Idiosyncratic Signatures for Authenticated Execution of Management Code

Mario Baldi[1], Yoram Ofek[2], and Moti Yung[3]
[1] Torino Polytechnic, Computer Engineering Department, Torino, Italy
mario.baldi@polito.it - www.polito.it/~baldi
[2] Synchrodyne Networks, Inc., New York, NY
ofek@synchrodyne.com
[3] Columbia University, Computer Science Department, New York, NY
moti@cs.columbia.edu

Abstract. TrustedFlow™ is a software solution to the problem of remotely authenticating code during execution. A continuous flow of idiosyncratic signatures assures that the software from which they have emanated is not changed prior to and during execution. TrustedFlow™ can be used to create a run-time trust relationship between the components of a distributed management system.

Software, especially in the context of data networks, suffers from some inherent problems. These include modifications by an either malicious or inadvertent attacker, malware distribution (e.g., viruses and "Trojan horses"), and the use of malicious software remotely for penetration, intrusion, denial-of-service (DoS), and distributed DoS (DDoS). Management software is particularly critical from this point of view since it is used to *monitor* and, especially, to *control* network devices. Hence, malicious modification and use of management code can be particularly harmful to the network and advantageous for the attacker. Moreover, distributed, possibly self-, management software, which is spread on various systems and possibly dynamically downloaded, is particularly exposed to manipulations.

The presented software solution aims at overcoming the above mentioned problems and at assuring (in many typical scenarios) that management operations are executed by a trusted software source. The solution is based on continuous authentication, ensuring at run-time that the correct software has been employed. This unique method for continuous authentication is based on a continuous flow of idiosyncratic signatures that are constantly being generated and emanated during execution. With reference to the architecture shown in Figure 1, the idiosyncratic signatures are generated by a secret function called Trusted Flow Generator (TTG) that is hidden (e.g., obfuscated) in the software and whose execution is subordinated to the proper execution of the software being authenticated. The flow of signatures is validated at a remote trusted component called Trusted Tag Checker (TTC). Consequently, this method guarantees that the correct software modules are used at run time, i.e., the authenticity of the executed software can be trusted. An explanation of the TrustedFlow™ architecture and the basic principles of the the TrustedFlow™ protocol can be found in [1].

The TrustedFlow™ protocol is an *add-on software protection component* intended to be included within other protocols, such as those, for example, of distributed computation (e.g., grid computing), traffic generation (e.g., TCP), and management (e,g., SNMP). In essense, the TrustedFlow™ protocol provides run-time continuous (multi-factor) authentication, certifying the authenticity of software modules that were used to compute,

generate, and send messages. As such, it becomes evident that the TrustedFlow™ protocol has broad applications in both networking and computing for military and commercial environments including, besides trusted network management.

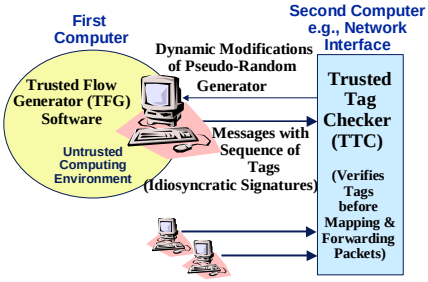

Figure 1: TrustedFlow™ architecture

The TrustedFlow™ protocol is complementary to many of the current enhancements for secure computing and networking protocols, including the security extensions to the Internet management model defined in SNMPv3. In fact, (to the best of our knowledge) no other authentication method certifies the software continuously during run-time by emanating idiosyncratic signatures. In other words, while other approaches provide privacy and authentication protecting from the attacks of a man in the middle, TrustedFlow™ protects from the attacks of *a man at the edge*. The TrustedFlow™ protocol has broad *synergistic implications* on various computing and networking protection means, as discussed in more detail the related work section of [1]. A prototype of the TrustedFlow™ architecture sponsored by Microsoft Research is under development at Torino Polytechnic.

In general, network management involves actions that are critical for proper network operation. SNMPv3 defines security mechanisms that enable authentication of management messages. However, these mechanisms do not protect from modified, possibly malicious, code that has access to proper authentication information. For example, by running on a network management station, the malicious code could get a hold of the certificates used for authentication with the management agent. The TrustedFlow™ protocol could be used to complement SNMP security mechanisms by including a TFG and a TTC in the management entity and in the management agent.

- A TFG in the management entity and a TTC in the management agent enables the latter to trust the management code requesting to store information in the MIB or to pass along information contained in the MIB.
- A TFG in the management agent and a TTC in the management station enables the latter to trust the agent code generating a trap or sending MIB information previously requested.

Embedding both a TFG and TTC in both the management entity and the management agent enables the creation of *mutual trust* in the execution of the other component. Hence, the TrustedFlow™ protocol can be used, possibly together with SNMP security mechanisms, to create a trusted run-time network management distributed environment. The resulting benefits increase with the degree of distribution and code mobility.

References

[1] M. Baldi, Y. Ofek, M. Yung, "Idiosyncratic Signatures for Authenticated Execution - The TrustedFlow™™ Protocol and its Application to TCP," IASTED CSN 2003, Benalmadena, Spain, 2003.