A Large-Scale Empirical Study of Security Patches

In this work we conduct a large-scale empirical study of security patches.
Among our findings we identify that: security patches have less impact on code bases and result in more localized changes than non-security bug patches.

security issues reside in code bases for years, with a third introduced more than 3 years prior to remediation; security fixes are poorly timed with public disclosures, allowing attackers who monitor open-source repositories to get a jump of weeks to months on targeting not-yet-patched systems prior to any public disclosure and patch distribution.

We relied on the NVD to find publicly disclosed vulnerabilities.

The NVD contains entries for each publicly released vulnerability assigned a CVE identifier.

As Git is arguably the most popular version control system for open-source software , we focused on references to Git web interfaces.

To find Git repositories and their security patches, we first reverse-engineered the URL paths and parameters used by popular Git web interfaces.

For each commit we collected (both security and non-security patches), we extracted the historical versions of affected files both before and after the commit.

The NVD quantifies the severity of vulnerabilities using a standardized method called CVSS.

Our collected dataset consists of a diverse set of security vulnerabilities across numerous software projects, for which we have downloaded the source code repositories and amassed a set of both security and non-security bug fixes.

Upon a vulnerability's first discovery, we might naturally ask how long it plagued a code base before a developer rectified the issue. We call this duration the vulnerability's code base life span.

As the development and distribution of a patch often occur at different times, the code base life span reflects the window of opportunity for attackers who silently discover a vulnerability to leverage it offensively, before any defensive measures are taken.

One might hypothesize that more severe vulnerabilities reside in code bases for shorter periods, as their more visible impact may correlate with more likely discovery and quicker remediation. To explore this aspect, we correlate CVSS severity scores with life spans, computing a Spearman's correlation coefficient of $\rho=-0.062$. This indicates that there is no substantial (monotonic) correlation between a vulnerability’s severity and its life span.

Given the public nature of open-source projects and their development, an attacker targeting a specific software project can feasibly track security patches and the vulnerabilities they address.

Our findings have revealed shortcomings in our ability to quickly identify vulnerabilities and reliably address them.