Making the Live Network the Honeypot

Michael Coughlin (student)[1], *Oliver Michel (student)*[1], *Eric Keller*[1], *Adam J. Aviv*[2]
[1]University of Colorado, [2]United States Naval Academy
{*michael.coughlin,oliver.michel,eric.keller*}@*colorado.edu, aviv@usna.edu*

With the ever increasing security threats that network infrastructures encounter, digital forensics has become an import tool in attempts to understand attacks, and ideally prevent future attacks through an automated, continuous feedback loop [2]. Forensic evidence, however, is only a snapshot of the state of an attack. We wish to go beyond simply collecting evidence, and be able to learn the behavior of an attacker. One approach is to use a honeypot system [1], where dedicated systems are left exposed to attract attacks and are monitored to capture the actions of an attacker. For a network to be able to learn information relevant to the specific network, the honeypot needs to closely mimic the real network – a difficult task. We are exploring another point in the space, where instead of exposing vulnerable systems which are isolated in advance, we effectively make the live network the honeypot.

The challenge is that we do not wish to allow an attack to continue on a live network, but we need the attacker to think it is still on the live network. To achieve this, we propose a new approach which, upon a machine becoming infected, clones the infected machine, disinfects the original machine, and quarantines the cloned (still infected) machine. There are two main challenges in this:

Preventing leakage of confidential data: As the data on the original system may contain confidential information, while cloning the machine we need to clean up this data. Instead of preventing access to it (e.g., by deleting it), the data can be falsified using various methods (e.g., with decoy documents [4]), in order to provide the attacker with misinformation.

Preventing further damage without alerting the attacker: In order to operate this system without alerting the attacker, the system must be able to make both machines appear as one to an outside observer. This can be achieved using software-defined networking (SDN) to modify packet headers and forwarding so that traffic is transparently redirected to the correct host (the infected clone or the original). When the infected machine tries to reach out to other machines on the local networks, we can, similar to GQ [3], spin up VMs to act as internal servers (using SDN to direct requests to the fake servers).

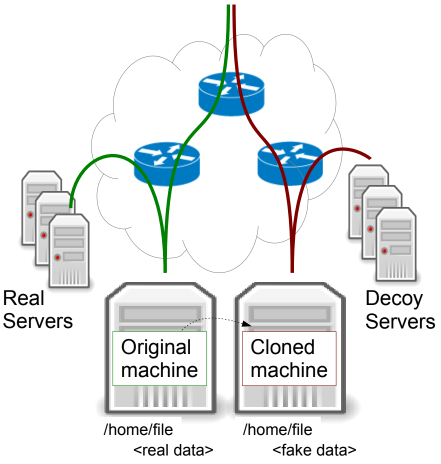

Figure 1: Live Honeypot architecture.

We have built an early prototype system that implements network redirection as an application within the Floodlight OpenFlow controller, and clones the machine through VM migration technology – currently we are simply pausing the VM, cloning it, then restarting both but will integrate live cloning. We are in the process of investigating modifications to enable misinformation.

References

[1] The honeypot project. `http://honeypot.org`.

[2] R. Hand, M. Ton, and E. Keller. Active Security. In *HotNets*, 2013.

[3] C. Kreibich, N. Weaver, C. Kanich, W. Cui, and V. Paxson. Gq: Practical containment for measuring modern malware systems. In *Proc. Conference on Internet Measurement Conference*, 2011.

[4] M. B. Salem and S. J. Stolfo. Decoy document deployment for effective masquerade attack detection. In *Proc. conference on Detection of intrusions and malware, and vulnerability assessment (DIMVA)*, 2011.