Розроблення та застосування експлойтів з подальшою інтеграцією в ботнет

УДК 004.738.5

Ростислав Ткачук, Артур Ткаченко, Роман Андріїв

Львівський державний університет безпеки життєдіяльності, rlvtk@ukr.net, tkachenko9720@ukr.net, roman.andriiv@icloud.com

Веб-браузери завжди будуть пріоритетною ціллю для пошуку вразливостей, оскільки ними користується кожна людина. Це означає, що якщо буде знайдена нова вразливість у веб-браузері – це стане ключем доступу до акаунтів дуже багатьох людей та великих потужностей обчислювальних ресурсів. Це дозволить у свою чергу проводити надзвичайно великі та складні кібероперації на зразок розвиненої сталої загрози [1].

Розроблення експлойта для використання ресурсів будь-якого веб-браузера

Наше дослідження зосереджено на пошуку нетипової поведінки та обходу різного роду обмежень в елементі, що міститься в кожному веб-браузері, а саме – JavaScript [2]. Цей експлойт в даній роботі фундаментальний, оскільки без нього всі решта не будуть працювати. Він працює на межі можливостей JavaScript виконуючи функціонал, який не передбачений навіть у відомих бібліотеках, при тому в коді якого не міститься жодної додаткової бібліотеки та використовує такі особливості JavaScript, як спільні змінні між скриптами, у дуже нестандартний спосіб, а саме використовує їх для отримання команд та скриптів із командного сервера, які в подальшому він динамічно розгортає на веб-сторінці. Тобто достатньо запустити один невеликий скрипт, а він може динамічно розгорнутись умовно у 20-30 та більше скриптів.

Це свого роду завантажувальна програма, тіло якої потрібно будь-яким способом запустити на веб-сторінці і яку добре використовувати в атаках типу XSS.

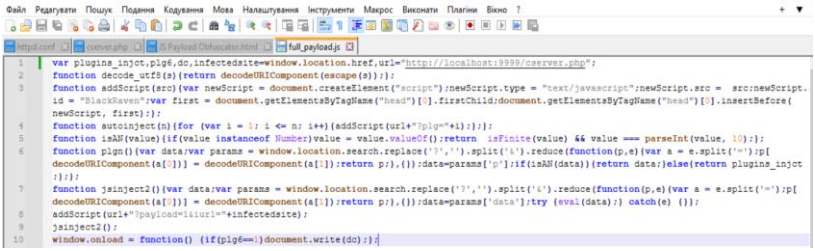

Рис. 1. Експлойт 1

Експлойт для записування всіх натиснутих клавіш на веб-сторінці з відправкою на віддалений сервер

Даний експлойт буде завантажуватися за допомогою попереднього. Він очікує коли нажата будь-яка клавіша і надсилає дані на командний сервер. Для цього було використано нетипове використання функції addEventListener, яка реєструє будь-яке натискання на клавішу, та повертає її код, який потім передається у функцію keylogger. Ця функція в залежності від числа повертає натиснуту клавішу та відправляє її на командний сервер. В ній здійснюється обхід безпеки веб-браузера,

так як у JavaScript закрита функціональність надсилати http/https get запити на сторонні веб-ресурсів, то для обходу цього обмеження було використано нетипове застосування html тега img, який призначений для завантажування зображень, а командний сервер зчитує ці дані і записує їх у файл [3].

Експлойт для проведення DDOS-атак за допомогою веб-браузер та IP-адреси відвідувача веб-сторінки

Цей експлойт аналогічно попереднім використовує спільні змінні між скриптами та спосіб обходу обмежень безпеки веб-браузера на заборону відправки http/https get запитів за допомогою HTML-тега img та iframe.

Після завантаження на веб-сторінці експлойт виконує запуск веб-браузера за допомогою експлойта-завантажувача, після чого він зчитує змінну ціль, яка вказана на командному сервері, зчитує обраний режим для атаки (звичайна http/https get атака, або вдосконалена http/https get атака що викликає помилку 404) та починає атакувати обрану ціль.

Експлойт має функцію створення контейнера із вигаданим HTML-тегом «zlo», в якому знаходиться тег img, або iframe в залежності від обраного типу атаки. Також має функцію самовидалення. Це зроблено для того, щоб у користувача не переповнювалась пам'ять через велику кількість створених тегів та не зависав веб-браузер. Далі ці функції за допомогою SetInterval постійно виконуються у веб-браузері з інтервалом у 0,5 секунди.

Командний сервер та інтеграція корисних навантажень і експлойтів

Для командного сервера потрібен сервер із підтримкою PHP. Тому було обрано Apache 2.4 сервер та PHP 8 версії із стандартними налаштуваннями. Їх потрібно завантажити та встановити на сервер. В нашому випадку це сервер під управлінням Windows Sever 2019 Base у хмарі Amazon (рис. 2).

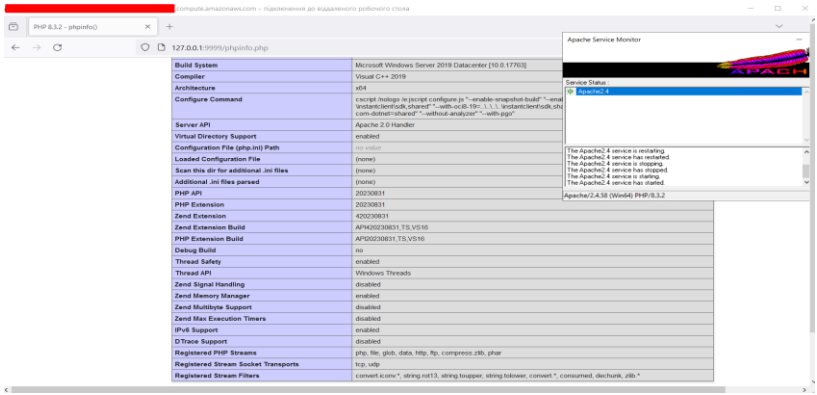

Рис. 2. Встановлений Apache2.4+PHP8

Тепер у папці htdocs необхідно створити конфігураційний файл командного сервера під назвою inject.js. Цей файл є файлом командного сервера, де відбувається управління ботнетом. За допомогою цього файлу можна увімкнути/вимкнути будь-який експлойт, вказати час обновлення команд, створити власний вміст веб-сторінки та зробити дефейс, вказати метод для ddos-атаки, додати власні скрипти.

Далі створюємо php-файл під назвою call.php, який записує дані з кейлогера, інформацію про відвідувача веб-сайту, вести логи, повертає код корисного навантаження та експлойтів завантажувачу, що запускається у веб-браузері, які відвідавши заражену веб-сторінку самі того не знаючи тимчасово стають учасниками ботнету. Цей файл є основою командного сервера.

Далі потрібно встановити проксі-ретранслятор командного сервера, який можна завантажувати на будь-який скомпроментований php-сайт.

Цей файл працює посередником між сервером та клієнтом, через який проходять всі дані і він є невід'ємною частиною командного сервера. Тепер, щоб інтегрувати багатоплатформне корисне навантаження на python достатньо просто створити файл code.txt у папці командного сервера [4] (рис. 3).

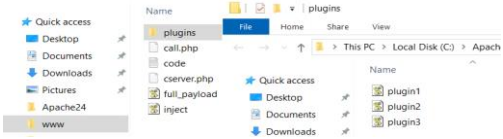

Рис. 3. Вигляд готового командного сервера

Далі необхідно перемістити експлойти у папку plugins під іменами plugin1,2,3 і код із файлу full_payload.js потрібно обробити обфускатором, щоб заплутати код. На цьому інтеграцію експлойтів та багатоплатформного корисного навантаження в ботнет є завершеною.

В роботі розглянута можливість написання простих експлойтів, їхній принцип роботи з подальшим їхнім об'єднанням в ботнет, а саме – створення простого командного сервера ботнета, який керує написаними експлойтами. Це необхідно для того, щоб розуміти як відбувається процес створення вище зазначених речей та як їм протидіяти, оскільки в процесі розробки завжди допускаються помилки, які потім можуть бути використані для проведення атак.

1. Беспалько О., Ткачук Р.Л., Андріїв Р.Р. Дослідження методів захисту інформації веб-сайтів на основі моделей розподілення доступу та моніторингу ідентифікаторів користувача Зб. тез доп. VI Всеукр. наук.-практ. конф. молодих учених, студентів і курсантів "Інформаційна безпека та інформаційні технології". Львів : ЛДУБЖД, 2023.
2. Експлойт [Електронний ресурс] – Режим доступу до ресурсу:https://ukeywaf.com/baza/eksplojt-shho-cze-take/
3. Metasploit Framework [Електронний ресурс] – Режим доступу до ресурсу: https://www.varonis.com/blog/what-is-metasploit
4. Відкритий вихідний код корисного навантаження на python [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/Xehos/Python-Socket-shell/tree/main