Review “Malware Detection”

Name: Wenhui Zhang

Paper Name：

Kolbitsch, C., Comparetti, P. M., Kruegel, C., Kirda, E., Zhou, X. Y., & Wang, X. (2009, August). Effective and Efficient Malware Detection at the End Host. In *USENIX security symposium* (Vol. 4, No. 1, pp. 351-366).

Contribution:

In this paper proposes a behavior based solution for malware detection. This solution first put malware in a controlled environment and get signature of its behavior. The signature contains information of information flow between malware invoked system calls. Then program slices are extracted for this information flow. For runtime detection, slices are matched against behavior of unknown programs.

Motivation:

Current host based intrusion detection system (IDS) is not effective. Malware invoked system calls are easily bypassing obfuscation and polymorphic techniques. System call reordering increases false positive rates of IDS system. However malware behavior is hard to obfuscate and randomize, and it is quite stable from one version of malware to another version.

Related works:

Network packets signature based methods, static analysis like symbolic execution and dynamic analysis techniques are compared with this paper’s idea. Network based method is only useful for detecting some network malware, could not deal with encrypted data. Static analysis is difficult to enforce. it could not detect metamorphic code, takes a heavy toll on system resources. Dynamic analysis techniques requires special hardware for taint tracking on the end host to perform detection, while this paper’s solution does not.

Methodology:

(1) Analyzing unknown binaries and disassembles instructions;
(2) Each byte is tainted to detect data dependencies between system calls;
(3) Any labels within a branch operation are labeled with the taint of the controlling instruction for control dependencies;
(4) Generate initial behavior graph, label all instructions, and put edges where dependency is found;
(5) Conduct program slicing;
(6) Scanner monitors running program for sys calls and match.

Results:

When restricting samples to 155 known variants, a 92% effectiveness was obtained with no false positives. When restricted data samples to 108 unknown variants, a 23% effectiveness was obtained with no false positives. It has a low overhead at runtime.

Take away:

Delays and time trigger malware behaviors could evade signature detection. Modification of malicious algorithms contained in a program could not be automated at the time when this paper is published.