Oblivorian Vulnerability Scanner

Nicholas Garrett
Computer Science
Idaho State University
Pocatello, Idaho, United States
garrnic3@isu.edu

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

I. Introduction

When the first comercially available personal computer, the Kenbak-1, was released to the public in 1971 [1], the concept of computer viruses was only that—a concept. It was not until the spread of the Creeper Virus of that same year that the need for cyber security became evident. Prior to Creeper, popularly known as the first computer virus [2], the practical need for implementations to avoid vulnerabilities only contained bugfixing and avoiding system errors. However, with the rising popularity and prevalence of computer viruses being spread from system to system, the importance of maintaining security and preventing cyber attack is rising as well.

In this paper, I describe a software system I have built intended to detect common exploitable issues easily detectable on a computer system. I will also explain these possible exploits and how they pose issue and threat to computer systems. Each of the exploits I will describe—or methods to reach exploit—are useful to an attacker intending to gain access to a computer system.

II. Oblivorian

The software program I have built to solve some of these issues in cybersecurity is called Oblivorian. Oblivorian is a system written in Python3 designed to scan a system for some of the simple vulnerabilities seen.

As with almost any software system, Oblivorian is a continuously changing software program designed to change and improve over time. Hopefully, given enough time, this program will be able to scan for more advanced and difficult software vulnerabilities. Of course, it is unfortunately limited by my abilities in programming and by the extent of my understanding in cybersecurity principles. As I become more competant and knowledgable about cybersecurity and vulnerability scannning principles, I will add to Oblivorian.

Python was chosen for the flexible and adaptable nature of its code. While a lower-level language may have been a better choice for the deeper control it can provide, Python is much easier to program in. In the future, I may implement C or similar scripts to perform some of the scans. However, this is only a possible plan for the future.

In this paper, I will describe various vulnerabilities and how Oblivorian works to fix or detect these vulnerabilities.

III. Port Scanner

Internet ports are an important part of connecting your computer to the outside world. Through them information and data can be both sent and received to the internet. Through this, a personal computer in Japan is capable to sending data to another system somewhere else in the world. However, while ports must be open in order to connect to, this is a double-edged sword: every open port is a potential vulnerability that an attacker could exploit. An attacker needs only to gain access to a port and they can then possibly get information about your syestem and possibly open doors for the execution of malicious software. Therefore, in order to ensure our computer system is more secure than otherwise, we should make sure that only ports we want to be open remain so, and all non-actively used ports are closed.

However, what defines a used verses an unused port? Simply put, a used port is one in which the daemon, a program that runs in the background without input from the user, is still listening on a port. Therefore, by the logical conjugate, we can say that unused ports are those in which no daemon is listening for incomming traffic.

Our first thought might be to simply close all the ports we do not want to use. This would prevent attackers from retrieving information about our system. However, this impulsive action could lead to necessary ports being closed. Suppose we close the ports necessary for HTTP or HTTPS (80 and 443, respectively), suddenly we would find that internet browsing programs no longer work, as HTTP and HTTPS are important in transferring the data used in internet browsing.

Thus, we can see that closing ports without first seeing what they do or closing ports often used, can lead to the systems using those ports to malfunction or fail. Therefore,

we must be careful not only to close unused ports, but also to not close open ports that are used frequently.

Therefore, instead of atomatically closing open ports when they are found, if we list the ports and give to the user the option of what ports to close, giving to them an explanation of what the purpose of these ports are, the risk of closing ports best left open can be decreased and control can be put into the hands of the user.

Following this plan, it is quite possible that the list of open ports may be quite large. For example, Mac-OS websites state that as about 132 ports are open by default on many Apple services. To imrove user experience, it would be more convenient to somehow filter through ports that should be closed, only leaving ports with an active owner up to the user to close if wished.

Thus, Oblivorian's scanning feature scans through port numbers 1 through 65535. The method of searching for open ports is through attempting to connect to the ports in iterative sequence. If a connection is able to be made, then that indicates that a service is listening on that port. So, Oblivorian adds each port where a connection could be made into an array and asks the user whether or not to terminate all processes listening-in on that port.

IV. Sniffer

V. System information

VI. Serial Port Scanning

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [**?**]—do not use "Ref. [**?**]" or "reference [**?**]" except at the beginning of a sentence: "Reference [**?**] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [3]. Papers that have been accepted for publication should be cited as "in press" [4]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [5].

References

[1] Baqué, Achim. n.d. "The First Personal Computer." The First Personal Computer. Accessed November 30, 2021. http://www.thefirstpc.com/..

[2] "Creeper: The World's First Computer Virus." 2019. Exabeam. March 5, 2019. http://www.exabeam.com/information-security/creeper-computer-virus/.

[3] "What Is Port Scanning and How Does It Work? — Avast." n.d. Www.avast.com. https://www.avast.com/business/resources/what-is-port-scanning#pc.

[4] "SecurityTrails — What Are Open Ports?" Securitytrails.com, securitytrails.com/blog/open-ports. Accessed 30 Nov. 2021.

[5] YHartwig, Chris. "Why Closing Unused Server Ports Is Critical to Cyber Security." Blog.getcryptostopper.com, blog.getcryptostopper.com/why-closing-unused-server-ports-is-critical-to-cyber-security. Accessed 30 Nov. 2021.

[6] "Is It Possible in Python to Kill Process That Is Listening on Specific Port, for Example 8080?" Stack Overflow, stackoverflow.com/questions/20691258/is-it-possible-in-python-to-kill-process-that-is-listening-on-specific-port-for.. Accessed 30 Nov. 2021.