Abstract

With the capability of infecting thousands of hosts in a fraction of a second, Internet Worms represent a major threat to the resources on the Internet. Worms exploit *vulnerabilities* in the programs to hijack the control flow to execute its own malicious code and take the control of the host. The defense against Internet worms is largely an open problem. Researchers have proposed various mechanisms to protect the network from the Internet Worm Attacks. Most worm detection mechanisms use signature based approach. Traditionally used signatures have limited coverage and can be easily evaded by an attacker with small variations of the exploit message.

To address the problem of low coverage, researchers proposed generation of vulnerability signatures. Such signatures have the potential to protect the system from all attacks that exploit that given vulnerability. Generation of a vulnerability signature is however a challenging task as it needs analyzing of the binary executables of the vulnerable program. Binary codes are complex due to lack of high level semantic structure and their dependence on compilers. It is anticipated that use of protocol information such as the handshake sequence, the packet format, the field types etc. will make the processing of these executable more feasible.

In this project we work towards developing a vulnerability signature generator utilizing protocol information. In this the generated vulnerability signature shall be in terms of protocol field parameters so that these can be used for packet filtering at the routers or hosts.

The project is divided into number of modules. Recently, we have worked with Receive Buffer identification module and protocol variable location modules. The main objective of the Receive buffer Identification module is to find the base address of Receive Buffer at Run-time. The objective of Protocol Variable Location Module is to identify the location of the protocol variables based on the Receive buffer access. We have developed algorithms for the two modules and implemented them successfully in IDA Pro environment. The experimental results show that the algorithms accurately identify the receive buffer as well as protocol variable locations successfully.