Mohit R. Punjabi
B22-93

Page No.
Date

27/11/20

# Assignment No. 4

Aim: To study implementation of Buffer overflow attack

Theory:

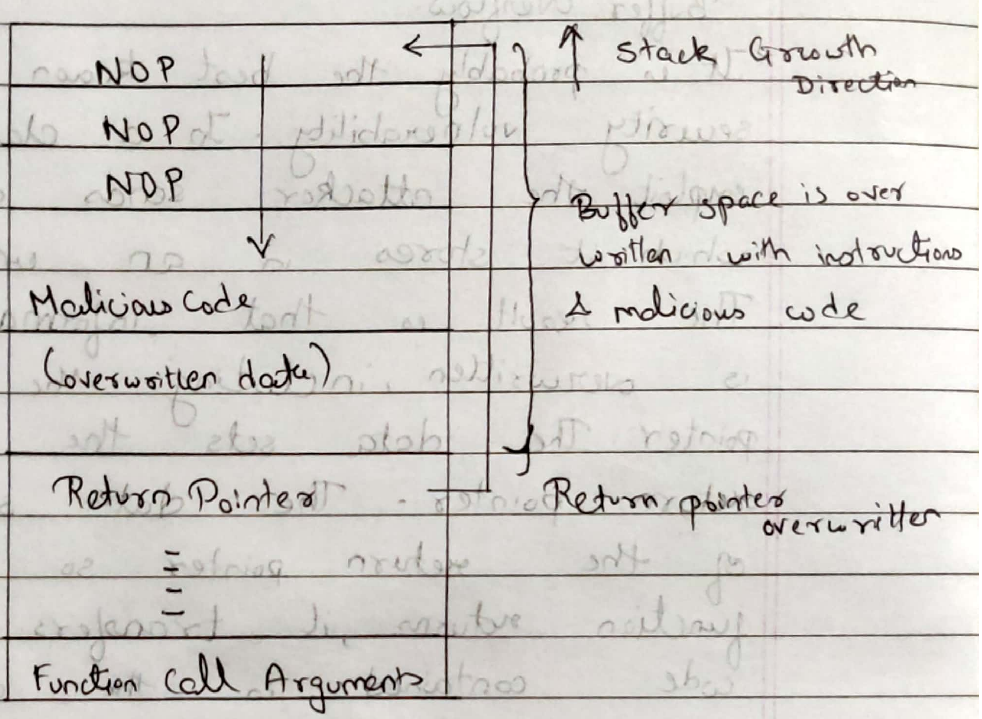
Buffer Overflow:

It is probably the best known form of software security vulnerability. In a classic buffer overflow exploit, the attacker sends data to a program which it stores in an undersized stack buffer. The result is that information on the cell is overwritten, including the function's return pointer. The data sets the value of the return pointer. The data sets the value of the return pointer so that when the function returns, it transfers control to malicious code contained in the attackers data

Buffer overflow vulnerabilities typically occurs in code that

- Relies on external data to control its behaviour
- Depends upon properties of the data that are enforced outside of the immediate scope of the code
- Is so complex that a programmer cannot accurately predict its behaviour

Buffer overflow occurs can be present in both the web server or application server products that serve the static & dynamic aspects of the site or the web application itself. Buffer overflows can also be more likely given the lack of security that web applications typically go through.



~~The~~ The above diagram shows how buffer overflow attack overwrites the Return Address & makes it to point to a location containing NOP (No Operation) so that the execution is directed towards some Malicious Code

~~Pred.~~ Protective Counter measures: Various techniques have been used to detect or prevent overflow are:

a) Choice of Programming language
b) Use of safe libraries

Page No.
Date

c) Pointer Protection
d) Executable Space Protection
e) Address space layout randomization
f) Deep Packet Inspection
g) Testing

Conclusion: Thus we have successfully studied about the Buffer overflow attack