Predikcija i klasifikacija softverskih ranjivosti

1. Definicija problema

Softverske ranjivosti predstavljaju značajan sigurnosni rizik jer omogućavaju napadačima da izvrše maliciozne akcije, uključujući krađu podataka i preuzimanje kontrole nad sistemom. Cilj ovog projekta je razvoj AI sistema koji kombinuje klasifikaciju za predviđanje da li je određeni softver ranjiv (ranjiv/neranjiv) i regresiju za predviđanje verovatnoće da ranjivost bude iskorišćena u budućnosti.

2. Motivacija

Ranjivosti u softveru predstavljaju ozbiljnu pretnju zbog krađe osetljivih podataka, finansijskih gubitaka, širenja malvera i problema u nacionalnoj bezbednosti. Razvijeni AI sistem ima za cilj da podrži bezbednosne timove omogućavajući im da prioritetizuju zakrpe na osnovu rizika, smanje vreme potrebno za analizu ranjivosti i efikasnije spreče sigurnosne incidente.

3. Skupovi podataka

Koriste se dva glavna izvora podataka. National Vulnerability Database (NVD) sadrži informacije o CVE ID-ovima, opisima ranjivosti, CVSS skorovima, tipovima ranjivosti i verzijama softvera sa preko 150.000 ranjivosti. Exploit Database (EDB) povezuje ranjivosti sa realnim exploit-ima i omogućava određivanje verovatnoće exploit-a. Ciljno obeležje za klasifikaciju je vulnerable (1/0), dok je za regresiju exploit_probability.

4. Pretprocesiranje podataka

Proces uključuje parsiranje NVD JSON fajlova radi izdvajanja atributa, ekstrakciju CVE ID-a iz EDB CSV fajlova, spajanje podataka preko CVE ID-a i kreiranje numeričke kolone num_exploits. Numeričke promenljive se normalizuju, kategorizovane se kodiraju pomoću one-hot encoding-a, a kompletni dataset se čuva u CSV/Parquet/Pickle format.

Napredni pristup uvodi pretrained transformer modele (BERT, RoBERTa, SecBERT) umesto TF-IDF za generisanje embedding vektora koji bolje hvataju semantičko značenje opisa ranjivosti. Arhitektura modela kombinuje tekstualni tok (opis ranjivosti → BERT embedding → Dense layers) sa tabularnim tokom (numerički i kategorijski atributi → Dense layers), a zatim spaja oba toka kroz Dense layers ka dva izlaza za klasifikaciju i regresiju.

5. Metodologija

Proces rešenja problema obuhvata prikupljanje i čišćenje podataka, feature engineering sa tekstualnim i numeričkim atributima, modeliranje koje koristi Logistic Regression, Random Forest, XGBoost i Neural Network za klasifikaciju i odgovarajuće regresore za regresiju, evaluaciju modela i integraciju rezultata kroz dashboard. Ulaz modela čine atributi ranjivosti (CVSS skor, opis, tip softvera, starost, tip ranjivosti), dok izlaz predstavljaju klasifikacija (ranjiv/neranjiv) i verovatnoća exploita.

Multi-task learning omogućava modelu da istovremeno uči klasifikaciju i regresiju, što poboljšava učenje zajedničkih reprezentacija podataka. Trening koristi BCEWithLogitsLoss za klasifikaciju i MSELoss za regresiju, AdamW optimizer i Linear LR warmup scheduler. Model hvata semantiku opisa ranjivosti, omogućava vizuelno rangiranje ranjivosti po riziku i interpretaciju ključnih feature-a kroz SHAP/LIME analizu.

6. Evaluacija modela

Podaci se dele na trening set (70%), validacioni set (15%) i test set (15%). Performanse klasifikacionog modela se mere pomoću accuracy, precision, recall i F1-score metrika, dok regresioni model koristi RMSE, MAE i R^2 score. Planirana je analiza važnosti feature-a i interpretacija rezultata kako bi se bolje razumele ključne karakteristike koje utiču na ranjivosti i verovatnoću exploit-a, kao i vizualizacija kroz TSNE/UMAP za embeddinge i prikaz sličnih ranjivosti.

7. Tehnologije

Implementacija se realizuje u Python-u sa bibliotekama pandas i numpy za obradu podataka, scikit-learn za mašinsko učenje, XGBoost za gradient boosting, i TensorFlow ili PyTorch za duboko učenje. Vizualizacija podataka i rezultata se omogućava kroz matplotlib, seaborn i plotly, dok se interaktivni dashboard realizuje koristeći Streamlit ili Dash.

8. Relevantna literatura

Projekt se zasniva na istraživanju "Predicting Exploitability of Software Vulnerabilities Using Machine Learning", NVD oficijalna dokumentacija (https://nvd.nist.gov), Exploit Database (https://www.exploit-db.com/) i Kaggle "Vulnerability Detection & Exploit Prediction" datasets (https://www.kaggle.com/).