Using Domain Name System to Thwart Automated Client-Based Attacks

Curtis Taylor
University of Tennessee – Knoxville
Knoxville, TN 37996

Craig Shue
Oak Ridge National Laboratory
Oak Ridge, TN 37830

On the Internet, attackers can compromise systems owned by other people and then use these systems to launch attacks automatically. When attacks such as phishing or SQL injections are successful, they can have negative consequences including server downtime and the loss of sensitive information. Current methods to prevent such attacks are limited in that they are application-specific, or fail to block attackers. Phishing attempts can be stopped with email filters, but if the attacker manages to successfully bypass these filters, then the user must determine if the email is legitimate or not. Unfortunately, they often are unable to do so. Since attackers have a low success rate, they attempt to compensate for it in volume. In order to have this high throughput, attackers take shortcuts and break protocols. We use this knowledge to address these issues by implementing a system that can detect malicious activity and use it to block attacks. To do this, our system constantly changes the location of all computers on the network by linking already created system tools. By changing the location of each server, clients must use the Domain Name Service protocol in order to find a server's current location. For example, we can configure computer A to have the IP address 1.2.3.4 and return this address when queried for computer A. However, we also stipulate that this address will be valid for only X seconds. After this time elapses, the client will have to send a new request asking for the new address of computer A. If the client fails to follow proper procedure, they can be classified as an attacker. Once an attacker has been discovered, they will be isolated and monitored. This can be accomplished using existing software in Ubuntu Linux applications, along with our custom wrapper application. After running the system and seeing its performance on three popular Web browsers Chromium, Firefox and Internet Explorer we found that not only is this system conceivable, it is effective and has low overhead. Overall, our approach provides a lightweight system that takes advantage of applications already being used on servers and can easily be integrated into current systems.