реалізації своїх ідей. Таким чином є можливість створити сайти з унікальним функціоналом, дизайном або надійністю систем безпеки.

Отже, онлайн-сервіси активно розвиваються та можуть надавати широкі можливості користувачам. Їх широка поширеність корелює з різноманіттям способів реалізації. Такі сайти може створювати навіть людина без досвіду, а для спеціалістів можливості вже майже не обмежені.

Список літератури

1. Techopedia. What is an online service?. Techopedia.
URL: https://www.techopedia.com/definition/3248/online-service (дата звернення:18.04.2022).
2. Що таке on-line сервіси?. Avada Media. URL: https://avada-media.ua/ua/services/on-line-servisy/ (дата звернення: 18.04.2022).
3. Як створити свій сайт?. ukraine.com.ua.
URL: https://www.ukraine.com.ua/uk/blog/marketing/kak-sozdat-svoj-sajt-podrobnij-gajd.html (дата звернення: 18.04.2022).
3. CMS або конструктор сайтів?. Onehostplanet.
URL: https://onehostplanet.ua/news/konstruktor-chi-cms-vibiraemo-instrumenti-dlya-stvorennya-saytu (дата звернення: 18.04.2022).

УДК 004:56

Чайковський П.А., студент 3 курсу, СО Бакалавр
Нескородєва Т. В., д-р. техн. наук, доцент, завідувач кафедри інформаційних технологій

МЕТОДИ ТА ПРАКТИКИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ КОРИСТУВАЦЬКИХ ДАНИХ У ОНЛАЙН СЕРВІСАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Під час створення сучасних електронних сервісів завжди варто мати на увазі аспект безпеки, це може бути безпека персональних даних користувачів сервісу, даних адміністрації, доступ до важливих ресурсів, таких як база приватних ключів та токенів авторизації.

Сучасні сайти часто менш схильні злому ніж раніше через покращення культури розробки, але проблема все ще залишається у полі зору, так як у будь-якого сервісу є точки доступу, що можна скомпрометувати, навіть якщо не sql-ін`єкцією, як це було 10-15 років тому, то як мінімум непродуманий сайт може перевантажувати сервер, що може допомогти недоброчесним агентам.

Даний матеріал має у собі ціль показати та пояснити яких саме принципів притримувались розробники електронного сервісу “Індивідуальна Освітня Траєкторія” задля забезпечення захисту персональних даних студентів, адміністрації сервісу, та інших аспектів.

Буде описано низка вразливостей, прикладів, принципів таких як HTTPS, SQLi та інше.

OWASP TOP 10

OWASP (Open Web Application Security Project) – це нон-профіт організація що працює над покращенням безпеки програмного забезпечення [1].

OWASP Top 10 – це 10 вразливостей, що є найпопулярнішими у певний рік. Нижче можна подивитись вразливості що були найпоширенішими у 2017 – 2021 роках.

Як бачимо, є речі що залишаються незмінними, а саме ін`єкції коду, невірна конфігурація безпеки, застарілі компоненти ПЗ, та небезпечна десеріалізація даних (помилки в цільності даних та ПЗ у 2021).

Варто виділити тут небезпечну десеаріалізацію даних. Якщо коротко: серіалізація – це приведення даних (або коду програми) до певного вигляду, що дозволить ці дані зручно пересувати по мережі, або між різними частинами певної однієї або групи програм.

Рисунок 1. Топ 10 вразливостей з 2017 по 2021 роки

В свою чергу десеріалізація – це приведення даних у такий вигляд, із яким може працювати середовище виконання (мова, скрипт, VM, т.д.). Часто це доволі безпечно, якщо мова йде про прості текстові дані, які можна передати наприклад у JSON. Але стає доволі небезпечним, коли це стосується коду. Щось подібне сталось у 2021 із популярним логгером для Java Log4j [2].

Діру в log4j часто порівнюють із вразливістю Heartbleed 2014 року. Тоді в бібліотеці OpenSSL була виявлена помилка, що призводить до витоку даних із пам'яті. Вона дозволяла віддалено витягувати секрети, і також принесла адміністраторам систем масу проблем. Складно було не так виявити всі вразливі інсталяції, як визначити, які дані теоретично могли бути викрадені. У Heartbleed і безіменної (крім креативу в MS Paint) вразливості в log4j є ще один загальний момент: це відкриті проекти, що розробляються в умовах перманентної нестачі ресурсів.

SQL ІН`ЄКЦІЇ

Сучасні веб-програми мають зараз досить складну структуру. Разом з цим для зберігання інформації стали активно використовуватися бази даних на основі мови SQL.

При зверненні до будь-якої сторінки, веб-програма формує спеціальний SQL-запит, який запитує у базі даних відповідний запис. Результат запиту повертається до веб-програми, яка відображає його на сторінці браузера для користувача. Використовувати подібну схему взаємодії зручно, але водночас зростає ризик появи SQL-ін'єкцій. Суть їх роботи полягає у впровадженні власного коду SQL-запит у базу даних з метою отримати додаткову інформацію від неї.[4]

Найчастіше таку вразливість можна використати у випадку, якщо розробник виконує свої SQL запити таким чином.

Розробники сервісу «Індивідуальна освітня траєкторія» постійно перевіряють дані, що передаються із зовнішніх джерел, таких як веб-застосунок, щоб попередити більшість ін`єкцій та інших вразливостей.

Далі необхідно вдатись до HTTPS та чому це важливо для більшості електронних сервісів, що мають доступ до користувацьких персональних даних.

HTTP ТА HTTPS

HTTP (HyperText Transfer Protocol) – протокол передачі даних, наразі використовується для передачі будь-яких даних через мережу.

HTTPS (s for Secure) – розширення протоколу HTTP в цілях підвищення безпеки трафіку, що передається через транспортні механізми SSL та TLS.

SSL (secure sockets layer) – криптопротокол для безпечного зв’язку. Працює використовуючи асимертичну та симетричну криптографію для надійності.

Let’s Encrypt CA – одна із організацій, що видає довірені сертифікати [5]. Сертифікат має бути випущений довіреною організацією, щоб користувач розумів, що комунікація із сайтом є безпечною та зашифрованою, а домене ім`я є вірним (щоб було зрозуміло що якщо ми перейшли на privat24.com, і сертифікат є вірним – то ми спокійно можемо ним користуватись).

Держава може заборонити доступ у зовнішню мережу для тих у кого державний сертифікат не встановлений, а у ти у кого він є – держава зможе розшифрувати трафік, можливо не весь, але все ж. Тобто свого роду man-in-the-middle attack, але на державному рівні. Але, якщо держава створить root сертифікат, вона зможе розшифрувати весь трафік що виходить із систем, в яких його встановлено. І таке вже було, детальніше в статті «Kazakhstan man-in-the-middle attack», де влада Казахстану у 15-му (і ще кілька раз) бажала просунити на території своєї країни сертифікати від **Qaznet Trust Network,** свого особистого CA, але дані сертифікати було заблоковано у браузерах Safari, Chrome, Firefox. Таким чином можна скомпрометувати користувацькі дані від паролів та логінів – до кредитних карт та ключів від крипто-гаманців.

XSS – CROSS-SITE SCRIPTING

XSS (Cross Site Scripting — «міжсайтовий скриптинг») — тип вразливості інтерактивних інформаційних систем у вебі. XSS виникає, коли на сторінки, які були згенеровані сервером, з якоїсь причини потрапляють користувацькі скрипти. Специфіка подібних атак полягає в тому, що замість безпосередньої атаки сервера зловмисники використовують вразливий сервер для атаки на користувача [6].

Довгий час програмісти не приділяли їм належної уваги, вважаючи їх безпечними. Однак ця думка помилкова: на сторінці або в HTTP-Cookie можуть бути досить вразливі дані (наприклад, ідентифікатор сесії адміністратора). На популярному сайті скрипт може влаштувати DoS-атаку.

Еван Джонсон, інженер компанії «CloudFlare» дослідив вебсайти з переліку «Топ мільйон» сайтів за відвідуваністю, обчислений компанією Alexa. Він виявив близько тисячі сайтів з цього переліку, налаштування вебсервера яких надають зловмисникові можливість атакувати користувача із допомогою міжсайтового скриптингу. Він помітив, що вебсервери вразливих сайтів повертають такий саме HTTP-заголовок Access-Control-Allow-Origin, як вказано в HTTP-заголовку Origin в запиті клієнта. На думку інженера, це створює умови, завдяки яким зловмисник може обійти обмеження правила «Єдиного походження» (англ. Same Origin Policy).

Висновок: Таким чином, ідентифікувавши популярні вразливості та виділивши певні практики щодо захисту персональних даних, можна забезпечити безпеку свого сервісу просто превентивно зробивши дослідження застосованих технологій та коду сервісу.

Список літературних джерел

1. *https://owasp.org/www-project-top-ten/*
2. *https://cyberint.com/blog/research/log4j-incident-update/*
3. *https://twitter.com/FiloSottile/status/1469441487175880711*
4. *https://www.acunetix.com/websitesecurity/sql-injection/*
5. *https://letsencrypt.org/*
6. *https://web.archive.org/web/20160224085509/https://thestack.com/security/2016/02/23/cross-site-scripting-enabled-on-1000-major-sites-including-financial-sites/*

УДК 004.9

Човган Д.С., студент 1 курсу СО Магістр спеціальності 122 «Комп'ютерні науки»

ДОСЛІДЖЕННЯ ГОЛОСОВОГО АСИСТЕНТА В АВТОМОБІЛЯХ
CARS VOICE ASSISTANT RESEARCH

Донецький національний університет імені Василя Стуса, м. Вінниця

Метою даної роботи є – дослідження роботи голосових асистентів які використовуються у автомобілях.

Важливу роль у прийняті рішення досліджень голосових асистентів які використовуються у автомобілях, стала їх активна розробка і розробка голосових асистентів у цілому.

Свою популярність голосові асистенти набували роками і беруть свій початок із далекого 2010 року. Але найбільш ріст стався почався у 2016 році, у момент появи Google Assistant. За появи Windows 10, на ринок голосових асистентів вийшла компанія Microsoft із помічником Cortana.