

УДК 004.89

Оптимизация средства поиска XSS-уязвимостей с помощью LSTM модели

Мицих Владислав Павлович

mitsikhvp@student.bmstu.ru

Вершинин Евгений Владимирович

vershinin@bmstu.ru

КФ МГТУ им. Н.Э. Баумана, Калуга, Россия

Проанализирована проблема повышения безопасности веб-систем от межсайтового скриптинга (XSS) посредством внедрения интеллектуальной системы на основе нейронных сетей в программное средство. Исследованы три стратегических направления: оптимизация обхода веб-сайтов, автоматическое классификация XSS-уязвимостей, а также фильтрация ложных срабатываний. Описываются сложности, связанные с разработкой таких систем. Приведены результаты обучения нейронной сети LSTM.

Ключевые слова: XSS, уязвимости, LSTM, искусственный интеллект, нейронные сети

Современные веб-системы становятся все более сложными и уязвимыми для атак, одной из которых является межсайтовый скриптинг (XSS). Одним из решений обеспечения безопасности интернет-ресурсов является разработка интеллектуальной системы, которая способна эффективно выявлять подобные уязвимости. Однако традиционные методы выявления уязвимостей подвержены ложным срабатываниям, требуют значительных ресурсов на сканирование сайтов, а также не всегда способны точно классифицировать эти угрозы.

В исследовании рассмотрен вопрос выбора направления для обучения нейронной сети для ее дальнейшей интеграции в систему. Были выбраны три направления: оптимизация процесса обхода веб-сайтов; автоматическая классификация найденных уязвимостей; фильтрация ложных срабатываний.

Эти направления рассматриваются с точки зрения актуальности применения, а также сложности выполнения и возможности интеграции в систему автоматизированной программы, по поиску XSS-уязвимостей в веб-приложениях.

Оптимизация обхода веб-сайтов направлена на то, чтобы уменьшить объем данных, которые необходимо сканировать, и сосредоточить внимание на наиболее вероятных местах возникновения уязвимостей. Это означает более быстрый и эффективный процесс сканирования веб-сайтов, с акцентом на анализ тех частей сайта, где вероятность XSS-уязвимостей наиболее высока [1]. Если система будет обходить веб-сайты более избирательно и сфокусируется на критически важных местах таких, как формы ввода данных, страницы с динамическим контентом, то сканирование станет более быстрым и точным.

Сложность заключается в создании алгоритмов, которые смогут эффективно предсказывать, какие части сайта содержат потенциальные уязвимости.

Для этого требуется анализ большого количества данных и выявление закономерностей.

Классификация основных XSS-уязвимостей на такие типы, как отраженные (Reflected), хранимые (Stored) и DOM-модели (DOM-based), может помочь более эффективно устранять проблемы безопасности [2]. Автоматическая классификация позволит быстро определить, какой тип атаки был выявлен, что поможет разработчикам быстрее реагировать на угрозы и устранять уязвимости.

Для успешного обучения модели, способной классифицировать вышеперечисленные уязвимости, необходим тщательно структурированный датасет, включающий разнообразные данные о веб-запросах, ответах, элементах HTML/JavaScript, а также информацию о типе уязвимости.

В датасете должна содержаться полезная нагрузка, которая отправляется через форму, URL или другой ввод данных для атаки. Эти данные могут включать скрипты на JavaScript, HTML-теги или комбинации из них [3].

Примеры:

`<script>alert('XSS')</script>` — классический пример Reflected;

`<img src=x onerror=alert(1)>` — может использоваться как Stored;

`document.write()` или `innerHTML` — примеры DOM-based, которые не зависят от серверного ответа.

Это основная часть, по которой модель должна учиться распознавать потенциальные атаки. Также она должна различать типы атак, основываясь на том, как вредоносный код внедряется в веб-страницу или отправляется серверу.

Основная трудность заключается в сложной природе XSS-атак. Некоторые уязвимости могут не попадать в одну четкую категорию и иметь признаки нескольких типов. Кроме того, для обучения модели потребуется значительный объем данных с точной разметкой.

В веб-приложениях ложные срабатывания затрудняют работу специалистов по безопасности, так как каждый ложный сигнал требует проверки. Это относится к ситуациям, когда система безопасности сообщает об угрозе, которой на самом деле не существует. В контексте XSS-уязвимостей, ложные срабатывания могут быть связаны с безопасным кодом, который система по ошибке определяет как вредоносный [4].

Основная сложность фильтрации ложных срабатываний заключается в нахождении баланса между выявлением реальных угроз и избеганием ошибок. Для этой задачи необходимо обучить модель на большом объеме данных с точно размеченными примерами безопасного и вредоносного кода.

Для выполнения этой задачи был найден размеченный датасет. Под него была создана и обучена нейронная сеть на основе двунаправленных LSTM слоев. Набор данных содержит полезную нагрузку, которая отмечена, как «1» в конце строки и безопасный код с индексом «0» в таком виде:

```
35,"<li><a href=""#CITEREFNilsson1998">Nilsson 1998 </a>, chpt. 8 </li> </ul>";0
36,<colgroup onblur=alert(1) tabindex=1 id=x></colgroup><input autofocus
```

```

37,"<style>@keyframes x{}</style><main style=""animation-name:x"" onanimation-
start=""alert(1)""></main>",1
38,"<article draggable=""true"" ondrag=""alert(1)"">test</article>",1
39,"<li><a href=""/wiki/Computational_physics"" title=""Computational phys-
ics"">Computational physics </a> </li>",0

```

Результаты обучения на 7 эпохе из 10 возможных получились следующими: accuracy: 1.0000 — точность на обучении достигла 100 %, что означает, что модель классифицировала все обучающие примеры правильно; loss: 1.9785e-04 — потери на обучении снизились до очень малого значения 1.9785×10^{-4} , что указывает на то, что модель почти идеально подстроилась под обучающие данные; val_accuracy: 0.9982 — точность на валидационных данных составила 99,82 %, что также очень высоко и свидетельствует о хорошей обобщающей способности модели; val_loss: 0.0089 — потери на валидации составляют 0.0089.

Модель остановила обучение на 7-й эпохе из-за использования метода ранней остановки, потому что валидационные потери перестали улучшаться на протяжении трех эпох подряд. Это помогло избежать переобучения и сохранить лучшие параметры модели, достигнутые на 4-й эпохе, когда валидационные потери были минимальными.

График точности (рис. 1) отображает изменение точности на обучающих и валидационных данных по эпохам. График потерь (рис. 2) отображает потери на обучающих и валидационных данных по эпохам. Оба графика показывают хорошее качество модели и стабильность обучения.



Рис. 1. График точности

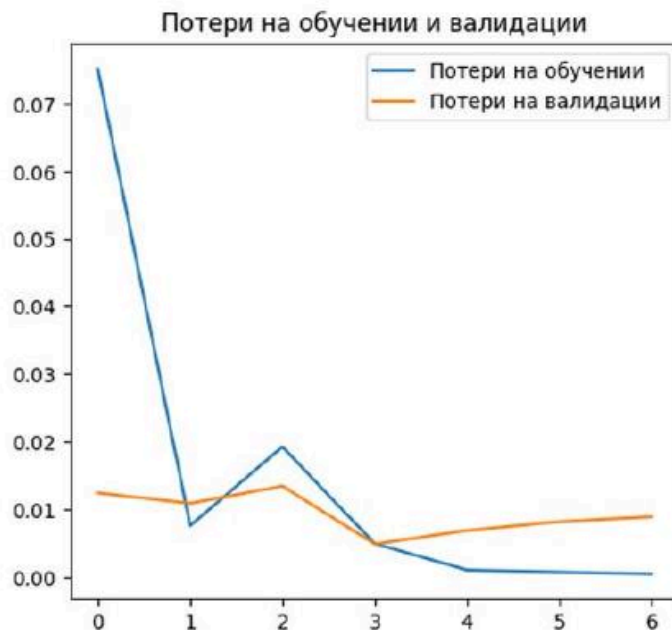


Рис. 2. График потерь

После окончания обучения модель оценивается на тестовом наборе данных, где точность составила 99,75 % (0.9974433779716492).

Из предложенных направлений наиболее целесообразным для интеграции в систему является «фильтрация ложных срабатываний». Это направление позволяет улучшить точность системы безопасности, так как снижение количества ложных срабатываний высвобождает ресурсы специалистов, уменьшая необходимость в ручной проверке несуществующих угроз. В отличие от других подходов, оно не требует существенного расширения текущего датасета, поскольку уже имеющиеся данные достаточны для обучения модели, способной фильтровать ложные сигналы. Кроме того, процесс внедрения модели фильтрации ложных срабатываний будет проще и менее затратен, что сокращает время до развертывания и снижает общие эксплуатационные расходы.

Литература

- [1] Крылов И.Д. Эффективные способы обнаружения и предотвращения XSS-уязвимостей сайтов. StudNet, 2021, т. 4, № 2, 19 с.
- [2] Рычкова А.А. Защита веб-приложений на основе детектирования XSS-уязвимостей. Современные научно-исследовательские и технологические аспекты программной инженерии. Всерос. науч.-техн. конф.: материалы. Оренбург, ОГУ, 2023, с. 218–221.
- [3] Азарнова Т.В., Полухин П.В. Обнаружение и анализ уязвимостей межсайтового скриптинга (XSS) современных интернет-приложений. Свидетельство о государственной регистрации программы для ЭВМ № 2016614976 РФ, 2016.
- [4] Горбунов А.А., Столяров Д.О., Белянин В.И. Методы защиты сайтов от XSS уязвимостей. Фундаментально-прикладные проблемы безопасности, живучести, надежности, устойчивости и эффективности систем. II Междунар. науч.-практ. конф., посв. 105-летию со дня рождения адмирала флота СССР дважды героя Советского Союза С.Г. Горшкова: материалы. Елец, ЕГУ им. И.А. Бунина, 2018, с. 133–135.

Optimizing a tool for finding XSS vulnerabilities using the LSTM model

Mitsikh V.P.

mitsikhvp@student.bmstu.ru

Vershinin E.V.

vershinin@bmstu.ru

BMSTU (KB), Kaluga, Russia

The article analyzes the problem of improving the security of web systems from cross-site scripting (XSS) through the introduction of an intelligent system based on neural networks into a software tool. Three strategic directions are being explored: optimization of website crawling, automatic classification of XSS vulnerabilities, and filtering of false positives. The difficulties associated with the development of such systems are described. The paper also presents the results of training the LSTM neural network.

Keywords: XSS, vulnerabilities, LSTM, artificial intelligence, neural networks