HTTP Time Bandit

Tigran Gevorgyan & Vaagn Toukharian

Who?

Tigran Gevorgyan
-Engineering Manager
Qualys

Vaagn Toukharian
-Principal Engineer
Qualys

- We fix stuff & accidentally break things
- Interested in time travel
- Love to tri (swim/bike/run)

What?

Yet another application layer DOS attack that strives for resource starvation through asymmetric resource utilization.

- Method
- Tool
- Stats
- Usage possibilities
- Defence

Why?

Classic Application Layer DOS/DDOS

DDOSing blindly

- GET index.html
- Repeat the above
- No feedback
- Symmetrical load

Smarter Bots

- SlowLoris
- Slowhttptest
- SlowRead
- PKI abuse
- SQL wildcards
- WebSockets connection hogging

The Proposed Method

Method of detection of the critical resource

- Spider over the web site and collect transfer times for each resource
- Calculate the average speed and distribution of transfers
- Identify the resources that have slower average transfer times

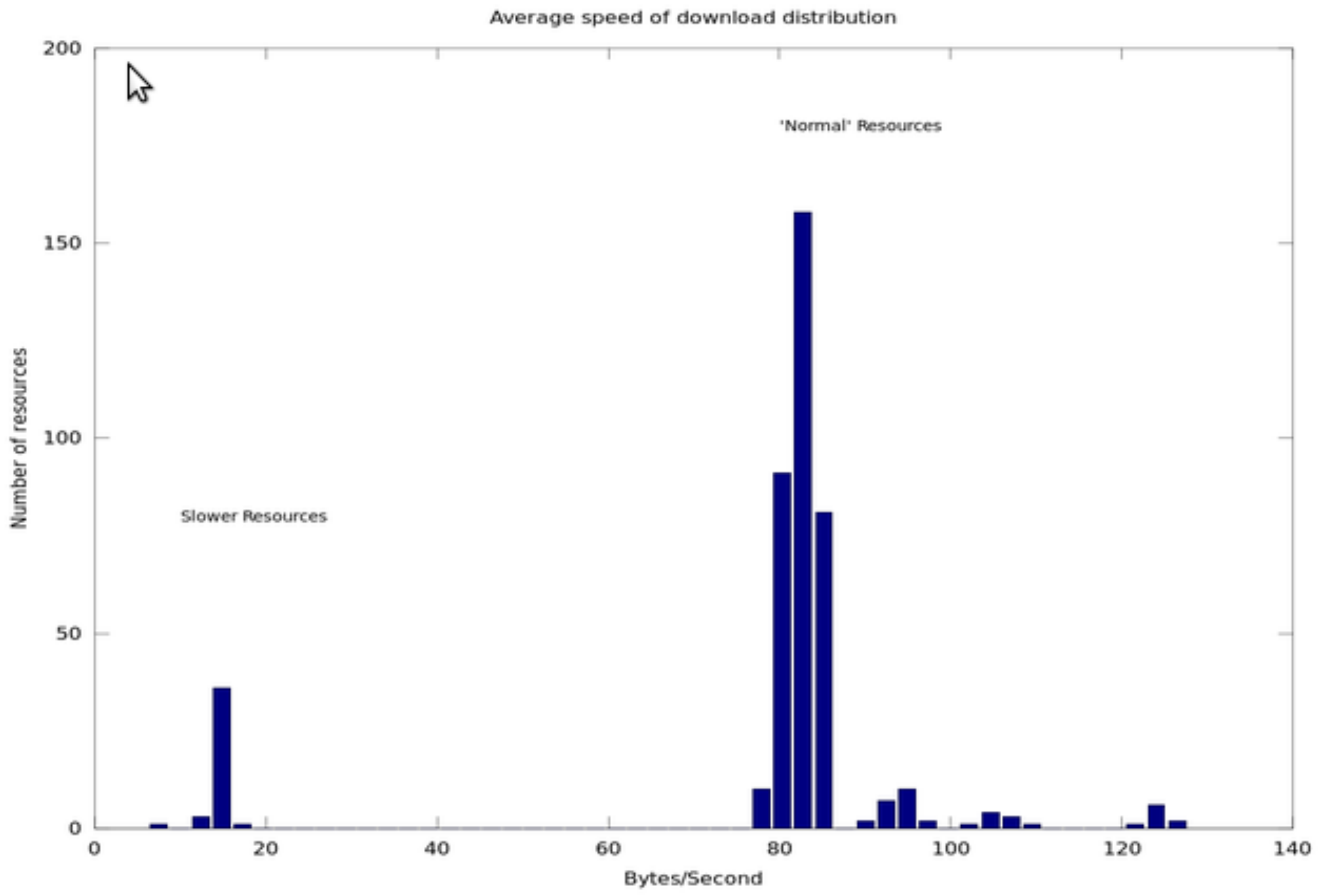
Transfer time's correlation with load

- CPU intensive resources take more time to response
- Resource size is not significant

Using statistics to normalize the data

- Mean as the measure of central tendency
  - Calculate the mean of all resource download speeds
  - Calculate the means of each resource download speeds
  - Select the resources whose download speeds are less (slower) than the mean of all download speeds
- Selecting resources with lower mean
- Discarding resources with large variance

Some Graphs


Average speed of download distribution
'Normal' Resources
Slower Resources
Number of resources
Bytes/Second
0
50
100
150
200
0
20
40
60
80
100
120
140

Demo

of HTTP Time Bandit

Usage

of HTTP Time Bandit

The Good

Find potential CPU/DB hogs in my web apps

The Bad

Automated iterative analyzer attacker

The Ugly

Probably should not be spelled out:)

Imagine “The Bad” x 1000

Back to the future

- Attack like stage of testing
    - Measurement of service degradation while doing a hard test for narrowing down the choice of links
- Understanding Load Balancers
- SQL wildcard usage
- State Reset cost analysis

Defence

- Load Balancing
- Identify/Fix resource hogs
- Simple mod_security protection [1]
- Advanced mos_security protection
  - Identification of regular flows
  - Out of ordinary flow filtering
  - State coherence checks

Thank you

tgevorgyan@qualys.com
vtoukharian@qualys.com

References

1. http://blog.cherouvim.com/simple-dos-protection-with-mod_security/