Було розраховано загальну кількість альтернатив (3840), з поміж яких обрано 20 найбільш актуальних для використання. Відносну важливість переваг було визначено за шкалою Сааті. Використавши матриці переваг було розраховано вектори локальних переваг та вектор глобальних пріоритетів, який дав загальну оцінку переваги для кожної альтернативи.

В результаті проведених розрахунків було визначено, що найоптимальнішим буде такий варіант реалізації: призначенням СЗІ є захист інформації на рівні глобальної мережі, метод виявлення контенту здійснюється за допомогою Exact Data Matching (EDM), застосовуються мережеві протоколи виду TCP, ICMP, UDP, здійснення захисту інформації IT-процесів інфраструктури підприємства відбувається ззовні, перевага надається підтримці операційної системи Windows, використовується активний контроль переміщення даних та робота системи в режимі реально часу.

Обраний варіант має такі переваги: захист на найбільш вразливому рівні глобальної мережі, можливість перевірки контролю валідності та ідентифікаторів користувачів баз даних через локальну мережу всередині підприємства задля посилення захисту конфіденційної інформації, підтримка найбільш популярної в корпоративних середовищах операційної системи, здійснення активного контролю переміщення даних та можливість захисту і внесення змін в режимі реального часу.

Отже, як можна помітити, вибраний варіант реалізації є одночасно ефективним та економічно доступним навіть для малих компаній, що і було метою даного дослідження.

Список використаних джерел

1. Ткач Ю. М. Прогнозування та моделювання. Методичні вказівки до виконання курсової роботи [Електронний ресурс] / Юлія Миколаївна Ткач. – 2017. – Режим доступу до ресурсу: https://eln.stu.cn.ua/pluginfile.php/84135/mod_resource/content/1/%D0%9F%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%9A%D0%A1%D0%A0%20%D0%BF%D0%B4%D1%84.pdf.
2. Грабовецкий Б. Є. Основи економічного прогнозування [Електронний ресурс] / Б. Є. Грабовецкий // ВФ ТАНГ. – 2000. – Режим доступу до ресурсу: https://buklib.net/books/32652/
3. Запобігання витоку інформації [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BE%D1%82%D0%B2%D1%80%D0%B0%D1%89%D0%B5%D0%BD%D0%B8%D0%B5_%D1%83%D1%82%D0%B5%D1%87%D0%B5%D0%BA_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8
4. Symantec Data Loss Prevention [Електронний ресурс] – Режим доступу до ресурсу: https://www.open-vision.ru/catalog/security/dlp-system/symantec-data-loss-prevention/
5. Баранов О. С. Огляд Symantec Data Loss Prevention 12.5 [Електронний ресурс] / Олексій Сергійович Баранов. – 2014. – Режим доступу до ресурсу: https://www.anti-malware.ru/reviews/Symantec_Data_Loss_Prevention_12_5
6. Вступ до кібернетики – Санкт-Петербург, 1959. – 432 с. – (Іноземна література)
7. Ієрархічна модель даних [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Ієрархічна_модель_даних

УДК 004.056.5

ОСНОВНІ ПРИНЦИПИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ТА СТАБІЛЬНОСТІ ВЕБ-СЕРВЕРІВ

Клименок В. О., студ. гр. КБ-161,
Ткач Ю. М., д.пед.н., доцент
Національний університет «Чернігівська політехніка»

За останнє десятиліття ми спостерігали неабиякий ріст мобільних додатків. І хоча нативні програми все ще популярні, зростає тенденція до веб-додатків, які працюють безпосередньо у вашому браузері. То що ж викликає ця привабливість до веб-додатків? Немає однієї основної відповіді, але є багато факторів:

1. Простіше створити веб-додатки за допомогою одного стека (HTML / CSS / JS).
2. Веб-додатки не вимагають окремого магазину додатків.

3. Більшість веб-додатків запускаються у всіх браузерах на всіх пристроях від настільних ПК до смартфонів.

Саме висока популярність веб-додатків може привернути увагу зловмисників. Неправильне налаштування веб-серверу та безпечного зв’язку може призвести до потенційної втрати або компрометації конфіденційної інформації.

Серед головних вразливостей веб-серверу та веб-додатку можна виділити наступні:

1. SQL Injection.

Ін'єкція - це вразливість безпеки, яка дозволяє зловмиснику змінювати серверні оператори SQL за допомогою маніпулювання наданими користувачем даними.Ін'єкція відбувається, коли користувальницькі дані надсилаються перекладачеві як частина команди чи запиту і вводять перекладача у виконання непередбачених команд та надають доступ до несанкціонованих даних. Команда SQL, яка виконується веб-додатком, також може відкрити резервну базу даних.

Наслідки

- Зловмисник може вводити шкідливий вміст у вразливі поля.
- Чутливі дані, такі як імена користувачів, паролі тощо, можна прочитати з бази даних.
- Дані бази даних можуть бути змінені.
- Операції з адміністрування можуть бути виконані у базі даних.

2. Cross Site Scripting.

Перехресні сценарії також коротко відомі як XSS. Сценарії націлювання на вразливості XSS, вбудовані в сторінку, яка виконується на стороні клієнта, тобто в браузері користувача, а не на стороні сервера. Ці недоліки можуть виникнути, коли програма приймає недовірені дані та надсилає їх у веб-браузер без належної перевірки.

Зловмисники можуть використовувати XSS для виконання шкідливих скриптів на користувачах у цьому випадку браузерах жертв. Оскільки браузер не може знати, справжній сценарій чи ні, сценарій буде виконаний, і зловмисник може викрасти сеансові файли cookie, знекровити веб-сайти або перенаправити користувача на небажані та зловмисні веб-сайти.

XSS - це атака, яка дозволяє зловмиснику виконувати скрипти в браузері жертви.

Наслідки:

Використовуючи цю вразливість безпеки, зловмисник може вводити сценарії в програму, може викрадати файли cookie сеансу, знекровити веб-сайти та запустити зловмисне програмне забезпечення на машинах жертви.

3. Неправильна конфігурація.

Конфігурація безпеки повинна бути визначена та розгорнута для програми, рамок, сервера додатків, веб-сервера, сервера баз даних та платформи. Якщо вони правильно налаштовані, зловмисник може мати несанкціонований доступ до конфіденційних даних або функцій. Іноді такі недоліки призводять до повної компрометації системи. Постійне оновлення програмного забезпечення – це також значно підвищує рівень безпеки системи.

Використання незахищеного протоколу передачі даних HTTP.

HTTP – це протокол передачі гіпертексту. HTTP пропонує набір правил і стандартів, які регулюють спосіб передачі будь-якої інформації у всесвітній мережі. HTTP забезпечує стандартні правила для спілкування веб-браузерів та серверів. Це мережевий протокол додаткового рівня, який будується поверх TCP. HTTP використовує структурований текст Hypertext, який встановлює логічний зв'язок між вузлами, що містять текст. Він також відомий як "протокол без стану", оскільки кожна команда виконується окремо, без використання посилання на попередню команду запуску.

Основні недоліки:

–Конфіденційності немає, оскільки кожен може бачити вміст.

–Цілісність даних є великою проблемою, оскільки хтось може змінити вміст. Ось чому протокол HTTP є небезпечним методом, оскільки не використовуються методи шифрування.

– Кожен, хто перехопить запит, може отримати ім’я користувача та пароль.

Установки та конфігурації операційної системи за замовчуванням не завжди є захищеними. У типовій установці за замовчуванням встановлено багато мережевих служб, які не використовуються в конфігурації веб-сервера, такі як: послуги віддаленого реєстру, сервер друку, RAS тощо. Чим більше служб, що працюють в операційній системі, тим більше буде портів залишати відкритими, тим самим залишаючи більш відкритими двері для зловмисних користувачів. Саме тому, усі непотрібні сервіси потрібно відключити, щоб наступного разу, коли сервер перезавантажиться, вони не запускалися автоматично. Вимкнення непотрібних служб також збільшить продуктивність сервера, звільнивши деякі ресурси.

Якщо потрібен віддалений доступ, потрібно переконатися, що віддалене з'єднання забезпечено належним чином, використовуючи протоколи тунелювання та шифрування. Використання токенів безпеки є хорошою практикою. Віддалений доступ також повинен бути обмежений певним числом IP-адрес і лише певними обліковими записами. Також дуже важливо не використовувати публічні комп’ютери чи публічні мережі для віддаленого доступу до корпоративних серверів, як в Інтернет-кафе чи загальнодоступних бездротових мережах. Якщо це з’єднання по протоколу SSH – слід змінити стандартний порт 22 на інший.

Рівні доступу на файлові та мережеві послуги відіграють важливу роль у захисті веб-серверів. Якщо двигун веб-сервера порушений через програмне забезпечення мережевого сервісу, зловмисник може використовувати обліковий запис, в якому працює мережева служба, для виконання завдань, таких як виконання певних файлів (скриптів). Тому дуже важливо завжди призначати найменші привілеї, необхідні для роботи певної мережевої послуги, наприклад програмного забезпечення веб-сервера. Також дуже важливо призначити мінімальним привілеям анонімному користувачеві, який необхідний для доступу до веб-сайту, файлів веб-додатків, а також резервних даних та баз даних.

Усі журнали, присутні на веб-сервері, в ідеалі повинні зберігатися в відокремленій області. Усі журнали мережевих служб, журнали доступу до веб-сайтів, журнали серверів баз даних (наприклад, Postgres, MySQL, Oracle) та журнали операційної системи слід регулярно контролювати та перевіряти. Завжди слід шукати дивні записи журналу. Файли журналів, як правило, дають всю інформацію про спробу нападу та навіть про вдалу атаку, але в більшості випадків вони ігноруються. Якщо хтось помітить дивну активність із журналів, це слід негайно посилити, щоб проблему можна було дослідити, щоб зрозуміти, що відбувається. Для автоматизації процесу аналізу лог-файлів можна використовувати програмне забезпечення Fail2ban. Коли fail2ban налаштований на моніторинг журналів сервісу, він переглядає фільтр, налаштований конкретно для цієї служби. Фільтр призначений для виявлення збоїв аутентифікації для цієї конкретної послуги за допомогою використання складних регулярних виразів. Дія за замовчуванням - заборона правопорушного хоста / IP-адреси шляхом зміни правил брандмауера iptables. Можна розширити цю дію, щоб також надіслати електронний лист адміністратору з повідомленням про зловмисника або з рядками журналу, які викликали дію. Ви також можете змінити ціль дії, щоб вона блокувала на іншому рівні, ніж iptables. За бажанням, дія може бути удосконалена.

Також слід слідкувати за використанням ресурсів на сервері. За допомогою таких систем, як Nagios, Prometheus або Zabbix, можна слідкувати за використанням процесорних ресурсів, постійної та тимчасової пам’яті, навантаження на диск тощо.

На сьогодні інформацію та поради щодо програмного забезпечення та операційної системи, що використовується, можна вільно знайти в Інтернеті. Дуже важливо бути інформованим та дізнаватися про нові напади та інструменти, читаючи журнали, пов’язані із безпекою та передплачуючи розсилки, форуми чи будь-який інший тип спільноти.

Тож, задля забезпечення максимальної стабільності та безпеки веб-серверу, потрібно тримати систему оновленою, слідкувати за ресурсами, відключати непотрібні сервіси, розмежовувати права доступу користувачів та програмного забезпечення на сервері. Слід окремо зберігати базу даних, робити резервні копії та репліки. Також, оскільки веб-додатками користуються клієнти, слід використовувати тільки захищений протокол – HTTPS з

використанням останніх версій TLS бібліотек. Перевірити на неправильну, або неактуальну конфігурацію веб-серверу можна за допомогою спеціальних аналізаторів, наприклад – ssllabs.com.

Список використаних джерел

1. Anicas M. 5 Common Server Setups For Your Web Application [Електронний ресурс] / Mitchell Anicas – Режим доступу до ресурсу: https://www.digitalocean.com/community/tutorials/5-common-server-setups-for-your-web-application.
2. Banga S. Web Application Architecture: Definition, Models, Types, and More [Електронний ресурс] / Swapnil Banga – Режим доступу до ресурсу: https://hackr.io/blog/web-application-architecture-definition-models-types-and-more.
3. Web Application Architecture: Definition, Models, Types, and More [Електронний ресурс] – Режим доступу до ресурсу https://www.commonplaces.com/blog/6-common-website-security-vulnerabilities/
4. Ellingwood J. How Fail2Ban Works to Protect Services on a Linux Server [Електронний ресурс] / Justin Ellingwood – Режим доступу до ресурсу: https://www.digitalocean.com/community/tutorials/how-fail2ban-works-to-protect-services-on-a-linux-server

---

УДК 004.056.5

ОПТИМІЗАЦІЯ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ АНОНІМНОСТІ ПЕРЕДАЧІ ТРАФІКУ НА БАЗІ TOR МЕРЕЖІ

Чулінда О. С., студ. гр. КБ-161
Науковий керівник: **Базилевич В. М.,** к.е.н., доцент
Національний університет «Чернігівська політехніка»

На даний момент в мережі анонімність є великою проблемою тому, що дані про відвідини інтернет-ресурсів можна зібрати з локальних пристроїв. Тому зараз все більш користувачів використовую програми для анонімності в мережі. У зв'язку з цим в мережі з'являється більше нових програм VPN, але не всі VPN дійсно надають анонімність. Деякі працюють з різними умовами такими як анонімність лише HTTP трафіку, або данні користувачів попадають в інтернет через неуважність розробників.

Для надання анонімності необхідно щоб трафік системи потрапляв на проміжні вузли для надійного маскування, чим більше вузлів тим краще маскування в мережі. Для цього можна використовувати Tor – це метод анонімного зв'язку, використовуваний для анонімної передачі мережевого трафіку. Повідомлення шифрується і відправляється на декілька вузлів. Кожен маршрутизатор розшифровує один шар повідомлення і передає на наступний маршрутизатор.

За допомогою Tor можна відключити вразливі сервіси, такі як скрипти та Flash. Через мережу Tor сайти не зможуть використовувати історію переглядів для створення таргетованої реклами. Tor забезпечить анонімність для обходу заблокованих сайтів. Але є і мінуси у використанні Tor, мережа Tor працює дуже повільно і не підходить для сервісів які вимагають швидкого підключення. Вихідні вузли можуть бути розкриті. Дані можуть бути вкрадені якщо не використовувати протокол HTTPS. Якщо використовувати Tor браузер, то захищений буде лише трафік браузера. На більшості вузлів Tor заборонено завантажувати торрент.

Tor славиться своїми функціями анонімізації інтернет-трафіку, однак можливості цієї мережі обмежені, а сама вона вразлива перед атаками й витоками даних. Вихідний вузол може створити хто завгодно в тому числі та зловмисник. Tor не використовує наскрізне шифрування, тому при переході на сайти які не використовують HTTPS власник вихідного вузла зможе дізнатися що і куди відправляє користувач. Якщо ви відправляєте через Tor конфіденційні дані або авторизуєтесь на сайті, то власник вихідного вузла отримує доступ до вашої інформації. VPN-сервіси, у свою чергу, використовують наскрізне шифрування, завдяки чому ваші дані на 100% захищені від хакерів і шпигунів.