

# KVS: A Tool for Knowledge-Driven Vulnerability Searching

Xingqi Cheng   Xiaobing Sun   Lili Bo   Ying Wei

Yangzhou University

November 16, 2022



# Outline

- 1 Introduction
- 2 Approach
- 3 Usage Example
- 4 Conclusion

# Outline

- 1 Introduction
- 2 Approach
- 3 Usage Example
- 4 Conclusion

# Objective

**CVE** CVE List • CNAs • WGs • Board • NVD Data Feed • CVE Source • CVE ID

Search CVE List Downloads Data Feeds Update a CVE Record Request CVE

TOTAL CVE Records: 185458

NOTICE: Transition to the all-new CVE website at [WWW.CVE.ORG](http://WWW.CVE.ORG) is underway and will last up to one year. (details)

NOTICE: Changes coming to CVE Record Format JSON and CVE List Content Downloads in 2022.

HOME > CVE > SEARCH RESULTS

### Search Results

There are **1247** CVE Records that match your search.

Name	Description
<a href="#">CVE-2022-40310</a>	Authenticated (subscriber+) Race Condition vulnerability in Rate my Post &#9211; WP Rating System plugin <= 3.3.4 at WordPress allows attackers to increase/decrease votes.
<a href="#">CVE-2022-40307</a>	An issue was discovered in the Linux kernel through 5.19.8, drivers/firmware/efi/capsule-loader.c has a race condition with a resultant use-after-free.
<a href="#">CVE-2022-39188</a>	An issue was discovered in include/asm-generic/tlb.h in the Linux kernel before 5.19. Because of a race condition (unmap, mapping, range versus mremap), a device driver can free a page while it still has stale TLB entries. This only occurs in situations with VM_PFNMAP VMAs.
<a href="#">CVE-2022-39006</a>	The MPTCP module has the race condition vulnerability. Successful exploitation of this vulnerability may cause the device to restart.
<a href="#">CVE-2022-38120</a>	In Apache Airflow prior to 2.3.4, an insecure umask was configured for numerous Airflow components when running with the "--daemon" flag which could result in a race condition giving world-writable files in the Airflow home directory and allowing local users to expose arbitrary file contents via the webserver.
<a href="#">CVE-2022-37035</a>	An issue was discovered in bgpd in FRRouting (FRR) 8.3. In bgp_notify_send_with_data() and bgp_process_packet() in bgp_packet.c, there is a possible use-after-free due to a race condition. This could lead to Remote Code Execution or Information Disclosure by sending crafted BGP packets. User interaction is not needed for exploitation.
<a href="#">CVE-2022-36422</a>	Rating increase/decrease via race condition in Lester "Gahzer" Chan WP-PostRatings plugin <= 1.89 at WordPress.

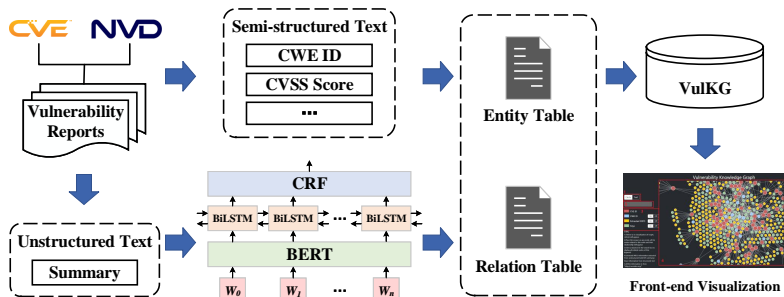


## How do we connect those vulnerabilities?

# Outline

- 1 Introduction
- 2 Approach
- 3 Usage Example
- 4 Conclusion

# Overview of construction process

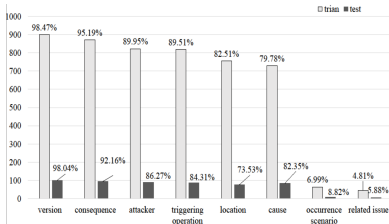


We divide the construction of KVS into three parts, namely the vulnerability named entity recognition (**VulNER**) part, the vulnerability knowledge graph (**VulKG**) storage part, and the **VulKG visualization** part.

# VuINER

## Data set preparation

Summary	Integer overflow in the pango_glyph_string_set_size function in pango/glyphstring.c in Pango before 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	<b>cause:</b> Integer overflow; <b>location:</b> pango_glyph_string_set_size function in pango/glyphstring.c; <b>version:</b> Pango before 1.24; <b>attacker:</b> context-dependent attackers; <b>consequence:</b> a denial of service; <b>consequence:</b> a heap-based buffer overflow; <b>triggering operation:</b> 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.

# VuINER

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

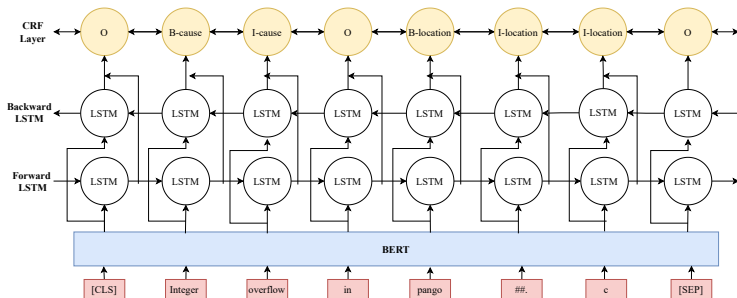


Figure: BERT-BiLSTM-CRF model



# VuINER

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
<b>Overall Result</b>	<b>0.86</b>	<b>0.85</b>	<b>0.85</b>	<b>0.88</b>	<b>0.89</b>	<b>0.88</b>

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

# VulKG Storage and Visualization

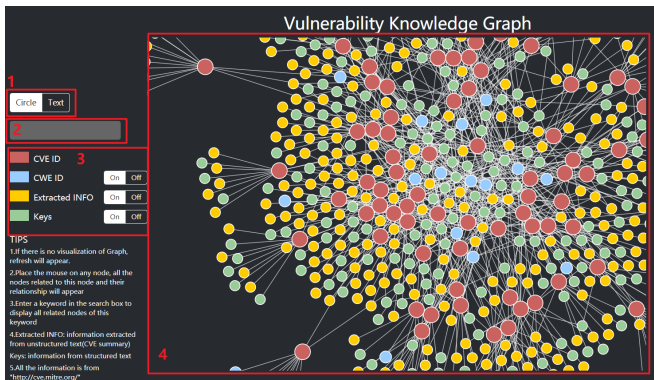
After the entity prediction, we have the corresponding entity tables and relation tables, which we further store as nodes and relations in Neo4j (a total of **20,631 nodes** and **50,961 relations**). In this work, we define 15 relation types and 16 node types. For the visualization of VulKG, we combined 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
Keys (green)	related issue	107	cve: related issue	126
	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
IDs	update date	688	cve: update date	4114
	CWE ID (blue)	92	cve: cwe id	4114
Total	CVE ID (red)	3755		
		20631		50961

# Outline

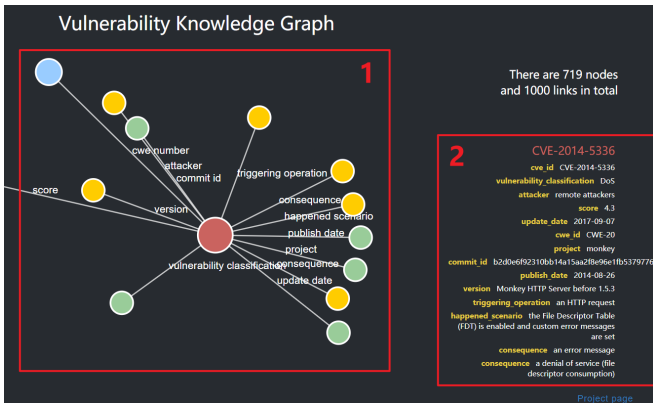
- 1 Introduction
- 2 Approach
- 3 Usage Example**
- 4 Conclusion

# KVS Usage



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

# KVS Usage



# Outline

- 1 Introduction
- 2 Approach
- 3 Usage Example
- 4 Conclusion**

# 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?**