ISCX
Information Security
Centre of Excellence

The impact of slow-rate application DDoS attacks on webservers

H. Gonzalez, M. Gosselin-Lavigne, N. Stakhanova and A. Ghorbani

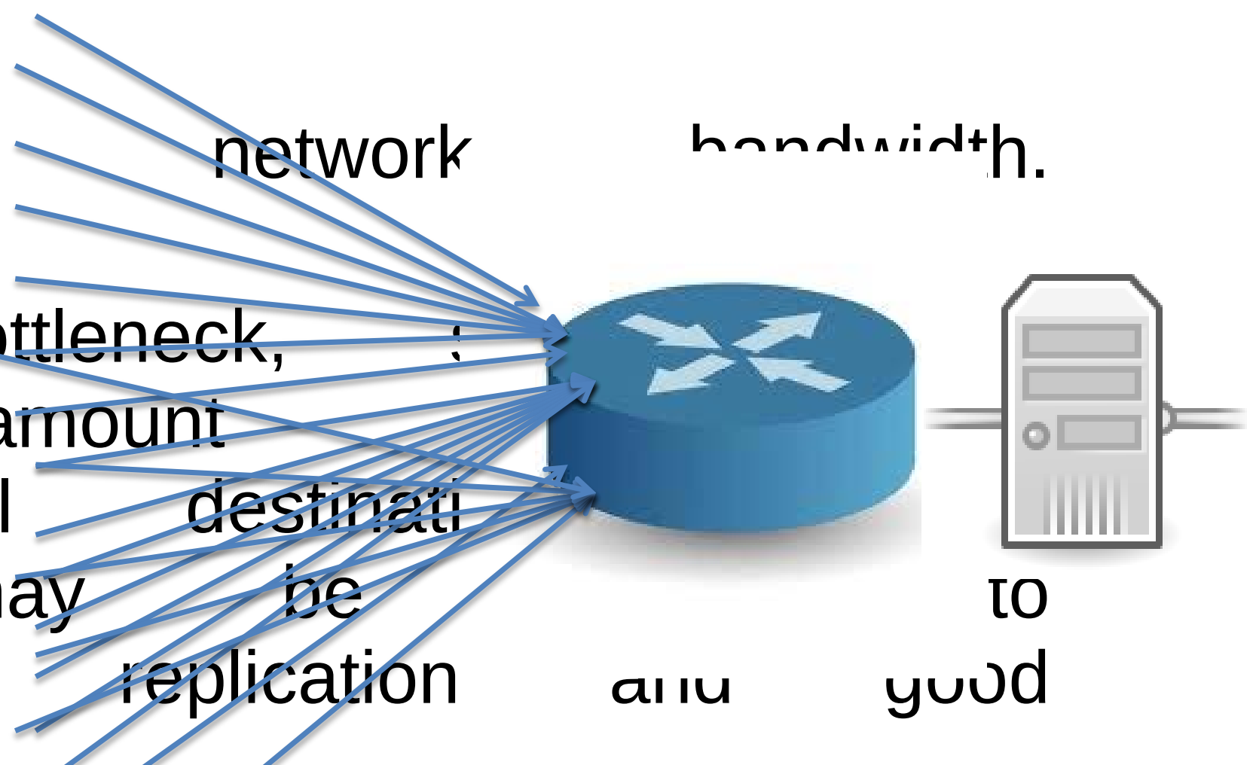
Faculty of Computer Science, University of new Brunswick

UNB

Introduction

- DDoS attacks are an old problem but continue to be a threat. Typically a DDoS attack tries to exhaust the resources from the victim, either a computer or a network, by using brute-force bandwidth flooding.There attacks attempt to block access to legitimate users.

- DDoS attacks were classified by victim type by Mirkovic as:
  - **Infrastructure**: The targets include core routers and critical services like DNS. Easy to detect due to high traffic and inoperability.
  - **Network**: The target is the network bandwidth. Easy to detect due to high traffic and inoperability.
  - **Resource**: The target is a bottleneck, router, where a large amount confluences even when the final destinati specifically the target. This may be to detect. It can be prevented by replication and good design of the network.



  - **Host**: The target of this type of attack is a host, preventing access to the host. The method can vary from flooding to attempting to cause the machine to crash. Easy to detect since it causes a host to go offline.
  - **Application**: The target is a specific service or application. The host can still be reached but the service or application cannot answer legitimate traffic.

- The most common, well known and effective techniques to perpetrate a DDoS attack is flooding. It could be performed at different layers, such as the Internet, transport or application layer. Flooding attacks can be mitigated by blocking and dropping packets from clients who's traffic increases.

- The HTTP GET flooding attack is considered by ** to be one of the most successful attack on application layer denial of service, because the detection is difficult due to its non-intrusive and evolving nature.

Partial results from default configurations

- How many resources does the attacker needs to take down a webserver with the default configuration?
- Easy : with only one PC in less than 20 seconds a server could be taken down.
- Hard: the low rate becomes a flooding attack to become effective.

	Slowheaders	Slowpost	Slowread
Apache / Linux	easy	easy	easy
Apache / Windows	easy	easy	easy
Tomcat / Linux	easy	easy	easy
Tomcat / Windows	hard	hard	hard
Nginx / Linux	hard	hard	hard
Nginx / Windows	hard	hard	hard
IIS / Windows	hard	hard	hard

Motivations

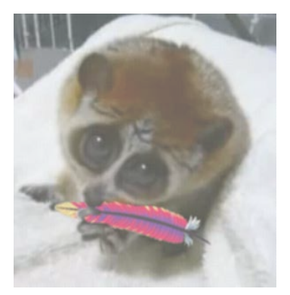
- Since web servers are now a very crucial part of the Internet, new kinds of DDoS attacks are gaining more attention because they try to exhaust the resources of the webserver without disrupting other services. They can also be executed with a small amount of bandwidth. Such attack is called a slow rate attack.

- We believe that the results of these attacks depends on the victim's software and its implementation, so we tested common webservers under different operating systems using actual slow rate attack tools.

Objective

Test different combinations of webservers (apache, nginx, tomcat, IIS), operating systems (Windows, Linux) and configurations (basic, hardener) against different attacks and explain the behaviour in each case.

Attacks beyond simple application flooding



Slow headers: (Slowloris tool) The attacker continually sends headers to the webserver just before the connection expire. This maintains the connection, committing resources. This attack is independent of the content of the website.

Slow post: (Rudy tool) The attacker sends a large amount of data on a single POST request, sending data just before the connection expires, maintain the connection open and resources busy. This attack requires a valid form on the website.

Slow read: (Slowhttptest tool) The attacker ask for a large file from the web server, but consume the data very slow. This keeps the connection open and the resources busy. The attack requires a big file on the website.

Partial findings

- The tools and attacks appear to have been designed for Apache.
- Nginx use a different approach for managing connections. It appears to be a good one.
- Tomcat was a surprise as it displays different behaviour on different platforms. We think that the implementation of Java may have something to do with that.
- IIS is hard to take down with a slow rate attacks.

Work in progress

1. Improve the configuration of the Apache webserver to resist the attacks.
2. Finding the differences with Tomcat on Linux and Windows.