# Introduction

## Summary

The aim of this project is to develop a vulnerability scanner for web applications that can, in an automated fashion and with minimal user input, scan a web application and generate a report detailing any vulnerabilities found along with general recommendations to mitigate them.

The program will be as minimalistic and easy to use as possible, so that a person without any technical know-how might be able to run it.

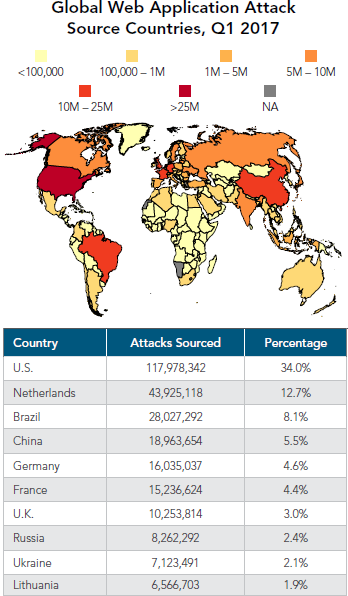
## Background

Computer security has always been a major concern in the field of technology. Humans first tried to purposely exploit vulnerabilities in networks back in the 60s, and then through the 70s, people would develop methods to make free phone calls by replicating the tone required to unlock the AT&T’s phone network. Worms and viruses would first appear years later, created by modifying programs intended to help computers. In the coming decades, self-modifying viruses would be developed, and new methods for massive world-wide distribution would be conceived. Attacks would extend even to government and military computer systems[1][2].

In the mid-2000s, the quick growth of the Internet would only help increase the reach of virus infections thanks to emails and social media, and credit and debit cards would become some of the new targets for attacks[3].

Nowadays, with the advent of the Internet of Things and the growing globalization, the issue of computer security has only grown in importance[4]. Viruses, malware and many other varied threats and attacks can exploit vulnerabilities in a system, requiring appropriate defenses from users and companies in order to avoid information theft or financial harm. Web applications have become a prominent target for attacks due to the multiple vectors of attack that exist and the fact that they tend to hold sensitive or valuable information.

The world has seen a 35% increase in total web application attacks in the first quarter of 2017 compared to 2016. As reference, the United States is both the most targeted country for web application attacks as well as their greater source. The United States accounted for roughly 34% of all web application attacks worldwide in the first quarter of 2017, up from 28% in the previous quarter. The list of most common sources for attacks after the United States consists of Netherlands, Brazil, China, Germany, France, United Kingdom, Russia, Ukraine, and Lithuania[5], as can be seen in the following figure:

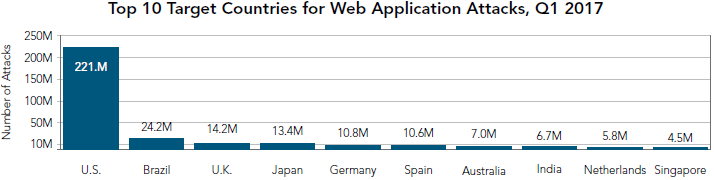


1 Global Web Application Attack Source Countries, Q1 2017

[Source: Akamai’s State of the Internet Security Report]

The Netherlands is the top source for web application attacks in Eurasia, as well as the 2nd worldwide, which is noteworthy given its relatively small population density. In comparison, the United States has nearly twenty times the population but is responsible only for roughly three times the number of attacks than the Netherlands.

As for the targets of the attacks, the United States leads the list as commented above, followed by Brazil, United Kingdom, Japan, Germany, Spain, Australia, India, Netherlands, and Singapore:



2 Top 10 Target Countries for Web Application Attacks, Q1 2017

[Source: Akamai’s State of the Internet Security Report]

As can be seen in the graphic, the attacks targeting the United States account for a vast majority of all attacks (nearly 70%), with the United States receiving almost ten times the number of attacks of the country in second place (Brazil). Even then, the number of attacks targeting the United States is actually down 9% compared to the last quarter. Conversely, Brazil and the United Kingdom have seen an increase in attacks of a 46% and a 30% respectively compared to the last quarter.

China and Canada, which appeared in this list in the last quarter, have been replaced by Spain and Singapore, both of which have been in this list in the past. These sudden swings may seem significant, but they are within the norms usually expected from such traffic.

## Motivation

Penetration testing, or pen-testing, is a controlled attack on a system with the aim of finding vulnerabilities the way an attacker would, in order to neutralize them. While it is an effective method to find breaches, many companies forgo its use due to its high cost, which might range from anything from $4,000 up to $20,000 (roughly from 3,500€ to 18,000€)[6].

Studies state that nearly half of breaches exceeded $500,000 in losses in 2015. A survey sent to security professionals reveals that 24% of the respondents had experienced a data or system breach in the last year, and yet 8% of them never perform penetration testing[7].

Thus, while penetration testing is certainly expensive, the losses incurred because of insecure web applications are far steeper. The relatively low budget spent in web application testing compared to other types of security practices might denote an insufficient awareness of the severity of breaches related to web applications.

The purpose of this project is not so much to build an extensive high-end vulnerability scanner, but to raise awareness about the importance of proper security testing. If such a simple and inexpensive tool is able to detect vulnerabilities in a target web application, a skilled attacker with sophisticated tools might put it in jeopardy with much more ease.

## Objectives

This project will attempt to automate a simple security audit and generate a report, with the assistance of the OWASP ZAP vulnerability scanner. The device will try different attacks or payloads in an automated fashion to search for vulnerabilities in a target web application. Web applications are susceptible to diverse kinds of attacks, such as XSS (cross-site scripting) or SQL injection, and this makes them common targets for attackers. The findings will be saved into a portable document format (pdf) for later analysis.

While the results might not be comparable to those obtained by a manual pen-testing from a professional, it should be able to find at least common vulnerabilities, and for a very low price. If such a simple, inexpensive tool is capable of compromising a system, then it should become immediately evident that any malicious attacker would also be able exploit those vulnerabilities, and such a system should be deemed insecure and unreliable.

Along with the report, the program might offer a set of general recommendations to secure a typical SME infrastructure against the sort of attacks performed by the program, based on the kinds of vulnerabilities found in the target.

This sort of device is not intended to actually solve the issues found, but merely inform about them. As such, it would be designed in such a way that any salesperson should be able to use it without requiring previous technical knowledge.

This sort of simple vulnerability scan could be used to encourage companies to enlist the services of a professional pen-tester that could remediate the found issues, as well as any other problems that an automated scan is unable to detect, in order to avoid financial damage or theft of sensitive data.