

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

SECURITY AND APPLIED LOGIC

LUCRARE DE DISERTAȚIE

Python Web Application Scanner

COORDONATOR ȘTIINȚIFIC: ABSOLVENT:

NĂIBOIU TEODOR

BUCUREȘTI, SEPTEMBRIE 2024

**Contents**

[I. Introduction 4](#_Toc173424623)

[I.1. Motivation 4](#_Toc173424624)

[I.2. Personal contribution 5](#_Toc173424625)

[I.3. Structure 5](#_Toc173424626)

**Abstract**

The following project represents a scanner software application written in Python programming language, designed to aid developers, white hat hackers (security researchers, red-teamers, bounty hunters) and any other cybersecurity enthusiast, in detecting vulnerabilities and assessing the attack surface and risk exposure of a Web Application. This new implementation is a redesigned and enhanced version of my previous Web Application Vulnerability Scanner software presented in my under-graduate thesis [1]. This new implementation is designed for Linux *Operating System* (OS) usage only. Users can interact with the application through the *Command Line Interface* (CLI) and by providing the scanning arguments such as the target website, scan type and login details.

The new scanner functionality has been enhanced by refining more accurate detections and by adding new features such as rogue comment identification, hierarchical site map graph, external uniform resource locators (URLs) references, hidden URL paths and Common Vulnerabilities and Exposures (CVE) detection. The reporting method has also been changed to an easier to read Hypertext Markup Language (HTML) report. The new version of the scanner has been built for scalability and ease of access, as the number of prerequisite configurations has been decreased compared to the previous version. Additionally, the majority of ‘future enhancements’ mentioned in the previous thesis were implemented in the new version along a completely new method of scanner-website communication channel that increases reliability and time of execution.

# Introduction

Cybersecurity is the art of protecting networks, devices, and data from unauthorized access or criminal use [2]. This formal definition of cybersecurity can be pictured as a border between two armies of people whose whole purpose is to conquer territory in the other’s lands. Over the last years, the need of understanding and educating people on cyber security has been increasing. Nowadays, not only software engineers and security researchers are impacted daily by digital data polluted with malicious additions, but any other person with a smart phone or any other digital device connected to the internet.

On average every day, approximately 30000 new websites fall to hacking attacks [3]. As the numbers of websites is increasing due to demand and digitalization, so are the attacks aimed at stealing data by exploiting misconfigurations, vulnerabilities or by simple social engineering attacks facilitated by lack of access restrictions. Currently, more and more solutions are transitioning to web applications designed to provide the users with a fast resolution for everyday activities (for example: buying a bus ticket, ordering a meal from the local restaurant, tracking your parcels or accessing your bank balance). Due to this reason, the need of having a secure website is more important than ever.

## Motivation

I have decided to continue my previous implementation of a web application vulnerability scanner and to enhance its functionality due to my ongoing passion for web application security and for my personal projects. A web application is a computer program stored on a remote server and run by its users via a web browser [4]. As the web application is designed to be publicly accessible, attackers have an easy target available to lock on their scanners to and wait for low hanging fruits to be found. Due to this reason, I have decided to create a scanner similar with the tools and methods attackers are using to detect misconfigured websites over the internet.

As information is constantly changing, so are software solutions and their configurations. Due to the ongoing modifications in web application service configurations and their increased complexity, researchers and red-teamers can have difficulties in detecting vulnerabilities in a timely manner. This also causes automated vulnerability detection tools to become deprecated really fast. As my project is an automated detection tools as well, I have decided to further continue its development and update it to the latest security risks while also having scalability in mind. In the following pages, I will present the new features of this tool, as well as the implementation methods I have chosen for ensuring future updates are done in a timely manner.

## Personal contribution

While the previous version was much more focused on the Open Worldwide Application Security Project (OWASP) Top 10 Standards [5], the new version is based on my work obtained experience as a Security Analyst and on Capture the Flag (CTF) competitions experience and observations, over the last 3 years.

CTFs are gamified competitive cybersecurity events that are based on different challenges or aspects of information security [6]. The focus was shifted from a OWASP Top10 approach to a more subjective means of detections based on my past observations from live models of vulnerable web applications. Almost all of the detection methods and capabilities were implemented based on my best judgement instead of generic approaches. This was done to further increase the accuracy of detections while also decreasing the false-negative findings. Due to this approach, there might be a significant number of false-positive findings as well, hence the scanner results should be confirmed by manual replication.

Furthermore, I have decided to bring this scanner up to a competitive level by lowering the number of prerequisite configurations needed and by providing a human readable report with dynamic and intuitive UI.

## Structure

TBD after capthers

# Technologies and instruments

**References**

|  |  |
| --- | --- |
| [1] | T. Naiboiu, Web Application Vulnerability Scanner, Bucharest: University Of Bucharest, 2021. |
| [2] | CISA, 01 02 2021. [Online]. Available: https://www.cisa.gov/news-events/news/what-cybersecurity. [Accessed 01 08 2024]. |
| [3] | BUGTI, "Website Hacking Statistics 2024," BUGTI, 07 05 2024. [Online]. Available: https://www.bugtiwallet.com/website-hacking-statistics-2024/. [Accessed 01 08 2024]. |
| [4] | A. Volle, "Web application," [Online]. Available: https://www.britannica.com/topic/Web-application. [Accessed 01 08 2024]. |
| [5] | "OWASP," [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed 01 08 2024]. |