Control Hijacking

Control Hijacking

- Hijacking is a type of network security attack.
- The attacker takes control of a communication.
- Just as an airplane hijacker takes control of a flight.
- Also known as a man in the middle attack.
- The attacker takes control of an established connection while it is in progress.
- The attacker intercepts messages in a public key exchange and then retransmits them, substituting their own public key for the requested one, so that the two original parties still appear to be communicating with each other directly.
- The attacker uses a program that appears to be the server to the client and appears to be the client to the server.
- This attack may be used simply to gain access to the messages, or

Attacker's goal in control hijacking-

a) Takeover target machine (for example web server)

b) Execute arbitrary code on target (web server) by hijacking application control flow

There are three types of control hijacking

a) Buffer overflow attacks

b) Integer Overflow attacks

c) Format String vulnerabilities

a) Buffer Overflow attacks

A buffer is a temporary area for data storage. When more data gets placed by a program or system process, the extra data overflow.

It causes some of that data to leak out into other buffers, which can corrupt or overwrite whatever data they are holding.

b) Integer Overflow attacks

An Integer Overflow attacks occurs when an attacker causes a value in the program to be large enough to overflow unexpectedly.

An Integer overflow is the condition that occure when the result of an arithmetic operation, such as multiplication or addition , exceeds the maximum size of the integer types used to store it.

c) Format String vulnerabilities

Format String vulnerabilities are a class of bug that takes advantage of an easily avoidable programmer error.

Format String vulnerabilities arise when user controllable input is passed as the format string parameter to a function that takes format specifiers that mat be misused, as in printf family in C.

Defense against Control Hijacking

Control Hijacking Attack Controlled or defense through:

a) Platform Defense

   i. Fixed the bug

   ii. Making Memory as non - Execute

b) Run time Defense

   i. Random Canary

a) **Platform Defense** – Through platform defense we can prevent target machine (server) by using

i. **Fixed the bug-**

a). Audit software through automated tools

b).Rewrite software in a safe language

c). Prevent code execution.

ii. **Making Memory as non – execute**: Prevent attack code execution by making stack and heap as non-executable.

b) **Run time Defense** – In run time defense we tests for stack integrity. We embed "canaries" in stack frames and verify their integrity prior to function return.