Alternative scalable HIDS with investigation capability

Degree programme: BSc in Computer Science | Specialisation: IT Security
Thesis advisor: Prof. Dr. Bruce Nikkel
Expert: Armin Blum

In this thesis, I show how a host-based intrusion detection system can be built for scalability. More specifically, I show how the sleuth kit can be used to create an intrusion detection system that does not rely on hashing, but on filesystem attributes. This way, it gains speed, which enables us to take the risk not to calculate hashes.

\chapter*{Abstract}
\label{chap:abstract}

Many tools exist to help to protect against cyber attacks. One family of those tools is called intrusion detection system (IDS). Those are created to detect attacks that somehow got through other measures and infected one or many hosts in the network. One type of IDS is called network-based intrusion detection system (NIDS). They operate on a network level and analyze the incoming and outgoing traffic for anomalies. Those anomalies usually signal an intrusion. When they find such an intrusion, they usually generate an alert for a system administrator or security professional to analyze.

There is also another type of IDS called host-based intrusion detection system (HIDS). They operate directly on the host and try to find attacks there. They are more effective at finding intrusions that are dormant and don‘t do anything for some time. They mostly operate on the file system and sometimes go beyond that. HIDS have been quite effective at finding intrusions on file basis in the past by creating cryptographic hashes of files and comparing them to previous executions. However, as file sizes have been growing, they began to struggle to execute within a short time frame. Calculating a hash is seen as the most reliable way to find changes to the file system, and with more data, it is taking increasingly long to calculate them. The situation has grown out of proportions because the time to scan now takes so long, that the intrusion detection system can‘t reliably find intrusions within a useful timespan.

In my thesis, I show a different solution to this problem. I created a host-based intrusion detection system that works at a file basis but does not calculate any hash. Instead, it finds intrusions by evaluating the file system attributes like modification time and permissions. This approach is risk-based because it is less reliable, but by increasing the speed, the host can be scanned multiple times more often than if hashes get calculated. I am using an open source forensic investigation tool called the sleuth kit (TSK). It offers much functionality for file system analysis and works on most operating systems. With this tool, I can extract the file system attributes reliably and fast, without touching the files themselves.

There is another advantage that offer with my system. Forensic investigators usually struggle to reliably create a timeline of what happened on a file system after an intrusion. This timeline is essential because it can lead them to find out what exactly happened and often can make future intrusions harder. Here I want to help as well. Different from the other HIDS, my system stores all the executions. This way, an investigator can look at this history and sees how the attack started. From this data I can then generate a timeline which shows what has happened when. One nice side-effect of storing all executions is that my system is very flexible. After changing something on the host, the system automatically adjusts.

With my tool, system administrators and forensic investigators have another option to tackle intrusion detection. Taking the risk-based approach can lead to many fast detections that otherwise would not be detected in time. Additionally, investigators have more data at their disposal to investigate incidents and learn valuable knowledge of the attacks and attackers.

Julian Jimmy Stampfli