Network delay as a IDS reponse

Christopher Wee wee@cs.ucdavis.edu

2 September, 1998

1. Introduction

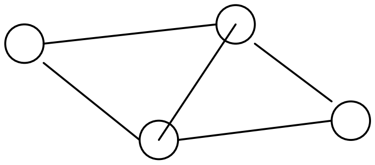
This is a working document. Please send your comments to the author.

So far, the only response we have considered is to block connections across links when intrusions are detected. The following are some variations on blocking traffic including a much more general idea of delaying traffic.

- Block new connections, but not existing connections.
- Delay all packets across a link a “d” amt of time
- Delay the formation of a new connection (SYN) by “d” amt of time.
- Allow new connections to form quickly, but delay rest of connection by “d” amt of time.

To implement the various delay model, we must detect and store different amounts of state/memory about connections & packets.

Adding delay to a network connection is a novel response to detected intrusions. In figure 1, the network circuit (link) is used for EDI transactions. The end-points of the network connections are programs that operate in batch-mode, exchanging records and like. Suppose that an intruder is using the circuit to interactively connect to a vulnerable host.



1.1 questions

What is the effect of latency upon application level behavior?

How to measure the impact of tolerable application level behavior

Real audio

Video

Jitter or distribution of delay

Latency distributions that encourage or discourage new connections from forming

How do we determine the threshold at which users will employ certain types of applications, e.g., interative telnet, POP mail, WWW surfing, ssh (or other encrypted logins), VPNs, file transfer, video conferencing, audio conferencing, X sessions.

Can we perform trace level simulations?

Find a paper that describes the impact of delay upon applications Find a paper that describes the impact of delay in a network.

Most importantly, we must have a set of metrics (or equivalent, a set of methods to measure) the impact of delay or blocking upon applications. The important aspect is “What is the effect upon the user?”

Rather than respond to attacks, we may also break the response chain into several steps

When considering approaches that utilize a set of pre-defined responses (e.g., expert system rules or table lookup), we must argue forcefully that there exists some expert (or method to draw out of an expert) the set of reasonable or even optimal responses to different situations. (Personally, I don’t have any hope for this approach because I do not believe that such experts exist).

```
detect        model of network          model of
attacks -->   capacity & failures --> impact upon --> responses
                                        services
```

2. Applications

Some applications that we should consider (I imagine that these applications exist in a C3I system).

2.1 database applications

communications is between client and server

- server-to-server database transactions may also be considered

metrics: delay, bandwidth of results returned, number of queries satisfied, number of failures because server cannot be contacted

2.2 Telnet-like interative applications

2.3 Network management applications

SNMP, polling, transaction oriented

2.4 Web browsers accessing web servers

2.5 Naming

Name services (e.g., DNS, YP, host-file) usually comprise name lookups and bulk name map transfers (in DNS, known as zone transfers). We can characterize the performance of naming by counting the number of name lookups that succeed or fail. We can measure the speed (time) of each lookup and compute the average and standard deviations of lookup response. For zone transfers, we can compute the bandwidth. Response time is probably not important for bulk data transfers. Finally, we would like to know the proportion of all network communications from a host that are devoted to name services.

2.6 Routing

Routing protocols

probes

2.7 file systems

client lookups

server-to-server backups

3. Performance metrics

Some first order metrics by which to consider how applications are affected. I feel it is important to develop credible methods that directly measure the applications performance according to these criteria. Otherwise, we may consider other metrics that are indirectly correlated to these metrics

- Data integrity
- Data privacy
- Availability especially timeliness

3.1 e.g. policy stmt on timeliness

if TELNET keystrokes to not echo with 1/2 second, then application behavior is not acceptable to user

if a database request is not satisfied within 10 sec, then app is not timely

3.2 Expectation profiles

instead of encoding a hard time constant (e.g., 1/2 sec or 10secs) into the policy statement, we instead encode as a % of an expectation paramter.

if database request not satisfied within 110% of expectation, then app not timely

Expectation are:

1) drawn from experts,
2) randomly chosen by researchers,
3) derived from user surveys, or
4) measured from actual application trials.

Expectations may be further parameterized by session, user, application

3.3 Methods to measure or characterize expectation

One suggestion for how to measure expectation from users is to provide a “I’m frustrated” button to applications. Users must be educated on how and when to press that button, i.e., button registers the fact that the user feels that the response is sufficiently timely, but not that the data is incorrect.

When is an application behaving correctly? When does a web search engine is returning the “correct” responses?

Application auditing/logging may also be a good way to measure application performance by providing intra-application, sub-task measurements of time, size and type of data processed, etc.

4. Conclusions