Practical: 1

Aim: List and practice various “net” commands on DOS & Linux.

The following are common Microsoft Windows network commands:

ipconfig
Ipconfig is a Console Command which can be issued to the Command Line Interpreter (or command prompt) to display the network settings currently assigned to any or all network adapters in the machine. This command can be utilised to verify a network connection as well as to verify your network settings.

netstat
Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, netstat displays active TCP connections.

tracert
The tracert command is used to visually see a network packet being sent and received and the amount of hops required for that packet to get to its destination.

ping

Helps in determining TCP/IP Networks IP address as well as determine issues with the network and assists in resolving them.

pathping
Provides information about network latency and network loss at intermediate hops between a source and destination. Pathping sends multiple Echo Request messages to each router between a source and destination over a period of time and then computes results based on the packets returned from each router.

telnet
Telnet is software that allows users to remotely access another computer such as a server, network device, or other computer. With telnet users can connect to a device or computer, manage a network device, setup a device, transfer files, etc.

ftp
FTP is short for File Transfer Protocol, this page contains additional information about the FTP command and help using that command in Unix and MS-DOS (Windows).

route
The function and syntax of the Windows ROUTE command is similar to the UNIX or Linux route command. Use the command to manually configure the routes in the routing table.

arp

Displays, adds, and removes arp information from network devices.

nslookup

Displays information that you can use to diagnose Domain Name System (DNS) infrastructure. Before using this tool, you should be familiar with how DNS works. The Nslookup command-line tool is available only if you have installed the TCP/IP protocol.

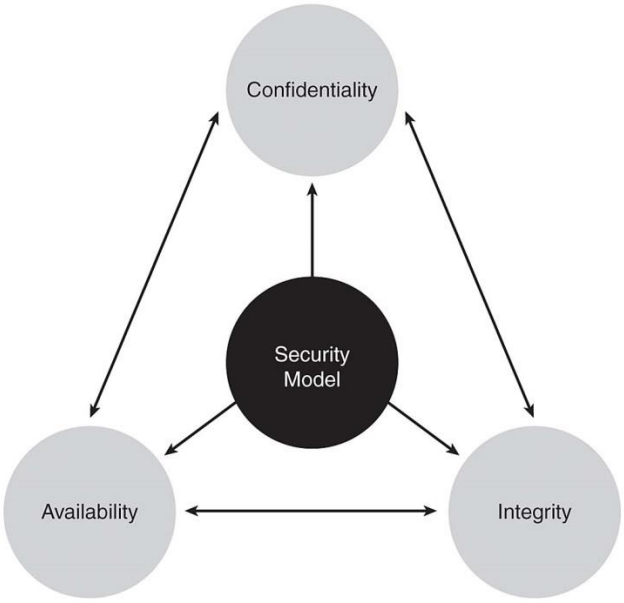
getmac

DOS command used to show both local and remote MAC addresses. When run with no parameters (ie. getmac) it displays MAC addresses for the local system. When run with the /s parameter (eg. getmac /s \\foo) it displays MAC addresses for the remote computer. When the /v parameter is used, it also displays the associated connection name and network adapter name.

Practical: 2

Aim: Draw the diagram for Confidentiality, Integrity & Availability.

- **Confidentiality** – ensures that sensitive information are accessed only by an authorized person and kept away from those not authorized to possess them. It is implemented using security mechanisms such as usernames, passwords, access control lists (ACLs), and encryption. It is also common for information to be categorized according to the extent of damage that could be done should it fall into unintended hands. Security measures can then be implemented accordingly.



- **Integrity** – ensures that information are in a format that is true and correct to its original purposes. The receiver of the information must have the information the creator intended him to have. The information can be edited by authorized persons only and remains in its original state when at rest. Integrity is implemented using security mechanism such as data encryption and hashing. Note that the changes in data might also occur as a result of non-human-caused events such as an electromagnetic pulse (EMP) or server crash, so it's important to have the backup procedure and redundant systems in place to ensure data integrity.

- **Availability** – ensures that information and resources are available to those who need them. It is implemented using methods such as hardware maintenance, software patching and network optimization. Processes such as redundancy, failover, RAID and high-availability clusters are used to mitigate serious consequences when hardware issues do occur. Dedicated hardware devices can be used to guard against downtime and unreachable data due to malicious actions such as distributed denial-of-service (DDoS) attacks.

Practical: 3

Aim: Configure Web browser security settings

Optimizing your browser's settings is a critical step in using the Internet securely and privately. Today's popular browsers include built-in security features, but users often fail to optimize their browser's security settings on installation. Failing to correctly set up your browser's security features can put you at a higher risk for malware infections and malicious attacks. This installation of our "Cybersecurity 101" series provides our tips for securing several of today's most popular browsers, including Google Chrome, Mozilla Firefox, and Microsoft Internet Explorer. While it is impossible to guarantee complete protection from cyber threats, following these tips will greatly increase the security of your web browser.

Secure Browsing with Google Chrome

These settings can be accessed through Chrome's "Advanced Settings" menu or by navigating to "chrome://settings/."

- **Enable phishing and malware protection:** Make sure that Chrome's phishing and malware protection feature is enabled under the "Privacy" section. This feature will warn you if a site you're trying to visit may be phishing or contain malware.

- **Turn off instant search:** The Instant search feature should be turned off for optimal security. While it offers some convenience in searching, having this feature enabled means that anything you type in the address bar is instantly sent to Google.

- **Don't sync:** Disconnect your email account from your browser under the "Personal Stuff" tab. Syncing your email account with your Chrome browser means that personal information such as passwords, autofill data, preferences, and more is stored on Google's servers. If you must use sync, select the "Encrypt all synced data" option and create a unique passphrase for encryption.

- **Configure content settings:** Click "Content settings" under the "Privacy" section and do the following:

    - *Cookies:* Select "Keep local data only until I quit my browser" and "Block third-party cookies and site data." These options ensure that your cookies will be deleted upon quitting Chrome and that advertisers will not be able to track you using third-party cookies.
    - *JavaScript:* Select "Do not allow any site to run JavaScript." It is widely recommended that JavaScript be disabled whenever possible to protect users from its security vulnerabilities.
    - *Pop-ups:* Select "Do not allow any site to show pop-ups.
    - *Location:* Select "Do not allow any site to track my physical location."

- **Configure passwords and forms settings:** Disable Autofill and deselect "Offer to save passwords I enter on the web" under the "Passwords and forms" section. Doing

so will prevent Chrome from saving your logins, passwords, and other sensitive information that you enter into forms.

Secure Browsing with Mozilla Firefox

These settings can be accessed through the "Options" menu.

- **Configure privacy settings:** Under the "Privacy" tab, complete the following steps. These measures ensure that Firefox is storing only as much of your information as it needs to function normally.
  - Select "Use custom settings for history."
  - Deselect "Remember my browsing and download history."
  - Deselect "Remember search and form history."
  - Deselect "Accept third-party cookies."
  - Set cookie storage to "Keep until I close Firefox."
  - Select "Clear history when Firefox closes."
- **Configure security settings:** Under the "Security" tab, choose the following settings. These steps prevent Firefox from saving your passwords and keep you from visiting potentially harmful sites.
  - Verify that "Warn me when sites try to install add-ons," "Block reported attack sites," and "Block reported web forgeries" are all selected.
  - Deselect "Remember passwords for sites."
- **Disable javaScript:** Deselect "Enable JavaScript" under the "Content" tab. JavaScript is notorious for containing security vulnerabilities and it is recommended that users only enable it for trusted sites.
- **Enable pop-up blocking:** Verify that "Block pop-up windows" is selected under the "Content" tab. This feature should be turned on by default as it protects users from unwarranted advertisements and windows.
- **Don't sync:** Avoid using Firefox Sync. By doing so you prevent Firefox from storing your logins, passwords, and other sensitive information.
- **Turn on automatic updates:** Verify that "Automatically install updates" is selected in the "Update" tab under "Advanced." Doing so will ensure that your browser receives critical security updates. Verify that "Automatically update Search Engines" is selected as well.
- **Use secure protocols:** Verify that "Use SSL 3.0" and "Use TLS 1.0" are selected in the "Encryption" tab under "Advanced."

Practical: 4

Aim: write ceaser's cipher algorithm & solve various examples based on encryption & decryption.

The Caesar Cipher technique is one of the earliest and simplest method of encryption technique. It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter some fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.

Thus to cipher a given text we need an integer value, known as shift which indicates the number of position each letter of the text has been moved down.

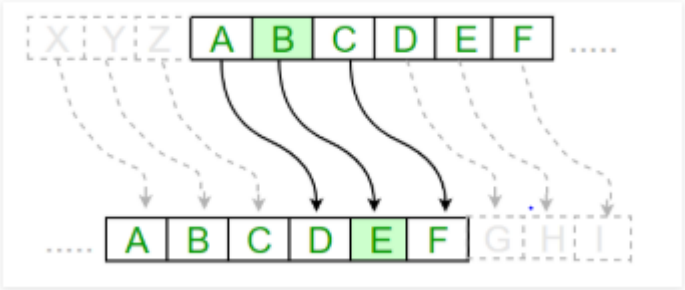
The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically as.

$$E_n(x) = (x + n) mod\ 26$$

(Encryption Phase with shift n)

$$D_n(x) = (x - n) mod\ 26$$

(Decryption Phase with shift n)



Examples:

Text : ABCDEFGHIJKLMNOPQRSTUVWXYZ

Shift: 23

Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

Text : ATTACKATONCE

Shift: 4

Cipher: EXXEGOEXSRGI

Algorithm for Caesar Cipher:

Input:

1. A String of lower case letters, called Text.
2. An Integer between 0-25 denoting the required shift.

Procedure:

Traverse the given text one character at a time.
For each character, transform the given character as per the rule, depending on whether we're encrypting or decrypting the text.
Return the new string generated.

Practical: 5

Aim: write test & debug ceaser cipher algorithm.

```
#include <iostream>
using namespace std;

// This function receives text and shift and
// returns the encrypted text
string encrypt(string text, int s)
{
    string result = "";

    // traverse text
    for (int i=0;i<text.length();i++)
    {
        // apply transformation to each character
        // Encrypt Uppercase letters
        if (isupper(text[i]))
            result += char(int(text[i]+s-65)%26 +65);

    // Encrypt Lowercase letters
    else
        result += char(int(text[i]+s-97)%26 +97);
    }

    // Return the resulting string
    return result;
}


int main()
{
    string text="ATTACKATONCE";
    int s = 4;
    cout << "Text : " << text;
    cout << "\nShift: " << s;
    cout << "\nCipher: " << encrypt(text, s);
    return 0;
}
```

Output:

```
Text : ATTACKATONCE

Shift: 4

Cipher: EXXEGOEXSRGI
```

Practical: 6

Aim: Write algorithm for shift cipher & solve various examples on it.

Shift Ciphers work by using the modulo operator to encrypt and decrypt messages. The Shift Cipher has a key K**, which is an** integer from 0 to 25**. We will only share this key with people that we want to see our message.**

How to Encrypt:

For every letter in the message **M** :

1. Convert the letter into the number that matches its order in the alphabet starting from 0, and call this number **X**.

(A=0, B=1, C=2, ...,Y=24, Z=25)

2. Calculate: **Y** = **(X + K)** mod **26**

3. Convert the number **Y** into a letter that matches its order in the alphabet starting from 0.

(A=0, B=1, C=2, ...,Y=24, Z=25)

For Example: We agree with our friend to use the Shift Cipher with **key K=19**for our message.

We encrypt the message **"KHAN",** as follows:

ENCRYPTION

	K	H	A	N	
	10	7	0	13	
+	19	19	19	19	
(	29	26	19	32	) mod 26
	3	0	19	6	
	D	A	T	G	

So, after applying the Shift Cipher with key K=19 our message text "**KHAN**" gave us **cipher text** "**DATG**".

We give the message "DATG" to our friend.

How to decrypt:

For every letter in the cipher text **C** :

1. Convert the letter into the number that matches its order in the alphabet starting from 0, and call this number Y.
(A=0, B=1, C=2, ..., Y=24, Z=25)

2. Calculate: **X= (Y - K)** mod **26**

3. Convert the number **X** into a letter that matches its order in the alphabet starting from 0.
(A=0, B=1, C=2, ..., Y=24, Z=25)

Our friend now decodes the message using our agreed upon **key K=19.** As follows:

DECRYPTION

	D	A	T	G	
	3	0	19	6	
-	19	19	19	19	
(	-16	-19	0	-13	) mod 26
	10	7	0	13	
	K	H	A	N	

So, after decrypting the Shift Cipher with key K=19 our friend deciphers the cipher text "**DATG**" into the message text "**KHAN**".

Practical: 7

Aim: Write algorithm for Hill cipher & solve various examples on it.

Hill cipher is a polygraphic substitution cipher based on linear algebra.Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.

The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26).

Input : Plaintext: ACT

Key: GYBNQKURP

Output : Ciphertext: POH

Input : Plaintext: GFG

Key: HILLMAGIC

Output : Ciphertext: SWK

Encryption

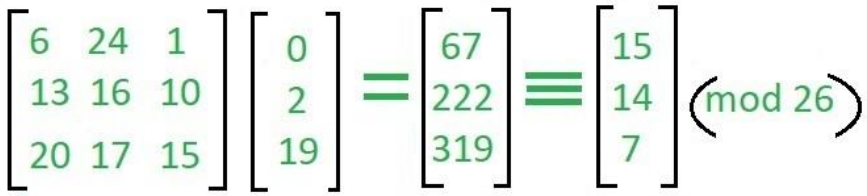
We have to encrypt the message ‘ACT’ (n=3).The key is ‘GYBNQKURP’ which can be written as the nxn matrix:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

The message ‘ACT’ is written as vector:

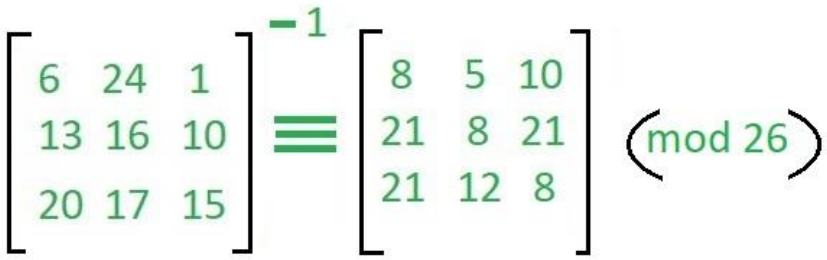
$$\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}$$

The enciphered vector is given as:

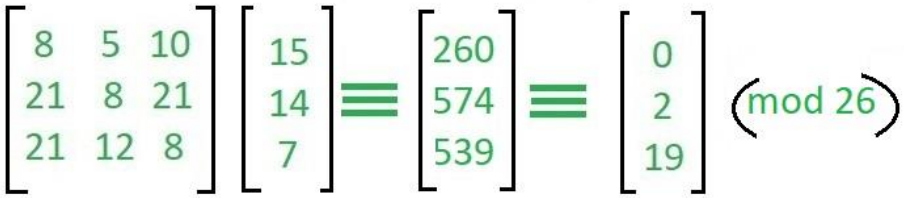


which corresponds to ciphertext of 'POH'

Decryption
To decrypt the message, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix (IFKVIVVMI in letters).The inverse of the matrix used in the previous example is:



For the previous Ciphertext 'POH':



which gives us back 'ACT'.
Assume that all the alphabets are in upper case.
Below is the the implementation of the above idea for n=3.

https://www.geeksforgeeks.org/hill-cipher/

Practical: 8

Aim: Write algorithm for Play fair cipher & solve various examples on it.

In this scheme, pairs of letters are encrypted, instead of single letters as in the case of simple substitution cipher.

In playfair cipher, initially a key table is created. The key table is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table as we need only 25 alphabets instead of 26. If the plaintext contains J, then it is replaced by I.

The sender and the receiver deicide on a particular key, say ‘tutorials’. In a key table, the first characters (going left to right) in the table is the phrase, excluding the duplicate letters. The rest of the table will be filled with the remaining letters of the alphabet, in natural order. The key table works out to be −

T	U	O	R	I
A	L	S	B	C
D	E	F	G	H
K	M	N	P	Q
V	W	X	Y	Z

Process of Playfair Cipher

- First, a plaintext message is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter. Let us say we want to encrypt the message “hide money”. It will be written as −

  HI DE MO NE YZ

- The rules of encryption are −
  - If both the letters are in the same column, take the letter below each one (going back to the top if at the bottom)

T	U	O	R	I
A	L	S	B	C
D	E	F	G	H
K	M	N	P	Q
V	W	X	Y	Z

‘H’ and ‘I’ are in same column, hence take letter below them to replace. HI → QC

If both letters are in the same row, take the letter to the right of each one (going back to the left if at the farthest right)

T	U	O	R	I	'D' and 'E' are in same row, hence take letter to the right of them to replace. DE → EF
A	L	S	B	C	
D	E	F	G	H	
K	M	N	P	Q	
V	W	X	Y	Z	

If neither of the preceding two rules are true, form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

T	U	O	R	I	'M' and 'O' nor on same column or same row, hence form rectangle as shown, and replace letter by picking up opposite corner letter on same row MO -> NU
A	L	S	B	C	
D	E	F	G	H	
K	M	N	P	Q	
V	W	X	Y	Z	

Using these rules, the result of the encryption of 'hide money' with the key of 'tutorials' would be −

QC EF NU MF ZV

Decrypting the Playfair cipher is as simple as doing the same process in reverse. Receiver has the same key and can create the same key table, and then decrypt any messages made using that key.

Practical: 10

Aim: Write algorithm for Vignere cipher & solve various examples on it.

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets .The encryption of the original text is done using the *Vigenère square or Vigenère table*.

- The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
- At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
- The alphabet used at each point depends on a repeating keyword.

Example:

Input : Plaintext : GEEKSFORGEEKS

Keyword : AYUSH

Output : Ciphertext : GCYCZFMLYLEIM

For generating key, the given keyword is repeated

in a circular manner until it matches the length of

the plain text.

The keyword "AYUSH" generates the key "AYUSHAYUSHAYU"

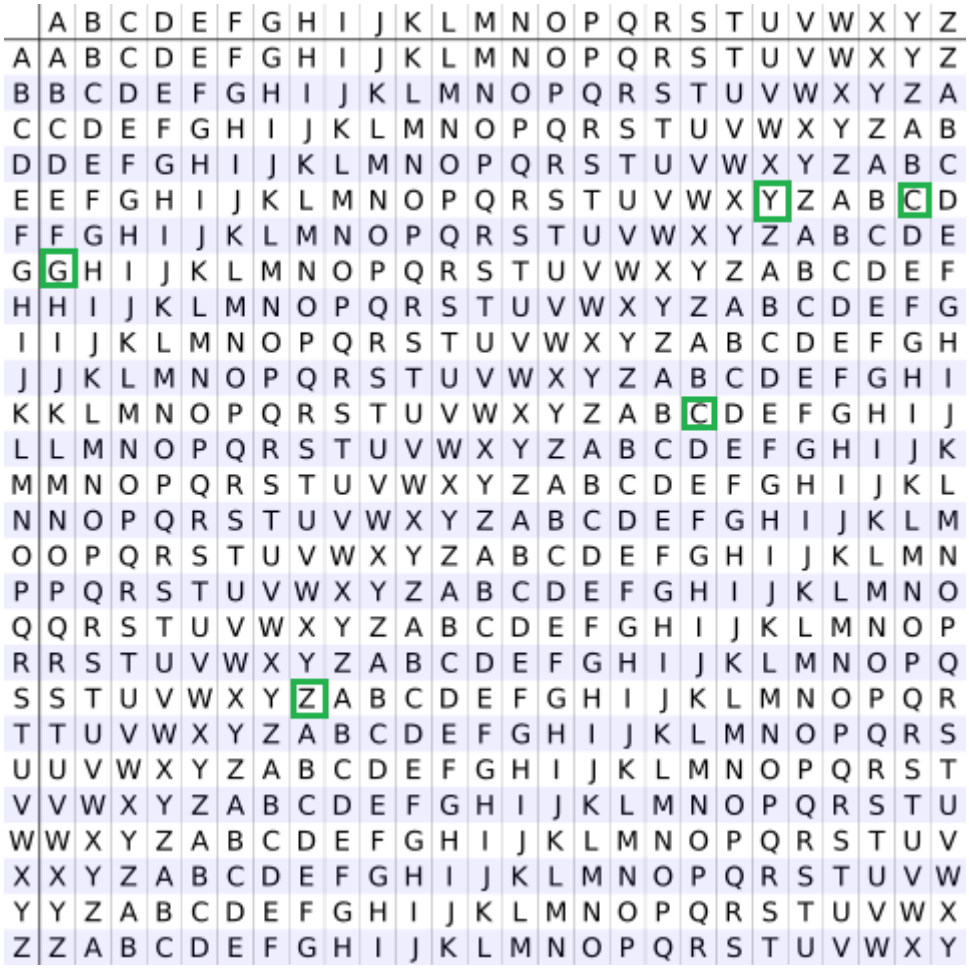
The plain text is then encrypted using the process

explained below.

Encryption

The first letter of the plaintext, G is paired with A, the first letter of the key. So use row G and column A of the Vigenère square, namely G. Similarly, for the second letter of the plaintext, the second letter of the key is used, the letter at row E and column Y is C. The rest of the plaintext is enciphered in a similar fashion.

Table to encrypt



Decryption

Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column's label as the plaintext. For example, in row A (from AYUSH), the ciphertext G appears in column G, which is the first plaintext letter. Next we go to row Y (from AYUSH), locate the ciphertext C which is found in column E, thus E is the second plaintext letter.

A more **easy implementation** could be to visualize Vigenère algebraically by converting [A-Z] into numbers [0–25].

Encryption

The the plaintext(P) and key(K) are added modulo 26.

$E_i = (P_i + K_i) \bmod 26$

Decryption

$D_i = (E_i - K_i + 26) \bmod 26$

https://www.geeksforgeeks.org/vigenere-cipher/

Practical: 11

Aim: Write algorithm for one time pad & solve various examples on it.

A one-time pad is the only (theoretically) 100% secure method of encryption currently available today. The one-time pad's security comes from it's key; the key (if chosen correctly) is EQUAL to the length of the plaintext and is COMPLETELY random. When both conditions are simultaneously fulfilled, the key is cryptographically secure.

The key destruction ensures that the key will not be reused. Should a key be reused, the OTP's security is compromised. The cipher is called the one-time pad because you have a "pad" of keys that are used only once and thrown out (hopefully not just "thrown out" but you get the picture
Next, we will explore how to encrypt and decrypt some example messages.

Pretend I want to send you the message "THE BRITISH ARE COMING". I must have a random key of length 19. How I obtain this key is not important, so long as it is truly random. Current methods of obtaining random keys include noise from webcams and key-generators connected to a radioactive substance that use the eccentricity of nuclear decay to construct their keys.

I'm going to use the key DKJFOISJOGIJPAPDIGN. All I did to obtain this key was bang on the keyboard, so it is not a "good" key, but it will do for purposes of this demonstration.

Step 1 - Write the PT above the key

T H E B R I T I S H A R E C O M I N G
D K J F O I S J O G I J P A P D I G N

Step 2 - Determine an algorithm

For a simple pen & paper implementation, I'm going to look up the numerical value of each letter in the alphabet (a=0, b=1, c=2, ..., z=25) for both PT and key, add them together and take it MOD the length of your alphabet (26 in this case; a-z). This gives us a new numerical value 1-26 which we can look up in our alphabet table and find the new encrypted CT character. I, personally, like to write out the table on my paper so it's easy for both me and the recipient of the message to encode and decode. It follows the formula "(plaintext + key) MOD alphabet length":

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Step 3 - Perform the encryption

(T(19)+D(03)=22) MOD 26 = 22 = W
(H(07)+K(10)=17) MOD 26 = 17 = R

(E(04)+J(09)=13) MOD 26 = 13 = N
(B(01)+F(05)=06) MOD 26 = 06 = G
(R(17)+O(14)=31) MOD 26 = 05 = F
(I(08)+I(08)=16) MOD 26 = 16 = Q
(T(19)+S(18)=37) MOD 26 = 11 = L
(I(08)+J(09)=17) MOD 26 = 17 = R
(S(18)+O(14)=32) MOD 26 = 06 = G
(H(07)+G(06)=13) MOD 26 = 13 = N
(A(00)+I(08)=08) MOD 26 = 08 = I
(R(17)+J(09)=26) MOD 26 = 00 = A
(E(04)+P(15)=19) MOD 26 = 19 = T
(C(02)+A(00)=02) MOD 26 = 02 = C
(O(14)+P(15)=29) MOD 26 = 03 = D
(M(12)+D(03)=15) MOD 26 = 15 = P
(I(08)+I(08)=16) MOD 26 = 16 = Q
(N(13)+G(06)=19) MOD 26 = 19 = T
(G(06)+N(13)=19) MOD 26 = 19 = T

From this fairly simple and quick process, we can determine that our CT is "WRNGFQLRGNIATCDPQTT". This is not vulnerable to simple frequency analysis because the same letter is not encrypted the same way twice (unless, of course, it aligns with the same key character twice). It should also be invulnerable to index of coincidence attacks because your key is not repeated; it is the length of the text.

Decryption is also quite straightforward. It follows the formula "(ciphertext - key + alphabet length) MOD alphabet length":

(W(22)-D(03)= 19 +26) MOD 26 = 19 = T
(R(17)-K(10)= 07 +26) MOD 26 = 07 = H
(N(13)-J(09)= 04 +26) MOD 26 = 04 = E
(G(06)-F(05)= 01 +26) MOD 26 = 01 = B
(F(05)-O(14)=-09 +26) MOD 26 = 17 = R
(Q(16)-I(08)= 08 +26) MOD 26 = 08 = I
(L(11)-S(18)=-07 +26) MOD 26 = 19 = T
(R(17)-J(09)= 08 +26) MOD 26 = 08 = I
(G(06)-O(14)=-08 +26) MOD 26 = 18 = S
(N(13)-G(06)= 07 +26) MOD 26 = 07 = H
(I(08)-I(08)= 00 +26) MOD 26 = 00 = A
(A(00)-J(09)=-09 +26) MOD 26 = 17 = R
(T(19)-P(15)= 04 +26) MOD 26 = 04 = E
(C(02)-A(00)= 02 +26) MOD 26 = 02 = C
(D(03)-P(15)=-12 +26) MOD 26 = 14 = O
(P(15)-D(03)= 12 +26) MOD 26 = 12 = M
(Q(16)-I(08)= 08 +26) MOD 26 = 08 = I
(T(19)-G(06)= 13 +26) MOD 26 = 13 = N
(T(19)-N(13)= 06 +26) MOD 26 = 06 = G

We can see the original message here: "The British are coming".

That's basically all there is to one time pads. Some key points to remember are:

1) Your key MUST be as long as your plaintext
2) Your key MUST NOT be reused
3) Your key MUST be random (not pseudorandom and PLEASE not a WORD!!)
4) The key MUST be exchanged over an existing secure channel

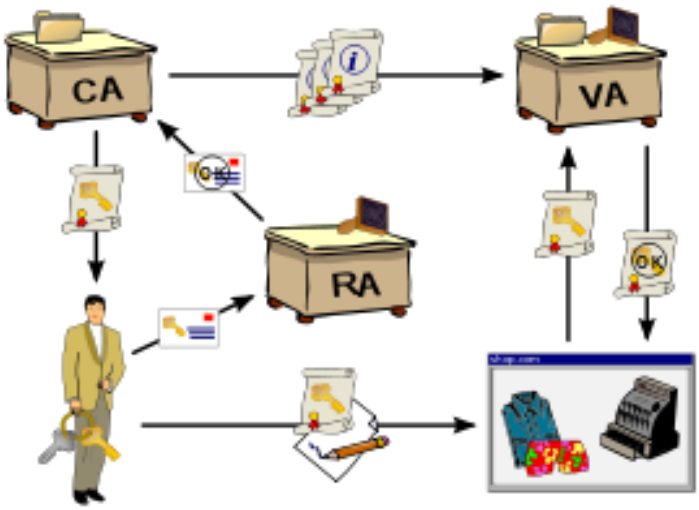
Practical: 12

Aim: Draw diagram of public key infrastructure.

A public key infrastructure (PKI) supports the distribution, revocation and verification of public keys used for public key encryption, and enables linking of identities with public key certificates. A PKI enables users and systems to securely exchange data over the internet and verify the legitimacy of certificate-holding entities, such as webservers, other authenticated servers and individuals.

PKI certificates include a public key used for encryption and cryptographic authentication of data sent to or from the entity that was issued the certificate. Other information included in a PKI certificate includes identifying information about the certificate holder, about the PKI that issued the certificate, and other data including the certificate's creation date and validity period.

Without PKI, sensitive information can still be encrypted, ensuring confidentiality, and exchanged between two entities, but there would be no assurance of the identity of the other party. Any form of sensitive data exchanged over the internet is reliant on the PKI for enabling the use of public key cryptography because the PKI enables the authenticated exchange of public keys.



Elements of PKI

A typical PKI includes the following key elements:

- A trusted party provides the root of trust for all PKI certificates and provides services that can be used to authenticate the identity of individuals, computers and other entities. Usually known as certificate authorities (CA), these entities provide assurance about the parties identified in a PKI certificate. Each CA maintains its own root CA, for use only by the CA.
- A registration authority (RA), often called a subordinate CA, issues PKI certificates. The RA is certified by a root CA and authorized to issue certificates for specific uses permitted by the root.
- A certificate database stores information about issued certificates. In addition to the certificate itself, the database includes validity period and status of each PKI certificate. Certificate revocation is done by updating this database, which must be queried to authenticate any data digitally signed or encrypted with the secret key of the certificate holder.
- A certificate store, which is usually permanently stored on a computer, can also be maintained in memory for applications that do not require that certificates be stored permanently. The certificate store enables programs running on the system to access stored certificates, certificate revocation lists and certificate trust lists.

A CA issues digital certificates to entities and individuals; applicants may be required to verify their identity with increasing degrees of assurance for certificates with increasing levels of validation. The issuing CA digitally signs certificates using its secret key; its public key and digital signature are made available for authentication to all interested parties in a self-signed CA certificate. CAs use the trusted root certificate to create a "chain of trust;" many root certificates are embedded in web browsers so they have built-in trust of those CAs. Web servers, email clients, smartphones and many other types of hardware and software -- including IoT devices -- also support PKI and contain trusted root certificates from the major CAs.

PKI certificates

Along with an entity's or individual's public key, digital certificates contain information about the algorithm used to create the signature, the person or entity identified, the digital

signature of the CA that verified the subject data and issued the certificate, the purpose of the public key encryption, signature and certificate signing, as well as a date range during which the certificate can be considered valid.

While PKI certificates are used for implementing cryptography over web and other internet connections, they are also used for other applications, including individual certification for code signing applications, for authenticating digital transactions and more.

Practical: 13

Aim: Draw the diagram of centralized/Decentralized Infrastructure.

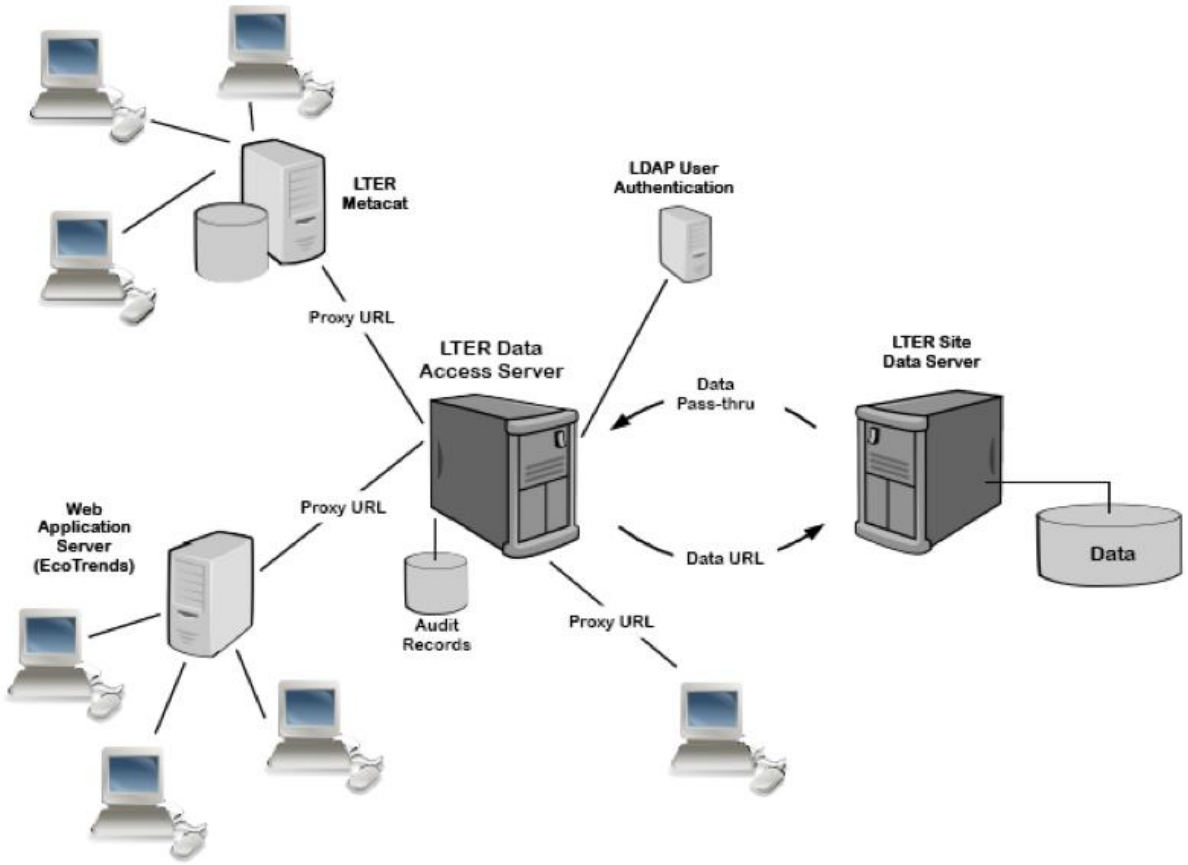


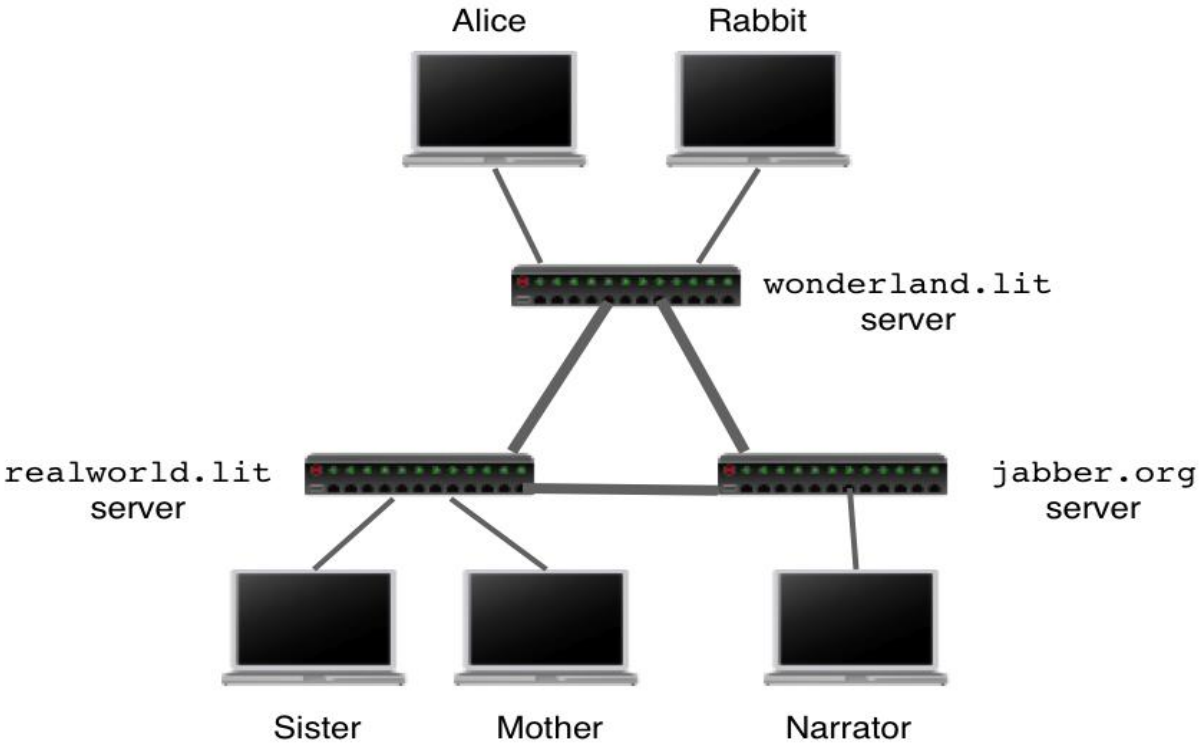
Figure 1: Hypothetical network topology of the Data Access Server.

Centralization means having one focus of control. One might have two DNS servers in every department of a company, but they all might be controlled by a single entity.

Centralization is an attempt to improve efficiency by taking advantage of potential economies of scale: improving the average; it may also improve reliability by minimizing opportunities for error.

centralized service that "do it yourself" has the potential of being better.

Decentralization:



Alternatively, decentralized systems distribute control to many parts. In our DNS example, each of those departments might maintain and control its own DNS server, being responsible for maintaining the skill set to stay on top of the technology as it changes, to architect the systems as it sees fit, and to monitor the service.

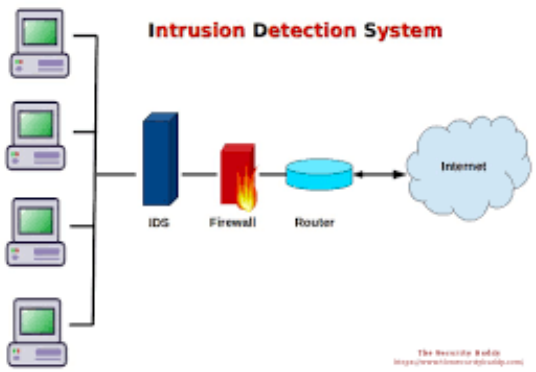
Decentralization is an attempt to improve speed and flexibility by reorganizing to increase local control and execution of a service.

Decentralization means breaking away from the prevailing hegemony, revolting against the frustrating bureaucratic ways of old.

Practical: 14

Aim: Explain intrusion detection Techniques in detail.

An **Intrusion Detection System (IDS)** is a system that monitors **network traffic** for suspicious activity and issues alerts when such activity is discovered. It is a software application that scans a network or a system for harmful activity or policy breaching. Any malicious venture or violation is normally reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system integrates outputs from multiple sources and uses alarm filtering techniques to differentiate malicious activity from false alarms.
Although intrusion detection systems monitor networks for potentially malicious activity, they are also disposed to false alarms. Hence, organizations need to fine-tune their IDS products when they first install them. It means properly setting up the intrusion detection systems to recognize what normal traffic on the network looks like as compared to malicious activity.



Intrusion prevention systems also monitor network packets inbound the system to check the malicious activities involved in it and at once sends the warning notifications.

Classification of Intrusion Detection System:

IDS is basically classified into 2 types:

Network Intrusion Detection System (NIDS):
Network intrusion detection systems (NIDS) are set up at a planned point within the network to examine traffic from all devices on the network. It performs an observation of passing traffic on the entire subnet and matches the traffic that is passed on the subnetts to the collection of known attacks.

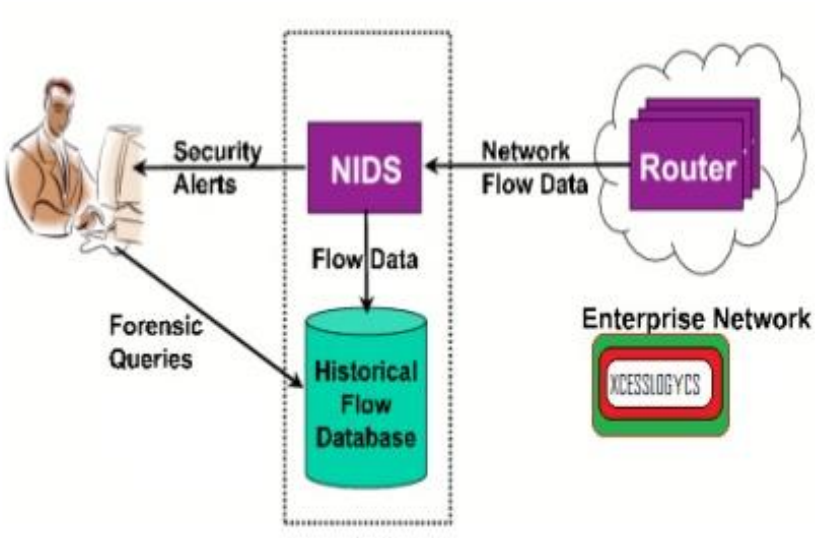
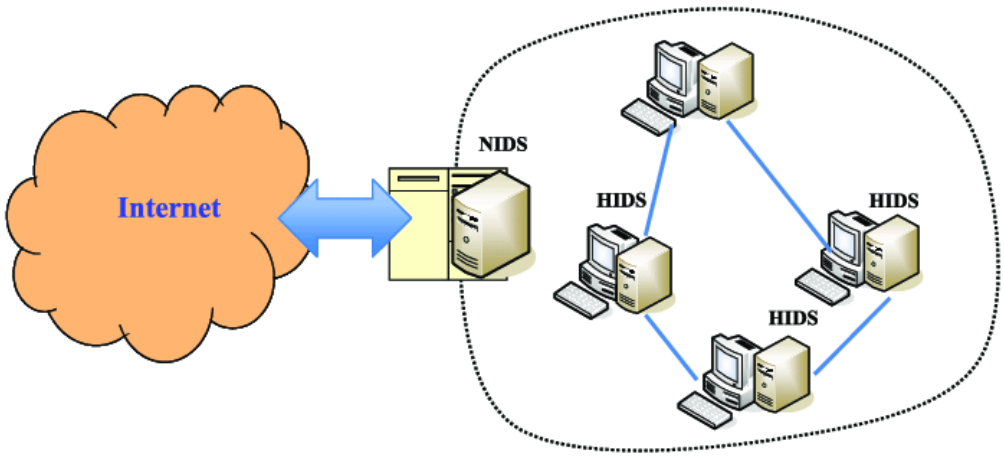


Fig. - Network Based IDS

Once an attack is identified or abnormal behavior is observed, the alert can be sent to the administrator. An example of an NIDS is installing it on the subnet where firewalls are located in order to see if someone is trying crack the firewall.

Host Intrusion Detection System (HIDS):
Host intrusion detection systems (HIDS) run on independent hosts or devices on the network. A HIDS monitors the incoming and outgoing packets from the device only and will alert the administrator if suspicious or malicious activity is detected. It takes a snapshot of existing system files and compares it with the previous snapshot.



If the analytical system files were edited or deleted, an alert is sent to the administrator to investigate. Anexample of HIDS usage can be seen on mission critical machines, which are not expected to change their layout.

Practical: 15

Aim: Demonstrate Cross Scripting.

- XSS is a web-based attack performed on vulnerable web applications.
- In XSS attacks, the victim is the user and not the application.
- In XSS attacks, malicious content is delivered to users using JavaScript.

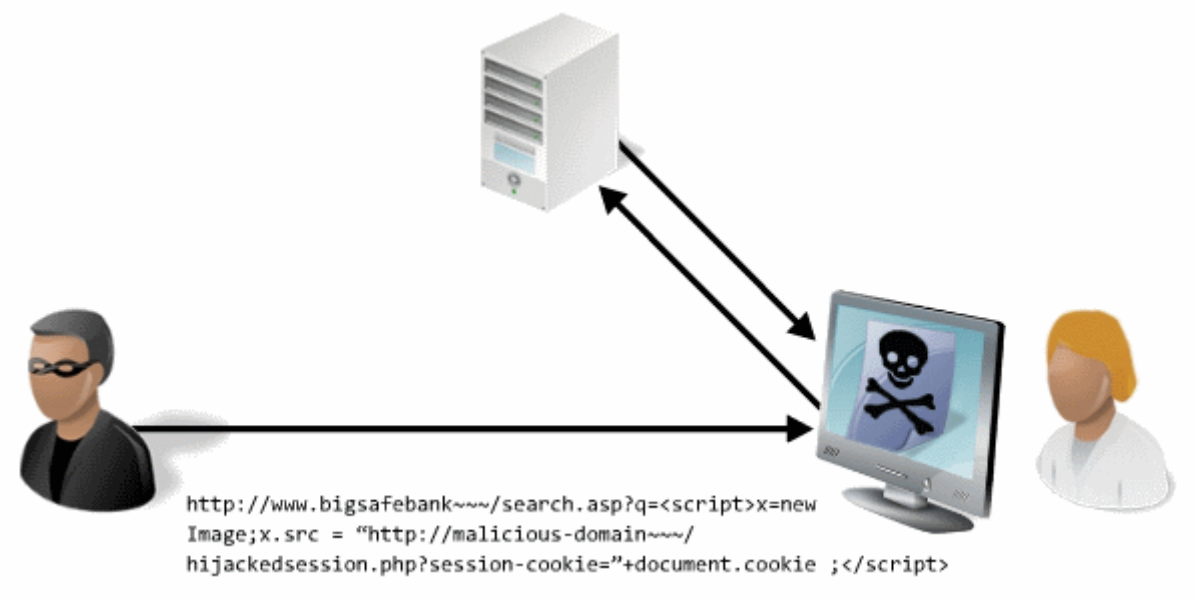
Cross-Site Scripting

An XSS vulnerability arises when web applications take data from users and dynamically include it in web pages without first properly validating the data. XSS vulnerabilities allow an attacker to execute arbitrary commands and display arbitrary content in a victim user's browser. A successful XSS attack leads to an attacker controlling the victim's browser or account on the vulnerable web application. Although XSS is enabled by vulnerable pages in a web application, the victims of an XSS attack are the application's users, not the application itself. The potency of an XSS vulnerability lies in the fact that the malicious code executes in the context of the victim's session, allowing the attacker to bypass normal security restrictions.

XSS Attack Examples

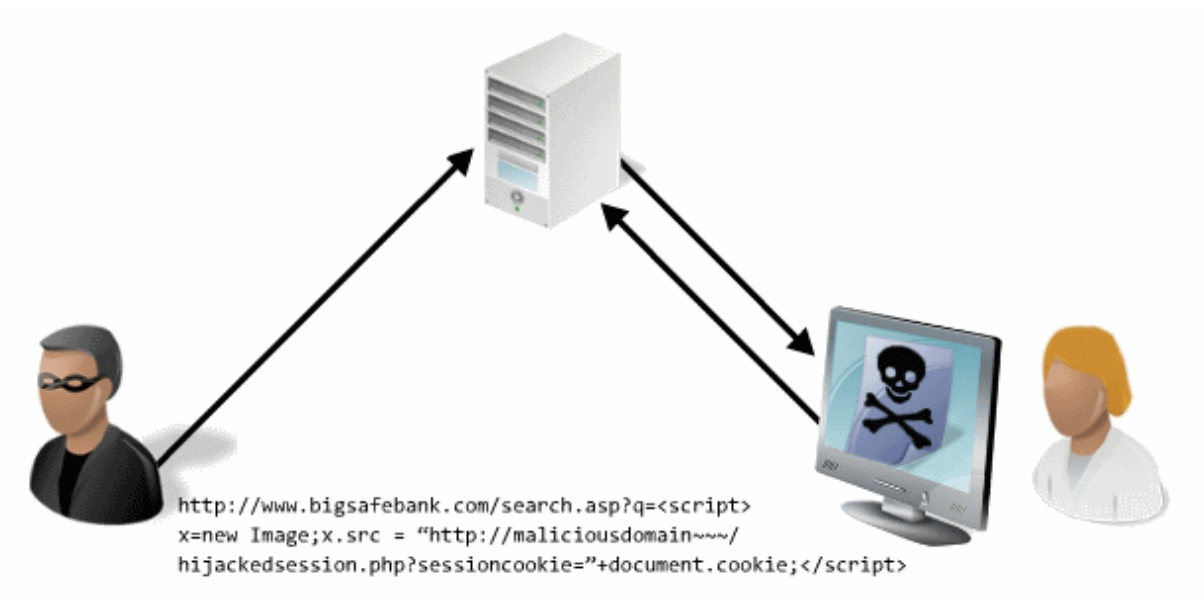
Reflective XSS

There are many ways in which an attacker can entice a victim into initiating a reflective XSS request. For example, the attacker could send the victim a misleading email with a link containing malicious JavaScript. If the victim clicks on the link, the HTTP request is initiated from the victim's browser and sent to the vulnerable web application. The malicious JavaScript is then reflected back to the victim's browser, where it is executed in the context of the victim user's session.



Persistent XSS

Consider a web application that allows users to enter a username that is displayed on each user's profile page. The application stores each username in a local database. A malicious user notices that the web application fails to sanitize the username field and inputs malicious JavaScript code as part of their username. When other users view the attacker's profile page, the malicious code automatically executes in the context of their session.



Impact of Cross-Site Scripting

When attackers succeed in exploiting XSS vulnerabilities, they can gain access to account credentials. They can also spread web worms or access the user's computer and view the user's browser history or control the browser remotely. After gaining control to the victim's system, attackers can also analyze and use other intranet applications.

By exploiting XSS vulnerabilities, an attacker can perform malicious actions, such as:

- Hijack an account.
- Spread web worms.
- Access browser history and clipboard contents.
- Control the browser remotely.
- Scan and exploit intranet appliances and applications.

Identifying Cross-Site Scripting Vulnerabilities

XSS vulnerabilities may occur if:

- Input coming into web applications is not validated
- Output to the browser is not HTML encoded

- **Example** **1.**

  For example, the HTML snippet:

-
- `<title>Example document: %(title)</title>`

  is intended to illustrate a template snippet that, if the variable title has value Cross-Site Scripting, results in the following HTML to be emitted to the browser:

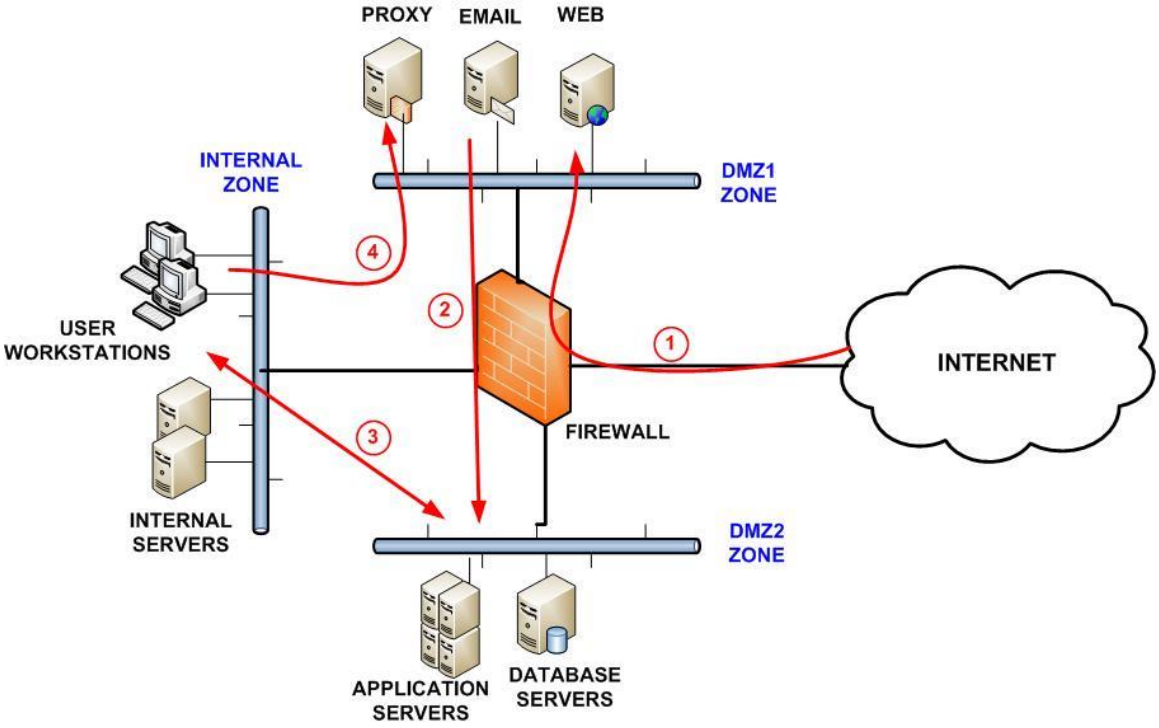
- `<title>Example document: XSS Doc</title>`

Practical: 16

Aim: Draw various Security topologies.

DMZ

The acronym *DMZ* originates from the military term Demilitarized Zone which refers to an area declared as a buffer between two sides in a war. In IT security the term DMZ is used to refer to what is essentially a buffer between the internet and the internal network. The DMZ is separated by an *outer firewall* on the internet facing side of the DMZ and an *inner firewall* on the internal network side of the DMZ. Any devices placed within the DMZ are accessible from both the internet and the internal network. There is no communication, however, from the internet directly though the DMZ to the internal network.



Any systems placed in the DMZ must be configured to the highest level of security possible (with the caveat that they must still be able to perform the role for which they are intended). These systems should always be considered to be compromised and must never be given direct and unrestricted access to the inner network. Servers typically placed in the DMZ are web, ftp, email and remote access servers.

INTERNET

The internet is the name given to the entire public network which provides the infrastructure for the transfer of data between remote points. Such data can take the form of email, web pages, files, multi-media and just about anything else that exists in digital form.

Whilst the internet seems like one giant network it is in reality a mesh of interconnected networks held together by routers which control and direct the flow of data from point to point until it reaches its destination.

The internet is completely open and as such there is no way to control what takes place on it. Whilst much of the activity on the internet is harmless it is also a fertile breeding ground for those with malicious intentions. It is for this reason that any computers or networks with access to the internet must be protected by a firewall.

INTRANET

An intranet can be described as a mini-internet build within the safety of a secure networking environment. Intranets are typically used to provide internal corporate web sites for employee only access. Because the intranet servers have internal, private IP addresses and reside behind firewalls they are generally not accessible to the outside world. If external access is needed to an intranet this is best achieved through the implementation of a Virtual Private Network (VPN).

EXTRANET

An extranet is a portion of an intranet which is made accessible to external partners. Access to an extranet is typically controlled by strict levels of authentication and authorization through the use of VPNs, firewalls and security policies.

VIRTUAL LOCAL AREA NETWORK (VLAN)

A local area network (LAN) is typically a collection of devices connected to a single switch. A virtual local area network (VLAN) typically involves grouping devices on a single switch into multiple broadcast domains and network segments. This provides a way to limit broadcast traffic on each segment of the network (improving overall performance) and increased security through the deployment of multiple isolated LANs on a single switch. A concept known as *trunking* can be used to create a VLAN which spans multiple switches. This enables users to be grouped on VLANs based on function rather than by physical location. For example all members of the accounting department can be placed in the same VLAN regardless of the switches to which they are physically connected.

NETWORK ADDRESS TRANSLATION (NAT)

Network Address Translation (NAT) provides a mechanism for using two sets of IP addresses for internal network devices, one set for internal use and another for external use.

NAT was originally developed to address the problem that the supply of available IPv4 IP addresses is beginning to run out.

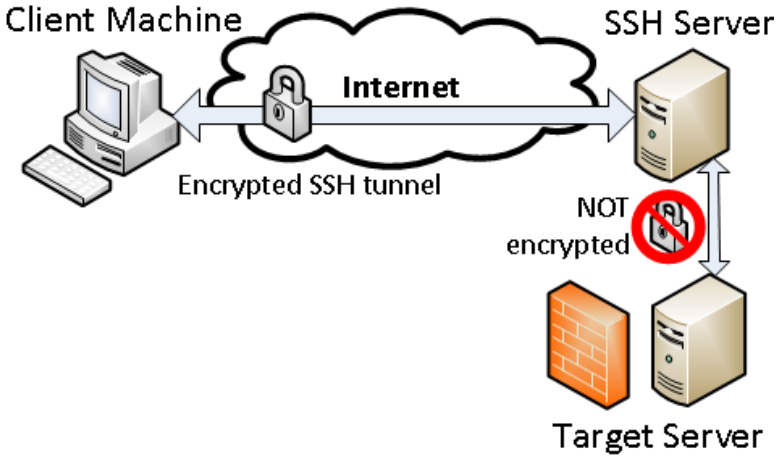
NAT translation typically takes place at a router or firewall and allows internal networks to assign so-called *non-routable* or *private* IP addresses for internal devices whilst using a single IP address for external communication across the internet.

Private IP addresses fall into specific ranges known as *classes*. Each of the following classes is considered to be non-routable on the internet:

- **Class A** - 10.0.0.0 - 10.255.255.255. Valid IP addresses are from 10.0.0.1 to 10.255.255.254.
- **Class B** - 172.16.0.0 - 172.31.255.255. Valid IP addresses are from 172.16.0.1 to 172.31.255.254.
- **Class C** - 192.168.0.0 - 192.168.255.255. Valid IP address are from 192.168.0.1 to 192.168.255.254

TUNNELING

Tunneling involves the packaging of data packets so that they can securely traverse a public network. In essence, the packets for one protocal are encapsulated in the packets of another protocol. An example is the Point-to-Point Tunneling Protocol which encapsulates its own packets into the TCP/IP protocol. Encapsulation is often combined with encryption to increase the level of security.



Practical: 17

Aim: Demonstrate traffic analysis of different network protocols using tool.

Wireshark is a free and open source network protocol analyzer that enables users to interactively browse the data traffic on a computer network. The development project was started under the name Ethereal, but was renamed Wireshark in 2006.

Many networking developers from all around the world have contributed to this project with network analysis, troubleshooting, software development and communication protocols. Wireshark is used in many educational institutions and other industrial sectors.

Wireshark is a network or protocol analyzer (also known as a network sniffer) available for free at the Wireshark website. It is used to analyze the structure of different network protocols and has the ability to demonstrate encapsulation. The analyzer operates on Unix, Linux and Microsoft Windows operating systems, and employs the GTK+ widget toolkit and pcap for packet capturing. Wireshark and other terminal-based free software versions like Tshark are released under the GNU General Public License.

Wireshark shares many characteristics with tcpdump. The difference is that it supports a graphical user interface (GUI) and has information filtering features. In addition, Wireshark permits the user to see all the traffic being passed over the network.

Features of Wireshark include:

- Data is analyzed either from the wire over the network connection or from data files that have already captured data packets.
- Supports live data reading and analysis for a wide range of networks (including Ethernet, IEEE 802.11, point-to-point Protocol (PPP) and loopback).
- With the help of GUI or other versions, users can browse captured data networks.
- For programmatically editing and converting the captured files to the editcap application, users can use command line switches.
- Display filters are used to filter and organize the data display.
- New protocols can be scrutinized by creating plug-ins.
- Captured traffic can also trace Voice over Internet (VoIP) calls over the network.
- When using Linux, it is also possible to capture raw USB traffic.

Practical: 18

Aim: Demonstrate sniffing using packet tool.

Packet Sniffer (aka Network Sniffer, Network Analyzer, Packet Analyzer) is a troubleshooting and network analyzing tool that is very useful and important to master, but is often forgotten.

Network analyzing tools come in many forms and are used to monitor the traffic conversations that occur across the network. Often the information obtained from a packet sniffer can be used to figure out exactly how devices are communicating, making it easier to figure out the root cause of the problem you're troubleshooting. But the use of a packet sniffer is not limited to troubleshooting, it can also be used to help train, design and operate devices on a network.

What is a Packet Sniffer?

So what exactly is a packet sniffer and how can you use it? At its most basic, a packet sniffer captures traffic directly from a network interface and allows the user the ability to interpret the information contained within this traffic. While this is not that complicated for an experienced network engineer, it does limit the use of the tool to junior level engineers and novices.

Figure 1 below shows an example of a capture being done while browsing the web using the Hypertext Transfer Protocol (HTTP).

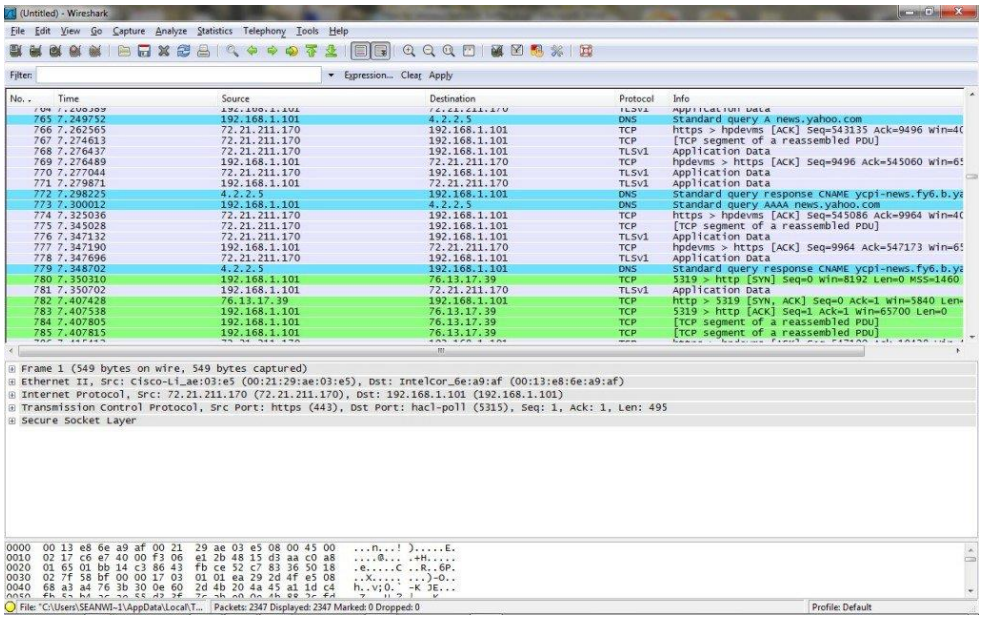


Figure 1: Packet Sniffer Capture

A specific packet can then be displayed showing all of the different information contained within; this is shown in Figure 2.

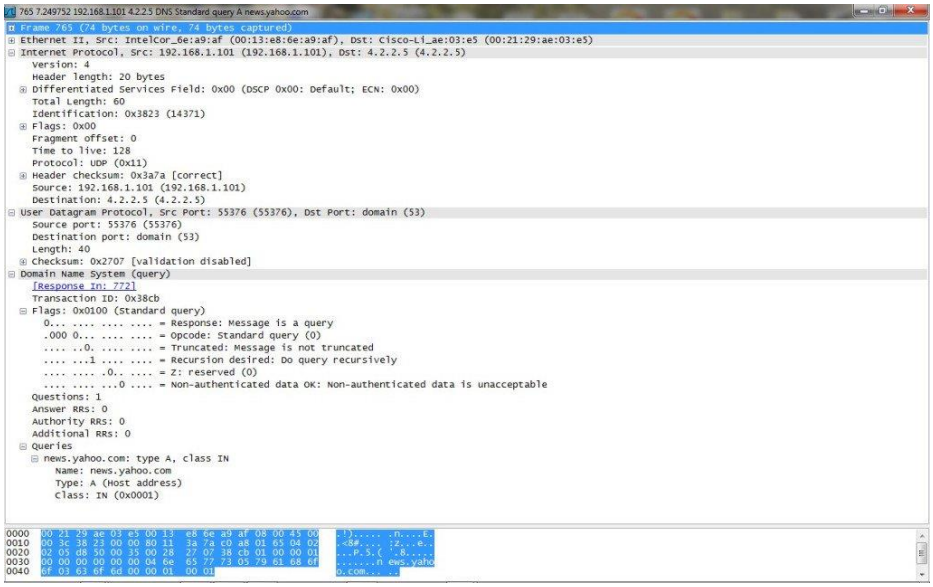


Figure 2: Example of a Packet Captured by the Packet Sniffer

There are also command line based scanners that are very popular but require a higher level of knowledge; one of the most popular is tcpdump which is typically used on Linux machines. Figure 3 below shows an example capture from tcpdump being run on a Linux based machine.

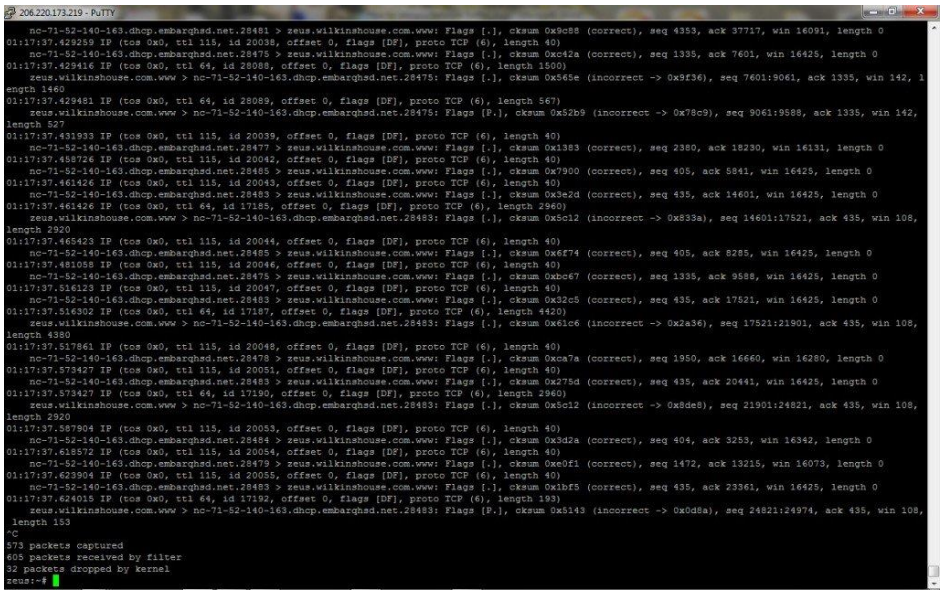


Figure 3: tcpdump Capture on Linux

More advanced functions that are possible on many packet sniffers include not only simple traffic capture but traffic analysis. This can include everything from simple tracking of conversations, statistical analysis, and stream analysis among many other options. Figure 4 below shows an example of a HTTP packet count statistical analysis.

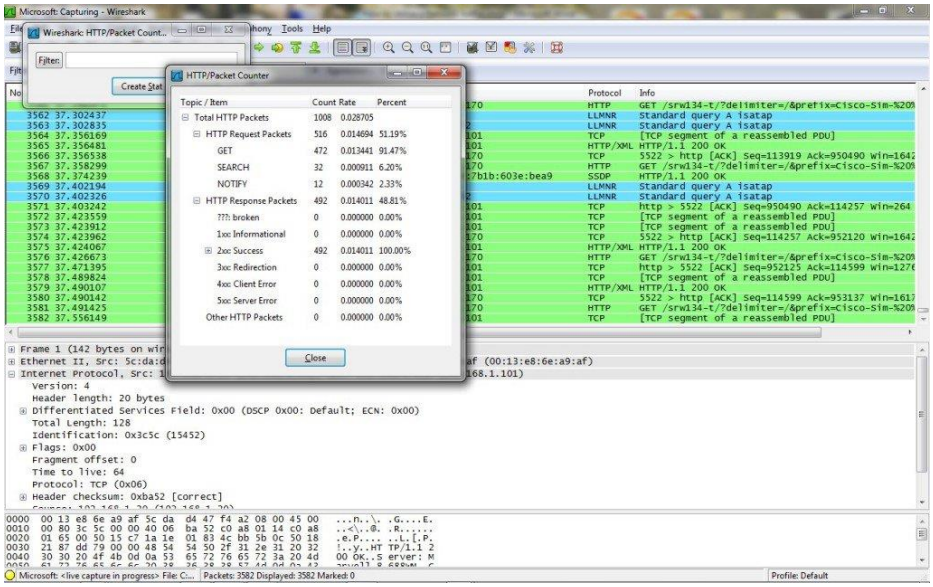


Figure 4: HTTP Packet Count Statistical Analysis

How to Use a Packet Sniffer

The basic capture of network traffic on most packet sniffers is relatively easy to start. Figure 5 below shows the capture options screen of a popular network protocol analyzer tool called Wireshark (more on Wireshark below).

This is the screen that is used to start a network capture. It is on this screen where the specific interface and options are selected before a capture is started.

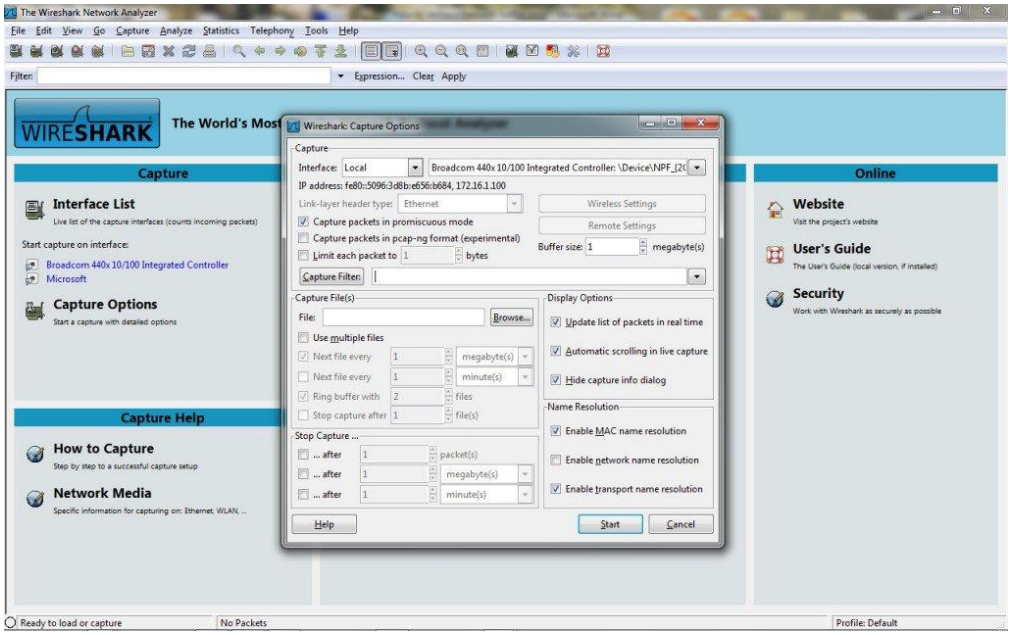


Figure 5: Wireshark Capture Options Screen

Once a capture is started in Wireshark, the screen will show the captured packets and permits the viewing of packet details as the capture continues. It is in these detailed packet screens where specific packet traffic analysis can be done; for example if a specific protocol conversation between hosts is being followed, this is where the details of this traffic can be seen. Figure 6 below shows an example of an FTP packet that was sent to initiate a file transfer; this can be verified as the only TCP flag used is SYN.

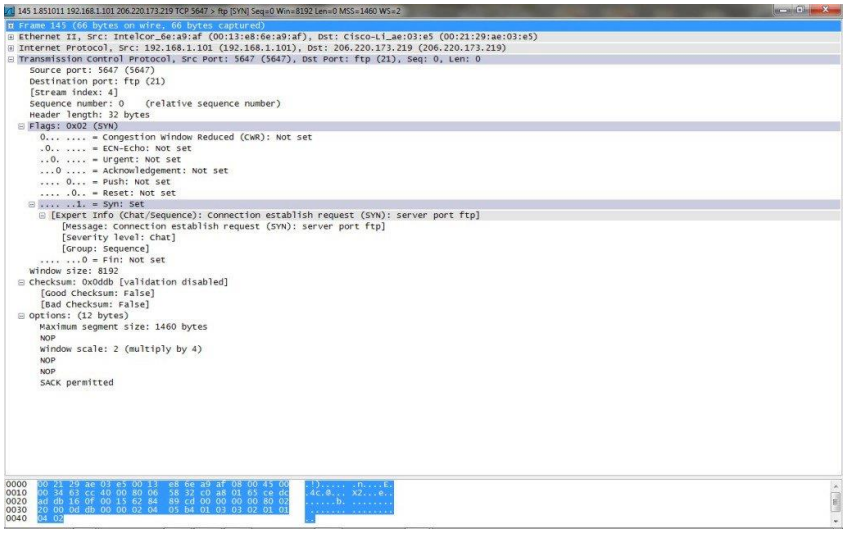


Figure 6: Example of an FTP Packet

Figure 7 below shows an example of a simple capture using the tcpdump utility on a Linux machine showing an initial ftp conversation.

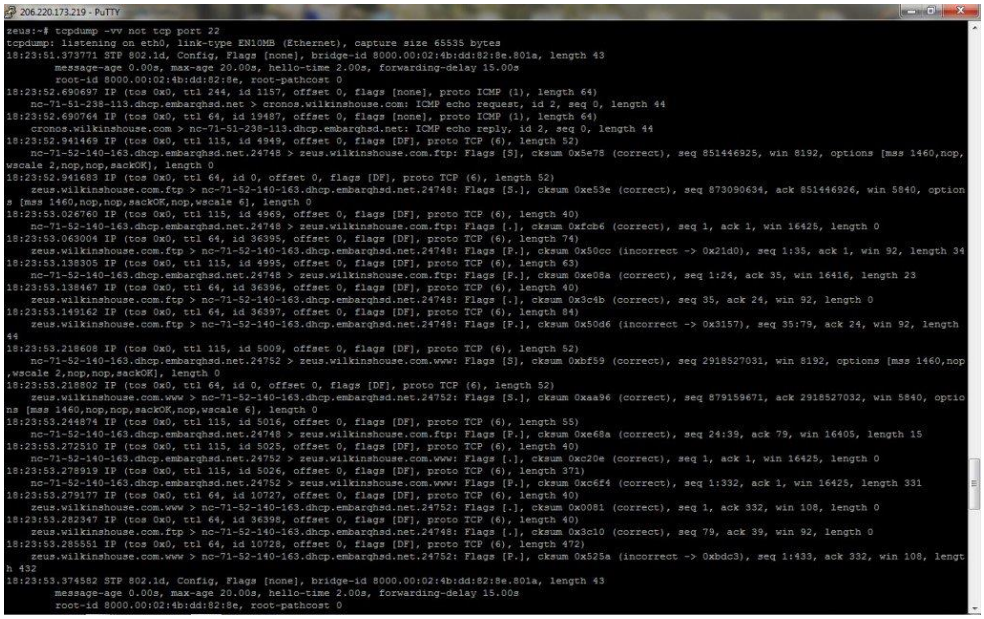


Figure 7: Capture using the tcpdump Utility

Wireshark can also be used to track the different conversations that are going on within the captured traffic. Figure 8 below shows how to use Wireshark to display this conversation list.

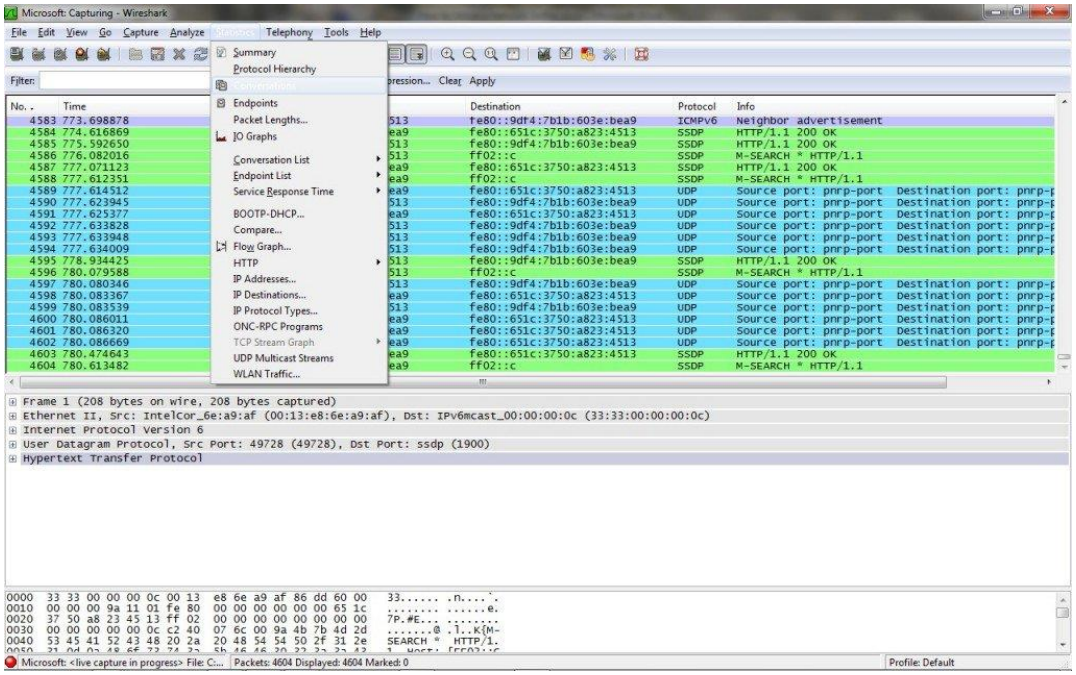


Figure 8: Example of Conversations within Captured Traffic

Figure 9 below shows an example of all the ongoing IPv4 conversations going on when capturing from a single computer. The packets going to and from the remote host via ftp are highlighted. This ability can be extended greatly when a packet sniffer is connected to an interface where multiple hosts are sending and receiving traffic.

In this type of configuration, the network sniffer can be used to troubleshoot network problems between a number of hosts and not just the traffic from the host running the packet sniffer software.

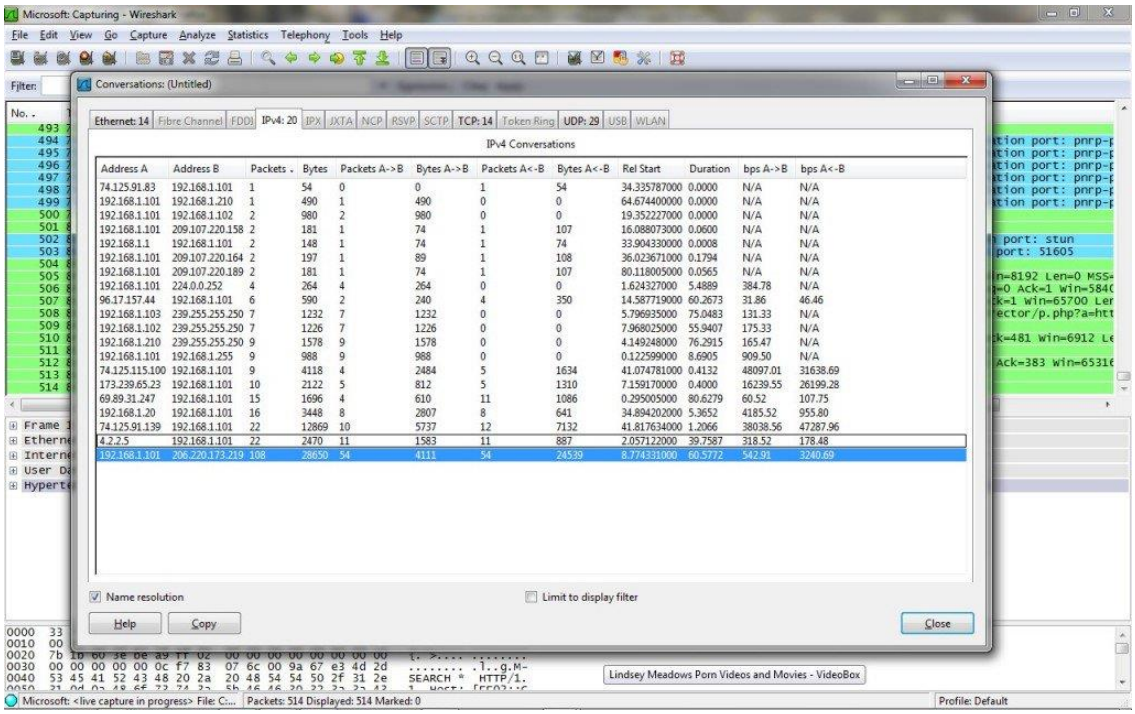


Figure 9: Example of IPv4 Conversations

The different things that can be done with a packet sniffer are really quite extensive and certainly cannot all be completely covered in a single article. The different tasks shown above are simply a few of the basic things that a network sniffer can be used for.

Practical: 19

Aim: Configure your e-mail account against various Threats.

Malware, short for malicious software, is frequently spread via e-mail on home networks. This type of security threat to home networks — and computers in general — may even appear to come from someone you know and trust. E-mail also has some original threats of its own, including spam, spoofing, and phishing attacks.

E-MAIL SECURITY THREAT: SPAM

Spam is the scourge of e-mail around the world. At times, it makes up as much as 95 percent of all e-mail on the Internet! Spammers get e-mail addresses from newsgroups, unscrupulous Web site operators who sell e-mail addresses to them, and malware that harvests e-mail addresses from hacked e-mail accounts. Spammers also guess e-mail addresses and sometimes just get lucky.

Spam causes a number of issues, including these:

- **Network congestion:** Spam clogs your network pipes. Although e-mail is relatively small in size, receiving enough of it will cause congestion on your network. Worse yet, if your computer has become part of a botnet, you will definitely see a negative effect on your network as you could be unknowingly *sending* thousands of spam e-mails to others!
- **Distraction and clutter:** Because spam can account for a large volume of e-mail, legitimate e-mails may get buried in your inbox or inadvertently deleted along with all the spam.
- **Malware:** A large proportion of spam contains malware, or links to Web sites that contain malware.

The best protection against spam (other than not using e-mail at all) is to use a spam filter. Of course, this may not be an option on your home network (although some Internet service

providers offer spam filtering as an additional service). If you don't **have a spam** filter, you should also use any junk mail filtering options available in your e-mail software.

E-MAIL SECURITY THREAT: SPOOFING

E-mail spoofing occurs when an attacker sends you an e-mail pretending to be someone you know. Spoofing is analogous to sending a letter to someone and forging the return address on the envelope. Unfortunately, e-mail spoofing is easy to do, and very difficult to trace to its real sender.

E-MAIL SECURITY THREAT: PHISHING

Phishing (pronounced like *fishing*) e-mails have become a favorite weapon of identity thieves, and they are becoming increasingly difficult to spot. Most phishing e-mails purport to be from a banking or other financial institution (as well as Web sites such as PayPal), and every once in a while they get lucky and actually send an e-mail pretending to be from *your* bank.

Phishing e-mails appear very authentic, and often include graphics and logos that are actually from your bank. There may even be a link that actually takes you to your bank's Web site. But buried somewhere in that e-mail is a link that takes you to a malicious Web site. Even if you don't enter any personal information, clicking the link can infect your computer with data-stealing malware.

- Never click a hyperlink in a suspect e-mail.
- Never reply to a suspect e-mail with personal information (such as social security numbers, account numbers, and passwords).
- Look for grammatical errors in the e-mail (but beware, identity thieves are getting more sophisticated).
- Contact your bank via telephone (get the number from your bank's Web site, not from the e-mail you received) if you suspect fraud.

- If you subscribe to e-mail or text alerts from your bank or financial institution, you should be familiar with the format, content, and address of these messages. Be suspicious of anything you receive that is out of the norm.
- Watch for small charges on your financial statements — to avoid detection, a thief is more likely to steal a few dollars from thousands of bank accounts rather than several hundred dollars from a few bank accounts.