ABSTRACT

The alarming rise in the number of computer security incidents since the late 80's has inspired researchers to devise new mechanisms for detection and containment of malwares like worms and viruses. Many approaches have been proposed to detect new attacks and generate corresponding attack signatures. These approaches roughly fall into two categories: coarse-grained detectors, that detect anomalous behavior, such as scanning or unusual activity at a certain port; and fine-grained detectors, that detect attacks on a program's vulnerabilities. Coarse-grained detectors may result in frequent false positives, and do not provide detailed information about the vulnerability and how it is exploited. Thus, it is desirable to develop fine-grained detectors that produce fewer false positives, and provides detailed information about the vulnerability and exploit which can be used in generation of effective signatures.

Fine grained detectors which worked on traditional attack signatures proved ineffective in detection of new age polymorphic and metamorphic worms because of their inherent limitations to deal with worm mutation, code obfuscation etc. Limitations of this sort spawned a new variety of polymorphic and metamorphic resilient vulnerability signatures which are based on the vulnerabilities present in the program and not on the exploit pattern. However it did not prove very effective because of its low coverage. Protocol level vulnerability signatures are one of the newest approaches in this direction which is not only based on the vulnerabilities in program but also incorporates protocol level information in the signature generation to attain better coverage.

The prime goal of this project is generation of such fast and effective protocol level vulnerability signatures, which are also resilient towards polymorphic and metamorphic variations of attack by internet worms.

Current work is motivated by the fact that, as for exploiting vulnerability it is the input that must lead the program execution to the vulnerability point. Thus the movement, activity and the set of possible paths that an input takes for a successful exploitation needs to be monitored and tracked properly. Commonly first part is known as taint analysis and the latter one is called control flow graph or CFG pruning.

This dissertation is the result of working in this direction. It is specifically focuses on CFG pruning and taint analysis for generating protocol level vulnerability signature.