This material is based upon work supported by the Defense
Advanced Research Projects Agency (DARPA) under Contract No.
HR001119C0102. The opinions, findings, and conclusions stated
herein are those of the authors and do not necessarily reflect
those of DARPA.

Zeek tools

Introduction

Zeek https://zeek.org is mature and widely-used suite of tools for summarizing and analyzing network traffic. Although Zeek is not primarily intended for capturing data IBR, it does provide a mechanism for capturing some of the information that we use.

This directory contains scripts for scanning through the Zeek connection logs, extracting information about the IBR observed entering a network, and storing it in a CSV format that can be used by some of our other tools. The data gathered in this manner is often somewhat incomplete, but there is enough data, in many cases, to do useful analysis.

Parameters

The file `default.params` in this directory describes the parameters that are used by the other scripts in this directory.

Tools

`zeek2csv`

This tool reads a Zeek connection log from stdin, finds records that appear to be IBR (connections that appear to have a single, incoming packet) and writes CSV records for each IBR packet to CSV.

The output CSV is the prefix of the CSV format created by `cpcap2csv` (for creating CSV directly from pcap files). It contains just the first eleven fields of the `cpcap2csv` format, and the IPID, TTL, and DATE fields are always set to a sentinel value of -1. (See `zeek2csv` for more information.)

`zeek-find-ibr.sh`

This tool is a wrapper around `zeek2csv`, which feeds Zeek connection logs into `zeek2csv` and stores the output according to the parameters it is passed (via environment variables or a parameter file).

`zeek-find-meanie.sh`

This tool uses a simple heuristic to find the most likely Meanie port for the given CSV, read from stdin. (See `../meanies` for more information about Meanie ports.)

This heuristic does not use all of the information we have about the Meanie, but it is good at detecting the Meanie quickly large subnet.

`zeek-extract-meanie-csv.sh`

Extracts the rows describing Meanie packets from the CSV files created by `zeek-find-ibr.sh`, and stores them in their own CSV files for later analysis.

Note that this script *does not* find the Meanie port for the current day, if it is not already known. It uses the Meanie ports that are *already* known and written to the file defined by the `ZEEK_MEANIE_PORTS` parameter. See `zeek_cron_meanie` for an example of a script that invokes `zeek-find-meanie.sh` to find the Meanie port for a given hour, updates the `$ZEEK_MEANIE_PORTS` file, and then invokes `zeek-extract-meanie-csv.sh`.

`zeek_cron_summ`

An example script that can be run as a `cron` job to run `zeek-find-ibr.sh` every hour, after Zeek rotates its logs.

`zeek_cron_meanie`

An example script that can be run as a `cron` job to run `zeek-extract-meanie.csv.sh` every hour, after Zeek rotates its logs.

Issues

Our experience with scanning Zeek connection logs to find IBR has had some issues. The first is that Zeek sometimes combines multiple packets into a single flow, even if no flow really exists. For example, if there is a scanner that sends three packets to the same five-tuple in a short period of time, Zeek will think (justifiably, given its set of assumptions) that all three of these packets are part of the same flow, rather than recognizing them as three separate probes. Unfortunately there is not enough information in the connection log to reconstruct the original packets, and our heuristics choose to drop all information rather than attempt to make inferences from the available information. This means that some scanners are "hidden" from our heuristics.

A second problem is not known to be specific to Zeek, but instead is attributable to provisioning. The Zeek server we were able to test on did not have the throughput to keep up with the processing demands of Zeek and all of the other tasks it was required to do, and at moments when other jobs were running (such as a storage-hungry backup task) Zeek was unable to keep up with the incoming packet stream. During heavy loads, our Zeek server fell behind and discarded as much as 90% of its input packets. If you want to gather accurate information from your Zeek connection logs, the first step is to make sure that the connection logs are complete – make sure that Zeek can keep up with the incoming data!