Abstract

With the capability of infecting thousands of host in a fraction of a second, Internet Worms represent a major threat to the resources on the Internet. Worms exploits *vulnerabilities* in the program to hijack the control flow to execute its own malicious code and take 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 potential to protect the system from all attacks that exploit the 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 executables 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 packer filtering at the routers or hosts.

The project is divided into a number of modules. In this project I have worked on Protocol Variable to Path Variable module. The main objective of the module is to identify the protocol variables and map protocol variables to path variables. Protocol variables hold subfields of protocol messages and they guide the execution flow of the program. Algorithm for detection of protocol variables and mapping protocol variables to path variables has been developed and successfully implemented in the IDA Pro environment. The experimental results show that the algorithms accurately identify the protocol variables and their mapping to path variables.