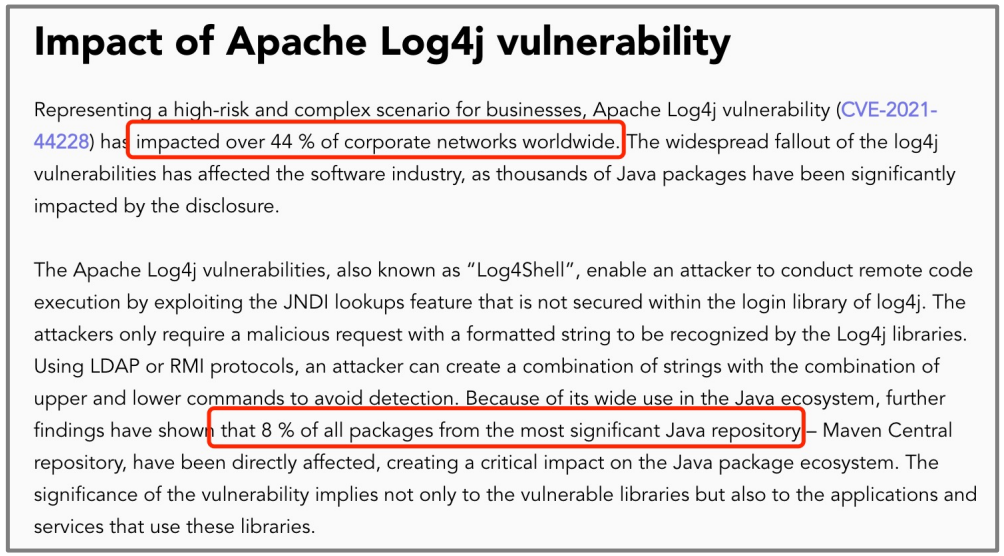
Achieving Interpretable DL-based Web Attack Detection through Malicious Payload Localization

Peiyang Li, Fukun Mei, Ye Wang, Zhuotao Liu, Ke Xu,
Chao Shen, Qian Wang, Qi Li

Web Attacks

- Web attacks are serious threats to Web security.
- Attackers may exploit vulnerabilities in Web applications by crafting malicious payloads in specific fields of HTTP requests.

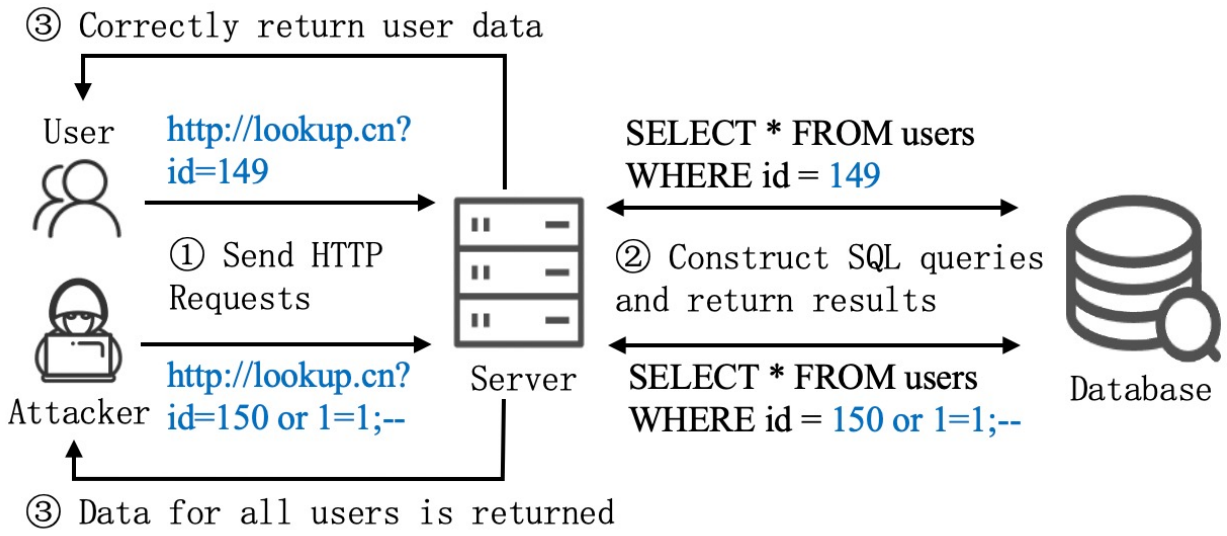


Impact of Apache Log4j vulnerability

Representing a high-risk and complex scenario for businesses, Apache Log4j vulnerability (CVE-2021-44228) has impacted over 44 % of corporate networks worldwide. The widespread fallout of the log4j vulnerabilities has affected the software industry, as thousands of Java packages have been significantly impacted by the disclosure.

The Apache Log4j vulnerabilities, also known as "Log4Shell", enable an attacker to conduct remote code execution by exploiting the JNDI lookups feature that is not secured within the login library of log4j. The attackers only require a malicious request with a formatted string to be recognized by the Log4j libraries. Using LDAP or RMI protocols, an attacker can create a combination of strings with the combination of upper and lower commands to avoid detection. Because of its wide use in the Java ecosystem, further findings have shown that 8 % of all packages from the most significant Java repository – Maven Central repository, have been directly affected, creating a critical impact on the Java package ecosystem. The significance of the vulnerability implies not only to the vulnerable libraries but also to the applications and services that use these libraries.

Log4Shell poses threats to over 44% of corporate networks worldwide

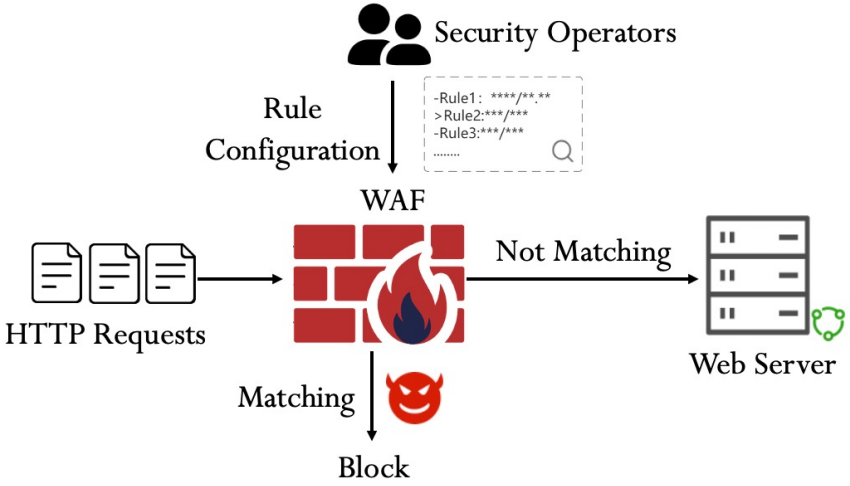


Example of Web attacks

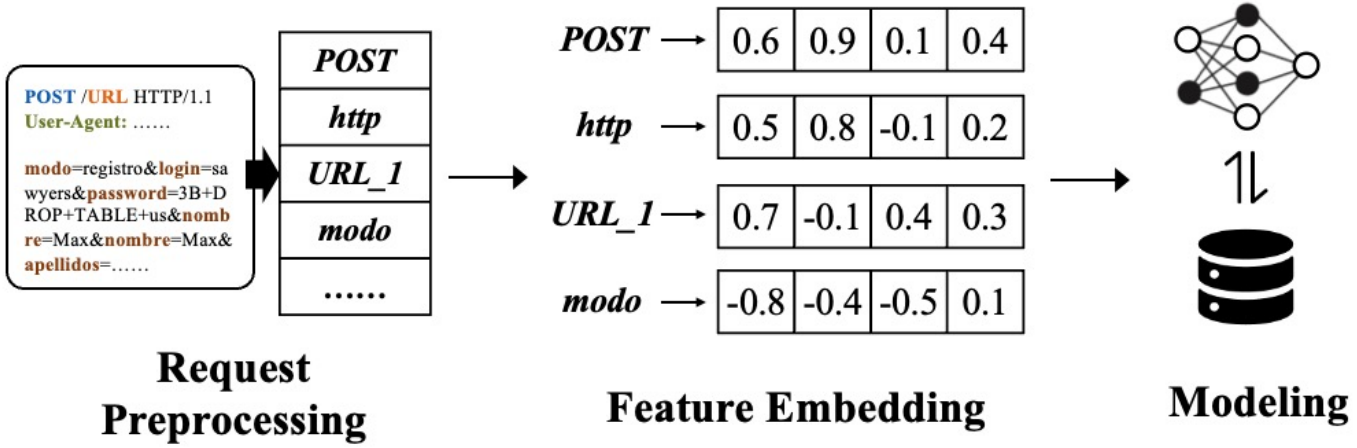
DL-based Web Attack Detection

Deep learning (DL) based methods are developed to detect Web attacks.

- Mapping the input HTTP requests to discrete labels.
- Learning normal/malicious patterns from the training data.
- Avoiding rule configurations and being able to detect unseen threats.



Traditional rule-based WAFs



DL-based Web attack detectors

Motivation

Existing DL-based detectors cannot produce interpretable results

- Only output a label (e.g., normal or malicious)
- Uninterpretable detection results require security operators to spend a significant amount of time and manual effort in analyzing the attacks
- The lack of interpretability could also cause attack desensitization

Existing interpretability frameworks are not applicable to Web attack detection

- HTTP requests have well-defined structures and contain various fields.
- Overlook the structure information of HTTP requests and only analyze the numerical features of DL models

We need to develop new interpretability framework for existing DL-based Web attack detection methods

Core idea

Achieving interpretable Web attack detection by identifying the locations of malicious payloads within attack requests.

```
POST /tienda1/miembros/editar.jsp HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.
8 (like Gecko)
Pragma: no-cache
Cache-control: no-cache
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,te
xt/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Host: localhost:8080
Cookie: JSESSIONID=D8B2A8778565BCF42799D84360295207
Content-Type: application/x-www-form-urlencoded
Content-Length: 335
…

modo=registro&login=sawyers&password=encUmb2ad3%27%3B+DRO
P+TABLE+usuarios&nombre=Max&apellidos=Sj%F6strand+Ampurias&
email=tamburi%40hispadis.ar&dni=30293636Z&direccion=C%2F+Virge
n+De+Las+Angustias+122+4%3FA&ciudad=Fuentespina&cp=48220&pr
ovincia=Asturias&ntc=8898020239162472&B1=Registrar
```

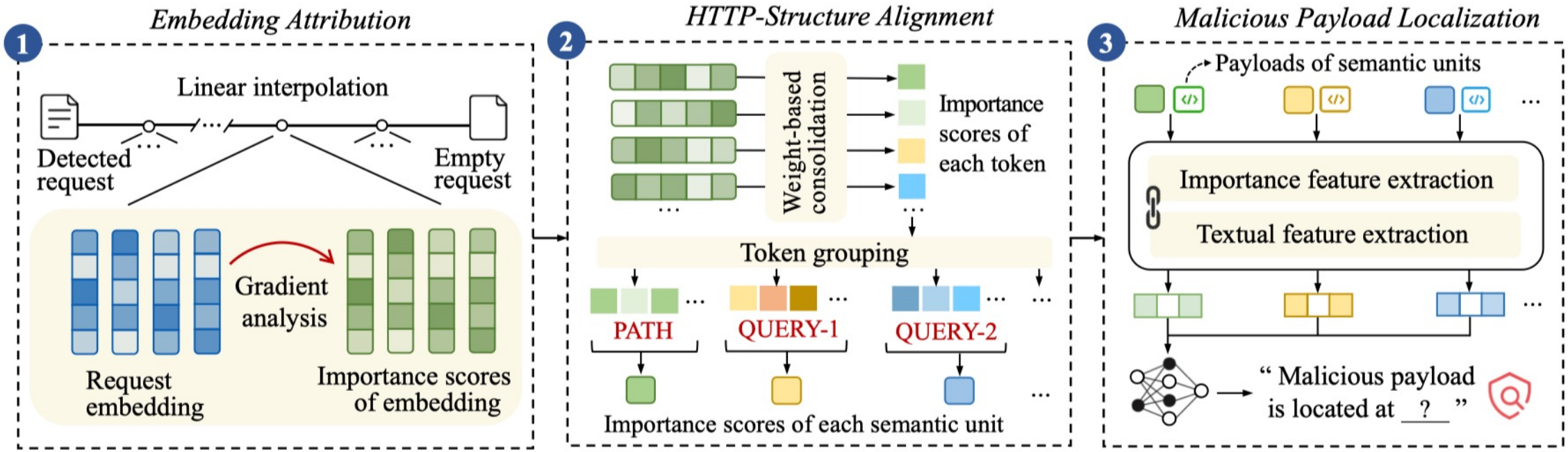
Attack request with malicious payload locations indicated in red

How to locate malicious payloads?

Observation: Malicious payloads often exert a considerable influence on the predictions of detection models.

We can quantify the importance of individual fields within HTTP requests to locate malicious payloads.

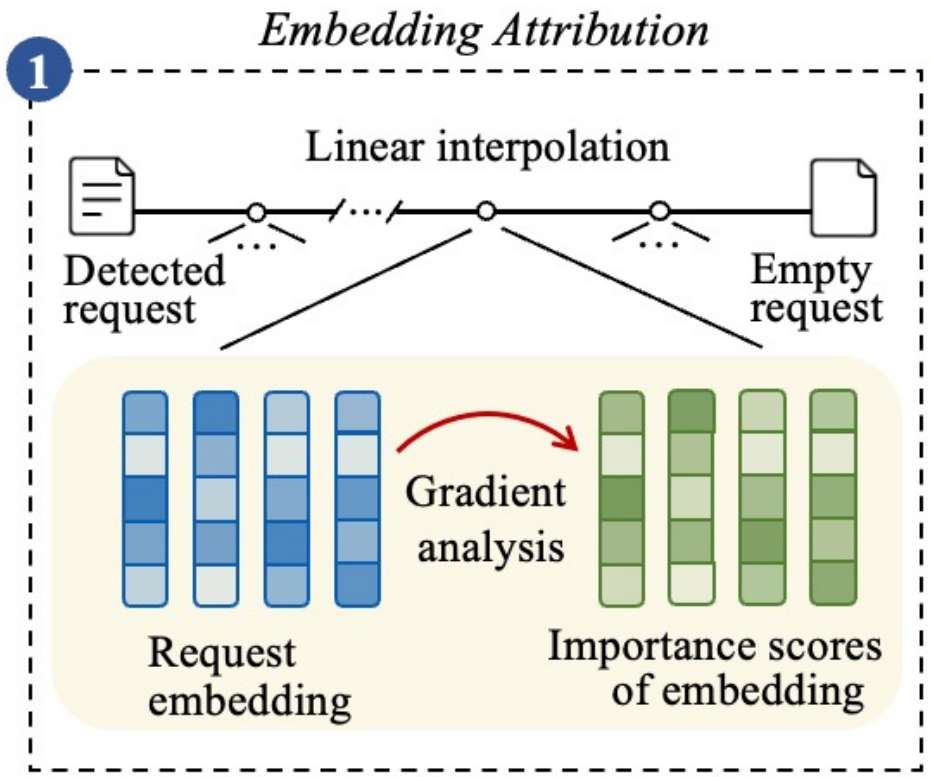
Design



Overall Workflow

- Assess the importance of embedding vectors in existing detection models.
- Assess the importance of each field in HTTP requests.
- Establish a robust connection between the location of the malicious payload and the importance of each HTTP field.

Design

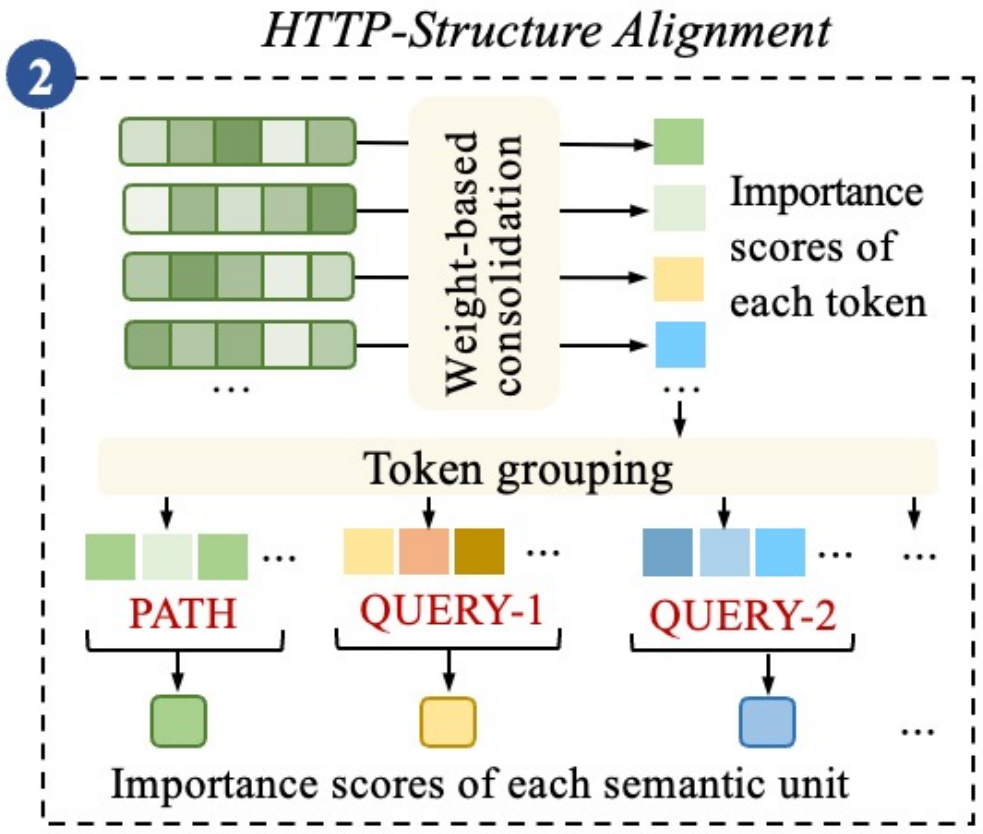


① Embedding Attribution

Assess the influence of each embedding value of the detection model on its predictions

- **Gradient Computing:** reflect how much each feature contributes to the output faithfully
- **Linear interpolation :** provide a global view of embedding importance

Design

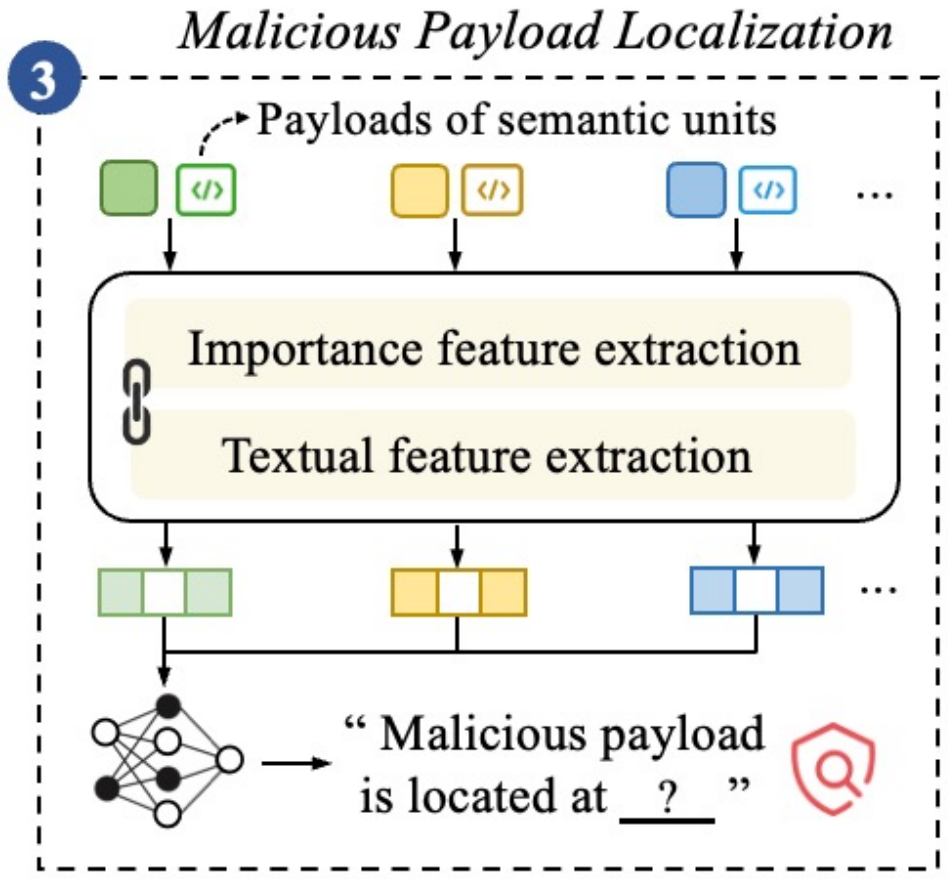


② HTTP-structure alignment

Align the embedding importance with each field of HTTP requests, and obtain the importance of these HTTP fields

- **Token-level aggregation:** consolidate embedding values of each token into the token importance score.
- **Protocol-level aggregation:** identify the tokens corresponding to each HTTP field and aggregate the importance of these tokens.

Design

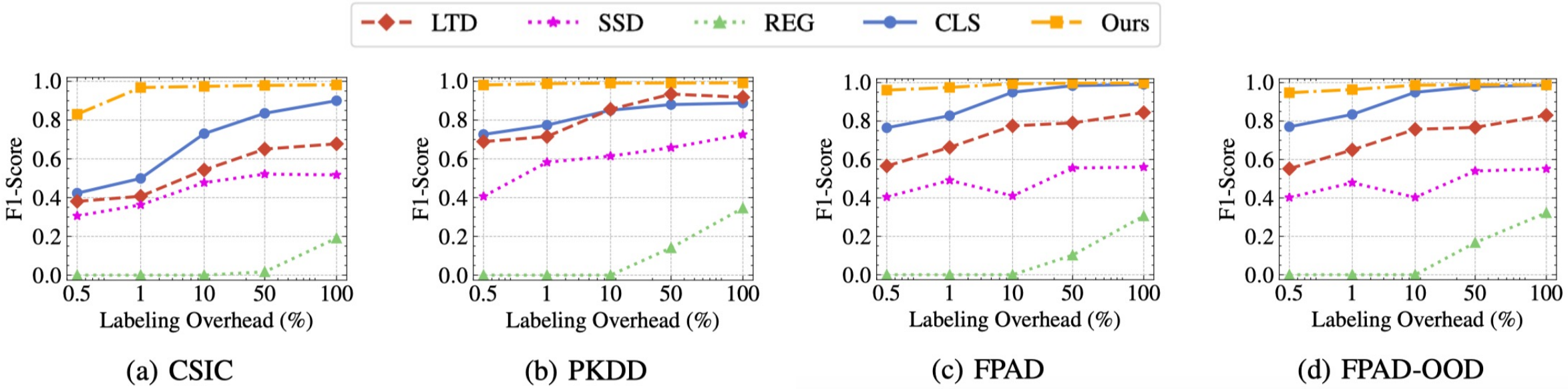


③ Malicious Payload Localization

Utilize the importance scores of each HTTP field derived from previous steps to identify which fields contain malicious payloads

- **Location features**: integrate the textual semantics and importance scores to enhance localization robustness.
- **Localization model**: train a lightweight binary classifier to predict the presence of malicious payloads in each HTTP field.

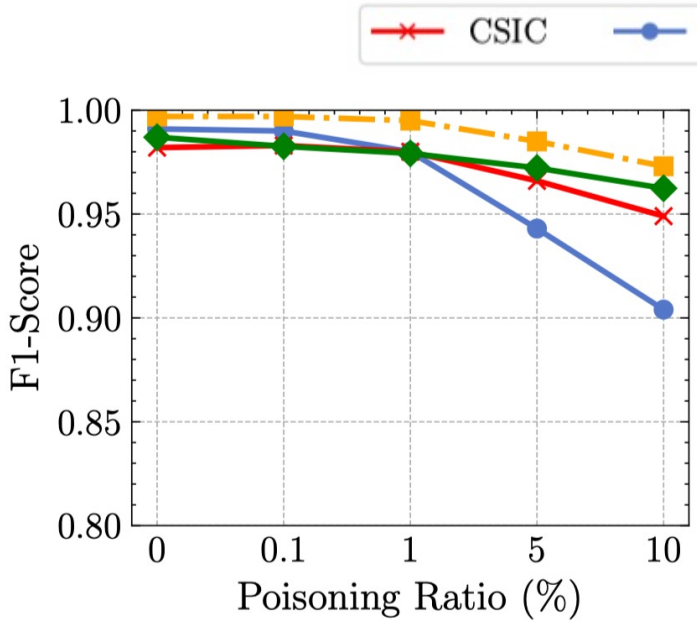
Performance Comparison



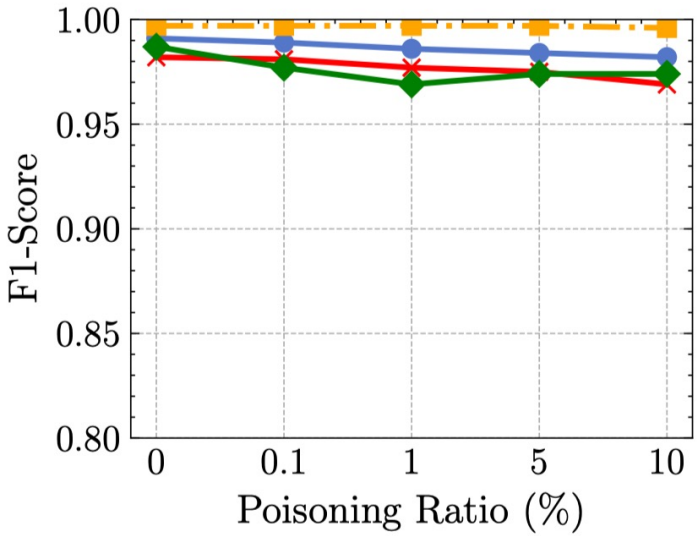
Remarks

- In most cases, the F1-score of our method approaches 1.0, demonstrating the effectiveness of malicious payload localization
- Our method is stable and consistently maintains a significant performance advantage across all datasets

Performance under Data Poisoning



Detection model poisoning

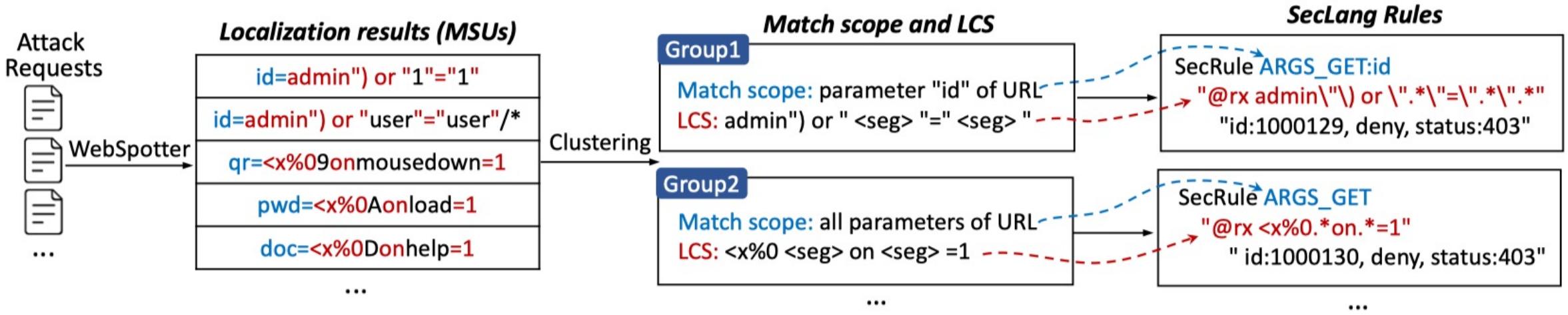


Localization model poisoning

Remark

- The localization performance only degrades slightly when the poisoning ratios of detection model and localization model increase.

Application: Generating WAF rules



Overall Workflow

- Cluster malicious payloads that exhibit similar semantics based on the edit distance
- Generate detection rules by determining the match scope and extracting the regular expressions that reflect their common pattern

Application: Generating WAF rules

Dataset	Pre		Rec		F1	
	Before	After	Before	After	Before	After
FPAD	0.999	0.996	0.742	0.934	0.852	0.964
CVE	0.999	0.992	0.840	0.947	0.913	0.969
G1	0.998	0.999	0.358	0.983	0.527	0.991
G2	0.999	0.999	0.744	0.996	0.853	0.998

Detection performance of WAFs before and after generating new WAF rules

Remarks

- The generated rules can effectively improve the attack detection rate of WAFs, while introducing a small number of false positives.

Conclusion

- Present an interpretability framework for existing DL-based Web attack detection methods
  - Embedding Attribution: assess the importance of embedding vectors in existing detection models.
  - HTTP-Structure Alignment: assess the importance of each field in HTTP requests.
  - Malicious Payload Localization: establish a connection between the location of the malicious payload and the importance of each HTTP field.
- Extensive evaluations with real-world data demonstrate the effectiveness of our method

NDSS SYMPOSIUM

Thank You!

Presenter: Jian Cui

Slide contributor: Peiyang Li

 peiyangli.20@gmail.com