# Internship and thesis

Today more and more services are offered as a web applications and no longer as a software distribution (that has to be installed and configured on the computer of the user). Web applications are easier to develop, flexible, independent from the operating system of the user and can be accessed from everywhere. But it has one major pitfall, they are very susceptible to cyber-attacks. Attackers can perform a devastating attack (from everywhere) on a web application developed by a less experienced developer which didn’t implement all the required security measures. Inexperienced developers are the main source of web application vulnerabilities. This is where the web application firewall comes in. This device can protect (multiple) web applications from attacks. It does this by recognizing specific patterns in requests that could be part of an attack and preventing this malicious from reaching the web application. For example: requests that contain JavaScript code in the query string can be blocked to prevent XSS, requests that contain the MySQL ‘UNION’ statement can be blocked to prevent SQL-Injection.

The web application firewall looks like perfect solution in preventing web based cyber-attacks but it has its downsides. The biggest flaw is that they are really difficult to configure in order to function correctly. What if a certain application requires JavaScript code in the query string or the MySQL UNION statement? These are normally blocked by the web application firewall and thus will prevent the web application from functioning correctly. This frustration among developers can lead to full out disabling the web application firewall making all the web applications vulnerable again.

In order to eliminate the complex process of configuring the static rules a web application firewall based on anomaly detection can be developed. This type of firewall doesn’t need any static configuration but will “learn” what type of requests are safe/legitimate and which are malicious. It does this by first observing legitimate requests (profiling of the application) and then later comparing an incoming request with the profile of legitimate requests to determine if the request can be marked as legitimate. This kind of mindset will not only adapt itself when the web application changes but should also able to detect zero-day attacks.

During this internship an application profiler and anomaly detection engine will be developed in order to proof the power of this type of web application firewall.

Index.php?page=1

Request 1

Index.php?page=’OR1=1#

Request 2

Index.php?page=az3

Request 3

Index.php (The “page” parameter normally has a numerical value)

Profile

NOT SAFE

Request 3

NOT SAFE

Request 2

SAFE

Request 1