KVS: A Tool for Knowledge-Driven Vulnerability Searching

Xingqi Cheng Xiaobing Sun Lili Bo Ying Wei

Yangzhou University

November 16, 2022

Outline

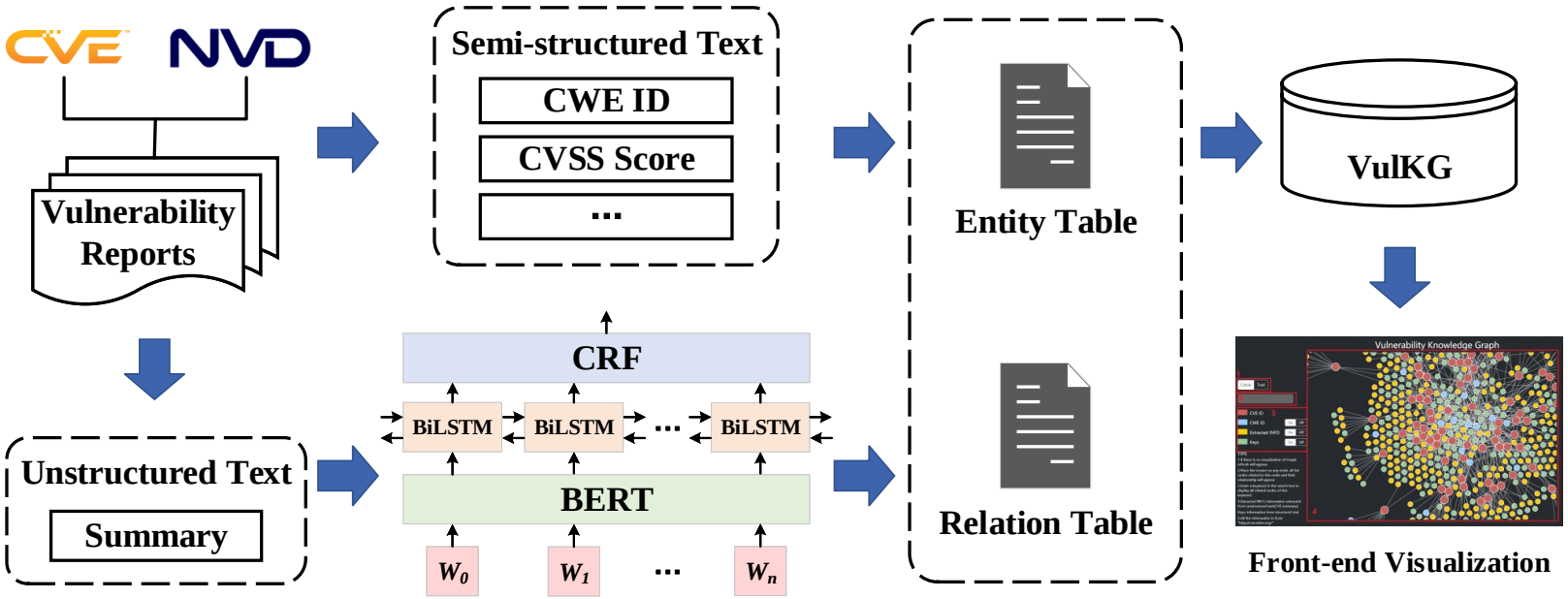
Outline

Objective

How do we connect those vulnerabilities?

Outline

Overview of construction process

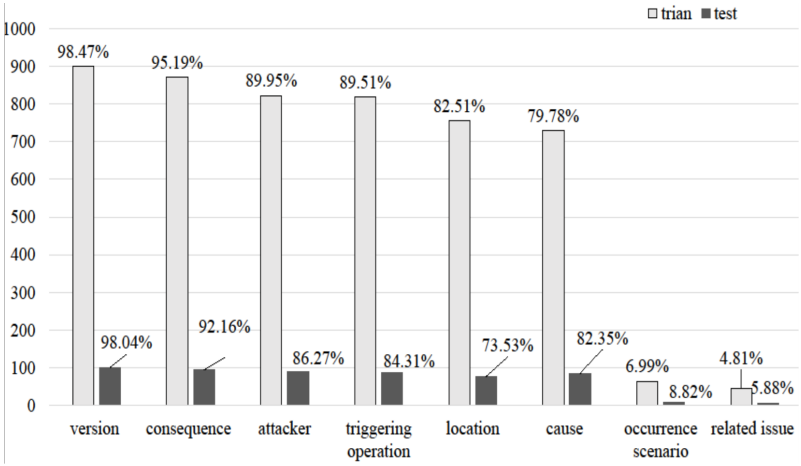


We divide the construction of KVS into three parts, namely the vulnerability named entity recognition **(VulNER)** part, the vulnerability knowledge graph **(VulKG) storage** part, and the **VulKG visualization** part.

VulNER

Data set preparation

Summary	Integer overflow in the pango_glyph_string_set_ size function in pango/glyphstring.c in Pango be- fore 1.24 allows context-dependent attackers to cause a denial of service via a long glyph string that triggers a heap-based buffer overflow.
Label	**cause**: Integer overflow; **location**: pango_glyph_ string_set_size function in pango/glyphstring.c; **version**: Pango before 1.24; **attacker**: context- dependent attac_kers; **consequence**: a denial of service **consequence**: a heap-based buffer over- flow; **triggering operation**: a long glyph string



We manually annotated 1017 CVE summaries, of which 915 were the training set and 102 were the test set.The prediction set is 4114.

VulNER

We perform the VulNER task based on two models, BERT-Softmax model and BERT-BiLSTM-CRF model.

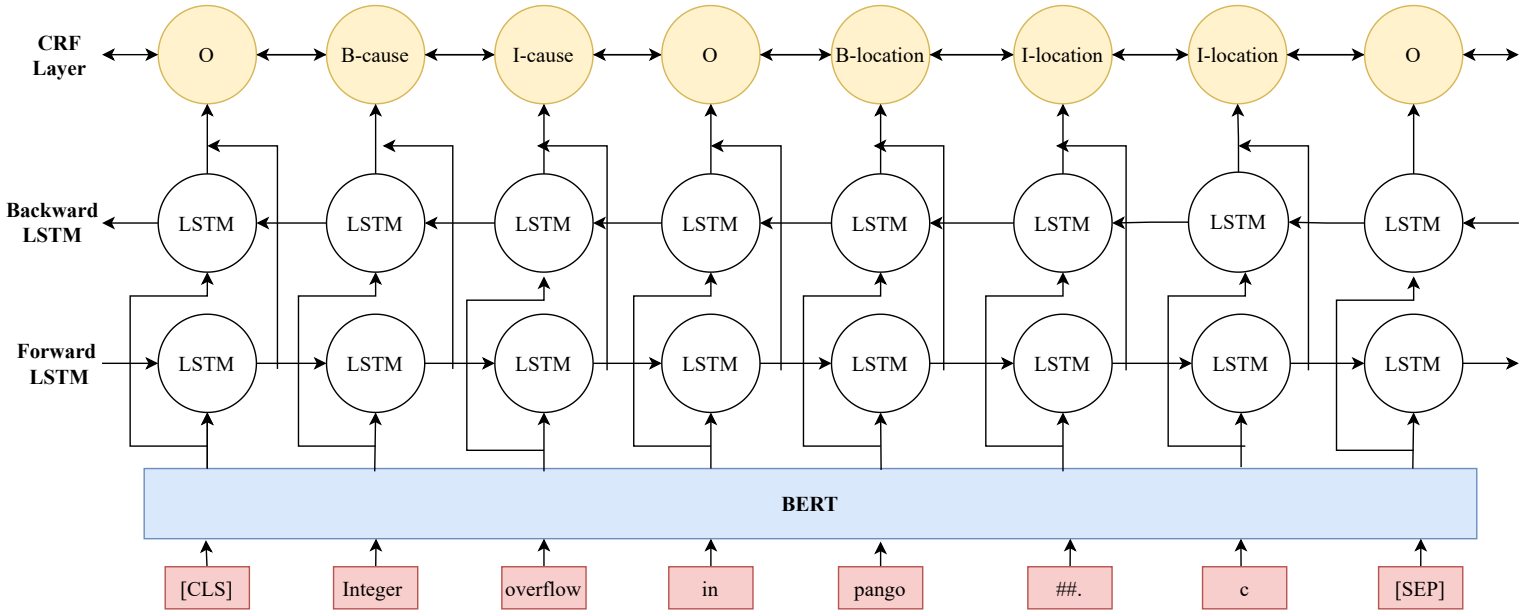


Figure: BERT-BiLSTM-CRF model

VulNER

Experiment result of two models.

Entities	BERT-Softmax			BERT-BiLSTM-CRF		
	P	R	F1	P	R	F1
version	0.90	0.91	0.90	0.93	0.93	0.93
consequence	0.81	0.79	0.80	0.85	0.85	0.85
attacker	0.93	0.92	0.93	0.94	0.93	0.93
triggering operation	0.82	0.86	0.84	0.88	0.89	0.89
location	0.90	0. 94	0.92	0.93	0.95	0.94
cause	0.80	0.79	0.79	0.83	0.83	0.83
occurrence scenario	0.86	0.67	0.75	0.75	0.67	0.71
related issue	0.33	0.11	0.17	0.45	0.56	0.50
Overall Result	**0.86**	**0.85**	**0.85**	**0.88**	**0.89**	**0.88**

Because the BERT-BiLSTM-CRF model has better performance, we use it to predict entities.

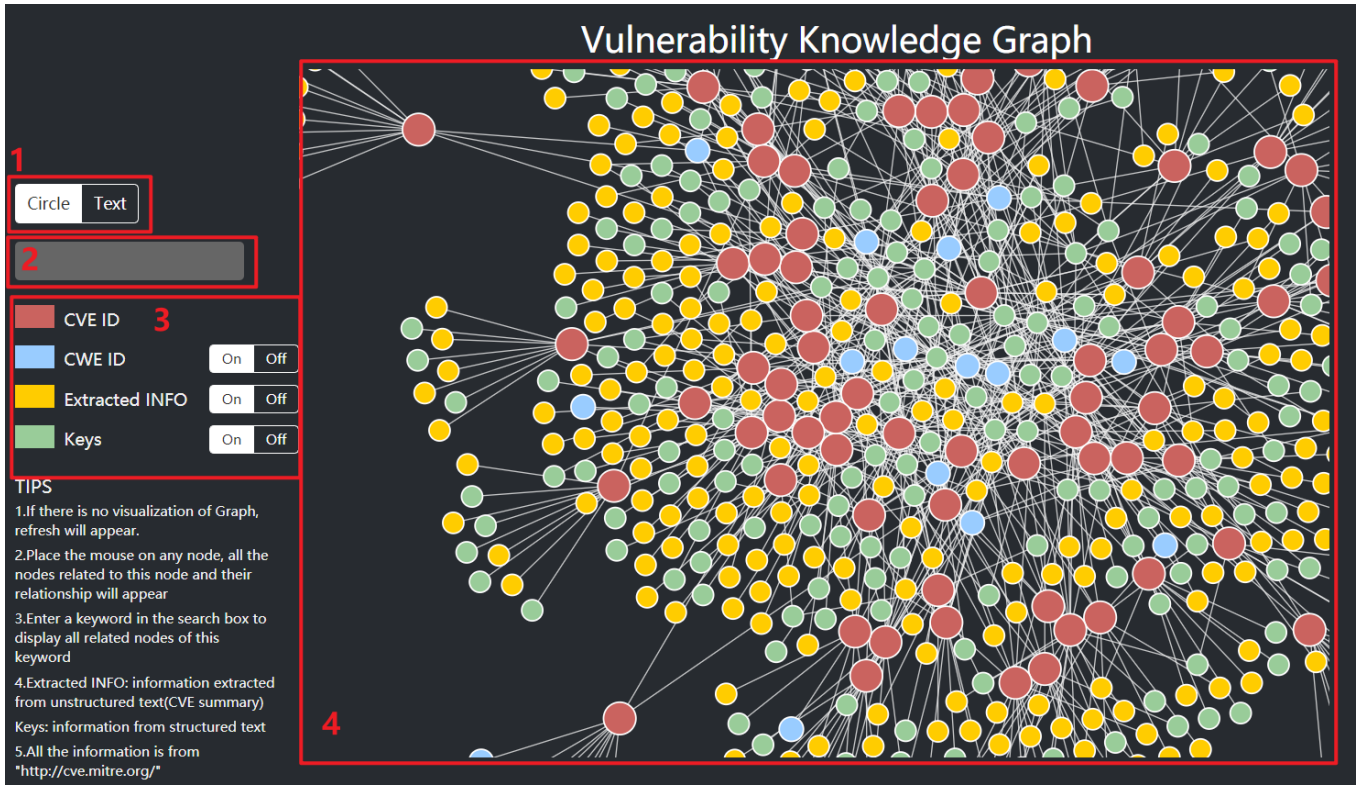
VulKG Storage and Visualization

After the entity prediction, we have the corresponding entity tables and relation tables, which we further store as nodes and relations in Neo4j (a total of **20,631 nodes** and **50,961 relations**). In this work, we define 15 relation types and 16 node types. For the visualization of VulKG, we combined D3.js to realize the front-end visualization.

nodes			relationships	
Extracted Info (yellow)	cause	1877	cve: cause	3398
	consequence	1567	cve: consequence	4264
	version	1955	cve: version	4052
	location	2530	cve: location	3160
	attacker	153	cve: attacker	3349
	triggering operation	1896	cve: triggering operation	3359
	occurrence scenario	224	cve: occurrence scenario	267
	related issue	107	cve: related issue	126
Keys (green)	publish date	838	cve: publish date	4114
	score	51	cve: score	4114
	project	348	cve: project	4114
	vulnerability classification	54	cve: vulnerability classification	4114
	commit id	4390	cve: commit id	4114
	update date	688	cve: update date	4114
IDs	CWE ID (blue)	92	cve: cwe id	4114
	CVE ID (red)	3755		
Total		20631		50961

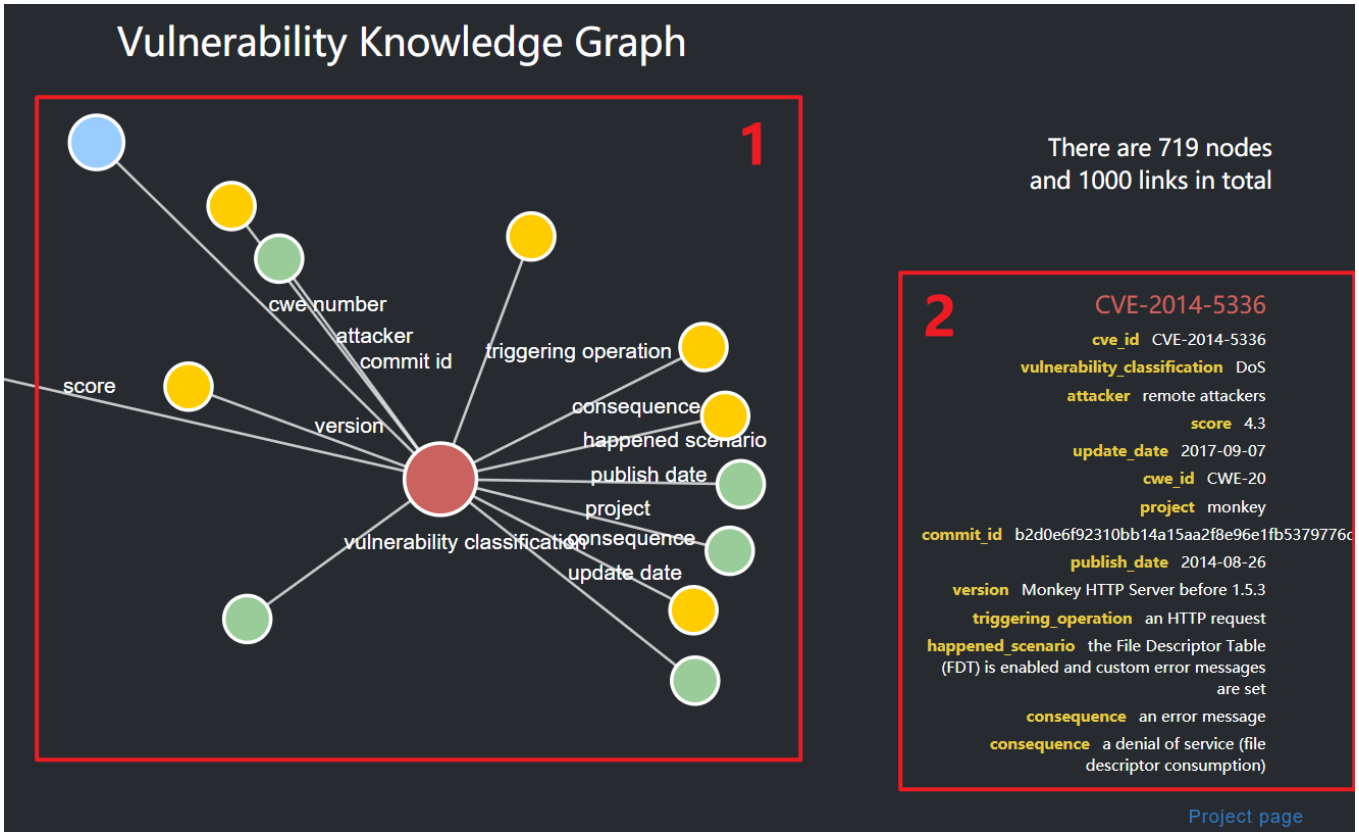
Outline

KVS Usage



KVS is available at
`https://cinnqi.github.io/Neo4j-D3-VKG/`.

KVS Usage


Vulnerability Knowledge Graph
1
cwe number
attacker
commit id
triggering operation
score
version
consequence
happened scenario
publish date
project
vulnerability classification
consequence
update date
There are 719 nodes
and 1000 links in total
2
CVE-2014-5336
cve_id CVE-2014-5336
vulnerability_classification DoS
attacker remote attackers
score 4.3
update_date 2017-09-07
cwe_id CWE-20
project monkey
commit_id b2d0e6f92310bb14a15aa2f8e96e1fb5379776c
publish_date 2014-08-26
version Monkey HTTP Server before 1.5.3
triggering_operation an HTTP request
happened_scenario the File Descriptor Table (FDT) is enabled and custom error messages are set
consequence an error message
consequence a denial of service (file descriptor consumption)
Project page

Outline

Conclusion

The main contributions of this work are:

- Constructing fine-grained NER datasets for vulnerability domain.
- The Bert-Softmax model and Bert-BiLSTM-CRF model are fine-tuned and applied to NER in the vulnerability domain to improve the accuracy of vulnerability entity recognition.
- Build a vulnerability knowledge graph based on dedicated defined entities and relations.

Future work includes:

- Pipeline the construction process, e.g. automatically build the knowledge graph.
- Further increase the data scale while ensuring the recognition accuracy.
- Explore more possibilities for practical usage.

Questions

Thank you for your attention!

Questions?