D36 Using Hashing to Improve Volatile Memory Forensic Analysis

Aaron R. Walters, MS, Center for Education and Research in Information Assurance and Security, Purdue University, West Lafayette, IN 47907; Douglas White, MS, National Institute of Standards and Technology, 100 Bureau Drive Stop 8970, Gaithersburg, MD 20899- 8970; and Blake Matheny, BS, Center for Education and Research in Information Assurance and Security, Purdue University, West Lafayette, IN 47907*

After attending this presentation, attendees will understand how information extracted from known files can improve volatile memory analysis and they will have insight into how automated tools can be built that leverage this information.

The presentation will impact the forensics community by demonstrating new methods of forensic volatile memory analysis, discussing how to derive trust in the data extracted from a potentially compromised system, and finally introducing a new resource being made freely available to the digital forensic community to support volatile memory analysis.

By extending hash databases to include immutable code sections of known executable files, an investigator can greatly augment the current techniques used for analyzing the volatile memory. This information can be used to automatically identify known aspects of the system's runtime state, thereby reducing the areas of volatile memory that need to be analyzed during the investigation. Since these immutable sections are compiler generated text, they can also be used to evaluate the runtime state of a potentially compromised system. This can be accomplished by evaluating the in-memory code sections of critical system executables that should not change if the system is in a trusted state. Also, by leveraging these static sections of code as anchors of trust, the investigator can evaluate the control flow integrity of the system at the time the memory image was acquired using a notion of transitive trust. This allows investigators to detect common mechanisms used to undermine the integrity of the system and compromise the integrity of the data collected from the system. By automatically identifying these changes to critical sections, it can also provide an investigator the mechanism to quickly triage the system and, if necessary, focus their attention on the time-intensive effort of determining the intent of the modifications.

This work will also discuss why the information currently stored in hash databases is inadequate to support the needs of the volatile memory community. It will also provide insight into the challenges associated with collecting this information. Outlines of the procedures that are needed to normalize the data extracted from the known files and of the procedures that need to be followed to extract and normalize the immutable code from volatile memory will also be presented.

A major goal of this effort is to provide freely available standard reference data, to support the needs of digital investigators and tool makers that are currently using this type of information. By leveraging the National Institute of Standards and Technology (NIST) National Software Reference Library (NSRL) it was possible to augment the Reference Data Set (RDS), a freely distributed collection of digital signatures for known applications, to include hashes of these immutable sections of data which will provide a valuable resource to the volatile memory analysis community.

Finally, the results of a number of experiments performed on both laboratory and real-world systems will also be presented. These experiments will attempt to demonstrate the impact of leveraging these techniques to improve volatile memory analysis.

Volatile Memory Analysis, Identification, Trust

 ** Presenting Author*