# Practical – 1

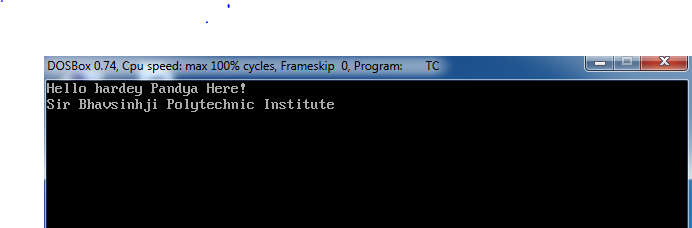
**(A) Generate an executable file from C compiler and generate its message digest sum (MD5). Note down MD5.**

**Create a C file:**



[1.1 C file]

**Output:**



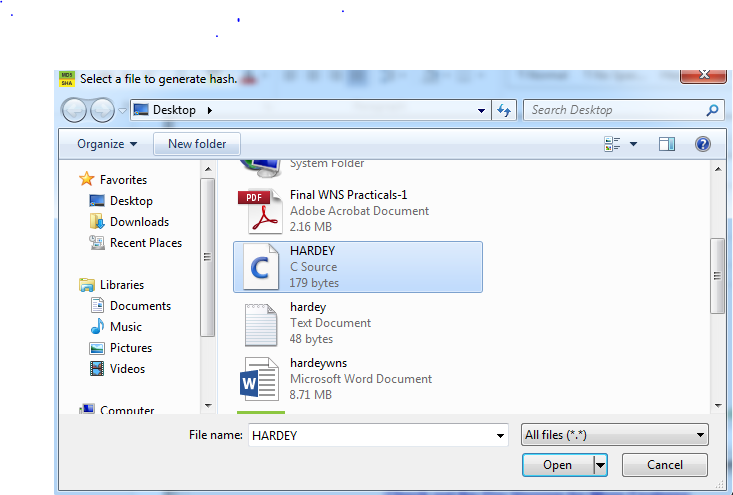
[1.2 Output]

**Open MD5 and SHA Checksum Utility Software:**



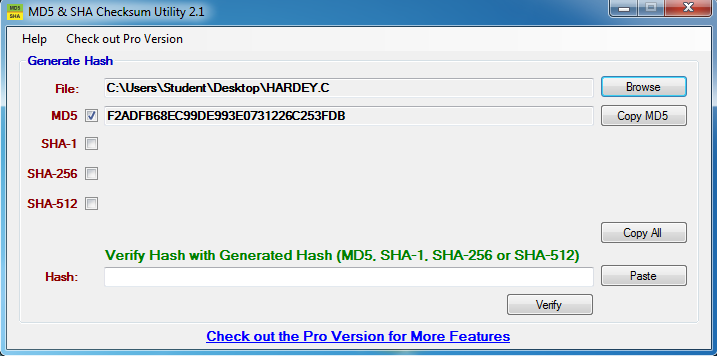
[1.3 MD5 and SHA Checksum Utility Software]

**Open C file:**



[1.4 Open file]

**Generate MD-5 hash value of C file:**



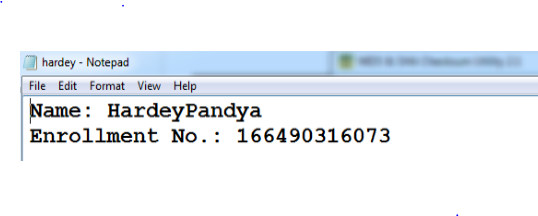
[1.5 MD5 Hash Value]

**Output:**

MD\_5: F2ADFB68EC99DE993E0731226C253FDB

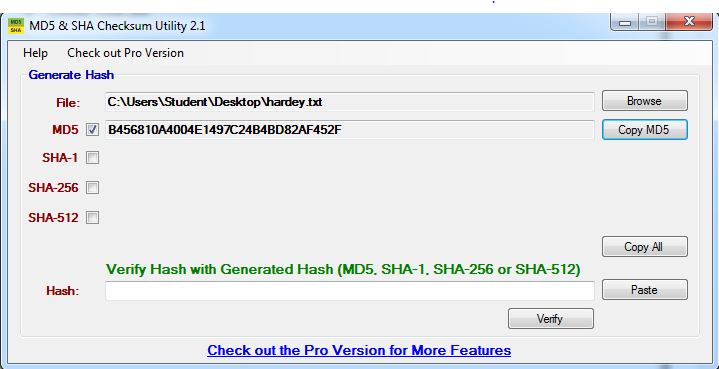
**(B)** **Change the above C program with a minor modification and again generate its executable. Check the MD5 of the new file. Verify the MD5 of both the files.**

**Create Text file:**



[1.6 Text File]

**Generate MD5 hash:**



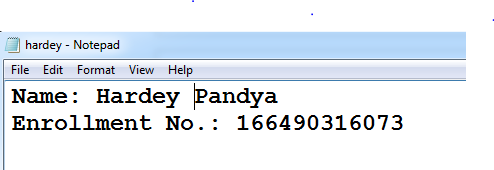
[1.7 MD5 Hash]

**Output:**

MD\_5: B456810A4004E1497C24B4BD82AF452F

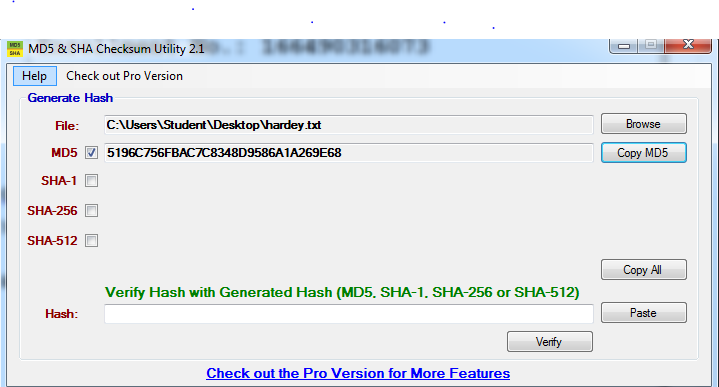
**(C) Change the above text file with minor modification and again generate its executable. Check the MD5 of the new file and verify the MD5 of both files.**

**Create Text file:**



[1.8 Text File with slight change]

**Generate MD\_5:**

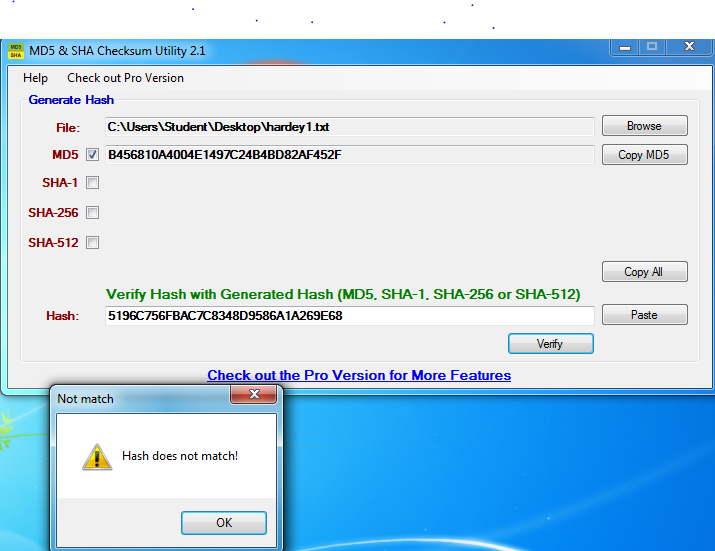


[1.9 MD5 Hash]

**Output:**

MD\_5: 5196C756FBAC7C8348D9586A1A269E68

**ii. Compare with previous hash:**



[1.10 Compare with Previous Hash]

MD\_5(file\_1): B456810A4004E1497C24B4BD82AF452F

MD\_5(file\_2): 5196C756FBAC7C8348D9586A1A269E68

**Conclusion:**

* Here in the given example we created a text file and generated MD5 hash value.Than we modified the text file and compiled both the values of MD5. As a result we concluded that both MD5 values were different at different MD5 value generation .
* So , by this we can conclude that even if there is a single change in the file , the hash value which was previously generated will not match.

# Practical: 2

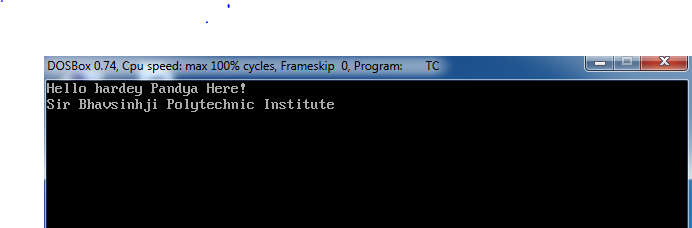
**(A) Generate an executable file from a C compiler and generate is Secure Hash Algorithm (SHA-1) sum. Note down the SHA value.**

**Create a C file:**



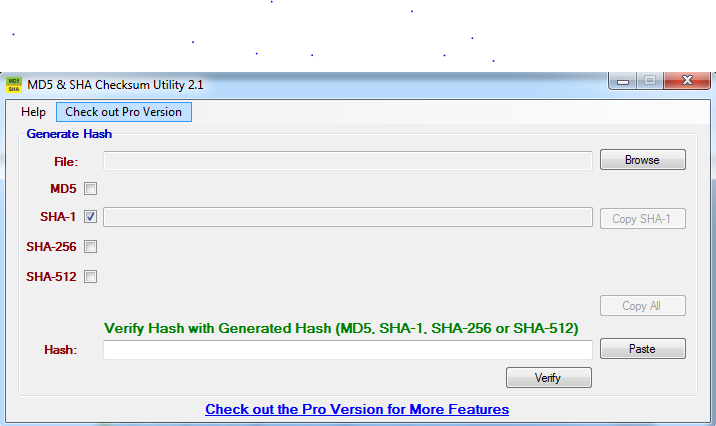
[2.1 C file]

**Output:**



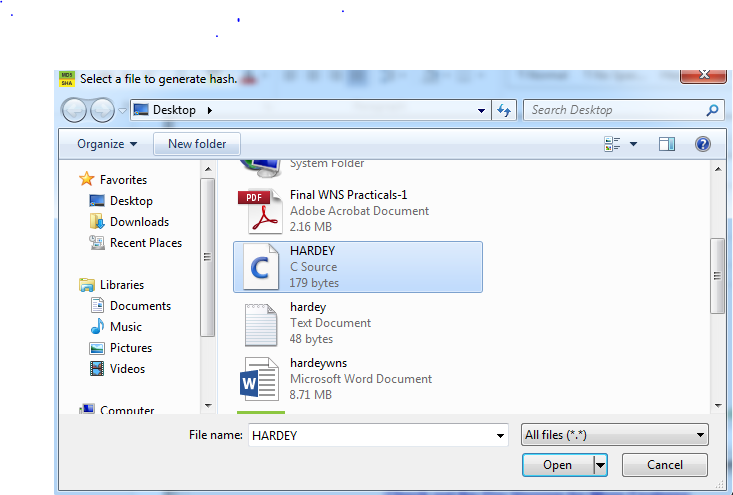
[2.2 Output]

**Open MD5\_and\_SHA\_Checksum\_Utility Software:**



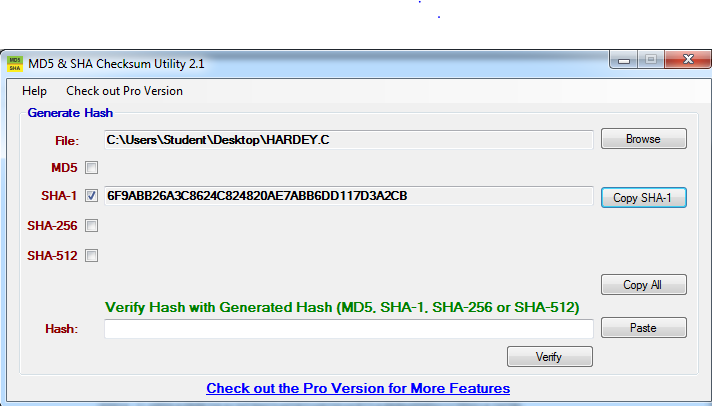
[2.3 MD5\_and\_SHA\_Checksum\_Utility Software]

**Open C file:**



[2.4 Open file]

**Generate SHA-1:**



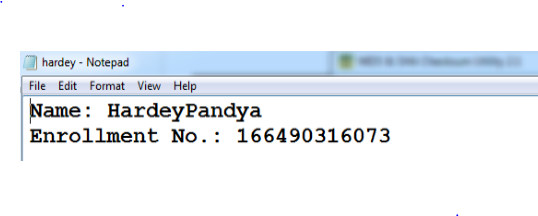
[2.5 Genereate SHA1]

**Output:**

SHA-1: 6F9ABB26A3C8624C824820AE7ABB6DD117D3A2CB

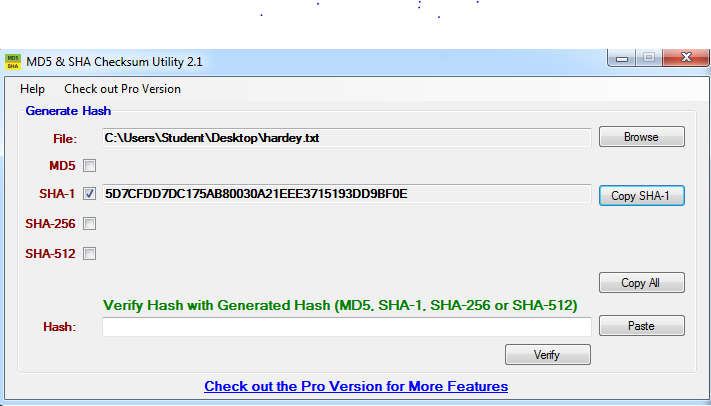
**(B) Change the above C program with a minor modification and again generate its executable. Check the SHA 256 and 512 of the new file. Verify the SHA values of both the files.**

**Text file:**



[2.6 Text File]

**Generate SHA-1:**



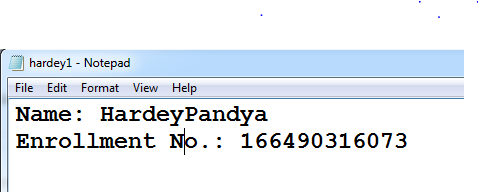
[2.7 Generate SHA1]

**Output:**

SHA-1: 5D7CFDD7DC175AB80030A21EEE3715193DD9BF0E

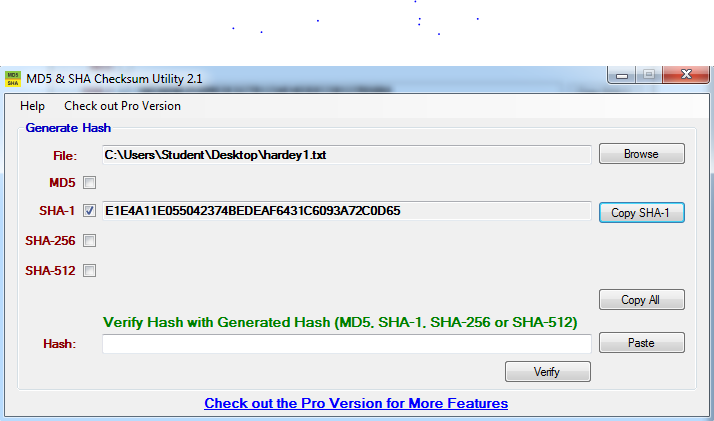
**(C) Change the above text file with a minor modification and again generate its executable.**

**Text file:**



[2.8 Text File with Slight Change]

**Generate SHA-1:**



[2.9 Generate SHA1]

**Output:**

SHA-1: E1E4A11E055042374BEDEAF6431C6093A72C0D65

**(D)Compare with previous SHA-1.**



[2.10 Compare with previous Hash]

SHA-1(file\_1): 5D7CFDD7DC175AB80030A21EEE3715193DD9BF0E

SHA-1(file\_2): E1E4A11E055042374BEDEAF6431C6093A72C0D65   
**Conclusion:**

* + Here in the given example we created a text file and generated SHA\_1 hash value. Then we modified the text file and compiled both the values of SHA-1. As a result, we concluded that both MD5 values were different at different SHA-1 value generation.
  + So, by this we can conclude that even if there is a single change in the file, the hash value which was previously generated will not match.

**Practical – 3**

**3. Prepare a presentation of RSA and explain its working and structure**.

* RSA algorithm was the first practicable implementation of asymmetric cryptosystem. It was invented by Rivest, Shamir & Adleman of MIT in 1977. It is best known & widely used public-key algorithm. It is based on fact that it is easy to find and multiply large prime numbers, but difficult to factor their products and it can be used for user authentication.

* It can be used for user authentication, data encryption and digital data signing. Hardware implementation of RSA is also possible. It is a public key algorithm. However, these two keys are interrelated. One of these two keys is used to encrypt the message on the sender side, another is used to decrypt the message on the receiver side.

* The private and public keys in RSA are made up by large prime numbers. So, selection and generation of large prime number is a real challenge in RSA implementation.

**There are two steps involved in RSA algorithm.**

1. Key Generation
2. Encryption/Decryption

**RSA Algorithm**

**1. Key generation**

Select two large prime number randomly, let’s say p and q.

1. Calculate n and (n). n=p\*q

(n)=(p-1)\*(q-1)

1. Select public key exponent e randomly, which should meet following condition. a. 1 < e < (n)

b. GCD(e,  (n)) = 1

3. Calculate private key exponent d using following formula:

d=e-1 mod  (n)

After generating private key exponent d and public key exponent e encryption and decryption of message can be done as follow:

**2. Encryption/Decryption**

For Encryption:

CT=PTe mod n

For Decryption:

PT=CTd mod n

**Example of RSA Algorithm**

Let’s take P=7 and Q=17. Now using key generation process, let’s find out N, (N), e and d.

**Step 1: Calculate N and**  **(N)**

N= P \* Q = 7 \* 17

N=119

 (N)=(P-1)\*(Q-1)=(6)\*(16)

(N)=96

**Step 2: Select public key exponent e**

Value of e should be randomly chosen in such a way so that a. 1< e <(N)

b. GCD(e, (N))=1

If we take e=5, then it satisfies above both condition. 5 is grater then 1 and less (N). And GCD of (5,96)=1.

**Step 3: Calculate private key exponent d**

d=e-1 mod (N) =5-1 mod 96 d=77

So, we have got public key exponent e=5, and private key exponent d=77. Let’s take plain text PT=6.

**Encryption:**

CT=PTe mod N

=65 mod 119

CT=41

So 5 plain text is converted to 41 cipher text using receivers public key exponent e. Thus, sender sends 41 after encryption.

**Decryption:**

PT=CTd mod N

=4177 MOD 119

PT=6

Receiver uses private key exponent d to get plain text back from cipher text. Using which, receiver gets plain text 6 back from 41 cipher text.

**The Security of RSA**

**Four possible approaches to attacking the RSA algorithm are**

* Brute force: This involves trying all possible private keys.
* Mathematical attacks: There are several approaches, all equivalent in effort to Factoring the product of two primes.
* Timing attacks: These depend on the running time of the decryption algorithm.
* Chosen cipher text attacks: This type of attack exploits properties of the RSA Algorithm.

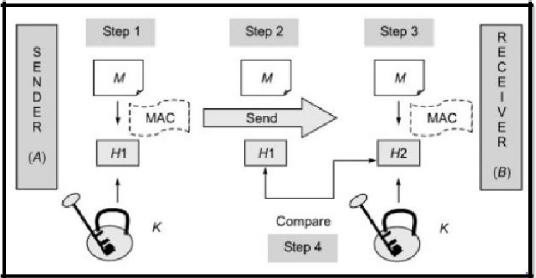
* The defence against the brute-force approach is the same for RSA as for other Cryptosystems, namely, to use a large key space. Thus, the larger the number of bits and, the better. However, because the calculations involved, both in key generation.

**Practical – 4**

**4. Prepare a Chart/model on Message Authentication Codes (MACs).**

* The concept of message authentication code (MAC) is quite similar to that of a message digest.
* The difference is that message digest is simply a fingerprint of a message. There is no cryptographic process involved in the case of a message digest whereas MAC involved cryptographic processing. Let’s us assume that the sender A wants to send a message M message M to a receiver B.

How the MAC processing works shown below.



[Fig:4.1 MAC Processing]

1. Key k, which is not known to anyone else. A calculates the MAC by applying key K to the message M.
2. A then sends the original message M and the MAC H1 to B.
3. When B receives the message, B also uses K to calculate its own MAC H2 over M.
4. B now compares H1 with H2. If the two match, B concludes that the message M has not been changes during transit. However if H1≠H2, B reject the message, realizing that the message was changed during transit.

**Requirements for Message Authentication Codes**

* + A MAC, also known as a cryptographic checksum, is generated by a function C of the form

T=MAC (K, M)

* + Where is a variable-length message, is a secret key shared only by sender and receiver, and MAC (,) is the fixed-length authenticator, sometimes called a tag. The tag isappended to the message at the source at a time when the message is assumed or known to be correct. The receiver authenticates that message by re computing the tag.
  + When an entire message is encrypted for confidentiality, using either symmetric or asymmetric encryption, the security of the scheme generally depends on the bit length of the key. Barring some weakness in the algorithm, the opponent must resort to a brute-force attack using all possible keys. On average, such an attack will require attempts for a -bit key. In particular, for a cipher text only attack, the opponent, given cipher text, performs for all possible key values until a is produced that matches the form of acceptable plaintext.

**Security of MACs**

* + Just as with encryption algorithms and hash functions, we can group attacks on MACs into two categories: brute-force attacks and cryptanalysis.

**Brute-Force Attacks**

* + A brute-force attack on a MAC is a more difficult undertaking than a brute-force attack on a hash function because it requires known message-tag pairs. Let us see why this is so. To attack a hash code, we can proceed in the following way. Given a fixed message with -bit hash code, a brute-force method of finding a collision is to pick a random bit string and check if. The attacker can do this repeatedly off line. Whether an off-line attack can be used on a MAC algorithm depends on the relative size of the key and the tag.

**Cryptanalysis**

* + As with encryption algorithms and hash functions, cryptanalytic attacks on MAC algorithms seek to exploit some property of the algorithm to perform some attack other than an exhaustive search. The way to measure the resistance of a MAC algorithm to cryptanalysis is to compare its strength to the effort required for a bruteforce attack. That is, an ideal MAC algorithm will require a cryptanalytic effort greater than or equal to the brute-force effort.
  + There is much more variety in the structure of MACs than in hash functions, so it is difficult to generalize about the cryptanalysis of MACs. Furthermore, far less work has been done on developing such attacks.

**Practical – 5**

**Aim: Prepare a chart/model to explain the working of Digital Signature.**

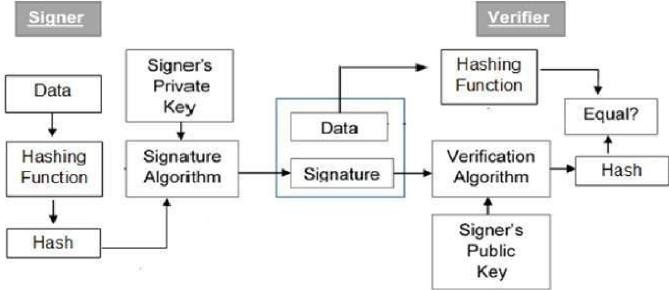
“A digital signature is an authentication mechanism that enables the creator of a message to attach a code that act as a signature.”

* The signature is formed by taking the hash of the message and encrypting the message with the creator’s private key. The signature guarantees the source and integrity of the message. The digital signature provides a set of security capabilities that would be difficult to implement any other way.

[

Fig : 5.1 Digital Signature

]



**How Digital Signature works Signing**

* Public key cryptography gives a reliable method for digital signing and signature verification based on public/private key pairs. A person can sign a given digital message (file, document, e-mail, and so forth) with his private key. From a technical point of view, the **digital signing of a message** is performed in two steps:

[

Fig : 5.2 D

igital

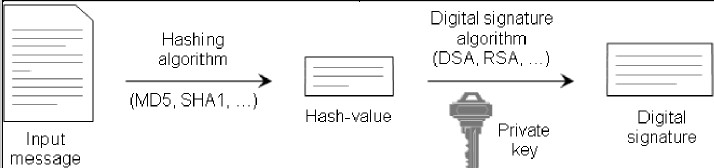
S

igning of a

M

essage

]



**Step 1: Calculate the Message Digest**

* In the first step of the process, a **hash-value** of the message (often called the **message digest**) is calculated by applying some cryptographic **hashing algorithm** (for example, MD2, MD4, MD5, SHA1, or other). The calculated hash-value of a message is a sequence of bits, usually with a fixed length, extracted in some manner from the message.

**Step 2: Calculate the Digital Signature**

* In the second step of digitally signing a message, the information obtained in the first step hash-value of the message (the message digest) is encrypted with the private key of the person who signs the message and thus an encrypted hash-value, also called **digital signature**, is obtained. For this purpose, some mathematical cryptographic **encrypting algorithm** for calculating digital signatures from given message digest is used.

**Verification**

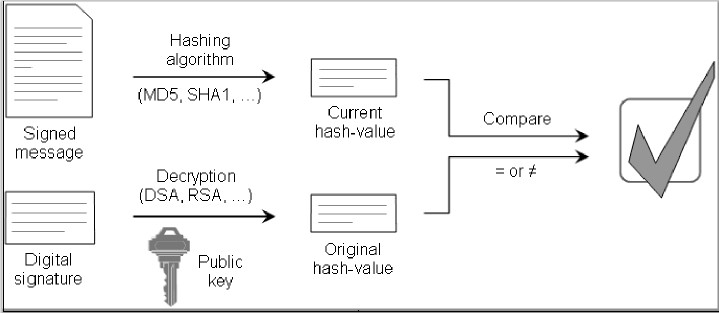
* Digital signature technology allows the recipient of given signed message to verify its real origin and its integrity. The process of **digital signature verification** is purposed to ascertain if a given message has been signed by the private key that corresponds to a given public key. The digital signature verification cannot ascertain whether the given message has been signed by a given person.
* The verification of a digital signature is performed in three steps:

[

Fig : 5.3

verification of a digital signature

]



**Step 1: Calculate the Current Hash-Value**

* In the first step, a hash-value of the signed message is calculated. For this calculation, the same hashing algorithm is used as was used during the signing process. The obtained hash-value is called the **current hash-value** because it is calculated from the current state of the message.

**Step 2: Calculate the Original Hash-Value**

* In the second step of the digital signature verification process, the digital signature is decrypted with the same encryption algorithm that was used during the signing process. The decryption is done by the public key that corresponds to the private key used during the signing of the message. As a result, we obtain the **original hashvalue** that was calculated from the original message during the first step of the signing process (the original message digests).

**Step 3: Compare the Current and the Original Hash-Values**

* In the third step, we compare the current hash-value obtained in the first step with the original hash-value obtained in the second step. If the two values are identical, the verification if successful and proves that the message has been signed with the private key that corresponds to the public key used in the verification process. If the two values differ from one another, this means that the digital signature is invalid and the verification is unsuccessful.

**Digital Signature Requirements**

1. The signature must be a bit pattern that depends on the message being signed.
2. The signature must use some information unique to the sender to prevent both forgery and denial.
3. It must be relatively easy to produce the digital signature.
4. It must be relatively easy to recognize and verify the digital signature.
5. It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
6. It must be practical to retain a copy of the digital signature in storage.

**A digital signature consists of three algorithms.**

1. **A key generation algorithm:** That selects a private key uniformly at random form a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
2. **Assigning algorithm:** That given a message and a private key produces signature.
3. **A signature verifying algorithm:** That given a message public key and a signature either accepts or rejects the message’s claim authenticity.

**Properties of Digital Signature**

1. The **authenticity** of a signature generated from a fixed message and fixed private key can be verified by using the corresponding public key.
2. It should be computationally infeasible to generate a valid signature for a party without knowing that party’s private key.

**Attacks and Forgeries**

1. **Key-only attack:** C only knows A‟s public key.
2. **Known message attack:** C is given access to a set of messages and their signatures.
3. **Generic chosen message attack:** C chooses a list of messages before attempting to breaks

A‟s signature scheme, independent of A‟s public key. C then obtains from a valid signature for the chosen messages. The attack is generic, because it does not depend on A‟s public key; the same attack is used against everyone.

1. **Directed chosen message attack:** Similar to the generic attack, except that the list of messages to be signed is chosen after C knows A‟s public key but before any signatures are seen.

1. **Adaptive chosen message attack:** C is allowed to use A as an “oracle.” This means the A may request signatures of messages that depend on previously obtained message– signature pairs.
2. **Total break:** C determines A’s private key.
3. **Universal forgery:** C finds an efficient signing algorithm that provides an equivalent way of constructing signatures on arbitrary messages.
4. **Selective forgery:** C forges a signature for a particular message chosen by C.
5. **Existential forgery:** C forges a signature for at least one message. C has no control over the message. Consequently, this forgery may only be a minor nuisance to A.

**Practical – 6**

**6. (A) Install Wireshark tool for packet capture**

* In late 1997 Gerald Combs needed a tool for tracking down network problems and wanted to learn more about networking so he started writing Ethereal (the original name of the Wireshark project) as a way to solve both problems.
* Ethereal was initially released after several pauses in development in July 1998 as version 0.2.0. Within days patches, bug reports, and words of encouragement started arriving and Ethereal was on its way to success. Not long after that Gilbert Ramirez saw its potential and contributed a low-level dissector to it.
* In October, 1998 Guy Harris was looking for something better than tcp view so he started applying patches and contributing dissectors to Ethereal.
* In late 1998 Richard Sharpe, who was giving TCP/IP courses, saw its potential on such courses and started looking at it to see if it supported the protocols he needed. While it didn’t at that point new protocols could be easily added. So he started contributing dissectors and contributing patches.
* The list of people who have contributed to the project has become very long since then, and almost all of them started with a protocol that they needed that Wireshark or did not already handle. So, they copied an existing dissector and contributed the code back to the team.
* In 2006 the project moved house and re-emerged under a new name: Wireshark.
* In 2008, after ten years of development, Wireshark finally arrived at version 1.0. This release was the first deemed complete, with the minimum features implemented. Its release coincided with the first Wireshark Developer and User Conference, called Sharkfest. In 2015 Wireshark 2.0 was released, which featured a new user interface.



**What is Wireshark?**

* Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.
* You could think of a network packet analyzer as a measuring device used to examine what’s going on inside a network cable, just like a voltmeter is used by an electrician to examine what’s going on inside an electric cable (but at a higher level, of course).
* In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, all that has changed.
* Wireshark is perhaps one of the best open source packet analyzers available today.

**Some intended purposes**

* Here are some examples people uses Wireshark for:
* Network administrators use it to troubleshoot network problems
* Network security engineers use it to examine security problems
* Developers use it to debug protocol implementations
* People use it to learn network protocol internals **Features**

* The following are some of the many features Wireshark provides:  Available for UNIX and Windows.
* Capture live packet data from a network interface.
* Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
* Import packets from text files containing hex dumps of packet data.
* Display packets with very detailed protocol information.
* Save packet data captured.
* Export some or all packets in a number of capture file formats.
* Filter packets on many criteria.
* Search for packets on many criteria.
* Colorize packet display based on filters.
* Create various statistics.

…and *a lot more!*

**System Requirements**

**Microsoft Windows**

* The current version of Wireshark should support any version of Windows that is still within its extended support lifetime. At the time of writing this includes Windows 10, 8, 7, Vista, Server 2016, Server 2012, Server 2008 R2, and Server 2008.
* Any modern 64-bit AMD64/x86-64 or 32-bit x86 processor.
* 400 MB available RAM. Larger capture files require more RAM.
* 300 MB available disk space. Capture files require additional disk space.
* 1024×768 (1280×1024 or higher recommended) resolution with at least 16 bit color. 8 bit color should work but user experience will be degraded. Power users will find multiple monitors useful.

**UNIX / Linux**

* Wireshark runs on most UNIX and UNIX-like platforms including OS X and Linux. The system requirements should be comparable to the Windows values listed above. Binary packages are available for most Unices and Linux distributions including the following platforms:
* Apple OS X
* Debian GNU/Linux
* FreeBSD
* Gentoo Linux
* HP-UX
* Mandriva Linux
* NetBSD
* OpenPKG
* Red Hat Enterprise/Fedora Linux
* Sun Solaris/i386
* Sun Solaris/SPARC
* Canonical Ubuntu

**Installing on Microsoft Windows Systems**

The first step when installing Wireshark under Windows is to obtain the latest installation build from the official Wireshark web page, http://www.wireshark.org/. Navigate to the Downloads section on the website and choose a mirror. Once you’ve downloaded the package, follow these steps:

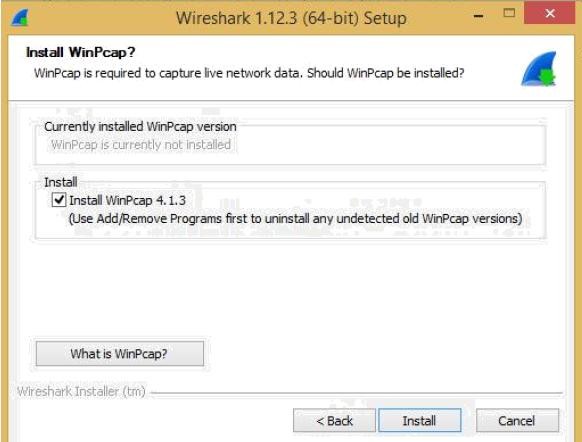
1. Double-click the .exe file to begin installation, and then click next in the introductory window.
2. Read the licensing agreement, and then click I Agree if you agree.
3. Select the components of Wireshark you wish to install, as shown in Figure. For our purposes, you can accept the defaults by clicking next.

[

Fig : 6.1

Wireshark

SetUp]

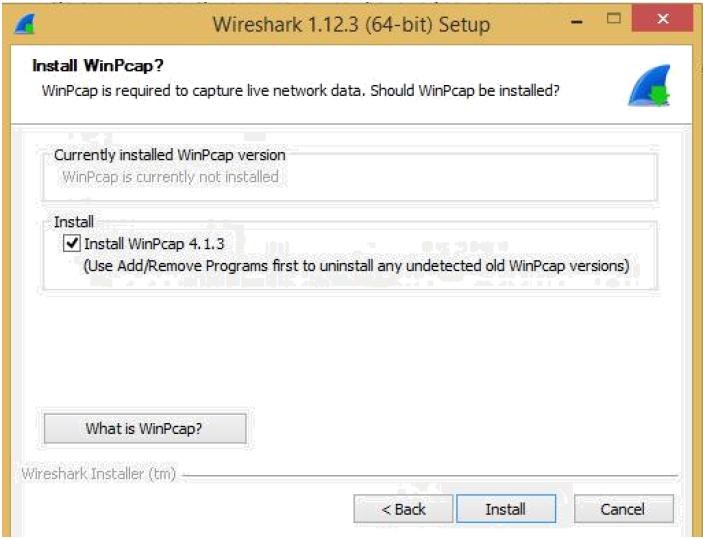
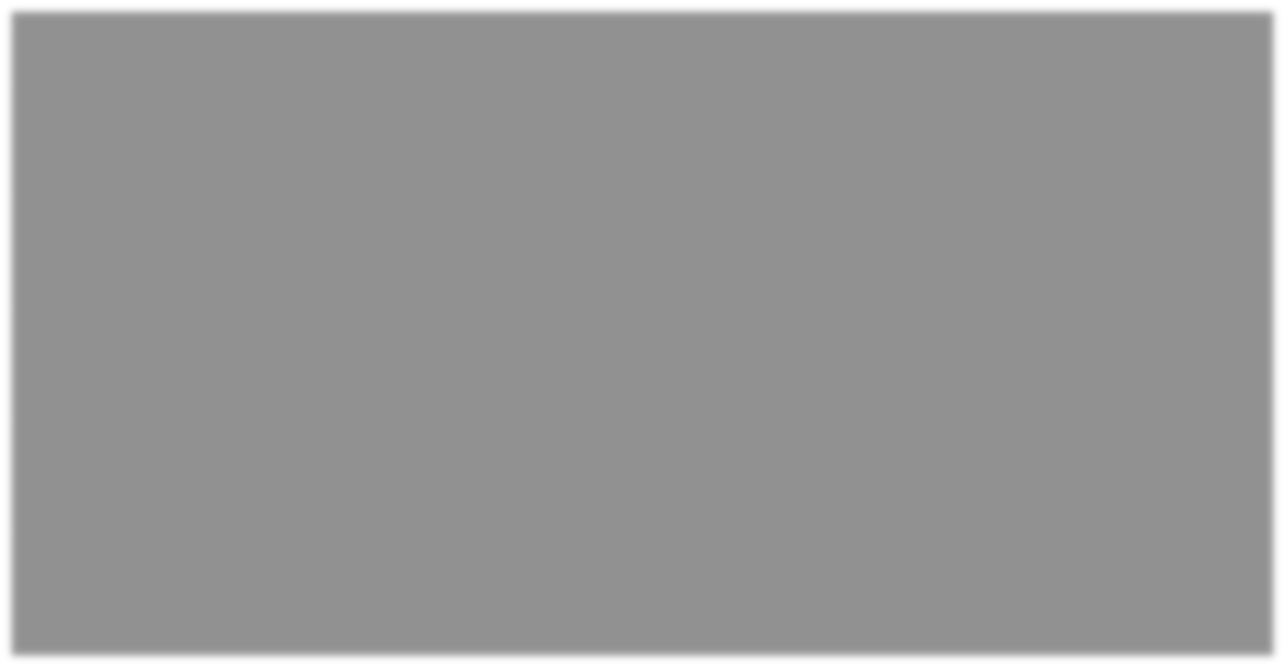


1. About halfway through the Wireshark installation, the WinPcap installation should start. When it does, click next in the introductory window, read the licensing agreement, and then click I Agree. Introduction to Wires hark 39
2. WinPcap should install on your computer. After this installation is complete, click Finish.
3. Wireshark should complete its installation. When it’s finished, click next.
4. In the installation confirmation window, click Finish.

[

Fig : 6.2 Installation of WireShark

]



**(B)inspect Packets and identify source and destination IP using Wireshark tools .**

**Packet Capture & Traffic Analysis with Wireshark**

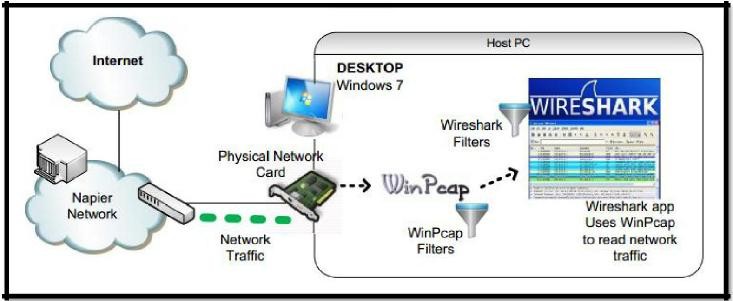
**Packet Capture (Packet Sniffing)**

* + A packet sniffer is an application which can capture and analyse network traffic which is passing through a system’s Network Interface Card (NIC). The sniffer sets the card to promiscuous mode which means all traffic is read, whether it is addressed to that machine or not. The figure below shows an attacker sniffing packets from the network, and the Wireshark packet sniffer/analyser (formerly known as ethereal).

**Packet Analysis**

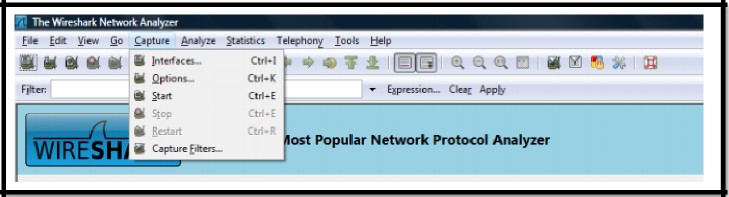
* + Wireshark is an open source cross-platform packet capture and analysis tool, with versions for Windows and Linux. The GUI window gives a detailed breakdown of the network protocol stack for each packet, colorizing packet details based on protocol, as well as having functionality to filter and search the traffic, and pick out TCP streams. Wireshark can also save packet datato files for offline analysis and export/import packet Captures to/from other tools. Statistics can also be generated for packet capture files.

**Using Wireshark to Capture Traffic**



[Fig : 6.2 Using Wireshark to Capture Traffic]

* + Start the Wireshark application. When Wireshark is first run, a default, or blank window is shown. To list the available network interfaces, select the CaptureInterfaces menu option.



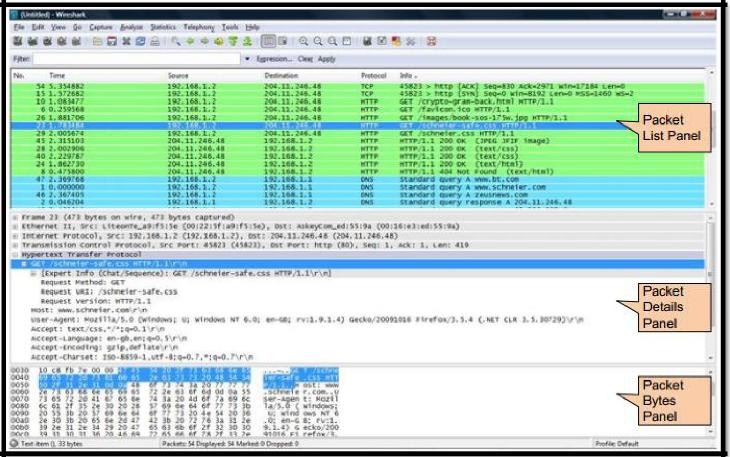
[Fig : 6.3the Wireshark application]

* + Wireshark should display a popup window such as the one shown in Figure. To capture network traffic click the Start button for the network interface you want to capture traffic on. Windows can have a long list of virtual interfaces, before the Ethernet Network Interface Card (NIC).



[Fig : 6.4 Wireshark Capture Interface]

* + Generate some network traffic with a Web Browser, such as Internet Explorer or Chrome. Your Wireshark window should show the packets, and now look something like.



[Fig : 6.5 Wireshark window]

* + To stop the capture, select the Capture->Stop menu option, Ctrl+E, or the Stop toolbar button. What you have created is a Packet Capture or pcap, which you can now view and analyse using the Wireshark interface, or save to disk to analyse later.

The capture is split into 3 parts:

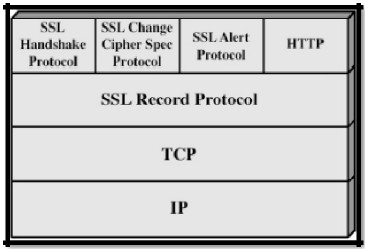
1. **Packet List Panel** – this is a list of packets in the current capture. It colours the packets based on the protocol type. When a packet is selected, the details are shown in the two panels below.
2. **Packet Details Panel** – this shows the details of the selected packet. It shows the different protocols making up the layers of data for this packet. Layers include Frame, Ethernet, IP, TCP/UDP/ICMP, and application protocols such as HTTP.
3. **Packet Bytes Panel** – shows the packet bytes in Hex and ASCII encodings.

**Practical – 7**

**7.Prepare a presentation and chart on SSL Protocol Stack.**

As mentioned, the Secure Sockets Layer (SSL) is a method for providing security for web based applications. It is designed to make use of TCP to provide a reliable end to-end secure service. SSL is not a single protocol but rather two layers of protocols as illustrated in figure. It can be seen that one layer makes use of TCP directly. This layer is known as the SSL Record Protocol and it provides basic security services to various higher layer protocols. An independent protocol that makes use of the record protocol is the Hypertext Markup Language (HTTP) protocol. Another three higher level protocols that also make use of this layer are part of the SSL stack. They are used in the management of SSL exchanges and are as follows:

1. Handshake Protocol.
2. Change Cipher Spec Protocol.
3. Alert Protocol.



[Fig : 7.1 SSL Block Diagram]

The SSL record protocol, which is at a lower layer and offers services to these three higher level protocols, is discussed first.

**SSL Record Protocol**

This protocol provides two services for SSL connections:

1. **Confidentiality** - using conventional encryption.
2. **Message Integrity** - using a Message Authentication Code (MAC).

In order to operate on data the protocol performs the following actions:

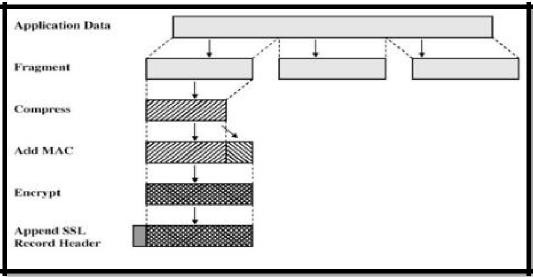
* + It takes an application message to be transmitted and fragments it into manageable blocks. These blocks are 214= 16, 384 bytes or less.

These blocks are then optionally compressed which must be lossless and may not increase the content length by more than 1024 bytes.

* + A message authentication code is then computed over the compressed data using a shared secret key. This is then appended to the compressed (or plaintext) block.
  + The compressed message plus MAC are then encrypted using symmetric encryption. Encryption may not increase the content length by more than 1024 bytes, so that the



* + total length may not exceed 214+ 2048. A number of different encryption algorithms are permitted.
  + The final step is to prepend a header, consisting of the following fields:



[Fig : 7.2]

* + Content type (8 bits) - The higher layer protocol used to process the enclosed fragment.
  + Major Version (8 bits) - Indicates major version of SSL in use. For SSLv3, the value is 3.
  + Minor Version (8 bits) - Indicates minor version in use. For SSLv3, the value is 0.
  + Compressed Length (16 bits) - The length in bytes of the compressed (or plaintext) fragment.



**Change Cipher Spec Protocol**

* + This consists of a single message which consists of a single byte with the value 1. Thisis used to cause the pending state to be copied into the current state which updates the cipher suite to be used on this connection.

**Alert Protocol**

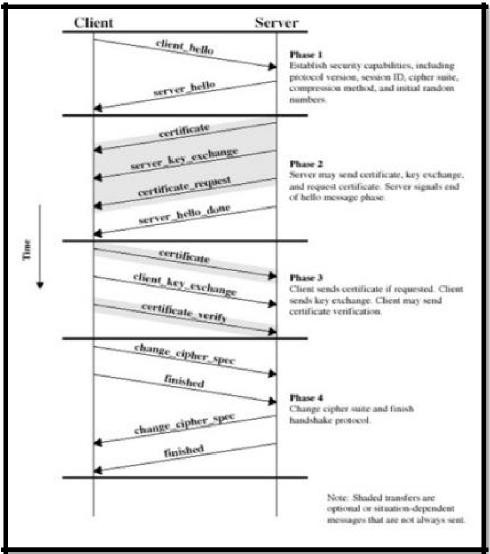
* + This protocol is used to convey SSL-related alerts to the peer entity. It consists of two bytes the first of which takes the values 1 (warning) or 2 (fatal). If the level is fatal SSL immediately terminates the connection. The second byte contains a code that indicates the specific alert. **Handshake Protocol**

This is the most complex part of SSL and allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. This protocol is used before any application data is sent. It consists of a series of messages exchanged by the client and server, all of which have the format shown in figure. Each message has three fields:

* + Type (1 byte): Indicates one of 10 messages such as “hello request”.  Length (3 bytes): The length of the message in bytes.



* + Content (≥ 0 byte): The parameters associated with this message such version of SSL being used.
  + The Handshake Protocol is shown in figure.



[Fig : 7.3 The Handshake Protocol]

**This consists of four phases:**

1. Establish security capabilities including protocol version, session ID, ciphersuite, compression method and initial random numbers. This phase consists of the client hello and server hello messages which contain the following (this is for the client however it‟s a little different for the server):

Version: The highest SSL version understood by client  Random: 32-bit timestamp and 28 byte nonce.

* + Session ID: A variable length session identifier.
  + Cipher Suite: List of crypto algorithms supported by client in decreasing order of preference. Both key exchange and Cipher Spec (this includes fields such as Cipher Algorithm, MAs Algorithm, Cipher Type, Hash Size, Key Material and IV Size) are defined.
  + Compression Method: List of methods supported by client

1. Server may send certificate, key exchange, and request certificate it also signals end of hello message phase. The certificate sent is one of a chain of X.509 certificates discussed earlier in the course. The server key exchange is sent only ifrequired. A certificate may be requested from the client if needs be by certificate request.
2. Upon receipt of the server done message, the client should verify that the server provided a valid certificate, if required, and check that the server hello parameters are acceptable. If all is satisfactory, the client sends one or more messages back to the server. The client sends certificate if requested (if none available then it sends a no certificate alert instead). Next the client sends client key exchange message. Finally, the client may send certificate verification.
3. Change cipher suite and finish handshake protocol. The secure connection is now setup and the client and server may begin to exchange application layer data.

**Practical – 8**

**8.Prepare a presentation on 3D authentication of monetary transaction.**

* + Secure electronic funds transfer is crucial to e-commerce.
  + Examination of how individuals and organizations conduct monetary transactions on the Internet.
  + Credit-card transactions, digital cash and e-wallets, smart cards, micro-payments and electronic bill presentment and payment.
  + Popular form of payment for online purchase.
  + Resistance due to security concerns.

**Many cards offer capabilities for online and offline purchases:**

**To accept credit-card payments, a merchant must have a merchant account.**

* + **Traditional merchant accounts accept only POS (point-of-sale) transactions:** Transactions that occur when you present your credit card at a store.
  + **Card-not-present (CNP) transaction:** Merchant does not see actual card being used in the purchase.
  + **Authentication:** The person is, in fact, who they say they are.
  + **Authorization:** The money is available to complete the transaction.
  + **Acquiring bank:** The bank with which the merchant holds an account.
  + **Issuing bank:** The bank from which the buyer obtained the credit card, and the credit-card association.

**Credit-Card Transaction Enablers:**

* + Companies that have established business relationships with financial institutions that will accept online credit-card payments for merchant clients.
  + iCat
  + Trintech
  + Cybercash
  + NextCard, Inc.
  + Enables businesses to receive payments through Internet.

**Credit Card Frauds:**

* + When a credit-card holder claims a purchase was made by an unauthorized individual, or when a purchase was not received.
  + The charges in question are not the responsibility of the credit-card holder.
  + On the Internet, neither a scan of the card nor a signature is registered nor is the cost incurred by the merchant.



[Fig : 8.1 ]

**Alternatives:**

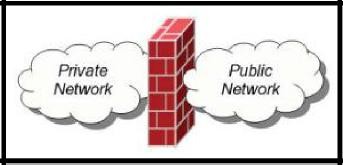
* + Cash on delivery (COD).
  + Debit cards :
  + Offer an alternative for card-holders to access their accounts.
  + Funds are instantly deducted from checking account.
  + Can withdraw cash from Automatic Teller Machines (ATMs).
  + Checking-account numbers :

* + Companies such as AmeriNet allow merchants to accept checking-account numbers as a valid form of payment.

**Practical – 9**

**9. (A) Install and configure few features of Firewall for network security .**

* + A firewall is an integrated collection of security measures designed to prevent unauthorized electronic access to a networked computer system.
  + A network firewall is similar to firewalls in building construction, because in both cases they are intended to isolate one "network" or "compartment" from another.



[Fig : 9.1 Firewall]

**Firewall Policies**

* + To protect private networks and individual machines from the dangers of the greater Internet, a firewall can be employed to filter incoming or outgoing traffic based on a predefined set of rules called firewall policies.

**Policy Actions**

**Packets flowing through a firewall can have one of three outcomes:**

* + Accepted: permitted through the firewall
  + Dropped: not allowed through with no indication of failure
  + Rejected: not allowed through, accompanied by an attempt to inform the source that the packet was rejected

**Policies used by the firewall to handle packets are based on several properties of the packets being inspected, including the protocol used, such as:**

* + TCP or UDP
  + the source and destination IP addresses
  + the source and destination ports
  + The application-level payload of the packet (e.g., whether it contains a Virus).

**The Need for Firewalls**

* + Information systems in corporations, government agencies, and other organizations have undergone a steady evolution. The following are notable developments:
  + Centralized data processing system, with a central mainframe supporting a number of directly connected terminals
  + Local area networks (LANs) interconnecting PCs and terminals to each other and the mainframe
  + Premises network, consisting of a number of LANs, interconnecting PCs, servers, and perhaps a mainframe or two
  + Enterprise-wide network, consisting of multiple, geographically distributed premises networks interconnected by a private wide area network (WAN)
  + Internet connectivity, in which the various premises networks all hook into the Internet and may or may not also be connected by a private WAN.
  + The firewall, then, provides an additional layer of defense, insulating the internal systems from external networks. This follows the classic military doctrine of “defense in depth,” which is just as applicable to IT security

**Firewall Characteristics**

**Lists the following design goals for a firewall:**

* + 1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible, as explained later in this chapter.
  + 2. Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies, as explained later in this chapter.
  + 3. The firewall itself is immune to penetration. This implies the use of a hardened system with a secured operating system. Trusted computer systems are suitable for hosting a firewall and often required in government applications.

**Scope of a firewall:**

* + 1. A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks. The use of a single choke point simplifies security management because security capabilities are consolidated on a single system or set of systems.
  + 2. A firewall provides a location for monitoring security-related events. Audits and alarms can be implemented on the firewall system.
  + 3. A firewall is a convenient platform for several Internet functions that are not security related. These include a network address translator, which maps local addresses to Internet addresses, and a network management function that audits or logs Internet usage.
  + 4. A firewall can serve as the platform for IPsec. Using the tunnel mode capability described in Chapter, the firewall can be used to implement virtual private networks.

**Firewalls have their limitations, including the following:**

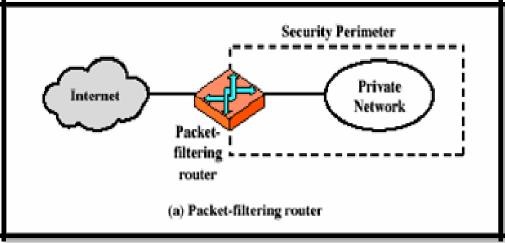
* + 1. The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-out capability to connect to an ISP. An internal LAN may support a modem pool that provides dial-in capability for traveling employees and telecommuters.
    2. The firewall may not protect fully against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.

* + 1. An improperly secured wireless LAN may be accessed from outside the organization. An internal firewall that separates portions of an enterprise network cannot guard against wireless communications between local systems on different sides of the internal firewall.

* + 1. A laptop, PDA, or portable storage device may be used and infected outside the corporate network, and then attached and used internally.

**Types of Firewalls**

* + A firewall may act as a packet filter. It can operate as a positive filter, allowing passing only packets that meet specific criteria, or as a negative filter, rejecting any packet that meets certain criteria. Depending on the type of firewall, it may examine one or more protocol headers in each packet, the payload of each packet, or the pattern generated by a sequence of packets. In this section, we look at the principal types of firewalls. **Packet Filtering Firewall** A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet (Figure 22.1b). The firewall is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on information contained in a network packet:
  + Source IP address: The IP address of the system that originated the IP packet (e.g., 192.178.1.1)
  + Destination IP address: The IP address of the system the IP packet is trying to reach (e.g., 192.168.1.2)
  + Source and destination transport-level address: The transport-level (e.g., TCP or UDP) port number, which defines applications such as SNMP or TELNET
  + IP protocol field: Defines the transport protocol
  + Interface: For a firewall with three or more ports, which interface of the firewall the packet came from or which interface of the firewall the packet is destined for.



[Fig : 9.2 Packet-Filtering Router]

**Lists the following weaknesses of packet filter firewalls:**

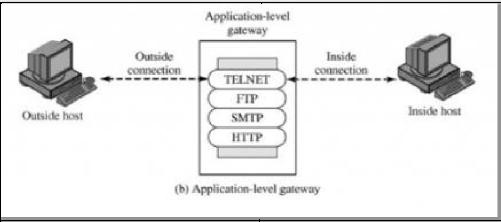
* + Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions. For example, a packet filter firewall cannot block specific application commands; if a packet filter
  + firewall allows a given application, all functions available within that application will be permitted.
  + Because of the limited information available to the firewall, the logging functionality present in packet filter firewalls is limited. Packet filter logs normally contain the same information used to make access control decisions (source address, destination address, and traffic type).
  + Most packet filter firewalls do not support advanced user authentication schemes. Once again, this limitation is mostly due to the lack of upper-layer functionality by the firewall.
  + Packet filter firewalls are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as network layer address spoofing. Many packet filter firewalls cannot detect a network packet in which the OSI Layer 3 addressing information has been altered. Spoofing attacks are generally employed by intruders to bypass the security controls implemented in a firewall platform.
  + Finally, due to the small number of variables used in access control decisions, packet filter firewalls are susceptible to security breaches caused by improper configurations. In other words, it is easy to accidentally configure a packet filter firewall to allow traffic types, sources, and destinations that should be denied based on an organization’s information security policy.

**Some of the attacks that can be made on packet filtering firewalls and the appropriate countermeasures are the following:**

* + **IP address spoofing:** The intruder transmits packets from the outside with a source IP address field containing an address of an internal host. The attacker hopes that the use of a spoofed address will allow penetration of systems that employ simple source address security, in which packets from specific trusted internal hosts are accepted. The countermeasure is to discard packets with an inside source address if the packet arrives on an external interface. In fact, this countermeasure is often implemented at the router external to the firewall.
  + **Source routing attacks:** The source station specifies the route that a packet should take as it crosses the Internet, in the hopes that this will bypass security measures that do not analyze the source routing information. The countermeasure is to discard all packets that use this option.
  + **Tiny fragment attacks:** The intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into a separate packet fragment. This attack is designed to circumvent filtering rules that depend on TCP header information. Typically, a packet filter will make a filtering decision on the first fragment of a packet. All subsequent fragments of that packet are filtered out solely on the basis that they are part of the packet whose first fragment was rejected. The attacker hopes that the filtering firewall examines only the first fragment and that the remaining fragments are passed through. A tiny fragment attack can be defeated by enforcing a rule that the first fragment of a packet must contain a predefined minimum amount of the transport header. If the first fragment is rejected, the filter can remember the packet and discard all subsequent fragments.

**Application-Level Gateway**

* + An application-level gateway, also called an application proxy, acts as a relay of application-level traffic. The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. Further, the gateway can be configured to support only specific features ofan application that the network administrator considers acceptable while denying all other features.

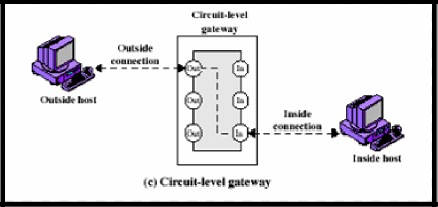


[Fig : 9.3 Application level gateway]

* + Application-level gateways tend to be more secure than packet filters. Rather than trying to deal with the numerous possible combinations that are to be allowed and forbidden at the TCP and IP level, the application-level gateway need only scrutinize a few allowable applications. In addition, it is easy to log and audit all incoming traffic at the application level.
  + A prime disadvantage of this type of gateway is the additional processing overhead on each connection. In effect, there are two spliced connections between the end users, with the gateway at the splice point, and the gateway must examine and forward all traffic in both directions.

**Circuit-Level Gateway**

* + A fourth type of firewall is the circuit-level gateway or circuit-level proxy. This can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications. As with an application gateway, a circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.



[Fig : 9.4 Circuit -level gateway]

* + A typical use of circuit-level gateways is a situation in which the system administrator trusts the internal users. The gateway can be configured to support application-level or proxy service on inbound connections and circuit-level functions for outbound connections. In this configuration, the gateway can incur the processing overhead of examining incoming application data for forbidden functions but does not incur that overhead on outgoing data.
  + An example of a circuit-level gateway implementation is the SOCKS package [KOBL92]; version 5 of SOCKS is specified in RFC 1928. The RFC defines SOCKS in the following fashion:
  + The protocol described here is designed to provide a framework for client-server applications in both the TCP and UDP domains to conveniently and securely use the services of a network firewall. The protocol is conceptually a “shim-layer” between the application layer and the transport layer, and as such does not provide network layer gateway services, such as forwarding of ICMP messages.

**SOCKS consist of the following components:**

* + The SOCKS server, which often runs on a UNIX-based firewall. SOCKS isalso implemented on Windows systems.
  + The SOCKS client library, which runs on internal hosts protected by the firewall.
  + SOCKS-ified versions of several standard client programs such as FTP and TELNET. The implementation of the SOCKS protocol typically involves either the recompilation or relinking of TCP-based client applications, or the use of alternate dynamically loaded libraries, to use the appropriate encapsulation routines in the SOCKS library.

**9. (B) Inspect the firewall at your department in CWN. Understand its functionality, identify the important configuration parameters for the same.**

* In our college we have FortiGate®-100D firewall.

**Basic information about FortiGate®-100D :**

* Today’s network security threats have evolved into highly sophisticated assaults usingmultipleattack vectors to penetrate networks and steal valuable information. Smalland mediumenterprises and remote branch offices of larger enterprises face theseattacks while storing andtransferring increasing amounts of sensitive data acrosstheir networks. To ensure the security of this sensitive data, legislation such as PCI,HIPAA, Sarbanes-Oxley, and others have warranted the implementation ofsecuritymeasures.

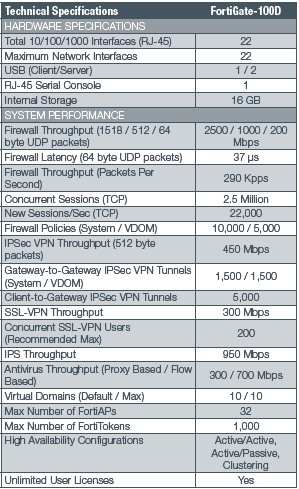
* In order to comply with legislation and secure the valuable data traversing networks,smalland medium enterprises and remote branch offices need a security solutionthat integrates multiple attack recognition technologies into a single device. Withlimited budgets and modest remote resources, these smaller networks desire a costeffective solution that is simple to install, connect and maintain. Just as importantly,networks are ever-expanding and need a solution that leaves them with room to growover time.

The FortiGate-100D is an ideal security solution for small and mediumenterprises or remote branch offices of larger networks. It combines firewall, IPSecand SSL VPN, application control, intrusion prevention, antivirus, antimalware,antispam, P2P security, and web filtering into a single device.

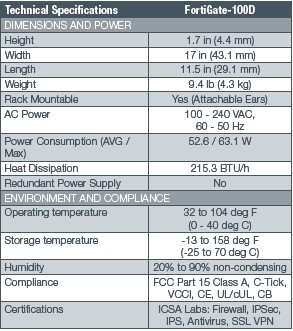
**Simple, Powerful, Secure**

* The FortiGate-100D installs in minutes, automatically downloading regular updates toprotectagainst the latest viruses, network vulnerabilities, worms, spam and phishingattacks, and malicious websites with no administrator intervention. Leveragingpatented FortiASIC acceleration, the FortiGate-100D offers marketleadingperformance, with twenty-two GbE interfaces that facilitate network growth andexpansion. Onboard storage provides local archiving of data for policy complianceand/or WAN Optimization. The WAN Optimization functionality increases network
* performance by reducing the amount of communication and data transmitted between applications and servers across a WAN.

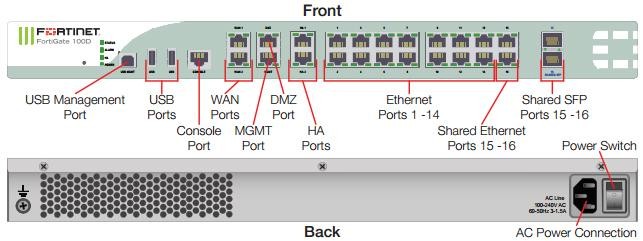
**Technical Specification of FortiGate®-100D :**



[Table : 9.1 Technical Specification of FortiGate®-100D ]



[Table : 9.2 Technical Specification of FortiGate®-100D ] **FortiGate®-100D :**



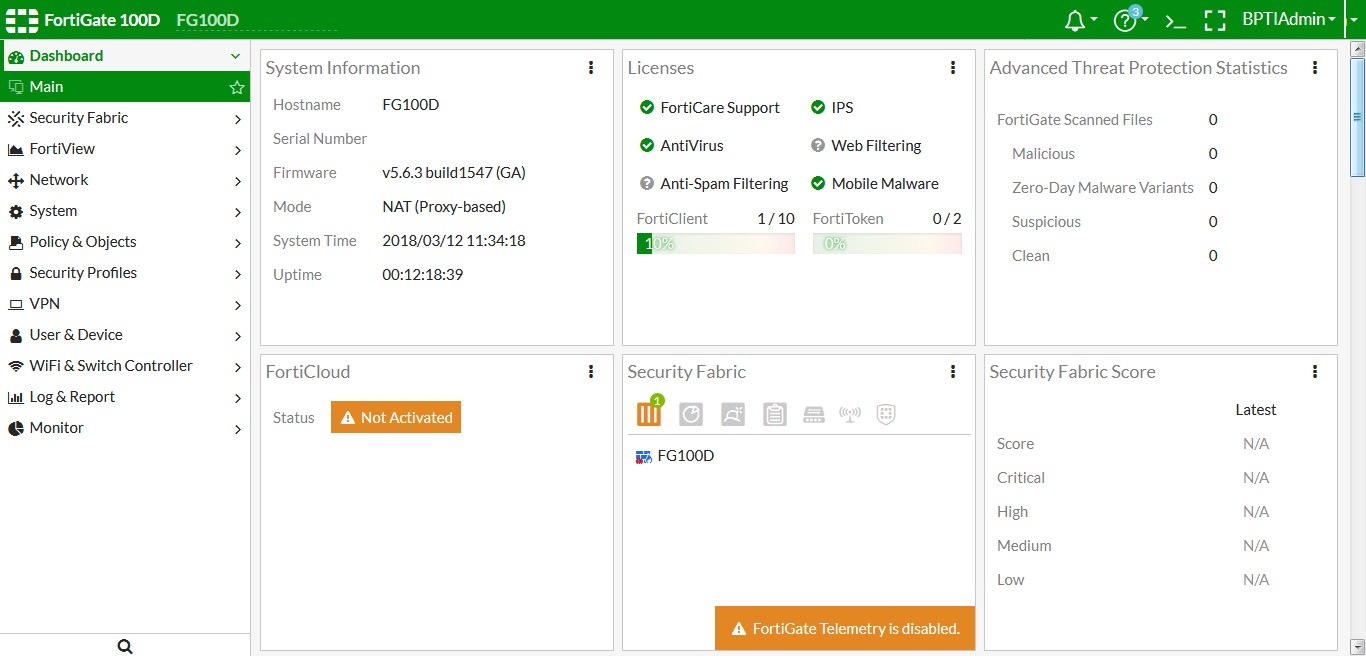
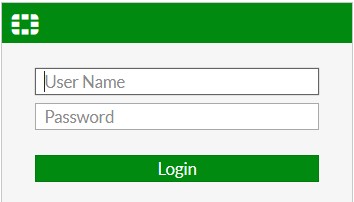
[Fig : 9(B).1 FortiGate®-100D Front Panel and Back Panel]

**Screenshots of our college on firewall. :**

[

Fig : 9(B).2 Login

]

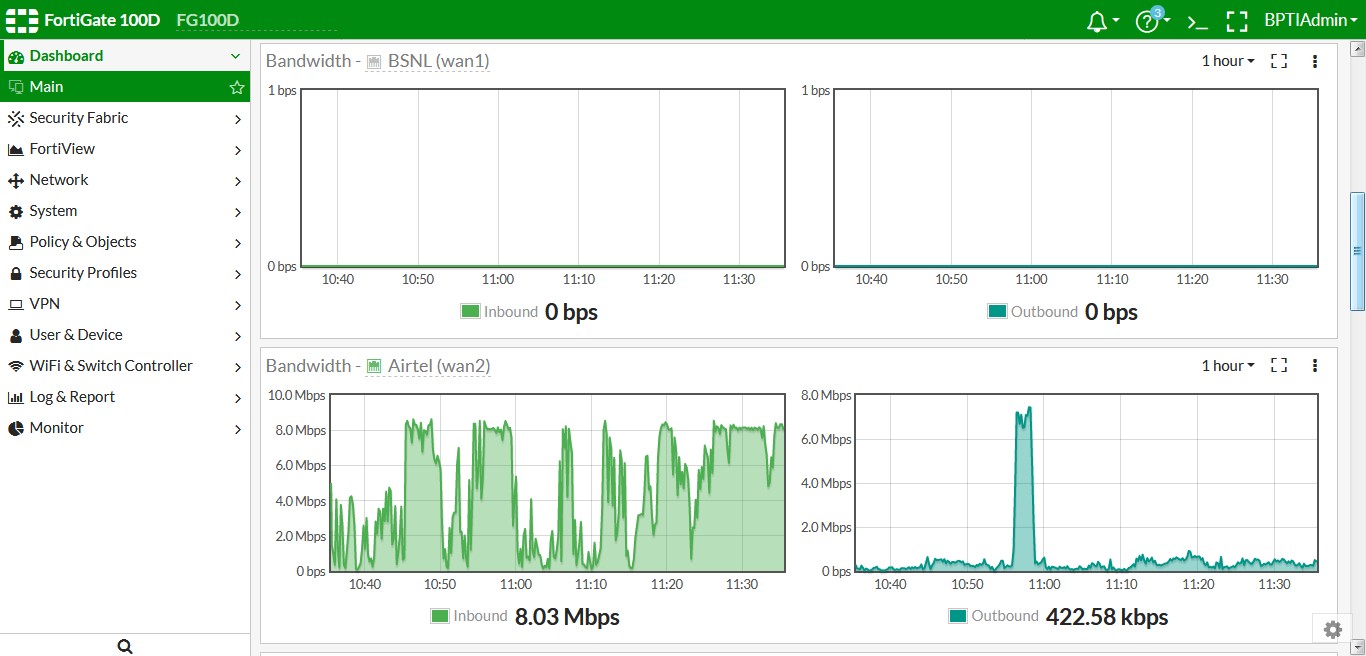


[Fig : 9(B).3 Dashboard]

[

Fig 9(B).4 Bandwidth Menegment

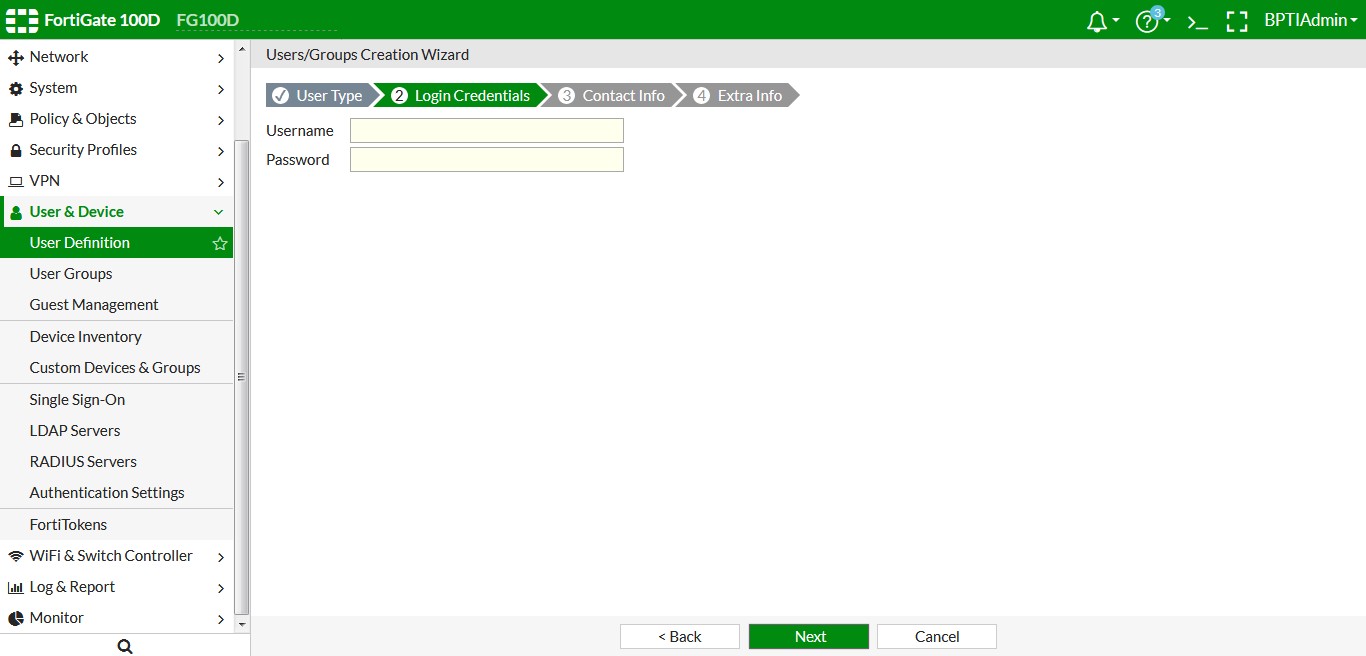
]



[

Fig : 9(B).5 User Creation

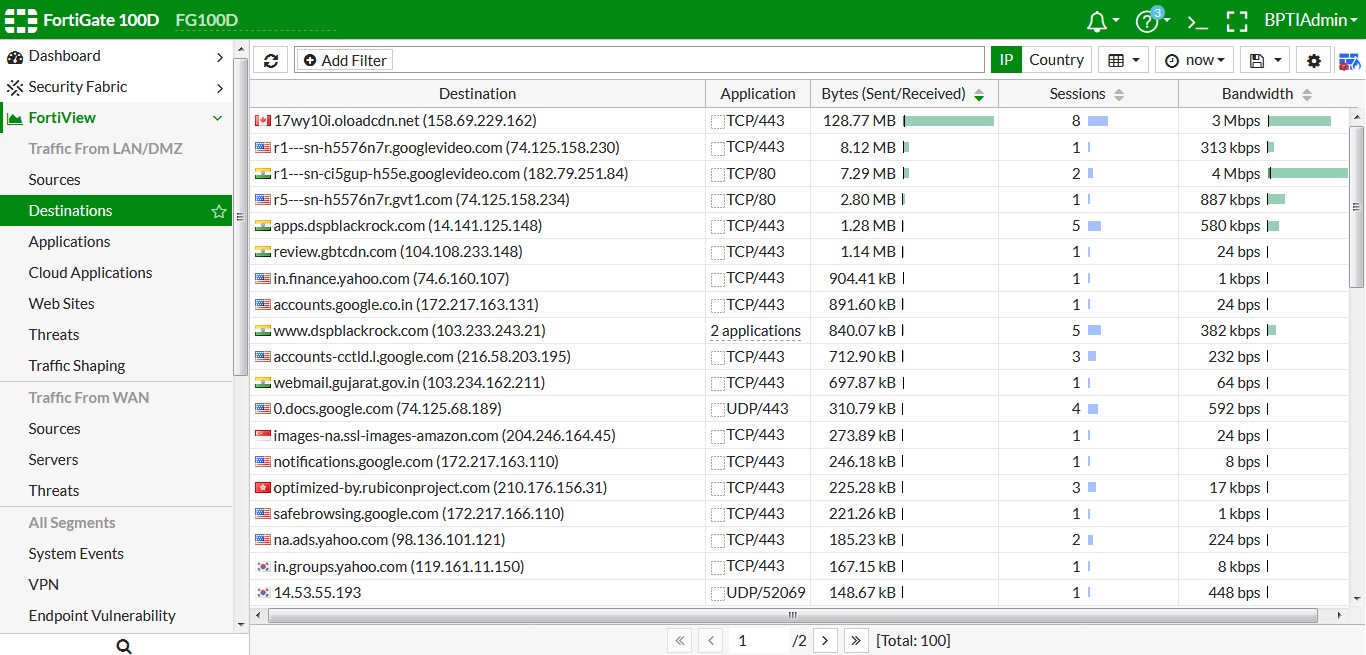
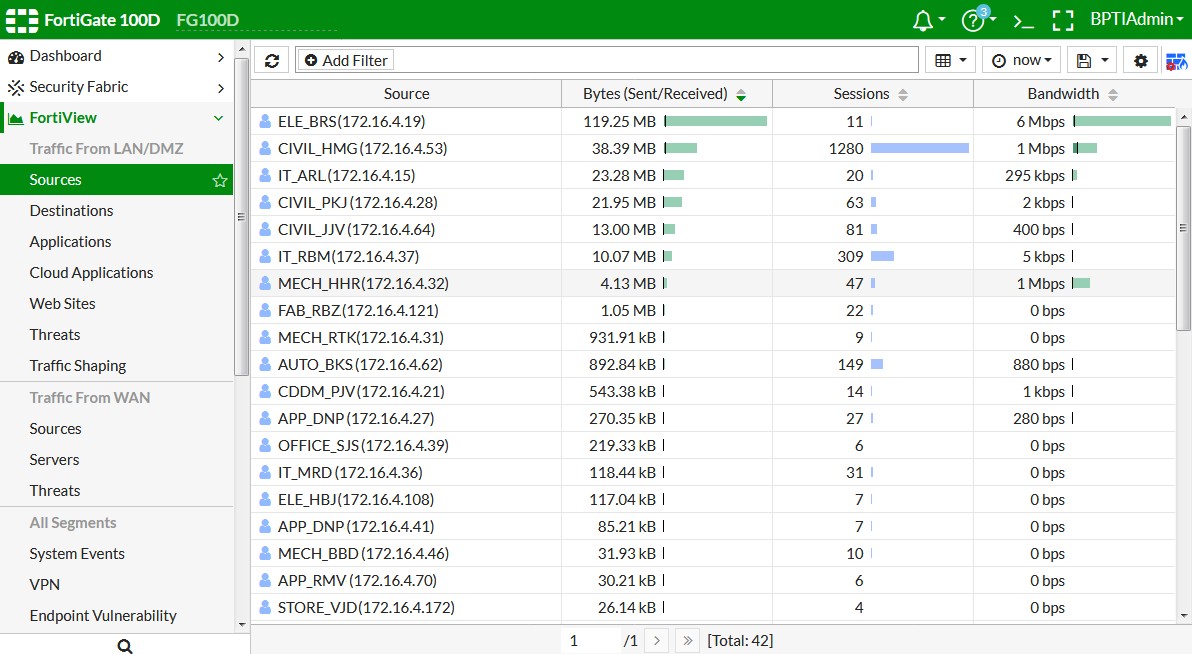
]



[

Fig : 9(B).6 Currunt

Users]



[Fig : 9(B).7 Accessed Destinations]

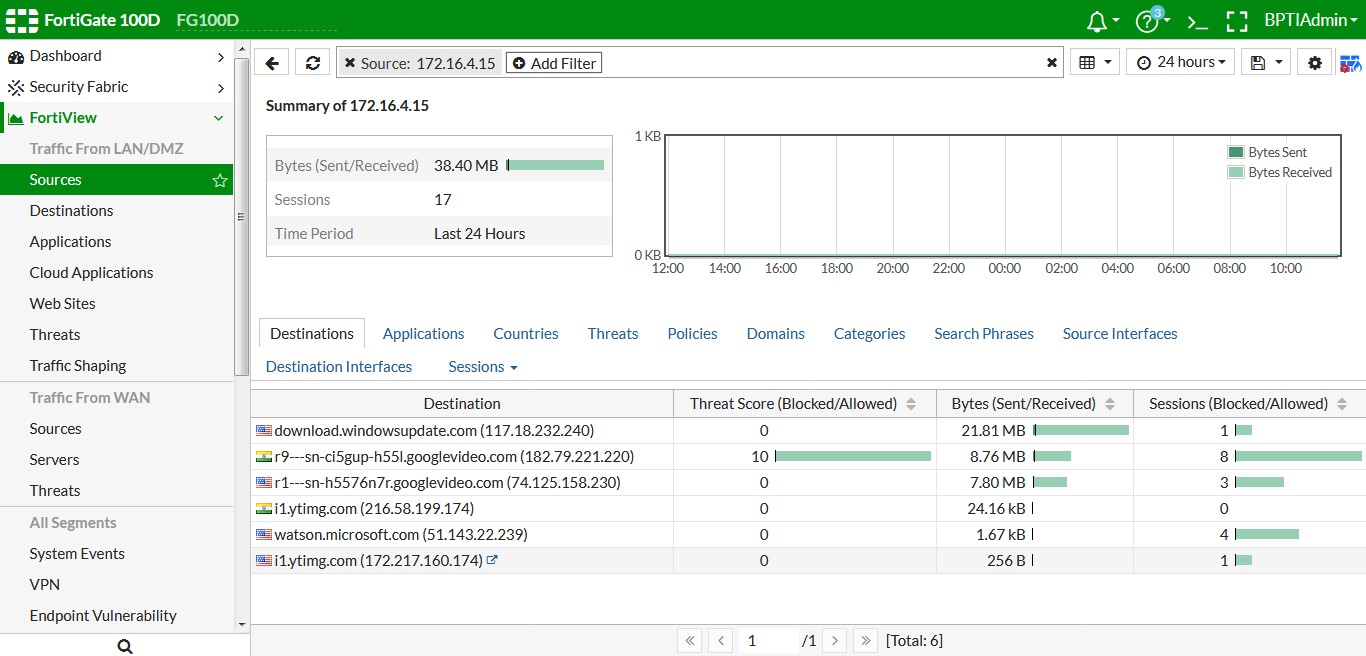
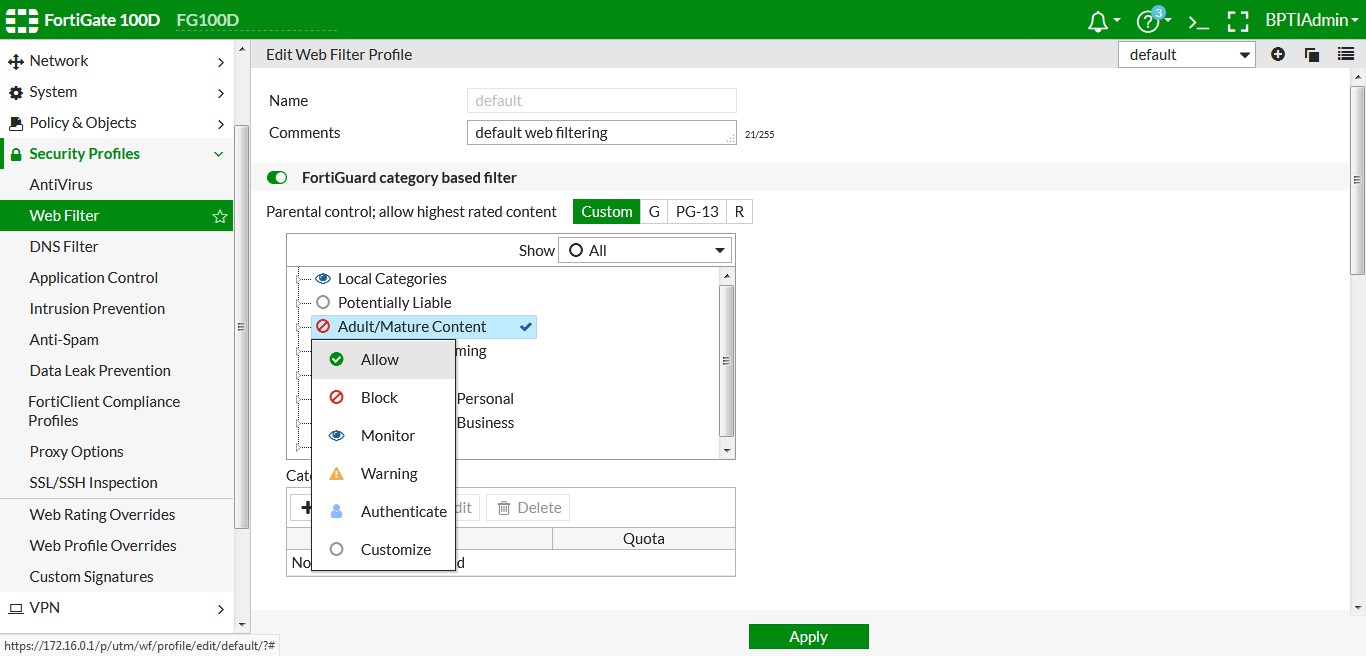
[

Fig. :

9(B).8

Web Sites Report

]



[Fig : 9(B).9 User wise Destinations Report]

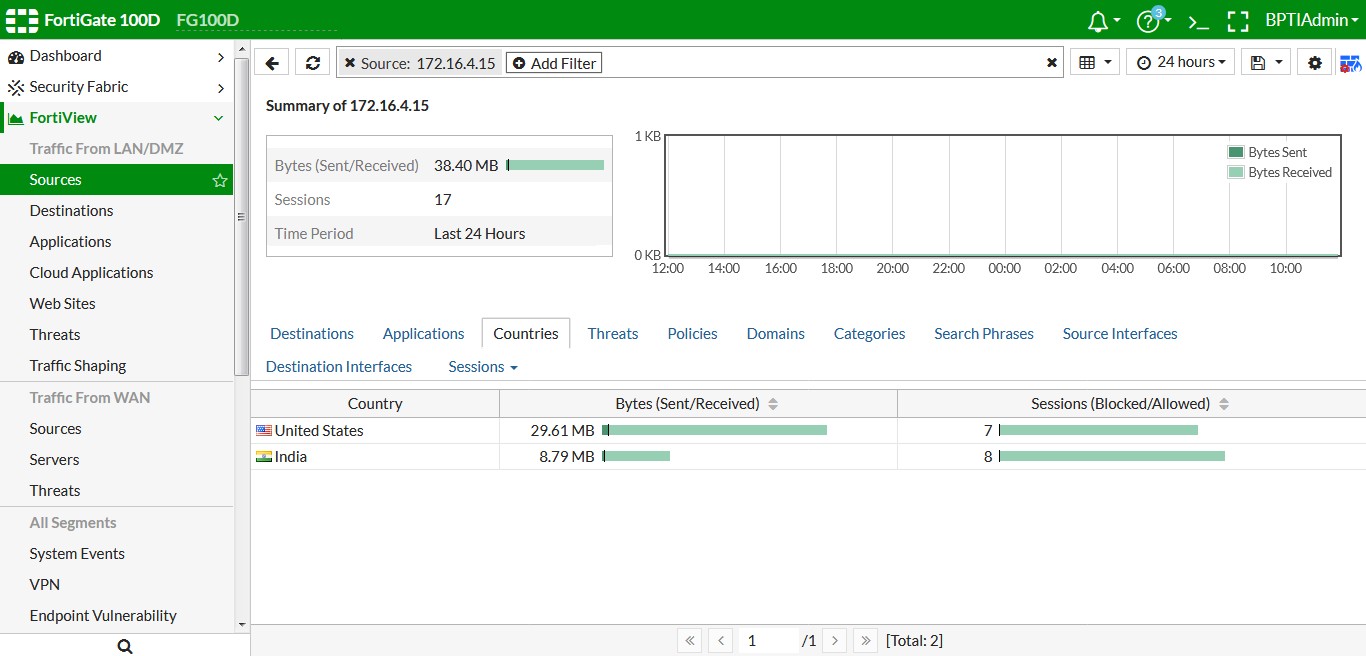
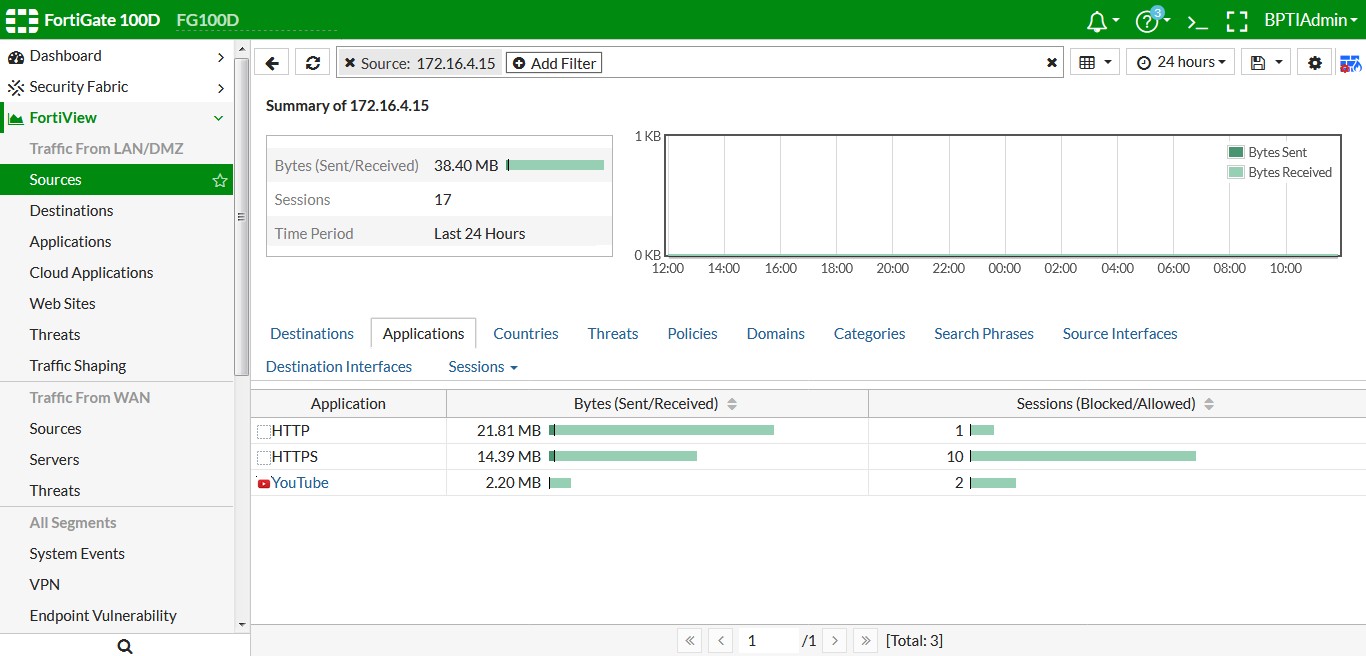
[

Fig

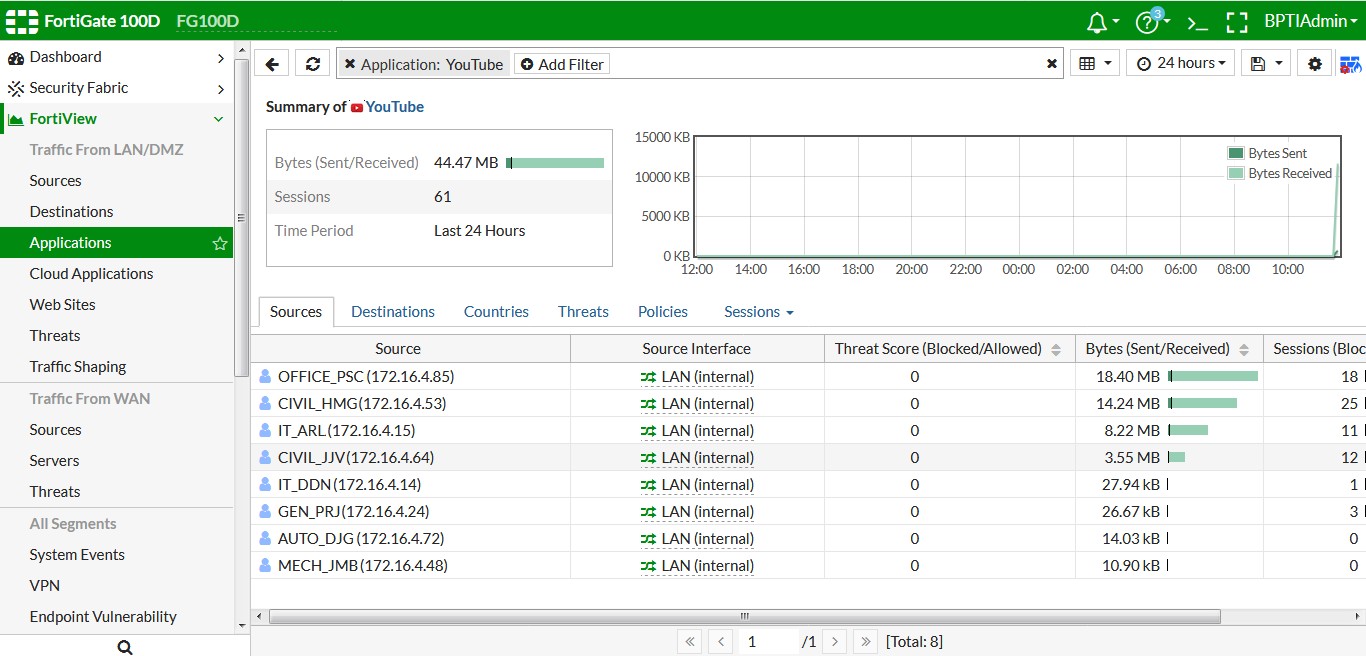
: 9(B).10

User wise Application Report

]



[Fig : 9(B).11 User wise Country Report]



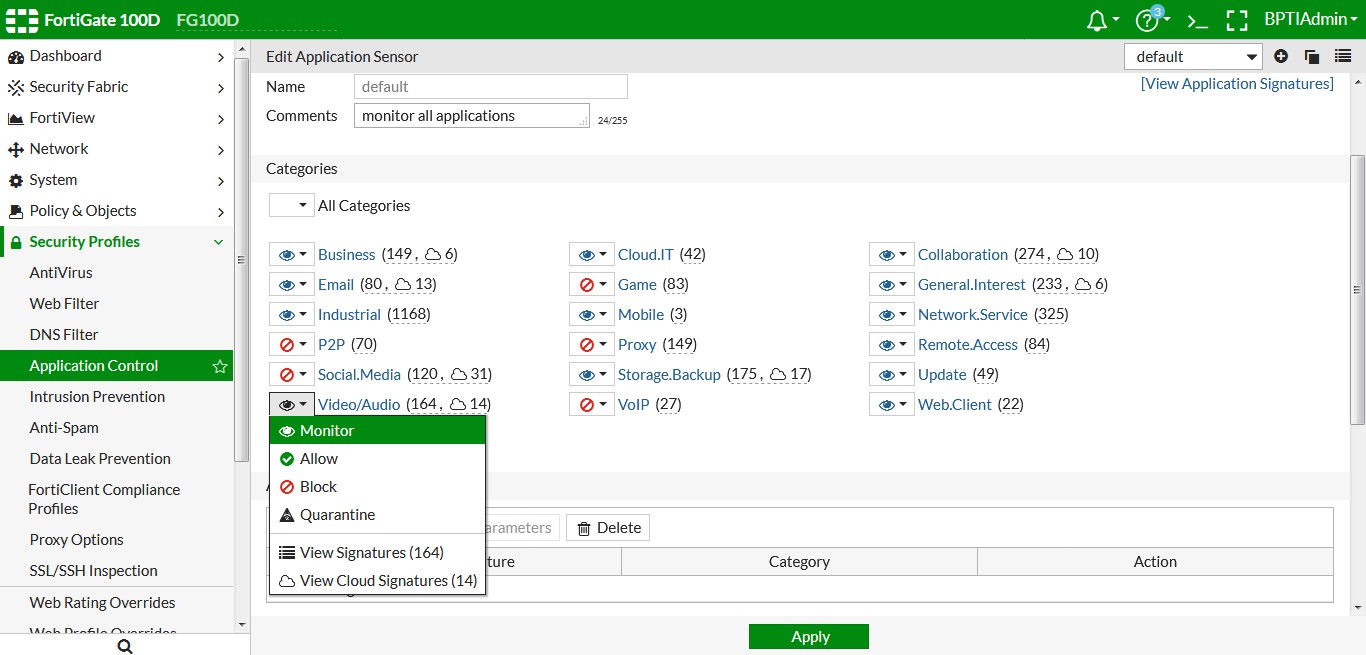
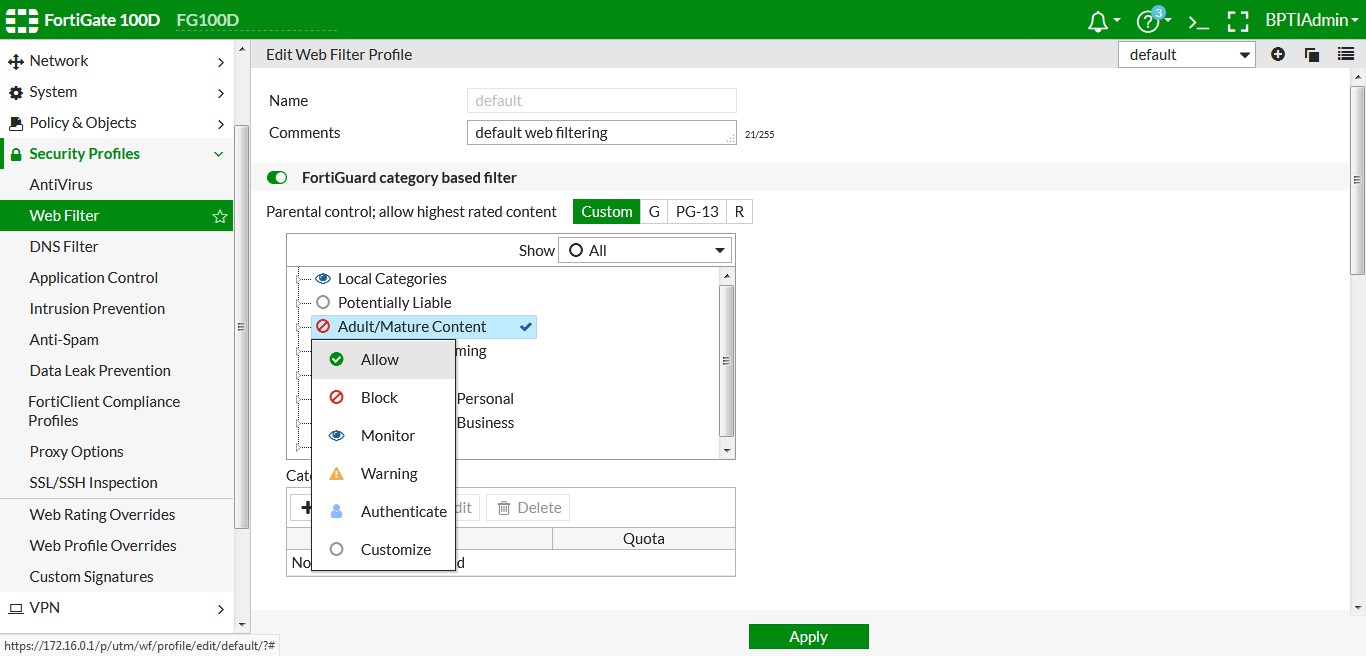
[Fig : 9(B).12 Application-Destination wise User Report]

[

Fig : 9(B).13

Web Filter Policy

]



[Fig. :9(B).14 Application Filter Policy]

# Practical – 10

**10. Chart on Wi-Fi Security.**

* Wi-Fi (Wireless Fidelity) is one of today’s leading wireless technologies, with Wi-Fi support being integrated into more and more devices: laptops, PDAs, mobile phones. However, one configuration aspect all too often goes unnoticed: security. Let's have a closer look at the level of security of encryption methods used in modern Wi-Fi implementations.
* Wireless security is the prevention of unauthorized access or damage to computers using wireless networks. The most common types of wireless security are Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA). WEP is a notoriously weak security standard. The password it uses can often be cracked in a few minutes with a basic laptop computer and widely available software tools. WEP is an old IEEE 802.11 standard from 1999, which was outdated in 2003 by WPA, or Wi-Fi Protected Access. WPA was a quick alternative to improve security over WEP. The current standard is WPA2; some hardware cannot support WPA2 without firmware upgrade or replacement. WPA2 uses an encryption device that encrypts the network with a 256-bit key; the longer key length improves security over WEP.
* Many laptop computers have wireless cards pre-installed. The ability to enter a network while mobile has great benefits. However, wireless networking is prone to some security issues. Hackers have found wireless networks relatively easy to break into, and even use wireless technology to hack into wired networks. As a result, it is very important that enterprises define effective wireless security policies that guard against unauthorized access to important resources. Wireless Intrusion Prevention Systems (WIPS) or Wireless Intrusion Detection Systems (WIDS) are commonly used to enforce wireless security policies.

**Modes of unauthorized access**

* The mode of unauthorized access to links, to functions and to data is as variable as the respective entities make use of program code. There does not exist a full scope model of such threat. To some extent the prevention relies on known modes and methods of attack and relevant methods for suppression of the applied methods. However, each new mode of operation will create new options of threatening. Hence prevention requires a steady drive for improvement. The described modes of attack are just a snapshot of typical methods and scenarios where to apply.

**Accidental association**

* Violation of the security perimeter of a corporate network can come from a number of different methods and intents. One of these methods is referred to as “accidental association”. When a user turns on a computer and it latches on to a wireless access point from a neighboring company’s overlapping network, the user may not even know that this has occurred. However, it is a security breach in that proprietary company information is exposed and now there could exist a link from one company to the other. This is especially true if the laptop is also hooked to a wired network.

**Malicious association**

* “Malicious associations” are when wireless devices can be actively made by attackers to connect to a company network through their laptop instead of a company access point (AP). These types of laptops are known as “soft APs” and are created when a cybercriminal runs some software that makes his/her wireless network card look like a legitimate access point.

**Ad hoc networks**

* Ad hoc networks can pose a security threat. Ad hoc networks are defined as [peer to peer] networks between wireless computers that do not have an access point in between them. While these types of networks usually have little protection, encryption methods can be used to provide security.

**Non-traditional networks**

* Non-traditional networks such as personal network Bluetooth devices are not safe from hacking and should be regarded as a security risk. Even barcode readers, handheld PDAs, and wireless printers and copiers should be secured. These non-traditional networks can be easily overlooked by IT personnel who have narrowly focused on laptops and access points.

**Identity theft (MAC spoofing)**

* Identity theft (or MAC spoofing) occurs when a hacker is able to listen in on network traffic and identify the MAC address of a computer with network privileges. Most wireless systems allow some kind of MAC filtering to allow only authorized computers with specific MAC IDs to gain access and utilize the network. However, programs exist that have network “sniffing” capabilities. Combine these programs with other software that allow a computer to pretend it has any MAC address that the hacker desires,[10] and the hacker can easily get around that hurdle.

**Man-in-the-middle attacks**

* A man-in-the-middle attacker entices computers to log into a computer which is set up as a soft AP (Access Point). Once this is done, the hacker connects to a real access point through another wireless card offering a steady flow of traffic through the transparent hacking computer to the real network. The hacker can then sniff the traffic. One type of man-in-the-middle attack relies on security faults in challenge and handshake protocols to execute a “de-authentication attack”.

**Denial of service**

* A Denial-of-Service attack (DoS) occurs when an attacker continually bombards a targeted AP (Access Point) or network with bogus requests, premature successful connection messages, failure messages, and/or other commands. These cause legitimate users to not be able to get on the network and may even cause the network to crash. These attacks rely on the abuse of protocols such as the Extensible Authentication Protocol (EAP).

**WEP, WPA, and WPA2: Wi-Fi Security through the Ages**

* Since the late 1990s, Wi-Fi security algorithms have undergone multiple upgrades with outright depreciation of older algorithms and significant revision to newer algorithms. A stroll through the history of Wi-Fi security serves to highlight both what‟s out there right now and why you should avoid older standards.



[Fig : 10.1 Wi-Fi]

**Wired Equivalent Privacy (WEP)**

* Wired Equivalent Privacy (WEP) is the most widely used Wi-Fi security algorithm in the world. This is a function of age, backwards compatibility, and the fact that it appears first in the encryption type selection menus in many router control panels.
* WEP was ratified as a Wi-Fi security standard in September of 1999. The first versions of WEP weren‟t particularly strong, even for the time they were released, because U.S. restrictions on the export of various cryptographic technologies led to manufacturers restricting their devices to only 64-bit encryption. When the restrictions were lifted, it was increased to 128-bit. Despite the introduction of 256-bit WEP encryption, 128-bit remains one of the most common implementations.
* Despite revisions to the algorithm and an increased key size, over time numerous security flaws were discovered in the WEP standard and, as computing power increased, it became easier and easier to exploit them. As early as 2001 proof-ofconcept exploits were floating around and by 2005 the FBI gave a public demonstration (in an effort to increase awareness of WEP‟s weaknesses) where they cracked WEP passwords in minutes using freely available software.
* Despite various improvements, work-around, and other attempts to shore up the WEP system, it remains highly vulnerable and systems that rely on WEP should be upgraded or, if security upgrades are not an option, replaced. The Wi-Fi Alliance officially retired WEP in 2004.

**Wi-Fi Protected Access (WPA)**

* Wi-Fi Protected Access was the Wi-Fi Alliance‟s direct response and replacement to the increasingly apparent vulnerabilities of the WEP standard. It was formally adopted in 2003, a year before WEP was officially retired. The most common WPA configuration is WPA-PSK (Pre-Shared Key). The keys used by WPA are 256-bit, a significant increase over the 64-bit and 128-bit keys used in the WEP system.
* Some of the significant changes implemented with WPA included message integrity checks (to determine if an attacker had captured or altered packets passed between the access point and client) and the Temporal Key Integrity Protocol (TKIP). TKIP employs a per- packet key system that was radically more secure than fixed key used in the WEP system. TKIP was later superseded by Advanced Encryption Standard (AES).
* Despite what a significant improvement WPA was over WEP, the ghost of WEP haunted WPA. TKIP, a core component of WPA, was designed to be easily rolled out via firmware upgrades onto existing WEP-enabled devices. As such it had to recycle certain elements used in the WEP system which, ultimately, were also exploited.
* WPA, like its predecessor WEP, has been shown via both proof-of-concept and applied public demonstrations to be vulnerable to intrusion. Interestingly the process by which WPA is usually breached is not a direct attack on the WPA algorithm (although such attacks have been successfully demonstrated) but by attacks on a supplementary system that was rolled out with WPA, Wi-Fi Protected Setup (WPS), designed to make it easy to link devices to modern access points.

**Wi-Fi Protected Access II (WPA2)**

* WPA has, as of 2006, been officially superseded by WPA2. One of the most significant changes between WPA and WPA2 was the mandatory use of AES algorithms and the introduction of CCMP (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol) as a replacement for TKIP (still preserved in WPA2 as a fallback system and for interoperability with WPA).
* Currently, the primary security vulnerability to the actual WPA2 system is an obscure one (and requires the attacker to already have access to the secured Wi-Fi network in order to gain access to certain keys and then perpetuate an attack against other devices on the network). As such, the security implications of the known WPA2 vulnerabilities are limited almost entirely to enterprise level networks and deserve little to no practical consideration in regard to home network security.
* Unfortunately, the same vulnerability that is the biggest hole in the WPA armor, the attack vector through the Wi-Fi Protected Setup (WPS), remains in modern WPA2capable access points. Although breaking into a WPA/WPA2 secured network using this vulnerability requires anywhere from 2-14 hours of sustained effort with a modern computer, it is still a legitimate security concern and WPS should be disabled (and, if possible, the firmware of the access point should be flashed to a distribution that doesn‟t even support WPS so the attack vector is entirely removed).

**How to Secure Your Wireless Network**

* The good news is that it is not very hard to make your wireless network secure, which will both prevent others from stealing your internet and will also prevent hackers from taking control of your computers through your own wireless network.

Here a few simple things that you should to secure your wireless network:

**Step 1. Open your router settings page**

* First, you need to know how to access your wireless router‟s settings. Usually you can do this by typing in “192.168.1.1” into your web browser, and then enter the correct user name and password for the router. This is different for each router, so first check your router’s user manual.

**Step 2. Create a unique password on your router**

* Once you have logged into your router, the first thing you should do to secure your network is to change the default password\* of the router to something more secure.

**Step 3. Change your Network’s SSID name**

* The SSID (or Wireless Network Name) of your Wireless Router is usually predefined as “default” or is set as the brand name of the router (e.g., linksys). Although this will not make your network inherently\* more secure, changing the SSID name of your network is a good idea as it will make it more obvious for others to know which network they are connecting to.

**Step 4. Enable Network Encryption**

* In order to prevent other computers in the area from using your internet connection, you need to encrypt your wireless signals.
* There are several encryption methods for wireless settings, including WEP, WPA (WPA-Personal), and WPA2 (Wi-Fi Protected Access version 2). WEP is basic encryption and therefore least secure (i.e., it can be easily cracked\*, but is compatible with a wide range of devices including older hardware, whereas WPA2 is the most secure but is only compatible with hardware manufactured since 2006.

**Step 5. Filter MAC addresses**

* Whether you have a laptop or a Wi-Fi enabled mobile phone, all your wireless devices have a unique MAC address (this has nothing to do with an Apple Mac) just like every computer connected to the Internet has a unique IP address. For an added layer of protection, you can add the MAC addresses of all your devices to your wireless router‟s settings so that only the specified devices can connect to your Wi-Fi network.

**Step 6. Reduce the Range of the Wireless Signal**

* If your wireless router has a high range but you are staying in a small studio apartment, you can consider decreasing the signal range by either changing the mode of your router to 802.11g (instead of 802.11n or 802.11b) or use a different wireless channel.

**Step 7. Upgrade your Router’s firmware**

* You should check the manufacturer’s site occasionally to make sure that your router is running the latest firmware. You can find the existing firmware version of your router using from the router’s dashboard at 192.168.\*.

**Connect to your Secure Wireless Network**

* To conclude, MAC Address filtering with WPA2 (AES) encryption (and a really complex passphrase) is probably the best way to secure your wireless network.
* Once you have enabled the various security settings in your wireless router, you need to add the new settings to your computers and other wireless devices so that they all can connect to the Wi-Fi network. You can select to have your computer automatically connect to this network, so you won’t have to enter the SSID, passphrase and other information every time you connect to the Internet.

Sign:

Date: