MSc. in Computing
Practicum Approval Form

About your Practicum

What is the topic of your proposed practicum?

The topic of our proposed practicum is: Measuring the Risk of Open-Source Project: Investigating the Time Lag Between A Published Vulnerability and a Fix Release.

In order to examine the level of risk associated with using specific open source projects, we want to employ various machine learning analyses. The longer a project waits to issue a patch once a vulnerability is discovered, the longer it is vulnerable to attacks using the published vulnerability. The duration of the project's insecurity will be predicted by the machine learning analysis.

Please provide details of the papers you have read on this topic.

1. Open or Sneaky? Fast or Slow? Light or Heavy?: Investigating Security Releases of Open Source Packages
   https://ieeexplore.ieee.org/document/9792380
2. Software fault prediction using data mining, machine learning and deep learning techniques: A systematic literature review
   https://www.sciencedirect.com/science/article/pii/S0045790622001744
3. Time Series Forecasting of Software Vulnerabilities Using Statistical and Deep Learning Models
   https://www.mdpi.com/2079-9292/11/18/2820
4. Software Vulnerability Prediction Using Feature Subset Selection And Support Vector Machine
   https://www.irjet.net/archives/V2/i4/Irjet-v2i471.pdf
5. Time Series Analysis of Open-Source Software Projects
   https://dl.acm.org/doi/pdf/10.1145/1566445.1566531
6. Securing the Software Supply Chain: Recommended Practices for SBOM Consumption
   https://media.defense.gov/2023/Nov/09/2003338086/-1/-1/0/SECURING%20THE%20SOFTWARE%20SUPPLY%20CHAIN%20RECOMMENDED%20PRACTICES%20FOR%20SOFTWARE%20BILL%20OF%20MATERIALS%20CONSUMPTION.PDF

How does your proposal relate to existing work on this topic described in these papers?

In many of the papers we investigated they emphasised the importance of fixing security vulnerabilities as quickly as possible in order to prevent any exploitation by attackers. 10% of organisations said that their organisation had experienced a breach in the past 12 months related to supply chain attacks. Malicious actors can take advantage of the published vulnerability details being published to enact an attack before a fix is published.

Two of the papers above employ the use of time series analysis, one to predict the number of vulnerabilities a project will have and another to predict the project activity. Our plan is to use similar methods to investigate the time lag of published vulnerabilities and fixes.

The impact of a major security event as discussed in several of the papers usually has the effect of an increase in activity on open-source projects which will be a factor in our machine learning analyses - it being the reason we cannot use an average time lag to predict how long the code will be vulnerable to attack.

None of the papers we investigated so far in our research have employed the use of machine learning to predict the time lag between a vulnerability being discovered and a vulnerability being fixed. Although there is a box plot analysis of the time lag between security fix being released and a security advisory being published in the investigation into security releases, this does not claim to be a predictor of the time lag.

What are the research questions that you will attempt to answer?

Q. Can machine learning analysis predict the length of time for a vulnerability to be fixed?

We aim to find out how well various machine learning analyses can forecast the interval between the publication of a security vulnerability patch and the publication of the vulnerability issue using CVE data. Machine learning will show how long a project will be susceptible when a security issue is released if it can be utilized to accurately estimate this time lag. An attacker can potentially take advantage of a vulnerability within the critical window of time between the publication of a security vulnerability and the delivery of a security remedy, since during this time the attacker will be aware of the specifics of the vulnerability. With this method we can assign a risk based on the length of time these patches take to be released. This will depend on the vulnerability level as of course, more high risk vulnerabilities will need a patch to be released faster to avoid severe attacks.

We plan to use different predictors in the analyses such as repository size, source language, project, severity of the vulnerability and the project commit history. We plan to investigate algorithms such as regression, Time Series Analyses and support vector machines.

How will you explore these questions?

- ➔ What software and programming environment will you use?
  - ◆ Python and visual studio code.
- ➔ What coding/development will you do?
  - ◆ Using a number of machine learning analyses we will analyse the data for a set of CVEs and find the most reliable predictor for how long it will take for a vulnerability to be fixed based on the severity level of the vulnerability.
- ➔ What data will be used for your investigations?
  - ◆ Data from vulnerability databases such as https://nvd.nist.gov/ and https://www.cve.org
  - ◆ Date of vulnerability published - severity level
  - ◆ Affected version - package name and details such as source language, project size
  - ◆ Fixed version

- Is this data currently available, if not, where will it come from?
    - Data from the NVD
        - https://nvd.nist.gov/
        - https://services.nvd.nist.gov/rest/json/cves/2.0
        - https://services.nvd.nist.gov/rest/json/cvehistory/2.0
    - https://www.cve.org/
    - GitHub’s API to gather predictor data
        - https://docs.github.com/en/rest?apiVersion=2022-11-28
- What experiments do you expect to run?
    - A number of machine learning analyses on various packages and CVEs to see what the best predictor is based on severity, package details and source language.
    - A box plot analysis to get the average number of days for a fix to be released based on severity, package details and source language.
- What output do you expect to gather?
    - Graphs and results for how long it will take for vulnerabilities to be fixed.
- How will the results be evaluated?
    - By which ones make the most accurate prediction - using and comparing against past data.