Chapter 4

Zero Day Vulnerability Assessment:

Exploit Detection and Various Design Patterns in Cyber Software

Vidhant Maan Thapa
University of Petroleum and Energy Studies, India

Sudhanshu Srivastava
University of Petroleum and Energy Studies, India

Shelly Garg
University of Petroleum and Energy Studies, India

ABSTRACT

In this technology-driven era, software development and maintenance are rapidly growing domains and are predestined to thrive over the coming decade. But the growing demand for software solutions also brings its own implications, and software vulnerabilities are the most crucial of these. Software vulnerabilities can be referred to as weaknesses or shortcomings of software solutions, which increase the risks of exploitation of resources and information. In the past few years, the number of exploits has been increasing rapidly, reaching an all-time high in 2021, affecting more than 100 million people worldwide. Even with the presence of existing vulnerability management models and highly secure tools and frameworks, software vulnerabilities are harder to identify and resolve as they may not be independent, and resolving them may cause other vulnerabilities. Moreover, a majority of the exploits are caused by known vulnerabilities and zero-day vulnerabilities.

DOI: 10.4018/978-1-6684-9018-1.ch004

INTRODUCTION

Zero-day vulnerabilities are the vulnerabilities that were previously unknown to the vulnerability management team (Dougherty et al., 2009). They have a high risk of being exploited even before identification. These turn into zero-day exploits, if vulnerabilities are exploited before mitigation (Bilge & Dumitraş, 2012). 2021 experienced exponential growth in these exploits with an estimate that more than 40 percent of these attacks occurred in the last year. From the 2006 Stuxnet attack to the 2019 Facebook and 2021 LinkedIn zero-day attacks, zero-day vulnerabilities are the cause of the majority of cyber attacks compromising the resources and information of millions of users (Chen et al., 2018; Kaushik et al., 2023; Khan & Mailewa, 2023; Nafees et al., 2018; Park et al., 2023; Yin et al., 2023).

Causes of Increased Zero-Day Exploits/Vulnerabilities

A major reason for an increasing number of zero-day vulnerabilities is the rising number of software solutions and updates that occur regularly. Although, these vulnerabilities are harder to detect before being exploited due to their unknown nature but often times developers do not really resolve these implications even identification as it may break other existing programs. This usually occurs due to ineffective design patterns or the existence of anti-patterns in the software solution. Moreover, large software solutions have hundreds (if not thousands) of existing vulnerabilities which may have higher risks of exploitation (Jaber & Fritsch, 2023). Furthermore, as a result of the increasing number of private companies that provide offensive cyber tools and services and malware vendors, global ransomware activity has escalated to a massive extent. This accessibility to developed exploit kits has hiked the number of zero-day exploits over the past few years (Blumbergs et al., 2023).

Handling Zero-Day Vulnerabilities

Due to the unknown nature and increasing existing presence in legacy and current software solutions, it's impossible to completely eradicate zero-day vulnerabilities but with specific practices and tools, we can reduce their growth, and codependencies and tackle these exploits much more effectively in the long run (Pérez-Díaz et al., 2023). For a secure software solution in regard to vulnerabilities and exploits we should successfully incorporate the following steps:

1. Preventing zero-day vulnerabilities to occur in the first place
2. Finding Existing Zero-day Vulnerabilities
3. Quick Recovery from zero-day exploits

Preventing Zero-Day Vulnerabilities to Occur in the First Place

One of the major reasons for the existence of zero-day vulnerabilities in software solutions is the implementation of improper or too co-dependent software/ architectural design patterns which often hinder troubleshooting or debugging processes. A suitable architectural design pattern can help mitigate security vulnerabilities. The basic procedure to choose a secure architectural design pattern for your software solution (Aryal et al., 2023).