Denial of Service (DOS)

Lester Dela Cruz

Computer Engineer, Santa Barbara City College

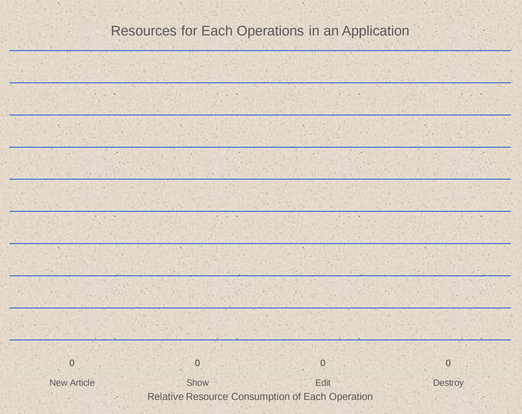
Adam Doupe - Christopher Kruegel, Giovanni Vigna, Richard Kemmerer - Computer Science

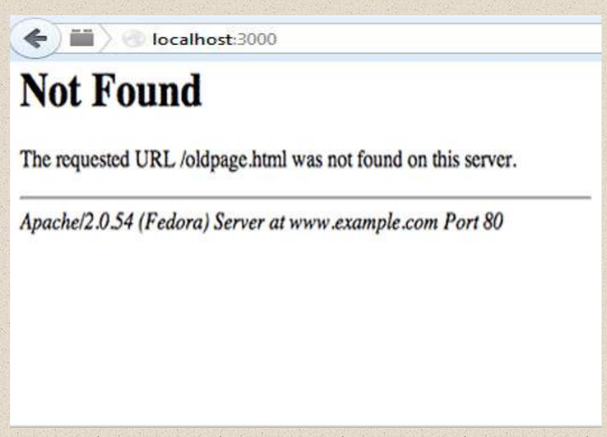
Abstract

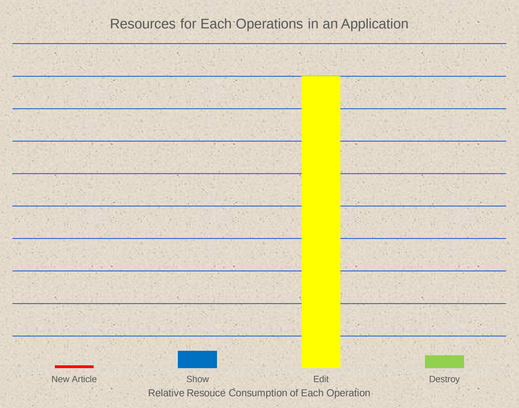
Denial of Service is a popular attack used for crashing websites. Companies that depend on the availability of their website are vulnerable to this type of attack, furthermore this attack is relatively simple to implement. Hackers implement this attack by first probing for parts of the website that consume the most website resources. Then, the hackers flood that part of the website to the point where the website consumes too many resources and crashes. Currently, when a company is under a Denial of Service attack, the most popular defense is to manually remove the part that is being attacked while rendering the rest of the website. However, this defense is expensive to implement and requires constant monitoring. This project attempts to solve these two downsides. Our goal for this project is to build a framework that automates the selection of the resource-hogging website functionality, and automatically removes them while rendering the rest of the website the same.

Problem

A website is composed of parts that take different amounts of website resources. This causes websites to have a dangerous vulnerability because website resources are limited. Hackers can then identify the part of the website that takes the most resources and overload that part until it consumes all the website resources which then causes the website to crash.







Approach

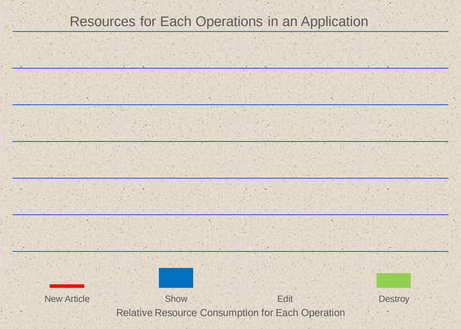
1. First, we need to launch our own web applications for testing.

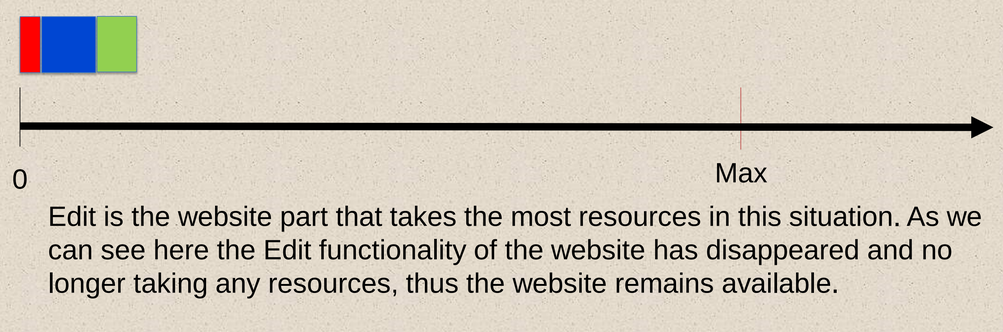
2. Second, we write a program that automates the selection of resources-hogging parts on the web application then removes them.

3. Once the part is removed, the rest of the website should be perfectly functional.

Expected Results

The project is still in its early stages therefore not all criteria has been met yet. However, here we have the same website. We want that website to still work even though the resource consumption tolerance has been exceeded. The following graph will demonstrate this idea.





Edit is the website part that takes the most resources in this situation. As we can see here the Edit functionality of the website has disappeared and no longer taking any resources, thus the website remains available.

Conclusions

Denial of Service can inflict major damages on businesses that depend on the availability of their websites. Hackers take advantage of the resource consumption of parts of a website and overload it until the resource consumption exceeds the limit. As a result, the website crashes. However, we believe that if we can create a software that integrates with a website that allows us to remove parts of the website without damaging the others, combined with an algorithm that selects which parts of the website that consume the most resources, it would be a viable defense against the Denial of Service attack. This way, when a website is under a Denial of Service attack, we can simply and automatically remove that part that takes the most resources, allowing the website to handle much more resources and survive the attack.

Bibliography

Client-Side Distributed Denial-of-Service: Valid Campaign Tactic or Terrorist or Terrorist Act? *DJNZ and The Action Tool Development Group of the Electrohippies Collective.* The MIT Press. *Leonardo,* Vol. 34, No.3 (2001), pp. 269-274.