. SYN synchronizes sequence numbers between devices, used to launch, and build a connection, while FIN shows that a device demands for connection termination, and ACK shows that a message is received (Kozierok Charles M. 2005).

Many different TCP state machine states exist, and they are as described as follows (Kozierok Charles M. 2005):

i. Closed: It is the main state where a connection begins ahead of the connection establishment.
ii. Listen: The server is waiting to receive SYN message from a client.
iii. Syn_rcvd: Await ACK to its SYN to complete the connection set up.
iv. Syn_sent: Client has sent SYN message and waiting for a similar message from the server.
v. Established: It is a steady state when data can be exchanged between both devices.
vi. Fin_wait_1: Device waiting for an ACK to a FIN it has sent.
vii. Fin_wait_2: Device received an ACK for its connection termination request and now waiting for a similar FIN from the other device.
viii. Close_wait: When a device received a FIN message from the other device.
ix. Last_ACK: Device already sent own FIN message and waiting for an ACK to its request. The device already received a close request.

It is not only limited to these given states, rather, these states incorporate more sub-events and transitions that occur between them.

The TCP sockets interaction with the application layer interface shows that the callback is triggered by the socket. The two functions of the socket are depicted by the sender and receiver which can be implemented at the same time. The TCP protocol uses congestion control to maximize the channel potentials for all nodes, connected to it by steadily increasing the rates, at which packets are being sent until congestion occurs in the network.

4.2. EstiNet

One of the most common network simulation and emulation tools is EstiNet. It is a software with a graphical user interface to simulate both wired and wireless network with real-life applications. And it in addition has some important attributes and advantages, such as the usage of the real-life protocol stacks, the capability to allow real-life Linux network applications, and the ability to communicate with the real-life network devices.

4.2.1. EstiNet Simulation Architecture

The EstiNet architecture as shown in Figure 17 below, is divided into three segments; namely the GUI, the Dispatcher and the Simulation Server. The graphical user interface client allows the user to draw, edit, run and playback simulation animation. The GUI offers the user the freedom to plot performance graphs and specifies parameters on the protocol module among many others. The job dispatcher is used for coordination, as it coordinates several simulation machines on the EstiNet. The dispatcher also picks the accessible machines to carry out the simulation of the network, and if none is accessible, it puts the simulation in the queue.

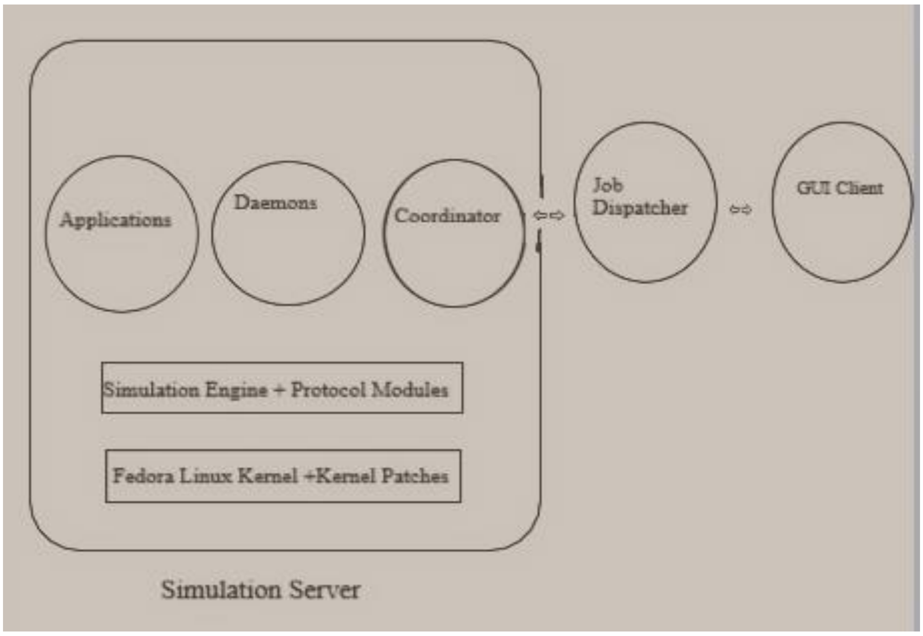


Figure 17. Typical EstiNet Architecture (Bernard Pasquier 2014)

Finally, the coordinator communicates directly with the job dispatcher, notifying the dispatcher which machine is in use or idle. The coordinator then receives tasks from the dispatcher and hands-over the specific protocol or network simulation to the simulation server.

For the Fedora operating system to work effectively with the software, the kernel is being modified by the inventive network simulator and emulator for wireless and mobile networks (NCTUns) so that the simulation server is able to work efficiently on it. This works in a manner that the simulation time is activated during simulation instead of the real-time timer.

4.2.2. EstiNet Flow Components

Open Flow (OF) Switch

In a software-defined network (SDN), this is responsible for the forwarding representation. There are three layers of abstraction, which are the specification, distribution and forwarding. The specification makes the application to show the exact network behaviour without implementation, while distribution protects the SDN applications from the distributed states and centralizing the distributed logical control. It is stated in the documentation, that the processing pipeline is responsible for packets flow. Finally, the forwarding permits forward behaviour by the network while hiding the hidden hardware specifications (Richard Ma 2015).

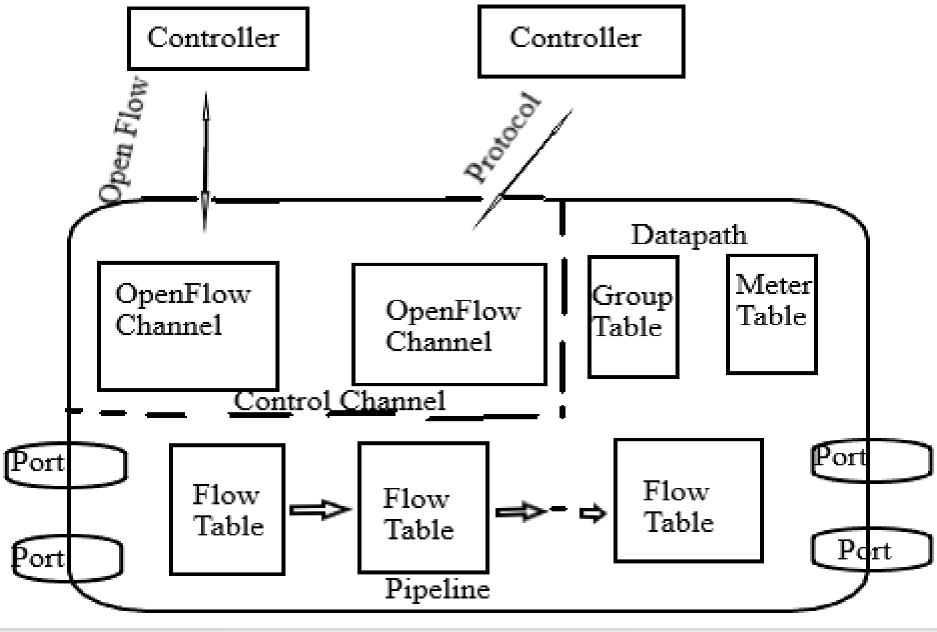


Figure 18. Typical OF Switch (Ma Richard 2015)

From the architecture shown in Figure 18 above, it could be seen that OF Switch has one group table, and one or more flow tables, that carry out packet forwarding and corrections, and a protected OpenFlow channels, that offer communication between the controller and the OpenFlow switches (Bhargavi Goswami et al 2017). Flow table is the basic data structure in an OpenFlow device, and each flow table consists of a group of flow entries. Flow tables altogether enable the appliance to check incoming packets according to a certain file, and carry out an appropriate action based on the contents of the packet information being received. Flow entries prioritize packets order by matching entries in each table that are being used. If a similar entry is found, the instructions related to the specific flow are carried out either to forward to a certain port; drop, flood to all ports; or move to the next flow table. If otherwise, the result depends on the configuration of the switch.

i. Operation Flow

The simulator operation flow has four different modes depicted as D, E, R and P in which D stands for the Draw topology mode, the E is the Edit property mode, R is the Run simulation and P for the Playback respectively. This is shown in Figure 19 below.

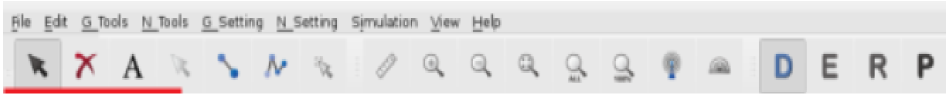


Figure 19. Typical EstiNet flow diagram

Drawing topology must be activated at first before the user could be allowed to have access to the nodes in favour of setting up the desired networks. The nodes take the form of point to point connections. Once the drawing is enabled, the edit property is activated, and the running program on the GUI will automatically create subnets under a fixed network. At that time, the nodes will be assigned IP and MAC addresses, and users will be allowed to add or modify the parameters on the nodes. The next mode in the list is the run simulation mode, which upon activation proceeds with running the configuration. It first generates a *.sim* directory, and then it exports the created configuration file describing the simulation scenario. The *.result* file is created automatically after the simulation is completed, which is the same as the topology file. The playback mode is activated when the simulation server sends back the simulation result files to the GUI program for storage in the *.result* directory.

4.3. Ettercap

Ettercap is a full access network sniffer package for the Man-in-the-Middle attacks (MitM) simulation. The package has distinctive attributes for detecting live connections, its contents, and other properties. Ettercap works with both active and passive analysis of many networks, and hosts' protocols. Active attacks are the type of attacks where attackers try to change the system resources or rather wipe out the information, e.g., ARP poisoning, MitM, DoS, or others. On the other hand, passive attacks are the monitoring attacks, where the attacker gains access to the information without altering them, e.g., Port scanning, idle scan.

Ettercap has two sniffing choices, the unified and bridge sniffing. In the former choice, the unified sniffing, all packets flowing through the wired or wireless link are detected, while the other packets not linked to the host on the ettercap will be directed automatically using network layer routing. This will allow someone to use MitM attack on different tools, and packets can be captured and modified by Ettercap. With the latter choice, bridge sniffing, upon activation, traffic is sent from one interface to the other during sniffing or filtering.

4.3.1. Address Resolution Protocol (ARP) Poisoning

Address Resolution Protocol (ARP) is a layer 3 protocol, which works by releasing ARP request packets. An ARP request taking the form of IP address is sent to all hosts in the network, to find out which host holds the given IP address. The host with that specific broadcast IP address replies with its MAC address. Every ARP packet broadcast is saved on the operating system's cache to reduce the number of broadcasts. The operating system regularly updates its cache when there is a new IP/MAC broadcast.

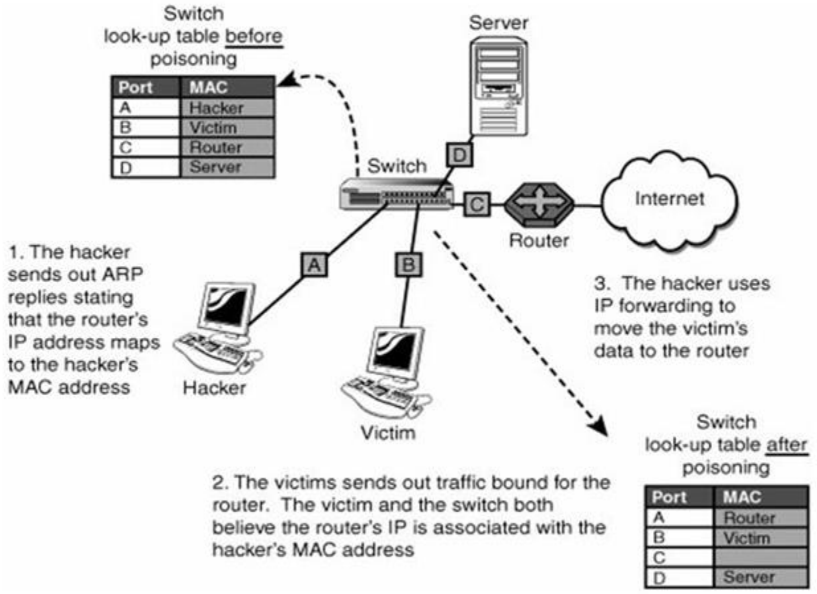

Switch
look-up table before
poisoning
Port | MAC
A | Hacker
B | Victim
C | Router
D | Server
Server
Switch
D
C
Router
Internet
A
B
1. The hacker sends out ARP replies stating that the router's IP address maps to the hacker's MAC address
Hacker
Victim
3. The hacker uses IP forwarding to move the victim's data to the router
Switch
look-up table after
poisoning
Port | MAC
A | Router
B | Victim
C |
D | Server
2. The victims sends out traffic bound for the router. The victim and the switch both believe the router's IP is associated with the hacker's MAC address

Figure 20. ARP Poisoning Process (Rickard Cayla 2015)

The MitM attacks happen when an attacker illegally positions own self between legally authorized users, so as to be able to be stealing or modifying information being exchanged either between or among them. The poisoning process in the MitM attack, as in Figure 20**,** occurs in a manner that, when ARP requests are sent to the target host to poison its ARP cache, it changes the Ethernet MAC address into the attacker's known MAC address to monitor the traffic. When the target has been poisoned, it will send all its packets to the attacker's device, which in turn forward the stolen information to the real destination and acts like a legitimate user. The attacker's machine thereby acts like a legitimate user during the communication process.

4.4. Damn Vulnerable Web Application (DVWA)

Another tool for the simulation of data security attacks is DVWA, and it is a PHP-MySQL damn susceptible web application and it is a free software available on GitHub (DVWA team). The application is an Internet-based tool, that is used for teaching, and to assist researchers understanding the techniques involved in securing web applications in a legitimate environment. It is also designed to practise other cloud attacks such as SQL injection, Brute force, and others in a virtual environment as it could be done in the real and legal environment.

4.4.1. SQL Injection Architecture

The browser is used as an interface to user input, and for the process of users' validation. The web application running on the web server authenticates inputs, handles the exception, allows the user into the system, and offers safe configuration. The architecture is as shown in Figure 21 below.

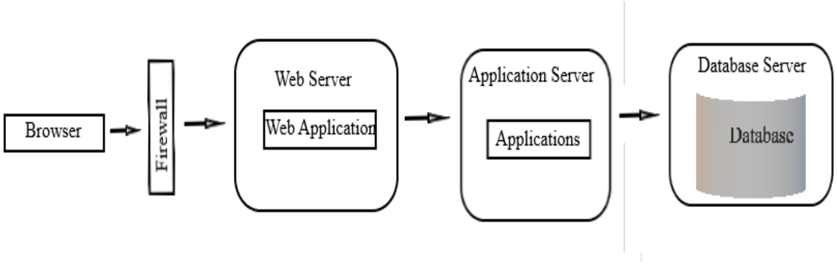


Figure 21. SQL Injection Architecture (Satyam Singh 2015)

In this architecture, the applications run on the application server for keeping logging activities, auditing, and validating crucial activities. This is followed by the database running on the database server which handles the security of the data by encrypting and hashing of crucial data (Satyam Singh 2015).

4.4.2. The procedure

There are two different conditions by which SQL injection attacks can occur, these are: through a relational database that uses SQL, and by controllable user input which is used exactly as in an SQL query (Govinda Raj 2017).

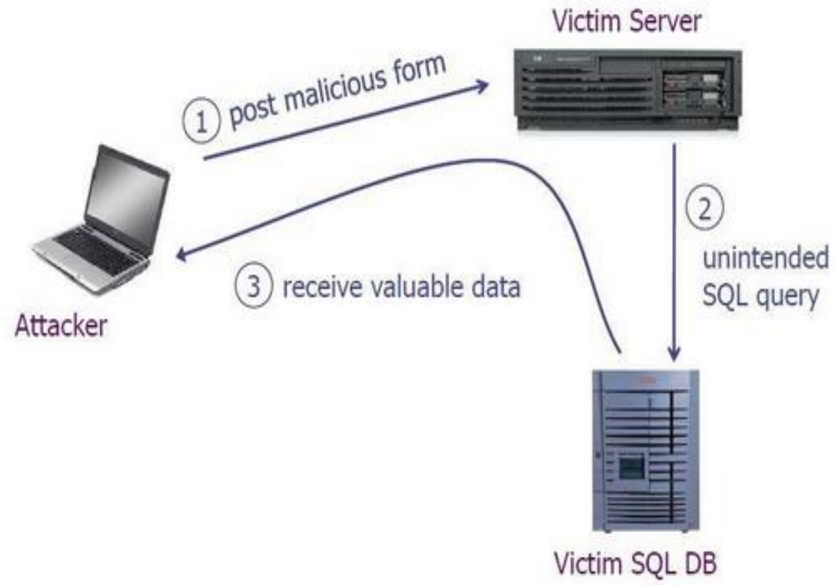


Figure 22. How SQL Injection works (Govinda Raj)

Attackers use SQL injection to manipulate data stored on a database or alter its integrity to render the data useless. The method could also be used to create vulnerability into the system, making deletion of data from the database possible. This SQL injection weakness could make all the data on the database server available to the attacker. The working steps, by which an attacker can send a query to the victim database in order to steal precious data, are as shown in Figure 22 above. From the figure, the attacker sends malicious data to the victim’s server to query his database; if the database is not well secured or configured, the attacker will be able to receive valuable information about the victim, but if otherwise, the database will not return anything to the attacker’s machine.

5. SIMULATION AND ANALYSIS

This chapter explains the experimental steps and analysis of the NS3, EstiNet, Damn Vulnerable Web Application, and Ettercap. The section studies the simulation performance metrics, systems and software used, the experimental model details, and the simulation performance results. This provides the evaluation metrics of the simulations of the network model on the network simulators.

5.1. DDoS- TCP Flooding using NS3

The system requirements for the NS3 simulation are as shown in the table below.

Table 5. NS3 System and Software requirements

System and Software	Details
Operating Systems	Linux Ubuntu 16.04 LTS, Intel Celeron Quad Core Processor N2940 @1.83GHz., 64-bit Operating System
NS3	ns-allinone-3.27

5.1.1. Simulation Topology

To achieve this, a network of twenty nodes having ten nodes at each end of a compromised link is implemented as shown in Figure 23 in order to analyse the TCP variant, measure nodes throughputs and the congestion size.

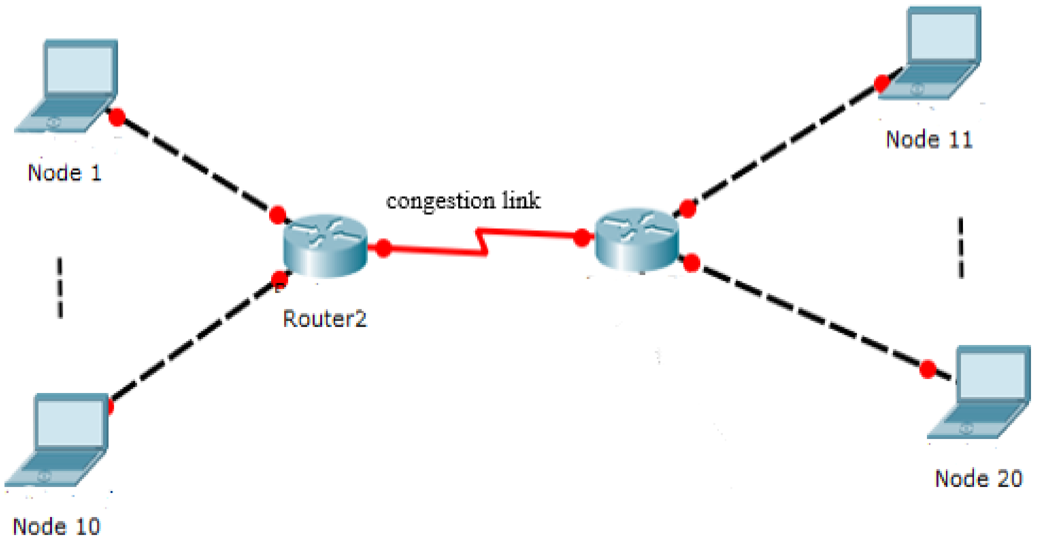


Figure 23. Point-to-point Network Topology

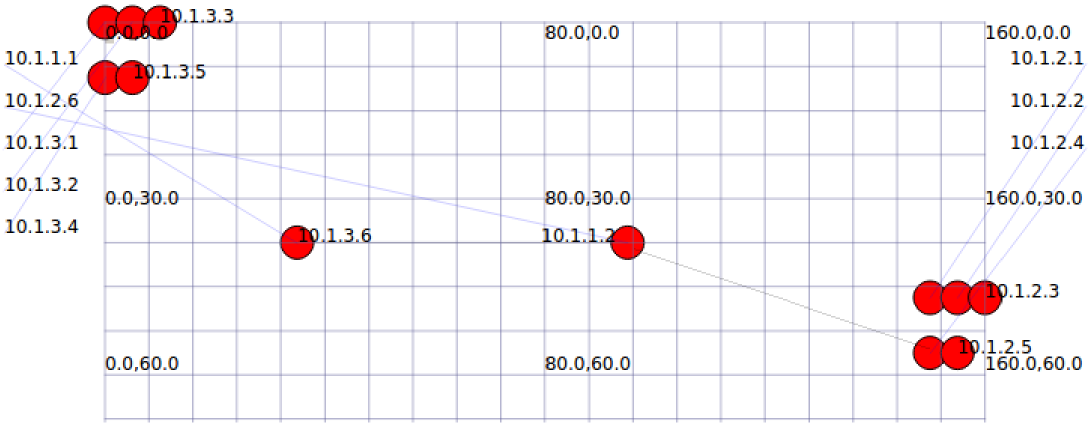


Figure 24. Simulation Topology on NetAnim

CALCULATION:

Data rate: 5 Mbps

Delay: 2ms

Throughput is calculated with this declaration;

*i->second.rxBytes * 8.0 / (i->second.timeLastRxPacket.GetSeconds() - i->second.timeFirstTxPacket.GetSeconds())/1024/1024*

The First time is calculated by;

i->second.timeFirstRxPacket.GetSeconds()

The last time is calculated using;

i->second.timeLastTxPacket.GetSeconds()

The received bytes;

i->second.rxBytes

Simulating the netAnim topology shown in Figure 24 above, which is an animated version of the topology drawn in Figure 23. NetAnim gives the following results, as shown in Figure 25 below which is a visualized animation similar to Figure 24 earlier in the simulation process, showing how packets are flowing in the network, in and out of one node to the other, and with which the performance metrics are measured.

: /home/ns3/ns-allinone-3.27/ns-3.27# ./waf --run "scratch/nbeck-wireless-tcp --nRightWifi = n≤10 –nLeftWifi=n≤10" --viz

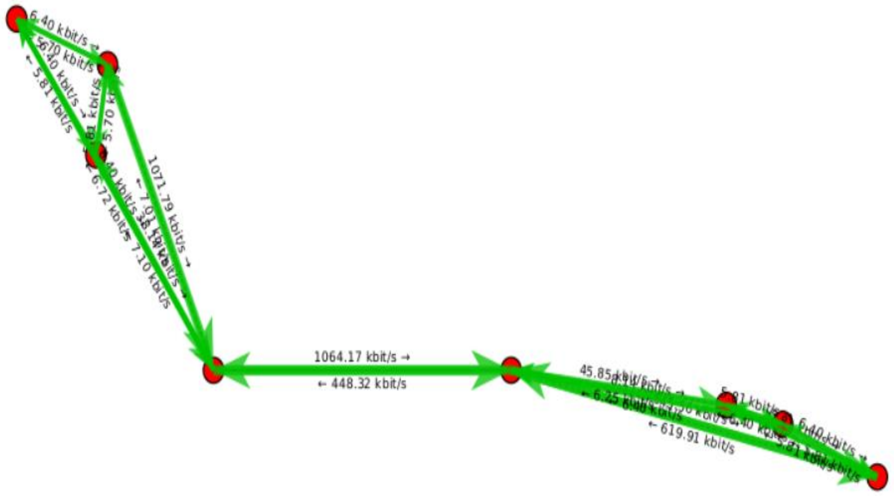


Figure 25. DDoS Simulation on Netanim

5.1.2. Performance Metrics

Node throughput is the number of packets received at the nodes and is measured in Mbps. The congestion control is the highest number of unauthorized packets that can be sent through a link, measured in bytes on Congestion Window (CWND), and its responsiveness is measured in milliseconds (ms).

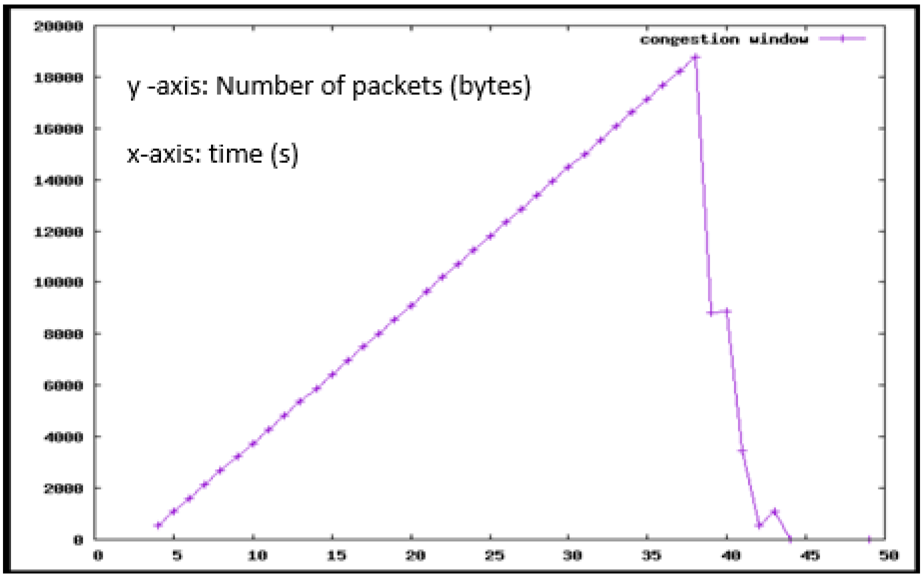


Figure 26. DDos Link Congestion with 10 nodes

The simulation was firstly carried out with ten nodes to see the effect of the link bottleneck on the nodes as plotted in the graph shown in Figure 26 above. It is observed that the link congestion caused a sharp drop of the packets after approximately 38s, and steadily dropped from 8000 bytes, 100 bytes and 300 bytes at 40s, 42s and 43s respectively. This explains the effect of the TCP variant on the link causing the bottleneck, thereby resulting in packet drop and subsequently affecting the quality of service.

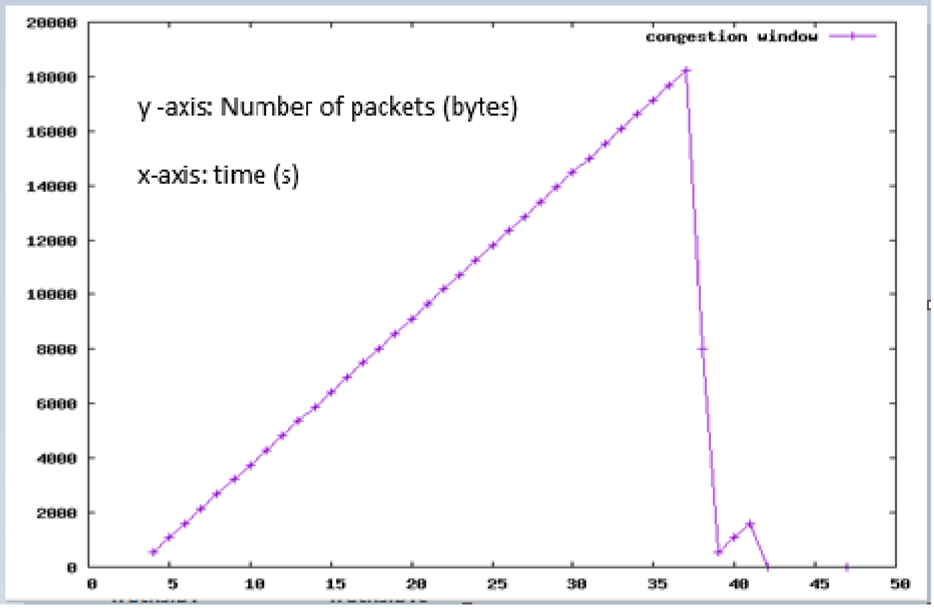


Figure 27. DDos Link Congestion at 16 nodes

From Figure 27**,** the number of nodes was increased to 16 to see their effect. Here, the simulation starts at around 4s. It is noticed that the packets dropped from 18000 bytes to less than 2000 bytes at around 38s and there was a little fluctuation, between 39s and 40s, where the packets sharply increased to 2000bytes but later dropped to 0 bytes at 41s. The drop at the congestion was because of the TCP variant due to the bottlenecks in the link.

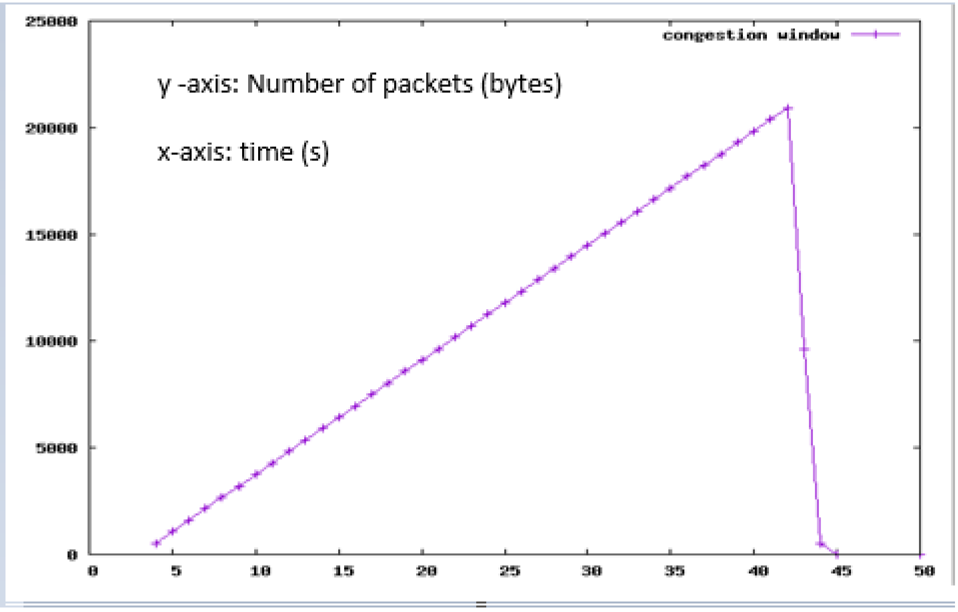
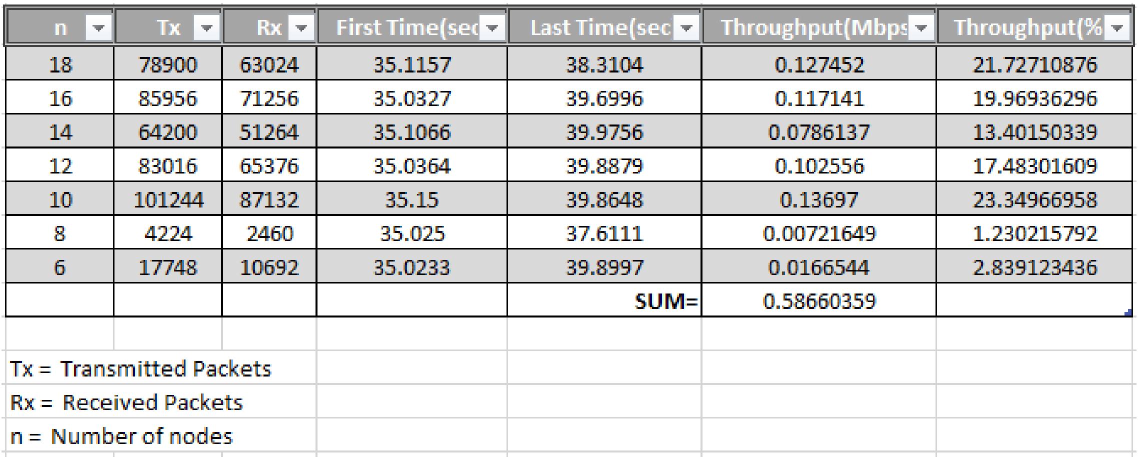


Figure 28. DDos Link Congestion at 18 nodes

In Figure 28 earlier, the number of nodes was increased to 18. It is noticed that there was a sharp change in the congestion, the packets dropped from around 21000bytes to 100 bytes at 44 seconds and later to 0 byte at 45s. It is observed that the more the nodes, the more the time it takes for the congestion to occur.

Table 6 below shows the network throughput of the nodes with their corresponding received and transmitted times. This gives a clearer picture of what is happening in the network per time.

Table 6. Network Throughputs data



n	Tx	Rx	First Time(sec	Last Time(sec	Throughput(Mbps	Throughput(%
18	78900	63024	35.1157	38.3104	0.127452	21.72710876
16	85956	71256	35.0327	39.6996	0.117141	19.96936296
14	64200	51264	35.1066	39.9756	0.0786137	13.40150339
12	83016	65376	35.0364	39.8879	0.102556	17.48301609
10	101244	87132	35.15	39.8648	0.13697	23.34966958
8	4224	2460	35.025	37.6111	0.00721649	1.230215792
6	17748	10692	35.0233	39.8997	0.0166544	2.839123436
				SUM=	0.58660359	

Tx = Transmitted Packets
Rx = Received Packets
n = Number of nodes

It can also be deduced from the table, that the link congestion increases as the number of the nodes increases. There was abnormal behaviour when the number of nodes was 12, 10 and 6; however, it is observed that the throughputs do not follow the sequence. This behaviour is due to the system error or fluctuation in the simulation.

Figure 29 shows the graphical representation of the increase in the network throughput and the number of nodes. Though the original data rate of 5Mbps was truncated to the average of around 0.1Mbps.

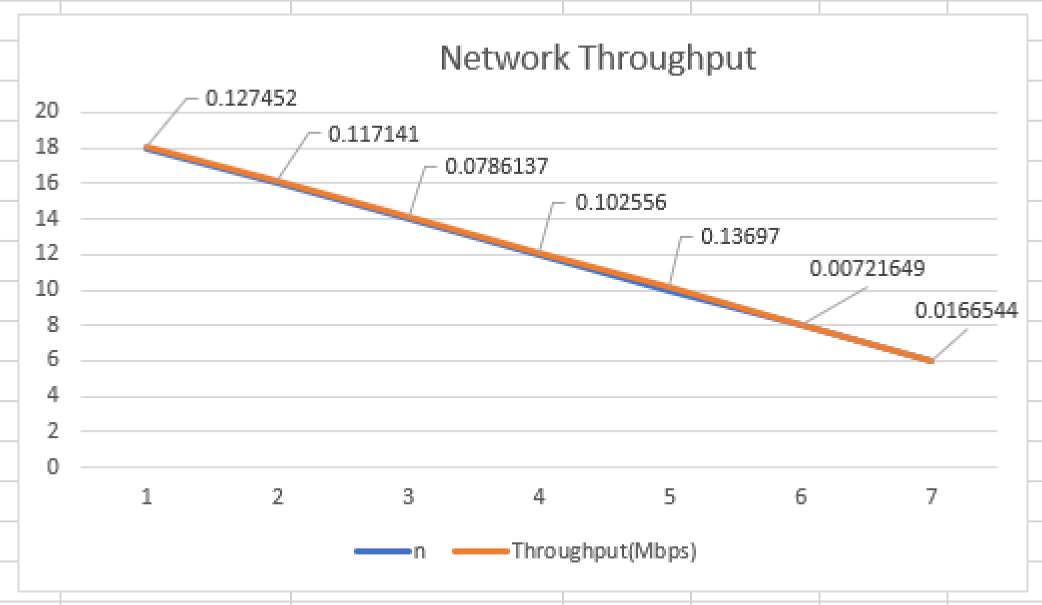


Figure 29. Network Throughput on graph

5.2. DDoS on EstiNet

The system and software recommendation for the simulation of the distributed denial of service using EstiNet is given in Table 7 below.

Table 7. EstiNet System and Software requirements

System and Software	Details
Operating Systems	Fedora 20 Intel Celeron Quad Core Processor N2940 @1.83GHz. 64-bit Operating System
EstiNet	EstiNet 9.0

5.2.1. Set up and Metrics

This section discusses the steps required to set up the EstiNet working environment and the topology for the simulation. The considered metrics, the nodes output/ throughput is measured in kilobyte per seconds (kBps)

The DDoS Botnet Simulator tool, Bonesi-0.3, is used to simulate Botnet Traffic to learn the effect of DDoS attacks on Internet networks in a testbed environment. It induces ICMP, TCP (HTTP), and UDP flow attacks from the user-defined botnet size (different IP addresses). Other parameters such as data volume, source IP address, URLs and others can be also configured. The tool is so robust that it has the capability of generating up to 50,000 IP addresses randomly.

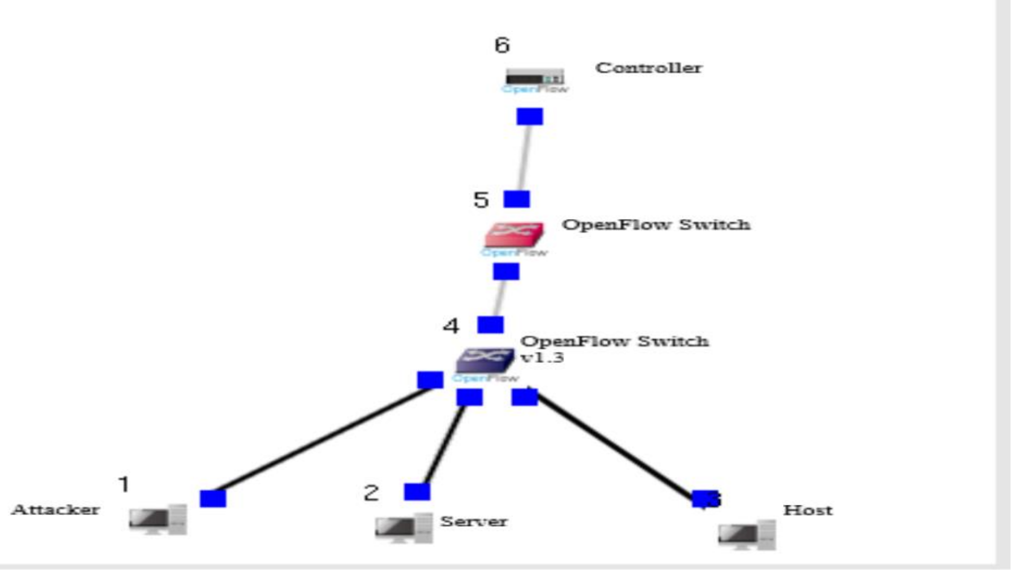
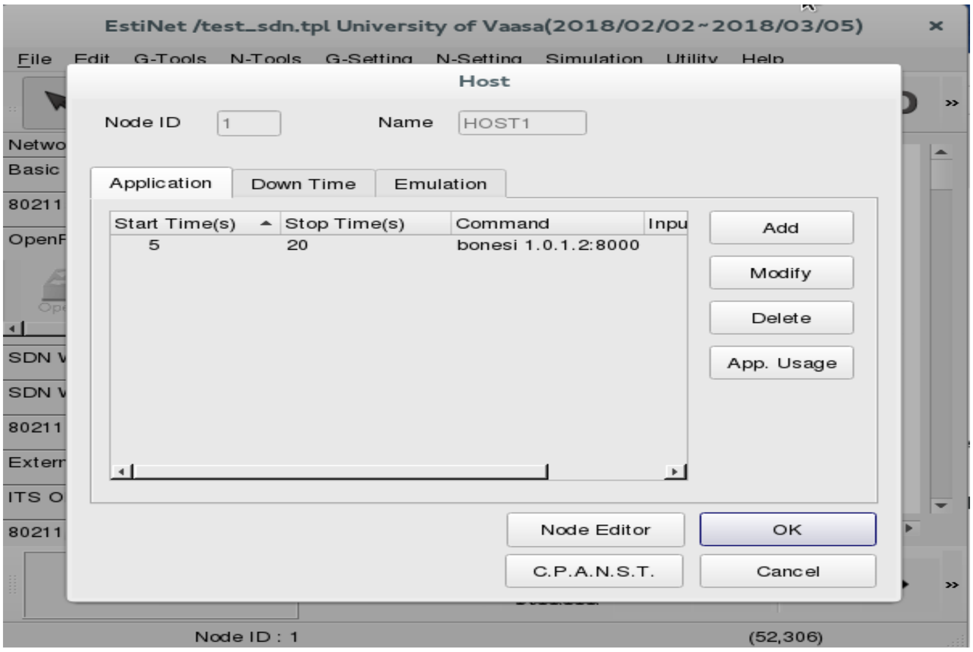


Figure 30. DDoS Topology on EstiNet

The network topology in Figure 30 is implemented to carry out the simulation. On the attacker’s workstation, an ICMP, TCP and UDP flow attack is defined at port 8000. This is according to the user-defined botnet size and is as shown in Figure 31.



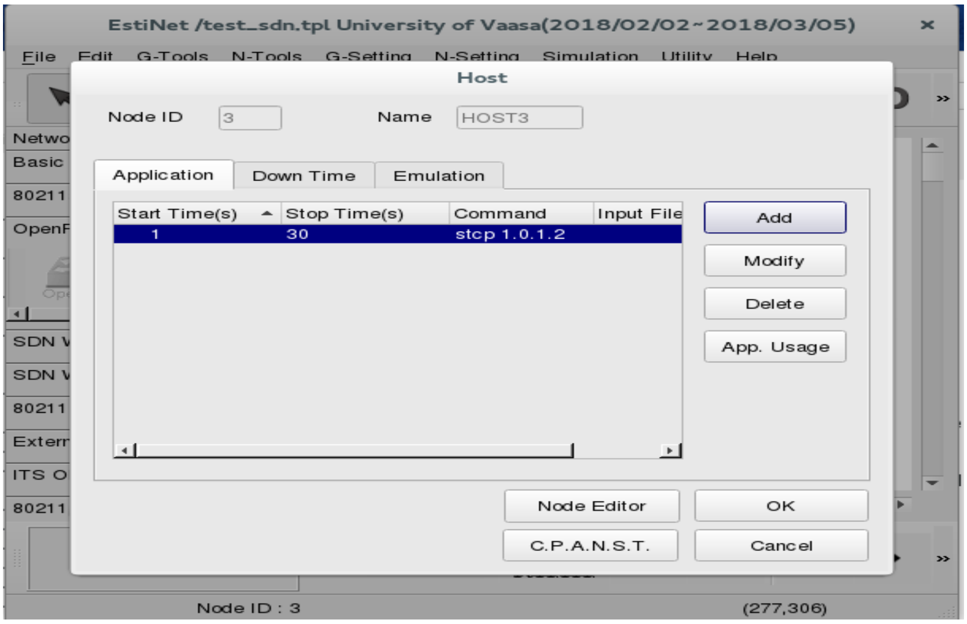
Node ID 1
Name HOST1

Start Time(s)	Stop Time(s)	Command	Inpu
5	20	bonesi 1.0.1.2:8000	

Node ID : 1 (52,306)

Figure 31. Attack configuration

The host’s workstation configuration is as shown in Figure 32 below. As seen, the host runs the Sensor Transmission Protocol (STCP), which is a transport layer protocol, authorized by the TCP, where most of its functions are applied at the base station. This will run every 30 seconds.



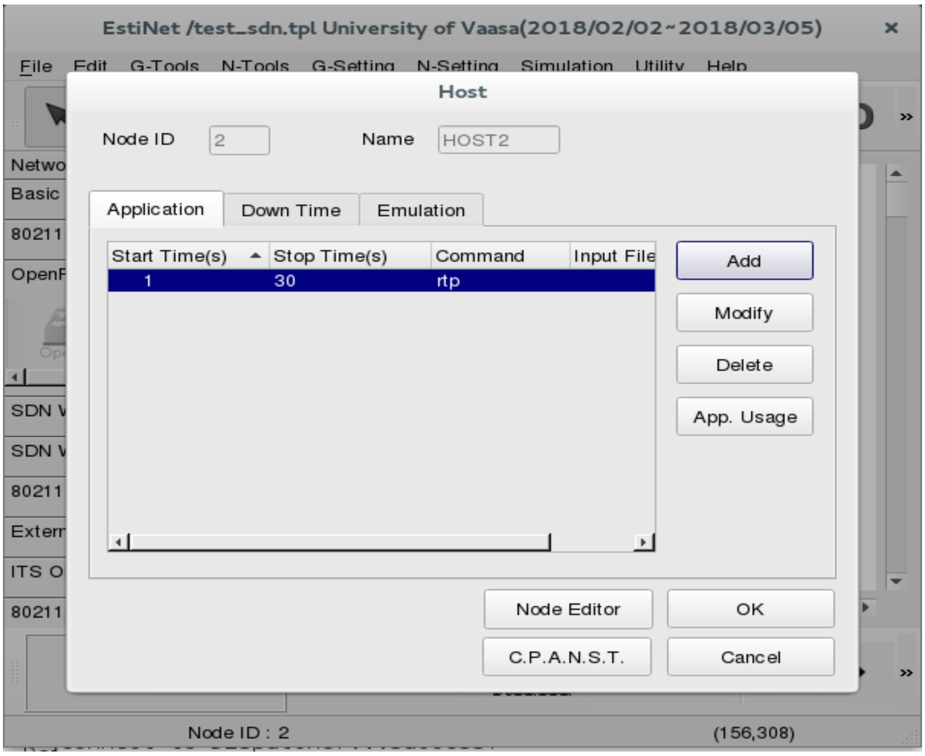
Node ID 3
Name HOST3

Start Time(s)	Stop Time(s)	Command	Input File
1	30	stcp 1.0.1.2	

Node ID : 3 (277,306)

Figure 32. Host configuration

The server workstation is configured to run the Real-Time Transport Protocol (RTP) for 30sec as shown in Figure 33. The RTP is responsible for offering point-to-point real-time data transmission (Schulzrinne et al 2003: p19).

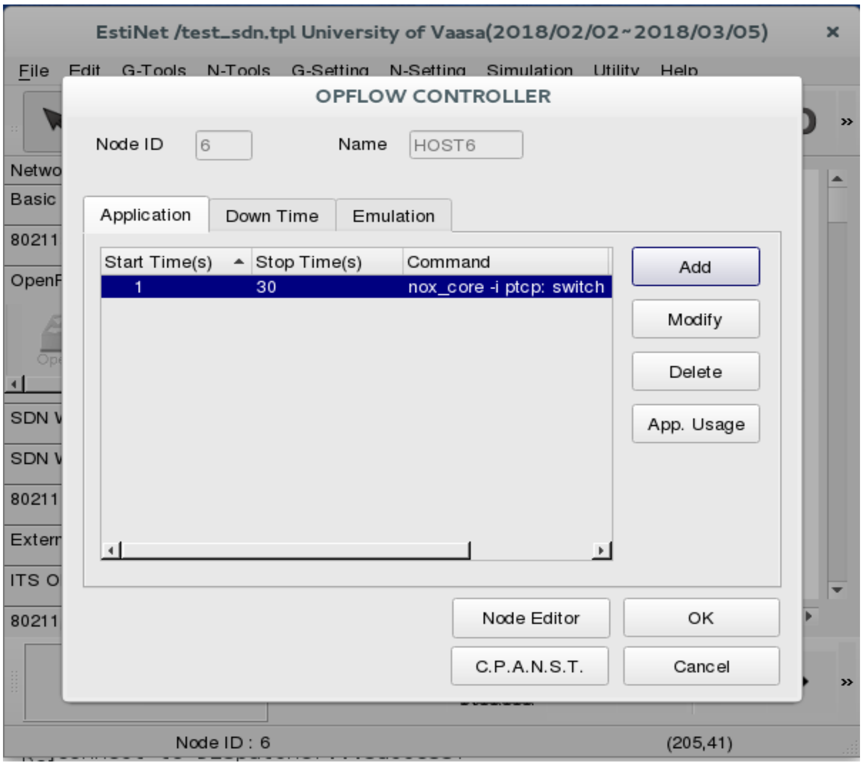


Node ID 2 Name HOST2

Start Time(s)	Stop Time(s)	Command	Input File
1	30	rtp	

Figure 33. Server configuration

The OpenFlow controller runs the Parallel TCP (PTCP) protocol. It is configured as shown in Figure 34 to run nox_core -I ptcp: switch. The PTCP is a point-to-point transport layer protocol, which enables a link to bask in the cumulative bandwidth resulted by different network paths, regardless of the attribute of the paths (Hsieh Hung-Yun et al 2002).


EstiNet /test_sdn.tpl University of Vaasa(2018/02/02~2018/03/05)
File Edit G-Tools N-Tools G-Setting N-Setting Simulation Utility Help
OPFLOW CONTROLLER
Application
Down Time
Emulation
Add
Modify
Delete
App. Usage
Node Editor
OK
C.P.A.N.S.T.
Cancel

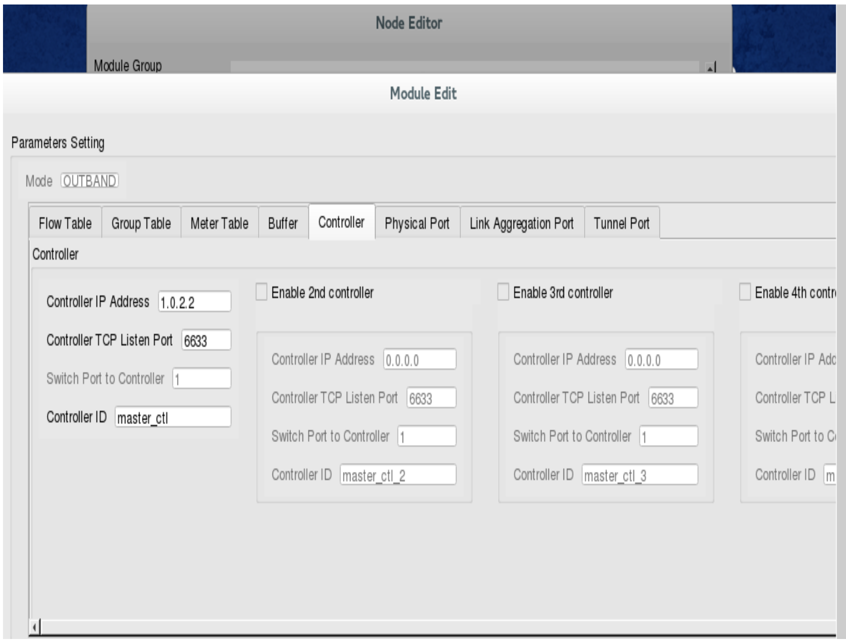
Node ID 6 Name HOST6

Start Time(s)	Stop Time(s)	Command
1	30	nox_core -i ptcp: switch

Node ID : 6 (205,41)

Figure 34. OpenFlow Controller configuration

On the Open Flow Switch, the network IP is modified as in Figure 35. This will allow the Open Flow controller and host workstation to be on the same network.



Mode OUTBAND

Controller IP Address 1.0.2.2
Controller TCP Listen Port 6633
Switch Port to Controller 1
Controller ID master_ctl

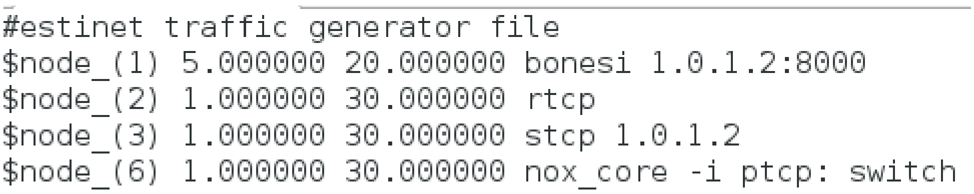
□ Enable 2nd controller
Controller IP Address 0.0.0.0
Controller TCP Listen Port 6633
Switch Port to Controller 1
Controller ID master_ctl_2

□ Enable 3rd controller
Controller IP Address 0.0.0.0
Controller TCP Listen Port 6633
Switch Port to Controller 1
Controller ID master_ctl_3

□ Enable 4th contr
Controller IP Add
Controller TCP L
Switch Port to C
Controller ID m

Figure 35. OF Switch configuration

To import the simulation data to the graph, MAC module is activated in the node editor, then log packet statistics are enabled. This will allow the user to select any input or output of interest. The generated transport traffic generated by bonesi at the configured nodes 1,2,3 and 6 is as shown in Figure 36 below. The figure also shows the type of protocol running on each node.



```
#estinet traffic generator file
$node_(1) 5.000000 20.000000 bonesi 1.0.1.2:8000
$node_(2) 1.000000 30.000000 rtcp
$node_(3) 1.000000 30.000000 stcp 1.0.1.2
$node_(6) 1.000000 30.000000 nox_core -i ptcp: switch
```

Figure 36. Network traffic generator

5.2.2. EstiNet Results

The server input throughput is shown in Figure 37 below. The input throughput of around 1.2MBps

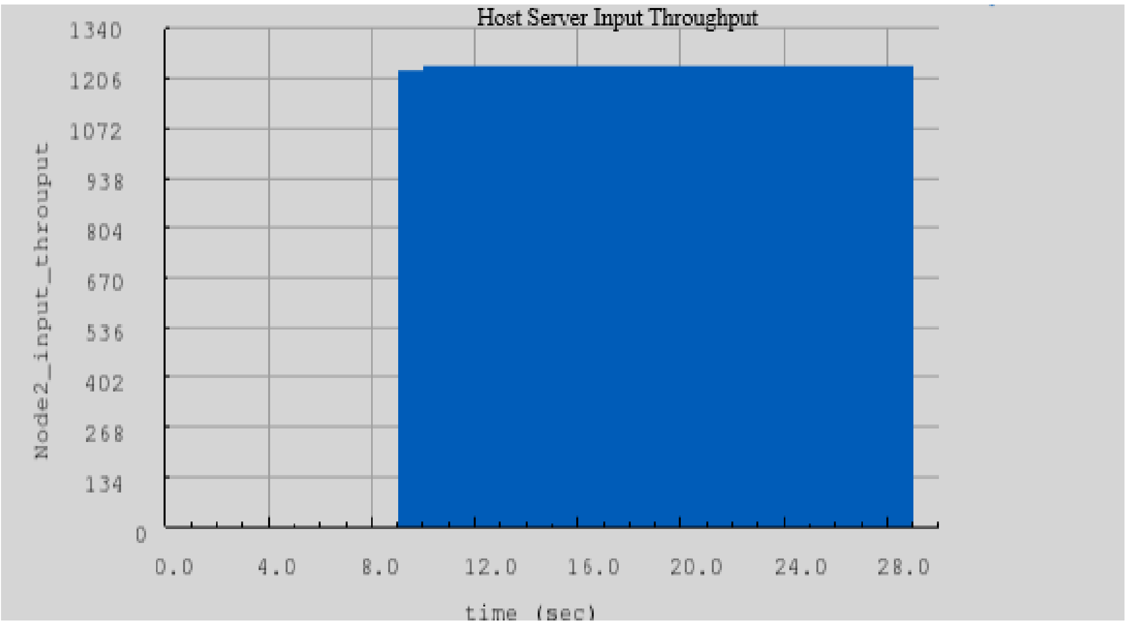


Figure 37. Host Server Input Throughput

The network throughput at the server is as shown in Figure 38. The figure shows the network after being compromised by TCP flood attacks. As could be seen, the network output dropped to approximately 35kBps compared to 1.2MBps as in the previous figure. This shows a great drop in the network packet and might result in network instability.

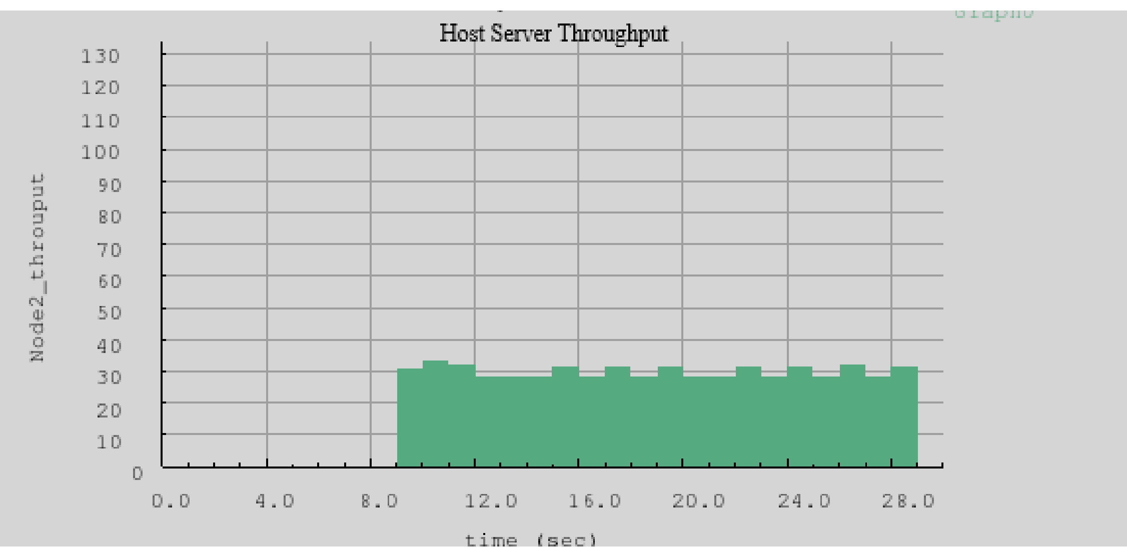


Figure 38. Host Server Throughput

The input traffic at both stations, the host and the server, is the same at 1.2MBps. This is displayed in Figure 39 below.

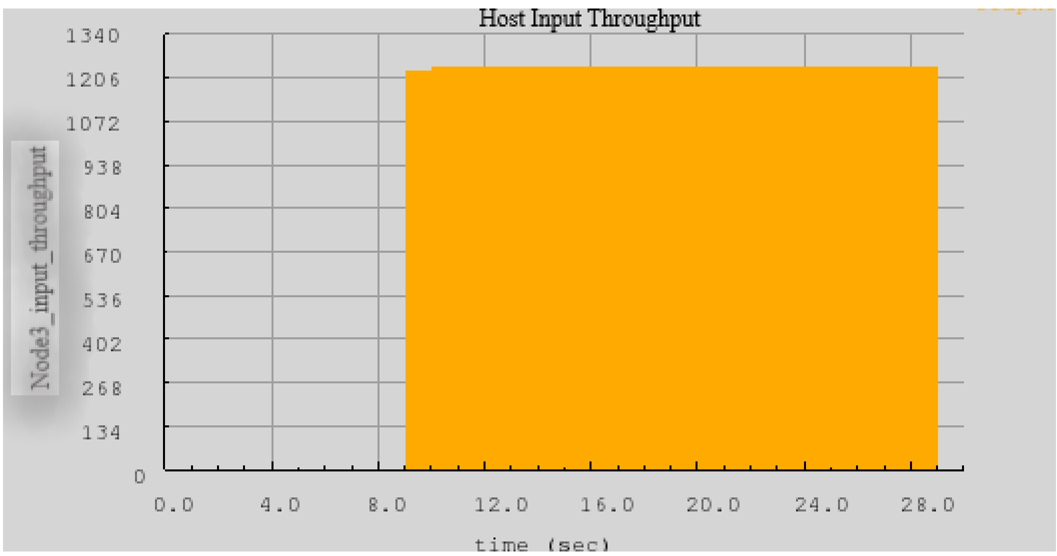


Figure 39. Host Input Throughput

The network output at the host is also dropped to approximately 35kBps. This tremendous drop in the network efficiency is shown in the node 3 throughput graph as in Figure 40 below.

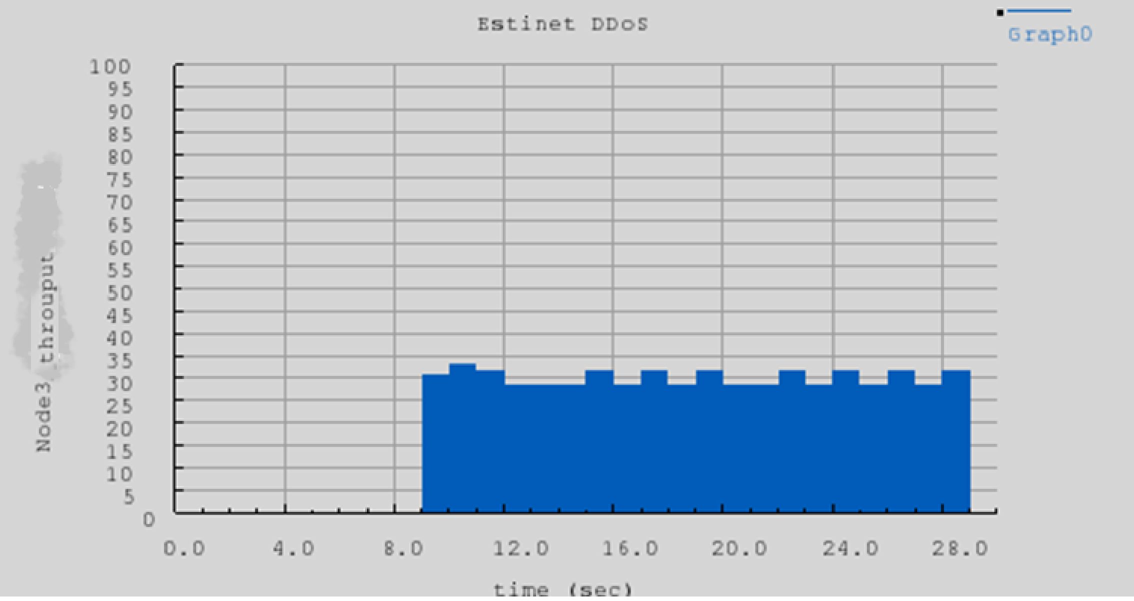


Figure 40. Host Throughput

Comparing the output at both stations, the host and the server, it is observed that the input throughput at both workstations has the same value of 1.2MBps and network output throughput of 35kBps as shown in Figure 41 below.

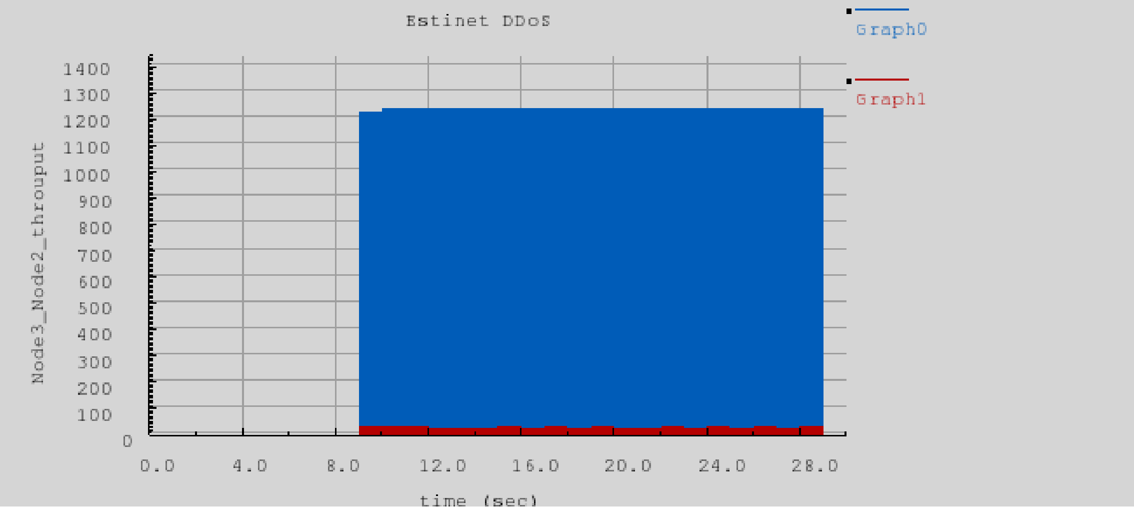


Figure 41. Host and Server Throughput

5.3. Damn Vulnerable Web Application for SQL Injection

There are four different levels of security on DVWA, namely low, medium, high, and impossible. These levels correspond to how easier it is to gain access to the information stored on the DVWA database. The testing database is designed with five default users, and queries are injected into the database to steal their usernames, passwords, as well as other vital information. Table 8 below shows the installation system and software requirements for DVWA.

Table 8. DVWA System and Software requirements

System and Software	Details
Operating Systems	Linux Ubuntu 16.04 LTS Intel Celeron Quad Core Processor N2940 @1.83GHz. 64-bit Operating System
Damn Vulnerability Web Application	DVWA

5.3.1. Results

After the installation, the four different levels, the low, medium, high, and the impossible can be tested. These test results shown in Figures 43, 44 and 45, were carried out at the low-level security for quick understanding of the concept. The other levels, medium and high, can be carried out by making few changes to the DVWA source codes accordingly. For example, to carry out any test with the medium level SQL Injection, the following steps as stated in the Oracle documentation should be taken:

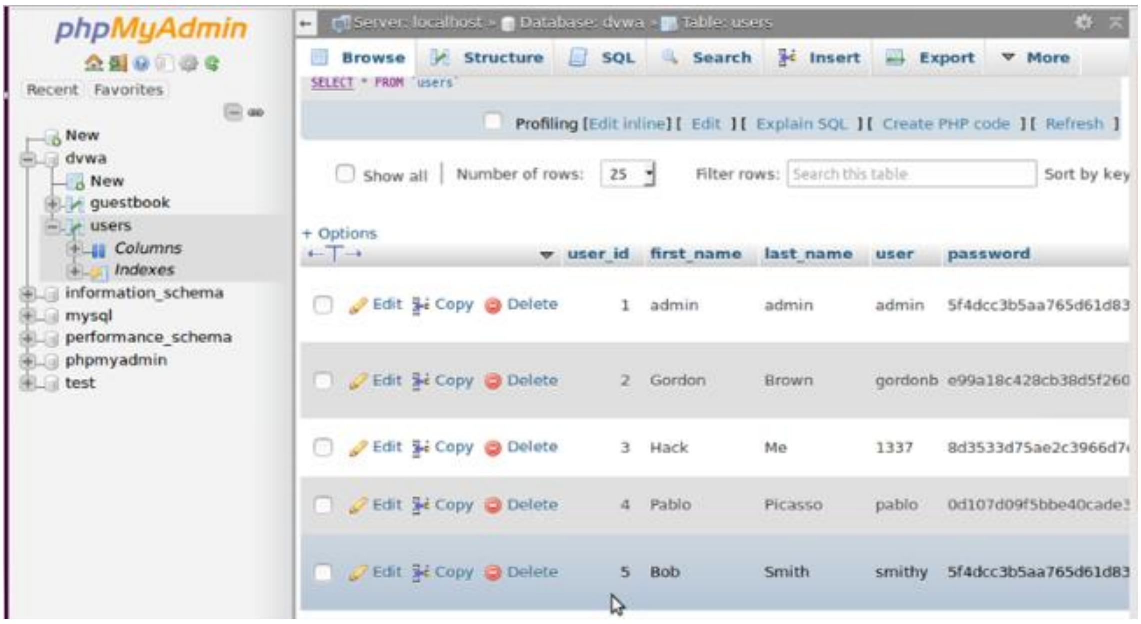
i. inside the source code, allow *mysql_real_escape_string()* to call mysql, which sorts `\’ to these characters, \x00,\n,\r,\,´, AND \x1a, and

ii. use post request and binding to the specific input options.

iii. then release the database.

For high-level security, Java assistance script query is used, which will add limit to output results, while the impossible level security level is such a situation, that the SQL Injection is not possible to be executed. This is because the database structure is well formed and protected.

The vulnerability of the database is tested by issuing the query, 1’. If the database is vulnerable, it will return a value, the user whose ID is equal to 1, but the query will return error or null if it is not vulnerable. Figure 42 below shows model database to be exploited.

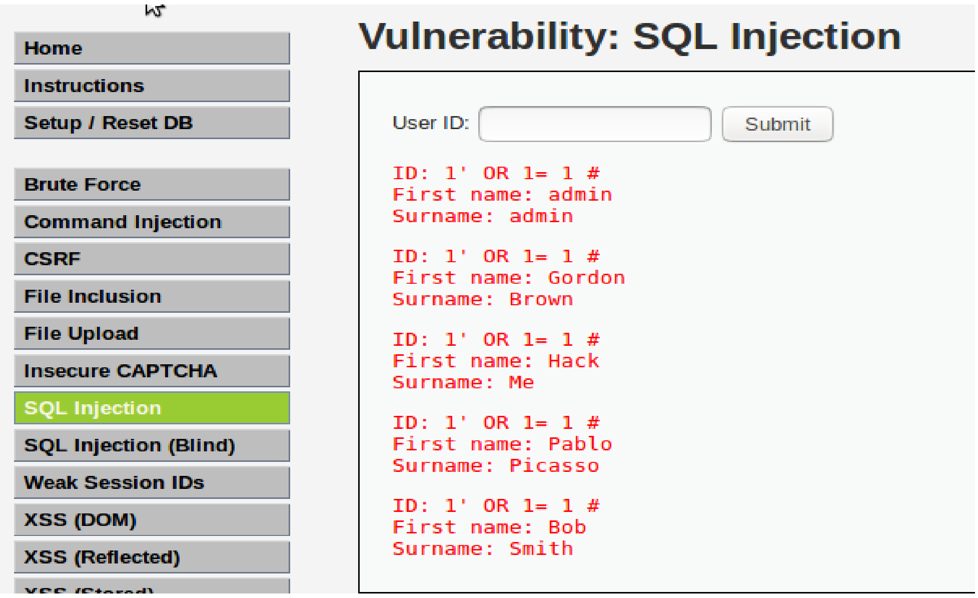


```
SELECT * FROM `users`
```

user_id	first_name	last_name	user	password
1	admin	admin	admin	5f4dcc3b5aa765d61d83
2	Gordon	Brown	gordonb	e99a18c428cb38d5f260
3	Hack	Me	1337	8d3533d75ae2c3966d7
4	Pablo	Picasso	pablo	0d107d09f5bbe40cade
5	Bob	Smith	smithy	5f4dcc3b5aa765d61d83

Figure 42. Database Interface

To exploit these vulnerabilities, the following tasks were performed. Firstly, get the list of all users on the database using this command; *1’ OR 1=1 #*



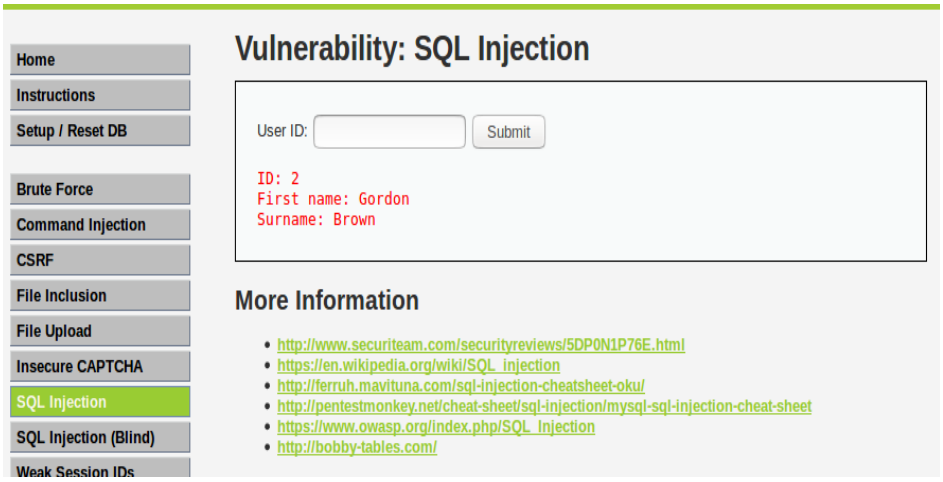
Vulnerability: SQL Injection

User ID:

```
ID: 1' OR 1= 1 #
First name: admin
Surname: admin

ID: 1' OR 1= 1 #
First name: Gordon
Surname: Brown

ID: 1' OR 1= 1 #
First name: Hack
Surname: Me

ID: 1' OR 1= 1 #
First name: Pablo
Surname: Picasso

ID: 1' OR 1= 1 #
First name: Bob
Surname: Smith
```

Figure 43. List of Users on the database

The user with ID = 2 is Gordon Brown



Vulnerability: SQL Injection

User ID:

```
ID: 2
First name: Gordon
Surname: Brown
```

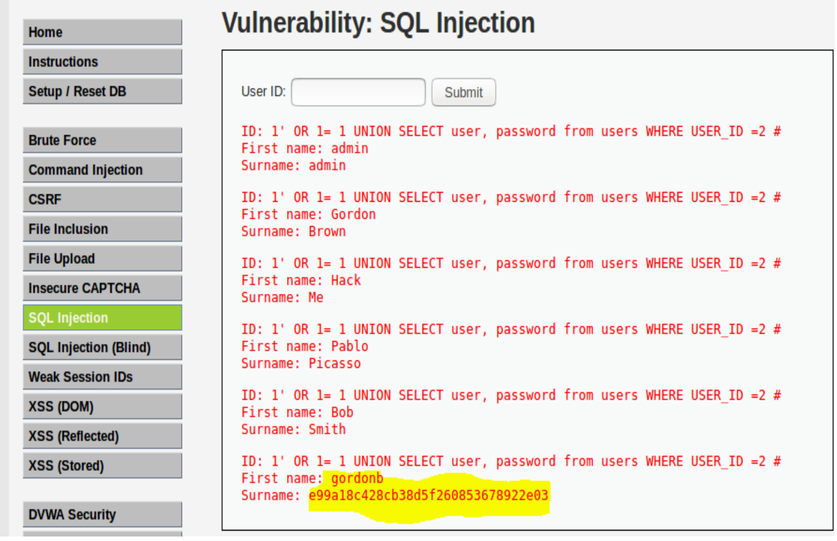
More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

Figure 44. User with ID = 2

To retrieve Gordon’s username and password,

1’ OR 1=1 UNION SELECT username, password FROM users WHERE user_ID=2 #



Vulnerability: SQL Injection

```
ID: 1' OR 1= 1 UNION SELECT user, password from users WHERE USER_ID =2 #
First name: admin
Surname: admin

ID: 1' OR 1= 1 UNION SELECT user, password from users WHERE USER_ID =2 #
First name: Gordon
Surname: Brown

ID: 1' OR 1= 1 UNION SELECT user, password from users WHERE USER_ID =2 #
First name: Hack
Surname: Me

ID: 1' OR 1= 1 UNION SELECT user, password from users WHERE USER_ID =2 #
First name: Pablo
Surname: Picasso

ID: 1' OR 1= 1 UNION SELECT user, password from users WHERE USER_ID =2 #
First name: Bob
Surname: Smith

ID: 1' OR 1= 1 UNION SELECT user, password from users WHERE USER_ID =2 #
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03
```

Figure 45. Username and password for use with ID=2

This encrypted password in MD5 hash could be broken using Dan’s tools for the MD5 hash generator.

5.4. Ettercap for MitM Attack

The system requirements and the network sniffing tool used for this simulation are as displayed in Table 9.

Table 9. Ettercap System and Software requirements

System and Software	Details
Operating Systems	Linux Ubuntu 16.04 LTS Intel Celeron Quad Core Processor N2940 @1.83GHz. 64-bit Operating System
Ettercap	ettercap 0.8.2

5.4.1. Topology

The network topology implemented for the simulation is shown in Figure 46. As seen, the network consists of the attacker's machine, on which Ettercap is running, and the victim machines. The attacker's machine, at the middle of the connection, will be stealing information being shared between the communicating Host1 and Host2.

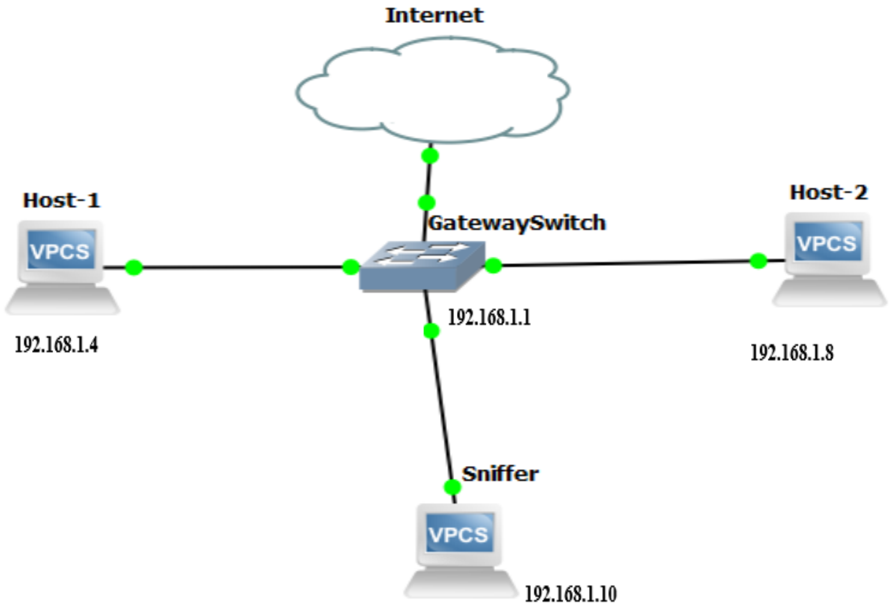


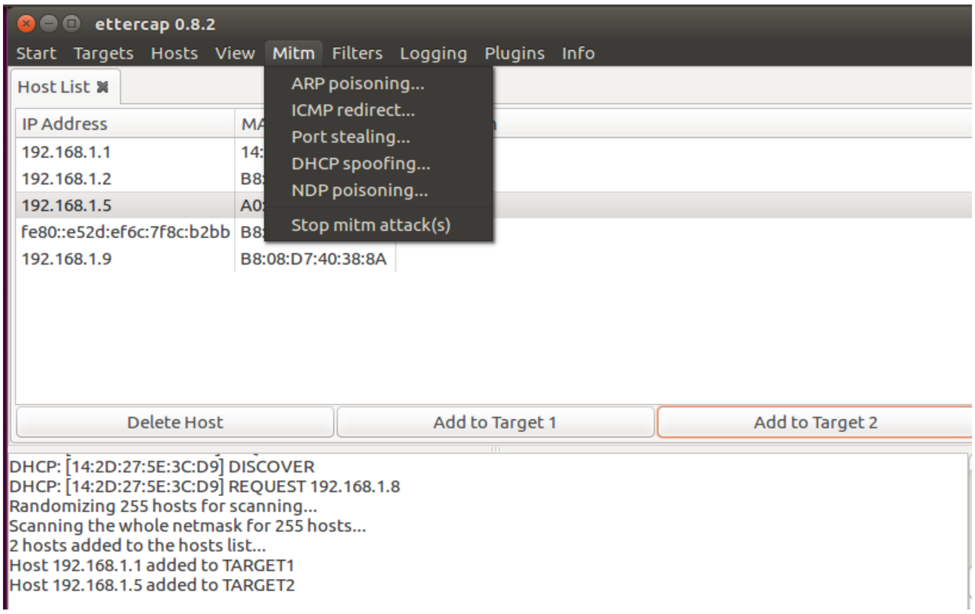
Figure 46. Man-in-the-middle attack topology

5.4.2. The set up on Ettercap

Activate the Unified sniffing for the Ettercap, to unify the attacker's machine MAC address with the legitimate user's machine MAC address, such that the server will not be able to distinguish the intruder's machine from the legitimate machine on the network. Activating the sniffing connection, it activates the vulnerability in the network so that layer 3 protocols could be tracked. This configuration will activate two-way poisoning, that is from the server to host-attacker and vice versa.

5.4.3. Results

On the Ettercap 0.8.2, as shown in Figure 47, the attacker’s machine matches the IP address of the authorized hosts to its own MAC address table on the network before establishing a false connection, poisoning the network, to steal data being communicated between the two legitimates hosts on the network.

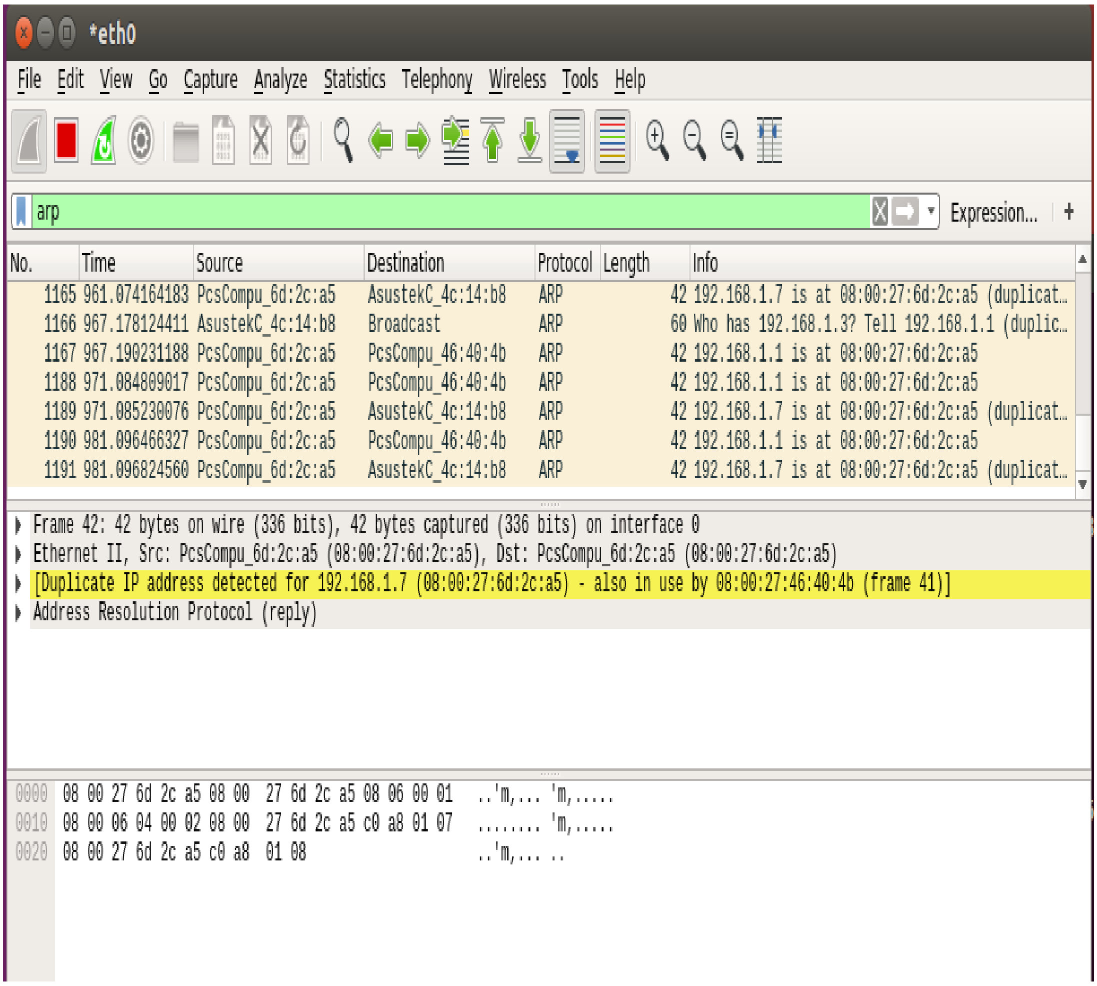


IP Address	MA
192.168.1.1	14:
192.168.1.2	B8:
192.168.1.5	A0:
fe80::e52d:ef6c:7f8c:b2bb	B8:
192.168.1.9	B8:08:D7:40:38:8A

```
DHCP: [14:2D:27:5E:3C:D9] DISCOVER
DHCP: [14:2D:27:5E:3C:D9] REQUEST 192.168.1.8
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
2 hosts added to the hosts list...
Host 192.168.1.1 added to TARGET1
Host 192.168.1.5 added to TARGET2
```

Figure 47. ARP Poisoning

The network traffic was captured using Wireshark as shown in Figure 48 when the sniffer was activated. On the Wireshark, we can see all the traffic on the network, the established connections and the activities on the connected hosts as captured on the attacker’s machine.


*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

arp

No.	Time	Source	Destination	Protocol	Length	Info
1165	961.074164183	PcsCompu_6d:2c:a5	AsustekC_4c:14:b8	ARP	42	192.168.1.7 is at 08:00:27:6d:2c:a5 (duplicat…
1166	967.178124411	AsustekC_4c:14:b8	Broadcast	ARP	60	Who has 192.168.1.3? Tell 192.168.1.1 (duplic…
1167	967.190231188	PcsCompu_6d:2c:a5	PcsCompu_46:40:4b	ARP	42	192.168.1.1 is at 08:00:27:6d:2c:a5
1188	971.084809017	PcsCompu_6d:2c:a5	PcsCompu_46:40:4b	ARP	42	192.168.1.1 is at 08:00:27:6d:2c:a5
1189	971.085230076	PcsCompu_6d:2c:a5	AsustekC_4c:14:b8	ARP	42	192.168.1.7 is at 08:00:27:6d:2c:a5 (duplicat…
1190	981.096466327	PcsCompu_6d:2c:a5	PcsCompu_46:40:4b	ARP	42	192.168.1.1 is at 08:00:27:6d:2c:a5
1191	981.096824560	PcsCompu_6d:2c:a5	AsustekC_4c:14:b8	ARP	42	192.168.1.7 is at 08:00:27:6d:2c:a5 (duplicat…

```
Frame 42: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: PcsCompu_6d:2c:a5 (08:00:27:6d:2c:a5), Dst: PcsCompu_6d:2c:a5 (08:00:27:6d:2c:a5)
[Duplicate IP address detected for 192.168.1.7 (08:00:27:6d:2c:a5) - also in use by 08:00:27:46:40:4b (frame 41)]
Address Resolution Protocol (reply)
```

```
0000  08 00 27 6d 2c a5 08 00  27 6d 2c a5 08 06 00 01   ..'m,... 'm,.....
0010  08 00 06 04 00 02 08 00  27 6d 2c a5 c0 a8 01 07   ........ 'm,.....
0020  08 00 27 6d 2c a5 c0 a8  01 08                     ..'m,... ..
```

Figure 48. ARP poisoning traffic on Wireshark

6. CONCLUSION

The importance and increase in the amount of data being transferred and saved have given birth to the idea of cloud computing. The technology has been offering lots of benefits to companies and organizations to own any cloud services of their choice and allows them to have access to their data regardless of the location at any time over the Internet. The security of the data has been the challenge to the cloud service providers, since data is kept in different places, and the fact that it needs to be always accessible to customers upon demand. It is observed in this work that the communication channels, storage infrastructure and the data itself are the target areas for the attackers. The data at the three different states, at rest, on motion and in use, suffer from various types of attacks, among which are the DDoS, ARP poisoning, password phishing, database injection and others. The main reasons why these attacks are carried out, are to tamper with the integrity, availability and the confidentiality of the data.

From the results of the simulation conducted in this work using simulating tools such as NS3, Ettercap, EstiNet and DVWA, so that more light is shed on the impact of these attacks on data, and network resources for our better understanding. The DDoS targets the availability of the resources by affecting the network throughput, and reducing the amount of the transmitted packets, thereby degrading the quality and availability of the given service. On the EstiNet simulation, it was observed, that the attacker infected the network with around 1.2MBps of traffic, which led the server and host workstation to be compromised with the packet loss to around 0.05kBps on the average. Similar scenarios are also witnessed on the NS3 simulation where the communication between the links was congested, the communication output in the network dropped from 5Mbps to 0.012Mbps and packets drop.

In the same vein, confidentiality of the stored data was compromised when the Ettercap was used to poison a network by sniffing the information being transmitted between authorized users. With this, it is obvious to observe the importance of secured communication links. The attack is aimed at the network layer to sniff packets moving to-and-from in the network. This kind of attack compromises the network, making communication between the legitimate users and/or the server prone to attack. The communication between users was captured using Wireshark when Ettercap was set up as an attacker on the network. The transmitted information was captured and attackers could easily use it against the legal user or for other personal use.

Finally, the integrity of a data is put at risk when an unauthorized user could be able to steal or manipulate valuable information from a database. This was explained using the DVWA simulation conducted in this work. As it is known that database is one of the most sensitive areas on a cloud system where bad configuration, and inappropriate security measures on this area will open it to attacks, and which will be detrimental to the integrity of the organization.

Some security techniques solutions were discussed in chapter 3 of this work as well, which have been in use, such as the cryptographic techniques; for the cloud attacks, database security, and digital signature; and offering solutions to protect data in the cloud. The effective implementation of these techniques and putting other security measures in place will help in securing important and valuable data in the cloud, regardless of the place where they are located and saved for easy access.

LIST OF REFERENCES

AbdElnapi Noha MM., Fatma A. Omara, Nahla F.Omran (2016). *A Hybrid Hashing Security Algorithm for Data Storage on Cloud Computing*. International Journal of Computer Science and Information Security (IJCSIS) Vol. 14, No. 4.

Arron Fu (2017). *7 Different Types of Cloud Computing Structures*. Uniprint.net Publication. Available: https://www.uniprint.net/en/7-types-cloud-computing-structures/

Berkeley University of California (2017). *Data Encryption in Transit Guideline*. Berkeley Information Security Policy. Available: https://security.berkeley.edu/data-encryption-transit-guideline

Bernard Pasquier (2014). *EstiNet Network Simulator and Emulator*. Presentation Script. Available: https://www.slideserve.com/benard/estinet-network-simulator-emulator

Bhalekar Seema Banduji, Thakare V.M., Junghare U.S., (2014). *Cloud Data Storage Security Techniques and Security Issues on Mobile Devices*. International Journal of Computer Applications (0975- 8887). Second National Conference on Recent Trends in Information Security, GHRCE, Nagpur, India.

Chris Kowalczyk (2017). *Crypto-IT: Meet-in-the-middle attack*. Available: http://www.crypto-it.net/eng/attacks/meet-in-the-middle.html

Corey Plett Liza Poggemeyer and Justinha (2016). *TLS-SSL (Schannel SSP) Overview*. Microsoft documentation. Available on: https://docs.microsoft.com/en-us/windows-server/security/tls/tls-ssl-schannel-ssp-overview

DocuSign (2018). *Understanding Digital Signature: What are digital signatures?*. Available: https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq

Gerald Kaefer (2010). *Cloud Computing Architecture*. 4th Generation Datacentre IEEE Spectrum, Feb. 2009. Siemens Corporate Research and Technologies, Munich Germany.

Govinda Raj (2017). *SQL Injection*. A Medium Corporation. Available: https://medium.com/@govinda_raj/sql-injection-71b39ff637fa.

Hsieh Hung-Yun, Sivakumar R (2002). *pTCP: an end-to-end transport layer protocol for stripped connections*. Published in Network Protocols, 2002. Proceedings of 10th IEEE International Conference. ISSN: 1092-1648. Available: http://ieeexplore.ieee.org/document/1181383/metrics

IBM Corporation (1999, 2014). *An Overview of the SSL or TLS Handshake*. Available: https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10660_.htm

IBM Corporation (2013). *IBM Knowledge Centre: TCP Connection Status*. Available: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.halu101/constatus.htm

Jain Ankit. *Cryptography- The Science of Secrecy: Types of Cryptosystem*. Available: http://www.ankitjain.info/articles/Cryptography_ankit4.htm

Jared Heinrich (2013). *Mastering the OSI & TCP/IP Models*. Available: http://jaredheinrichs.com/mastering-the-osi-tcpip-models.html

Khan Shakeeba S and Tuteja R. R. (Prof.), (2015). *Security in Cloud Computing using Cryptographic Algorithms*. International Journal of Innovative Research in

Computer and Communication Engineering (ISO 3297:2007 Certified Organization) Vol.3 Issue 1.

Kozierok Charles M. (2005). *TCP/IP Guide – TCP Operational Overview and the TCP Finite State Machine (FSM)*. Available: http://tcpipguide.com/free/t_TCPOperationalOverviewandtheTCPFiniteStateMachineF-2.htm

Ma Richard T.B (2015). *Software Defined Network – CS 4226: Internet Architecture*. School of Computing. The National University of Singapore. Available: http://docplayer.net/15055142-software-defined-networking.html

Manikandasaran S.S. (Dr.) (2016). *Security Attacks and Cryptography Solutions for Data Stored in Public Cloud Storage*. IRACST- International Journal of Computer Science and Information Technology and Security (IJCSITS). ISSN: 2249-9555, Vol.6, No.1.

Martin Rakhmanov (2010). Trustwave Holdings, Inc. *Encrypting Data at Rest*. Available: https://www.trustwave.com/Resources/SpiderLabs-Blog/Encrypting-Data-at-Rest/

Michelle Boisvert and Stephen J. Bigelow (2013). *Infrastructure as a Service*. TechTarget Network. Public Cloud – IaaS Cloud Deployment. Available: https://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS

Nadra Waheed (2013). *Attacks on Cloud Computing*. Carleton University Course Material. Available: http://people.scs.carleton.ca/~maheshwa/courses/4109/cloud-attacks.pdf

Nick Sullivan (2014). *Keyless SSL: The Nitty Gritty Technical Details. CloudFlare Blog Inc*. Available: https://blog.cloudflare.com/keyless-ssl-the-nitty-gritty-technical-details/

Nick Sullivan and Douglas Stebila (2015). *An Analysis of TLS Handshake Proxying*. Available: https://crypto.dance/projects/6239269

NS3 (2011-2017). *Documentation: A Discrete Event Simulator - Key Technologies*. Available: https://www.nsnam.org/overview/key-technologies/

NS3 (2011-2017). *Documentation: A Discrete Event Simulator - TCP Models in ns3*. Available: https://www.nsnam.org/docs/models/html/tcp.html

Oktay U and O.K. Sahingoz (2013). *Attack Types and Intrusion Detection Systems in Cloud Computing*. 6th International Information Security & Cryptology Conference, ISC, Turkey.

Oracle Corporation (2018). *MySQL: Documentation- mysql_real_escaping_string*. Available: https://dev.mysql.com/doc/apis-php/en/apis-php-function.mysql-real-escape-string.html

Randeep Kaur and Supriya Kinger (2014). *Analysis of Security Algorithms in Cloud Computing*. Volume 3, Issue 3, March 2014. International Journal of Application or Innovation in Engineering and Management (IJAIEM). ISSN 2319- 4847.

Raj Jain (2011). *Advanced Encryption Standards*. Washington University in Saint Louis, Saint Louis, MO 63130. Available: https://www.cse.wustl.edu/~jain/cse571-11/ftp/l_05aes.pdf

Rickard Cayla (2015). *Security Lab 2: Man in the Middle Attack*. Available: http://slideplayer.com/slide/1647221/

Sagar Aman, Bineet Kumar Joshi and Nishant Mathur (2013). *A study of Distributed Denial of Service Attack in Cloud Computing (DDoS)*. Edition on Cloud and Distributed Computing: Advanced and Applications. e-ISSN: 2321-6980. Available: http://stl.hctl.org/vol2/STL_Article_201308002.pdf

Saleh Asadollahi, Dr Bhargavi Goswami, Dr Atul M Gonsai (2017). *Software Defined Network, Controller Comparison*. International Journal of Innovative Research in Computer and Communication Engineering. ISSN (online): 2320-9801

Satyam Singh, (2015). *Application Architecture Review. Infosec Institute*. Available: http://resources.infosecinstitute.com/application-architecture-review/#gref

Schulzrinne H, Frederick R and Jacobson V, (2003). *Standards Track – A Transport Protocol for Real-Time Applications*. Available: https://tools.ietf.org/html/rfc3550#page-19

Stallings William (2011). *Cryptography and Network Security: Principles and Practice*. 5th edition. Published by Pearson Education. ISBN-13: 978-0-13-705632-3. PP 270-280.

Stephen R. Smoot, Nam K. Tan (2012). *Private Cloud Computing: Consolidation, Virtualization and Service Oriented Infrastructure*. Published by MK publications. ISBN: 978-0-12-384919-9.

Tim Rains (2015). *SCloud Security Controls Series: Encrypting Data in Transit*. Microsoft Secure Publication. Available: https://cloudblogs.microsoft.com/microsoftsecure/2015/08/10/cloud-security-controls-series-encrypting-data-in-transit/

Teare Diane (1999). *Designing CISCO Networks*. Indianapolis: CISCO Press, July 1999. Available:

http://docwiki.cisco.com/wiki/Internetworking_Basics#OSI_Model_Transport_Layer

The EstiNet Technologies Inc., (2000-2016). *The GUI User Manual for the EstiNet 9.0 Network Simulator*. Available: http://www.estinet.com/ns/wp-content/uploads/2015/12/EstiNet_9.0_GUIManual_20150803.01.pdf

Vairagade Rauli Sachin, Nitin Ashokrao Vairagade (2012). *Cloud Computing Data Storage and Security Enhancement*. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Volume 1, Issue 6, August 2012

ViSolve IT Security Team (2013). *Securing Data at Rest*. ViSolve Inc. Open Source Solutions. Available: http://www.visolve.com/uploads/resources/Securing-DataAtRestwithEncryption.pdf

Waleed Ahmed (2015). *Simulation and Evaluation of Wired and Wireless Networks with NS2, NS3 and OMNET++*. Master's thesis for the degree of Master of Science in Technology. University of Vaasa, Finland.

Yassin Ali A., Hikmat Z. Neima, and Haider Sh. Hashim (2014). *Security and Integrity of Data in Cloud Computing Based on Feature Extraction of Handwriting Signature.* International Journal of Cyber-Security and Digital Forensic (IJCSDF) 3(2):93-105. The Society of Digital Information and Wireless Communication, 2014 (ISSN:2305-0012).