Identifying DDoS Attacks Using Flexible Server Logs

Title Slide

* Consumer internet bandwidth
* You can consume more information faster
* When expected, all is good
* When unexpected, serious negative side effects
* An organized exploitation is DDoS
* I will explain
  + How DDoS impacts you
  + What a DDoS is exactly
  + How I can detect them before negative impacts
* <CLICK>

Context Slide

* How would a DDoS affect you?
* Working on an 8.01 PSet, need to Google a constant
* You open your browser and…
* <CLICK>
* Get this message
* After intense troubleshooting, not your internet connection
* Other websites load
* Potential cause: DDoS
* <CLICK>

Background Slide

* So what is a DDoS?
* <CLICK>
* Several computers trying to access a webserver
* Not all at same time
* <CLICK>
* Maybe this one needs a webpage
* <CLICK>
* Then this one needs a picture
* <CLICK>
* Then this one needs a video
* All other computers remain idle
* <CLICK>
* Now suppose all these computers are constantly making bogus requests
* <CLICK>
* Then a legitimate user comes and needs a webpage
* <CLICK>
* But they can’t get through
* <CLICK>
* These malicious computers, are executing a DDoS
* You are stuck with the error message
* <CLICK>

Contribution Slide

* How do we combat DDoS attacks?
* Many complicated methods
* I focused on first step: ease of detection
* Through detailed, flexible logging, detection is simple
* <CLICK>
* My program takes user supplied format sentence
* Parses the format sentence determining what the user is looking for
* Parses all network traffic
* Sorts out all unnecessary information
* <CLICK>
* Outputs parsed information according to user format
* In this example, it is clear to recognize a DDoS attack
* The same users making thousands of requests per second
* Not waiting for server response before requesting
* <CLICK>

Conclusion

* But we haven’t done anything to stop DDoS!
* No, but we did make the first major step
* Now this information can be fed into a relational database or program
* Firewall rules can be temporarily applied to block users
* The beauty of it is: nothing is hardcoded
* Everything is
  + Easily modified
  + Supremely flexible
* Next time you get the took too long to respond error, there may be a DDoS
* That DDoS could’ve been detected by my program