ДВНЗ «ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики

«До захисту допущено»
в. о. завідувача кафедри ПМІ
Наталія Маслова
(підпис) (ініціали, прізвище)
«____»_____2022 р.

Випускна кваліфікаційна робота
магістра
(освітньо-кваліфікаційний рівень)

на тему: «Дослідження підходів до забезпечення безпеки та захисту веб-додатків на основі JS»

Виконав: студент 2 курсу, групи ІПЗм-21а
(шифр групи)

спеціальності 121 Інженерія програмного забезпечення
(шифр і назва напряму підготовки)

Ясінський Олександр Юрійович _____
(прізвище та ініціали) (підпис)

Керівник доц. кафедри ПМІ, к.т.н., доц. Маслова Н. О. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Нормоконтроль
к.т.н., доц., доц.каф.ПМІ

_____ Ірина Назарова
(підпис)

Засвідчую, що у цій випускній кваліфікаційній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Луцьк – 2022

ДВНЗ «Донецький національний технічний університет»

Факультет Комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики
Освітній ступінь магістр
Спеціальність 121 Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ
в. о. завідувача кафедри
_____ /Н. О. Маслова/
«____»_____2022 року

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Ясінському Олександру Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження підходів до безпеки та захисту веб-додатків на Основі JS
Спецчастина
Керівник роботи Маслова Н. О., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від « 28 » вересня 20 22 року № 446
2. Строк подання студентом роботи 6 грудня 2022 року
3. Вихідні дані до роботи результати переддипломної практики
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз WEB–додатків, опис схеми реалізацій атак ;
2) огляд методів захисту WEB–додатків від XSS;
3)розробка моделі зловмисника та класифікація векторів атаки
4) Реалізація та розробка програмного додатку

5. Перелік графічного матеріалу
Рисунків - 15 , таблиць - 7, діаграм – 1, в цілому в кількості, достатній для демонстрації матеріалів випускної кваліфікаційної роботи

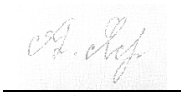
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 3 вересня 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд WEB–додатків та веб–сайтів	03.10.22 – 09.10.22	
2	Аналіз вразливостей WEB–додатків та фактори успішного розвитку міжсайтинговогу скриптингу	10.10.22 – 20.10.22	
3	Опис схеми реалізацій атак на WEB–додатки	21.10.22 – 11.11.22	
4	Розробка моделі зловмисника, класифікація векторів міжсайтинговогу скриптингу	31.10.22 – 27.11.22	
5	Пошук середовища розробки програмного додатку, перевірка властивостей Brython	28.11.22 – 05.12.22	
6	Розробка та тестування програмного додатку	06.12.22 – 19.12.22	
7	Захист роботи	20.12.22	

Студент  Ясінський О.Ю.

(підпис) (прізвище та ініціали)

Керівник роботи _____ Маслова Н.О.

(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Ясінський О.Ю. кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 121 Інженерія програмного забеспечення на тему: «Дослідження підходів до безпеки та захисту веб–додатків на основі JS»– ДВНЗ ДонНТУ, Луцьк, 2022.

Мета роботи – дослідити вразливості WEB–додатків на JS, розробити програмний додаток для пошуку XSS–вразливостей.

Об’єкт роботи – технології захисту WEB–додатків від вразливостей по типу XSS.

Предмет роботи – XSS–вразливості та методи пошуку вразливостей у WEB–додатках.

В процесі виконання роботи опановано аналітичні методи, методи пошуку й роботи з науковим матеріалом. Проведена конкретизація та реалізація знайдених підходів до створення блоку сканування вразливостей.

Застосовано методи експерименту, тестування, програмування

Основні задачі дипломної роботи:

–огляд WEB–додатків та їх вразливостей;

–дослідження існуючих методів атаки на вразливості WEB–додатків;

–обрання технології та середовища розробки;

–розробка програми сканування WEB–додатків на вразливості.

Публікації автора:

Ясінський О.Ю. “Дослідження підходів до забезпечення безпеки та захисту web–додатків на основі js” // Материали Всеукраїнської науково-практичної конференції молодих учених «ТАК»- «Телекомунікації, автоматизація, комп’ютерно-інтегровані та інформаційні технології», 2022 р., ДВНЗ «ДонНТУ. – Покровськ: ДВНЗ «ДонНТУ», 2022. – С. 3

ЗМІСТ

Стор.

Перелік умовних позначень, символів, одиниць, скорочень і терміні 7
Вступ 8
1 Системний аналіз досліджуваної області, постановка задачі 9
1.1 Веб–додатки 9
1.2 Відмінність веб–додатків від веб–сайтів 10
1.3 Огляд вразливостей WEB–додатків 12
1.4 Опис та схеми реалізацій атак, що використовують XSS–уразливості 16
1.5 Висновки за розділом 21
2 Методи захисту web–додатків від xss 22
2.1 Аналіз процесу виконання JavaScript як середовища виконання міжсайтових атак 22
2.2 Аналіз інформаційних впливів при атаці міжсайтовим виконанням сценаріїв 25
2.3 Фактори успішного розвитку атаки міжсайтове виконання сценаріїв 32
2.4 Аналіз існуючих рішень на ринку 35
2.5 Висновки за розділом 42
3 Апаратне забезпечення середовища моделювання для вивчення атаки, міжсайтове виконання сценаріїв 43
3.1 Розробка моделі зловмисника, що виконує атаку міжсайтового виконання сценаріїв 43
3.2 Розробка класифікації векторів міжсайтового виконання сценаріїв ... 46
3.3 Висновки за розділом 48
4 Реалізація та розробка програмного додатку 49
4.1 Середовище розробки для пошуку XSS–вразливостей у WEB–додатках 49

4.2 Розробка та тестування програмного додатку…… ……...50
4.3Висновки за розділом 61
Висновки..... 62
Список використаних джерел..... 63
Додаток А Зауваження нормоконтролера 66
Додаток Б Лістинг..... 67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

XSS – Cross Site Scripting

HTML – Hypertext Markup Language

WAP–сайти – мобільні сайти для телефонів

SQL–ін'єкція – один с засобів злому сайтів

URL – Uniform Resource Locator

JS – Java Script мова програмування

DOM – Document object model

CSS – Cascading Style Sheets

API – Application Programming Interface

WAF – Web Application Firewall

ВСТУП

В рамках навчання за фахом 121 Інженерія програмного забезпечення була обрана тематика роботи. Робота передбачає дослідження підходів для захисту безпеки в веб–додатках.

Обрана тематика дипломної роботи обґрунтована необхідністю дослідження безпеки веб–додатків на JS.

Робота передбачає підготовку для дослідження навчальних матеріалів, що вміщує системний аналіз досліджуваної області, постановку завдання, оцінку проблематики безпеки веб–додатків, теоретичні аспекти системи безпеки веб–додатків на основі JS.

Мета роботи – дослідити вразливості WEB–додатків на JS, розробити програмний додаток для пошуку XSS–вразливостей.

Об’єкт роботи – технології захисту WEB–додатків від вразливостей по типу XSS.

Предмет роботи – XSS–вразливості та методи пошуку вразливостей у WEB–додатках.

Основні задачі дипломної роботи:

–огляд WEB–додатків та їх вразливостей;

–дослідження існуючих методів атаки на вразливості WEB–додатків;

–обрання технології та середовища розробки;

–розробка програми сканування WEB–додатків на вразливості

Загальний обсяг роботи складає – 112 сторінок, що містять: 4 розділи, матеріали у вигляді 15 рисунків, 2 таблиць, 24 джерел інформації, загалом обсяг роботи – 88 сторінок.

1 СИСТЕМНИЙ АНАЛІЗ ДОСЛІДЖУВАННОЇ ОБЛАСТІ, ПОСТАНОВКА ЗАДАЧІ

Число компаній, які застосовують веб–технології зростає з кожним роком. Інтернет–сервіси мають безліч переваг, але зі збільшенням числа застосувань, збільшується і кількість загроз безпеки. Тому дослідження вразливостей інтернет–додатків й розробка інструментів їх захисту є важливою та актуальною проблемою. У роботі розглянуто один з найпоширеніших типів атак на сайти, пов'язаний з наявністю XSS–вразливості, яка дозволяє перенаправляти запити користувачів на шкідливі веб–сторінки, й, як наслідок, отримувати доступ до персональної або конфіденційної інформації.

1.1 Веб–додатки

Веб–додаток – клієнт–серверна програма, в якій клієнтом виступає браузер, а сервером – веб–сервер. Логіка веб–додатка розподілена між сервером та клієнтом, зберігання даних здійснюється переважно на сервері, обмін інформацією відбувається по мережі. Однією з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб–програми є міжплатформні сервісами.

Веб–програми стали широко популярними наприкінці 1990–х – на початку 2000–х років.

Незважаючи на те, що і веб–сайтом, і веб–програмою користуються за допомогою браузера, між ними є суттєві відмінності.

Веб–додаток носить інтерактивний характер і дозволяє вводити дані, динамічно обробляти їх та отримувати деякі результати. Крім того, веб–програми можна настроювати відповідно до своїх вимог. Різні фрагменти

сторінки веб–застосунку оновлюються «на льоту» після взаємодії з користувачами, як, наприклад, стрічка новин у соціальній мережі.

Веб–сайт – у комп'ютерній мережі об'єднана під однією адресою сукупність документів приватної особи чи організації.

За умовчанням мається на увазі, що сайт розташовується в мережі Інтернет. Усі веб–сайти Інтернету разом складають Всесвітню павутину. Для прямого доступу клієнтів до веб–сайтів на серверах було розроблено протокол HTTP. Веб–сайти інакше називають Інтернет–представництвом людини чи організації. Коли кажуть "своя сторінка в Інтернеті", то мається на увазі цілий веб–сайт або особиста сторінка у складі чужого сайту. Крім веб–сайтів у мережі Інтернет, також доступні WAP–сайти для мобільних телефонів.

Спочатку веб–сайти являли собою сукупність статичних документів. Нині більшості їх властива динамічність і інтерактивність. Для таких випадків фахівці використовують термін веб–додаток – готовий програмний комплекс для вирішення задач веб–сайту. Веб–програма входить до складу веб–сайту, але веб–додаток без даних сайтом є лише технічно.

1.2 Відмінність веб–додатків від веб–сайтів

Щоб користуватися можливостями веб–програми, потрібно пройти авторизацію, інакше будь–хто може отримати доступ до ваших даних та налаштувань. Як приклад веб–програми можна навести соціальну мережу. Приклад відмінностей веб–сайту від WEB–додатку наведено в таблиці 1.1.

Таблиця 1.1 – Основні відмінності між веб–додатком та веб–сайтом

Веб–сайт	WEB–додаток
Дозволяє переглядати дані	Дозволяє маніпулювати даними
Можна користуватися без автентифікації	Потрібна автентифікація
Видає заздалегідь підготовлені HTML–сторінки в основному зі статичними файлами	Фрагменти HTML–сторінки генеруються та оновлюються на високій швидкості

Продовження таблиці 1.1

Веб–сайт	Веб–додаток
Простіше у розробці; менше налаштувань для відвідувача	Потребує розробки; дає більше налаштувань для користувача. Це породжує складність, зворотний бік чого — потенційні помилки.

Незважаючи на всі відмінності, в деяких випадках межі між веб–сайтами та веб–додатками стираються – веб–сайти можуть включати веб–додатки або мати деякі характеристики. Наприклад, WordPress зберігає інформацію в реляційній базі даних MySQL та надає безліч інтерактивних функцій для адміністрування. Загальна схема WEB–додатків наведена на рисунку 1.1.

Сучасні веб–додатки складаються:

– клієнтська частина;
– серверна частина;
– сховище даних;
– інші компоненти.

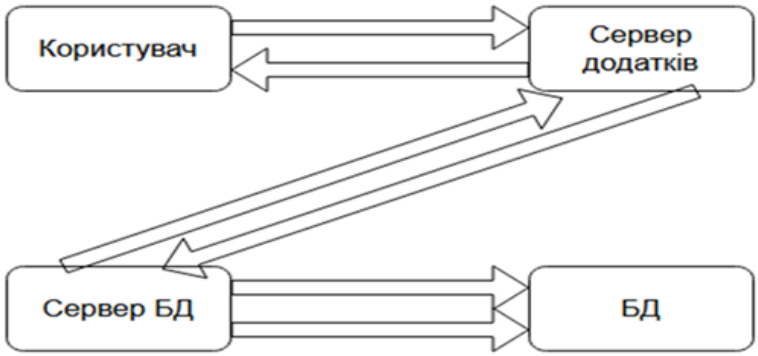


Рисунок 1.1 – Схема веб–додатків

На наш час веб–додатки включають у себе дуже багато компонентів.(таблиця 1.2) Розробили дуже багато середовищ обробки та зберігання даних.

Таблиця 1. 2. Компоненти сучасних веб–додатків

Клієнтська частина	Серверна частина	Зберігання даних	Інші компоненти
Electron	Nginx	Files System	Reverse–Proxy
JavaScript	Apache	MySQL	Load Balancer
TypeScript	Apache Tomcat	PostgreSQL	Proxy Server
ReactJS	Microsoft IIS	mongoDB	Cache Server
Redux	Gunicorn	Oracle	WebSocket
AngularJS	uWSGI	IBM DB 2	Web Application Firewall
Vue.js	Java	Microsoft SQL Server	
jQuery	Python	SQLite	
	PHP	Redis	
	NodeJs	Memcached	
	C#	Neo4j	
	Go	Tarantool	
	RubyOnRails	XML	
		YAML	

1.3 Огляд вразливостей WEB–додатків

У комп'ютерній безпеці термін «уразливість» використовується для позначення нестачі в системі, використовуючи який можна навмисно порушити її цілісність і викликати неправильну роботу.

Комп'ютерна атака – цілеспрямований несанкціонований вплив на інформацію, ресурс автоматизованої інформаційної системи або отримання несанкціонованого доступу до них із застосуванням програмних або програмно–апаратних засобів.

У процесі еволюції WEB–додатків розробники стикалися з низкою проблем, серед яких особняком стоїть проблема безпеки WEB–додатків. Проблеми безпеки випливають з основних особливостей WEB–додатків, наприклад, через неможливість ізолювати WEB–додаток від спроб несанкціонованого доступу ззовні через їх доступність.

Завдяки поширенню мережі Інтернет все більше вразливостей, загрожують мільярдам користувачів мережі. Тому виникає потреба у безпечному середовищі для Інтернет–серфінгу. Фахівці з інформаційної безпеки поділяють усі веб–вразливості на дві великі категорії.

Перша категорія включає захист веб–додатків, що складається з користувацьких та серверних програм. Друга категорія – це мережна безпека, яка включає бази даних, програми, операційні системи та мережі (Рисунок 1.2).

Як правило, порушника цікавить уразливості на рівні веб–сервера та прикладного програмного забезпечення, які часто експлуатує при проведенні різних видів атак, зокрема XSS.

Безпека мережевих технологій знаходиться на вищому рівні, на відміну від безпеки веб–додатків, тому необхідно приділити більше уваги більше слабкій ланці, щоб уникнути ситуацій, що завдають шкоди користувачам і прийняти заходи щодо скорочення втрат. Одним із небезпечних комп'ютерних атак на веб–додатки є міжсайтовий скриптинг.(рисунок 1.2)

«Міжсайтовий скриптинг» – це комп'ютерна атака, що полягає в

введення шкідливого коду у вхідні параметри веб–сторінки, що надсилаються веб–браузеру користувача. Впроваджуваний код схожий на SQL–ін'єкцію і може бути використаний у різний спосіб [1]. XSS прийнято класифікувати за двома критеріям: за вектором та способом впливу (Рисунок 1.3).

За способом впливу XSS–атака може бути [2]:

– «активна» XSS – не потребує будь–яких зайвих дій з боку користувача з погляду функціоналу веб–програми;

– "пасивна" XSS – спрацьовує при виконанні користувачем певної дії ("клік" або наведення покажчика миші і т.п.).

За вектором впливу XSS–атаки поділяються на такі види:

– «відбита» (reflected) XSS;

– «зберігана» (stored) XSS;
– "XSS заснована на DOM–моделі" (DOM–XSS).

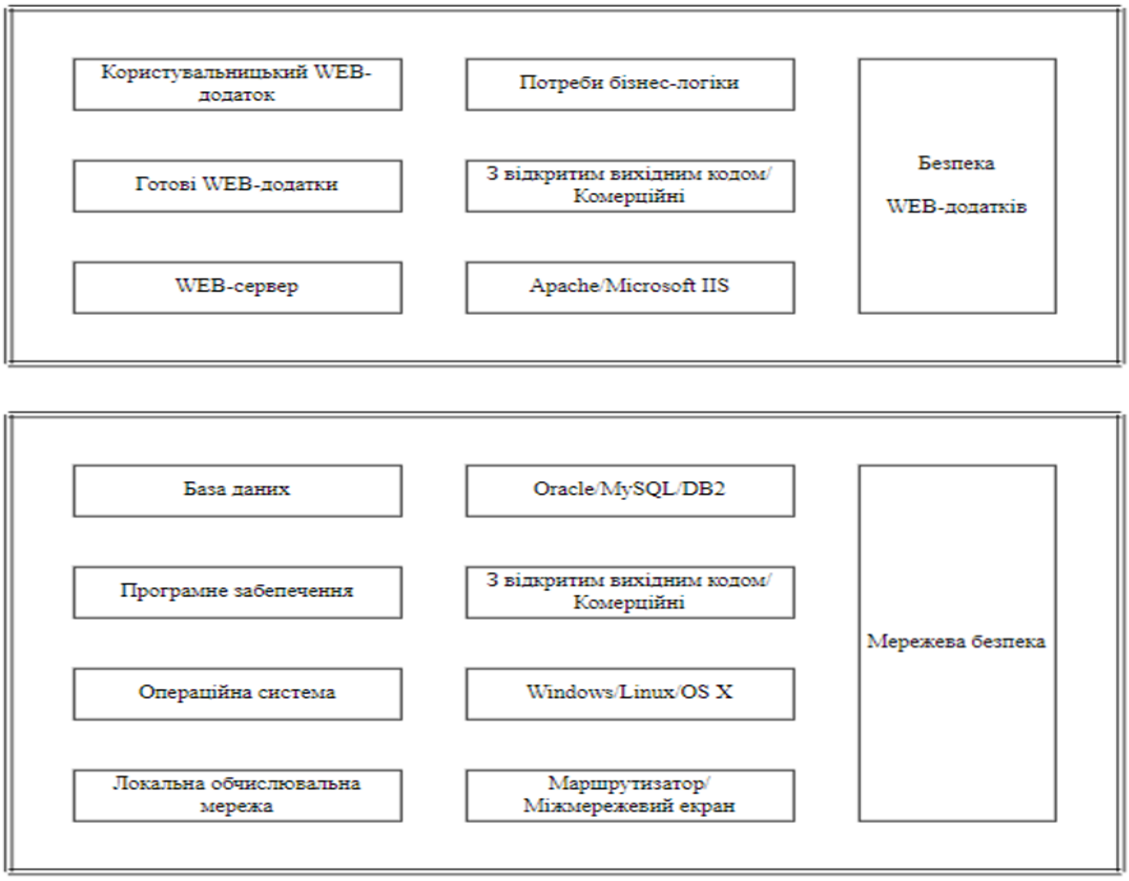


Рисунок 1.2 – Категорії WEB–уразливостей

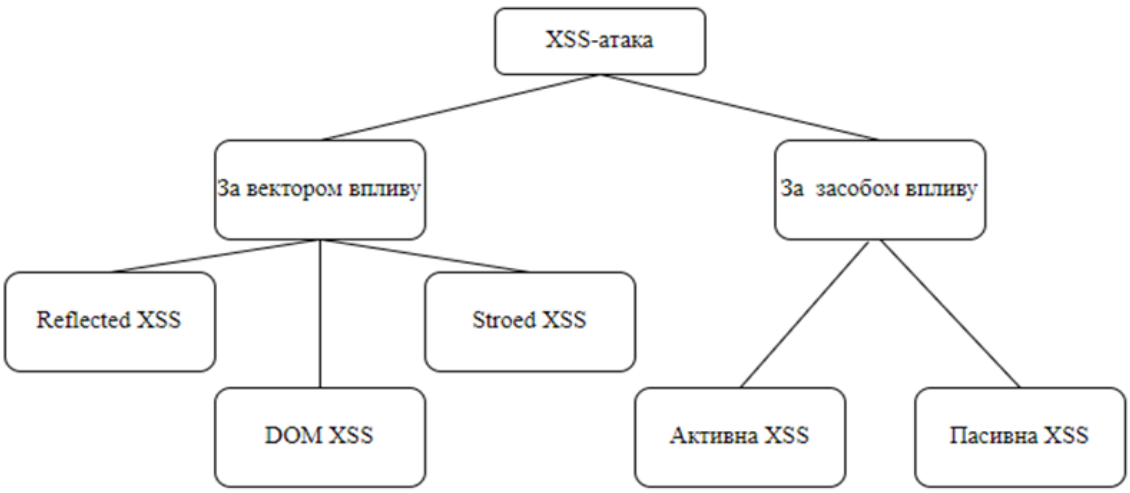


Рисунок 1.3 – Класифікація XSS

Stored XSS–атаки можливі тоді, коли порушнику вдасться впровадити на сервер шкідливий код, який виконується в браузері користувача щоразу при зверненні до оригінальної сторінки. Класичним прикладом цієї вразливості є форуми, на яких можна залишати коментарі в HTML–форматі.

Відмінною особливістю «відбитої» XSS–атаки від «зберігається» є те, що комп'ютерна атака спрацьовує тоді, коли користувач переходить за посиланням порушника. Ці вразливості з'являються, коли дані, надані веб–клієнтом, виконуються безпосередньо серверними скриптами для синтаксичного аналізу та відображення сторінки. Найчастіше ці вразливості перебувають у параметрах HTTP–запиту чи формі HTML через некоректної обробки.

XSS–уразливості засновані на «DOM–моделі» також часто використовуються зловмисниками для проведення атак. Насамперед необхідно дати визначення терміну DOM.

DOM (від англ. Document object model – "об'єктна модель документа") – об'єктна модель, що використовується для XML/HTML–документів [3]. За допомогою DOM можна генерувати деревоподібну структуру документів HTML. Тому легко можна змінювати або контролювати об'єктну модель документа. Цей вид XSS–атаки повністю відрізняється від збереженої та відображеної так як шкідливий код не впроваджується у веб–сторінку.

XSS–вразливість, заснована на об'єктній моделі документа, виникає безпосередньо в JavaScript сценарію. Це є серйозною вразливістю, і звичайний захист від XSS–атак не буде працювати під час цієї атаки.

1.4 Опис та схеми реалізацій атак, що використовують XSS–уразливості

«Відбита» XSS–атака є найбільш поширеною, і вразливість, яку експлуатує даний вид комп'ютерної атаки, легка у детектуванні.

При «відбитій» XSS–атаці передача коду серверу та повернення його до користувача здійснюється в рамках одного HTTP–запиту. Процес проведення

міжсайтового скриптингу з використанням відбитої XSS–уразливості складний. Це пов'язано з тим, що шкідливий код необхідно впровадити в URL–адресу, потім URL із впровадженим шкідливим кодом надіслати користувачеві, щоб він перейшов за посиланням для запуску шкідливого коду. Однак складність реалізації не є перешкодою для порушників.

При проведенні базової відбитої XSS–атаки не переслідується мета крадіжки конфіденційної інформації. Суть атаки полягає в тому, що при переході на вебсайт з «відбитою» XSS–уразливістю виводиться попереджувальне вікно. Це результат виконання скриптового коду Посилання для переходу на веб–сторінку з базовою «відбитою» XSS–уразливістю виглядає приблизно так:

http://site.ru/<script>alert ("XSS успішно виконаний"</script>).

Якщо порушник відправить цю URL свою жертву в такому вигляді, то швидше за все просунутому користувачеві веб–адреса здасться дивною, і він не перейде по даній засланні. Тому одним із методів приховування шкідливого посилання є використання кодування RFC 1738 [4]. Як видно з фрагмента коду, наведеного нижче, після кодування шкідливого URL–адреса скриптовий код стає важко розпізнаваним:

```
URL–адреса з XSS:
http://site.ru/<script>alert('XSS')<script>
URL–адреса з XSS після кодування:
http:%3a%2f%2fsite.ru%2f%3cscript%3ealert(%27XSS%27)%3cscript%
```

Порушники використовують «відбиту» XSS–уразливість з іншими комп'ютерними атаками, щоб підвищити ефект від експлуатації вразливості. У зокрема, з'явився новий вид комп'ютерної атаки, так званий клікджекінг. Його суть полягає в тому, що користувач, здійснюючи «Клік» на легітимній

сторінці, насправді здійснює перехід за посиланням, сформованою зловмисником [5].

Особливості «клікджекінгу»:

– крадіжка посилання відбувається, коли користувач натискає кнопку, сформовану порушником, або переходить за посиланням;

– немає необхідності в ін'єктованих скриптах, що робить атаку простіше в реалізації;

– комп'ютерна атака заснована на Dynamic HTML;

– для захоплення посилання зловмисник лише повинен контролювати деяку частину веб–сайту.

У свою чергу завдяки «клікджекінгу» з'явився інший вид комп'ютерної атаки, яка поєднує JavaScript і тег Iframe. Фахівці з ІБ назвали цю комп'ютерну атаку Cross–Frame Scripting (XFS). Ця комп'ютерна атака поєднує в собі комбінацію шкідливого JavaScript–коду (JS–код) з HTML тегом Iframe, який завантажує легітимну сторінку для крадіжки даних користувача. XFS–атака зазвичай буває успішною лише у поєднанні із соціальною інженерією.

Розглянемо приклад реалізації XFS–атаки [6]. Порушник, застосовуючи методи соціальної інженерії, переконує користувача перейти на спеціально сформовану сторінку. Після переходу на сторінку порушника у веб–браузер користувача підвантажуються шкідливий JavaScript–код (JS–код) та Iframe. Коли користувач вводить облікові дані в Iframe, що вказує на легітимний сайт, шкідливий JS–код краде натискання клавіш.

Існує ще один вид XSS–атаки, що експлуатує «відбиту» XSS вразливість веб–програми, так званий «Redirection XSS». Це вид XSS–атаки, в якому шкідливий код підвантажується з ненадійного джерела та впроваджується в різні скрипти веб–програми. Атака "Redirection XSS" виглядає наступним чином:

– дані надходять до веб–ресурсу з ненадійного джерела у вигляді веб–запиту;

– отримані дані «вклинюються» в динамічний контент веб–додатку, який надсилається веб–користувачу без перевірки на наявність шкідливого коду.

Шкідливий контент, що надсилається до веб–браузера користувача, часто приймає вигляд JS–коду, але може також включати HTML, Flash або будь–який інший тип коду, який може бути оброблений веб–браузером.

Комп'ютерні атаки, що використовують «збережені» XSS–уразливості, є, порівняно з «відбитими» XSS, більш небезпечними, оскільки вони можуть бути використані протягом тривалого часу. «Зберігані» XSS–атаки в основному використовуються для крадіжки сесійної інформації (cookies чи іншої інформації про сеанс). Також можливо використовувати цю атаку для перенаправлення користувача на веб–ресурс зловмисника або виконувати інші шкідливі операції на ПК користувача.

Комп'ютерна атака під назвою «крадіжка куків» (від англ. Stealing cookies). Як відомо, cookies – це невеликий фрагмент даних, що зберігається у веб–браузері користувача і включає в себе дані користувача і багато корисної інформації, використовується для розпізнавання користувача під час доступу до веб–програми.

Cookies генеруються на стороні веб–сервера і надсилається веб–браузеру користувача для збереження її у локальному сховищі веб–браузера. При подальшому зверненні користувача до веб–застосунку, збережені cookies будуть відправлені веб–серверу для розпізнавання користувача та надання доступу до веб–ресурсу, ґрунтуючись на інформації, записаної у cookies [7].

Іноді порушники для здійснення довгострокової комп'ютерної атаки

використовують XSS–вразливість для перенаправлення користувачів на завантаження троянського коня. Фахівці з ІБ називають цю атаку "XSS Based Trojan Horse", в перекладі з англійської означає «XSS атака з використанням троянського коня» [7].

При проведенні цієї атаки порушнику необхідно, щоб користувач скачав троянську програму. Для цього порушник використовує «збережену» XSS вразливість веб–застосунку. В область веб–програми з «збереженою» XSS–вразливістю впроваджується шкідливий код для завантаження троянської програми. Будь–хто, хто відвідає цю сторінку автоматично перейде за посиланням зі шкідливим JS–кодом. JS–код обробиться браузером користувача і у фоновому режимі завантажить частину, що залишилася шкідливого коду на комп'ютері користувача. При цьому жертва не помітить комп'ютерну атаку, так як використовується Iframe для створення дочірнього вікна, налаштуваннях якого прописано нульовий розмір.

«XSS–worm» для посилення ефективності атаки, об'єднує «збережену» XSS вразливість та методику проведення комп'ютерної атаки за допомогою «мережевого черв'яка». Для успішного розповсюдження "хробака" необхідно компактно написати код, так як у веб–додатках встановлюють певні обмеження введення даних користувачем.

Для обходу обмежень порушник використовує певні методи. Іноді порушник викликає шкідливий код зовні, якщо у веб–додатках немає прав користувача. Існує ще один вид комп'ютерної атаки експлуатує «зберігаючу» XSS–уразливість, так звана «XSS tunneling» (у перекладі з англійської «XSS тунелювання»).

Завдяки комп'ютерній атаці XSS tunneling порушник має можливість постійно контролювати комп'ютер користувача. На рисунку 1.4 представлена схема реалізації «XSS–тунелювання».

Порушник через XSS–Shell сервер посилає команди, які виконуватимуться на комп'ютері жертви. Таким чином XSS–тунелювання забезпечує передачу даних HTTP трафіку між порушником та користувачем.

Наступна комп'ютерна атака експлуатує XSS–уразливість у DOM–моделі, що виникає на стороні користувача під час обробки даних у JavaScript сценарію. [13]

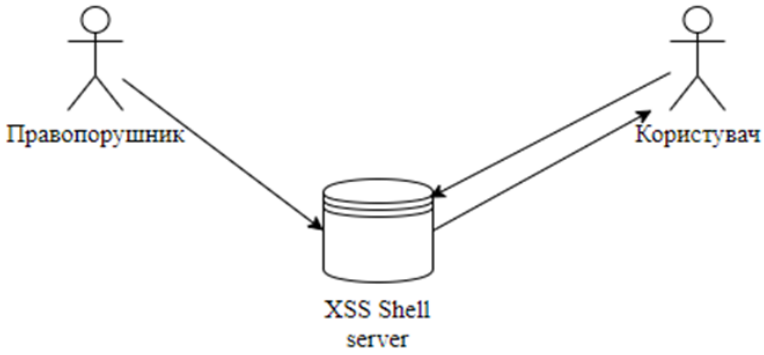


Рисунок 1.4 – Схема реалізації «XSS–тунелювання»

Даний тип XSS–атаки отримав таку назву, оскільки реалізується через DOM (Document Object Model), який не залежить від платформи та мови програмного інтерфейсу, що дозволяє програмам і сценаріям отримувати доступ до вмісту HTML та XML–документів, а також змінювати вміст, структуру та оформлення таких документів [8]. При некоректній фільтрації можна модифікувати DOM атакованого сайту і домогтися виконання JS–коду в контексті сайту, що атакується.

Розглянемо докладніше механізм проведення DOM XSS. Цей вид XSSатаки використовує довірений, керований сервером скрипт, що надсилається користувачеві веб–додатки. Наприклад, JS–код здійснює перевірку працездатності форми перед відправкою серверу заповнених даних користувачем. Скрипт обробляє введені дані, потім вставляє їх назад у веб–сторінку (наприклад, з Dynamic HTML). Це дозволяє реалізувати «DOM XSS» тобто. "вклинити" шкідливий код у сценарій JS–коду.[14]

1.5 Висновки за розділом

У даному розділі було досліджено веб–додатки. Реалізацію атак на веб–додатки, а також відмінність веб–додатків від веб–сайтів. За цими даними визначенно що WEB–додатки мають ряд вразливостей які надають доступ до кофедеційних даних. Визначено можливі підходи до атаки на WEB–додатки. Розглянуто вразливості WEB–додатків та їч наслідки.

2 МЕТОДИ ЗАХИСТУ WEB–ДОДАТКІВ ВІД XSS

Комп'ютерні атаки на основі міжсайтового скриптингу потенційно небезпечні для більшості веб–програм і браузерів. Порушники постійно винаходять нові типи комп'ютерних атак, але наступні заходи зможуть допомогти захистити системи від цих комп'ютерних атак. Для захисту від XSS–атак застосовуються різні методи як на серверах, і у браузерах клієнтів.

Завдяки неуважності та недбалості користувачів, порушник з легкістю може отримати несанкціонований доступ до даних користувача.

Тому виникає необхідність підвищення обізнаності користувачів про вразливість та способи захисту від XSS–атак у процесі Інтернет–серфінгу. Основним способом захисту від XSS–атак є відключення всіх мов сценаріїв. На жаль, користувачі не можуть відключити мову сценаріїв, тому що вона спричинить відключення багатьох функцій веб–сайтів.

2.1 Аналіз процесу виконання JavaScript як середовища виконання міжсайтових атак

Історія появи вразливості міжсайтового скриптингу бере свій початок із 1996 року. Це період перших років створення World Wide Web. У США тільки почала з'являтися електронна комерція, активно розвивалися компанії Netscape, Yahoo. У ті роки тисячі веб–сторінок були лише на стадії розробки. Для створення веб–сайтів використовувалася гіпертекстова мова розміток (HTML), яка не містила інтерактивних елементів і була статичною. Все змінилося із запровадженням JavaScript.[16] Веб–програми знаходять широке застосування в сьогоднішньому житті. Все більше людей та організацій сильно залежать від їхнього правильного функціонування мережі Інтернет, внаслідок чого збільшується попит на надійність та безпеку. JavaScript – це мова програмування, що інтерпретується, з можливостями об'єктно–орієнтованого

програмування. З погляду синтаксису базова мова JavaScript нагадує C, C++ та Java такими програмними конструкціями, як інструкція if, цикл while та оператор &&. Однак ця подоба обмежується синтаксичною схожістю. Спочатку JavaScript розроблявся з метою вбудовування в будь–які програми та надання можливості виконувати сценарії.[17]

Ядро JavaScript містить базовий набір об'єктів, таких як масиви (Array), дата (Date) та арифметичні (Math), а також базовий набір мовних елементів, наприклад, оператори, що управляють структурами та виразами. Ядро JavaScript може бути розширене для різних цілей шляхом доповнення додаткових об'єктів. JavaScript на клієнтській стороні (Client–side JavaScript) розширює базову мову об'єктами для керування браузером та його об'єктною моделлю документа (DOM).[15]

Наприклад, клієнтські розширення дозволяють додаткам розміщувати елементи в HTML–формах і тим самим реагувати на події користувача, такі як клік миші, введення у форму і навігація по сторінках. JavaScript на стороні сервера розширює базову мову об'єктами, що стосуються виконання JavaScript на сервері. Наприклад, серверні розширення дозволяють програмі обмінюватися даними з реляційною базою даних забезпечують цілісність інформації від одного запуску додатка до іншого або виконують маніпуляції з файлами на сервері [18].

Netscape винайшла JavaScript, і JavaScript був уперше використаний у браузерах фірми Netscape. Але Netscape працює спільно з CMA (European Computer Manufacturers Association/Європейська Асоціація Виробників Комп'ютерів) для створення стандартизованої міжнародної мови програмування на базі ядра JavaScript [19]. CMA є асоціацією міжнародних стандартів у галузі інформації та систем комунікації. Ядро мови JavaScript підтримує роботу з такими простими типами даних, як числа, рядки та булеві значення [20]. Стандартизована версія JavaScript, звана CMAScript,

поводиться абсолютно однаково у всіх додатках, що підтримують цей стандарт.

Компанії можуть використовувати цю відкриту стандартну мову для розробки своїх реалізацій JavaScript. Перше видання стандарту CMA задокументовано у специфікації CMA–262. Друге видання CMA–262 містило лише правки та не вносило змін. Метою оновлення стандарту було приведення у сувору відповідність із ISO/I C–16262, при якому не допускалося доповнення, зміни або видалення тексту, зазначеного у стандарті. CMAScript не використовують друге видання як міру відповідності. Третє видання CMA–262 було першим повноцінним оновленням стандарту. Здійснено оновлення обробки рядкових параметрів, визначення помилок та робота з виведенням числових значень. Також додано підтримку регулярних виразів, нові керуючі оператори, обробки винятків.

Для багатьох це ознаменувало прихід CMAScript як справжньої мови програмування. Спочатку веб–сервери компанії Netscape включали інтерпретатор.[21]

При відсутності контролю типів JavaScript дозволяє допускати помилки, яких припускаються недосвідчені програмісти. Завдяки цьому багато веб–дизайнерів можуть використовувати JavaScript для вирішення обмеженого кола завдань, що виконуються за точними рецептами [17].

Хакери знайшли свої плюси у цій мові. Коли користувач відкривав свою веб–сторінку в HTML фреймі, то міг бути перенаправлений на будь–який інший веб–сайт в межах того самого вікна браузера. Таким чином відбувалася крадіжка імен користувачів, паролів, файлів cookie, компрометація будь–якої інформації на екрані. ЗМІ повідомили про цю проблему як про вразливість веб–браузера [22].

Захист коду JavaScript, що виконується, заснована на механізмі пісочниці, в якій може виконуватися тільки обмежене коло дій, а не завдання програмування загального призначення. Тобто JavaScript програми

розглядаються як ненадійні програмні компоненти, які отримують доступ тільки до обмеженої кількості ресурсів у браузері. JavaScript браузера використовує механізм пісочниці, обмежуючи сценарії доступу до ресурсів, пов'язаних із сайтом. На жаль, ці механізми захисту не приносять користі, якщо користувача заманять на завантаження шкідливого коду JavaScript з проміжного, надійного сайту. У цьому випадку шкідливий скрипт отримує повний доступ до всіх ресурсів (наприклад, маркери автентифікації та cookie), які належать до довіреного сайту. Крім того, JavaScript програми, завантажені з різних сайтів, захищені від одного механізмом, званим "Правило обмеження домену". Політика дозволяє сценаріям, що знаходяться на сторінках одного сайту, доступ до методів та властивостей один одного без обмежень, але запобігає доступу до більшості методів та властивостей сторінок на різних сайтах. Хоча інтерпретатори JavaScript мають ряд недоліків, більшість веб–ресурсів використовують функціональність JavaScript.

Проблема існуючих механізмів безпеки JavaScript полягає в тому, що сценарії можуть бути обмежені механізмом пісочниці та відповідати правилу обмеження домену, але порушувати політику безпеки системи. Таке порушення може бути досягнуто під час завантаження шкідливого коду самим користувачем із довіреного веб–сайту.

2.2 Аналіз інформаційних впливів при атаці міжсайтовим виконанням сценаріїв

У грудні 1999 року Девід Росс продемонстрував, що вміст веб–сторінок вразливий до включення скриптів, обійшовши захист, вбудований у код Internet Explorer. Але виявилося, що помилка існує за сервера, а застосовується за клієнта. Девід Росс описав проблему у внутрішньому бюлетені безпеки Microsoft, названому "Script Injection". Стаття описувала суть проблеми, як експлуатувати вразливість, як напад може бути відновлено за допомогою cookie. Техніка використання вразливості отримала назву міжсайтового

виконання сценаріїв. Під вразливістю у разі розуміється слабкість, яка дозволяє зловмиснику знизити рівень захищеності інформаційної системи.[12]

Одночасно з появою "Script Injection" виявлено схожий метод атаки, у якому зловмисник може використовувати передачі довільних команд на дочірній сервер. У такому випадку програма дозволяє зробити включення довільного коду, приймає введення з ненадійних джерел і вставляє його в команди, які надсилаються операційній системі без відповідної перевірки вхідного або вихідного кодування. Такий метод отримав назву HTML injection.

2 лютого 2000 року фахівцями з безпеки компанії «Microsoft», спільно з фахівцями CERN випущено бюлетень безпеки, який об'єднав обидві проблеми безпеки. Описані в бюлетені атаки дозволяли підключити сторонній код з недовіреного сайту, водночас браузер коректно відображав вміст усередині надійного сайту. Специфіка полягає в тому, що для атаки на сервер як засіб атаки використовується авторизований на цьому сервері клієнт. Новий вид загрози отримав назву «Cross site scripting». [20]

20 лютого 2000 року CR опублікувала інформацію про знову виявлені вразливості, що стосуються всіх серверних програм. Ця вразливість відома як Cross–Site Scripting (XSS), в результаті якої програми помилково довіряють даним, що повертаються від клієнтів. Наприклад, поле адреси сайту може використовуватися для вставки скриптів, що виконуються. У 2011 році за оцінкою Мітрі XSS посіла 4 місце з 25 у категорії найнебезпечніших програмних помилок. Найбільш популярні види веб–застосунків, які стають жертвами XSS атак: пошукові системи, дошки оголошень, електронні поштові системи та системи миттєвих повідомлень. Навіть найвідоміші сайти в сучасному світі, такі як Google, Yahoo!, Facebook, PayPal та Вікіпедія ставали жертвами, і все ще дуже чутливі до багатьох видів XSS атак.

Найчастіше використовувані мови програмування під час атаки XSS: H ML, XH ML, JavaScript та Adobe Flash. Однак найпопулярнішим і потенційно найбільш шкідливим в руках зловмисника є JavaScript . Точкою ін'єкції є

параметр (POST, GET, Cookie і т.д.), дані якого можуть бути передані на сервер, а потім відображаються в відповіді HTML сервера . Розробники веб–додатків повинні гарантувати, що всі введені користувачем дані аналізуються та фільтруються належним чином. Під введенням користувача розуміються рядки запитів, URL, і взагалі всі постійні дані, які передаються між браузером і веб–сервером.

Сервера, які використовують лише статичні сторінки, мають імунітет до атаки міжсайтового виконання сценаріїв, оскільки мають повний контроль над тим, як інтерпретуватиметься вміст їхніх сторінок.

Міжсайтовий скриптинг дозволяє здійснювати перехоплення сесії. Оскільки ідентифікатори сесії найчастіше розповсюджуються через cookie, JavaScript дозволяє прочитати їх та передавати на контрольований зловмисником сервер. Потім зловмисник може відтворити цю сесію на вразливому сайті замість жертви. XSS атаки є серйозною проблемою, що призводить до поширення шкідливих програм та автоматизації дій браузера жертви з метою подальших атак.

Нижче наводиться короткий список потенційного шкоди, яка може бути викликана XSS атаками:

– крадіжка та продовження сесії авторизованої жертви;

– записуючі натискання клавіш жертви, у тому числі при введенні паролів;

– перенаправлення трафіку на сторонні ресурси з метою організації атаки на відмову в обслуговуванні;

– зондування внутрішньої мережі жертви;

– запуск сторонніх програм доступних браузеру.

Веб–розробники використовують безліч веб–технологій для створення кращого інтерфейсу користувача, здатного ефективно розподіляти пропускну здатність.

Реалізація таких нових технологій підвищує вразливість веб–додатків для XSS атак. Знання новітніх веб–технологій та широке використання інтерактивного контенту через Інтернет є важливим для розробки програмного забезпечення. Розробникам необхідно знати та дотримуватися належних основ безпеки під час усього життєвого циклу програмного забезпечення. За механізмом виконання атаки XSS умовно поділяються на три категорії: відбиті, стійкі та засновані на DOM.[22]

Відображені XSS мають на увазі, що скрипт не зберігається на сервері вразливого сайту або він не може автоматично виконатися в браузері жертви. Для спрацювання пасивної XSS потрібна певна додаткова дія, яку має виконати браузер жертви. Ця версія XSS напад нагадує атаку "людина посередині", тому що вимагає від атакуючого відправити шкідливе посилання електронною поштою або вставити її на довіреному сайті, щоб направити жертву до шкідливого скрипту. У разі успіху ми можемо перехопити та змінити інформацію, яка надсилається жертві з мережі. Оскільки міжсайтовий скриптинг дозволяє перенаправляти користувача на сторонні ресурси, атака часто застосовується при фішингу з метою перенаправлення користувачів на сайт зловмисника. Їх також називають першим типом XSS.

При стійких XSS шкідливий скрипт зберігається на сервері і спрацьовує в браузері жертви при відкритті будь–якої сторінки зараженого сайту. Атака зазвичай відбувається, коли HTML або Javascript отримує дані, які зберігаються на сайті, а потім відображаються на сторінці. Використання цього методу не обмежується лише крадіжкою даних браузера. Зловмисник може отримати у своє розпорядження розширений код JavaScript, завдяки якому моделюється, наприклад, вихід із авторизованого сайту. Пропонується хибна форма входу, яка збереже введені користувачем дані авторизації. Як тільки буде зібрано інформацію, сценарій поверне авторизовану сторінку користувачеві і передасть вкрадену інформацію атакуючому. Виявлення атак XSS під час використання асинхронного JavaScript є важчим завданням для

інструментів автоматичного аналізу, ніж у класичних веб–додатках. Це з тим, що сканерам доводиться розбирати JavaScript код сайту за сервера щоб одержати імен сценаріїв. Крім того, вилучення імен параметрів може вимагати виконання коду JavaScript. Їх також називають другим типом XSS. Стійкому скриптингу піддаються сайти, так званого веб–2.0 – форуми, блоги, гостьові книги та соціальні мережі.

При атаці через DOM ознакою вразливого сайту може бути наявність H ML сторінки, яка використовує дані з document.location, document.URL чи document.referrer (або будь–яких інших об'єктів, на які може впливати атакуючий) небезпечним способом. При виконанні Javascript коду в браузері він отримує доступ до кількох об'єктів, представлених в рамках DOM (Document Object Model – Об'єктна модель документа). Об'єкт document є головним серед цих об'єктів і надає доступ до більшості властивостей сторінки. Цей об'єкт містить багато вкладених об'єктів, таких як location, URL та referrer, що використовуються небезпечним способом.

При проведенні атаки оригінальний код Javascript на сторінці не перевіряє параметри за замовчуванням, що містять H ML розмітку, і повторює шкідливий сценарій атакуючого на сторінку (DOM) під час завантаження. При неправильній реалізації викликів об'єктів DOM, таких як val () або document() в полі, зловмисник може викликати атаку, де DOM і H ML5 об'єкти, можуть бути використані одночасно. Це розширює можливості атаки та дозволяє отримати більше точок входу для зловмисників. Наслідком буде XSS напад на H ML 5 додаток, що працюють віджети, об'єкти Mashup і т.п..В даний час можливість виявлення стійких XSS засобами автоматичного пошуку вразливостей вкрай мала. Практично немає можливості виявлення DOM–based атак (Duchene et al. 2012). Наведена вище класифікація приймається багатьма дослідниками безпеки веб–застосунків. Fogie використовує цю класифікацію вказуючи, що шкідливий код, зазвичай написаний мовою гіпертекстової розмітки чи Javascript, не виконується на сервері. Сервер тільки зберігає код,

тоді як атака виконується у браузері. Хакер лише використовує надійний сайт як канал для виконання атаки.[19]

Mcallister також розрізняє відбитий і стійкий XSS, але розмотує класифікацію з погляду застосування атак сканерами. Він визначає стійкий XSS так: «У разі стійкої XSS вразливості зловмисно введені дані не відразу повертаються клієнту, але зберігаються в базі даних, а потім включаються в інший запит».

Адам Доуп використовує дещо іншу класифікацію. Він розрізняє відбитий XSS, стійкий XSS і багатоступеневий XSS, що зберігається, при виконанні якого від користувача потрібно виконати додаткову дію, перш ніж вразливість дозріє, наприклад, клацнути за посиланням або на кнопці .

Усі три публікації використовують практично однакову класифікацію атак, що відрізняється лише незначною мірою. Неофіційно аналогічна класифікація описана у документі WASC Thread Classification, що є неофіційним класифікатором загроз Інтернету.

Фонд OWASP пропонує ширшу класифікацію, вказуючи атаки XSS, як підмножина атак уразливості Input Validation Data. Input Validation Data є найбільш поширеною проблемою безпеки додатків мережі Інтернет і полягає у відсутності коректної обробки введених користувачем даних або в неправильній обробці даних, що одержуються додатком ззовні. Найбільш поширені форми атак:

– SQL Injection дозволяє зловмиснику керувати базою даних через погану перевірку вхідних даних. SQL команда передається веб–додатком внутрішній базі даних, а потім виконує зміну інструкції SQL. Атака може призвести до розголошення даних, що зберігаються в базі та, можливо, повної компрометації сервера бази даних;

– LDAP Injection вразливість дозволяє контролювати зловмисником зміну LDAP запитів з сервера хостингу веб–застосунків. Ці вразливості

можуть призвести до розкриття інформації та несанкціонованого доступу зловмисника;

– RCE дистанційне виконання коду дозволяє виконати довільний код на цільовій системі. Успішна атака може вимагати координації між кількома запитами із збереженням їхнього стану та кількома скриптами на стороні сервера.

Відомо, що ін'єкції сценаріїв залежать від контексту. Наприклад, виклик expression: Alert(a) є нешкідливим у контексті H ML тега, але може призвести до виконання JavaScript, коли є частиною атрибуту у контексті CSS.

У деяких випадках XSS напад можна виконати підключенням коду, який генерується на веб–сервері на різних сторінках. Ця атака нагадує класичну атаку return–to–libc лише з використанням власного коду сервера. Поворотно–орієнтоване програмування передбачає, що вразливість може просто знаходиться в libc, що може призвести до виконання довільного коду від імені нападника. Більше того, атака може здійснюватися без виклику функцій, але з використанням допустимих комбінацій існуючого коду. Залежно від атаки об'єктом може виступати будь–яке програмне забезпечення, що має можливість отримання даних від користувача. Це можуть бути документи, що використовують мови сценаріїв, плеєри, меседжери, бази даних, але найпоширенішим об'єктом атак є Інтернет–браузер. На відміну від проблем, пов'язаних, наприклад, з SQL–ін'єкціями, XSS атаки не впливають на стороні сервера, фактично експлуатація відбувається у браузері жертви. Таким чином, адміністратор веб–додатку має лише обмежені свідчення успішних атак XSS. Тому атаки часто не беруться до уваги або виявляються досить пізно.

Міжсайтовий скриптинг дозволяє створювати сценарії, що самовідтворюються, що містять зловмисний код. Такі сценарії називаються XSS черви. Звичайні віруси можуть мати доступ до файлової системи та пам'яті, але XSS черв'яки повинні виконуватися у веб–браузері віртуальної машини. Незважаючи на це обмеження, XSS черв'яки можуть здійснювати

атаки DDoS, розсилку спаму, а також надавати інформацію про браузер. Поєднання знання доступних кодувань на стороні сервера для обходу механізму фільтрації та стороні клієнта для коректної інтерпретації JavaScript може генерувати XSS атаки автоматично. Моделювання поведінки хробаків показує їхню високу небезпеку і підпорядкованість статечному закону розподілу.[21]

2.3 Фактори успішного розвитку атаки міжсайтове виконання сценаріїв

Браузер – це програмне забезпечення для перегляду веб–сайтів, тобто для запиту веб–сторінок, їх обробки, виведення та переходу від однієї сторінки до іншої. В експериментальній частині роботи показано, що наявна класифікація типів атак є достатньою при атаках на серверну частину JavaScript, але значну роль в успішному проведенні атак грає браузер, який використовується на боці об'єкта атаки. Політика безпеки в браузері спрямована на обробку та переривання виконання ненадійного коду на стороні клієнта. Політика ґрунтується на таких припущеннях:

- веб–браузери використовують усі необхідні інструменти з метою виявлення та відображення сценарію;

- розробник веб–програм точно знає, які сценарії є довіреними для браузера. Основною частиною браузера є програма для перетворення H ML–розмітки сайту у зрозуміле для користувача подання у браузері, так званий браузерний движок, що перетворює вміст веб–сторінок (файли H ML, XML, цифрові зображення тощо) та інформацію про форматування (у форматах CSS, XSL тощо) в інтерактивне зображення форматованого вмісту на екрані. Цей термін набув поширення після того, як код движка став незалежною від браузера програмою.

Найвідоміші:

– Trident – пропрієтарний двигун Microsoft Internet xplorer, використовується багатьма програмами для Microsoft Windows;

– Gecko – відкритий двигун проекту Mozilla, використовується у великій кількості програм, заснованих на коді Mozilla;

– KHTML – розроблений у рамках проекту KD , використовується у браузері Konqueror та послужив основою WebKit;

– WebKit – движок для браузера Apple Safari, включеного в операційну систему Mac OS X, і браузера Google Chrome;

– Presto – пропрієтарний двигун, розроблений Opera Software, є базою для браузера Opera.

Проблема XSS виникає внаслідок двох причин. По–перше, розроблені браузери не є безпечними. Вони були створені з метою реалізації обробки запитів, завдання браузера не входить виявлення шкідливих частин коду. По–друге, через відсутність навичок безпечного програмування та обмеження термінів розробки веб–додатків.

Проблема запобігання атакам на браузери пов'язана з тим, що існує безліч браузерів, і на виході веб–програми можуть по–різному інтерпретуватися. При цьому більшість браузерів дозволяють налаштувати виконання будь–яких клієнтських сценаріїв фільтрації або дозволяти сценарії лише з довірених хостів, доменів або зон.

На додаток до движка відображення веб–сторінок у браузері використовується інтерпретатор JavaScript, що є незалежною вбудованою програмою. Він дозволяє виконувати JavaScript програми та має можливість вбудовування в інші додатки, що представляють робоче оточення JavaScript. Браузери, засновані на KH ML: двигун Safari виділено з Konqueror у 2002 році. Konqueror в даний час складається з приблизно 200 000 рядків коду C++, в той час як в Safari складається з близько 350000 рядків коду C++. В обох браузерах кожен кадр у документі обробляється сортувальником H ML і движком відображення, який у свою чергу викликає інтерпретатор JavaScript. Цей інтерфейс є двонаправленим. Рух H ML викликає інтерпретатор JavaScript для виконання скриптів, з якими вона стикається при сортуванні документа, а

функція JavaScript може змінювати дерево документа, яке знаходиться у веденні двигуна H ML.

Браузер Opera підтримує так званий User JavaScript, призначений для автоматичного налаштування веб–сторінок довільних сайтів. Наприклад, якщо сайт розроблений для нестандартної поведінки Internet xplorer, User JavaScript може бути викликаний для перезапису вмісту під правильне представлення в Opera.

Програмний інтерфейс дозволяє реєструвати функції зворотного виклику JavaScript для обробки подій, що відбуваються в процесі аналізу та відображення. User JavaScript виконується до запуску сценаріїв на сторінці, що може дозволити запобігти виконанню будь–якого зі сценаріїв.

Додаткові проблеми виникають під час використання у веб–застосунках сторонніх бібліотек JavaScript. Як правило, розробники позначають дані бібліотеки як довірені, незважаючи на їх розширений функціонал. Так, графічний редактор у браузері дозволяє інтегрувати сторонній код, який перевершує його основні функції. А обмеживши доступ до стороннього коду, розробник також обмежує функціонал редактора.

Сучасні веб–програми надзвичайно різноманітні. Якість перевірок даних на стороні сервера залежить від обробки браузером клієнта даних отриманих від сервера. Внаслідок різноманітності браузерів виникають відмінності у обробці даних, що дозволяє уникнути виявлення атаки. Для правильної роботи багато хто використовує особливості браузерів, таких як Internet Explorer 6, що не відповідають стандартам W3C. Деякі програмні функції, які забезпечують функціональність для браузера, наприклад ActiveX, Java, різні сценарії (VBScript і т.д.), можуть також створювати вразливості у комп'ютерній системі [19]. Фахівці з безпеки також виділяють проблему атаки міжсайтового виконання сценаріїв із використанням небезпечних методів. Дослідження показали, що впровадження потенційно небезпечного коду ActionScript з 200 SWF файлів, які Google дав як результат пошукового запиту

"filetype: SWF inurl: click ag", схильні до 120. Історія браузера – це глобальний список сторінок, які були відвідані за допомогою вкладок браузера. При натисканні кнопок "Назад" та "Вперед" у браузері користувач переходить через всю історію браузера. Якщо сторінка містить IFRAM , будь–яка зміна місця всередині IFRAM також записується історія браузера. Отже, при відкритті однієї URL–адреси кілька разів вставиться лише один запис у списку Історії. Правило обмеження домену запобігає доступу до URL лише в історії об'єкта. Однак це правило не заважає доступу до history.length, який містить загальну кількість елементів у списку історії.

Розробники браузерів повинні розпочинати побудову захисту з формалізації підходу до захисту та реалізації обмеження завантаження вмісту сайтів. Реальність така, що марно чекати зменшення кількості атак XSS на веб–додатки, не кажучи вже про повне зникнення вразливості. Дослідження показують, що сучасні засоби захисту браузерів Opera, Chrome, Firefox, навіть у поєднанні із вбудованими політиками безпеки, вразливі до атак XSS. Internet Explorer версії нижче 10 також вразливий до атак XSS.

2.4 Аналіз існуючих рішень на ринку

Найчастіше об'єктом атаки XSS є браузер. Потенційно вразливі усі додатки, які не виконують JavaScript фільтрації на вході або виході. В результаті JavaScript дозволяє отримати доступ до H ML документа, що відображається в браузері користувача, а також дозволяє впливати на обробку даних, що передаються браузеру. XSS атака використовує той факт, що параметри не перевіряються при передачі з веб сайтів, що динамічно генеруються. Це дозволяє вставляти шкідливий код замість очікуваного значення параметрів. У браузерах на мережах великих компаній, як правило, дозволено запускати активний вміст, через свою функціональність веб–сайти не можуть коректно працювати без нього. Для зниження ризиків від атак пропонується розглянути питання фізичного відділення від Інтернету з цих

комп'ютерів . Поки поточна політика систем безпеки HTML недостатня для сучасних веб–додатків, у тому числі через їх проблеми з продуктивністю та недостатніми вимогами побудови сайтів, веб–додатки будуть наділені безліччю недокументованих функцій. Дослідження повинні оцінювати політику безпеки HTML, покращуючи заходи захисту, що вживаються.

Автори більшості робіт дотримуються думки, що джерелом уразливості для атак XSS є програмні продукти чи сайти, які здійснюють перевірку та очищення даних, які вводить користувач. Внаслідок чого вразливість вивчається з точки зору безпеки веб–додатку, а не безпеки об'єкта атаки. Класифікація атаки міжсайтового виконання сценаріїв наведена у статті Galan, вона поділяє всі атаки на 8 напрямків. Reis описує три фактори маркування стовпів безпеки браузера, які на його думку включаються ступінь небезпеки загроз, популярність вразливості і частота впливу. Wenliang Du вважає, що основна причина XSS – рівень деталізації сесії користувача. Інтернет стає дедалі складнішим, і довіряти вмісту за клієнта більше неможливо, як і запитам, ініціював доступом до цього вмісту. Таким чином, даючи різні запити з одними привілеями протягом однієї сесії неможливо задовольнити потребу в захисті інформації. Щоб не примушувати розробників додатків нести повну відповідальність за реалізацію цих потреб у захисті, необхідно створювати серверні системи контролю доступу. Крім того, дані часу відповіді сайту можуть стати свідченням витоку приватної інформації, припускаючи, що підключення стороннього каналу зловмисника дає додаткову затримку. Бортз представляє ряд прямих і непрямих методів вимірювання, які можуть ефективно виявляти реальні витоку приватної інформації, у тому числі при атаках XSS, дозволяючи виявити особисті дані користувача. У статті Patil та Bamnote, наводячи приклади захисту від XSS, вказує основною причиною атаки недостатність перевірки вхідних даних. «Ми можемо врятувати наші додатки від міжсайтового виконання сценаріїв, лише відкинувши

припущення, що дані, що вводяться, є довіреними. Сервер не повинен довіряти стороні клієнта, всі дані перед обробкою необхідно перевіряти».[18]

Для захисту від міжсайтового виконання сценаріїв розроблено безліч різних методик, проте найефективнішою залишається рекомендація Центру реагування на комп'ютерні інциденти, що наказує для захисту від XSS атак відключення JavaSctipt в браузері. Найбільш простий спосіб захисту від XSS – застосування спеціальних функцій. Наприклад, мова програмування PHP дозволяє використовувати функцію htmlspecialchars(), що перетворює спеціальні символи в H ML–сутності.

Практичне застосування функції захисту від XSS атак показує значне підвищення рівня безпеки. На жаль, ця функція працює з малою кількістю кодувань, що дозволяє легко обійти такий захист. Mario Heiderich у своїй роботі зазначає, що зниження ризику атаки XSS не обов'язково пов'язане із закриттям уразливості, достатньо щоб він перешкоджав виконання належним чином шкідливої частини коду. Це можна здійснити за допомогою фільтрації вхідних даних на проксі, а також розмітки (токенізації) елементів H ML. З цією метою на проксі аналізуються H P трафік обміну між браузером користувача та сервером. По–перше, запити клієнта перевіряються на наявність спеціальних HML символів (наприклад, символ "<"). У разі виявлення застосовується відображення цих імовірно шкідливих параметрів запиту користувача. Або символи кодуються перед відповіддю та доставляються до браузера користувача. Тобто будь–який символ введення користувача, який, як правило, має особливе значення, змінюється на текст, який передає в браузер інструкцію для відображення символів, а не інтерпретує структури сторінки. За таким принципом працюють BrowserShield і CoreScript, пропонуючи для захисту від JavaScript атак переписувати перехоплені скрипти відповідно до політики безпеки перед їх виконанням у браузері.

У BrowserShield процес перезапису включає довірені JavaScript функції, будучи посередником доступу до дерева документа для ненадійних скриптів.

CoreScript політики задаються за допомогою автозамін. Ще один інструмент аналізу перевірки вхідних даних на основі проксі – Stranger дозволяє виявляти атаки з використанням різних кодувань, у тому числі в автоматичному режимі. IPAAS є засобом розпізнавання типів даних вхідних параметрів, а також включає додаткову перевірку цих даних для підвищення безпеки веб–додатків. Дослідження, проведене сканером IPAAS, дозволило запобігти 83% SQL ін'єкцій та 65% XSS атак. Схожий метод пропонує Florian Kerschbaum, описуючи рішення, яке запобігає відображеному міжсайтовому скриптингу та підробку запитів атак за допомогою шлюзу на сервері.

Подібну реалізацію рекомендує Prateek Saxena, проте замість проксі пропонується використовувати клієнт хмарних обчислень. Таке рішення пов'язане з тим, що даний клієнт розроблений без використання сторонніх бібліотек та забезпечує безпеку виконання коду. Ця методика спирається на мову розмітки підтвердження безпеки, що базується на мові XML стандарт, розроблений для обміну даними про автентифікацію та авторизацію між захищеними доменами. Однією з важливих проблем, яку намагається вирішити SAML, є забезпечення наскрізної автентифікації за допомогою технології єдиного входу при роботі через Web–браузер. Alessandro Armado у своїй статті показує можливість застосування цієї технології для захисту від міжсайтового виконання сценаріїв. На думку Matthew A. Smith важливо розуміти, коли дані, що вводяться, повинні бути відфільтровані. Для максимальної безпеки необхідно мати фільтрацію на стороні сервера перед збереженням введення та мати можливість очищення даних перед відображенням. Це має бути зроблено на стороні сервера, оскільки будь–яким клієнтським сценарієм можна маніпулювати, оскільки він повністю доступний клієнтській стороні [20].

Обмеженням даного підходу і те, що він орієнтований передачу даних користувачем і дозволяє виявити стійких XSS, що зберігаються на сервер.

Крім того, деякі сервери помилково визначаються як вразливі. Ця проблема виникає внаслідок поступової, а не комплексної обробки.

У своїй науковій роботі Bojan Simic пропонує використання програми XSS Validator, яка перевіряє за списком дозволених дій отримані вхідні дані. Однак такий метод вимагає постійного оновлення списку та не дозволить виявити нові вектори атак. XSS атаки по структурі прості, проте їх важко запобігти через високу гнучкість мови H ML, а використання різних кодувань забезпечує зловмиснику додаткову можливість обходу фільтрації з боку сервера. Shalini для захисту від XSS пропонує ідентифікувати всі поля введення користувача в додатку і забезпечити обробку коду цих полів. Для цієї мети необхідно для кожного поля прописати кодування, символи, довжину рядка. Таке рішення дозволить запобігти більшій частині атак.

Рішення, запропоноване Zhendong Su, забезпечує перевірку виконання команд SQL–ін'єкції під час виконання запиту, і такий підхід дозволяє також запобігти атакі XSS. Є досить багато рішень, що пропонуються за тією ж схемою досліджень, які перевіряють виконання команд під час виконання запиту. Masri та Podgurski стверджують, що потік оброблюваної інформації, що підвищується, підвищить число помилкових спрацьовувань, наведений спосіб ефективний, якщо потік інформації великий. Для фільтрації даних, що вводяться користувачем, можуть використовуватися системи виявлення і запобігання вторгнень. Даний метод заснований на сигнатурному аналізі та дозволяє захиститися тільки від відомих загроз. Розроблені нині системи можуть виявляти факт застосування зловмисного коду, виробляючи семантичний аналіз, і виявляти тип атаки, і навіть блокувати шкідливе включення. Стаття Yacin Nadji описує методику, що дозволяє запобігати зміні структури довіреного коду, отриманого з ненадійних джерел. Така DSI система забезпечує ізоляцію вбудованих даних користувача, дозволяючи блокувати 98% відомих XSS атак. Також існує гібридне рішення щодо виявлення вразливостей на веб–ресурсах, що поєднує статичний та

динамічний аналіз для виявлення вразливостей на стороні клієнта. У ході проведення дослідження програма змогла виявити 2940 проблем Javascript на 188 із 675 сайтів. Проте 10% повідомлень виявилися помилковими.

Для виконання фільтрації даних, що вводяться, будь то параметри введення або H P заголовки, необхідно включати додаткові фільтри H ML тегів. Тим не менш, на стороні клієнта остаточного рішення для захисту немає, у I фільтр регулярних виразів легко обійти. Крім того, фільтр підданий критиці в реалізації самого фільтра, застосування якого призводило до атаки XSS. Код XSSAuditor і Chrome були об'єднані в Webkit у 2010 року, пізніше XSSAuditor був відключений за умовчанням через проблеми сумісності. Більш того, він захищає лише від XSS атак, які вводять повністю автономні сценарії. NoScript є найякіснішим захистом, проте потребує занадто багато знань від середнього користувача браузера. Фільтри, що блокують атаки, в основному відстежують жарту http запитів і http відповідей, які згенерував запит. Побудову фільтра, що містить усі негативні запити, створити неможливо, оскільки необхідно враховувати всі можливі вектори атаки та методи кодування тексту.

Phung запропонував дещо інший спосіб захисту. Суть методу самоконтролю JavaScript шаром, заснованим на полягає в мета–програмування спеціалізованими функціями JavaScript.[23] Такий підхід пропонує перехоплення та подальше повторне відтворення для отримання та встановлення доступу до об'єктної моделі документа. При цьому використовуються методи, схожі на аспектно–орієнтоване програмування. Публікацію оцінки надійності та захищеності від несанкціонованого доступу вищезазначеної обгорткової методики було випущено Jonas Magazinius, в ній обговорюється застосування реальних атак та обхід запропонованого механізму захисту [24].

Вирішенням проблеми XSS черв'яків може стати перехоплення H P запиту на стороні клієнта та його порівняння з сценаріями на сервері, що використовуються в даний час. Детальне рішення наведено в роботі, що

описує поведінку XSS черв'яка, що відстежується за допомогою вимірювання створюваного ним навантаження. Yinzhi Cao пропонує інший підхід, блокуючи два основні шляхи розповсюдження JavaScript черв'яків: доступ до об'єктів DOM та несанкціоновані запити на H P сервер.

Успішні випробування рішення були проведені на п'яти реальних хробаках та двох концептах.

Для перевірки можливості виконання атаки можуть використовуватися як програмні продукти за клієнта, звані сканери безпеки, і серверні рішення з урахуванням Java додатків. Дані продукти дозволяють автоматизувати створення потенційно зловмисних запитів, показати кількість успішно проведених атак та провести цілеспрямований аналіз із використанням різних векторів атак. Відомі успішні випробування програм, що дозволяють автоматизувати пошук вразливих полів, у тому числі для атак XSS на базі системи WebGoat. Однак такі продукти не враховують кінцеву успішність атаки, що проводиться, при використанні певних браузерів. За оцінками WSAC 80–96% веб–додатків має вразливість високого рівня ризику, їх 13% може бути скомпрометовано засобами автоматичного пошуку вразливостей. Велика кількість вразливостей веб–застосунків робить тестування більш продуктивним рішенням у виявленні проблем безпеки, на відміну від формальної перевірки моделі. Крім того, існуючі сканери безпеки не в змозі виявляти багатоступінчасті атаки, що зберігаються XSS, так як сканерам не вистачає розуміння взаємодії атаки між різними сторінками одного сайту. До того ж сканер не може відрізнити перенаправлення на іншу сторінку від атаки.

У комплексному дослідженні доктора Jayamsakthi Shanmugam, присвяченому захисту від XSS атак, зазначається, що в даний час існують мільярди веб–сторінок, які були розроблені в різних мовах, таких як PHP, ASP, JSP, H ML, CGI–P RL. N і т.д. Для них не розроблено єдиного доступного рішення, яке може бути застосоване для захисту веб–програм від міжсайтового виконання сценаріїв. Робота Joaquin Garcia–Alfaro показує

методи виявлення небезпечних даних. Пропонується не блокувати шкідливий запит, а попереджати користувача про його появу. І хоча така реалізація захисту підвищує рівень безпеки, більшість користувачів не обізнані про можливі ризики, і, ймовірно, приймуть дані запитані браузером. При статичному виявленні XSS проводиться аналіз вихідного програмного коду. Однак, даний метод є недостатнім, що доводиться успішним проведенням атак на виправлену програму [23].

Причиною недостатньої статичної перевірки є динамічне генерування веб–сторінок, що завантажуються, код HTML створюється динамічно. Хоча DOM–based XSS є серйозною проблемою безпеки, в даний час відсутнє програмне забезпечення, що підтримує динамічну оцінку коду, що завантажується користувачеві.

2.5 Висновки за розділом

У даному розділі було проведено аналіз виконання JavaScript. Також проаналізовано фактори атаки та її вплив на інформаційну систему. Ще проведено аналіз існуючих рішень на ринку. Проаналізовано інформаційні впливи при атаці на WEB–додатки. Виявлено фактори успішного розвитку атаки міжсайтинговигу скриптингу.

3 АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ СЕРЕДОВИЩА МОДЕЛЮВАННЯ ДЛЯ ВИВЧЕННЯ АТАКИ МІЖСАЙТОВЕ ВИКОНАННЯ СЦЕНАРІЇВ

3.1 Розробка моделі зловмисника, що виконує атаку міжсайтового виконання сценаріїв

При моделюванні проведення атак міжсайтове виконання сценаріїв на браузер користувача для виключення впливу фільтрів на стороні сервера використовується схема, подібна до стійкого типу атаки. При реалізації такої схеми зловмисник впроваджує шкідливий сценарій через активні форми на сайт, що атакується. Шкідливий сценарій у такому разі стає частиною даних веб–програми. У такому випадку користувач, який відкриває сторінку із впровадженим сценарієм, завантажує не тільки код веб–програми, але й шкідливий сценарій, який передає довільні дані атакувальному. Схема атаки показано на рис 3.1.

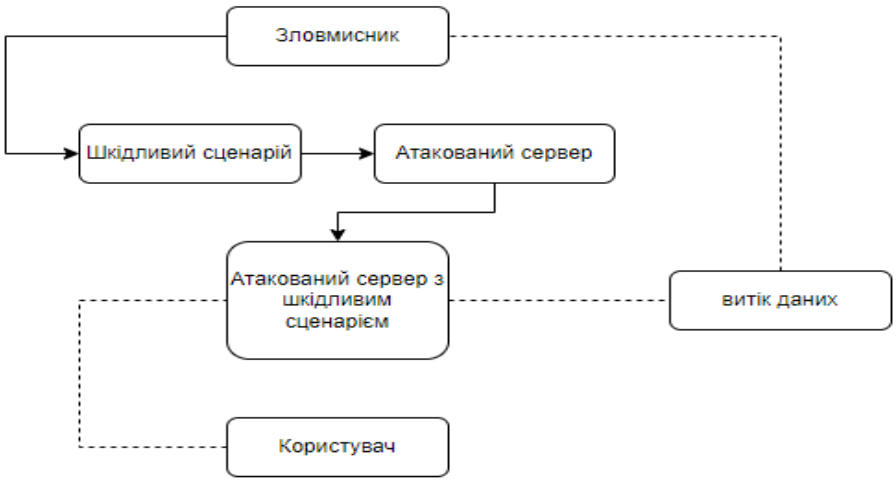


Рисунок 3.1 – Схема стійкого типу міжсайтового виконання сценаріїв

Схема, зазначена вище, є узагальненою, і для подальшого розуміння необхідно доповнити схему елементами, що дозволяють предметно описати реалізацію атак. Для цього введемо у схему засоби програмного забезпечення, що дозволяють обробляти код, що пересилається. Таким засобом є обробники коду. Схема атаки показано на рис. 3.2.

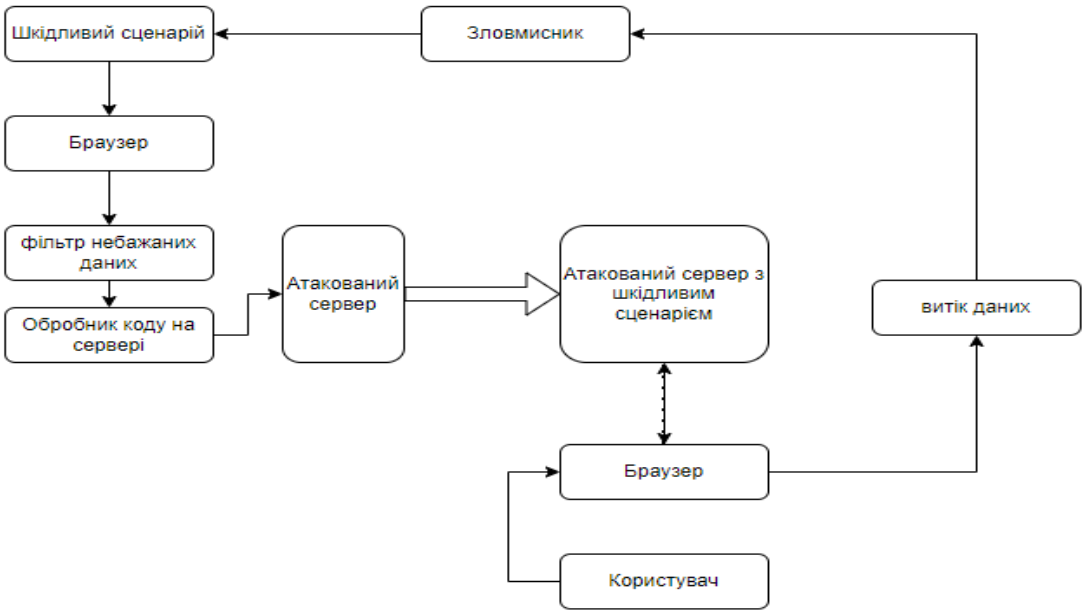


Рисунок 3.2 – Розширена схема стійкого типу міжсайтового виконання сценаріїв

При реалізації вектора атаки зловмисник передає шкідливий сценарій за допомогою браузера – програмного забезпечення для перегляду, обробки та виведення веб–сторінок.

Сценарій, що передається, обробляється на стороні сервера фільтром даних, що надходять, і обробником коду на сервері. Різні реалізації фільтрів – використання зовнішнього фільтра на проксі–сервері чи внутрішнього – з допомогою можливостей сервера, є принциповим і порушують працездатності запропонованої схеми. За відсутності небажаних для сервера даних сценарій передається на зберігання на сервері, стаючи атакованим сервером зі

шкідливим сценарієм. Далі користувач за допомогою браузера обробляє дані з атакованого сервера. За наявності шкідливого коду виконується без обмежень, т.к. браузер у загальноприйнятому поданні вважається засобом правильної інтерпретації коду та не містить фільтрів небажаних даних. При успішній обробці браузером шкідливого сценарію дані передаються зловмиснику.

Прийнято вважати, що атака міжсайтового виконання сценаріїв є наслідком відсутності фільтрації даних, що вводяться на стороні сервера. З цієї причини відповідальність за міжсайтове виконання сценаріїв перекладається на виробників сайтів, а більшість безпекових рішень носить серверний характер. Частина схеми, де реалізуються механізми безпеки, показано на рис. 3.3.

Рисунок 3.3 – Схема реалізації механізмів захисту за сервера

При цьому реалізація механізмів безпеки веб–застосунків не є обов'язковою для розробників, що не дозволяє кінцевому користувачеві захиститися від атакованих сайтів.

3.2 Розробка класифікації векторів міжсайтового виконання сценаріїв

Атаки є векторами при виконанні яких вразливий компонент може містити шкідливий довільний код. Однак семантичний аналіз застосовуваних векторів дозволяє розробити класифікацію, на підставі якої всі вектори можна віднести до кількох груп, що мають загальними ознаками. Така класифікація дозволяє виявити найуспішнішу групу векторів. Для цього дослідження пропонується використовувати методику, що групує вектори за загальними ознаками. Під загальною ознакою в даній класифікації розумітиметься місцезнаходження вразливості, показано на рисунку 3.4.

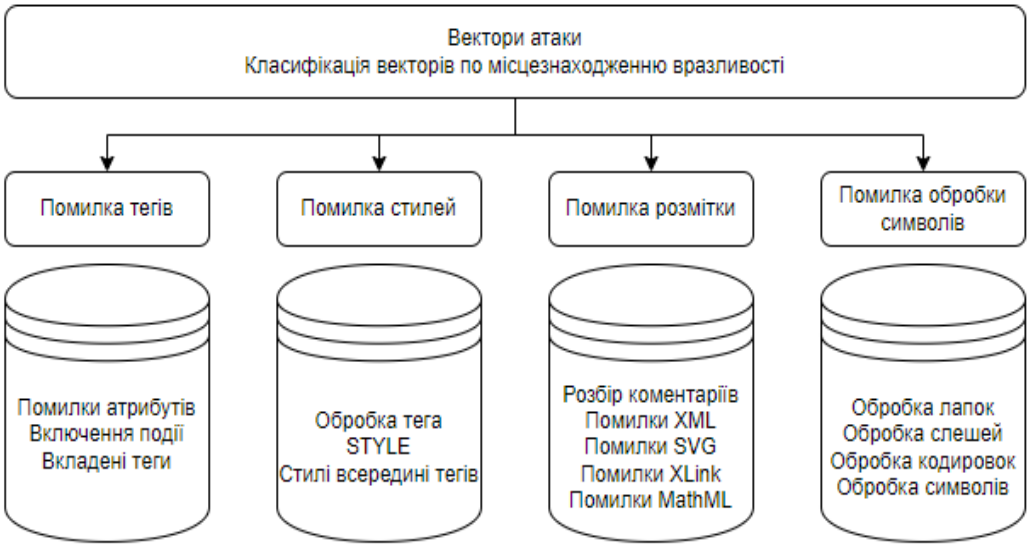


Рисунок 3.4 – Вектори Атаки

До першої групи атак належать вектори, які використовують помилки обробки HTML тегів. Групу можна умовно розділити на помилкове виконання сценаріїв усередині атрибутів, підключення сценаріїв за допомогою подій, а також виклик Javascript за допомогою вкладення тегів один в одного. До виконання сценаріїв всередині атрибутів відносяться вектори, що дозволяють параметром атрибута задати JavaScript, що успішно виконується при неправильній інтерпретації отриманих браузером даних. Підключення

сценаріїв за допомогою подій здійснюють вектори, що дозволяють внаслідок помилок обробки вмісту тега або навмисного виконання дії користувачем викликати подію, що містить код JavaScript. Виклик JavaScript за допомогою вкладення тегів передбачає закриття одних тегів відкриттям інших, при цьому теги залишаються функціонуючими.

До другої групи атак належать вектори, які використовують помилки обробки стилів, які допускають вихід сценарію за межі селектора. Групу можна умовно розділити на атаки за допомогою тега STYLE та атаки з додаванням стильової інформації до інших тегів. Тег STYLE використовується для впровадження CSS коду безпосередньо в HML документ, таким чином, формуючи внутрішню таблицю стилів. Так як всередині тега міститься інформація, призначена лише для браузерів, а чи не для користувачів. Атрибут використовується для стилізації елементів безпосередньо в коді. Мета атрибуту style полягає у наданні способу використання CSS стилів у будь–якому HML елементі.

До третьої групи атак належать вектори, які застосовують помилки обробки розмітки. Дана група представлена векторами атак, що використовують коментування вмісту, а також підключають можливості інших мов розмітки, що підтримуються браузерами. До цієї групи віднесено атаки, що використовують мову розмітки XML, що розширюється. Мова називається розширюваною, оскільки вона не фіксує розмітку, що використовується в документах: розробник вільний створити розмітку відповідно до потреб конкретної області, будучи обмеженим лише синтаксичними правилами мови. Крім того, до цієї групи можна віднести атаки за допомогою мов, що належать до підмножини XML.

Мова розмітки масштабованої векторної графіки SVG, призначена для опису двовимірної графіки у форматі XML. Мовою підтримуються скриптові мови на основі специфікації CMAScript. SVG–елементами можна керувати з допомогою JavaScript. Мова розмітки, що дозволяє вставляти в XML

документи елементи для опису посилань між ресурсами XLink. Він використовує XML синтаксис для створення структури складних посилань HML. Мова математичної розмітки є застосуванням XML для представлення математичних символів і формул у HTML документах MathML.

До четвертої групи атак належать вектори, які використовують помилки обробки символів. Ця група представлена векторами атак, що використовують слеши, лапки, а також обробку різних кодувань та символів. Вектори, використовуючи незбалансовані лапки і слеши.

В результаті класифікація дозволяє виявити найбільш успішний напрямок векторів атак і виявити можливі причини, що призводять до їх виконання. Ця методика рекомендується до використання при подальших класифікаціях атак, оскільки її використання не знижує можливості зміни векторів і одночасно дозволяє класифікувати їх відповідно до загальних ознак.

3.3 Висновки за розділом

У даному розділі проаналізовано модель зловмисника. Розроблено класифікацію векторів міжсайтових атак. Та обрано середовище розробки програмного додатку для пошуку XSS–вразливостей. Розписано схему класифікації векторів міжсайтових.

4 РЕАЛІЗАЦІЯ ТА РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ

4.1 Середовище розробки для пошуку XSS–вразливостей у WEB–додатках

У даній роботі для розробки було обрано систему Linux Ubuntu 20.04. У випадку з Windows запроси які використовуються у пошуку XSS–вразливостей у WEB–додатках викликають конфлікти та не працюють коректно. Про те Linux дає більше можливостей для розробки та ініціалізації програми.

Операційна система Windows потребує багато бібліотек та частково викликає конфлікти при використанні розробки. Linux вже має непоганий набір вбудованих бібліотек та добре працює з веб–тестингом різного роду.

Переваги запуску Python у браузері

Хоча JavaScript є повсякденною мовою інтерфейсної веб–розробки, є моменти коли потрібна інша мова програмування яка буде виконувати функції неможливі для реалізації у даній мові програмування.

Якою б не була причина, багато розробників віддали перевагу альтернативі JavaScript на основі Python для використання можливостей браузера. Запуск Python у браузері дає кілька переваг, що дозволяє:

– виконувати один і той же код Python на сервері та в браузері

– працювати з різними API–інтерфейсами браузера за допомогою Python

– керувати об'єктною моделлю документа (DOM) за допомогою Python

– використовувати Python для взаємодії з існуючими бібліотеками JavaScript, такими як Vue.js та jQuery.

Одним із побічних ефектів використання Python у браузері є втрата продуктивності порівняно з тим же кодом JavaScript. Однак, цей недолік не переважує жодної з перерахованих вище переваг.

Реалізація ізоморфної веб–розробки.Ізоморфний JavaScript або Universal JavaScript підкреслює, що програми JavaScript повинні працювати як на клієнті, так і на сервері. Передбачається, що серверна частина ґрунтується на JavaScript, а саме на сервері Node. Розробники Python, які використовують Flask або Django, також можуть застосовувати принципи ізоморфізму до Python, за умови, що вони можуть запускати Python у браузері.

Brython дозволяє створювати інтерфейс на Python та обмінюватися модулями між клієнтом та сервером. Наприклад, ви можете поділитися функціями перевірки, такими як наступний код, який нормалізує та перевіряє номери телефонів у США:

```
import re
def normalize_us_phone(phone: str) –> str:
""" Виймати номери та цифри із заданого номера телефону """
return re.sub(r"[^\da–zA–z]", "", phone)
def is_valid_us_phone(phone: str) –> bool:
""" Підтвердити 10–значний номер телефону """
normalized_number = normalize_us_phone(phone)
return re.match(r"^\d{10}$", normalized_number) is not None
```

normalize_us_phone() видаляє будь–які буквенно–цифрові символи, тоді як is_valid_us_phone() повертає True, якщо вхідний рядок містить рівно десять цифр і не містить буквених символів. Один і той же код може використовуватися спільно процесами, запущеними на сервері Python, та клієнтом, створеним за допомогою Brython.

Інтернет–браузери надають стандартизовані Веб–API для JavaScript. Ці стандарти є частиною стандарту HTML життя. Brython дозволяє не тільки використовувати веб–API, але й взаємодіяти з JavaScript.

4.2 Розробка та тестування програмного додатку

Для розробки програми обрана операційна система Linux Ubuntu Mint 21 та мову програмування Python та доповнення Brython.

Python достатньо популярна мова програмування та може реалізувати необхідні мені функції для пошуку WEB–вразливостей. Також я використовував доповнення Brython для інтеграції моїх запитів з програмним додатком, щоб усе проходило в автоматичному режимі.

У розробці використовувалася система Linux дистрибутив Mint. Обрана саме ця система за її легке використування та простоту настройки. Авжеж найбільш розвинена система для тестування WEB–додатків Kali Linux, але для розробленого програмного додатку підходить майже будь який дистрибутив Linux.

На перших етапах розробки розлоблено блок–схему програми та її основних функцій. У блок–схемі показано варіанти з WAF(WEB Application Firewall) у статусі онлайн та офлайн. Різні варіанти роботи програми у різних випадках. Блок схему показано на рисунку 4.1.

Програмний додаток поділено на 3 блоки. Перший блок це ядро програмного додатку яке складається с декількох бібліотек. Другий блок це дані необхідні для пошуку уразливостей наприклад генератор пайлоадів або пайлоад листи. Третій блок це сама програма яка компелює усі бібліотеки.

Спочатку на Python було додано бібліотеку “fuzzer.py”, вона дозволяє порівнювати строки або записи та відноситься до першого блоку програми. Ця бібліотека дозволяє порівнювати відповіді на запити, та виявлити ступень вразливості.

Фрагмент бібліотеки fuzzer.py:

```
import copy
from random import randint
from time import sleep
from urllib.parse import unquote

from core.colors import end, red, green, yellow
from core.config import fuzzes, xsschecker
from core.requester import requester
from core.utils import replaceValue, counter
from core.log import setup_logger

logger = setup_logger(__name__)
```

Далі розроблено програмний код та створено файл “WAFdetector.py” який відноситься до першого блоку, цей код перевіряє наявність WEB Application Firewall.

У коді використовується також ще язик програмування JavaScript. Було утворено файл з сигнатурами для перевірки WAF з назвою “WafSignatures.json”, який відноситься до другого блоку.

WAF (Web Application Firewall) – міжмережевий екран для веб–застосунків. Це інструмент для фільтрації трафіку, що працює на прикладному рівні та захищає веб–додатки методом аналізу трафіку HTTP/HTTPS та семантики XML/SOAP. WAF може встановлюватися на фізичний або віртуальний сервер та виявляє найрізноманітніші види атак.

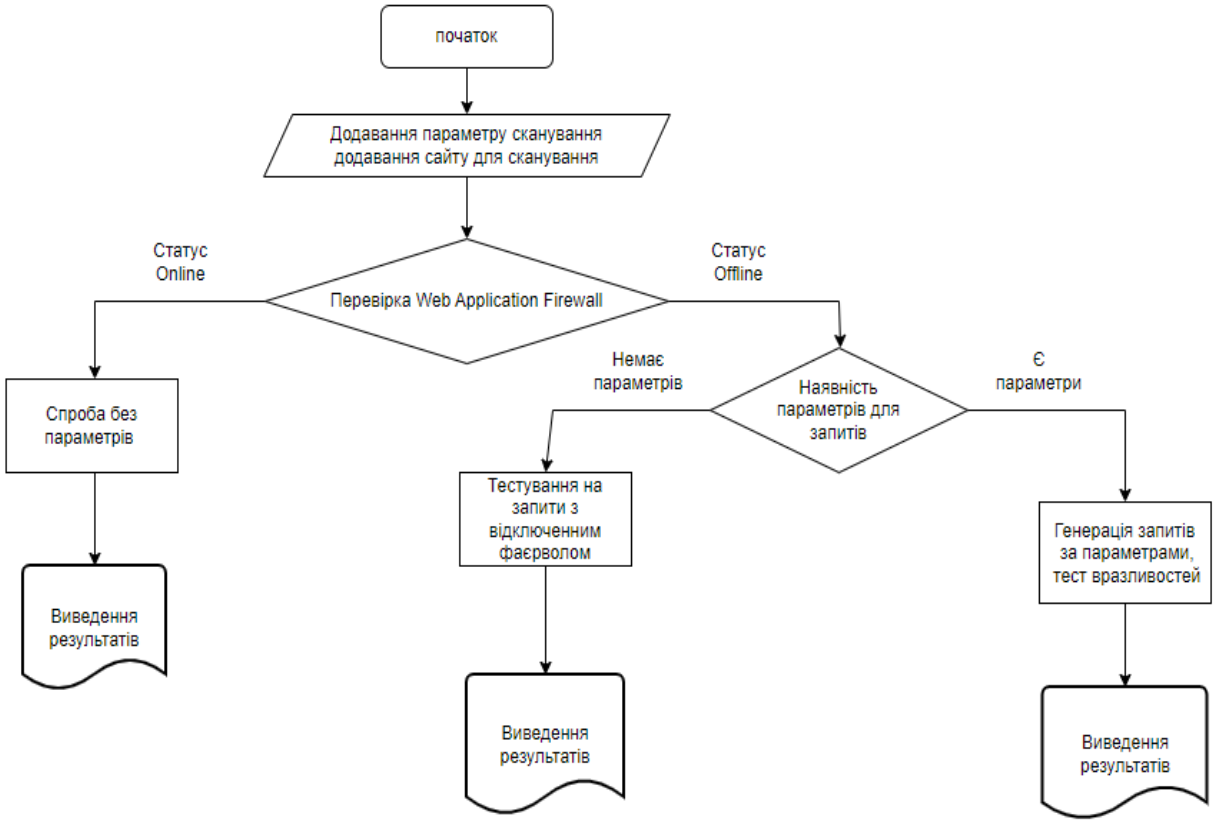


Рис. 4.1 Блок–схема програмного додатку

Наступним етапом було розроблено блоки програмного додатку згідно блок схеми. Утворено три блоки, з яких :

– перший блок відповідає за основні функції програмного додатку;
– другий блок відповідає за допомогу першому блоку в перевірці та транслірування інформації;
– третій блок ядро програми за яким функціонує програмний додаток.

Класифікації блоків програмного додатку наведено на рисунку 4.2.

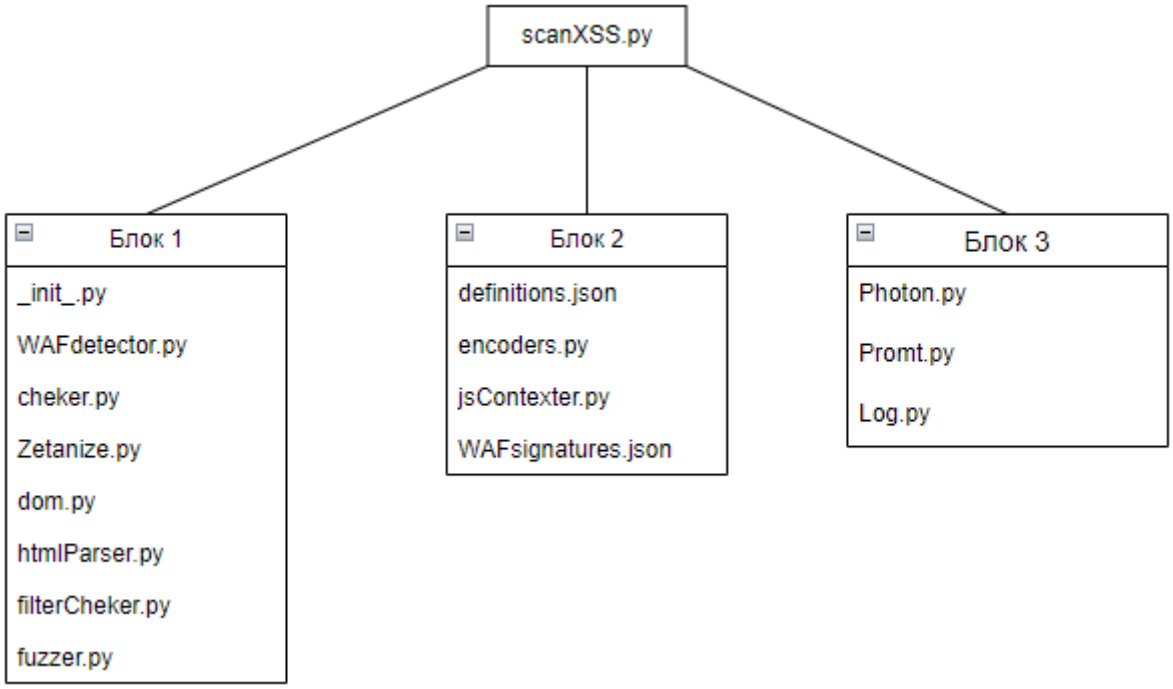


Рисунок 4.2 – Класифікація блоків програмного додатку

Результат роботи цього коду - це виведення одного рядка з інформацією статусу та підключення інших бібліотек у разі отримання значення Offline, показано на рисунку 4.3 і 4.4.

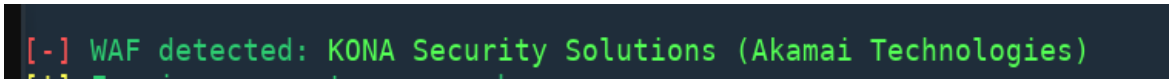


Рис. 4.3 – Результат статусу Online WAF

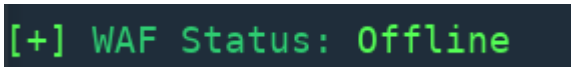


Рис. 4.4 – Результат статусу Offline WAF

Наступний програмний код відноситься до першого блоку та перевіряє наявність вразливості DOM (Document object model) та використовує її.

Фрагмент програмного коду “dom.py”:

```
def dom(response):
 highlighted = []
sources =
r'"\b(?:document\.(URL|documentURI|URLUnencoded|baseURI|cookie|referrer)|locatio
n\.(href|search|hash|pathname)|window\.name|history\.(pushState|replaceState)(local|ses
sion)Storage)\b"'
            sinks =
r'"\b(?:eval|evaluate|execCommand|assign|navigate|getResponseHeaderopen|showModal
Dialog|Function|set(Timeout|Interval|Immediate)|execScript|crypto.generateCRMFRequ
est|ScriptElement\.(src|text|textContent|innerText)|.*?\.onEventName|document\.(write|
writeln)|.*?\.innerHTML|Range\.createContextualFragment|(document|window)\.locatio
n)\b"'
```

Перевіряє параметри:

- document.URL
- document.URLUnencoded
- document.location (і безліч його властивостей)
- document.referrer
- window.location (і безліч його властивостей)
- висновок необробленого HTML
- document.write(), document.writeln()
- document.body.innerHTML=
- пряма модифікація DOM (включаючи події DHTML)
- document.forms[0].action= (та інші колекції)
- document.atatchEvent()
- document.create...()
- document.execCommand()

– document.body. ... (доступ до DOM через об'єкт body)

– window.atatchEvent()

– методи сторонніх бібліотек, наприклад jQuery: after(), append(), html(), wrap() тощо.

– document.location= (і присвоювання href, host, hostname)

– document.location.replace()

– document.location.assign()

– document.location.URL=

– window.navigate()

– document.open()

– window.open()

– window.location.href= (і привласнення host, hostname)

Наступним єтапом був розроблений програмний код для пошуку параметрів тестування відбитих вразливостей “cheker.py” який відноситься до першого блоку програмного додатку. За цими параметрами генеруються необхідні запити для перевірки на вразливості. Параметри наведені у таблиці 4.1.

Таблиця 4.1

n	Сканування без параметрів
p	Скануваня за шаблонами або готовими скриптами
s	Сканування з генерацією запитів
k	Неможливе сканування

Програмний код для пошуку параметрів тестування відбитих вразливостей “cheker.py”:

```
def checker(url, params, headers, GET, delay, payload, positions, timeout, encoding):
    checkString = 'st4r7s' + payload + '3nd'
    if encoding:
        checkString = encoding(unquote(checkString))
    response = requester(url, replaceValue(
```

третього блоку програми. У програмного коду запрограмовано декілька запитів які можуть змінюватися завдяки генератору запитів.

Генератор запитів використовує декілька бібліотек а також використовує базу створену на JavaScript. У код JS додано шаблони для генерації запитів а також бібліотеки відомих вразливостей за це відвовідає файл “definitions.json” який відноситься до другого блоку програми.

Фрагмент структури бази, створена на JavaScript:

```
        "jquery": {
                "bowername": [ "jQuery" ],
                "vulnerabilities" : [
                        {
                                "below" : "1.6.3",
                                "severity" : "medium",
                                "identifiers" : {
                                        "CVE": [ "CVE-2011-4969" ],
                                        "summary": "XSS with location.hash"
                                },
                                "info" : [ "https://nvd.nist.gov/vuln/detail/CVE-
2011-4969" , "http://research.insecurelabs.org/jquery/test/",
"https://bugs.jquery.com/ticket/9521" ]
```

Для функції перевірки посилань на вразливості та пошуку параметрів для вразливостей створено “htmlParser.py” який відноситься до другого блоку програми. Стандартний парсер сторінок HTML.

Далі додано дуже необхідну бібліотеку Photon.py який відноситься до третього блоку програмного додатку.

Фрагмент дуже необхідної бібліотеки Photon.py:

```
        checkedDOMs.append(clean_highlighted)
                logger.good('Potentially vulnerable objects found at %s' % url)
                logger.red_line(level='good')
                for line in highlighted:
```

– URL–адреси з параметрами
– файли (pdf, png, xml і т.п.)
– файли JavaScript
– субдомени, дані пов’язані з DNS

Наступним було додано “Promt.py” який відноситься до третього блоку програмного додатку, для більш гнучкого налаштовування програмного додатку.

Також для JS додано “jsContexter.py” та “encoders.py”, обидва файли відносятся до другого блоку програмного додатку. Вони допомагають адаптувати код JS під Python.

За допомогою “Zetanize.py” виконується передача запиту на код Python, бібліотека відноситься до першого блоку програмного додатку. Тобто він перекодує запити JS, HTML у код який взаємодіє з Python.

У процесі програмування потрібен був журнал помилок та недоліків для цього утворен файл log.py який відноситься до третьогу блоку. Він виводить дані виконаної роботи.

Основні функції програмного додатку:

– відбите та DOM XSS сканування;
– багатопотокове сканування;
– контекстний аналіз;
– конфігуроване ядро;
– виявлення та ухилення від WAF;
– сканування застарілої бібліотеки JS;
– інтелектуальний генератор корисного навантаження;
– саморобний парсер HTML та JavaScript;
– потужний фазинговий двигун;
– сліпа підтримка XSS;
– ретельно вивчений робочий процес;
– повна підтримка HTTP;

– перебір корисних даних із файлу;

– кодування корисного навантаження;

Для встановлення програми потребується дистрибутив Linux та такі бібліотеки як “pip” та ”fuzzywuzzy”.

Запуск програми виконується за допомогою терміналу Linux .

Наступні команди запускають програмний додаток за різними шляхами:

– python scanxss.py -u http://example.com (сканування одного URL)

– python xsstrike.py -u "http://example.com " –crawl (сканування за картою сайту)

– python xsstrike.py -u "http://example.com " -d 2 (сканування одного URL з затримкою секунд, для меншого навантаження серверів)

– python xsstrike.py -u "http://example.com " –fuzzer (перевірка фільтрів та браундмаєрів)

– python xsstrike.py -u "http://example.com " --skip-dom (пропуск сканування вразливості DOM, за замовчуванням перевіряється кожного разу)

– python xsstrike.py -u "http://example.com/search.php?q=query" --console-log-level DEBUG (вивід журналу роботи програми у файл, був необхідний при розробці програмного додатку)

Сканування на вразливість DOM–XSS сайту https://donntu.edu.ua показано на рисунку 4.5. На даному прикладі видно що знайдено частину коду сторінки яка може приймати команди script,print та інші, також можливо виявити файли cookies які є конфеденційні. Ця вразливість є небезпечною тому що може надати

Програмний додаток також наглядно показує код за яким знайдено вразливість, за цією інформацією є можливість власноруч найти вразливість за допомогою перевірки коду сторінки в будь якому браузері.

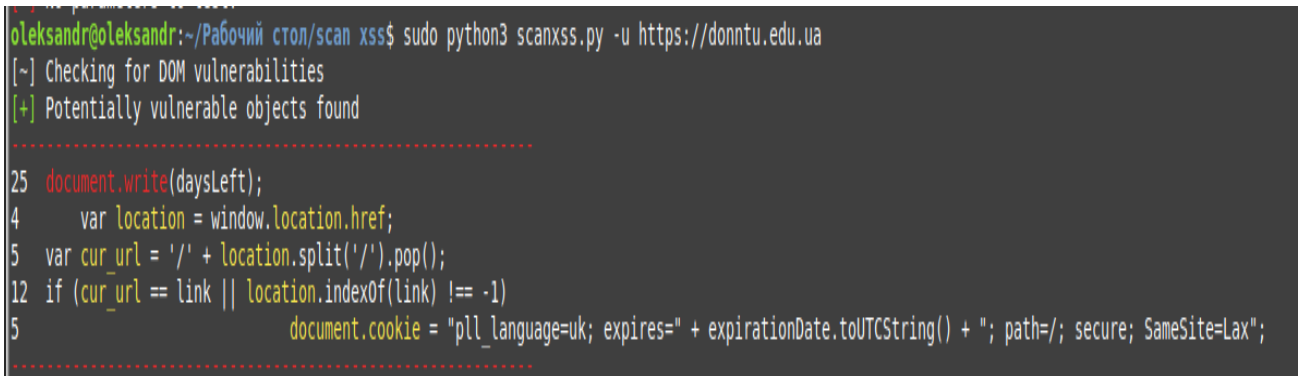


Рис. 4.5 – DOM–XSS сканування

Наступний тест відбите XSS. Программа знаходить параметри вразливості, генерує запити та перевіряє кожний на степінь вразливості, що й показано на рисунку 4.6. Цей параметр це частина коду яка дозволяє вводити дані, за цим параметром починається генерація запитів, вводиться туди та перевіряється відповідь на запит, якщо запит виконався то це вразливість за якою можливо отримати доступ до БД або конфеденційних даних.

Рис. 4.6 – Відбите XSS сканування

Далі є функція повзання, тобто перевірка посилань за картою сайту. Функція може бути дуже затратна за часом виконання але досконало перевіряє сайт на вразливості показано на рисунку 4.7. Затрати на час викликано тим що сайт може мати велику кількість сторінок, кожна сторінка перевіряється за

допомогою запитів по шаблону. З цього іноді виникає затрати часу тому що все перевіряється поступово.

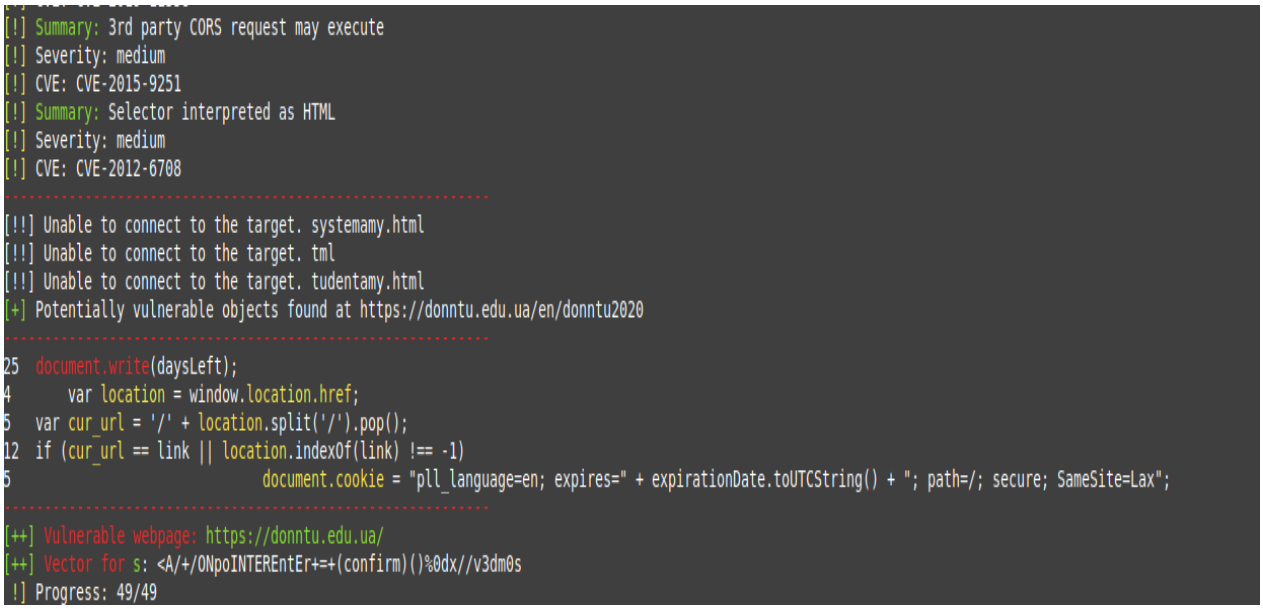


Рис. 4.7 – Функція “повзання” результат роботи

Наступна функція це фазинг. Фазинг – це техніка перевірки ПО. Функція перевіряє стандартні команди скриптингу за шаблоном для вразливості показано на рисунку 4.8.

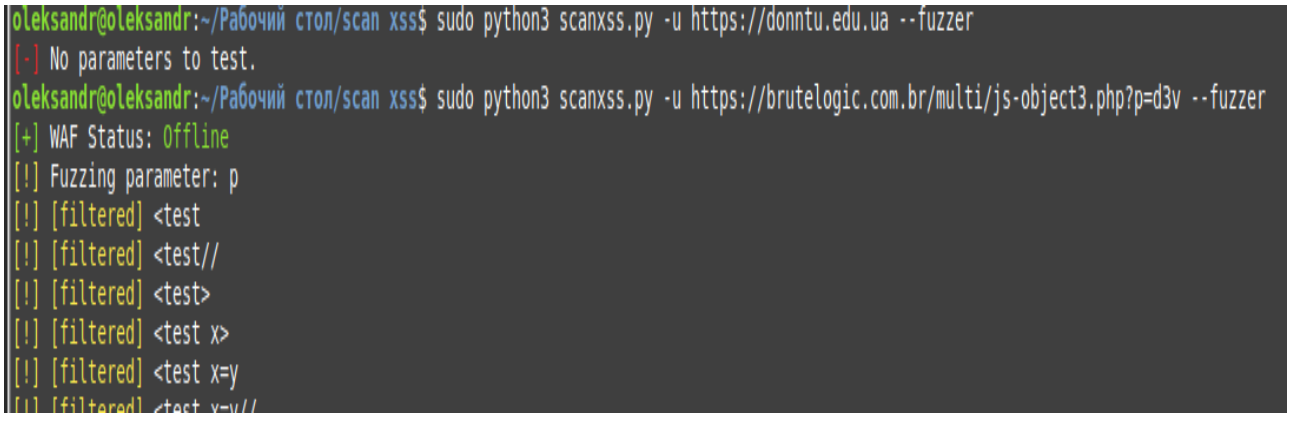


Рис.4.8 – Сканування за допомогою фазингу

За фазингом іноді знаходяться дуже серйозні вразливості, які можуть надавати доступ до БД.

4.3 Висновки за розділом

В даному розділі пояснювальної записки розроблено та протестовано програмний додаток. Описано середу програмування, мову програмування та параметри які може знаходити програмний додаток. Протестовано сканування за деякими параметрами. Також наведено приклади роботи програми за різними параметрами які наведені у таблиці 4.1. Протестовано функцію знаходження WAF та надано приклади роботи алгоритму. Програмний додаток досить інтелектуальний, вміє сам шукати вразливості та працювати з ними. Автоматично генерую запити та підбирає їх до вразливостей. Наглядно показує частину коду в якій знайшло уразливість.

ВИСНОВКИ

На час дипломування була поставлена задача дослідити методи пошуку, атаки та захисту на WEB–додатки. Досліджено WEB–додатки, відмінність WEB–додатків від веб–сайтів. Оглянуто можливі вразливості на WEB–додатки. Описана схема реалізацій атак на WEB–додатки. Проаналізовано процес виконання JS як середовище міжсайтингових атак. Оглянуто фактори успішного розвитку міжсайтингового скриптингу. Також проаналівано існуючі рішення на ринку.Створена модель зловмисника, що виконує міжсайтову атаку. Розроблено класифікація векторів міжсайтингового скриптування, програмний додаток для сканування WEB–додатків, описано функції, наведено приклади роботи та пройден етап тестування.

Оглянуті WEB–додатки та їх характеристики. Досліджено існуючі методи атаки на вразливості WEB–додатків. У процесі обрано технології та середовища розробки. Розроблено програму сканування WEB–додатків на вразливості, наведено блок схему програми, класифікацію блоків програми та протестовано основні функції.

Розроблено програмний додаток для розробників. Програмний додаток виконує такі функції:

– сканування на вразливості за заданою URL–адресою;
– сканування на вразливості за картою сайту;
– перевірка наявності WAF, та визначення до якої фірми він належить;
– генерує запити за шаблоном для вразливостей без параметрів;

Програмний додаток досить інтелектуальний та робить основну частину роботи самостійно достатньо вибрати яким шляхом сканувати WEB–додатки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Євтєєв Д. SQL Injection від А до Я // Positive Technologies [Електронний ресурс]. 11.10.2014. – Режим доступу: URL: www.ptsecurity.com/upload/corporate/ru–ua/analytics/PT–devteevAdvanced–SQL–Injection.pdf (дата звернення 09.10.2022).

2. КБлог Вільного Вебмайстра // Міжсайтовий скриптинг, як захистити сайт від атаки XSS [Електронний ресурс]. 29.01.2018 – URL: https://webliberty.ru/xss/ (дата звернення 12.10.2022).

3. Klein A. Третій тип XSS: Міжсайтовий скриптинг через DOM // SecurityLab.ru [Електронний ресурс]. 22.10.2022. – URL: https://www.securitylab.ru/analytics/275087.php (Дата звернення 18.03.2018).

4. Berners–Lee T. Uniform Resource Locators // RFC 1738 – IETF [Електронний ресурс] – URL: www.ietf.org/rfc/rfc1738.txt (дата звернення 23.09.2022).

5. Do Son. Clickjacking Attack // Penetration Testing Security Training Share [Електронний ресурс] 27.07.2017. – URL: https://securityonline.info/clickjacking–attack/(Дата звернення 24.10.2022).

6. Cross–frame–scripting // OWASP free and open software security community [Електронний ресурс]. 22.06.2017. – URL: www.owasp.org/index.php/Cross_Frame_Scripting (дата звернення 22.10.2022).

7. Типи XSS: Stored XSS, Reflected XSS і DOM–based XSS // Acunetix Blog [Електронний ресурс]. 17.01.2018. – URL: www.acunetix.com/websitesecurity/xss/ (дата звернення 17.09.2022).

8. DOM Based XSS // OWASP free and open software security com–munity [Електронний ресурс]. 22.06.2017. – URL: www.owasp.org/index.php/DOM_Based_XSS (Дата звернення 24.03.2018).

9. Fogie S., Grossman J., Hansen R., Rager A., Petkov P. XSS attacks: Cross Site Scripting exploits and defense – Seth Fogie, Oxford: Elsevier Limited, 2007. – 448 p.

10. CSP (Політика Захист Контенту) // MDN Web Docs [Електронний ресурс].21.12.2016. – URL: https://developer.mozilla.org/ru/docs/Web/Security/CSP (дата звернення 12.10.2022).

11. XSpider // Positive Technologies [Електронний ресурс]. 12.05.2013. – URL: www.ptsecurity.com/ua–ru/products/xspider/ (дата звернення 20.10.2022).

12. Nemesida Scanner // PENTESTIT [Електронний ресурс]. 03.01.2017. – URL: www.pentestit.ru/nemesida–scanner/ (дата звернення 20.10.2022).

13. Audit Your Web Security with Acunetix Vulnerability Scanner // Acunetix [Електронний ресурс]. 03.01.2017. – URL: www.acunetix.com/vulnerability–scanner/ (дата звернення 20.10.2022).

14. Джатана Н., Агравал А., Собті К. Пост–експлуатація XSS: просунуті методи та способи захисту // SecurityLab.ru [Електронний ресурс]. 12.05.2013. – URL: www.securitylab.ru/analytics/440187.php (дата звернення 20.10.2022).

15. Xenotix XSS Exploit Framework // OWASP [Електронний ресурс]. 22.06.2017. – URL: www.owasp.org/index.php/OWASP_Xenotix_XSS_Exploit_Framework (дата звернення 21.10.2022).

16. Fogie S., Grossman J., Hansen R., Rager A., Petkov P. D. XSS attacks: Cross Site Scripting exploits and defense. Elsevier Limited, Oxford, 2007. 448 c.

17. White A. JavaScript Programmer's Reference. Indiana, Wiley Publishing, Inc., 2009. P.1032.

18. Raman P. JaSPIn: JavaScript based anomaly detection of Cross–site scripting attacks // A thesis submitted in partial fulfillment of the requirements for the degree of Master of computer science, 2008. [Електронний ресурс]. URL:

http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.216.6860 (дата звернення 24.10.2022).

19. Dormann W., Rafail J. Securing your Web browser // CERT. 2006.URL: http://www.cert.org/tech_tips/securing_browser (дата звернення 22.10.2022).

20. ECMA Script Language. Specification Standard ECMA–262. 3rd Edition // Standardizing Information and Communication Systems. 1999. [Електронний ресурс]URL:http://www.ecma–inter national.org/publications/files/ECMA–ST–ARCH/ECMA262,%203rd%20edition,%20 December%201999.pdf (дата звернення 24.10.22).

21. Флэнаган Д. JavaScript. Подробное руководство, 5–е издание. Символ–Плюс, 2008. 992 с.

22. Li Y. Cross–Site–Scripting (XSS) Attacking and Defending // Bachelor´s thesis abstract. 2009. P. 1–73. [Електронний ресурс] URL: http://publications.theseus.fi/bitstream/handle/10024/1 3013/Li_ Yonghao.pdf?sequence=1(дата звернення 22.10.2022)

23. Phung P. H., Sands D., Chudnov A. Lightweight Self–Protecting JavaScript // Volume March 2009 of Computer and Communications Security. 2009. P. 47–60. URL: http://www.cse.chalmers.se/~dave/papers/ASIACCS09.pdf (дата звернення 24.10.2022)

24. Magazinius J., Phung P. H., Sands D. Safe wrappers and sane policies for self protecting JavaScript // Chalmers University of Technology. 2010. [Електронний ресурс] URL: http://www.cse.chalmers.se/~ dave/papers/SafeWrappers.pdf (дата звернення 22.10.2022).

Додаток А

Зауваження нормоконтролера

Таблиця А.1 – Зауваження нормоконтролера

Сторінка	Позначення	Зміст зауваження
ПЗ		Кожний розділ повинен починатися зі службового слова РОЗДІЛ
С. 16,…56		Фрагменти коду оформлюються рисунками

Додаток Б

Лістинг

Блок 1. Лістинг допоміжних бібліотек

“_init_.py” (стандартний файл, зазвичай пустий)
“WAFdetector”

```
import json
import re
import sys

from core.requester import requester
from core.log import setup_logger

logger = setup_logger(__name__)

def wafDetector(url, params, headers, GET, delay, timeout):
    with open(sys.path[0] + '/db/wafSignatures.json', 'r') as file:
        wafSignatures = json.load(file)
    noise = '<script>alert("XSS")</script>'
    params['xss'] = noise
    response = requester(url, params, headers, GET, delay, timeout)
    page = response.text
    code = str(response.status_code)
    headers = str(response.headers)
    logger.debug('Waf Detector code: {}'.format(code))
    logger.debug_json('Waf Detector headers:', response.headers)

    if int(code) >= 400:
        bestMatch = [0, None]
        for wafName, wafSignature in wafSignatures.items():
            score = 0
            pageSign = wafSignature['page']
            codeSign = wafSignature['code']
            headersSign = wafSignature['headers']
            if pageSign:
                if re.search(pageSign, page, re.I):
                    score += 1
            if codeSign:
                if re.search(codeSign, code, re.I):
                    score += 0.5
            if headersSign:
                if re.search(headersSign, headers, re.I):
                    score += 1
            if score > bestMatch[0]:
                del bestMatch[:]
                bestMatch.extend([score, wafName])
        if bestMatch[0] != 0:
            return bestMatch[1]
        else:
            return None
    else:
        return None
```

“cheker.py”

```
import copy
from fuzzywuzzy import fuzz
import re
from urllib.parse import unquote
```

```
from core.config import xsschecker
from core.requester import requester
from core.utils import replaceValue, fillHoles


def checker(url, params, headers, GET, delay, payload, positions, timeout, encoding):
    checkString = 'st4r7s' + payload + '3nd'
    if encoding:
        checkString = encoding(unquote(checkString))
    response = requester(url, replaceValue(
        params, xsschecker, checkString, copy.deepcopy), headers, GET, delay, timeout).text.lower()
    reflectedPositions = []
    for match in re.finditer('st4r7s', response):
        reflectedPositions.append(match.start())
    filledPositions = fillHoles(positions, reflectedPositions)
    num = 0
    efficiencies = []
    for position in filledPositions:
        allEfficiencies = []
        try:
            reflected = response[reflectedPositions[num]
                :reflectedPositions[num]+len(checkString)]
            efficiency = fuzz.partial_ratio(reflected, checkString.lower())
            allEfficiencies.append(efficiency)
        except IndexError:
            pass
        if position:
            reflected = response[position:position+len(checkString)]
            if encoding:
                checkString = encoding(checkString.lower())
            efficiency = fuzz.partial_ratio(reflected, checkString)
            if reflected[:-2] == ('\\%s' % checkString.replace('st4r7s', '').replace('3nd', '')):
                efficiency = 90
            allEfficiencies.append(efficiency)
            efficiencies.append(max(allEfficiencies))
        else:
            efficiencies.append(0)
        num += 1
    return list(filter(None, efficiencies))
```

“zetanize.py”

```
import re


def zetanize(response):
    def e(string):
        return string.encode('utf-8')

    def d(string):
        return string.decode('utf-8')
    response = re.sub(r'(?s)<!--.*?-->', '', response)
    forms = {}
    matches = re.findall(r'(?i)(?s)<form.*?</form.*?>',
                         response)
    num = 0
    for match in matches:
        page = re.search(r'(?i)action=[\'"](.*?)[\'"]', match)
        method = re.search(r'(?i)method=[\'"](.*?)[\'"]', match)
        forms[num] = {}
        forms[num]['action'] = d(e(page.group(1))) if page else ''
        forms[num]['method'] = d(
            e(method.group(1)).lower()) if method else 'get'
```

```
        forms[num]['inputs'] = []
        inputs = re.findall(r'(?i)(?s)<input.*?>', response)
        for inp in inputs:
            inpName = re.search(r'(?i)name=[\'"](.*?)[\'"]', inp)
            if inpName:
                inpType = re.search(r'(?i)type=[\'"](.*?)[\'"]', inp)
                inpValue = re.search(r'(?i)value=[\'"](.*?)[\'"]', inp)
                inpName = d(e(inpName.group(1)))
                inpType = d(e(inpType.group(1)))if inpType else ''
                inpValue = d(e(inpValue.group(1))) if inpValue else ''
                if inpType.lower() == 'submit' and inpValue == '':
                    inpValue = 'Submit Query'
                inpDict = {
                    'name': inpName,
                    'type': inpType,
                    'value': inpValue
                }
                forms[num]['inputs'].append(inpDict)
        num += 1
    return forms
```

“dom.py”

```
import re

from core.colors import end, red, yellow

if len(end) < 1:
    end = red = yellow = '*'

def dom(response):
    highlighted = []
    sources =
r'''\b(?:document\.(URL|documentURI|URLUnencoded|baseURI|cookie|referrer)|location\.(href|search|hash|pathnam
e)|window\.name|history\.(pushState|replaceState)(local|session)Storage)\b'''
    sinks =
r'''\b(?:eval|evaluate|execCommand|assign|navigate|getResponseHeaderopen|showModalDialog|Function|set(Timeou
t|Interval|Immediate)|execScript|crypto.generateCRMFRequest|ScriptElement\.(src|text|textContent|innerText)|.*?\.o
nEventName|document\.(write|writeln)|.*?\.innerHTML|Range\.createContextualFragment|(document|window)\.loca
tion)\b'''
    scripts = re.findall(r'(?i)(?s)<script[^>]*>(.*?)</script>', response)
    sinkFound, sourceFound = False, False
    for script in scripts:
        script = script.split('\n')
        num = 1
        allControlledVariables = set()
        try:
            for newLine in script:
                line = newLine
                parts = line.split('var ')
                controlledVariables = set()
                if len(parts) > 1:
                    for part in parts:
                        for controlledVariable in allControlledVariables:
                            if controlledVariable in part:
                                controlledVariables.add(re.search(r'[a-zA-Z$_][a-zA-Z0-9$_]+', part).group().replace('$', '\$'))
                pattern = re.finditer(sources, newLine)
                for grp in pattern:
                    if grp:
                        source = newLine[grp.start():grp.end()].replace(' ', '')
                        if source:
                            if len(parts) > 1:
                                for part in parts:
```

```
                        if source in part:
                            controlledVariables.add(re.search(r'[a-zA-Z$_][a-zA-Z0-9$_]+', part).group().replace('$',
'\$'))
                    line = line.replace(source, yellow + source + end)
            for controlledVariable in controlledVariables:
                allControlledVariables.add(controlledVariable)
            for controlledVariable in allControlledVariables:
                matches = list(filter(None, re.findall(r'\b%s\b' % controlledVariable, line)))
                if matches:
                    sourceFound = True
                    line = re.sub(r'\b%s\b' % controlledVariable, yellow + controlledVariable + end, line)
            pattern = re.finditer(sinks, newLine)
            for grp in pattern:
                if grp:
                    sink = newLine[grp.start():grp.end()].replace(' ', '')
                    if sink:
                        line = line.replace(sink, red + sink + end)
                        sinkFound = True
            if line != newLine:
                highlighted.append('%-3s %s' % (str(num), line.lstrip(' ')))
            num += 1
        except MemoryError:
            pass
    if sinkFound or sourceFound:
        return highlighted
    else:
        return []
```

“htmlParser.py”

```
import re

from core.config import badTags, xsschecker
from core.utils import isBadContext, equalize, escaped, extractScripts


def htmlParser(response, encoding):
    rawResponse = response
    response = response.text
    if encoding:
        response = response.replace(encoding(xsschecker), xsschecker)
    reflections = response.count(xsschecker)
    position_and_context = {}
    environment_details = {}
    clean_response = re.sub(r'<!--[.\s\S]*?-->', '', response)
    script_checkable = clean_response
    for script in extractScripts(script_checkable):
        occurences = re.finditer(r'(%s.*?)$' % xsschecker, script)
        if occurences:
            for occurence in occurences:
                thisPosition = occurence.start(1)
                position_and_context[thisPosition] = 'script'
                environment_details[thisPosition] = {}
                environment_details[thisPosition]['details'] = {'quote' : ''}
                for i in range(len(occurence.group())):
                    currentChar = occurence.group()[i]
                    if currentChar in ('/', '\'', '`', '"') and not escaped(i, occurence.group()):
                        environment_details[thisPosition]['details']['quote'] = currentChar
                    elif currentChar in (')', ']', '}', '}') and not escaped(i, occurence.group()):
                        break
                script_checkable = script_checkable.replace(xsschecker, '', 1)
    if len(position_and_context) < reflections:
        attribute_context = re.finditer(r'<[^>]*?(%s)[^>]*?>' % xsschecker, clean_response)
```

```
        for occurence in attribute_context:
            match = occurence.group(0)
            thisPosition = occurence.start(1)
            parts = re.split(r'\s', match)
            tag = parts[0][1:]
            for part in parts:
                if xsschecker in part:
                    Type, quote, name, value = '', '', '', ''
                    if '=' in part:
                        quote = re.search(r'=([\'`"])?', part).group(1)
                        name_and_value = part.split('=')[0], '='.join(part.split('=')[1:])
                        if xsschecker == name_and_value[0]:
                            Type = 'name'
                        else:
                            Type = 'value'
                        name = name_and_value[0]
                        value = name_and_value[1].rstrip('>').rstrip(quote).lstrip(quote)
                    else:
                        Type = 'flag'
                    position_and_context[thisPosition] = 'attribute'
                    environment_details[thisPosition] = {}
                    environment_details[thisPosition]['details'] = {'tag' : tag, 'type' : Type, 'quote' : quote, 'value' : value,
'name' : name}
    if len(position_and_context) < reflections:
        html_context = re.finditer(xsschecker, clean_response)
        for occurence in html_context:
            thisPosition = occurence.start()
            if thisPosition not in position_and_context:
                position_and_context[occurence.start()] = 'html'
                environment_details[thisPosition] = {}
                environment_details[thisPosition]['details'] = {}
    if len(position_and_context) < reflections:
        comment_context = re.finditer(r'<!--[\s\S]*?(%s)[\s\S]*?-->' % xsschecker, response)
        for occurence in comment_context:
            thisPosition = occurence.start(1)
            position_and_context[thisPosition] = 'comment'
            environment_details[thisPosition] = {}
            environment_details[thisPosition]['details'] = {}
    database = {}
    for i in sorted(position_and_context):
        database[i] = {}
        database[i]['position'] = i
        database[i]['context'] = position_and_context[i]
        database[i]['details'] = environment_details[i]['details']

    bad_contexts = re.finditer(r'(?s)(?i)<(style|template|textarea|title|noembed|noscript)>[.\s\S]*(%s)[.\s\S]*</\1>' %
xsschecker, response)
    non_executable_contexts = []
    for each in bad_contexts:
        non_executable_contexts.append([each.start(), each.end(), each.group(1)])

    if non_executable_contexts:
        for key in database.keys():
            position = database[key]['position']
            badTag = isBadContext(position, non_executable_contexts)
            if badTag:
                database[key]['details']['badTag'] = badTag
            else:
                database[key]['details']['badTag'] = ''
    return database
```

“filterCheker.py”

```
from core.checker import checker


def filterChecker(url, params, headers, GET, delay, occurences, timeout, encoding):
    positions = occurences.keys()
    sortedEfficiencies = {}
    environments = set(['<', '>'])
    for i in range(len(positions)):
        sortedEfficiencies[i] = {}
    for i in occurences:
        occurences[i]['score'] = {}
        context = occurences[i]['context']
        if context == 'comment':
            environments.add('-->')
        elif context == 'script':
            environments.add(occurences[i]['details']['quote'])
            environments.add('</scRipT/>')
        elif context == 'attribute':
            if occurences[i]['details']['type'] == 'value':
                if occurences[i]['details']['name'] == 'srcdoc':
                    environments.add('&lt;')
            if occurences[i]['details']['quote']:
                environments.add(occurences[i]['details']['quote'])
    for environment in environments:
        if environment:
            efficiencies = checker(
                url, params, headers, GET, delay, environment, positions, timeout, encoding)
            efficiencies.extend([0] * (len(occurences) - len(efficiencies)))
            for occurence, efficiency in zip(occurences, efficiencies):
                occurences[occurence]['score'][environment] = efficiency
    return occurences
```

“fuzzer.py”

```
import copy
from random import randint
from time import sleep
from urllib.parse import unquote

from core.colors import end, red, green, yellow
from core.config import fuzzes, xsschecker
from core.requester import requester
from core.utils import replaceValue, counter
from core.log import setup_logger

logger = setup_logger(__name__)


def fuzzer(url, params, headers, GET, delay, timeout, WAF, encoding):
    for fuzz in fuzzes:
        if delay == 0:
            delay = 0
        t = delay + randint(delay, delay * 2) + counter(fuzz)
        sleep(t)
        try:
            if encoding:
                fuzz = encoding(unquote(fuzz))
            data = replaceValue(params, xsschecker, fuzz, copy.deepcopy)
            response = requester(url, data, headers, GET, delay/2, timeout)
        except:
            logger.error('WAF is dropping suspicious requests.')
            if delay == 0:
                logger.info('Delay has been increased to %s6%s seconds.' % (green, end))
```

```
            delay += 6
        limit = (delay + 1) * 50
        timer = -1
        while timer < limit:
            logger.info('\rFuzzing will continue after %s%i%s seconds.\t\t\r' % (green, limit, end))
            limit -= 1
            sleep(1)
        try:
            requester(url, params, headers, GET, 0, 10)
            logger.good('Pheww! Looks like sleeping for %s%i%s seconds worked!' % (
                green, ((delay + 1) * 2), end))
        except:
            logger.error('\nLooks like WAF has blocked our IP Address. Sorry!')
            break
    if encoding:
        fuzz = encoding(fuzz)
    if fuzz.lower() in response.text.lower():
        result = ('%s[passed]  %s' % (green, end))

    elif str(response.status_code)[:1] != '2':
        result = ('%s[blocked] %s' % (red, end))
    else:
        result = ('%s[filtered]%s' % (yellow, end))
    logger.info('%s %s' % (result, fuzz))
```

Блок 2. Допоміжні бібліотеки для JS

“definition.json”

```
{
        "retire-example": {
                "vulnerabilities" : [
                        {
                                "below" : "0.0.2",
                                "severity" : "low",
                                "identifiers" : {
                                        "CVE" : [ "CVE-XXXX-XXXX" ],
                                        "bug" : "1234",
                                        "summary" : "bug summary"
                                },
                                "info" : [ "http://github.com/eoftedal/retire.js/" ]
                        }
                ],
                "extractors" : {
                        "func" : [ "retire.VERSION" ],
                        "filename" : [ "retire-example-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "filecontent"       : [ "/\\*!? Retire-example v([0-9][0-9.a-z_-]+)" ],
                        "hashes" : { "07f8b94c8d601a24a1914a1a92bec0e4fafda964" : "0.0.1" }
                }
        },

        "jquery": {
                "bowername": [ "jQuery" ],
                "vulnerabilities" : [
                        {
                                "below" : "1.6.3",
                                "severity" : "medium",
                                "identifiers" : {
                                        "CVE": [ "CVE-2011-4969" ],
                                        "summary": "XSS with location.hash"
                                },
                                "info" : [ "https://nvd.nist.gov/vuln/detail/CVE-2011-4969" ,
"http://research.insecurelabs.org/jquery/test/", "https://bugs.jquery.com/ticket/9521" ]
```

```
                },
                {
                        "below" : "1.9.0b1",
                        "identifiers": {
                                "CVE" : [ "CVE-2012-6708" ],

                                "bug": "11290",
                                "summary": "Selector interpreted as HTML"
                        },
                        "severity": "medium",
                        "info" : [ "http://bugs.jquery.com/ticket/11290" ,
"https://nvd.nist.gov/vuln/detail/CVE-2012-6708", "http://research.insecurelabs.org/jquery/test/" ]
                },
                {
                        "atOrAbove" : "1.4.0",
                        "below" : "1.12.0",
                        "identifiers": {
                                "issue" : "2432",
                                "summary": "3rd party CORS request may execute",
                             "CVE": [ "CVE-2015-9251" ]
                        },
                        "severity": "medium",
                        "info" : [ "https://github.com/jquery/jquery/issues/2432",
"http://blog.jquery.com/2016/01/08/jquery-2-2-and-1-12-released/", "https://nvd.nist.gov/vuln/detail/CVE-2015-
9251", "http://research.insecurelabs.org/jquery/test/" ]
                },
                {
                        "atOrAbove" : "1.12.3",
                        "below" : "3.0.0-beta1",
                        "identifiers": {
                                "issue" : "2432",
                                "summary": "3rd party CORS request may execute",
                             "CVE": [ "CVE-2015-9251" ]
                        },
                        "severity": "medium",
                        "info" : [ "https://github.com/jquery/jquery/issues/2432",
"http://blog.jquery.com/2016/01/08/jquery-2-2-and-1-12-released/", "https://nvd.nist.gov/vuln/detail/CVE-2015-
9251", "http://research.insecurelabs.org/jquery/test/" ]
                },
                {
                        "atOrAbove" : "1.8.0",
                        "below" : "1.12.0",
                        "identifiers": {
                                "CVE" : [ "CVE-2015-9251" ],
                                "issue" : "11974",
                                "summary": "parseHTML() executes scripts in event handlers"
                        },
                        "severity": "medium",
                        "info" : [ "https://bugs.jquery.com/ticket/11974",
"https://nvd.nist.gov/vuln/detail/CVE-2015-9251", "http://research.insecurelabs.org/jquery/test/" ]
                },
                {
                        "atOrAbove" : "1.12.2",
                        "below" : "2.2.0",
                        "identifiers": {
                                "CVE" : [ "CVE-2015-9251" ],
                                "issue" : "11974",
                                "summary": "parseHTML() executes scripts in event handlers"
                        },
                        "severity": "medium",
```

```
                                        "info" : [ "https://bugs.jquery.com/ticket/11974",
"https://nvd.nist.gov/vuln/detail/CVE-2015-9251", "http://research.insecurelabs.org/jquery/test/" ]
                                },
                                {
                                        "atOrAbove" : "2.2.2",
                                        "below" : "3.0.0",
                                        "identifiers": {
                                                "CVE" : [ "CVE-2015-9251" ],
                                                "issue" : "11974",
                                                "summary": "parseHTML() executes scripts in event handlers"
                                        },
                                        "severity": "medium",
                                        "info" : [ "https://bugs.jquery.com/ticket/11974",
"https://nvd.nist.gov/vuln/detail/CVE-2015-9251", "http://research.insecurelabs.org/jquery/test/" ]
                                },
                                {
                                        "below" : "3.4.0",
                                        "identifiers": {
                                                "CVE" : [ "CVE-2019-11358" ],
                                                "summary": "jQuery before 3.4.0, as used in Drupal, Backdrop CMS,
and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution"
                                        },
                                        "severity" : "low",
                                        "info" : [ "https://blog.jquery.com/2019/04/10/jquery-3-4-0-released/",
"https://nvd.nist.gov/vuln/detail/CVE-2019-11358",
"https://github.com/jquery/jquery/commit/753d591aea698e57d6db58c9f722cd0808619b1b" ]
                                }



                        ],
                        "extractors" : {
                                "func"                  : [
                                                                        "(jQuery || $ || $jq || $j).fn.jquery",
                                                                        "require('jquery').fn.jquery"

                                ],
                                "uri"                           : [ "/([0-9][0-9.a-z_-]+)/jquery(\\.min)?\\.js" ],
                                "filename"                      : [ "jquery-([0-9][0-9.a-z_-]+)(\\.min)?\\.js" ],
                                "filecontent"       : [
                                                                        "/\\*!? jQuery v([0-9][0-9.a-z_-]+)", "\\*
jQuery JavaScript Library v([0-9][0-9.a-z_-]+)",
                                                                        "\\* jQuery ([0-9][0-9.a-z_-]+) - New Wave
Javascript", "// \\$Id: jquery.js,v ([0-9][0-9.a-z_-]+)",
                                                                        "/\\*! jQuery v([0-9][0-9.a-z_-]+)",
                        "[^a-z]f=\"([0-9][0-9.a-z_-]+)\",.*[^a-z]jquery:f,",
                        "[^a-z]m=\"([0-9][0-9.a-z_-]+)\",.*[^a-z]jquery:m,",
                        "[^a-z.]jquery:[ ]?\"([0-9][0-9.a-z_-]+)\"",
                        "\\$\\.documentElement,Q=e.jQuery,Z=e\\.\\$,ee=\\{\\},te=\\[\\],ne=\"([0-9][0-9.a-z_-]+)\""
                                ],
                                "filecontentreplace" : [
                                        "/var [a-z]=[a-z]\\.document,([a-z])=\"([0-9][0-9.a-z_-]+)\",([a-
z])=.{130,160};\\3\\.fn=\\3\\.prototype=\\{jquery:\\1/$2/"
                                ],
                                "hashes"                : {}
                        }
                },
                "jquery-migrate" : {
                        "vulnerabilities" : [
                                {
                                        "below" : "1.2.0",
                                        "severity": "medium",
```

```
                                "identifiers": {
                                        "release": "jQuery Migrate 1.2.0 Released",
                                        "summary": "cross-site-scripting"
                                },
                                "info" : [ "http://blog.jquery.com/2013/05/01/jquery-migrate-1-2-0-released/" ]
                        },
                        {
                                "below" : "1.2.2",
                                "severity": "medium",
                                "identifiers": {
                                        "bug": "11290",
                                        "summary": "Selector interpreted as HTML"
                                },
                                "info" : [ "http://bugs.jquery.com/ticket/11290" ,
"http://research.insecurelabs.org/jquery/test/" ]
                        }
                ],
                "extractors" : {
                        "filename"                      : [ "jquery-migrate-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "filecontent"       : [ "/\\*!?(?:\n \\*)? jQuery Migrate(?: -)? v([0-9][0-9.a-z_-]+)" ],
                        "hashes"            : {}
                }
        },
        "jquery.validator" : {
                "bowername": [ "jquery-validator" ],
                "vulnerabilities" : [
                ],
                "extractors" : {
                        "func"              : [ "jQuery.validation.version" ],
                        "filename"                      : [ "jquery.validation-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "uri"                           : [ "/([0-9][0-9.a-z_-]+)/jquery.validation(\\.min)?\\.js" ],
                        "filecontent"       : [ "/\\*!?(?:\n \\*)? jQuery Validation Plugin v([0-9][0-9.a-z_-]+)" ],
                        "hashes"            : {}
                }
        },
        "jquery-mobile" : {
                "bowername": [ "jquery-mobile", "jquery-mobile-min", "jquery-mobile-build", "jquery-mobile-
dist", "jquery-mobile-bower" ],
                "vulnerabilities" : [
                        {
                                "below" : "1.0RC2",
                                "severity": "high",
                                "identifiers": {"osvdb": ["94563", "93562", "94316", "94561", "94560"]},
                                "info" : [ "http://osvdb.org/show/osvdb/94563",
"http://osvdb.org/show/osvdb/94562", "http://osvdb.org/show/osvdb/94316", "http://osvdb.org/show/osvdb/94561",
"http://osvdb.org/show/osvdb/94560" ]
                        },
                        {
                                "below" : "1.0.1",
                                "severity": "high",
                                "identifiers": {"osvdb": ["94317"]},
                                "info": [ "http://osvdb.org/show/osvdb/94317" ]
                        },
                        {
                                "below" : "1.1.2",
                                "severity": "medium",
                                "identifiers": {
                                        "issue": "4787",
                                        "release": "http://jquerymobile.com/changelog/1.1.2/",
                                        "summary": "location.href cross-site scripting"
                                },
```

```
                                "info": [ "http://jquerymobile.com/changelog/1.1.2/",
"https://github.com/jquery/jquery-mobile/issues/4787" ]
                        },
                        {
                                "below" : "1.2.0",
                                "severity": "medium",
                                "identifiers": {
                                        "issue": "4787",
                                        "release": "http://jquerymobile.com/changelog/1.2.0/",
                                        "summary": "location.href cross-site scripting"
                                },
                                "info": [ "http://jquerymobile.com/changelog/1.2.0/",
"https://github.com/jquery/jquery-mobile/issues/4787" ]
                        },
                        {
                                "below" : "100.0.0",
                                "severity": "medium",
                                "identifiers": {
                                        "summary": "open redirect leads to cross site scripting"
                                },
                                "info": [ "http://sirdarckcat.blogspot.no/2017/02/unpatched-0day-jquery-mobile-
xss.html" ]
                        }
                ],
                "extractors" : {
                        "func"          : [ "jQuery.mobile.version" ],
                        "filename"              : [ "jquery.mobile-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "uri"                   : [ "/([0-9][0-9.a-z_-]+)/jquery.mobile(\\.min)?\\.js" ],
                        "filecontent"   : [ "/\\*!?(?:\n \\*)? jQuery Mobile(?: -)? v([0-9][0-9.a-z_-]+)" ],
                        "hashes"        : {}
                }
        },
        "jquery-ui-dialog" : {
                "bowername": [ "jquery-ui", "jquery.ui" ],
                "vulnerabilities" : [
                        {
                                "atOrAbove": "1.8.9",
                                "below" : "1.10.0",
                                "severity": "medium",
                                "identifiers": {
                                        "CVE": [ "CVE-2010-5312" ],
                                        "bug": "6016",
                                        "summary": "Title cross-site scripting vulnerability"
                                },
                                "info" : [ "http://bugs.jqueryui.com/ticket/6016",
"https://nvd.nist.gov/vuln/detail/CVE-2010-5312" ]
                        },
                        {
                                "below" : "1.12.0",
                                "severity": "high",
                                "identifiers": {
                                        "CVE": [ "CVE-2016-7103" ],
                                        "bug": "281",
                                        "summary": "XSS Vulnerability on closeText option"
                                },
                                "info" : [ "https://github.com/jquery/api.jqueryui.com/issues/281",
"https://nvd.nist.gov/vuln/detail/CVE-2016-7103", "https://snyk.io/vuln/npm:jquery-ui:20160721" ]
                        }
                ],
                "extractors" : {
                        "func"          : [ "jQuery.ui.dialog.version" ],
```

```
                        "filecontent"       : [
                                "/\\*!? jQuery UI - v([0-9][0-9.a-z_-]+)(.*\n){1,3}.*jquery\\.ui\\.dialog\\.js",
                                "/\\*!?[\n *]+jQuery UI ([0-9][0-9.a-z_-]+)(.*\n)*.*\\.ui\\.dialog",
                                "/\\*!?[\n *]+jQuery UI Dialog ([0-9][0-9.a-z_-]+)",
                                "/\\*!? jQuery UI - v([0-9][0-9.a-z_-]+)(.*\n){1,3}\\* Includes: .* dialog\\.js"
                        ],
                        "hashes"            : {}
                }
        },
        "jquery-ui-autocomplete" : {
                "bowername": [ "jquery-ui", "jquery.ui" ],
                "vulnerabilities" : [ ],
                "extractors" : {
                        "func"              : [ "jQuery.ui.autocomplete.version" ],
                        "filecontent"       : [
                                "/\\*!? jQuery UI - v([0-9][0-9.a-z_-
]+)(.*\n){1,3}.*jquery\\.ui\\.autocomplete\\.js",
                                "/\\*!?[\n *]+jQuery UI ([0-9][0-9.a-z_-]+)(.*\n)*.*\\.ui\\.autocomplete",
                                "/\\*!?[\n *]+jQuery UI Autocomplete ([0-9][0-9.a-z_-]+)",
                                "/\\*!? jQuery UI - v([0-9][0-9.a-z_-]+)(.*\n){1,3}\\* Includes: .*
autocomplete\\.js"
                        ],
                        "hashes"            : {}
                }
        },
        "jquery-ui-tooltip" : {
                "bowername": [ "jquery-ui", "jquery.ui" ],
                "vulnerabilities" : [
                        {
                                "atOrAbove": "1.9.2",
                                "below" : "1.10.0",
                                "severity": "high",
                                "identifiers": {
                                        "CVE" : [ "CVE-2012-6662" ],
                                        "bug": "8859",
                                        "summary": "Autocomplete cross-site scripting vulnerability"
                                },
                                "info" : [ "http://bugs.jqueryui.com/ticket/8859",
"https://nvd.nist.gov/vuln/detail/CVE-2012-6662" ]
                        }
                ],
                "extractors" : {
                        "func"              : [ "jQuery.ui.tooltip.version" ],
                        "filecontent"       : [
                                "/\\*!? jQuery UI - v([0-9][0-9.a-z_-]+)(.*\n){1,3}.*jquery\\.ui\\.tooltip\\.js",
                                "/\\*!?[\n *]+jQuery UI ([0-9][0-9.a-z_-]+)(.*\n)*.*\\.ui\\.tooltip",
                                "/\\*!?[\n *]+jQuery UI Tooltip ([0-9][0-9.a-z_-]+)"
                        ],
                        "hashes"            : {}
                }
        },
        "jquery.prettyPhoto" : {
                "bowername": [ "jquery-prettyPhoto" ],
                "vulnerabilities" : [
                        {
                                "below" : "3.1.5",
                                "severity" : "high",
                                "identifiers" : { "CVE" : [ "CVE-2013-6837" ] },
                                "info" : [ "https://nvd.nist.gov/vuln/detail/CVE-2013-6837" ]
                        },
                        {
```

```
                                "below" : "3.1.6",
                                "severity" : "high",
                                "info" : [ "https://github.com/scaron/prettyphoto/issues/149",
"https://blog.anantshri.info/forgotten_disclosure_dom_xss_prettyphoto" ]
                        }

                ],
                "extractors" : {
                        "func"          : [ "jQuery.prettyPhoto.version" ],
                        "filecontent"   : [
                                "/\\*(?:.*[\n\r]+){1,3}.*Class: prettyPhoto(?:.*[\n\r]+){1,3}.*Version: ([0-9][0-
9.a-z_-]+)",
                                "\\.prettyPhoto[ ]?=[ ]?\\{version:[ ]?(?:'|\")([0-9][0-9.a-z_-]+)(?:'|\")\\}"
                        ],
                        "hashes"        : {}
                }
        },
        "jPlayer" : {
                "bowername": [ "jPlayer" ],
                "vulnerabilities" : [
                        {
                                "below" : "2.3.1",
                                "severity": "high",
                                "identifiers": {
                                        "CVE": [ "CVE-2013-2023" ],
                                        "release" : "2.3.1",
                                        "summary" : "XSS vulnerability in actionscript/Jplayer.as in the Flash
SWF component" },
                                "info" : [ "http://jplayer.org/latest/release-notes/",
"https://nvd.nist.gov/vuln/detail/CVE-2013-2023" ]
                        },
                        {
                                "below" : "2.3.23",
                                "severity": "high",
                                "identifiers": {
                                        "CVE": [ "CVE-2013-2022" ],
                                        "release": "2.3.23",
                                        "summary": "XSS vulnerabilities in actionscript/Jplayer.as in the Flash
SWF component"
                                },
                                "info" : [ "http://jplayer.org/latest/release-notes/",
"https://nvd.nist.gov/vuln/detail/CVE-2013-2022" ]
                        },
                        {
                                "below" : "2.2.20",
                                "severity": "high",
                                "identifiers": {
                                        "CVE": [ "CVE-2013-1942" ],
                                        "release": "2.2.20",
                                        "summary": "XSS vulnerabilities in actionscript/Jplayer.as in the Flash
SWF component"
                                },
                                "info" : [ "http://jplayer.org/latest/release-notes/",
"https://nvd.nist.gov/vuln/detail/CVE-2013-1942" ]
                        }
                ],
                "extractors" : {
                        "func"          : [ "new jQuery.jPlayer().version.script" ],
                        "filecontent"   : [
                                "/\\*(?:.*[\n\r]+){1,3}.*jPlayer Plugin for jQuery(?:.*[\n\r]+){1,10}.*Version:
([0-9][0-9.a-z_-]+)"
```

```
                ],
                "hashes"            : {}
        }
    },
    "knockout": {
        "vulnerabilities" : [
            {
                "below" : "3.5.0-beta",
                "severity": "medium",
                "identifiers": {"summary": "XSS injection point in attr name binding for
browser IE7 and older"},
                "info" : [ "https://github.com/knockout/knockout/issues/1244" ]
            }
        ],
        "extractors" : {
            "func"              : [ "ko.version" ],
            "filename"                  : [ "knockout-([0-9][0-9.a-z_-]+)(.min)?\\.js"],
            "filecontent"       : [
                "\\* Knockout JavaScript library v([0-9][0-9.a-z_-]+)"
            ],
            "hashes"            : {}
        }
    },
    "sessvars": {
        "vulnerabilities" : [
            {
                "below" : "1.01",
                "severity": "low",
                "identifiers": {"summary": "Unsanitized data passed to eval()"},
                "info" : [ "http://www.thomasfrank.se/sessionvars.html" ]
            }
        ],
        "extractors" : {
            "filename"                  : [ "sessvars-([0-9][0-9.a-z_-]+)(.min)?\\.js"],
            "filecontent"       : [ "sessvars ver ([0-9][0-9.a-z_-]+)"],
            "hashes"            : {}
        }
    },
    "swfobject": {
        "bowername": [ "swfobject", "swfobject-bower" ],
        "vulnerabilities" : [
            {
                "below" : "2.1",
                "severity": "medium",
                "identifiers": {"summary": "DOM-based XSS"},
                "info" : [ "https://github.com/swfobject/swfobject/wiki/SWFObject-Release-
Notes#swfobject-v21-beta7-june-6th-2008" ]
            }
        ],
        "extractors" : {
            "filename"                  : [ "swfobject_([0-9][0-9.a-z_-]+)(.min)?\\.js"],
            "filecontent"       : [ "SWFObject v([0-9][0-9.a-z_-]+) "],
            "hashes"            : {}
        }
    },

    "tinyMCE" : {
        "bowername": [ "tinymce", "tinymce-dist" ],
        "vulnerabilities" : [
            {
                "below" : "1.4.2",
```

```
                                "severity" : "high",
                                "identifiers" : {
                                        "summary" : "Static code injection vulnerability in
inc/function.base.php",
                                        "CVE" : [ "CVE-2011-4825" ]
                                },
                                "info" : [ "http://www.cvedetails.com/cve/CVE-2011-4825/" ]
                        },
                        {
                                "below" : "4.2.4",
                                "severity" : "medium",
                                "identifiers" : { "summary" : "xss issues with media plugin not properly filtering
out some script attributes." },
                                "info" : [ "https://www.tinymce.com/docs/changelog/" ]

                        },
                        {
                                "below" : "4.2.0",
                                "severity" : "medium",
                                "identifiers" : { "summary" : "FIXED so script elements gets removed by default
to prevent possible XSS issues in default config implementations" },
                                "info" : [ "https://www.tinymce.com/docs/changelog/" ]

                        },
                        {
                                "below" : "4.7.12",
                                "severity" : "medium",
                                "identifiers" : { "summary" : "FIXED so links with xlink:href attributes are
filtered correctly to prevent XSS." },
                                "info" : [ "https://www.tinymce.com/docs/changelog/" ]

                        }
                ],
                "extractors" : {
                        "filecontent"           : [ "// ([0-9][0-9.a-z_-]+) \\([0-9\\-
]+\\)[\n\r]+.{0,1200}l=.tinymce/geom/Rect." ],
                        "filecontentreplace" : [

        "/tinyMCEPreInit.*majorVersion:.([0-9]+).,minorVersion:.([0-9.]+)./$1.$2/",
                                                                                "/majorVersion:.([0-
9]+).,minorVersion:.([0-9.]+).,.*tinyMCEPreInit/$1.$2/"
                                                                        ],
                        "func"                                  : [ "tinyMCE.majorVersion + '.'+
tinyMCE.minorVersion" ]
                }
        },

        "YUI" : {
                "bowername": [ "yui", "yui3" ],
                "vulnerabilities" : [
                        {
                                "atOrAbove" : "3.5.0" ,
                                "below" : "3.9.2",
                                "severity": "high",
                                "identifiers": {"CVE": [ "CVE-2013-4942" ] },
                                "info" : [ "http://www.cvedetails.com/cve/CVE-2013-4942/" ]
                        },
                        {
                                "atOrAbove" : "3.2.0" ,
                                "below" : "3.9.2",
                                "severity": "high",
```

```
            "identifiers": {"CVE": [ "CVE-2013-4941" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2013-4941/" ]
        },
        {
            "atOrAbove" : "3.0.0",
            "below" : "3.10.3",
            "severity": "high",
            "identifiers": {"CVE": [ "CVE-2013-4940" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2013-4940/" ]
        },
        {
            "atOrAbove" : "3.0.0" ,
            "below" : "3.9.2",
            "severity": "high",
            "identifiers": {"CVE": [ "CVE-2013-4939" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2013-4939/" ]
        },
        {
            "atOrAbove" : "2.8.0" ,
            "below" : "2.9.1",
            "severity": "high",
            "identifiers": {"CVE": [ "CVE-2012-5883" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2012-5883/" ]
        },
        {
            "atOrAbove" : "2.5.0" ,
            "below" : "2.9.1",
            "severity": "high",
            "identifiers": {"CVE": [ "CVE-2012-5882" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2012-5882/" ]
        },
        {
            "atOrAbove" : "2.4.0" ,
            "below" : "2.9.1",
            "severity": "high",
            "identifiers": {"CVE": [ "CVE-2012-5881" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2012-5881/" ]
        },
        {
            "below" : "2.9.0",
            "severity": "medium",
            "identifiers": {"CVE": [ "CVE-2010-4710" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2010-4710/" ]
        },
        {
            "atOrAbove" : "2.8.0" ,
            "below" : "2.8.2",
            "severity": "high",
            "identifiers": {"CVE": [ "CVE-2010-4209" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2010-4209/" ]
        },
        {
            "atOrAbove" : "2.5.0" ,
            "below" : "2.8.2",
            "severity": "high",
            "identifiers": {"CVE": [ "CVE-2010-4208" ] },
            "info" : [ "http://www.cvedetails.com/cve/CVE-2010-4208/" ]
        },
        {
            "atOrAbove" : "2.4.0" ,
            "below" : "2.8.2",
```

```
                                "severity": "high",
                                "identifiers": {"CVE": [ "CVE-2010-4207" ] },
                                "info" : [ "http://www.cvedetails.com/cve/CVE-2010-4207/" ]
                        }
                ],
                "extractors" : {
                        "func"          : [ "YUI.Version", "YAHOO.VERSION" ],
                        "filename"              : [ "yui-([0-9][0-9.a-z_-]+)(.min)?\\.js"],
                        "filecontent"   : [ "/*\nYUI ([0-9][0-9.a-z_-]+)", "/yui/license.(?:html|txt)\nversion:
([0-9][0-9.a-z_-]+)"],
                        "hashes"        : {}
                }
        },
        "prototypejs" : {
                "bowername": [ "prototypejs", "prototype.js", "prototypejs-bower" ],
                "vulnerabilities" : [
                        {
                                "atOrAbove" : "1.6.0",
                                "below" : "1.6.0.2",
                                "severity": "high",
                                "identifiers": {"CVE": [ "CVE-2008-7220" ] },
                                "info" : [ "http://www.cvedetails.com/cve/CVE-2008-7220/",
"http://prototypejs.org/2008/01/25/prototype-1-6-0-2-bug-fixes-performance-improvements-and-security/" ] },
                        {
                                "below" : "1.5.1.2",
                                "severity": "high",
                                "identifiers": {"CVE": [ "CVE-2008-7220" ] },
                                "info" : [ "http://www.cvedetails.com/cve/CVE-2008-7220/",
"http://prototypejs.org/2008/01/25/prototype-1-6-0-2-bug-fixes-performance-improvements-and-security/" ] }
                ],
                "extractors" : {
                        "func"          : [ "Prototype.Version" ],
                        "uri"                   : [ "/([0-9][0-9.a-z_-]+)/prototype(\\.min)?\\.js" ],
                        "filename"              : [ "prototype-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "filecontent"   : [ "Prototype JavaScript framework, version ([0-9][0-9.a-z_-]+)",
                                                                "Prototype[ ]?=[ ]?\\{[ \r\n\t]*Version:[
]?(?:'|\")([0-9][0-9.a-z_-]+)(?:'|\")" ],
                        "hashes"        : {}
                }
        },
        "ember" : {
                "vulnerabilities" : [
                        {
                                "atOrAbove" : "1.8.0",
                                "below" :"1.11.4",
                                "severity" : "medium",
                                "identifiers": {"CVE": [ "CVE-2015-7565" ] },
                                "info": [ "https://groups.google.com/forum/#!topic/ember-
security/OfyQkoSuppY" ]
                        },
                        {
                                "atOrAbove" : "1.12.0",
                                "below" :"1.12.2",
                                "severity" : "medium",
                                "identifiers": {"CVE": [ "CVE-2015-7565" ] },
                                "info": [ "https://groups.google.com/forum/#!topic/ember-
security/OfyQkoSuppY" ]
                        },
                        {
                                "atOrAbove" : "1.13.0",
                                "below" : "1.13.12",
```

```
                        "severity" : "medium",
                        "identifiers": {"CVE": [ "CVE-2015-7565" ] },
                        "info": [ "https://groups.google.com/forum/#!topic/ember-
security/OfyQkoSuppY" ]
                },
                {
                        "atOrAbove" : "2.0.0",
                        "below" : "2.0.3",
                        "severity" : "medium",
                        "identifiers": {"CVE": [ "CVE-2015-7565" ] },
                        "info": [ "https://groups.google.com/forum/#!topic/ember-
security/OfyQkoSuppY" ]
                },
                {
                        "atOrAbove" : "2.1.0",
                        "below" : "2.1.2",
                        "severity" : "medium",
                        "identifiers": {"CVE": [ "CVE-2015-7565" ] },
                        "info": [ "https://groups.google.com/forum/#!topic/ember-
security/OfyQkoSuppY" ]
                },
                {
                        "atOrAbove" : "2.2.0",
                        "below" : "2.2.1",
                        "severity" : "medium",
                        "identifiers": {"CVE": [ "CVE-2015-7565" ] },
                        "info": [ "https://groups.google.com/forum/#!topic/ember-
security/OfyQkoSuppY" ]
                },
                {
                        "below" : "1.5.0",
                        "severity": "medium",
                        "identifiers": {
                                "CVE": [ "CVE-2014-0046" ],
                                "summary": "ember-routing-auto-location can be forced to redirect to
another domain"
                        },
                        "info" : [ "https://github.com/emberjs/ember.js/blob/v1.5.0/CHANGELOG.md"
]
                },
                {
                        "atOrAbove" : "1.3.0-*",
                        "below" : "1.3.2",
                        "severity": "medium",
                        "identifiers": {"CVE": [ "CVE-2014-0046" ] },
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/1h6FRgr8lXQ" ]
                },
                {
                        "atOrAbove" : "1.2.0-*",
                        "below" : "1.2.2",
                        "severity": "medium",
                        "identifiers": {"CVE": [ "CVE-2014-0046" ] },
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/1h6FRgr8lXQ" ] },
                {
                        "atOrAbove" : "1.4.0-*",
                        "below" : "1.4.0-beta.2",
                        "severity": "high",
                        "identifiers": {"CVE": ["CVE-2014-0013", "CVE-2014-0014"]},
```

```
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/2kpXXCxISS4", "https://groups.google.com/forum/#!topic/ember-security/PSE4RzTi6l4" ]
                    },
                    {
                        "atOrAbove" : "1.3.0-*",
                        "below" : "1.3.1",
                        "severity": "high",
                        "identifiers": {"CVE": ["CVE-2014-0013", "CVE-2014-0014"]},
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/2kpXXCxISS4", "https://groups.google.com/forum/#!topic/ember-security/PSE4RzTi6l4" ]
                    },
                    {
                        "atOrAbove" : "1.2.0-*",
                        "below" : "1.2.1",
                        "severity": "high",
                        "identifiers": {"CVE": ["CVE-2014-0013", "CVE-2014-0014"]},
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/2kpXXCxISS4", "https://groups.google.com/forum/#!topic/ember-security/PSE4RzTi6l4" ]
                    },
                    {
                        "atOrAbove" : "1.1.0-*",
                        "below" : "1.1.3",
                        "severity": "high",
                        "identifiers": {"CVE": ["CVE-2014-0013", "CVE-2014-0014"]},
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/2kpXXCxISS4", "https://groups.google.com/forum/#!topic/ember-security/PSE4RzTi6l4" ]
                    },
                    {
                        "atOrAbove" : "1.0.0-*",
                        "below" : "1.0.1",
                        "severity": "high",
                        "identifiers": {"CVE": ["CVE-2014-0013", "CVE-2014-0014"]},
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/2kpXXCxISS4", "https://groups.google.com/forum/#!topic/ember-security/PSE4RzTi6l4" ]
                    },
                    {
                        "atOrAbove" : "1.0.0-rc.1",
                        "below" : "1.0.0-rc.1.1",
                        "severity": "medium",
                        "identifiers": {"CVE": [ "CVE-2013-4170" ] },
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/dokLVwwxAdM" ]
                    },
                    {
                        "atOrAbove" : "1.0.0-rc.2",
                        "below" : "1.0.0-rc.2.1",
                        "severity": "medium",
                        "identifiers": {"CVE": [ "CVE-2013-4170" ] },
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/dokLVwwxAdM" ]
                    },
                    {
                        "atOrAbove" : "1.0.0-rc.3",
                        "below" : "1.0.0-rc.3.1",
                        "severity": "medium",
                        "identifiers": {"CVE": [ "CVE-2013-4170" ] },
                        "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/dokLVwwxAdM" ]
                    },
                    {
                        "atOrAbove" : "1.0.0-rc.4",
```

```
                                "below" : "1.0.0-rc.4.1",
                                "severity": "medium",
                                "identifiers": {"CVE": [ "CVE-2013-4170" ] },
                                "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/dokLVwwxAdM" ]
                        },
                        {
                                "atOrAbove" : "1.0.0-rc.5",
                                "below" : "1.0.0-rc.5.1",
                                "severity": "medium",
                                "identifiers": {"CVE": [ "CVE-2013-4170" ] },
                                "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/dokLVwwxAdM" ]
                        },
                        {
                                "atOrAbove" : "1.0.0-rc.6",
                                "below" : "1.0.0-rc.6.1",
                                "severity": "medium",
                                "identifiers": {"CVE": [ "CVE-2013-4170" ] },
                                "info" : [ "https://groups.google.com/forum/#!topic/ember-
security/dokLVwwxAdM" ]
                        },
                        {
                                "below" : "0.9.7.1",
                                "info" : [ "https://github.com/emberjs/ember.js/blob/master/CHANGELOG" ]
                        },
                        {
                                "below" : "0.9.7",
                                "severity": "high",
                                "identifiers": {
                                        "bug": "699",
                                        "summary": "Bound attributes aren't escaped properly"
                                },
                                "info" : [ "https://github.com/emberjs/ember.js/issues/699" ]
                        }
                ],
                "extractors" : {
                        "func"                  : [ "Ember.VERSION" ],
                        "uri"                   : [ "/(?:v)?([0-9][0-9.a-z_-]+)/ember(\\.min)?\\.js" ],
                        "filename"              : [ "ember-([0-9][0-9.a-z_-]+)(\\.min)?\\.js" ],
                        "filecontent"   : [
                                "Project:   Ember -(?:.*\n){9,11}// Version: v([0-9][0-9.a-z_-]+)",
                                "// Version: v([0-9][0-9.a-z_-]+)(.*\n){10,15}(Ember Debug|@module
ember|@class ember)",
                                "Ember.VERSION[ ]?=[ ]?(?:'|\")([0-9][0-9.a-z_-]+)(?:'|\")"
                        ],
                        "hashes"                : {}
                }
        },
        "dojo" : {
                "vulnerabilities" : [
                        {
                                "atOrAbove" : "0.4",
                                "below" : "0.4.4",
                                "severity": "high",
                                "identifiers": {"CVE": ["CVE-2010-2276", "CVE-2010-2272"]},
                                "info" : [ "http://dojotoolkit.org/blog/dojo-security-advisory",
"http://www.cvedetails.com/cve/CVE-2010-2276/", "http://www.cvedetails.com/cve/CVE-2010-2272/" ]
                        },
                        {
                                "atOrAbove" : "1.0",
```

```
                                "below" : "1.0.3",
                                "severity": "high",
                                "identifiers": {"CVE": ["CVE-2010-2276", "CVE-2010-2274", "CVE-2010-
2273"]},
                                "info" : [ "http://dojotoolkit.org/blog/dojo-security-advisory",
"http://www.cvedetails.com/cve/CVE-2010-2276/", "http://www.cvedetails.com/cve/CVE-2010-2274/",
"http://www.cvedetails.com/cve/CVE-2010-2273/" ]
                        },
                        {
                                "atOrAbove" : "1.1",
                                "below" : "1.1.2",
                                "severity": "high",
                                "identifiers": {"CVE": ["CVE-2010-2276", "CVE-2010-2274", "CVE-2010-
2273"]},
                                "info" : [ "http://dojotoolkit.org/blog/dojo-security-advisory",
"http://www.cvedetails.com/cve/CVE-2010-2276/", "http://www.cvedetails.com/cve/CVE-2010-2274/",
"http://www.cvedetails.com/cve/CVE-2010-2273/" ]
                        },
                        {
                                "atOrAbove" : "1.2",
                                "below" : "1.2.4",
                                "severity": "high",
                                "identifiers": {"CVE": ["CVE-2010-2276", "CVE-2010-2274", "CVE-2010-
2273"]},
                                "info" : [ "http://dojotoolkit.org/blog/dojo-security-advisory",
"http://www.cvedetails.com/cve/CVE-2010-2276/", "http://www.cvedetails.com/cve/CVE-2010-2274/",
"http://www.cvedetails.com/cve/CVE-2010-2273/" ]
                        },
                        {
                                "atOrAbove" : "1.3",
                                "below" : "1.3.3",
                                "severity": "high",
                                "identifiers": {"CVE": ["CVE-2010-2276", "CVE-2010-2274", "CVE-2010-
2273"]},
                                "info" : [ "http://dojotoolkit.org/blog/dojo-security-advisory",
"http://www.cvedetails.com/cve/CVE-2010-2276/", "http://www.cvedetails.com/cve/CVE-2010-2274/",
"http://www.cvedetails.com/cve/CVE-2010-2273/" ]
                        },
                        {
                                "atOrAbove" : "1.4",
                                "below" : "1.4.2",
                                "severity": "high",
                                "identifiers": {"CVE": ["CVE-2010-2276", "CVE-2010-2274", "CVE-2010-
2273"]},
                                "info" : [ "http://dojotoolkit.org/blog/dojo-security-advisory",
"http://www.cvedetails.com/cve/CVE-2010-2276/", "http://www.cvedetails.com/cve/CVE-2010-2274/",
"http://www.cvedetails.com/cve/CVE-2010-2273/" ]
                        },
                        {
                                "below" : "1.4.2",
                                "severity": "medium",
                                "identifiers": {"CVE": [ "CVE-2010-2275" ] },
                                "info" : [ "http://www.cvedetails.com/cve/CVE-2010-2275/"]
                        },
                        {
                                "below" : "1.1",
                                "severity": "medium",
                                "identifiers": {"CVE": [ "CVE-2008-6681" ] },
                                "info" : [ "http://www.cvedetails.com/cve/CVE-2008-6681/"]
                        },
                        {
```

```
                                "below" : "1.10.10",
                                "severity": "medium",
                                "identifiers": { "PR" : "307" },
                                "info" : [ "https://github.com/dojo/dojo/pull/307" ,
"https://dojotoolkit.org/blog/dojo-1-14-released"]
                        },
                        {
                                "atOrAbove" : "1.11.0",
                                "below" : "1.11.6",
                                "severity": "medium",
                                "identifiers": { "PR" : "307" },
                                "info" : [ "https://github.com/dojo/dojo/pull/307" ,
"https://dojotoolkit.org/blog/dojo-1-14-released"]
                        },
                        {
                                "atOrAbove" : "1.12.0",
                                "below" : "1.12.4",
                                "severity": "medium",
                                "identifiers": { "PR" : "307" },
                                "info" : [ "https://github.com/dojo/dojo/pull/307" ,
"https://dojotoolkit.org/blog/dojo-1-14-released"]
                        },
                        {
                                "atOrAbove" : "1.13.0",
                                "below" : "1.13.1",
                                "severity": "medium",
                                "identifiers": { "PR" : "307" },
                                "info" : [ "https://github.com/dojo/dojo/pull/307" ,
"https://dojotoolkit.org/blog/dojo-1-14-released"]
                        }
                ],
                "extractors" : {
                        "func"                                  : [ "dojo.version.toString()" ],
                        "uri"                                   : [ "/(?:dojo-)?([0-9][0-9.a-z_-]+)/dojo(\\.min)?\\.js"
],
                        "filename"                              : [ "dojo-([0-9][0-9.a-z_-]+)(\\.min)?\\.js" ],
                        "filecontentreplace" : [ "/dojo.version=\\{major:([0-9]+),minor:([0-9]+),patch:([0-
9]+)/$1.$2.$3/"],
                        "hashes"                        : {
                                "73cdd262799aab850abbe694cd3bfb709ea23627" : "1.4.1",
                                "c8c84eddc732c3cbf370764836a7712f3f873326" : "1.4.0",
                                "d569ce9efb7edaedaec8ca9491aab0c656f7c8f0" : "1.0.0",
                                "ad44e1770895b7fa84aff5a56a0f99b855a83769" : "1.3.2",
                                "8fc10142a06966a8709cd9b8732f7b6db88d0c34" : "1.3.1",
                                "a09b5851a0a3e9d81353745a4663741238ee1b84" : "1.3.0",
                                "2ab48d45abe2f54cdda6ca32193b5ceb2b1bc25d" : "1.2.3",
                                "12208a1e649402e362f528f6aae2c614fc697f8f" : "1.2.0",
                                "72a6a9fbef9fa5a73cd47e49942199147f905206" : "1.1.1"
                        }

                }
        },
        "angularjs" : {
                "bowername": [ "angularjs", "angular.js" ],
                "vulnerabilities" : [
                        {
                                "atOrAbove" : "1.5.0",
                                "below" : "1.6.9",
                                "severity": "low",
                                "identifiers": {
                                        "summary": "XSS through SVG if enableSvg is set"
```

```
                        },
                        "info" : [
"https://github.com/angular/angular.js/blob/master/CHANGELOG.md#169-fiery-basilisk-2018-02-02",
"https://vulnerabledoma.in/ngSanitize1.6.8_bypass.html" ]
                    },
                    {
                        "atOrAbove" : "1.3.0",
                        "below" : "1.5.0-rc2",
                        "severity": "medium",
                        "identifiers": {
                            "summary": "The attribute usemap can be used as a security exploit"
                        },
                        "info" : [
"https://github.com/angular/angular.js/blob/master/CHANGELOG.md#1230-patronal-resurrection-2016-07-21" ]
                    },
                    {
                        "atOrAbove" : "1.0.0",
                        "below" : "1.2.30",
                        "severity": "medium",
                        "identifiers": {
                            "summary": "The attribute usemap can be used as a security exploit"
                        },
                        "info" : [
"https://github.com/angular/angular.js/blob/master/CHANGELOG.md#1230-patronal-resurrection-2016-07-21" ]
                    },
                    {
                        "below" : "1.6.3",
                        "severity": "medium",
                        "identifiers": {
                            "summary": "Universal CSP bypass via add-on in Firefox"
                        },
                        "info" : [ "https://github.com/mozilla/addons-linter/issues/1000#issuecomment-
282083435", "http://pastebin.com/raw/kGrdaypP" ]
                    },
                    {
                        "below" : "1.6.3",
                        "severity": "medium",
                        "identifiers": {
                            "summary": "DOS in $sanitize"
                        },
                        "info" : [
"https://github.com/angular/angular.js/blob/master/CHANGELOG.md" ]
                    },
                    {
                        "below" : "1.6.5",
                        "severity": "low",
                        "identifiers": {
                            "summary": "XSS in $sanitize in Safari/Firefox"
                        },
                        "info" : [
"https://github.com/angular/angular.js/commit/8f31f1ff43b673a24f84422d5c13d6312b2c4d94" ]
                    }
                ],
                "extractors" : {
                    "func"              : [ "angular.version.full" ],
                    "uri"               : [ "/([0-9][0-9.a-z_-]+)/angular(\\.min)?\\.js" ],
                    "filename"          : [ "angular(?:js)?-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                    "filecontent"       : [
                        "/\\*[ \n]+AngularJS v([0-9][0-9.a-z_-]+)",
                        "http://errors\\.angularjs\\.org/([0-9][0-9.a-z_-]+)/"
                    ],
```

```
                        "hashes"            : {}
                }
        },
        "backbone.js" : {
                "bowername": [ "backbonejs", "backbone" ],
                "vulnerabilities" : [
                        {
                                "below" : "0.5.0",
                                "severity": "medium",
                                "identifiers": {
                                        "release": "0.5.0",
                                        "summary": "cross-site scripting vulnerability"
                                },
                                "info" : [ "http://backbonejs.org/#changelog" ]
                        }
                ],
                "extractors" : {
                        "func"                  : [ "Backbone.VERSION" ],
                        "uri"                   : [ "/([0-9][0-9.a-z_-]+)/backbone(\\.min)?\\.js" ],
                        "filename"              : [ "backbone(?:js)?-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "filecontent"       : [ "//[ ]+Backbone.js ([0-9][0-9.a-z_-]+)",
"a=t.Backbone={}}a.VERSION=\"([0-9][0-9.a-z_-]+)\"" ],
                        "hashes"            : {}
                }
        },
        "mustache.js" : {
                "bowername": [ "mustache.js", "mustache" ],
                "vulnerabilities" : [
                        {
                                "below" : "0.3.1",
                                "severity": "high",
                                "identifiers": {
                                        "bug": "112",
                                        "summary": "execution of arbitrary javascript"
                                },
                                "info" : [ "https://github.com/janl/mustache.js/issues/112" ]
                        },
                        {
                                "below" : "2.2.1",
                                "severity": "medium",
                                "identifiers": {
                                        "bug": "pull request 530",
                                        "summary": "weakness in HTML escaping"
                                },
                                "info" : [ "https://github.com/janl/mustache.js/releases/tag/v2.2.1",
"https://github.com/janl/mustache.js/pull/530" ]
                        }
                ],
                "extractors" : {
                        "func"                  : [ "Mustache.version" ],
                        "uri"                   : [ "/([0-9][0-9.a-z_-]+)/mustache(\\.min)?\\.js" ],
                        "filename"              : [ "mustache(?:js)?-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "filecontent"       : [ "name:\"mustache.js\",version:\"([0-9][0-9.a-z_-]+)\"",
                                                                "[^a-z]mustache.version[ ]?=[ ]?(?:'|\")([0-
9][0-9.a-z_-]+)(?:'|\")",
                                                                "exports.name[ ]?=[ ]?\"mustache.js\";[\n
]*exports.version[ ]?=[ ]?(?:'|\")([0-9][0-9.a-z_-]+)(?:'|\");"
                                                                ],
                        "hashes"            : {}
                }
        },
```

```
        "handlebars.js" : {
                "bowername": [ "handlebars", "handlebars.js" ],
                "vulnerabilities" : [
                        {
                                "below" : "1.0.0.beta.3",
                                "severity": "medium",
                                "identifiers": {
                                        "summary": "poorly sanitized input passed to eval()"
                                },
                                "info" : [ "https://github.com/wycats/handlebars.js/pull/68" ]
                        },
                        {
                                "below" : "4.0.0",
                                "severity": "medium",
                                "identifiers": {
                                        "summary": "Quoteless attributes in templates can lead to XSS"
                                },
                                "info" : [ "https://github.com/wycats/handlebars.js/pull/1083" ]
                        }
                ],
                "extractors" : {
                        "func"                  : [ "Handlebars.VERSION" ],
                        "uri"                   : [ "/([0-9][0-9.a-z_-]+)/handlebars(\\.min)?\\.js" ],
                        "filename"              : [ "handlebars(?:js)?-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "filecontent"       : [
                                "Handlebars.VERSION = \"([0-9][0-9.a-z_-]+)\";",
"Handlebars=\\{VERSION:(?:'|\")([0-9][0-9.a-z_-]+)(?:'|\")",
                                "this.Handlebars=\\{\\};[\n\r \t]+\\(function\\([a-z]\\)\\{[a-
z].VERSION=(?:'|\")([0-9][0-9.a-z_-]+)(?:'|\")",
                                "/\\*![\n\r \t]+handlebars v([0-9][0-9.a-z_-]+)"
                        ],
                        "hashes"            : {}
                }
        },
        "easyXDM" : {
                "vulnerabilities" : [
                        {
                                "below" : "2.4.18",
                                "severity": "high",
                                "identifiers": {"CVE": [ "CVE-2013-5212" ] },
                                "info" : [ "http://blog.kotowicz.net/2013/09/exploiting-easyxdm-part-1-not-
usual.html", "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-5212" ]
                        },
                        {
                                "below" : "2.4.19",
                                "severity": "high",
                                "identifiers": {"CVE": [ "CVE-2014-1403" ] },
                                "info" : [ "http://blog.kotowicz.net/2014/01/xssing-with-shakespeare-name-
calling.html", "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1403" ]
                        }
                ],
                "extractors" : {
                        "uri"                   : [ "/(?:easyXDM-)?([0-9][0-9.a-z_-
]+)/easyXDM(\\.min)?\\.js" ],
                        "filename"              : [ "easyXDM-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
                        "filecontent"       : [ " \\* easyXDM\n \\* http://easyxdm.net/(?:\r|\n|.)+version:\"([0-9][0-
9.a-z_-]+)\"",
                                                                        "@class easyXDM(?:.|\r|\n)+@version ([0-
9][0-9.a-z_-]+)(\r|\n)" ],
                        "hashes"            : { "cf266e3bc2da372c4f0d6b2bd87bcbaa24d5a643" : "2.4.6"}
                }
```

```
	},

	"plupload" : {
		"bowername": [ "Plupload", "plupload" ],
		"vulnerabilities" : [
			{
				"below" : "1.5.4",
				"severity": "high",
				"identifiers": {"CVE": [ "CVE-2012-2401" ] },
				"info" : [ "http://www.cvedetails.com/cve/CVE-2012-2401/" ]
			},
			{
				"below" : "1.5.5",
				"severity": "high",
				"identifiers": {"CVE": [ "CVE-2013-0237" ] },
				"info" : [ "http://www.cvedetails.com/cve/CVE-2013-0237/" ]
			},
			{
				"below" : "2.1.9",
				"severity": "medium",
				"identifiers": {"CVE": [ "CVE-2016-4566" ] },
				"info" : [ "https://github.com/moxiecode/plupload/releases" ]
			}
		],
		"extractors" : {
			"func"			: [ "plupload.VERSION" ],
			"uri"			: [ "/([0-9][0-9.a-z_-]+)/plupload(\\.min)?\\.js" ],
			"filename"		: [ "plupload-([0-9][0-9.a-z_-]+)(.min)?\\.js" ],
			"filecontent"	: [ "\\* Plupload - multi-runtime File Uploader(?:\r|\n)+ \\* v([0-9][0-
9.a-z_-]+)",
								"var g=\\{VERSION:\"([0-9][0-9.a-z_-
]+)\",.*;window.plupload=g\\}"
								],
			"hashes"		: {}
		}
	},

	"DOMPurify" : {
		"bowername": [ "dompurify", "DOMPurify" ],
		"vulnerabilities" : [
			{
				"below" : "0.6.1",
				"severity": "medium",
				"identifiers": { },
				"info" : [ "https://github.com/cure53/DOMPurify/releases/tag/0.6.1" ]
			},
			{
				"below" : "0.8.6",
				"severity": "medium",
				"identifiers": { },
				"info" : [ "https://github.com/cure53/DOMPurify/releases/tag/0.8.6" ]
			},
			{
				"below" : "0.8.9",
				"severity": "low",
				"identifiers": { "summary": "safari UXSS" },
				"info" : [ "https://github.com/cure53/DOMPurify/releases/tag/0.8.9",
"https://lists.ruhr-uni-bochum.de/pipermail/dompurify-security/2017-May/000006.html" ]
			},
			{
				"below" : "0.9.0",
```

```
                                "severity": "low",
                                "identifiers": { "summary": "safari UXSS" },
                                "info" : [ "https://github.com/cure53/DOMPurify/releases/tag/0.9.0" ]
                        }
                ],
                "extractors" : {
                        "func"                          : [ "DOMPurify.version" ],
                        "filecontent"       : [
                                "DOMPurify.version = '([0-9][0-9.a-z_-]+)';",
                                "DOMPurify.version=\"([0-9][0-9.a-z_-]+)\"",
                                "DOMPurify=.[^\\r\\n]{10,500}\\.version=\"([0-9][0-9.a-z_-]+)\""
                        ],
                        "hashes"            : {}
                }
        },

        "react" : {
                "vulnerabilities" : [
                        {
                                "atOrAbove" : "0.4.0", "below" : "0.4.2",
                                "severity" : "low",
                                "identifiers" : {
                                        "CVE": [ "CVE-2013-7035" ] ,
                                        "summary":"potential XSS vulnerability can arise when using user data
as a key"
                                },
                                "info": [ "https://facebook.github.io/react/blog/2013/12/18/react-v0.5.2-
v0.4.2.html" ]
                        },
                        {
                                "atOrAbove" : "0.5.0", "below" : "0.5.2",
                                "severity" : "low",
                                "identifiers" : {
                                        "CVE": [ "CVE-2013-7035" ],
                                        "summary":"potential XSS vulnerability can arise when using user data
as a key"
                                },
                                "info": [ "https://facebook.github.io/react/blog/2013/12/18/react-v0.5.2-
v0.4.2.html" ]
                        },
                        {
                                "below" : "0.14.0",
                                "severity" : "low",
                                "identifiers" : { "summary":" including untrusted objects as React children can
result in an XSS security vulnerability" },
                                "info": [ "http://danlec.com/blog/xss-via-a-spoofed-react-element",
"https://facebook.github.io/react/blog/2015/10/07/react-v0.14.html" ]
                        },
                        {
                                "atOrAbove" : "16.0.0", "below" : "16.0.1",
                                "severity" : "low",
                                "identifiers" : {
                                        "CVE": [ "CVE-2018-6341" ],
                                        "summary":"potential XSS vulnerability when the attacker controls an
attribute name"
                                },
                                "info": [ "https://github.com/facebook/react/blob/master/CHANGELOG.md",
"https://reactjs.org/blog/2018/08/01/react-v-16-4-2.html" ]
                        },
                        {
                                "atOrAbove" : "16.1.0", "below" : "16.1.2",
```

```
                                        "severity" : "low",
                                        "identifiers" : {
                                                "CVE": [ "CVE-2018-6341" ],
                                                "summary":"potential XSS vulnerability when the attacker controls an
attribute name"
                                        },
                                        "info": [ "https://github.com/facebook/react/blob/master/CHANGELOG.md",
"https://reactjs.org/blog/2018/08/01/react-v-16-4-2.html" ]
                                },
                                {
                                        "atOrAbove" : "16.2.0", "below" : "16.2.1",
                                        "severity" : "low",
                                        "identifiers" : {
                                                "CVE": [ "CVE-2018-6341" ],
                                                "summary":"potential XSS vulnerability when the attacker controls an
attribute name"
                                        },
                                        "info": [ "https://github.com/facebook/react/blob/master/CHANGELOG.md",
"https://reactjs.org/blog/2018/08/01/react-v-16-4-2.html" ]
                                },
                                {
                                        "atOrAbove" : "16.3.0", "below" : "16.3.3",
                                        "severity" : "low",
                                        "identifiers" : {
                                                "CVE": [ "CVE-2018-6341" ],
                                                "summary":"potential XSS vulnerability when the attacker controls an
attribute name"
                                        },
                                        "info": [ "https://github.com/facebook/react/blob/master/CHANGELOG.md",
"https://reactjs.org/blog/2018/08/01/react-v-16-4-2.html" ]
                                },
                                {
                                        "atOrAbove" : "16.4.0", "below" : "16.4.2",
                                        "severity" : "low",
                                        "identifiers" : {
                                                "CVE": [ "CVE-2018-6341" ],
                                                "summary":"potential XSS vulnerability when the attacker controls an
attribute name"
                                        },
                                        "info": [ "https://github.com/facebook/react/blob/master/CHANGELOG.md",
"https://reactjs.org/blog/2018/08/01/react-v-16-4-2.html" ]
                                }
                        ],
                        "extractors" : {
                                "func" : [
                                        "react.version",
                                        "require('react').version"
                                ],
                                "filecontent" : [
                                        "/\\*\\*\n +\\* React \\(with addons\\) ?v([0-9][0-9.a-z_-]+)",
                                        "/\\*\\*\n +\\* React v([0-9][0-9.a-z_-]+)",
                                        "\"\\./ReactReconciler\":[0-9]+,\"\\./Transaction\":[0-9]+,\"fbjs/lib/invariant\":[0-
9]+\\}\\],[0-9]+:\\[function\\(require,module,exports\\)\\{\"use strict\";module\\.exports=\"([0-9][0-9.a-z_-]+)\"\\}",
                                        "ReactVersion\\.js[\\*! \\\\/\n\r]{0,100}function\\(e,t\\)\\{\"use
strict\";e\\.exports=\"([0-9][0-9.a-z_-]+)\"",
                                        "expected a ReactNode.[\\s\\S]{0,1800}?function\\(e,t\\)\\{\"use
strict\";e\\.exports=\"([0-9][0-9.a-z_-]+)\""
                                ]
                        }
                },
```

```
	"flowplayer" : {
		"vulnerabilities" : [
			{
				"below" : "5.4.3",
				"severity": "medium",
				"identifiers": { "summary" : "XSS vulnerability in Flash fallback" },
				"info" : [ "https://github.com/flowplayer/flowplayer/issues/381" ]
			}
		],
		"extractors" : {
			"uri"                    : [ "flowplayer-([0-9][0-9.a-z_-]+)(\\.min)?\\.js" ],
			"filename"               : [ "flowplayer-([0-9][0-9.a-z_-]+)(\\.min)?\\.js" ]
		}
	},

	"DWR" : {
		"vulnerabilities" : [
			{
				"below" : "1.1.4",
				"severity": "high",
				"identifiers": { "CVE" : [ "CVE-2007-01-09" ] },
				"info" : [ "http://www.cvedetails.com/cve/CVE-2014-5326/",
"http://www.cvedetails.com/cve/CVE-2014-5326/" ]
			},
			{
				"below" : "2.0.11",
				"severity": "medium",
				"identifiers": { "CVE" : ["CVE-2014-5326", "CVE-2014-5325"] },
				"info" : [ "http://www.cvedetails.com/cve/CVE-2014-5326/",
"http://www.cvedetails.com/cve/CVE-2014-5326/" ]
			},
			{
				"above" : "3",
				"below" : "3.0.RC3",
				"severity": "medium",
				"identifiers": { "CVE" : ["CVE-2014-5326", "CVE-2014-5325"] },
				"info" : [ "http://www.cvedetails.com/cve/CVE-2014-5326/",
"http://www.cvedetails.com/cve/CVE-2014-5326/" ]
			}
		],
		"extractors" : {
			"func"                   : [ "dwr.version" ],
			"filecontent"        : [
				" dwr-([0-9][0-9.a-z_-]+).jar"
			]
		}
	},

	"moment.js" : {
		"bowername": [ "moment", "momentjs" ],
		"vulnerabilities" : [
			{
				"below" : "2.11.2",
				"severity": "low",
				"identifiers": { "summary":"reDOS - regular expression denial of service" },
				"info" : [ "https://github.com/moment/moment/issues/2936" ]
			}
		],
		"extractors" : {
			"func"                   : [ "moment.version" ],
			"filecontent"        : [ "//! moment.js(?:[\n\r]+)//! version : ([0-9][0-9.a-z_-]+)" ]
```

```
		}
	},

	"bootstrap": {
		"vulnerabilities" : [
			{
				"below" : "4.3.1",
				"atOrAbove" : "4.0.0",
				"identifiers": {
					"issue" : "28236",
					"summary": "XSS in data-template, data-content and data-title
properties of tooltip/popover",
					"CVE" : ["CVE-2019-8331"]
				},
				"severity" : "high",
				"info" : [ "https://github.com/twbs/bootstrap/issues/28236" ]
			},
			{
				"below" : "3.4.1",
				"identifiers": {
					"issue" : "28236",
					"summary": "XSS in data-template, data-content and data-title
properties of tooltip/popover",
					"CVE" : ["CVE-2019-8331"]
				},
				"severity" : "high",
				"info" : [ "https://github.com/twbs/bootstrap/issues/28236" ]
			},
			{
				"below" : "4.1.2",
				"atOrAbove" : "4.0.0",
				"identifiers": {
					"issue" : "20184",
					"summary": "XSS in data-target property of scrollspy",
					"CVE" : ["CVE-2018-14041"]
				},
				"severity" : "medium",
				"info" : [ "https://github.com/twbs/bootstrap/issues/20184" ]
			},
			{
				"below" : "3.4.0",
				"identifiers": {
					"issue" : "20184",
					"summary": "XSS in data-target property of scrollspy",
					"CVE" : ["CVE-2018-14041"]
				},
				"severity" : "medium",
				"info" : [ "https://github.com/twbs/bootstrap/issues/20184" ]
			},
			{
				"below" : "4.1.2",
				"atOrAbove" : "4.0.0",
				"identifiers": {
					"issue" : "20184",
					"summary": "XSS in collapse data-parent attribute",
					"CVE" : ["CVE-2018-14040"]
				},
				"severity" : "medium",
				"info" : [ "https://github.com/twbs/bootstrap/issues/20184" ]
			},
			{
```

```
                "below" : "3.4.0",
                "identifiers": {
                        "issue" : "20184",
                        "summary": "XSS in collapse data-parent attribute",
                        "CVE" : ["CVE-2018-14040"]
                },
                "severity" : "medium",
                "info" : [ "https://github.com/twbs/bootstrap/issues/20184" ]
        },
        {
                "below" : "4.1.2",
                "atOrAbove" : "4.0.0",
                "identifiers": {
                        "issue" : "20184",
                        "summary": "XSS in data-container property of tooltip",
                        "CVE" : ["CVE-2018-14042"]
                },
                "severity" : "medium",
                "info" : [ "https://github.com/twbs/bootstrap/issues/20184" ]
        },
        {
                "below" : "3.4.0",
                "identifiers": {
                        "issue" : "20184",
                        "summary": "XSS in data-container property of tooltip",
                        "CVE" : ["CVE-2018-14042"]
                },
                "severity" : "medium",
                "info" : [ "https://github.com/twbs/bootstrap/issues/20184" ]
        },
        {
                "below" : "2.1.0",
                "severity": "medium",
                "identifiers": {
                        "summary": "cross-site scripting vulnerability"
                },
                "info" : [ "https://github.com/twbs/bootstrap/pull/3421" ]
        },
        {
                "below" : "4.3.1",
                "atOrAbove" : "4.0.0",
                "identifiers": {
                        "issue" : "28236",
                        "summary": "XSS in data-template, data-content and data-title
properties of tooltip/popover",
                        "CVE" : ["CVE-2019-8331"]
                },
                "severity" : "high",
                "info" : [ "https://github.com/twbs/bootstrap/issues/28236" ]
        },
        {
                "below" : "3.4.1",
                "identifiers": {
                        "issue" : "28236",
                        "summary": "XSS in data-template, data-content and data-title
properties of tooltip/popover",
                        "CVE" : ["CVE-2019-8331"]
                },
                "severity" : "high",
                "info" : [ "https://github.com/twbs/bootstrap/issues/28236" ]
        }
```

```
                ],
                "extractors" : {
                        "uri"                   : [ "/([0-9][0-9.a-z_-]+)/bootstrap(\\.min)?\\.js" ],
                        "filename"              : [ "bootstrap-([0-9][0-9.a-z_-]+)(\\.min)?\\.js" ],
                        "filecontent"   : [
                                                                "/\\*!? Bootstrap v([0-9][0-9.a-z_-]+)",
                                                                "\\* Bootstrap v([0-9][0-9.a-z_-]+)",
                                                                "/\\*! Bootstrap v([0-9][0-9.a-z_-]+)"
                        ],
                        "hashes"        : {}
                }
        },

        "ckeditor" : {
                "vulnerabilities": [
                        {
                                "below" : "4.4.3",
                                "identifiers" : {
                                        "summary" : "XSS"
                                },
                                "severity" : "medium",
                                "info": [ "https://github.com/ckeditor/ckeditor-
dev/blob/master/CHANGES.md#ckeditor-443" ]
                        },
                        {
                                "below" : "4.4.6",
                                "identifiers" : {
                                        "summary" : "XSS"
                                },
                                "severity" : "medium",
                                "info": [ "https://github.com/ckeditor/ckeditor-
dev/blob/master/CHANGES.md#ckeditor-446" ]
                        },
                        {
                                "below" : "4.4.8",
                                "identifiers" : {
                                        "summary" : "XSS"
                                },
                                "severity" : "medium",
                                "info": [ "https://github.com/ckeditor/ckeditor-
dev/blob/master/CHANGES.md#ckeditor-448" ]
                        },
                        {
                                "below" : "4.5.11",
                                "identifiers" : {
                                        "summary" : "XSS"
                                },
                                "severity" : "medium",
                                "info": [ "https://github.com/ckeditor/ckeditor-
dev/blob/master/CHANGES.md#ckeditor-4511" ]
                        },
                        {
                                "below" : "4.9.2",
                                "atOrAbove" : "4.5.11",
                                "identifiers" : {
                                        "summary" : "XSS if the enhanced image plugin is installed"
                                },
                                "severity" : "medium",
                                "info": [ "https://ckeditor.com/blog/CKEditor-4.9.2-with-a-security-patch-
released/", "https://ckeditor.com/cke4/release-notes" ]
                        },
```

```
                {
                        "atOrAbove" : "4.0.0",
                        "below" : "4.11.0",
                        "identifiers" : {
                                "summary" : "XSS vulnerability in the HTML parser"
                        },
                        "severity" : "medium",
                        "info" : [
                                "https://ckeditor.com/blog/CKEditor-4.11-with-emoji-dropdown-and-auto-link-on-typing-released/",
                                "https://snyk.io/vuln/SNYK-JS-CKEDITOR-72618"
                        ]
                }
        ],
        "extractors" : {
                "uri"                   : [ "/([0-9][0-9.a-z_-]+)/ckeditor(\\.min)?\\.js" ],
                "filename"              : [ "ckeditor-([0-9][0-9.a-z_-]+)(\\.min)?\\.js" ],
                "filecontent"   : [
"ckeditor..js.{4,20}=\\{timestamp:\"[^\"]+\",version:\"([0-9][0-9.a-z_-]+)",
                                                                "window.CKEDITOR=function\\(\\)\\{var [a-z]=\\{timestamp:\"[^\"]+\",version:\"([0-9][0-9.a-z_-]+)"
                        ],
                "hashes"        : {},
                "func"                  : [ "CKEDITOR.version" ]
        }
},


"vue" : {
        "vulnerabilities" : [
                {
                        "below" : "2.5.17",
                        "severity" : "medium",
                        "identifiers" : {
                                "summary" : "potential xss in ssr when using v-bind"
                        },
                        "info" : [ "https://github.com/vuejs/vue/releases/tag/v2.5.17" ]
                },
                {
                        "below" : "2.4.3",
                        "severity" : "medium",
                        "identifiers" : {
                                "summary" : "possible xss vector "
                        },
                        "info" : [ "https://github.com/vuejs/vue/releases/tag/v2.4.3" ]
                }
        ],
        "extractors" : {
                "uri"                   : [
                "/vue@([0-9][0-9.a-z_-]+)/dist/vue\\.js"
                ],
                "filename"              : [ "vue-([0-9][0-9.a-z_-]+)(\\.min)?\\.js" ],
                "filecontent"   : [
                        "/\\*!\\n * Vue.js v([0-9][0-9.a-z_-]+)",
                        "Vue.version = '([0-9][0-9.a-z_-]+)';",
                        "'([0-9][0-9.a-z_-]+)'[^\\n]{0,8000}Vue compiler"
                ],
                "func"                  : [ "Vue.version" ]
        }
},
```

```
	"ExtJS" : {
		"vulnerabilities" : [
			{
				"below"      : "6.6.0",
				"atOrAbove"  : "4.0.0",
				"severity"   : "high",
				"identifiers" : {
					"CVE"    : [
						"CVE-2018-8046"
					],
					"summary" : "XSS in Sencha Ext JS 4 to 6 via getTip() method of
Action Columns"
				},
				"info"       : [
					"http://seclists.org/fulldisclosure/2018/Jul/8",
					"https://nvd.nist.gov/vuln/detail/CVE-2018-8046"
				]
			},
			{
				"below"      : "6.0.0",
				"severity"   : "high",
				"identifiers" : {
					"CVE"    : [
						"CVE-2007-2285"
					],
					"summary" : "Directory traversal and arbitrary file read"
				},
				"info"       : [
					"https://www.cvedetails.com/cve/CVE-2007-2285/",
					"https://packetstormsecurity.com/files/132052/extjs-Arbitrary-File-
Read.html",
					"https://www.akawebdesign.com/2018/08/14/should-js-frameworks-
prevent-xss/"
				]
			},
			{
				"below"      : "4.0.0",
				"atOrAbove"  : "3.0.0",
				"severity"   : "high",
				"identifiers" : {
					"CVE"    : [
						"CVE-2010-4207",
						"CVE-2012-5881"
					],
					"summary" : "XSS vulnerability in ExtJS charts.swf"
				},
				"info"       : [
					"https://www.acunetix.com/vulnerabilities/web/extjs-charts-swf-cross-
site-scripting",
					"https://typo3.org/security/advisory/typo3-core-sa-2014-001/",
					"https://www.akawebdesign.com/2018/08/14/should-js-frameworks-
prevent-xss/"
				]
			}
		],
		"extractors" : {
			"uri" : [
				"/extjs/([0-9][0-9.a-z_-]+)/.*\\.js"
			],
			"filename" : [
```

```
                                        "/ext-all-([0-9][0-9.a-z_-]+)(\\.min)?\\.js",
                                        "/ext-all-debug-([0-9][0-9.a-z_-]+)(\\.min)?\\.js",
                                        "/ext-base-([0-9][0-9.a-z_-]+)(\\.min)?\\.js"
                                ],
                                "filecontent" : [
                                        "/*!\n * Ext JS Library ([0-9][0-9.a-z_-]+)"
                                ],
                                "func"     : [
                                        "Ext && Ext.versions && Ext.versions.extjs.version",
                                        "Ext && Ext.version"
                                ]
                        }
                },


                "dont check" : {
                        "extractors" : {
                                "uri" : [
                                        "^http[s]?://(ssl|www).google-analytics.com/ga.js",
                                        "^http[s]?://apis.google.com/js/plusone.js",
                                        "^http[s]?://cdn.cxense.com/cx.js"
                                ]
                        }
                }
}
“encoder.py”
import base64 as b64
import re
def base64(string):
    if re.match(r'^[A-Za-z0-9+\/=]+$', string) and (len(string) % 4) == 0:
        return b64.b64decode(string.encode('utf-8')).decode('utf-8')
    else:
        return b64.b64encode(string.encode('utf-8')).decode('utf-8')
“jsContexter.py”
import re

from core.config import xsschecker
from core.utils import stripper


def jsContexter(script):
    broken = script.split(xsschecker)
    pre = broken[0]
    pre = re.sub(r'(?s)\{.*?\}|(?s)\(.*?\)|(?s)".*?"|(?s)\'.*?\'', '', pre)
    breaker = ''
    num = 0
    for char in pre:
        if char == '{':
            breaker += '}'
        elif char == '(':
            breaker += ';)'
        elif char == '[':
            breaker += ']'
        elif char == '/':
            try:
                if pre[num + 1] == '*':
                    breaker += '/*'
            except IndexError:
                pass
        elif char == '}':
            breaker = stripper(breaker, '}')
```

```
        elif char == ')':
            breaker = stripper(breaker, ')')
        elif breaker == ']':
            breaker = stripper(breaker, ']')
        num += 1
    return breaker[::-1]
“WAFsignatures.json”
{
        "360 Web Application Firewall (360)" : {
                "code" : "493",
                "page" : "/wzws-waf-cgi/",
                "headers" : "X-Powered-By-360wzb"
        },
        "aeSecure" : {
                "code" : "",
                "page" : "aesecure_denied.png",
                "headers" : "aeSecure-code"
        },
        "Airlock (Phion/Ergon)" : {
                "code" : "",
                "page" : "",
                "headers" : "AL[_-]?(SESS|LB)"
        },
        "Anquanbao Web Application Firewall (Anquanbao)" : {
                "code" : "405",
                "page" : "/aqb_cc/error/|hidden_intercept_time",
                "headers" : "X-Powered-By-Anquanbao"
        },
        "Armor Protection (Armor Defense)" : {
                "code" : "",
                "page" : "This request has been blocked by website protection from Armor",
                "headers" : ""
        },
        "Application Security Manager (F5 Networks)" : {
                "code" : "",
                "page" : "The requested URL was rejected\\. Please consult with your administrator\\.",
                "headers" : ""
        },
        "Amazon Web Services Web Application Firewall (Amazon)" : {
                "code" : "403",
                "page" : "",
                "headers" : "\\bAWS"
        },
        "Yunjiasu Web Application Firewall (Baidu)" : {
                "code" : "",
                "page" : "",
                "headers" : "yunjiasu-nginx"
        },
        "Barracuda Web Application Firewall (Barracuda Networks)" : {
                "code" : "",
                "page" : "",
                "headers" : "barra_counter_session=|(|\\b)barracuda_"
        },
        "BIG-IP Application Security Manager (F5 Networks)" : {
                "code" : "",
                "page" : "",
                "headers" : "BigIP|F5\\Z|\\bTS[0-9a-f]+=|X-WA-Info:|X-Cnection:"
        },
        "BinarySEC Web Application Firewall (BinarySEC)" : {
                "code" : "",
                "page" : "",
```

```
		"headers" : "binarysec"
	},
	"BlockDoS" : {
		"code" : "",
		"page" : "",
		"headers" : "BlockDos\\.net"
	},
	"ChinaCache (ChinaCache Networks)" : {
		"code" : "400",
		"page" : "",
		"headers" : "Powered-By-ChinaCache"
	},
	"Cisco ACE XML Gateway (Cisco Systems)" : {
		"code" : "",
		"page" : "",
		"headers" : "ACE XML Gateway"
	},
	"Cloudbric Web Application Firewall (Cloudbric)" : {
		"code" : "",
		"page" : "Cloudbric|Malicious Code Detected",
		"headers" : ""
	},
	"CloudFlare Web Application Firewall (CloudFlare)" : {
		"code" : "",
		"page" : "Attention Required! \\| Cloudflare|CloudFlare Ray ID:|var
CloudFlare=|CLOUDFLARE_ERROR_500S_BOX",
		"headers" : "cloudflare|__cfduid=|cf-ray"
	},
	"CloudFront (Amazon)" : {
		"code" : "",
		"page" : "",
		"headers" : "Error from cloudfront"
	},
	"Comodo Web Application Firewall (Comodo)" : {
		"code" : "",
		"page" : "",
		"headers" : "Protected by COMODO WAF"
	},
	"CrawlProtect (Jean-Denis Brun)" : {
		"code" : "",
		"page" : "This site is protected by CrawlProtect",
		"headers" : ""
	},
	"IBM WebSphere DataPower (IBM)" : {
		"code" : "",
		"page" : "",
		"headers" : "X-Backside-Transport.*?(OK|FAIL)"
	},
	"Deny All Web Application Firewall (DenyAll)" : {
		"code" : "",
		"page" : "Condition Intercepted",
		"headers" : "sessioncookie"
	},
	"Distil Web Application Firewall Security (Distil Networks)" : {
		"code" : "",
		"page" : "",
		"headers" : "x-distil-cs"
	},
	"DOSarrest (DOSarrest Internet Security)" : {
		"code" : "",
		"page" : "",
```

```
                "headers" : "DOSarrest|X-DIS-Request-ID"
        },
        "dotDefender (Applicure Technologies)" : {
                "code" : "",
                "page" : "dotDefender Blocked Your Request|<meta name=\\.description\\. content=\\.Applicure is the leading provider of web application security|Please contact the site administrator, and provide the following Reference ID:EdgeCast Web Application Firewall (Verizon)",
                "headers" : "X-dotDefender-denied"
        },
        "EdgeCast Web Application Firewall (Verizon)" : {
                "code" : "400",
                "page" : "",
                "headers" : "SERVER.*?ECDF"
        },
        "ExpressionEngine (EllisLab)" : {
                "code" : "",
                "page" : "Invalid (GET|POST) Data",
                "headers" : ""
        },
        "FortiWeb Web Application Firewall (Fortinet)" : {
                "code" : "",
                "page" : "\\.fgd_icon|\\.blocked|\\.authenticate",
                "headers" : "FORTIWAFSID=|cookiesession1="
        },
        "Hyperguard Web Application Firewall (art of defence)" : {
                "code" : "",
                "page" : "",
                "headers" : "ODSESSION="
        },
        "Incapsula Web Application Firewall (Incapsula/Imperva)" : {
                "code" : "",
                "page" : "Incapsula incident ID|_Incapsula_Resource|subject=WAF Block Page|If you feel you have been blocked in error, please contact Customer Support",
                "headers" : "X-Iinfo|incap_ses|visid_incap"
        },
        "ISA Server (Microsoft)" : {
                "code" : "",
                "page" : "The server denied the specified Uniform Resource Locator (URL)\\. Contact the server administrator\\.",
                "headers" : ""
        },
        "Jiasule Web Application Firewall (Jiasule)" : {
                "code" : "403",
                "page" : "static\\.jiasule\\.com/static/js/http_error\\.js|notice-jiasule",
                "headers" : "jiasule-WAF|__jsluid=|jsl_tracking"
        },
        "KS-WAF (Knownsec)" : {
                "code" : "",
                "page" : "ks-waf-error\\.png'",
                "headers" : ""
        },
        "KONA Security Solutions (Akamai Technologies)" : {
                "code" : "400|403|501",
                "page" : "",
                "headers" : "AkamaiGHost"
        },
        "ModSecurity: Open Source Web Application Firewall (Trustwave)" : {
                "code" : "",
                "page" : "This error was generated by Mod_Security|One or more things in your request were suspicious|rules of the mod_security module",
                "headers" : "Mod_Security|NOYB"
```

```
	},
	"NAXSI (NBS System)" : {
		"code" : "",
		"page" : "",
		"headers" : "naxsi/waf"
	},
	"NetContinuum Web Application Firewall (NetContinuum/Barracuda Networks)" : {
		"code" : "",
		"page" : "",
		"headers" : "NCI__SessionId="
	},
	"NetScaler (Citrix Systems)" : {
		"code" : "",
		"page" : "",
		"headers" : "ns_af=|citrix_ns_id|NSC_|NS-CACHE"
	},
	"Newdefend Web Application Firewall (Newdefend)" : {
		"code" : "",
		"page" : "",
		"headers" : "newdefend"
	},
	"NSFOCUS Web Application Firewall (NSFOCUS)" : {
		"code" : "",
		"page" : "",
		"headers" : "NSFocus"
	},
	"Palo Alto Firewall (Palo Alto Networks)" : {
		"code" : "",
		"page" : "has been blocked in accordance with company policy",
		"headers" : ""
	},
	"Profense Web Application Firewall (Armorlogic)" : {
		"code" : "",
		"page" : "",
		"headers" : "PLBSID=|Profense"
	},
	"AppWall (Radware)" : {
		"code" : "",
		"page" : "Unauthorized Activity Has Been Detected.*?Case Number:",
		"headers" : "X-SL-CompState"
	},
	"Reblaze Web Application Firewall (Reblaze)" : {
		"code" : "",
		"page" : "",
		"headers" : "rbzid=|Reblaze Secure Web Gateway"
	},
	"ASP.NET RequestValidationMode (Microsoft)" : {
		"code" : "500",
		"page" : "ASP\\.NET has detected data in the request that is potentially dangerous|Request
Validation has detected a potentially dangerous client input value|HttpRequestValidationException",
		"headers" : ""
	},
	"Safe3 Web Application Firewall" : {
		"code" : "",
		"page" : "",
		"headers" : "Safe3"
	},
	"Safedog Web Application Firewall (Safedog)" : {
		"code" : "",
		"page" : "",
		"headers" : "WAF/2\\.0|safedog"
```

```
        },
        "SecureIIS Web Server Security (BeyondTrust)" : {
                "code" : "",
                "page" : "SecureIIS.*?Web Server
Protection|http://www\\.eeye\\.com/SecureIIS/|\\?subject=[^>]*SecureIIS Error",
                "headers" : ""
        },
        "SEnginx (Neusoft Corporation)" : {
                "code" : "",
                "page" : "SENGINX-ROBOT-MITIGATION",
                "headers" : ""
        },
        "TrueShield Web Application Firewall (SiteLock)" : {
                "code" : "",
                "page" : "SiteLock Incident ID|sitelock-site-verification|sitelock_shield_logo",
                "headers" : ""
        },
        "SonicWALL (Dell)" : {
                "code" : "",
                "page" : "This request is blocked by the SonicWALL|#shd|#nsa_banner|Web Site
Blocked.*?\\bnsa_banner",
                "headers" : "SonicWALL"
        },
        "UTM Web Protection (Sophos)" : {
                "code" : "",
                "page" : "Powered by UTM Web Protection",
                "headers" : ""
        },
        "Stingray Application Firewall (Riverbed / Brocade)" : {
                "code" : "403|500",
                "page" : "",
                "headers" : "X-Mapping-"
        },
        "CloudProxy WebSite Firewall (Sucuri)" : {
                "code" : "403",
                "page" : "Access Denied.*?Sucuri Website Firewall|Sucuri WebSite Firewall.*?Access
Denied|Questions\\?.*?cloudproxy@sucuri\\.net",
                "headers" : "Sucuri/Cloudproxy|X-Sucuri"
        },
        "Tencent Cloud Web Application Firewall (Tencent Cloud Computing)" : {
                "code" : "405",
                "page" : "waf\\.tencent-cloud\\.com",
                "headers" : ""
        },
        "Teros/Citrix Application Firewall Enterprise (Teros/Citrix Systems)" : {
                "code" : "",
                "page" : "",
                "headers" : "st8(id|_wat|_wlf)"
        },
        "TrafficShield (F5 Networks)" : {
                "code" : "",
                "page" : "",
                "headers" : "F5-TrafficShield|ASINFO="
        },
        "UrlScan (Microsoft)" : {
                "code" : "",
                "page" : "Rejected-By-UrlScan",
                "headers" : "Rejected-By-UrlScan"
        },
        "USP Secure Entry Server (United Security Providers)" : {
                "code" : "",
```

```
                "page" : "",
                "headers" : "Secure Entry Server"
        },
        "Varnish FireWall (OWASP)" : {
                "code" : "",
                "page" : "Request rejected by xVarnish-WAF|\\bXID: \\d+",
                "headers" : ""
        },
        "Wallarm Web Application Firewall (Wallarm)" : {
                "code" : "",
                "page" : "",
                "headers" : "nginx-wallarm"
        },
        "WatchGuard (WatchGuard Technologies)" : {
                "code" : "",
                "page" : "",
                "headers" : "WatchGuard"
        },
        "WebKnight Application Firewall (AQTRONIX)" : {
                "code" : "999",
                "page" : "WebKnight Application Firewall Alert|AQTRONIX WebKnight",
                "headers" : "WebKnight"
        },
        "Wordfence (Feedjit)" : {
                "code" : "",
                "page" : "This response was generated by Wordfence|Your access to this site has been limited",
                "headers" : ""
        },
        "Zenedge Web Application Firewall (Zenedge)" : {
                "code" : "",
                "page" : "zenedge/assets/",
                "headers" : "ZENEDGE"
        },
        "Yundun Web Application Firewall (Yundun)" : {
                "code" : "",
                "page" : "",
                "headers" : "YUNDUN"
        },
        "Yunsuo Web Application Firewall (Yunsuo)" : {
                "code" : "",
                "page" : "<img class=.yunsuologo.",
                "headers" : "yunsuo_session"
        }
}
```

Блок 3. Лістинг необхідних бібліотек для роботи програмного додатку

“Photon.py”

```
import re
import concurrent.futures
from urllib.parse import urlparse

from core.dom import dom
from core.log import setup_logger
from core.utils import getUrl, getParams
from core.requester import requester
from core.zetanize import zetanize
from plugins.retireJs import retireJs

logger = setup_logger(__name__)


def photon(seedUrl, headers, level, threadCount, delay, timeout, skipDOM):
```

```
forms = []
processed = set()
storage = set()
schema = urlparse(seedUrl).scheme
host = urlparse(seedUrl).netloc
main_url = schema + '://' + host
storage.add(seedUrl)
checkedDOMs = []

def rec(target):
    processed.add(target)
    printableTarget = '/'.join(target.split('/')[3:])
    if len(printableTarget) > 40:
        printableTarget = printableTarget[-40:]
    else:
        printableTarget = (printableTarget + (' ' * (40 - len(printableTarget))))
    logger.run('Parsing %s\r' % printableTarget)
    url = getUrl(target, True)
    params = getParams(target, '', True)
    if '=' in target:
        inps = []
        for name, value in params.items():
            inps.append({'name': name, 'value': value})
        forms.append({0: {'action': url, 'method': 'get', 'inputs': inps}})
    response = requester(url, params, headers, True, delay, timeout).text
    retireJs(url, response)
    if not skipDOM:
        highlighted = dom(response)
        clean_highlighted = ''.join([re.sub(r'^\d+\s+', '', line) for line in highlighted])
        if highlighted and clean_highlighted not in checkedDOMs:
            checkedDOMs.append(clean_highlighted)
            logger.good('Potentially vulnerable objects found at %s' % url)
            logger.red_line(level='good')
            for line in highlighted:
                logger.no_format(line, level='good')
            logger.red_line(level='good')
    forms.append(zetanize(response))
    matches = re.findall(r'<[aA].*href=["\']{0,1}(.*?)["\']', response)
    for link in matches:
        link = link.split('#')[0]
        if link.endswith(('.pdf', '.png', '.jpg', '.jpeg', '.xls', '.xml', '.docx', '.doc')):
            pass
        else:
            if link[:4] == 'http':
                if link.startswith(main_url):
                    storage.add(link)
            elif link[:2] == '//':
                if link.split('/')[2].startswith(host):
                    storage.add(schema + link)
            elif link[:1] == '/':
                storage.add(main_url + link)
            else:
                storage.add(main_url + '/' + link)
try:
    for x in range(level):
        urls = storage - processed
        threadpool = concurrent.futures.ThreadPoolExecutor(
            max_workers=threadCount)
        futures = (threadpool.submit(rec, url) for url in urls)
        for i in concurrent.futures.as_completed(futures):
            pass
```

```
    except KeyboardInterrupt:
        return [forms, processed]
    return [forms, processed]
```

“Promt.py”

```
import os
import tempfile

from core.config import defaultEditor
from core.colors import white, yellow
from core.log import setup_logger

logger = setup_logger(__name__)


def prompt(default=None):
    editor = os.environ.get('EDITOR', defaultEditor)
    with tempfile.NamedTemporaryFile(mode='r+') as tmpfile:
        if default:
            tmpfile.write(default)
            tmpfile.flush()
        child_pid = os.fork()
        is_child = child_pid == 0
        if is_child:
            try:
                os.execvp(editor, [editor, tmpfile.name])
            except FileNotFoundError:
                logger.error('You don\'t have either a default $EDITOR \
value defined nor \'nano\' text editor')
                logger.info('Execute %s`export EDITOR=/pat/to/your/editor` \
%sthen run XSStrike again.\n\n' % (yellow,white))
                exit(1)
        else:
            os.waitpid(child_pid, 0)
            tmpfile.seek(0)
            return tmpfile.read().strip()
```

“Log.py”

```
import logging
from .colors import *

__all__ = ['setup_logger', 'console_log_level', 'file_log_level', 'log_file']

console_log_level = 'INFO'
file_log_level = None
log_file = 'xsstrike.log'

"""
Default Logging Levels
CRITICAL = 50
ERROR = 40
WARNING = 30
INFO = 20
DEBUG = 10
"""

VULN_LEVEL_NUM = 60
RUN_LEVEL_NUM = 22
GOOD_LEVEL_NUM = 25


logging.addLevelName(VULN_LEVEL_NUM, 'VULN')
logging.addLevelName(RUN_LEVEL_NUM, 'RUN')
```

```
logging.addLevelName(GOOD_LEVEL_NUM, 'GOOD')


def _vuln(self, msg, *args, **kwargs):
    if self.isEnabledFor(VULN_LEVEL_NUM):
        self._log(VULN_LEVEL_NUM, msg, args, **kwargs)


def _run(self, msg, *args, **kwargs):
    if self.isEnabledFor(RUN_LEVEL_NUM):
        self._log(RUN_LEVEL_NUM, msg, args, **kwargs)


def _good(self, msg, *args, **kwargs):
    if self.isEnabledFor(GOOD_LEVEL_NUM):
        self._log(GOOD_LEVEL_NUM, msg, args, **kwargs)


logging.Logger.vuln = _vuln
logging.Logger.run = _run
logging.Logger.good = _good


log_config = {
    'DEBUG': {
        'value': logging.DEBUG,
        'prefix': '{}[*]{}'.format(yellow, end),
    },
    'INFO': {
        'value': logging.INFO,
        'prefix': info,
    },
    'RUN': {
        'value': RUN_LEVEL_NUM,
        'prefix': run,
    },
    'GOOD': {
        'value': GOOD_LEVEL_NUM,
        'prefix': good,
    },
    'WARNING': {
        'value': logging.WARNING,
        'prefix': '[!!]'.format(yellow, end),
    },
    'ERROR': {
        'value': logging.ERROR,
        'prefix': bad,
    },
    'CRITICAL': {
        'value': logging.CRITICAL,
        'prefix': '{}[--]{}'.format(red, end),
    },
    'VULN': {
        'value': VULN_LEVEL_NUM,
        'prefix': '{}[++]{}'.format(green, red),
    }
}


class CustomFormatter(logging.Formatter):
    def format(self, record):
```

```
        msg = super().format(record)
        if record.levelname in log_config.keys():
            msg = '%s %s %s' % (log_config[record.levelname]['prefix'], msg, end)
        return msg


class CustomStreamHandler(logging.StreamHandler):
    default_terminator = '\n'

    def emit(self, record):
        """
        Overrides emit method to temporally update terminator character in case last log record character is '\r'
        :param record:
        :return:
        """
        if record.msg.endswith('\r'):
            self.terminator = '\r'
            super().emit(record)
            self.terminator = self.default_terminator
        else:
            super().emit(record)


def _switch_to_no_format_loggers(self):
    self.removeHandler(self.console_handler)
    self.addHandler(self.no_format_console_handler)
    if hasattr(self, 'file_handler') and hasattr(self, 'no_format_file_handler'):
        self.removeHandler(self.file_handler)
        self.addHandler(self.no_format_file_handler)


def _switch_to_default_loggers(self):
    self.removeHandler(self.no_format_console_handler)
    self.addHandler(self.console_handler)
    if hasattr(self, 'file_handler') and hasattr(self, 'no_format_file_handler'):
        self.removeHandler(self.no_format_file_handler)
        self.addHandler(self.file_handler)


def _get_level_and_log(self, msg, level):
    if level.upper() in log_config.keys():
        log_method = getattr(self, level.lower())
        log_method(msg)
    else:
        self.info(msg)


def log_red_line(self, amount=60, level='INFO'):
    _switch_to_no_format_loggers(self)
    _get_level_and_log(self, red + ('-' * amount) + end, level)
    _switch_to_default_loggers(self)


def log_no_format(self, msg='', level='INFO'):
    _switch_to_no_format_loggers(self)
    _get_level_and_log(self, msg, level)
    _switch_to_default_loggers(self)


def log_debug_json(self, msg='', data={}):
    if self.isEnabledFor(logging.DEBUG):
```

```
        if isinstance(data, dict):
            import json
            try:
                self.debug('{} {}'.format(msg, json.dumps(data, indent=2)))
            except TypeError:
                self.debug('{} {}'.format(msg, data))
        else:
            self.debug('{} {}'.format(msg, data))


def setup_logger(name='xsstrike'):
    from types import MethodType
    logger = logging.getLogger(name)
    logger.setLevel(logging.DEBUG)
    console_handler = CustomStreamHandler(sys.stdout)
    console_handler.setLevel(log_config[console_log_level]['value'])
    console_handler.setFormatter(CustomFormatter('%(message)s'))
    logger.addHandler(console_handler)
    no_format_console_handler = CustomStreamHandler(sys.stdout)
    no_format_console_handler.setLevel((log_config[console_log_level]['value']))
    no_format_console_handler.setFormatter(logging.Formatter(fmt=''))
    logger.console_handler = console_handler
    logger.no_format_console_handler = no_format_console_handler

    if file_log_level:
        detailed_formatter = logging.Formatter('%(asctime)s %(name)s - %(levelname)s - %(message)s')
        file_handler = logging.FileHandler(log_file)
        file_handler.setLevel(log_config[file_log_level]['value'])
        file_handler.setFormatter(detailed_formatter)
        logger.addHandler(file_handler)
        no_format_file_handler = logging.FileHandler(log_file)
        no_format_file_handler.setLevel(log_config[file_log_level]['value'])
        no_format_file_handler.setFormatter(logging.Formatter(fmt=''))
        logger.file_handler = file_handler
        logger.no_format_file_handler = no_format_file_handler

    logger.red_line = MethodType(log_red_line, logger)
    logger.no_format = MethodType(log_no_format, logger)
    logger.debug_json = MethodType(log_debug_json, logger)
    return logger
```