

Echos: A Distributed, Passive, Generic Network Activity Logging Architecture Proof of Concept (Phase I)

Erin Johnson, John Koenig, Darren Kulp, Joe Myre, Dave Rush

{johnsone, koenigjm, kulpdm, myrejm, rushd} @ uwec.edu

Abstract

Echos is a distributed, passive, generic network activity logging architecture. The goal of Phase I was to implement a basic proof of concept for this architecture.

Introduction

Network activity logging is a balancing act where the amount of data collected must be weighed against how much time and how many resources are required to analyze that data [1]. To complicate matters, if a system is successfully compromised, attackers often sanitize systems' logs [2].

Echos

In order to address these two issues, the Echos project was developed as a distributed, passive, generic network activity logging architecture. A distributed implementation enables logging at critical points in the network as well as allowing data across multiple networks to be easily analyzed. The passive approach lessens the risk of detection and thus provides a greater level of confidence in the data's validity. By making the architecture generic, the recorded data can be used in varying domains such as network load analysis and incident response.

The Architecture

The Echos architecture is broken down into two components: one or more collectors and one aggregator. Collectors passively copy packets off of the network, perform minimal header processing, write the data to file, and then send that file to the aggregator. The aggregator can then perform a variety of tasks dictated by the users.

Collectors

The collector, in part inspired by The HoneyNet Project's Honeywall tool, is a physical appliance that acts as a layer two-bridge that is placed in between mission critical systems and the network [3]. When in Promiscuous Mode, the collector logs all traffic. While in Target Mode, only traffic to and from a specific IP is logged. The collector performs basic header processing and then writes that data,

as well as the raw packet, to file. That file is then sent to the aggregator. To assure that the collector is simply experiencing a lull in traffic and not malfunctioning, the collector also sends timed "heartbeat" packets.

Aggregators

Aggregators receive collector files. The tasks performed on the collected data are determined by the user. The data files can be parsed into a database or, because the raw packets are intact, the data can be transformed and then loaded into a tool such as Wireshark [4]. The aggregator also watches for collector "heartbeat" packets and throws an alert if one is not received within an acceptable time frame.

Phase I Activities

The goal of Phase I was to develop the Echos architecture and implement a basic collector and aggregator. Due to this project being wholly independent of any organization, testing was performed on the investigators' home networks. These tests yielded a working prototype.

Collectors

The collector was developed native to FreeBSD 6.2 using Perl 5.8.8 utilizing libpcap [5]. Data files are sent to the aggregator using a third network interface.

Aggregators

The aggregator was written in C, and is also native to FreeBSD 6.2.

References

- [1] Mauro, D. R. and Schmidt, K.J. 2001. *SNMP*. O'Reilly.
- [2] Young, S. and Aitel, D. 2003. *The Hackers Handbook*. CRC Press.
- [3] Honeywall. <http://www.honeynet.org/tools/cdrom/>. [Aug. 26, 2006].
- [4] Wireshark <http://www.wireshark.org/>. [Accessed: Aug. 02, 2007].
- [5] libpcap. <http://sourceforge.net/projects/libpcap/>. [Accessed: Aug. 02, 2007].