

Anomaly Detection of Web-based Attacks

Steve Hanna

April 18, 2008

Stepping Stones

Context

To review the context of the discussion: We would like to determine if given connections $A \rightarrow B$ and $C \rightarrow D$, that $B \rightarrow C$ exists, for A and D outside of the cloud being monitored and B and C inside cloud. We assume that we can only monitor input and output from cloud. Here cloud is used as a term for a nebulous network where internals are not known; we can only examine links into and out from the network.

Keystrokes

Recall that keystrokes in aggregation approximate the Pareto distribution (note: infinite variance).

25% keystrokes \geq 500ms

15% keystrokes \geq 1sec

1.6% leystrokes \geq 10sec

Remember that in detection research, possible evasions don't make a paper worthless. Unless the attack is trivial, the fact that we've made it harder for attackers suffices for a contribution.

Filtering

We filter in the following way:

Interactive Only, 22(SSH), 23 (Telnet), 513 (Rlogin)

Small packets

On/Off Transitions

Idle for ΔT second implies off. $\Delta T \approx 500\text{ms}$.

The transition from Off to On will occur on A and D at the same time. Therefore we need to look for On/Off coincidences. Consider δ up to 80 ms. Because of the fact we only look Off to On transition, we have relatively little work.

Basic Detection Algorithm

$\text{Off}_{1,2}/(\min(\text{off}_1, \text{off}_2)) \geq 30\%$

Flow in one direction, counting done on both ends, $1 \rightarrow 2$ $2 \rightarrow 1$

Look for either 2 or 4 consecutive coincides. 2 is for $A \rightarrow B \rightarrow D$ case, 4 is for $A \rightarrow B \rightarrow C \rightarrow D$.

Methods of gaining truth from the observations

Ground Truth

Analyze all lines of text, find lines occurring twice. Twice because we examine the route from A into the cloud, and then out of the cloud to D . In any true stepping stone, there will likely be lines repeated that didn't occur in other sessions. Consider those repeated twice that do not occur in any other session. These are candidates for inspection.

Cheap Hack #1

Find those connections with the same `$DISPLAY` environment variable. It is propagated.

However: often find `localhost`, so not very useful.

Cheap Hack #2

Examine string “Last login from $a.b.c.d$ at $time$ ”; should be unique per login session, and propagated on any stepping-stone login.

Evaluation

Results taken at two sites. Numbered results indicate number of stepping stones detected.

LBL

Brute Force: 23

Timing: 21

UCB

Brute Force: 47

Timing: 74

\$DISPLAY: 3

login string: 20

Lessons

This isn't an actionable detection because a lot of people do this as a part of their daily lives for completely benign reasons. It turns out pretty much every site has this property. *However* it provides a high-quality determination of a (fairly rare) type of activity, which can be incorporated into further analysis. For example, at LBL the detection of a stepping stone leads to recording of the contents of the session.

Web Based Attack Detection

This is an exemplar paper. They bring in a lot of the domain structure into the problem. They examine parameters and identify problems based on that. Other work in the space often uses completely unrelated data, like IP header fields. Important to analyze at the application later. Additionally there are a wide variety of statistical arguments.

Methodology

Detection stuff could be done in real time but in this case they analyze logs. They only go after GET requests and they say that they can generalize to other forms POST, headers, etc. They only consider success HTTP return codes. These seem like reasonable limitations given that the main contribution is methodology — remember, their goal is to flag as actionable, not block traffic.

Their learning techniques train on polluted data. This is a generally hard problem, how can we obtain non-polluted data to improve our learning modes? This is a hard problem. An upcoming Oakland S&P has a paper entitled **Casting out the Demons** relating to sanitizing data for correct analysis.

Attributes

A discussion follows regarding the attributes used in the paper.

Context

Recall that a request has the following format:

`http://host/foo.cgi?arg1=something&arg2=somethingelse`

Attribute Length

Look at the length of all of the different arguments and compute the mean and variance.

They use *Chebyshev's Inequality*:

$$\Pr(|X - \mu| \geq k\sigma) < \frac{1}{k^2}$$

This reads that the difference between a sample and the sample mean will exceed t times the standard deviation with probability less than $1/t^2$. It's a very general result (applies regardless of distribution), though also a *weak* result (usually the probability is much smaller than the bound). However, in this case the weakness is an asset: it means that if we find an attribute that is improbable according to Chebyshev's Inequality, then it's *definitely* quite improbable.