User Interface for Interpretation of Code Vulnerabilities

Timothy Cain
Jordan Liebe
Luis de la Torre
John Miller

MITRE perspective: weaknesses make codes vulnerable to attacks

CVEs: Publicly reported vulnerabilities

CWEs: Expert assigned weaknesses

NVD: National Vulnerability Database (managed by NIST)

CVE - CVE (mitre.org)
Link to CVE download homepage

NOTICE: Changes are coming to CVE List Content Downloads in 2023.

The mission of the CVE® Program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.

Newest CVE Records

Follow @CVEnew >>

TOTAL CVE Records: 202,705

CVE-ID

CVE-2023-2726 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Inappropriate implementation in WebApp Installs in Google Chrome prior to 113.0.5672.126 allowed an attacker who convinced a user to install a malicious web app to bypass install dialog via a crafted HTML page. (Chromium security severity: Medium)

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:https://chromereleases.googleblog.com/2023/05/stable-channel-update-for-desktop_16.html
- URL:https://chromereleases.googleblog.com/2023/05/stable-channel-update-for-desktop_16.html
- MISC:https://crbug.com/1442018
- URL:https://crbug.com/1442018

NATIONAL VULNERABILITY DATABASE

NATIONAL VULNERABILITY DATABASE
NVD

VULNERABILITIES

CVE-2023-2726 Detail

AWAITING ANALYSIS

This vulnerability is currently awaiting analysis.

Automation of CVE to CWE mapping

V2W-BERT: A Framework for Effective Hierarchical Multiclass Classification of Software Vulnerabilities

Siddhartha Shankar Das
Purdue University
West Lafayette, IN, USA
das90@purdue.edu

Edoardo Serra
Boise State University
Boise, ID, USA
edoardoserra@boisestate.edu

Mahantesh Halappanavar
Pacific Northwest National Lab
Richland, WA, USA
hala@pnnl.gov

Alex Pothen
Purdue University
West Lafayette, IN, USA
apothen@purdue.edu

Ehab Al-Shaer
Carnegie Mellon University
Pittsburgh, PA, USA
ehab@cmu.edu

Natural language processing of CVE descriptions produced 768-component vectors, many of which were labeled by an expert-assigned CWE. Supervised machine learning generated a model to predict CWEs based on natural language processing of CVE descriptions. The model returns a probability to associate the input CVE with each of 124 CWEs in the NVD system.

NVD and V2W-BERT associate CVEs with a subset of MITRE's CWEs
Most often, a CVE is mapped to one of 27 CWEs

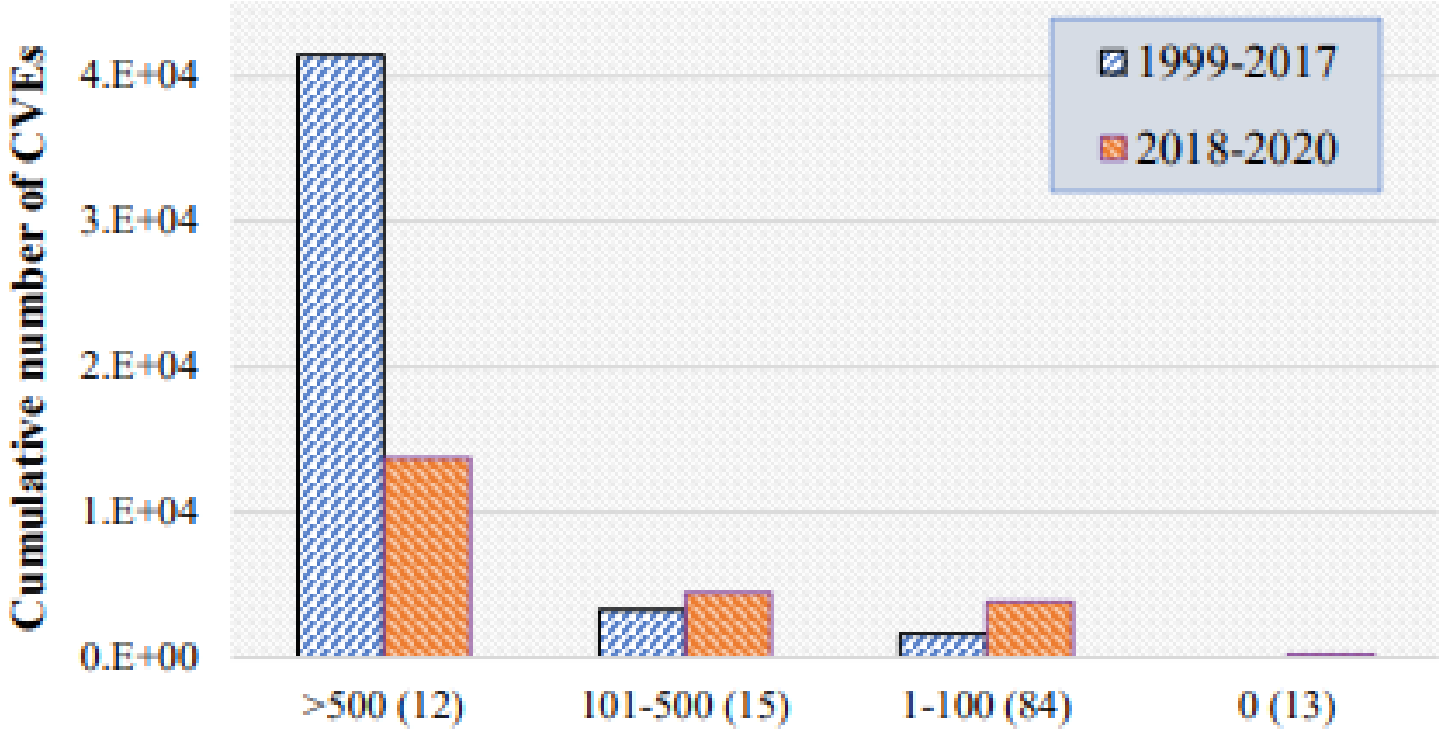


Figure 1: Distribution of the number of CVEs per CWE in the National Vulnerability Database, bucketed into four categories: 12 CWEs with 500 or more CVEs per CWE, 15 CWEs with 100 to 500 CVEs per CWE, 84 CWEs with 1 to 100 CVEs per CWE, and 13 CWEs with zero CVE.

From Das et al. V2W-BERT, 2021

Efficient Clustering of Software Vulnerabilities using Self Organizing Map (SOM)

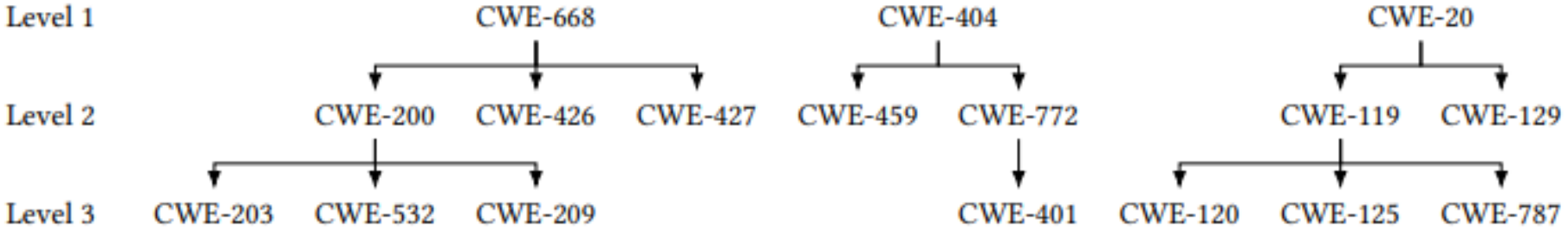
Khyati Panchal, Siddhartha Das, Luis De La Torre, John Miller, Mahantesh Halappanavar

Through our association with Mahantesh at PNNL, we obtained a database of approximately 170K 768-component vectors from natural language processing of CVEs, each labeled by the top 2 predictions of V2W-BERT model.

We clustered the vectors in this database using the SOM-Kmeans method.

Interpretation of these clusters provides a context for the CWE assigned to a given CVE, which will make the predictions of the V2W-BERT model more meaningful.

MITRE's CWEs have a hierarchical structure.
Most trees have 3 levels with CWEs becoming more specific root->leaf
Most of our CVE clusters are associated with CWEs in the tree rooted at CWE-707 or in the tree rooted at CWE-664.
Jordan Liebe will discuss these results in more detail.



From Das et al. V2W-BERT, 2021

Self-Organizing Maps (SOM)

The basic aim of SOM is data compression. Given a large dataset of input vectors, find a smaller set of prototypes, vectors with the same features at the input, that provide a good approximation to the whole input dataset.

Generally, we expect at least a factor of 10 compression.

The process of generating prototypes topologically orders them on a 2-dimensional array in clusters of similar prototypes.

We consider is array to be “map” of the underlying dataset.

Application of SOM for clustering the underlying dataset

Use Kmeans to quantify clusters of prototypes in the 2D array. Use a value of K that minimizes dispersion (i.e., produces tight clusters with large separation of their centroids).

For any vector in the dataset, find the most similar prototype, which we call the “best matching unit” (BMU). Clusters in the underlying dataset have BMUs in the same cluster of prototypes in the 2D array.

Any questions?