Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно–інформаційних систем і програмної інженерії
(назва факультету)
Комп’ютерних наук
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи)

Магістр
(освітній ступінь (освітньо-кваліфікаційний рівень))

на тему: Дослідження поширених вразливостей веб-сайтів та методики їх усунення

Виконав: студент (ка) 6 курсу, групи САм-61
спеціальності (напряму підготовки) 124
Системний аналіз
(шифр і назва спеціальності (напряму підготовки))

		Шутко Б.Ю.
	(підпис)	(прізвище та ініціали)
Керівник		Назаревич О.Б.
	(підпис)	(прізвище та ініціали)
Нормоконтроль		Мацюк О.В.
	(підпис)	(прізвище та ініціали)
Рецензент		Лупенко С.А.
	(підпис)	(прізвище та ініціали)

м. Тернопіль – 2019

АНОТАЦІЯ

Дослідження поширених вразливостей веб-сайтів та методики їх усунення // Дипломна робота ОР «Магістр» // Шутко Богдан Юрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група САм-61 // - Тернопіль, 2019 // с. – , рис. – , табл. – , кресл. – , додат. – , бібліогр. – .

Ключові слова: АУТЕНТИФІКАЦІЯ, ВРАЗЛИВОСТІ, ВЕБ-САЙТ, ВЕБ-ДОДАТОК, АВТОРИЗАЦІЯ, МЕТОДИКА, ЗАХИСТ, ПОШИРЕНИЙ, СКАНЕР ВРАЗЛИВОСТЕЙ.

Дипломна робота присвячена питанням дослідження систем захисту веб-сайтів.

В першому розділі проведено аналіз вразливостей веб-серверів, зокрема звернуто увагу на аутентифікацію, авторизацію, атаки на клієнтів, атаки типу виконання коду, розголошення інформації, логічні атаки. В другому розділі наведено статистику вразливостей веб-додатків. В третьому розділі описаний захист веб-сайтів та веб-проектів. В четвертому розділі наведено аналіз поширених вразливостей веб-сайтів та створено методику їх усунення. В п'ятому розділі описано практичне використання проведених досліджень, а саме описано процедуру модифікації під власні потреби сканера вразливостей Nikto-online.

Об'єкт дослідження – інформаційний портал ТНТУ.

Предмет дослідження – захищеність веб-сайтів ТНТУ.

Мета дослідження – розробка система підвищення рівня безпеки веб-сайтів ТНТУ.

Основні результати – Проведено аналіз статистики вразливостей веб-додатків; проаналізовано сканери вразливостей веб-застосувань; модифіковано

сканер Nikto-Online; розроблено рекомендації щодо підвищення захищеності веб-сайтів.

ANNOTATION

Study of websites common vulnerabilities and techniques of their removal // Thesis educational degree "Master" // Shutko Bohdan // Ternopil Ivan Puluj national technical university, Department of Computer Information Systems and Software Engineering, Department of Computer Science, a group SAm- 61 // Ternopil – 2019 // Page – , Fig. - , Table - , Draws - .

Web servers are constantly exposed to many variety of hazards. And the most serious threat to these hackers and viruses. The first can gain access to confidential information available on the server, hack sites and replace its contents, as well as bring down the server using distributed attack (DDoS-attack). Viruses are, infecting Web servers, turn themselves in nursery infection. In addition, they significantly slow down performance, and take Internet channel. At first glance it seems that these threats on the basis of very different from each other. But it is not quite true. It turns out many viruses, especially Internet worms use to spread vulnerabilities in software. And hackers also prefer to use an attack aimed at well-known "hole" in the software. And there is absolutely nothing strange. Using the vulnerability, they both get very easy access to a remote computer even if the latter is well protected.

Virtually any program is vulnerability. And the source code of its bigger, the more it can find different "holes". Vulnerabilities explained very easily. Man - not a machine, it may be wrong. There is even a special rate program, which indicates how many bugs could allow an expert in writing a certain number of lines of code. Also, do not forget that great software says not one person but a group. And often there are errors in layout modules created by different programmers. In addition, vulnerabilities are not always determined by the quality of writing software.

When it comes to vulnerability of web servers, the vast majority of people at once mentions the "holes" in their software. This applies to most applications servers such as Apache, Microsoft Internet Information Server, etc. And there is absolutely

nothing strange. Still, this software is quite extensive and complex, so that "holes" in it necessarily. Also, do not forget that modern Web server can not imagine not many extra features such as no support for programming languages such as Perl, PHP, etc., and no database management systems. This becomes possible due to installation of additional web server software. Yet it can also contain their vulnerability.

Today on sites dedicated to information security, there are always reports of the discovery of new vulnerabilities in software, web servers. This process involved as experts on data protection, and hackers. At last the detection of a new "holes" can be silent about it and try to use it for their own purposes. But often the opposite. Hacker tries to tell you about a new vulnerability to everyone, including software developers.

The main feature of industrial vulnerabilities is their commitment to certain versions of software. The fact that the "holes" often not the entire line of web servers, but only in some of their releases. Also worth noting that the popular way or the other software, the more for it find new vulnerabilities. And it does not depend on the quality of writing software. Just study it more professional, so that the probability of detection and "holes" relatively large.

Protect yourself from the considered type of software vulnerabilities can be only one way - the timely installation of all manufacturers developed updates and patches. The fact that software developers regularly spread on the official site update for its products. If there are critical security "holes" patch released quickly. If the newly found vulnerabilities are theoretical rather than real threat, as they accumulate issued cumulative updates. Vulnerabilities can occur due to incorrect software setup web server.

Perhaps it is no secret that the security of any thing depends on how you use it. The same can be said about the web server. A lot depends on how its configured software. Generally, most web servers have a fairly large set of parameters related to virtually all aspects of their activities. Thus, security is largely dependent on the administrators involved in their care. But do not forget that the administrators - people. This means that they through their carelessness, lack of training or even for

any reason may be wrong. These errors can open the way to the web server a hacker or virus.

From incorrect setting can help plant patches. Indeed, when updating its software configuration does not change. This means that the vulnerability in the system after installing the security patch is likely to remain. Thus, the main threat considered as "holes" is the difficulty of detection. So the only way to really secure protection against these vulnerabilities - the use of special security scanners. These programs with special protection methods examine web servers and all are potentially dangerous places. Scripts web site may also contain vulnerabilities.

The modern Web server and related software are often a kind of database to run programs written by himself. It is, of course, the scripts that run on most modern sites. The fact that most web programming languages is the server. This means that scripts written for them, are performed directly on the server and the user's computer (in this case - Visitor) leaves only the results of their work.

And then it lies a very serious danger. The fact that the scripts for sites not always worked really good specialists. Many web projects using distributed free program or software of their own writing. Naturally, it also may contain vulnerabilities. And some of them can be very serious, that could allow an attacker to gain unauthorized access to the server. And keep in mind that some scripts are executed with elevated privileges. So what vulnerabilities they may be a good tool for hackers.

Identify the "holes" in the scripts you can use security scanners. So that each Web server really cares about the security of your site should periodically review it. And the word "periodically" in the previous sentence was not accidental. The fact that hackers are constantly inventing new ways to remote attacks. In addition, the constantly revealed new "holes" in the original software, which can in combination with scripts that were previously considered safe, be a real threat.

The research is to develop a system to enhance safety websites TNTU.

Under the goal to solve the following problem:

1. Analysis of vulnerabilities Web servers;

2. Analysis of statistics vulnerabilities Web applications;

3. Analysis of the common vulnerabilities of Web sites;

4. Analysis of crawlers vulnerabilities Web applications;

5. Rationale for the adaptation and vulnerability scanner to work in the Ukrainian segment of Internet.

6. Development of methods to reduce the vulnerability of websites.

The first section analyzes vulnerabilities Web servers, in particular referred to the authentication, authorization, attacks on customers, such as attack code execution, information disclosure, logical attack. The second section provides statistics vulnerabilities Web applications. In the third section describes the protection of sites and web projects. In the fourth section, the analysis of common vulnerabilities sites and established methods to address them. In the fifth chapter describes the practical use of the research and describes how to modify it to your needs vulnerabilities scanner Nikto-online.

Object of research - information portal TNTU.

The subject of research - security websites TNTU.

Key Findings – The analysis of statistics web application vulnerabilities, vulnerabilities scanners analyzed Web applications; modified scanner Nikto-Online; designed to maximize security of Web sites.

Newest of the studies was the use of vulnerabilities scanner Nikto-Online in conjunction with the use of methods of increasing the security of web information portal TNTU.

Keywords: AUTHENTICATION VULNERABILITY WEBSITE, WEB APPLICATION, AUTHORIZATION, METHODS, PROTECTION, HARBOR, SCANNER VULNERABILITY.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ASP – active server pages (активні сторінки сервера);

CCK – content construction kit (набір конструкцій контенту);

CGI – common gateway snterface (універсальний інтерфейс шлюзів);

CSS – cascading style sheets (каскадні таблиці стилів);

DBI – data base interface (інтерфейс баз даних);

DoS – denial of service (відмова в обслуговуванні);

DOS – disk operating system (дискова операційна система);

FLV – flash video (флеш відео);

FTP – file transfer protocol (протокол передачі файлів);

HTML – hyper text markup language (мова розмітки гіпертекстових документів);

HTTP – hyper text transfer protocol (протокол передачі даних);

IP – internet protocol (Інтернет протокол);

LDAP – lightweight directory access protocol (полегшений протокол доступу до папок);

MIME – multipurpose internet mail extensions (комплексні розширення для інтернет–пошти);

PDF – portable document format;

PERL – practical extraction and report language (високорівнева, інтерпретована, динамічна мова програмування загального призначення);

PHP – hypertext preprocessor (гіпертекстовий препроцесор);

SMP – symmetric multiprocessing (симетричне мультипроцесування);

SSI – server side include (включення на стороні сервера);

URI – universal resource identification (універсальна форма адресації інформаційних ресурсів);

URL – uniform resource locator (уніфікований вказівник ресурсу);

usability – зручність використання;

VBScript – visual beginners all–purpose symbolic instruction code script (візуальний символічний універсальний командний код для початківців);

WMT –WebMoney transfer (переказ коштів WebMoney);

WWW – world wide web (всесвітня павутина);

XHTML – extensible hypertext markup language (розширювана мова розмітки гіпертексту);

XML – extensible markup language (розширювана мова розмітки);

БД – база даних;

ЕОМ – електронна обчислювальна машина;

ІС – інформаційна система;

ОС – операційна система;

ПЕОМ – персональна електронна обчислювальна машина;

ПК – персональний комп’ютер;

СУБД – система управління базами даних.

ЗМІСТ

ВСТУП 14
1 ВРАЗЛИВОСТІ ВЕБ-СЕРВЕРІВ 18
1.1 Аутентифікація (Authentication) 18
1.1.1 Підбір (Brute Force) 19
1.1.2 Недостатня аутентифікація (Insufficient Authentication) 20
1.1.3 Небезпечне відновлення паролів (Weak Password Recovery Validation) 20
1.2 Авторизація (Authorization) 21
1.2.1 Передбачуване значення ідентифікатора сесії (Credential/Session Prediction) 21
1.2.2 Недостатня авторизація (Insufficient Authorization) 22
1.2.3 Відсутність таймауту сесії (Insufficient Session Expiration) 22
1.2.4 Фіксація сесії (Session Fixation) 23
1.3 Атаки на клієнтів (Client-side Attacks) 24
1.3.1 Підміна вмісту (Content Spoofing) 24
1.3.2 Міжсайтове виконання сценаріїв (Cross-site Scripting, XSS) 25
1.3.3 Розщеплення HTTP-запиту (HTTP Response Splitting) 26
1.4 Виконання коду (Command Execution) 28
1.4.1 Переповнення буфера (Buffer Overflow) 28
1.4.2 Атака на функції форматування рядків (Format String Attack) 29
1.4.3 Впровадження операторів LDAP (LDAP Injection) 29
1.4.4 Виконання команд ОС (OS Commanding) 30
1.4.5 Впровадження операторів SQL (SQL Injection) 30
1.4.6 Впровадження серверних розширень (SSI Injection) 31
1.4.7 Впровадження операторів XPath (XPath Injection) 31
1.5 Розголошення інформації (Information Disclosure) 32
1.5.1 Індексування директорій (Directory Indexing) 32
1.5.2 Ідентифікація додатків (Web Server/Application Fingerprinting) 33

1.5.3 Витік інформації (Information Leakage) 34
1.5.4 Зворотний шлях в директоріях (Path Traversal)..... 35
1.5.5 Передбачуване розміщення ресурсів (Predictable Resource Location) 36
1.6 Логічні атаки (Logical Attacks)..... 37
1.6.1 Зловживання функціональними можливостями (Abuse of Functionality) 37
1.6.2 Відмова в обслуговуванні (Denial of Service)..... 38
1.6.3 Недостатня протидія автоматизації (Insufficient Anti-automation) 39
1.6.4 Недостатня перевірка процесу (Insufficient Process Validation)..... 39
1.7 Висновки до першого розділу..... 40
4 ПРАКТИЧНЕ ВИКОРИСТАННЯ ПРОВЕДЕНИХ ДОСЛІДЖЕНЬ 67
4.1 Класифікація сканерів вразливості веб–застосувань..... 67
4.2 Сканер визначення небезпек Nikto-online..... 67
4.2.1 Робота сканера Nikto-Online..... 68
4.2.2 Інтерфейс сканера 69
4.2.3 Результати сканування 71
4.3 Висновки до четвертого розділу..... 73
5 СПЕЦІАЛЬНА ЧАСТИНА..... 74
5.1 Сканер Shadow Security 74
5.2 Acunetix Web Vulnerability Scanner 75
5.3 Сканер Nessus 79
5.4 Сканер вразливостей XSpider 7 79
5.5 Висновки до шостого розділу 83
6 ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ..... 84
6.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР 84
6.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи... 85
6.3 Розрахунок матеріальних витрат 88
6.4 Розрахунок витрат на електроенергію 89
6.5 Розрахунок суми амортизаційних відрахувань 89

6.6 Обчислення накладних витрат 90
6.7 Складання кошторису витрат та визначення собівартості НДР 91
6.8 Розрахунок ціни науково-дослідної роботи 92
6.9 Визначення економічної ефективності і терміну окупності капітальних вкладень 93
6.10 Висновки до сьомого розділу 94
7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 96
7.1 Охорона праці 96
7.1.1 Дії роботодавця за результатами атестації робочих місць за умовами праці 96
7.1.2 Організація робочого місця: природне та штучне освітлення 99
7.2 Безпека в надзвичайних ситуаціях 102
7.2.1 Забезпечення захисту працівників суб'єктів господарювання та населення від впливу іонізуючих випромінювань 102
8.2.2 Оцінка стійкості роботи промислового підприємства до впливу вторинних вражаючих факторів 104
8 ЕКОЛОГІЯ 108
8.1 Етапи та техніка збору та опрацювання екологічної інформації 108
8.2 Індексний метод в екології 111
8.3 Висновки до сьомого розділу 114
ВИСНОВКИ 115
ПЕРЕЛІК ПОСИЛАНЬ 117
ДОДАТКИ

ВСТУП

Веб-сервери постійно піддаються безлічі самих різноманітних небезпек. Причому найсерйознішу загрозу становлять для них хакери і віруси. Перші можуть отримати доступ до конфіденційної інформації, розміщеної на сервері, зламати сайти і підмінити їх вміст, а також вивести з ладу сервер за допомогою розподіленої атаки (DDoS-атака). Віруси ж, заражаючи веб-сервери, перетворюють їх самих у розсадник інфекції. Крім того, вони істотно сповільнюють його роботу, а також займають Інтернет-канал. На перший погляд здається, що ці загрози за принципом роботи дуже сильно відрізняються один від одного. Але насправді це не зовсім так. Виявляється, багато вірусів, особливо Інтернет-черв'яки, використовують для поширення уразливості в програмному забезпеченні. Так і хакери теж воліють застосовувати атаки, спрямовані на відомі «дірки» в ПЗ. І в цьому немає абсолютно нічого дивного.

Використовуючи уразливості, і ті й інші одержують досить легкий доступ до віддаленого комп'ютера навіть у тому випадку, якщо останній добре захищений.

Практично в будь-якій програмі є вразливості. І чим її вихідний код більший за об’ємом, тим більше в ній можна знайти різних «дірок». Наявність вразливостей пояснюється дуже легко. Людина – не машина, вона може помилятися. Існує навіть спеціальна норма програмування, в якій зазначено, скільки помилок може допустити фахівець при написанні певного числа рядків коду. Крім того, не можна забувати, що велике ПЗ пише не одна людина, а ціла група. І досить часто помилки виникають при компонуванні модулів, створених різними програмістами. Крім того, наявність вразливостей далеко не завжди визначається якістю написання ПЗ.

Коли мова заходить про вразливість веб-серверів, то переважна більшість людей відразу ж згадує «дірки» в їх програмному забезпеченні. Це відноситься до самих програм-серверів, таким як Apache, Microsoft Internet Information Server і т. д. І в цьому немає абсолютно нічого дивного. Все-таки це

програмне забезпечення досить об'ємне і складне, так що «дірки» в ньому обов'язково є. Крім того, не можна забувати, що сучасний веб-сервер неможливо уявити собі без багатьох додаткових функцій, наприклад без підтримки мов програмування типу Perl, PHP і т. д., а також без систем управління базами даних. Все це стає можливим завдяки установці на веб-сервер додаткового програмного забезпечення. І все воно теж може містити свої вразливості.

Сьогодні на сайтах, присвячених інформаційній безпеці, постійно з'являються повідомлення про виявлення нових вразливостей в програмному забезпеченні веб-серверів. У цьому процесі беруть участь як фахівці із захисту даних, так і хакери. Причому останні при виявленні нової «діри» можуть замовчати про неї і спробувати використовувати її у своїх цілях. Але часто буває навпаки. Хакер намагається розповісти про нову уразливість всім, в тому числі і розробникам ПЗ.

Головною особливістю виробничих вразливостей є їх прихильність до певних версій програмного забезпечення. Справа в тому, що «дірки» часто зустрічаються не у всій лінійці веб-серверів, а тільки в деяких їх релізах. А ще варто відзначити, що чим популярніше те чи інше програмне забезпечення, тим частіше для нього знаходять нові уразливості.

І це залежить не від якості написання ПЗ. Просто його вивчають більше фахівців, так що і ймовірність виявлення «дір» відносно велика.

Захиститися від розглянутого типу вразливостей програмного забезпечення можна тільки одним способом – своєчасним встановленням всіх розроблених виробниками оновлень та патчів. Справа в тому, що розробники ПЗ регулярно викладають на офіційних сайтах оновлення для своїх продуктів. При виявленні критичної для безпеки «дірки» патч випускається швидко. Якщо ж знову знайдені вразливості несуть швидше теоретичну, ніж реальну загрозу, то в міру їх накопичення випускаються кумулятивні оновлення.

Вразливості можуть виникати через некоректне налаштування програмного забезпечення веб-сервера.

Напевно, ні для кого не секрет, що безпека будь-якої речі залежить від того, як її застосовувати. Те ж саме можна сказати і про веб-сервер. Дуже багато залежить від того, як налаштоване його програмне забезпечення. Взагалі, переважна більшість веб-серверів мають досить великий набір параметрів, що стосуються практично всіх аспектів їхньої діяльності. Таким чином, безпека багато в чому залежить від адміністраторів, що займаються їх обслуговуванням. Але не можна забувати, що адміністратори – це люди. А це означає, що вони через свою неуважність, недостатньої кваліфікації чи ще з якихось причин можуть помилятися. І ці помилки можуть відкрити дорогу до веб-сервера хакеру або вірусу.

Від некоректного налаштування не може допомогти установка патчів. І дійсно, при оновленні програмного забезпечення його конфігурація не змінюється. А це означає, що уразливість в системі захисту після інсталяції патча швидше за все залишиться. Таким чином, головною небезпекою розглянутого типу «дірок» є складність їх виявлення. Отже єдиний спосіб дійсно надійного захисту від таких вразливостей – використання спеціальних сканерів безпеки. Ці програми за допомогою спеціальних методів досліджують захист веб-серверів і знаходять всі потенційно небезпечні місця.

Скрипти веб-сайтів теж можуть містити вразливості.

Сучасний веб-сервер і супутнє програмне забезпечення дуже часто служать своєрідною базою для виконання програм, які написані власноруч користувачем. Мова йде, звичайно ж, про скрипти, які працюють на більшості сучасних сайтів. Справа в тому, що більшість мов веб-програмування є серверними. Це означає, що скрипти, написані на них, виконуються прямо на сервері, а на комп'ютер користувача (в даному випадку – відвідувача сайту) відправляються тільки результати їх роботи.

Тут-то і криється досить серйозна небезпека. Справа в тому, що скрипти для сайтів далеко не завжди розробляються дійсно гарними спеціалістами. На багатьох веб-проектах використовуються безкоштовно поширювані програми або ж ПЗ власного написання. Природно, у ньому теж можуть міститися

уразливості. Причому деякі з них можуть бути дуже серйозними, що дозволяють зловмисникам отримати несанкціонований доступ до самого серверу. Причому потрібно враховувати, що деякі скрипти виконуються з підвищеними привілеями. Так що уразливості в них можуть виявитися гарною підмогою для хакерів.

Виявити «дірки» в скриптах можна за допомогою сканерів безпеки. Так що кожен власник веб-сервера, дійсно піклується про безпеку свого сайту, повинен періодично перевіряти його. Причому слово «періодично» в попередньому реченні з'явилося зовсім не випадково. Справа в тому, що хакери постійно вигадують нові способи віддалених атак. Крім того, постійно виявляються нові «дірки» в оригінальному ПЗ, які можуть у поєднанні із скриптами, які раніше вважалися безпечними, являти собою реальну загрозу.

Метою роботи є розробка система підвищення рівня безпеки веб-сайтів ТНТУ.

Згідно поставленої мети потрібно вирішити наступні задачі:

1. Аналіз вразливостей веб-серверів;
2. Аналіз статистики вразливостей веб-додатків;
3. Аналіз поширених вразливостей веб-сайтів;
4. Аналіз сканерів вразливостей веб-застосувань;
5. Обґрунтування вибору та адаптація сканера вразливостей для роботи в українському сегменті Інтернет.
6. Розробка методики зменшення вразливостей веб-сайтів.

Новизною проведених досліджень стало використання сканера вразливостей Nikto-Online в поєднанні із використанням методики збільшення захищеності веб-сайтів інформаційного порталу ТНТУ.

1 ВРАЗЛИВОСТІ ВЕБ-СЕРВЕРІВ

В даному розділі дослідимо теоретичні основи та класифікацію вразливостей і атак. Дана класифікація представляє собою спільну спробу зібрати воєдино і організувати загрози безпеки веб-серверів. Члени Web Application Security Consortium створили даний проект для розробки та популяризації стандартної термінології опису цих проблем. Це дасть можливість розробникам додатків, фахівцям в області безпеки, виробникам програмних продуктів і аудиторам використовувати єдину мову для взаємодії.

У багатьох організаціях веб-додатки використовуються як критично важливі системи, які повинні щодня обслуговувати багатомільйонні транзакції. Однак справжня цінність веб-сайтів повинна оцінюватися на основі потреб кожної організації. Важливість чого-небудь досить важко уявити тільки у вигляді певної суми.

Вразливості у веб-додатках досить давно становили небезпеку для користувачів. Після ідентифікації вразливості для здійснення атаки використовується одна з кількох технік.

Дипломна робота містить компіляцію відомих класів атак, які представляли загрозу для веб-додатків в минулому і представляють зараз. Кожному класу атак присвоєно стандартну назву і описані його ключові особливості. Класи організовані в ієрархічну структуру.

1.1 Аутентифікація (Authentication)

В даному розділі описано атаки, спрямовані на використовувані веб-додатком методи перевірки ідентифікатора користувача, служби або програми. Аутентифікація використовує як мінімум один з трьох механізмів (факторів):

"щось, що ми маємо", "щось, що ми знаємо" або "щось, що ми є". Опишемо атаки, спрямовані на обхід або експлуатацію вразливостей в механізмах реалізації аутентифікації веб-серверів.

1.1.1 Підбір (Brute Force)

Підбір – автоматизований процес проб і помилок, що використовується для того, щоб вгадати ім'я користувача, пароль, номер кредитної картки, ключ шифрування і т.д.

Багато систем дозволяють використовувати слабкі паролі або ключі шифрування, і користувачі часто вибирають парольні фрази, які легко вгадати або ті, які містяться у словниках.

Використовуючи цю ситуацію, зловмисник може скористатися словником і спробувати використати тисячі або навіть мільйони комбінацій символів в якості пароля.

Якщо випробуваний пароль дозволяє отримати доступ до системи, атака вважається успішною і атакуючий може використовувати обліковий запис.

Подібна техніка проб і помилок може бути використана для підбору ключів шифрування. У разі використання ключів недостатньою довжини, зловмисник може отримати необхідний ключ, перебравши всі можливі комбінації.

Існує два види підбору: прямий і зворотний. При прямому підборі використовуються різні варіанти пароля для одного імені користувача. При зворотному перебираються різні імена користувачів, а пароль залишається незмінним. В системах з мільйонами облікових записів ймовірність використання різними користувачами одного пароля досить висока. Не дивлячись на популярність і високу ефективність, підбір може займати кілька годин, днів або років.

1.1.2 Недостатня аутентифікація (Insufficient Authentication)

Ця уразливість виникає, коли веб-сервер дозволяє атакуючому отримувати доступ до важливої інформації або функцій сервера без належної аутентифікації. Інтерфейси адміністрування через Web – яскравий приклад критичних систем.

Залежно від специфіки програми, подібні компоненти не повинні бути доступні без належної аутентифікації. Щоб не використовувати аутентифікацію деякі ресурси "ховаються" за певною адресою, на яку немає посилань на основних сторінках сервера або інших загальнодоступних ресурсах. Однак, подібний підхід не більш ніж "безпека через приховування". Важливо розуміти, що, не дивлячись на те, що зловмисник не знає адреси сторінки, вона все одно доступна через Web.

Необхідний URL може бути знайдений перебором типових файлів і директорій (таких як /admin/), з використанням повідомлень про помилки, журналів перехресних посилань або шляхом простого читання документації. Подібні ресурси повинні бути захищені адекватно важливості їх вмісту і функціональних можливостей.

1.1.3 Небезпечне відновлення паролів (Weak Password Recovery Validation)

Ця уразливість виникає, коли веб-сервер дозволяє атакуючому несанкціоновано отримувати, модифікувати або відновлювати паролі інших користувачів.

Часто аутентифікація на веб-сервер вимагає від користувача запам'ятовування паролю або парольної фрази. Тільки користувач повинен знати пароль, причому пам'ятати його виразно. З часом пароль забувається. Ситуація ускладнюється, оскільки в середньому користувач відвідує близько 20 сайтів, що вимагають введення пароля.

Таким чином, функція відновлення паролю є важливою складовою, що надається веб-серверами сервісу.

Прикладом реалізації подібної функції є використання "секретного питання", відповідь на яке вказується в процесі реєстрації. Питання або вибирається зі списку або вводиться самим користувачем. Ще один механізм дозволяє користувачу вказати "підказку", яка допоможе йому згадати пароль. Інші способи вимагають від користувача вказати частину персональних даних, таких як номер соц. страхування, ІПН, домашню адресу, поштовий індекс і т.д., які потім будуть використовуватися для встановлення особи. Після того як користувач доведе свою ідентичність, система відобразить новий пароль або перешле його поштою.

1.2 Авторизація (Authorization)

В даному розділі опишемо атаки, направлені на методи, які використовуються веб-сервером для визначення того, чи має користувач, служба або програма необхідний дозвіл для вчинення дії. Багато веб-сайтів дозволяють тільки певним користувачам отримувати доступ до деякого вмісту або функцій програми. Доступ іншим користувачам повинен бути обмежений. Використовуючи різні технології, зловмисник може підвищити свої привілеї і отримати доступ до захищених ресурсів.

1.2.1 Передбачуване значення ідентифікатора сесії (Credential/Session Prediction)

Передбачуване значення ідентифікатора сесії дозволяє перехоплювати сесії інших користувачів. Подібні атаки виконуються шляхом передбачення або вгадування унікального ідентифікатора сесії користувача. Ця атака також як і перехоплення сесії (Session Hijacking) у разі успіху дозволяє зловмисникові послати запит веб-сервера з правами скомпрометованого користувача. Дизайн багатьох серверів припускає аутентифікацію користувача при першому зверненні та подальше відстеження його сесії. Для цього користувач вказує комбінацію імені та пароля. Замість повторної передачі ім'я користувача та

паролю при кожній транзакції, веб-сервер генерує унікальний ідентифікатор, який присвоюється сесії користувача. Наступні запити користувача до сервера містять ідентифікатор сесії як доказ того, що аутентифікація була успішно пройденою. Якщо атакуючий може передбачити або вгадати значення ідентифікатора іншого користувача, це може бути використано для проведення атаки.

1.2.2 Недостатня авторизація (Insufficient Authorization)

Недостатня авторизація виникає, коли веб-сервер дозволяє атакуючому отримувати доступ до важливої інформації або функцій, доступ до яких повинен бути обмежений. Те, що користувач пройшов аутентифікацію не означає, що він повинен отримати доступ до всіх функцій і вмісту сервера. Крім аутентифікації повинно бути реалізовано розмежування доступу.

Процедура авторизації визначає, які дії може здійснювати користувач, служба або додаток. Правильно побудовані правила доступу повинні обмежувати дії користувача відповідно до політики безпеки. Доступ до важливих ресурсів сайту повинен бути дозволений тільки адміністраторам.

1.2.3 Відсутність таймауту сесії (Insufficient Session Expiration)

У випадку, якщо для ідентифікатора сесії або облікових даних не передбачений таймаут або має значення дуже велике, зловмисник може скористатися старими даними для авторизації. Це підвищує уразливість сервера для атак, пов'язаних з крадіжкою ідентифікаційних даних. Оскільки протокол HTTP не передбачає контроль сесії, веб-сервери зазвичай використовують ідентифікатори сесії для визначення запитів користувача. Таким чином, конфіденційність кожного ідентифікатора повинна бути забезпечена, щоб запобігти багаторазового доступу користувачів з одним профілем. Викрадений ідентифікатор може використовуватися для доступу до даних користувача або здійснення шахрайських транзакцій. Відсутність таймауту сесії збільшує ймовірність успіху різних атак. Наприклад, зловмисник може отримати

ідентифікатор сесії, використовуючи мережевий аналізатор або вразливість типу міжсайтового виконання сценаріїв. Хоча таймаут не допоможе у випадку, якщо ідентифікатор буде використаний негайно, обмеження часу допоможе у випадку більш пізніх спроб використання ідентифікатора.

В іншій ситуації, якщо користувач отримує доступ до сервера з публічного комп'ютера (бібліотека, Internet-кафе і т.д.) відсутність таймауту сесії може дозволити зловмисникові скористатися історією браузера для перегляду сторінок користувача.

Велике значення таймауту збільшує шанси підбору чинного ідентифікатора. Крім того, збільшення цього параметра веде до збільшення одночасно відкритих сесій, що ще більше підвищує ймовірність успішного підбору.

1.2.4 Фіксація сесії (Session Fixation)

Використовуючи даний клас атак, зловмисник присвоює ідентифікатору сесії користувача задане значення. Залежно від функціональних можливостей сервера, існує декілька способів "зафіксувати" значення ідентифікатора сесії. Для цього можуть використовуватися атаки типу міжсайтового виконання сценаріїв або підготовка сайту з допомогою попереднього HTTP запиту. Після фіксації значення ідентифікатора сесії атакуючий очікує момент, коли користувач увійде в систему. Після входу користувача, зловмисник використовує ідентифікатор сесії для отримання доступу до системи від імені користувача.

Можна виділити два типи систем управління сесіями на основі ідентифікаторів. Перший з них, "дозволяючий", дає змогу браузеру вказувати будь-який ідентифікатор. Системи другого "суворого" типу обробляють тільки ідентифікатори, згенеровані сервером. Якщо використовуються "дозволяючі" системи, зловмисник може вибрати будь-який ідентифікатор сесії. У випадку із "суворими" серверами зловмисникові доводиться підтримувати "сесію-

заглушку" і періодично з'єднуватися з сервером для уникнення закриття сесії за таймаут.

Без наявності активного захисту від фіксації сесії, ця атака може бути використана проти будь-якого сервера, аутентифікує користувачів за допомогою ідентифікатора сесії. Більшість веб-серверів зберігає ID в cookie, але це значення так само може бути присутнім в URL або прихованому полі форми.

На жаль, системи, що використовують cookie, є найбільш уразливими. Більшість відомих на даний момент варіантів фіксації сесії спрямовані саме на значення cookie.

1.3 Атаки на клієнтів (Client-side Attacks)

Опишемо атаки на користувачів веб-сервера. Під час відвідування сайту, між користувачем і сервером встановлюються довірчі відносини, як у технологічному, так і в психологічному аспектах. Користувач очікує, що сайт надасть йому легітимний вміст. Крім того, користувач не очікує атак з боку сайту. Експлуатуючи цю довіру, зловмисник може використовувати різні методи для проведення атак на клієнтів сервера.

1.3.1 Підміна вмісту (Content Spoofing)

Використовуючи цю техніку, зловмисник змушує користувача повірити, що сторінки згенеровані веб-сервером, а не передана із зовнішнього джерела.

Деякі веб-сторінки створюються з використанням динамічних джерел HTML-коду. Наприклад, розташування фрейму

```
<frame src=" http://foo.example/file.html">
```

може передаватися у параметрі

URL (http://foo.example/page?fr_src=http://foo.example/file.html).

Атакуючий може замінити значення параметра "frame_src" на

"frame_src = http://attacker.example/spoof.html".

Коли буде відображатися результуюча сторінка, в рядку адреси браузера користувача відображатиметься адреса сервера (foo.example), але так само на сторінці буде присутній вміст із зовнішнього джерела, завантажене з сервера атакуючого (attacker.example), замасковане під легальний контент.

Спеціально створене посилання може бути надіслане електронною поштою, системі моментального обміну повідомленнями, опублікована на дошці повідомлень або відкрита в браузері користувача з використанням міжсайтового виконання сценаріїв. Якщо атакуючий спровокував користувача на перехід по спеціально створеному посиланню, у користувача може скластися враження, що він переглядає дані з сервера, в той час як частина їх була згенерована зловмисником.

Таким чином, відбудеться "дефейс" сайту http://foo.example на стороні користувача, оскільки вміст сервера буде додано з сервера http://attacker.example. Ця атака так само може використовуватися для створення помилкових сторінок, таких як форми введення пароля, прес-релізи і т.д.

1.3.2 Міжсайтове виконання сценаріїв (Cross-site Scripting, XSS)

Наявність уразливості Cross-site Scripting дозволяє атакуючому передати серверу код, який буде виконуватися перенаправлено браузеру користувача. Цей код зазвичай створюється на мовах HTML / JavaScript, але можуть бути використані VBScript, ActiveX, Java, Flash, або інші мови, які підтримує браузер.

Переданий код виконується в контексті безпеки (або зоні безпеки) уразливого сервера. Використовуючи ці привілеї, код отримує можливість

читати, модифікувати або передавати важливі дані, доступні за допомогою браузера. У атакованого користувача може бути скомпрометований аккаунт (крадіжка cookie), його браузер може бути перенаправлений на інший сервер або здійснена підміна вмісту сервера. У результаті ретельно спланованої атаки зловмисник може використовувати браузер жертви для перегляду сторінок сайту від імені атакованого користувача. Код може передаватися зловмисником в URL, в заголовках HTTP запиту (cookie, user-agent, refferer), значеннях полів форм і т.д.

Існує два типи атак, що приводять до міжсайтового виконання сценаріїв: постійні (збережені) і непостійні (відображені). Основною відмінністю між ними є те, що у другому варіанті передача коду сервером та повернення його клієнту здійснюється в рамках одного HTTP-запиту, а в першому - в різних.

Здійснення непостійної атаки вимагає, щоб користувач перейшов за посиланням, згенерованому зловмисником (посилання може бути передане за email, ICQ тощо). У процесі завантаження сайту код, впроваджений в URL або заголовки запиту буде переданий клієнту і виконаний у його браузері. Збережений різновид уразливості виникає, коли код передається серверу і зберігається на ньому на деякий проміжок часу. Найбільш популярними цілями атак у цьому випадку є форуми, пошта з веб-інтерфейсом і чати. Для атаки користувачеві не обов'язково переходити по посиланню, досить відвідати

1.3.3 Розщеплення HTTP-запиту (HTTP Response Splitting)

При використанні даної уразливості зловмисник посилає серверу спеціальним чином сформований запит, відповідь на який інтерпретується метою атаки як дві різні відповіді. Друга відповідь повністю контролюється зловмисником, що дає йому можливість підробити відповідь сервера.

В реалізації атак із розщепленням HTTP-запиту беруть участь як мінімум три сторони:

– веб-сервер, який містить подібну уразливість;

– мета атаки, що взаємодіють з веб-сервером під управлінням зловмисника. Типово в якості мети атаки виступає кешуючий сервер-посередник або кеш браузера.

– атакуючий, який ініціює атаку.

Можливість здійснення атаки виникає, коли сервер повертає дані, надані користувачем в заголовках HTTP відповіді. Зазвичай це відбувається при перенаправленні користувача на іншу сторінку (коди HTTP 3xx) або коли дані, отримані від користувача, зберігаються в cookie.

В першій ситуації URL, на який відбувається перенаправлення, є частиною заголовка Location HTTP відповіді, а в другому випадку значення cookie передається в заголовку Set-Cookie.

Основою розщеплення HTTP-запиту є впровадження символів переведення рядка (CR і LF) таким чином, щоб сформувати дві HTTP транзакції, в той час як реально буде відбуватися тільки одна. Переклад рядка використовується для того, що б закрити першу (стандартну) транзакцію, і сформувати другу пару питання/відповідь, повністю контрольовану зловмисником і абсолютно непередбачувану логікою програми.

У результаті успішної реалізації цієї атаки зловмисник може виконати наступні дії:

– міжсайтове виконання сценаріїв;

– модифікація даних кешу сервера-посередника. Деякі кешуючі сервери-посередники (Squid 2.4, NetCache 5.2, Apache Proxy 2.0 і ряд інших), зберігають підроблений зловмисником відповідь на жорсткому диску і на подальші запити користувачів за цією адресою повертають кешовані дані. Це призводить до заміни сторінок сервера на стороні клієнта. Крім цього, зловмисник може переправити собі значення Cookie користувача або присвоїти їм певне значення. Так само ця атака може бути спрямована на індивідуальний кеш браузера користувача.

– міжкористувацька атака (один користувач, одна сторінка, тимчасова підміна сторінки). При реалізації цієї атаки зловмисник не посилає

додатковий запит. Замість цього використовується той факт, що деякі сервери-посередники поділяють одне TCP-з'єднання до сервера між декількома користувачами. В результаті другий користувач отримує у відповідь сторінку, сформовану зловмисником. Крім підміни сторінки зловмисник може також виконати різні операції з cookie користувача.

– перехоплення сторінок, що містять дані користувача. В цьому випадку зловмисник отримує відповідь сервера замість самого користувача. Таким чином, він може отримати доступ до важливої або конфіденційної інформації.

1.4 Виконання коду (Command Execution)

Опишемо атаки, спрямовані на виконання коду на веб-сервері. Всі сервери використовують дані, віддані користувачем при обробці запитів. Часто ці дані використовуються при складанні команд, що застосовуються для генерації динамічного вмісту. Якщо при розробці не враховуються вимоги безпеки, зловмисник отримує можливість модифікувати виконавчі команди.

1.4.1 Переповнення буфера (Buffer Overflow)

Експлуатація переповнення буфера дозволяє зловмисникові змінити шлях виконання програми шляхом перезапису даних у пам'яті системи. Переповнення буфера є найбільш поширеною причиною помилок у програмах. Воно виникає, коли об’єм даних перевищує розмір пам’яті буфера, виділеної під дані. Коли буфер переповнюється, дані переписують інші області пам'яті, що призводить до виникнення помилки. Якщо зловмисник має можливість управляти процесом переповнення, це може викликати ряд серйозних проблем.

Переповнення буфера може викликати відмови в обслуговуванні, приводячи до пошкодження пам'яті і викликаючи помилки в програмах. Більш серйозні ситуації дозволяють змінити шлях виконання програми і виконати в її контексті різні дії. Це може відбуватися в кількох випадках.

Використовуючи переповнення буферу, можна перезаписувати службові області пам'яті, наприклад, адресу повернення з функцій у стеці. Також, при переповненні можуть бути переписані значення змінних у програмі.

Переповнення буфера є найбільш поширеною проблемою в безпеці і нерідко зачіпає веб-сервери. Проте атаки, що експлуатують цю уразливість, використовуються проти веб-додатків не дуже часто. Причина цього криється в тому, що атакуючому, як правило, необхідно проаналізувати вихідний код або образ програми. Оскільки атакуючому доводиться експлуатувати нестандартну програму на віддаленому сервері, йому доводиться атакувати "всліпу", що знижує шанси на успіх.

Переповнення буфера в основному виникає при створенні програм на мовах C і C + +. Якщо частина сайту створена з використанням цих мов, сайт може бути вразливий для переповнення буферу.

1.4.2 Атака на функції форматування рядків (Format String Attack)

При використанні цих атак шлях виконання програми модифікується методом перезапису областей пам'яті за допомогою функцій форматування символьних змінних. Вразливість виникає, коли дані користувача застосовуються в якості аргументів функцій форматування рядків, таких як fprintf, printf, sprintf, setproctitle, syslog і т.д. Якщо атакуючий передає додаткам рядок, що містить символи форматування ("% f", "% p", "% n" і т.д.), то в нього з'являється можливість:

- виконати довільний код на сервері;
- зчитати значення зі стеку;
- викликати помилки в програмі / відмова в обслуговуванні.

1.4.3 Впровадження операторів LDAP (LDAP Injection)

Атаки цього типу спрямовані на веб-сервери, що створюють запити до служби LDAP на основі даних, які вводяться користувачем. Спрощений протокол доступу до служби каталогу (Lightweight Directory Access Protocol,

LDAP) – відкритий протокол для створення запитів і управління службами каталогу сумісними зі стандартом X.500.

Протокол LDAP працює поверх транспортних протоколів Internet (TCP / UDP). веб-додаток може використовувати дані, надані користувачем для створення запитів по протоколу LDAP при генерації динамічних веб-сторінок. Якщо інформація, отримана від клієнта, належним чином не верифікується, атакуючий отримує можливість модифікувати LDAP-запит.

Запит буде виконуватися з тим же рівнем привілеїв, з яким працює компонент програми, що виконує запит (сервер СКБД, веб-сервер і т.д). Якщо цей компонент має права на читання або модифікацію даних у структурі каталогу, зловмисник отримує ті ж можливості.

Техніка експлуатації даної уразливості мало відрізняється від впровадження операторів SQL, описаної далі.

1.4.4 Виконання команд ОС (OS Commanding)

Атаки цього класу спрямовані на виконання команд операційної системи на веб-сервері шляхом маніпуляції вхідними даними. Якщо інформація, отримана від клієнта, належним чином не верифікується, атакуючий отримує можливість виконати команди ОС. Вони будуть виконуватися з тим же рівнем привілеїв, з яким працює компонент програми, що виконує запит (сервер СУБД, веб-сервер і т.д).

1.4.5 Впровадження операторів SQL (SQL Injection)

Ці атаки спрямовані на веб-сервери, що створюють SQL запити до серверів СКБД на основі даних, які вводяться користувачем.

Мова запитів Structured Query Language (SQL) являє собою спеціалізований мова програмування, що дозволяє створювати запити до серверів СКБД. Більшість серверів підтримують цю мову у варіантах, стандартизованих ISO та ANSI. У більшості сучасних СКБД присутні розширення діалекту SQL, специфічні для даної реалізації (T-SQL в Microsoft

SQL Server, - PL SQL в Oracle і т.д.). Багато веб-додатків використовують дані, передані користувачем, для створення динамічних веб-сторінок.

Якщо інформація, отримана від клієнта, належним чином не верифікується, атакуючий отримує можливість модифікувати запит до SQL-серверу, що відправляється додатком. Запит буде виконуватися з тим же рівнем привілеїв, з яким працює компонент програми, що виконує запит (сервер СКБД, веб-сервер і т.д). У результаті зловмисник може отримати повний контроль над сервером СКБД і навіть його операційною системою.

1.4.6 Впровадження серверних розширень (SSI Injection)

Атаки даного класу дозволяють зловмисникові передати виконуваний код, який надалі буде виконаний на веб-сервері. Уразливості, що приводять до можливості здійснення даних атак, зазвичай полягають у відсутності перевірки даних, наданих користувачем, перед збереженням.

Перед генерацією HTML сторінки сервер може виконувати сценарії, наприклад, Server-site Includes (SSI). У деяких ситуаціях вихідний код сторінок генерується на основі даних, наданих користувачем.

Якщо атакуючий передає серверу оператори SSI, він може отримати можливість виконання команд операційної системи або включити до неї заборонений вміст при наступному відображенні.

1.4.7 Впровадження операторів XPath (XPath Injection)

Ці атаки спрямовані на веб-сервери, що створюють запити на мові XPath на основі даних, які вводяться користувачем.

Мова XPath 1.0 розроблена для надання можливості звернення до частин документа на мові XML. Він може бути використаний безпосередньо або в якості складової частини XSLT-перетворення XML-документів або виконання запитів XQuery.

Синтаксис XPath близький до мови SQL запитів. Припустимо, що існує документ XML, що містить елементи, відповідні іменам користувачів, кожен з

яких містить три елементи – ім'я, пароль та номер рахунку. Наступний вираз мовою XPath дозволяє визначити номер рахунку користувача "jsmith" з паролем "Demo1234":

```
string (/ / user [name / text () = 'jsmith' and
password / text () = 'Demo1234'] / account / text ())
```

Якщо запити XPath генеруються під час виконання на основі користувацького вводу, в атакуючого з'являється можливість модифікувати запит з метою обходу логіки роботи програми.

1.5 Розголошення інформації (Information Disclosure)

Атаки даного класу направлені на отримання додаткової інформації про веб-додаток. Використовуючи ці уразливості, зловмисник може визначити використовувані дистрибутиви ПЗ, номери версій клієнта і сервера і встановлені оновлення. В інших випадках, у витікаючій інформації може міститися розташування тимчасових файлів або резервних копій. У багатьох випадках ці дані не потрібні для роботи користувача. Більшість серверів надають доступ до надмірного об’єму даних, однак необхідно мінімізувати обсяг службової інформації. Чим більші знання про програму відомі зловмиснику, тим легше йому буде скомпрометувати систему.

1.5.1 Індексування директорій (Directory Indexing)

Надання списку файлів в директорії являє собою нормальну поведінку веб-сервера, якщо сторінка, яка відображається по замовчуванню (index.html / home.html / default.htm) відсутня.

Коли користувач запитує основну сторінку сайту, він зазвичай вказує доменне ім'я сервера без імені конкретного файлу (http://www.example). Сервер переглядає основну папку, знаходить в ній файл, який використовується по

замовчуванню, і на його основі генерує відповідь. Якщо такий файл відсутній, в якості відповіді може повернутися список файлів в директорії сервера.

Ця ситуація аналогічна виконанню команди "ls" (Unix) або "dir" (Windows) на сервері і форматування результатів у вигляді HTML.

У цій ситуації зловмисник може отримати доступ до даних, не призначених для вільного доступу. Досить часто адміністратори покладаються на "безпеку через приховування", припускаючи, що оскільки гіперпосилання на документ відсутнє, то він недоступний. Сучасні сканери вразливостей, такі як Nikto, можуть динамічно додавати файли і папки до списку сканованих залежно від результатів запитів. Використовуючи інформацію із /robots.txt або отриманого списку директорій сканер може знайти прихований вміст або інші файли.

Таким чином, зовні безпечне індексування директорій може призвести до витоку важливої інформації, яка в подальшому буде використана для проведення атак на систему.

1.5.2 Ідентифікація додатків (Web Server/Application Fingerprinting)

Визначення версій додатків використовується зловмисником для отримання інформації про використовувану сервером і клієнтом операційну систему, веб-сервер і браузери. Також ця атака може бути спрямована на інші компоненти веб-додатків, наприклад, службу каталогу або сервер баз даних.

Зазвичай подібні атаки здійснюються шляхом аналізу різної інформації, що надається веб-сервером, наприклад:

- особливості реалізації протоколу HTTP;
- заголовки HTTP-відповідей;
- використовувані сервером розширення файлів (. Asp або. Jsp);
- значення Cookie (ASPSESSION і т.д.);
- повідомлення про помилки;
- структура каталогів і використовувана угода про імена (Windows / Unix);

– інтерфейси підтримки розробки веб-додатків (Frontpage / WebPublisher);

– інтерфейси адміністрування сервера (iPlanet / Comanche);

– визначення версій операційної системи.

Для визначення версій клієнтських додатків зазвичай використовується аналіз HTTP-запитів (порядок слідування заголовків, значення User-agent і т.д.). Проте, для цих цілей можуть застосовуватись і інші техніки. Так, наприклад, аналіз заголовків листів, створених за допомогою клієнта Microsoft Outlook, дозволяє визначити версію встановленого на комп'ютері браузера Internet Explorer.

Наявність детальної та точної інформації про використовувані додатки дуже важливо для зловмисника, оскільки реалізація багатьох атак (наприклад, переповнення буфера) специфічна для кожного варіанту операційної системи або програми. Крім того, детальна інформація про інфраструктуру дозволяє зменшити кількість помилок, і як наслідок – загальний “шум”, що виробляється атакуючим. Даний факт відзначено в HTTP RFC 2068, що рекомендує, щоб значення заголовка Server HTTP відповіді було налаштованим параметром.

1.5.3 Витік інформації (Information Leakage)

Ці уразливості виникають у ситуаціях, коли сервер публікує важливу інформацію, наприклад коментарі розробників або повідомлення про помилки, яка може бути використана для компрометації системи. Цінні з точки зору зловмисника дані можуть міститися в коментарях HTML, повідомленнях про помилки або просто бути присутнім у відкритому вигляді. Існує величезна кількість ситуацій, в яких може статися витік інформації. Не обов'язково вона приводить до виникнення уразливості, але часто дає атакуючому чудовий посібник для розвитку атаки. З витоком важливої інформації можуть виникати ризики різного ступеня, тому необхідно мінімізувати кількість службової інформації, доступної на клієнтській стороні.

Аналіз доступної інформації дозволяє зловмисникові зробити розвідку і отримати уявлення про структуру директорій сервера, що використовуються у SQL запитах, назвах ключових процесів і програм сервера.

Часто розробники залишають коментарі у HTML сторінках і коді сценаріїв для полегшення пошуку помилок і підтримки програми. Ця інформація може варіюватися від простих описів деталей функціонування програми до, в гірших випадках, імен користувачів і паролів, які використовуються при налагодженні.

Витік інформації може відноситися і до конфіденційних даних, які обробляються сервером. Це можуть бути ідентифікатори користувача (ІПН, номери водійських посвідчень, паспортів і т.д.), а також поточна інформація (баланс особового рахунку або історія платежів).

Більшість атак цієї категорії виходять за рамки захисту веб-додатків і переходять в область фізичної безпеки. Витік інформації в цьому випадку часто виникає, коли в браузері відображається інформація, яка не повинна виводитися у відкритому вигляді навіть користувачеві. Як приклад можна навести паролі користувача, номери кредитних карток і т.д.

1.5.4 Зворотний шлях в директоріях (Path Traversal)

Дана техніка атак спрямована на отримання доступу до файлів, папок та командам, які знаходяться поза основною директорією веб-сервера. Зловмисник може маніпулювати параметрами URL з метою отримати доступ до файлів або виконати команди, розташовувані у файловій системі веб-сервера. Для подібних атак потенційно вразливий будь-який пристрій, що має веб-інтерфейс.

Більшість веб-серверів обмежують доступ користувача певною частиною файлової системи, звичайно названої "web document root" або "CGI root". Ці директорії містять файли, призначені для користувача і програми, необхідні для отримання доступу до функцій веб-додатки.

Більшість базових атак, що експлуатують зворотний шлях, засновані на впровадженні в URL символів "../", для того, щоб змінити розташування ресурсу, який буде оброблятися сервером. Оскільки більшість веб-серверів фільтрують цю послідовність, зловмисник може скористатися альтернативними кодуваннями для представлення символів переходу по директоріях. Популярні прийоми включають використання альтернативних кодувань, наприклад Unicode ("..% u2216 "або" ..% c0% af "), використання зворотного слешу (" .. \ ") в Windows-серверах, символів URLEncode ("% 2e% 2e % 2f ") чи подвійна кодування URLEncode ("..% 255c").

Навіть якщо веб-сервер обмежує доступ до файлів певним каталогом, ця вразливість може виникати в сценаріях або CGI-програмах. Можливість використання зворотного шляху в каталогах досить часто виникає в додатках, що використовують механізми шаблонів або завантажують їх текст сторінок з файлів на сервері. У цьому варіанті атаки зловмисник модифікує ім'я файлу, що передається як параметр CGI програми або серверного сценарію. У результаті зловмисник може отримати вихідний код сценаріїв. Досить часто до імені запитуваного файлу додаються спеціальні символи, такі як "% 00", з метою обходу фільтрів.

1.5.5 Передбачуване розміщення ресурсів (Predictable Resource Location)

Передбачуване розміщення ресурсів дозволяє зловмисникові отримати доступ до прихованих даних або функціональним можливостям. Шляхом підбору зловмисник може отримати доступ до вмісту, не призначеному для публічного перегляду. Тимчасові файли, файли резервних копій, файли конфігурації або стандартні приклади часто є метою подібних атак. У більшості випадків перебір може бути оптимізований шляхом використання стандартної угоди про імена файлів і директорій сервера. Отримувані зловмисником файли можуть містити інформацію про дизайн програми, інформацію з баз даних, імена машин або паролі, шляхи до директорій. Також «приховані» файли

можуть містити уразливості, відсутні в основному додатку. На цю атаку часто посилаються як на перерахування файлів і директорій (Forced Browsing, File Enumeration, Directory Enumeration).

1.6 Логічні атаки (Logical Attacks)

Атаки даного класу спрямовані на експлуатацію функцій програми або логіки його функціонування. Логіка програми представляє собою очікуваний процес функціонування програми при виконанні певних дій. В якості прикладів можна навести відновлення паролів, реєстрацію облікових записів, аукціонні торги, транзакції в системах електронної комерції. Додаток може вимагати від користувача коректного виконання кількох послідовних дій для виконання певного завдання. Зловмисник може обійти або використовувати ці механізми в своїх цілях.

1.6.1 Зловживання функціональними можливостями (Abuse of Functionality)

Дані атаки спрямовані на використання функцій веб-додатків з метою обходу механізмів розмежування доступу. Деякі механізми веб-додатків, включаючи функції забезпечення безпеки, можуть бути використані для цих цілей. Наявність уразливості в одному з, можливо, другорядних компонентів системи може призвести до компрометації всього додатку. Рівень ризику та потенційні можливості зловмисника в разі проведення атаки дуже сильно залежать від конкретного додатка.

Зловживання функціональними можливостями дуже часто використовується спільно з іншими атаками, такими як зворотний шлях в директоріях і т.д. Приміром, за наявності уразливості типу міжсайтового виконання сценаріїв в HTML-чаті зловмисник може використовувати функції чату для розсилки URL, який експлуатує уразливість, всім поточним користувачам.

З глобальної точки зору, всі атаки на комп'ютерні системи є зловживаннями функціональними можливостями. Особливо це стосується до атак, спрямованих на веб-додатки, які не вимагають модифікації функцій програми.

1.6.2 Відмова в обслуговуванні (Denial of Service)

Даний клас атак спрямований на порушення доступності веб-сервера. Зазвичай атаки, спрямовані на відмову в обслуговуванні реалізуються на мережевому рівні, проте вони можуть бути спрямовані і на прикладний рівень. Використовуючи функції веб-додатків, зловмисник може вичерпати критичні ресурси системи, або скористатися уразливістю, що приводить до припинення функціонування системи.

Зазвичай DoS атаки спрямовані на вичерпання критичних системних ресурсів, таких як обчислювальна потужність, оперативна пам'ять, дисковий простір або пропускна спроможність каналів зв'язку. Якщо якийсь із ресурсів досягне максимального завантаження, додаток цілком буде недоступний.

Атаки можуть бути спрямовані на будь-який із компонент веб-додатків, наприклад, такі як сервер СКБД, сервер аутентифікації і т.д. На відміну від атак на мережевому рівні, що вимагають значних ресурсів зловмисника, атаки на прикладному рівні зазвичай легше реалізувати.

DoS на інший сервер.

Зловмисник може розмістити на популярному веб-форумі посилання (наприклад, у вигляді зображення в повідомленні) на інший ресурс. При заході на форум, користувачі будуть автоматично завантажувати дані з атакованого серверу вказаний ресурс, використовуючи його ресурси. Якщо на атакуючому сервері використовується система запобігання атак з функцією блокування IP-адреси атакуючого, у посиланні може використовуватися сигнатура атаки (наприклад ../../../../../etc/passwd), що призведе до блокування користувачів, що зайшли на форум.

Атаки на сервер СКБД.

Зловмисник може скористатися впровадженням коду SQL для видалення даних з таблиць, що призведе до відмови в обслуговуванні програми.

1.6.3 Недостатня протидія автоматизації (Insufficient Anti-automation)

Недостатня протидія автоматизації виникає, коли сервер дозволяє автоматично виконувати операції, які повинні проводитися вручну. Для деяких функцій програми необхідно реалізовувати захист від автоматичних атак.

Автоматизовані програми можуть варіюватися від нешкідливих роботів пошукових систем до систем автоматизованого пошуку вразливостей і реєстрації облікових записів. Подібні роботи генерують тисячі запитів в хвилину, що може призвести до падіння продуктивності всього додатку.

Протидія автоматизації полягає в обмеженні можливостей подібних утиліт. Наприклад, файл robots може запобігати індексування деяких частин сервера, а додаткові затрати ідентифікації запобігати автоматичну реєстрацію сотень облікових записів системи електронної пошти.

1.6.4 Недостатня перевірка процесу (Insufficient Process Validation)

Уразливості цього класу виникають, коли сервер не достатньо перевіряє послідовність виконання операцій програми. Якщо стан сесії користувачів та програми належним чином не контролюється, додаток може бути уразливий для шахрайських дій.

У процесі доступу до деяких функцій програми очікується, що користувач виконає ряд дій в певному порядку. Якщо деякі дії виконуються невірно або у неправильному порядку, виникає помилка, що приводить до порушення цілісності. Прикладами подібних функцій виступають переклади, відновлення паролів, підтвердження покупки, створення облікового запису і т.д. У більшості випадків ці процеси складаються з ряду послідовних дій, здійснюваних у чіткому порядку.

Для забезпечення коректної роботи подібних функцій веб-додаток повинен чітко відслідковувати стан сесії користувачів та відслідковувати її

відповідність поточним операціям. У більшості випадків це здійснюється шляхом збереження стану сесії в cookie або прихованому полі форми HTML. Але оскільки ці значення можуть бути модифіковані користувачем, обов'язково має проводитися перевірка цих значень на сервері. Якщо цього не відбувається, зловмисник отримує можливість обійти послідовність дій, і як наслідок - логіку програми.

Приклад.

Система електронної торгівлі може пропонувати знижку на продукт В, у випадку купівлі продукту А. Користувач, який не хоче купувати продукт А, може спробувати придбати продукт В зі знижкою. Заповнивши замовлення на купівлю обох продуктів, користувач отримає знижку. Потім користувач повертається до форми підтвердження замовлення і видаляє продукт А, шляхом модифікації значень у формі. Якщо сервер повторно не перевірить можливість покупки продукту В за вказаною ціною без продукту А, буде здійснено закупівлю за низькою ціною.

1.7 Висновки до першого розділу

В розділі проаналізовано вразливості веб-серверів. Таким чином розглянуто компіляцію відомих класів атак, які представляють загрозу для веб-додатків в минулому і представляють зараз. Кожному класу атак присвоєно стандартну назву і описані його ключові особливості. Класи організовані в ієрархічну структуру

2 ВРАЗЛИВОСТІ ВЕБ-САЙТІВ ТА ВЕБ-ДОДАТКІВ

Веб-додатки можуть мати декілька вразливих моментів, які важко виявити вручну. Проведемо огляд найпоширеніших проблем безпеки, як перевірити стан своєї програми та отримати кращі поради щодо безпеки сайту.

Кожен веб-додаток може мати вразливі місця, будь то пов’язаний з основною системою ядра або одним веб-сайтом на основі популярного рішення CMS.

Багато додатків розробляються протягом тривалого періоду часу за участі різних середовищ та людей. Якщо ви самі програмували, то знаєте, що метою є, як правило, виправити помилку або змусити щось працювати. Коли все вирішено після багатогодинної роботи, просто запустити додаток таким, яким він є, і не замислюватися над питаннями безпеки [10].

«Найбільша помилка безпеки в усіх сайтах – пропуск роботи з виявлення вразливості. Інша проблема – це усвідомлення вразливих місць, щоб нічого не робити. Це частіше, ніж ви можете собі уявити. Зазвичай основні вразливості безпеки виправляються, тоді як залишаються менші. Кілька питань із низьким рівнем ризику швидко додаються до головного питання безпеки», – каже Лінус Саруд, дослідник безпеки компанії Detectify.

2.1 DDoS атаки

Випадки розподіленої відмови в сервісі (DDoS) зустрічаються частіше, ніж будь-коли раніше, і досі є найпопулярнішою формою атаки на веб-сайт.

Враховуючи це, не дивно, що в 2018 році відбулася найбільша в історії атака DDoS. За даними NETSCOUT, "американський провайдер" став ціллю атаки рефлексії/посилення, яка потрапила на їх веб-сайт із 1,7 терабайт зловмисних запитів в секунду. З певної точки зору, це еквівалентна пропускна здатність потокового телевізійного телебачення в 200000 одночасно.

На DDoS також припадає значна частина витрат на кіберзлочинність. Щорічний звіт про кібербезпеку від 2019 року виявив, що DDOS-атака зазвичай коштує біля 2-х мільйонів доларів, а найменш – 120 000 доларів.

Це не дивно, враховуючи, що DDoS-набори для нападу, доступні для придбання в Темній павутині, коштують приблизно 20 доларів, згідно зі статті Ars Technica.

Середня тривалість часу, на який пристрій, щойно підключений до Інтернету, потребує нападу на запит DDoS – 5 хвилин, повідомляє NETSCOUT.

Всі ці статистики знайомі, але DDoS-атаки також демонструють деякі нові функції. За словами Касперського, наприклад, в Китаї на кінець 2018 року припадало понад 50% DDoS-атак.

Ще одне занепокоєння полягає в тому, що з більшою кількістю пристроїв IoT, ніж коли-небудь підключених до Інтернету, потужність DDoS-атак може лише збільшитися. Gartner підрахував, що кількість пристроїв IoT до 2020 року досягне 20,4 мільярда, і це зробить атаки DDoS більш небезпечними, ніж будь-коли [10].

2.3 Malware

Зловмисне програмне забезпечення все ще залишається величезною проблемою. Насправді шкідливі програми зустрічаються частіше, ніж будь-коли.

Електронна пошта все ще є найпоширенішим способом розповсюдження зловмисних програм. CSO Online повідомили, що електронна пошта відповідає за поширення до 92% випадків зловмисного програмного забезпечення. Але це не означає, що веб-сайти не вразливі до зловмисного програмного забезпечення.

Більшість шкідливих програм зараз поширюється як шкідливі сценарії. Сценарії PowerShell давно стали величезним джерелом вразливості, але Symantec виявив, що використання шкідливих скриптів Powershell підскочило

на 1000% у 2018 році. У цьому ж звіті було встановлено, що сценарії утворюють 47,5% зловмисних вкладень електронної пошти.

Зловмисне програмне забезпечення впливає на всі типи пристроїв і може становити загрозу для веб-сайтів із ноутбуків, планшетів та смартфонів. Насправді смартфони цілком можуть стати найбільшим джерелом зловмисного програмного забезпечення в наступне десятиліття: мобільні покупки програмного забезпечення збільшився на 33% за останній рік, повідомляє Symantec.

Зловмисне програмне забезпечення також зараз є величезною загрозою для бізнесу. Зловмисне програмне забезпечення, спеціально націлене на підприємства, зросло на 12% у 2019 році, як виявило Symantec [11].

2.4 Людина, як джерело вразливості сайтів

Основним джерелом вразливості веб-сайтів є людина. Для погано захищених веб-сайтів хакерам досить легко вставити себе між клієнтами та власниками веб-сайтів та перехопити всю інформацію, що надсилається між ними.

Атаки MITM, як відомо, також збільшуються.

Наприклад, методи MITM були задіяні у 35% експлуатації веб-сайтів у 2018 році, відповідно до індексу розвідки X-Force Threat Intelligence Index 2019.

Це не дивно, враховуючи, наскільки багато підприємств підготовлені до атак MITM. Netcraft встановив, наприклад, що 95% серверів HTTP були вразливими до MiTM у 2016 році, і з того часу мало зроблено для виправлення цих вразливих місць.

Більше тривожним є той факт, що лише 10% компаній впровадили HSTS для своїх веб-сайтів, що залишає їх відкритими для атаки. W3Techs провели це дослідження, а також рекомендували всім веб-сайтам реалізувати протокол якомога швидше [12].

2.5 Атаки веб-додатків

Зараз веб-додатки є невід’ємною частиною майже кожного веб-сайту, і зростання їх використання супроводжувалося аналогічним зростанням їх експлуатації. Наприклад, згідно з дослідженнями компанії Imperva, більше половини веб-додатків мають публічний доступ, який доступний для хакерів, і більше третини цих проблем не мають рішення.

Згідно з доповіддю TrustWave, найбільш поширеними формами атак на веб-додатки є ті, що експлуатують міжсайтовий сценарій (XSS), який становить близько 40% таких атак, та ін'єкції SQL, на які припадає 24%.

Вразливості веб-додатків також надзвичайно поширені. Acunetix виявив, що 46% веб-сайтів мають такий тип вразливості.

Цей тип вразливості веб-сайтів також зростає. За результатами дослідження Akamai, у 2018 році на 38% зросла ін'єкція SQL та атаки міжсайтового сценарію. WordPress, найпопулярніший на сьогоднішній день CMS, є загальною ціллю ін'єкцій SQL, тому що більшість популярних хостів WordPress використовують SQL за замовчуванням.

За даними Acunetix, 2% веб-додатків також сприйнятливі до віддаленого виконання коду, що дозволяє зловмиснику виконувати свій (зловмисний) код у сценаріях вашого веб-сайту. І хоча 2% можуть не здаватися настільки високими, враховуючи велику кількість веб-сайтів там, це представляє величезну кількість вразливих веб-сайтів.

Насправді переважна більшість проникнень локальної мережі (LAN) у 2019 році відбулася через слабкі місця в веб-додатках, згідно з дослідженнями позитивних технологій [13].

Розглянемо масштаби вразливості веб-сайтів у 2019 році та найпоширеніші форми взлому.

Ці цифри можуть бути шокуючими, але вони підтверджують істину, яку ми всі знаємо вже декілька років. Масштаби кіберзлочинності – це величезна проблема, яку ми ніде не вирішуємо.

Проведення декількох основних кроків для захисту вашого веб-сайту може допомогти обмежити вашу сприйнятливість до цих кібератак і потенційно врятувати ваш бізнес від них. І ви також повинні пам’ятати, що ви не самотні – якщо ви використовуєте одну з найкращих веб-хостингових компаній Канади, вони допоможуть, надаючи інструменти безпеки, які дозволять захистити ваш веб-сайт [13].

2.6 Найпоширеніші вразливості веб-безпеки

Для дуже багатьох компаній, лише після того, як сталося порушення безпеки, найкращі практики веб-безпеки стануть пріоритетним. Протягом багатьох років роботи професіонали з безпеки IT неодноразово бачили, як виникають проблеми в програмістів з веб-розробок.

Ефективний підхід до загроз веб-безпеці, за визначенням, повинен мати активний захист. Тому метою цієї роботи, є спрямування по огляду проблем безпеки веб-розробок.

Зокрема, в цій роботі зосереджено 10 загальних та значних підводних проблем, які слід пам’ятати, включаючи рекомендації щодо їх усунення. Основна увага приділяється топ-10 вразливостям веб-сторінок, визначених проектом захисту відкритих веб-застосунків (OWASP), міжнародною некомерційною організацією, метою якої є поліпшення безпеки програмного забезпечення в усьому світі [14].

2.6.1 Автентифікація та авторизація, як система захисту

Розмовляючи з іншими програмістами та IT-професіоналами, ми часто стикаємося з плутаниною щодо відмінності між авторизацією та автентифікацією. І звичайно, той факт, що абревіатура auth часто використовується для обох, допомагає посилити цю загальну плутанину. Ця плутанина настільки поширена, що, можливо, ця проблема повинна бути включена в цю посаду як «Загальна вразливість веб-сторінок».

Отже, перш ніж продовжувати, давайте уточнимо різницю між цими двома термінами:

Перевірка автентичності: перевірка того, що людина є (або, принаймні, здається, є) певним користувачем, оскільки він / вона правильно надав свої дані безпеки (пароль, відповіді на питання безпеки, сканування відбитків пальців тощо) [14].

Авторизація: підтвердження того, що певний користувач має доступ до певного ресурсу або має дозвіл на виконання певної дії.

Заявлений іншим способом, автентифікація – це знання, хто є суб'єктом господарювання, а авторизація – це знання, що може зробити даний суб'єкт. Зважаючи на це, давайте вступимо в топ-10 проблем безпеки в Інтернеті.

2.6.2 Поширена помилка веб-безпеки №1: недоліки ін'єкції

Недоліки ін'єкції є результатом класичного невдалого фільтрування ненадійного введення. Це може статися, коли ви передаєте нефільтровані дані на SQL-сервер (SQL-ін'єкція), у браузер (XSS – про це ми поговоримо пізніше), на сервер LDAP (ін'єкція LDAP) або де-небудь ще. Проблема тут полягає в тому, що зловмисник може вводити команди цим об'єктам, що призводить до втрати даних та викрадення веб-переглядачів клієнтів.

Все, що ваша програма отримує з недовірених джерел, має бути відфільтровано, бажано відповідно до білого списку. Ви майже ніколи не використовуєте чорний список, оскільки отримати це право дуже важко і, як правило, легко обійти. Антивірусні програмні продукти, як правило, надають приклади збоїв у чорних списках. Відповідність шаблонів не працює.

Хороша новина полягає в тому, що захист від ін'єкцій – це просто питання правильної фільтрації ваших даних та роздумів про те, чи можна довіряти вкладенню. Але погана новина полягає в тому, що весь вхід потрібно правильно фільтрувати, якщо тільки йому можна беззаперечно довіряти (згідно з прислів’я «ніколи не кажи ніколи»).

Наприклад, у системі з 1000 входами, успішної фільтрації 999 з них недостатньо, оскільки це все одно залишає одне поле, яке може послужити завадою у вашій системі. І ви можете подумати, що введення результату SQL-запиту в інший запит – це гарна ідея, оскільки довіряють базі даних, але якщо периметр не вказаний, введення поступає опосередковано від хлопців із недостатнім запитом. Це називається інжекцією SQL другого порядку, якщо ви зацікавлені [15].

Оскільки фільтрування досить складно зробити правильно (як крипто), те, що я зазвичай раджу, – покладатися на функції фільтрації вашої рамки: вони перевірені, що вони працюють і ретельно перевіряються. Якщо ви не використовуєте фреймворки, вам дійсно потрібно добре подумати над тим, чи не використовувати їх справді має сенс у контексті безпеки вашого сервера. 99% часу це не робить.

2.6.3 Поширена помилка веб-безпеки №2: зламана автентифікація

Це сукупність безлічі проблем, які можуть виникнути під час порушеної автентифікації, але всі вони не випливають із однієї першопричини.

Припускаючи, що хтось все ще хоче скрутити власний код автентифікації у 2014 році (що ви думаєте ??), я раджу проти цього. Вийти вкрай важко, і існує безліч можливих підводних каменів, лише кілька:

URL може містити ідентифікатор сеансу та просочувати його у заголовку реферала для когось іншого.

Паролі не можуть бути зашифровані ні на зберіганні, ні на транзиті.

Ідентифікатори сеансу можуть бути передбачуваними, тому отримання доступу є тривіальним.

Фіксація сеансу може бути можливою.

Можливе викрадення сеансу, тайм-аути не реалізовані правильно або з використанням HTTP (відсутність безпеки SSL) тощо…

Запобігання: Найпростіший спосіб уникнути цієї вразливості безпеки в Інтернеті – це використання рамки. Ви можете це зробити правильно, але це

набагато простіше. У випадку, якщо ви хочете скрутити власний код, будьте вкрай параноїчні та навчіться, якими є підводні камені. Їх досить багато [15].

2.6.4 Поширена помилка веб-безпеки №3: Сценарій між веб-сайтами (XSS)

Це досить розповсюджена помилка санітарії введення (по суті, особливий випадок поширеної помилки №1). Зловмисник надає ваші веб-програми теги JavaScript на вході. Коли цей вхід повернеться користувачу несанкціонованому, його веб-переглядач виконає. Це може бути так само просто, як створити посилання та переконати користувача натиснути його, або це може бути щось набагато зловісніше. На завантаження сторінки виконується сценарій, і, наприклад, можна використовувати для розміщення файлів cookie на зловмисника. Профілактика. Існує просте рішення для веб-безпеки: не повертайте HTML-теги клієнту. Це має додаткову перевагу від захисту від введення HTML, аналогічної атаки, при якій зловмисник вводить звичайний вміст HTML (наприклад, зображення або гучні невидимі флеш-плеєри) – не сильно впливає, але, безумовно, дратує ("будь ласка, зупиніть!"). Зазвичай вирішення способу – це просто перетворення всіх HTML-об'єктів, так що <script> повертається як & lt; script & gt;. Інший часто застосовуваний метод санітарії – це використання регулярних виразів, щоб позбавити теги HTML, використовуючи регулярні вирази на <i>, але це небезпечно, оскільки багато браузерів інтерпретують сильно зламаний HTML просто чудово. Краще конвертувати всі символи в їх схожі колеги[14].

2.6.5 Загальна помилка веб-безпеки №4: Небезпечні посилання

Це класичний випадок довіритися вкладенню користувачів та сплатити ціну внаслідок вразливості безпеки. Пряма посилання на об'єкт означає, що внутрішній об’єкт, такий як файл або ключ бази даних, піддається користувачу. Проблема з цим полягає в тому, що зловмисник може надати цю посилання, і

якщо авторизація або не застосовується (або порушена), зловмисник може отримати доступ або робити речі, від яких вони повинні бути виключені.

Наприклад, у коді є модуль download.php, який читає і дозволяє користувачу завантажувати файли, використовуючи параметр CGI для визначення імені файлу (наприклад, download.php? File = something.txt). Або помилково, або через лінь розробник опустив авторизацію з коду. Тепер зловмисник може використовувати це для завантаження будь-яких системних файлів, до яких користувач під управлінням PHP має доступ, як-от сам код програми або інші дані, що залишаються на сервері, як резервні копії. Ой-ой.

Інший поширений приклад вразливості – це функція скидання пароля, яка спирається на введення користувача, щоб визначити, чий пароль ми скидаємо. Після натискання дійсної URL-адреси зловмисник може просто змінити поле імені користувача в URL-адресі, щоб сказати щось на зразок "адміністратор".

Між іншим, обидва ці приклади – це речі, які я сам бачив, часто з’являються «в дикій природі».

Запобігання. Виконуйте авторизацію користувачів належним чином та послідовно та додавайте до списку вибір. Частіше за все, всю проблему можна уникнути, зберігаючи дані внутрішньо і не покладаючись на те, що вони передаються від клієнта через параметри CGI. Для цієї мети добре підходять змінні сесії в більшості фреймів [15].

2.6.6 Поширена помилка веб-безпеки №5: неправильна конфігурація безпеки

На мій досвід, веб-сервери та програми, які були неправильно налаштовані, набагато частіше, ніж ті, які були налаштовані належним чином. Можливо, це тому, що немає дефіциту способів викрутити. Деякі приклади:

- Запуск програми з налагодженням з правами адміністратора.
- Увімкнено перелік каталогів на сервері.

– Запуск застарілого програмного забезпечення (плагіни WordPress, старий PhpMyAdmin).

– Маючи непотрібні служби на машині.

– Не змінюючи ключі та паролі за замовчуванням.

– Виявлення помилок щодо обробки інформації для зловмисників, таких як сліди стека.

Провести хороший (бажано автоматизований) процес "побудови та розгортання", який може запускати тести на розгортання. Рішення неправильної конфігурації – це проблеми після введення, щоб запобігти виходу коду із вбудованими паролями за замовчуванням та / або вбудованими елементами розробки [15].

2.6.7 Поширена помилка веб-безпеки №6: чутливий вплив даних

Ця вразливість веб-безпеки стосується захисту криптовалюти та ресурсів. Чутливі дані повинні бути зашифровані в будь-який час, включаючи транзит і в спокої. Немає винятків. Інформація про кредитні картки та паролі користувачів ніколи не повинні подорожувати та зберігатись незашифрованими, а паролі – завжди хешированими. Очевидно, що алгоритм крипто / хешування не повинен бути слабким – коли виникають сумніви, стандарти безпеки веб-сайтів рекомендують AES (256 біт і вище) та RSA (2048 біт і вище).

І хоча само собою зрозуміло, що ідентифікатори сеансу та конфіденційні дані не повинні пересуватись у URL-адресах, а на конфіденційних файлах cookie повинен бути захищений прапор.

Використовуйте HTTPS з належним сертифікатом та PFS (Perfect Forward Secret). Не приймайте нічого через з'єднання, що не є HTTPS. Майте захищений прапор на файлах cookie [15].

На зберіганні: Це складніше. Перш за все, вам потрібно знизити експозицію. Якщо вам не потрібні конфіденційні дані, подрібніть їх. Дані, яких

у вас немає, не можна вкрасти. Не зберігайте інформацію про кредитні картки ніколи, оскільки, напевно, не потрібно мати справу з сумісністю з PCI. Підпишіться за допомогою платіжного процесора, такого як Stripe або Braintree. По-друге, якщо у вас є конфіденційні дані, які вам насправді потрібні, зберігайте їх у зашифрованому вигляді та переконайтеся, що всі паролі хешировані. Для хешування рекомендується використовувати bcrypt. Якщо ви не використовуєте bcrypt, навчайте себе на засолювальних та веселкових столах.

І, ризикуючи сказати очевидним, не зберігайте ключі шифрування поруч із захищеними даними. Це як зберігання велосипеда із замком, у якому є ключ. Захистіть резервні копії за допомогою шифрування і збережіть ваші ключі дуже приватними. І звичайно, не втрачайте ключі/ [15]

2.6.8 Загальна помилка веб-безпеки № 7: Відсутність контролю рівня доступу до функції

Це просто помилка авторизації. Це означає, що коли функція викликається на сервері, належна авторизація не виконувалася. Багато разів розробники покладаються на те, що серверна сторона створила інтерфейс, і вони думають, що функціонал, який не надається сервером, не може отримати доступ клієнт. Це не так просто, оскільки зловмисник завжди може підробляти запити на "прихований" функціонал і не буде відляканий тим, що інтерфейс не робить цю функцію легко доступною. Уявіть, що є панель / адміністратор, і кнопка в інтерфейсі присутня лише у тому випадку, якщо користувач насправді є адміністратором. Ніщо не перешкоджає зловмиснику виявляти цю функціональність і не використовувати її в разі відсутності авторизації.

Профілактика: на стороні сервера завжди потрібно робити авторизацію. Так, завжди. Ніякі винятки чи вразливості не призведуть до серйозних проблем [15].

2.6.9 Поширена помилка веб-безпеки № 8: Підробка міжпрофільних заявок (CSRF)

Це приємний приклад заплутаної депутатської атаки, завдяки якій браузер обманює якусь іншу сторону в зловживанні своїми повноваженнями. Наприклад, сторонній веб-сайт може змусити браузера користувача неправомірно використовувати повноваження робити щось для зловмисника.

Що стосується CSRF, сторонній сайт надсилає запити на цільовий сайт (наприклад, ваш банк), використовуючи ваш веб-переглядач із файлами cookie / сеансом. Якщо ви, наприклад, увійшли на одну вкладку на домашній сторінці свого банку, і вони вразливі до цієї атаки, інша вкладка може змусити ваш веб-переглядач неправильно використовувати свої дані від імені зловмисника, що спричинить заплутану депутатську проблему. Заступник – це веб-переглядач, який зловживає своїми повноваженнями (файли cookie сесії), щоб зробити те, що зловмисник доручає йому робити [15].

2.6.10 Загальна помилка веб-безпеки №9: Використання компонентів з відомими вразливими місцями

У заголовку все сказано. Я знову класифікую це як більшу проблему з технічного обслуговування / розгортання. Перш ніж включити новий код, проробіть деякі дослідження, можливо, якийсь аудит. Використання коду, який ви отримали від випадкової людини на GitHub або на якомусь форумі, може бути дуже зручним, але це не без ризику серйозної вразливості веб-безпеки.

Є багато випадків, наприклад, коли сайти отримали право власності (тобто, коли сторонній користувач отримує адміністративний доступ до системи), не тому, що програмісти були дурними, а тому, що стороннє програмне забезпечення залишалося незавершеним протягом багатьох років у виробництві. Наприклад, це відбувається постійно з плагінами WordPress. Якщо ви думаєте, що вони не знайдуть вашу приховану установку phpmyadmin, дозвольте мені ознайомити вас з dirbuster.

Урок тут полягає в тому, що розробка програмного забезпечення не закінчується при розгортанні програми. Повинна бути документація, тести та плани щодо того, як підтримувати та оновлювати її, особливо якщо вона містить компоненти сторонніх чи відкритих джерел.

Будьте обережні. Крім того, очевидно, що обережно використовуйте такі компоненти, не будьте кодером для копіювання та вставки. Уважно перегляньте фрагмент коду, який ви збираєтеся вкласти у своє програмне забезпечення, оскільки він може бути зламаний після ремонту (або в деяких випадках навмисно зловмисних. Атаки веб-безпеки іноді мимоволі запрошуються таким чином).

Будьте в курсі. Переконайтеся, що ви використовуєте останні версії всього, чому ви довіряєте, і плануйте регулярно їх оновлювати. Принаймні підпишіться на розсилку нових вразливих місць безпеки щодо продукту [15].

2.11 Загальна помилка веб-безпеки №10: Неправільне переспрямування та переадресація

Це ще раз проблема вхідної фільтрації. Припустимо, що на цільовому сайті є модуль redirect.php, який приймає URL як параметр GET. Маніпулювання параметром може створити URL-адресу на targetite.com, яка перенаправляє браузер на malwareinstall.com. Коли користувач побачить посилання, він побачить targetite.com/blahblahblah, який користувач вважає довіреним і його безпечно клацати. Мало хто знає, що це насправді перенесе їх на падіння зловмисного програмного забезпечення (або будь-яку іншу шкідливу) сторінку. Крім того, зловмисник може перенаправити браузер на targetite.com/deleteprofile?confirm=1.

Варто зазначити, що заповнення несанізованого введеного користувачем вводу в HTTP-заголовок може призвести до введення заголовка, що є досить поганим.

Профілактика: варіанти включають:

– Не робіть переадресації взагалі (вони рідко потрібні).

– Майте статичний список дійсних локацій, на які потрібно переадресувати.

– Додайте до списку визначений користувачем параметр, але це може бути складним [15].

2.12 Тестування безпеки сайту

Основними джерелами небезпеки для веб-сайту є SQL-ін’єкції та XSS.

XSS (Cross site scripting)

Cross site scripting (також відома, як XSS) відбувається у разі отримання web-додатком небезпечних даних від користувача. Дані зазвичай беруться з форми або з гіперпосилання, зміст якого небезпечний. Зазвичай користувач переходить по цьому посиланню з іншого web-сайту, форуму, з повідомлення, отриманого по електронній пошті або по інтернет-пейджеру. Той, що зазвичай атакує кодує небезпечну частину посилання в hex-кодах (або іншими способами кодування), щоб посилання виглядало менш підозріло. Після отримання даних web-додатком воно виводить користувачеві сторінку, що містить небезпечні дані, які були спочатку послані йому, але ці дані відображаються, як дійсний зміст web-сайту.

Ті, що зазвичай атакують упроваджують Java Script, VBScript, Active X, HTML або Flash, щоб насолити користувачеві, або щоб отримати його інформацію. Можливо все що завгодно, від захоплення аккаунта, зміни настройок користувача, крадіжки/підміни cookie.

SQL-ін’єкція (англ. SQL injection) — одна з розповсюджених уразливостей. Атака з використанням даної може бути можлива при некоректній обробці вхідних даних, використовуваних в SQL-запитах. SQL-ін'єкція зазвичай дає можливість під час атаки виконати довільний запит зокрема до бази даних, та прочитати вміст таблиць або в свою чергу видалити всі дані.

Захист від XSS та SQL ін’єкцій [16].

Для того щоб надійно захистити свій сайт від XSS та SQL ін’єкцій потрібно дуже ретельно фільтрувати усі вхідні дані від клієнта. Треба не допускати передачу «небезпечних» символів. Такими символами є ті символи, які використовуються в синтаксисі програмних засобів, які використовуються при розробці сайту. В нашому випадку це HTML, PHP, MySQL.

Лістинг 2.1 – Використання програми захисту

```
<?php
$url=$QUERY_STRING;
$hack=strstr($url,"-");
if(strlen($hack)>=1) header("Location:
http://diplom-shop.ho.com.ua");
$hack=strstr($url,"select");
if(strlen($hack)>=1) header("Location:
http://diplom-shop.ho.com.ua");
$hack=strstr($url,"..");
if(strlen($hack)>=1) header("Location:
http://diplom-shop.ho.com.ua");
$hack=strstr($url,"where");
if(strlen($hack)>=1) header("Location:
http://diplom-shop.ho.com.ua");
?>
```

Для захисту свого Інтернет-магазину я від XSS атак та SQL ін’єкцій було використано програму показану на лістингу 2.1.

3 АТАКИ І ЗАХИСТ ВІД АТАК ВЕБ-САЙТІВ

3.1 Оновлення програмного забезпечення

Це може здатися очевидним, але забезпечення постійного оновлення всього програмного забезпечення є життєво важливим для забезпечення безпеки вашого сайту. Це стосується як операційної системи сервера, так і будь-якого програмного забезпечення, яке ви можете працювати на своєму веб-сайті, наприклад, CMS або форуму. Коли в програмному забезпеченні виявляються дірки в безпеці, хакери швидко намагаються зловживати ними.

Якщо ви використовуєте кероване рішення хостингу, то вам не потрібно так сильно хвилюватися щодо застосування оновлень безпеки для операційної системи, оскільки хостинг-компанія повинна про це подбати.

Якщо ви використовуєте стороннє програмне забезпечення на своєму веб-сайті, наприклад, CMS або форум, вам слід забезпечити швидке застосування будь-яких патчів безпеки. Більшість постачальників мають список розсилки або RSS-канал із детальним описом будь-яких проблем із безпекою веб-сайту. WordPress, Umbraco та багато інших CMS повідомляють вас про доступні оновлення системи під час входу.

Багато розробників використовують такі інструменти, як Composer, npm або RubyGems, щоб керувати своїми програмними залежностями та вразливими місцями безпеки, що з’являються в пакеті, від якого ви залежите, але не звертаєте на них ніякої уваги, – це один з найпростіших способів виходу з ладу. Переконайтеся, що ви постійно оновлюєте свої залежності та використовуйте такі інструменти, як Gemnasium, щоб отримувати автоматичні сповіщення, коли в одному з компонентів оголошено вразливість [16].

3.2 Слідкування за ін'єкцією SQL

Атаки введення SQL – це те, коли зловмисник використовує поле веб-форми або параметр URL для отримання доступу до вашої бази даних або маніпулювання ними. Коли ви використовуєте стандартний Transact SQL, легко несвідомо вставити у свій запит негідний код, який може бути використаний для зміни таблиць, отримання інформації та видалення даних. Ви можете легко запобігти цьому, завжди використовуючи параметризовані запити, більшість веб-мов мають цю функцію, і це легко реалізувати.

Розглянемо цей запит:

```
"SELECT * FROM table WHERE column = '" + parameter + "';"
```

Якщо зловмисник змінив параметр URL для передачі на 'або' 1 '=' 1, це призведе до вигляду запиту таким чином:

```
"SELECT * FROM table WHERE column = '' OR '1'='1';"
```

Оскільки '1' дорівнює '1', це дозволить зловмиснику додати додатковий запит до кінця оператора SQL, який також буде виконуватися [16].

Ви можете виправити цей запит, чітко параметризуючи його. Наприклад, якщо ви використовуєте MySQLi в PHP, це повинно стати:

```
$stmt = $pdo->prepare('SELECT * FROM table WHERE column = :value');
$stmt->execute(array('value' => $parameter));
```

3.3 Захист від атак XSS

Атаки міжсайтового скриптування (XSS) вводять на ваші сторінки зловмисний JavaScript, який потім запускається у браузерах ваших користувачів і може змінювати вміст сторінки або вкрадати інформацію для повернення зловмиснику. Наприклад, якщо ви показуєте коментарі на сторінці без валідації, зловмисник може надсилати коментарі, що містять теги скриптів

та JavaScript, які можуть працювати в браузері кожного іншого користувача та вкрасти їх файли cookie, що дозволяє атаці взяти під контроль кожен обліковий запис. користувач, який переглянув коментар. Вам потрібно переконатися, що користувачі не можуть вводити активний вміст JavaScript на ваші сторінки.

Це викликає особливе занепокоєння у сучасних веб-додатках, де сторінки тепер будуються головним чином із вмісту користувача, і які у багатьох випадках генерують HTML, який потім також інтерпретується фронтальними рамками, такими як Angular та Ember. Ці рамки забезпечують безліч захистів XSS, але змішування сервера та клієнта надає нові і більш складні способи атаки: не тільки введення JavaScript у HTML ефективний, але ви також можете вводити вміст, який буде запускати код, вставляючи кутові директиви або використовуючи Ember помічники [16].

Тут головне – зосередитись на тому, як створений користувачем вміст може вийти за межі, які ви очікуєте, і інтерпретувати браузер як щось інше, ніж те, що ви задумали. Це схоже на захист від ін'єкції SQL. При динамічному генеруванні HTML використовуйте функції, які явно вносять потрібні вам зміни (наприклад, використовуйте element.setAttribute та element.textContent, який автоматично вийде браузером, а не встановлюючи element.innerHTML вручну) або використовуйте функції у вашому інструменті для створення шаблонів, який автоматично виконує відповідні ескалації, а не об'єднання рядків або встановлення сировинного вмісту HTML.

Ще один потужний інструмент інструментаря захисника XSS – це політика безпеки вмісту (CSP). CSP – це заголовок, на який може повернутися ваш сервер, який вказує браузеру обмежувати, як і який JavaScript виконується на сторінці, наприклад, заборонити запуск будь-яких сценаріїв, які не розміщені у вашому домені, заборонити вбудований JavaScript або відключити eval (). У Mozilla є чудовий путівник із деякими прикладними конфігураціями. Це ускладнює роботу скриптів зловмисників, навіть якщо вони можуть потрапити на вашу сторінку.

3.4 Повідомлення про помилки

Будьте уважні до того, скільки інформації ви видаєте у своїх повідомленнях про помилки. Надайте користувачам лише мінімальні помилки, щоб вони не просочувались секретами, наявними на вашому сервері (наприклад, ключі API або паролі бази даних). Не вказуйте також детальну інформацію про винятки, оскільки це може значно спростити складні атаки, такі як ін'єкція SQL. Зберігайте докладні помилки у своїх журналах сервера та показуйте користувачам лише необхідну інформацію [17].

3.5 Підтвердження з обох сторін

Перевірку завжди слід робити як на веб-переглядачі, так і на стороні сервера. Веб-переглядач може зафіксувати прості помилки, такі як обов'язкові поля, які порожні, і коли ви вводите текст у поле лише для цифр Однак їх можна обійти, і ви повинні переконатися, що ви перевіряєте наявність цих даних і більш глибокої сторони сервера перевірки, оскільки цього не вдасться призвести до того, що в базу даних буде вставлено зловмисний код або код сценарію або призведе до небажаних результатів на вашому веб-сайті [18].

3.6 Перевірка своїх паролі

Усі знають, що вони повинні використовувати складні паролі, але це не означає, що вони завжди є. Важливо використовувати надійні паролі для адміністратора вашого сервера та веб-сайту, але не менш важливо наполягати на належній практиці використання паролів для захисту своїх облікових записів.

Настільки, наскільки користувачам це може не сподобатися, виконання вимог до пароля, як мінімум, близько восьми символів, включаючи великі

літери та цифри, допоможе захистити їх інформацію в довгостроковій перспективі.

Паролі завжди повинні зберігатися у вигляді зашифрованих значень, бажано з використанням одностороннього алгоритму хешування, такого як SHA. Використання цього методу означає, що під час аутентифікації користувачів ви лише порівнюєте зашифровані значення. Для додаткової безпеки веб-сайтів корисно солити паролі, використовуючи нову сіль на пароль.

У випадку, коли хтось увірветься та вкраде ваші паролі, використання хешованих паролів може допомогти обмежити шкоду, оскільки розшифрувати їх неможливо. Найкраще, що хтось може зробити, – це атака словника або груба сила, по суті, здогадуючись про кожну комбінацію, поки вона не знайде відповідності. При використанні солоних паролів процес злому великої кількості паролів відбувається навіть повільніше, оскільки кожну здогадку потрібно хешировать окремо для кожної солі + пароля, що обчислюється дуже дорого.

На щастя, багато CMS надають користувачеві управління з коробки безліч цих вбудованих функцій безпеки веб-сайту, хоча для використання солоних паролів (до Drupal 7) або для встановлення мінімальної потужності пароля можуть знадобитися деякі конфігураційні або додаткові модулі. Якщо ви використовуєте .NET, тоді варто скористатися постачальниками членства, оскільки вони дуже налаштовані, забезпечте вбудовану безпеку веб-сайту та включіть готові елементи керування для входу та скидання пароля [19].

3.7 Уникання завантаження файлів

Дозволити користувачам завантажувати файли на ваш веб-сайт може бути великим ризиком для безпеки веб-сайту, навіть якщо просто змінити аватар. Ризик полягає в тому, що будь-який завантажений файл, як би невинний

він не виглядав, міг би містити сценарій, який при виконанні на вашому сервері повністю відкриває ваш веб-сайт.

Якщо у вас є форма для завантаження файлів, тоді вам потрібно ставитися до всіх файлів з великою підозрою. Якщо ви дозволяєте користувачам завантажувати зображення, ви не можете розраховувати на розширення файлу або тип mime, щоб перевірити, що файл є зображенням, оскільки вони легко підробляються. Навіть відкриття файлу та читання заголовка чи використання функцій для перевірки розміру зображення не є надійною. Більшість форматів зображень дозволяють зберігати розділ коментарів, який може містити PHP-код, який може бути виконаний сервером.

То що ви можете зробити, щоб цього не допустити? Зрештою, ви хочете, щоб користувачі не могли виконувати будь-який завантажений ними файл. За замовчуванням веб-сервери не намагаються виконувати файли з розширеннями зображень, але не покладаються тільки на перевірку розширення файлу як файлу з ім'ям image.jpg.php, як відомо.

Деякі варіанти – перейменувати файл для завантаження, щоб забезпечити правильне розширення файлу, або змінити дозволи файлу, наприклад, chmod 0666, щоб він не міг бути виконаний. Якщо ви використовуєте * nix, ви можете створити .htaccess файл (див. лістинг 3.1), який дозволить отримати доступ лише до встановлених файлів, що запобігають нападу подвійного розширення, згаданому раніше [20].

Лістинг 3.1 – Створення файлу .htaccess

```
deny from all
    <Files ~ "^\w+\.(gif|jpe?g|png)$">
    order deny,allow
    allow from all
    </Files>
```

Зрештою, рекомендованим рішенням є взагалі не допускати прямого доступу до завантажених файлів. Таким чином, будь-які файли, завантажені на ваш веб-сайт, зберігаються у папці поза веб-коренею чи в базі даних у вигляді краплі. Якщо ваші файли не доступні безпосередньо, вам потрібно буде створити сценарій, щоб отримати файли з приватної папки (або обробник HTTP в .NET) та доставити їх у браузер. Зображення тегів підтримують атрибут src, який не є прямою URL-адресою до зображення, тому ваш атрибут src може вказувати на ваш сценарій доставки файлів за умови встановлення правильного типу вмісту в заголовку HTTP.

Лістинг 3.2 – Вміст заголовку HTTP

```
<img src="/imageDelivery.php?id=1234" />
<?php
      // imageDelivery.php
           // Fetch image filename from database based on $_GET["id"]
      ...
           // Deliver image to browser
       Header('Content-Type: image/gif');
      readfile('images/'.$fileName);

?>
```

Більшість хостинг-провайдерів мають справу з налаштуванням сервера для вас, але якщо ви розміщуєте свій веб-сайт на своєму власному сервері, ви хочете перевірити кілька речей [21].

Переконайтеся, що у вас встановлений брандмауер та блокуються всі неістотні порти. Якщо можливо, встановлення DMZ (Демілітаризована зона), що дозволяє лише доступ до портів 80 та 443 із зовнішнього світу. Хоча це може бути неможливим, якщо у вас немає доступу до вашого сервера з внутрішньої мережі, оскільки вам потрібно буде відкрити порти, щоб

дозволити завантажувати файли та віддалено входити на ваш сервер через SSH або RDP.

Якщо ви дозволяєте завантажувати файли з Інтернету, використовуйте лише безпечні способи транспортування на ваш сервер, такі як SFTP або SSH.

Якщо можливо, ваша база даних працює на іншому сервері, ніж веб-сервер. Це означає, що до сервера баз даних не можна отримати доступ безпосередньо із зовнішнього світу, лише ваш веб-сервер може отримати доступ до нього, мінімізуючи ризик впливу ваших даних.

Нарешті, не забувайте про обмеження фізичного доступу до вашого сервера.

3.8 Використання HTTPS

HTTPS – це протокол, що використовується для забезпечення безпеки через Інтернет. HTTPS гарантує, що користувачі розмовляють із сервером, якого вони очікують, і що ніхто більше не може перехопити або змінити вміст, який вони бачать у дорозі.

Якщо у вас є що-небудь, що ваші користувачі можуть захотіти приватним, настійно рекомендується використовувати лише HTTPS для його доставки. Звичайно, це означає, що кредитна картка та сторінки для входу (та URL-адреси, які вони надсилають), але зазвичай також набагато більше вашого сайту. Форма для входу часто встановлює, наприклад, файл cookie, який надсилається разом з будь-яким іншим запитом на ваш сайт, який робить користувач, який увійшов у систему, і використовується для автентифікації цих запитів. Зловмисник, який викраде це, зможе ідеально імітувати користувача та перейняти його сеанс входу. Щоб перемогти подібні атаки, ви майже завжди хочете використовувати HTTPS для всього свого сайту.

Це вже не так хитро або дорого, як це було колись. Let's Encrypt надає абсолютно безкоштовні та автоматизовані сертифікати, які вам знадобляться для ввімкнення HTTPS, і є наявні інструменти спільноти, доступні для

широкого спектру загальних платформ і рамок, щоб автоматично налаштувати це для вас.

Зокрема, Google оголосив, що вони піднімуть вас у рейтингу пошуку, якщо ви використовуєте HTTPS, надаючи цьому й перевагу SEO. Невпевнений HTTP вже виходить, і тепер настав час оновлення.

Вже використовуєте HTTPS скрізь? Подивіться далі і подивіться на налаштування суворої безпеки HTTP-транспорту (HSTS) – простий заголовок, який ви можете додати до відповідей вашого сервера, щоб заборонити небезпечний HTTP для всього вашого домену [22].

3.9 Засоби безпеки веб-сайту

Як тільки ви думаєте, що зробили все, що можете, тоді прийшов час перевірити безпеку вашого веб-сайту. Найефективніший спосіб зробити це через використання деяких інструментів безпеки веб-сайту, які часто називають тестуванням на проникнення або коротким тестуванням пера.

Існує багато комерційних та безкоштовних продуктів, які допоможуть вам у цьому. Вони працюють на аналогічній основі з хакерами скриптів, оскільки вони перевіряють усі відомі подвиги та намагаються скомпрометувати ваш сайт, використовуючи деякі з попередніх згаданих методів, таких як SQL Injection.

Деякі безкоштовні інструменти, які варто переглянути:

– Netsparker (доступна безкоштовна версія спільноти та пробна версія). Добре підходить для тестування SQL ін'єкції та XSS.

– OpenVAS претендує на найсучасніший сканер безпеки з відкритим кодом. Добре підходить для тестування відомих уразливостей, наразі сканує понад 25 000. Але це може бути важко налаштовано, і для цього потрібно встановити сервер OpenVAS, який працює лише на * nix. OpenVAS є виделкою Nessus до того, як він став комерційним продуктом із закритим кодом.

– SecurityHeaders.io (безкоштовна онлайн-перевірка). Інструмент для швидкого повідомлення про те, які згадані вище заголовки безпеки (такі як CSP і HSTS) домену ввімкнув і правильно налаштувати.

– Xenotix XSS Exploit Framework Інструмент від OWASP (Open Web Security Security Project), який включає в себе величезний вибір прикладів атаки XSS, який можна запустити, щоб швидко підтвердити, чи вразливі дані вашого сайту в Chrome, Firefox та IE.

Результати автоматизованих тестів можуть бути приголомшливими, оскільки вони представляють безліч потенційних проблем. Важливим є спочатку зосередитись на критичних питаннях. Кожен випуск, про який повідомляється, зазвичай має чітке пояснення потенційної вразливості. Ви, ймовірно, виявите, що деякі середні / низькі проблеми не стосуються вашого сайту [23].

Є кілька додаткових кроків, які ви можете вжити вручну, щоб спробувати порушити свій сайт, змінивши значення POST / GET. Тут може допомогти проксі-сервер налагодження, оскільки він дозволяє перехоплювати значення HTTP-запиту між вашим браузером та сервером. Популярна безкоштовна програма під назвою Fiddler – хороша відправна точка.

То що б ви намагалися змінити у запиті? Якщо у вас є сторінки, які повинні бути видимими лише зареєстрованому користувачеві, спробуйте змінити параметри URL-адреси, такі як ідентифікатор користувача або значення cookie, намагаючись переглянути деталі іншого користувача. Інша область, яку варто перевірити, – це форми, зміни значення POST для спроби подати код для виконання XSS або завантаження сценарію на стороні сервера.

4 ПРАКТИЧНЕ ВИКОРИСТАННЯ ПРОВЕДЕНИХ ДОСЛІДЖЕНЬ

4.1 Класифікація сканерів вразливості веб–застосувань

На даний момент в компаніях майже не замислюються про безпеку інформаційних сторінок в мережі Інтернет, і практично не виділяють цьому питанню уваги, зокрема часто починають частіше вживати заходів тоді коли відбувся витік важливої інформації.

Щоб усунути цю проблему, яка пов’язана з захистом інформації і захистити веб-сайт від зловмисників, потрібно розглянути захист бази даних всередині мережі, а також провести моніторинг мережі. Моніторинг мережі і діагностику можна провести за допомогою сканерів мережі, або за допомогою спеціалізованого обладнання [24].

Сканери в свою чергу діляться на групи:

- сканери великих мереж;
- сканери вразливостей веб-додатків.

Запропоновані програми дають можливості і засоби для ефективного захисту і управлінням завад вразливостям,

4.2 Сканер визначення небезпек Nikto-online

Сканер веб-сервера, який виконує комплексні тести на веб-серверах для декількох предметів, включаючи понад 6700 потенційно небезпечних файлів / програм, перевіряє наявність застарілих версій понад 1250 серверів та проблем, пов’язаних з версією, на понад 270 серверах. Ця програма робить перевірку елементів конфігурації сервера, зокрема, як наявність декількох файлів індексу, параметри HTTP-сервера та намагається встановити встановлені веб-сервери та програмне забезпечення.

Вивчіть веб-сервер, щоб знайти потенційні проблеми та вразливості

безпеки, включаючи:

- Неправильні налаштування сервера та програмного забезпечення.
- Файли та програми за замовчуванням.
- Небезпечні файли та програми.
- Застарілі сервери та програми.

Для сканування потрібен час (до декількох годин).

4.2.1 Робота сканера Nikto-Online

Для підвищення безпеки веб-сайту було використано сканер Nikto-Online. Під час того коли відбувається сканування сайту, зокрема відбувається генерація 2-ч файлів, в яких записуються результати сканування. Зокрема це текстовs файлb *.txt і *.pdf/

Проте коли відбувається сканування система проводить аналіз результатів, і тоді можна розділити небезпечні вразливості.

Проведемо аналіз структури бази даних, яка наведена на рисунку 4.1

Проаналізуємо структуру бази даних наведеної на рисунку 4.1. База даних системи являться, як об’єднана база даних OSVDB.

Для роботи системи нам необхідно було внести наступні зміни до OSVDB базу вразливостей [25]:

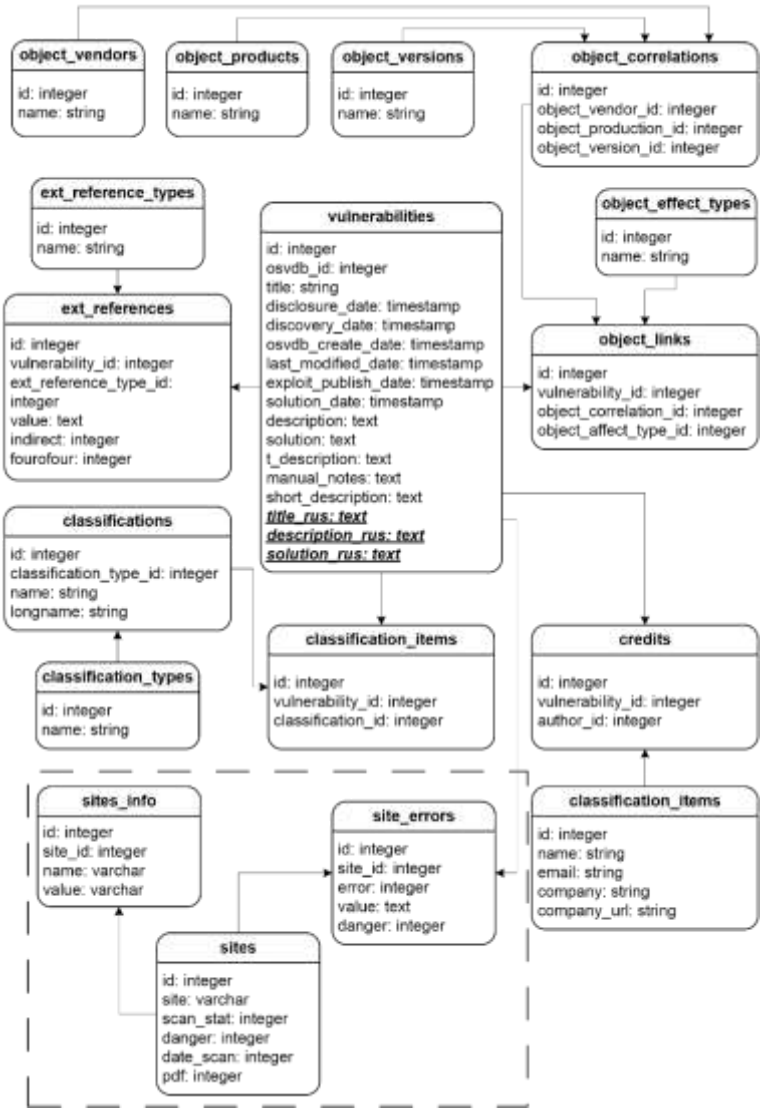

Рисунок 4.1 – Структура БД модифікованого сканера Nikto-Online

– *«site_errors»* містить помилки сайтів і помітку про небезпечні уразливості.

4.2.2 Інтерфейс сканера

Розглянемо інтерфейс і зовнішній вигляд сканера, а також його роботу і структуру. Даний сканер складається з сторінки, яка представлена на русунку 4.2.

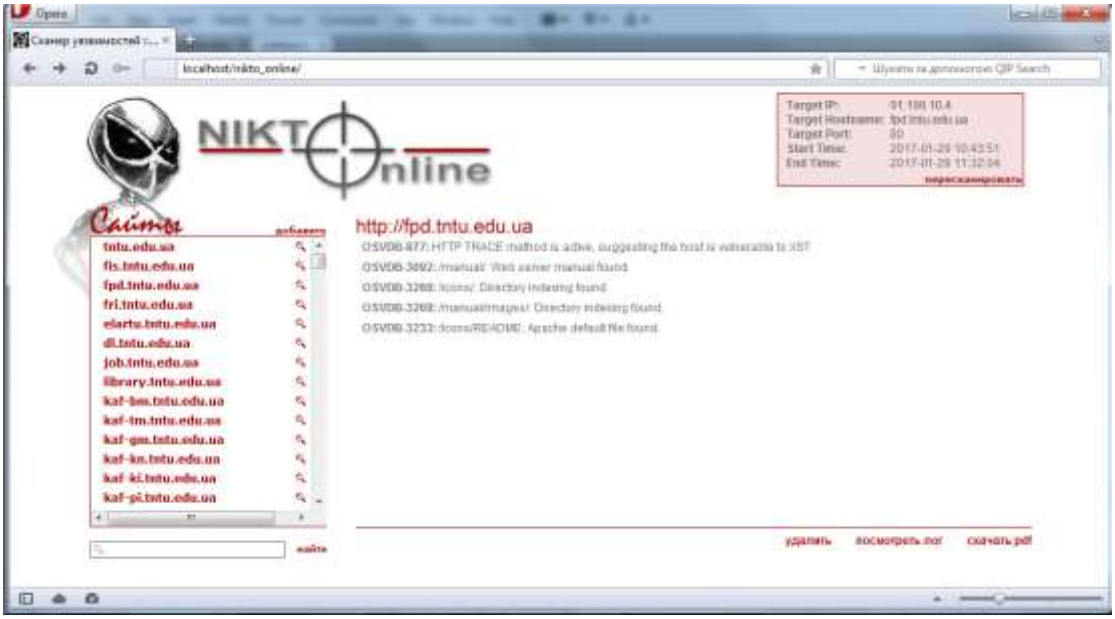

Рисунок 4.2 – Інтерфейс сканера Nikto-Online

З правої сторони показано сайти, які ми додали раніше, а також показано форму для пошуку і додавання сайтівю

Додавання Сайту.

Рисунок 4.3 – Додавання сайту

Після додавання сайту відкривається форма, яка зображена на рисунку 4.4:

Рисунок 4.4 – Форма додавання сайту

Під час спроби намагання додати сайт, який вже був доданий раніше, в свою чергу система автоматично перейде на перелік вразливостей сайту, що

додається.

На рисунку 4.5 показано, як можна пересканувати сайти.

Рисунок 4.5 – Блок з інформацією про сайт

На рисунку 4.6 показано «Перехід між сайтами»

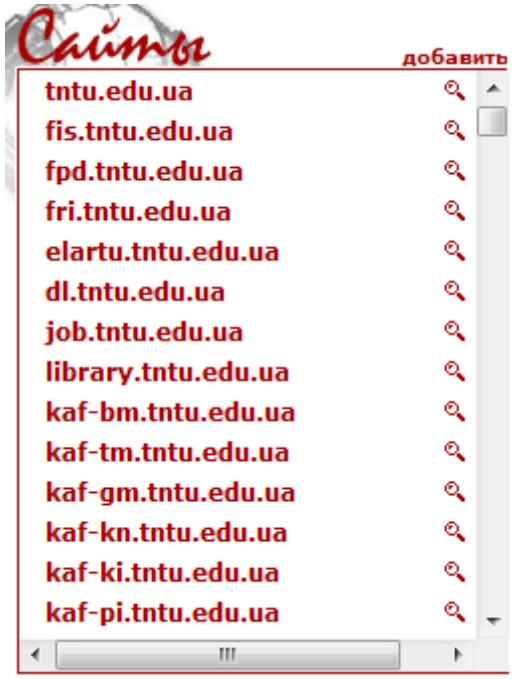

Рисунок 4.6 – Блок навігації по сайтам

Власне кажучи для покращення навігації існує пошук по сайтам.

У другому випадку буде обраний перший-ліпший сайт, який задовольняє умову запиту [26].

4.2.3 Результати сканування

Розглянемо, яким чином відображаються результати сканування.

Лістинг 4.1 – Результати сканування

```
      - Nikto Online
      ---------------------------------------------------------------
---------------
      + Target IP:          91.198.10.4
      + Target Hostname:    fpd.tntu.edu.ua
      + Target Port:        80
      + Start Time:         2012-07-29 10:43:51
      ---------------------------------------------------------------
-
      + Server: Apache/2.2.3 (Red Hat)
      + Number of sections in the version string differ from
those in the database, the server reports: apache/2.2.3 while
the database has: 2.2.16. This may cause false positives.
      + Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
      + OSVDB-877: HTTP TRACE method is active, suggesting the
host is vulnerable to XST
      + OSVDB-3092: /manual/: Web server manual found.
      + OSVDB-3268: /icons/: Directory indexing found.
      + OSVDB-3268: /manual/images/: Directory indexing found.
      + OSVDB-3233: /icons/README: Apache default file found.
      + 6417 items checked: 128 error(s) and 7 item(s) reported
on remote host
      + End Time:           2012-07-29 11:32:04 (2893 seconds)
      ---------------------------------------------------------------
-
      + 1 host(s) tested

**********************************************************************
**
      Portions of the server's ident string (Apache/2.2.3) are
not in the Nikto database or is newer than the known string.
Would you like to submit this information (*no server specific
data*) to CIRT.net
```

Після цього відкриваєм звіт, який показано на рисунку 4.7

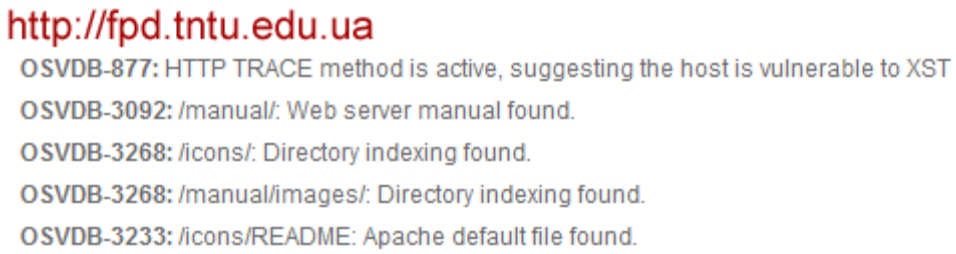

Рисунок 4.7 – Результат сканування сайту

На завершення можна сказати, що при розробці системи, було задіяно декілька технологій, що дає можливість взаємодіяти одна з одною.

4.3 Висновки до четвертого розділу

На даний момент в компаніях майже не замислюються про безпеку інформаційних сторінок в мережі Інтернет, і практично не виділяють цьому питанню уваги, зокрема часто починають частіше вживати заходів тоді коли відбувся витік важливої інформації.

Щоб усунути цю проблему, яка пов’язана з захистом інформації і захистити веб-сайт від зловмисників, потрібно розглянути захист бази даних всередині мережі, а також провести моніторинг мережі. Моніторинг мережі і діагностику можна провести за допомогою сканерів мережі, або за допомогою спеціалізованого обладнання.

5 СПЕЦІАЛЬНА ЧАСТИНА

В даному розділі проаналізуємо інші сканери вразливостей веб-сайтів.

5.1 Сканер Shadow Security

Цей сканер виявляє проломи в системі безпеки серверів як на платформі * Unix, так і на платформі Windows. Shadow Security Scanner має унікальну технологію виявлення можливих дірок в системі. На даний момент база складає майже 10000 аудитів. В неї входить аудит таких модулів як TCP / IP, UDP, FTP, DNS, SMTP, POP3, HTTP, CGI,

NetBIOS, Registry, Users accounts, Password checks, Services, LDAP, DoS атаки і багато іншого. Також є генератор звітів. Програма підтримує російську мову. Інтерфей сканера представлено на рисунках 5.1 та 5.2.

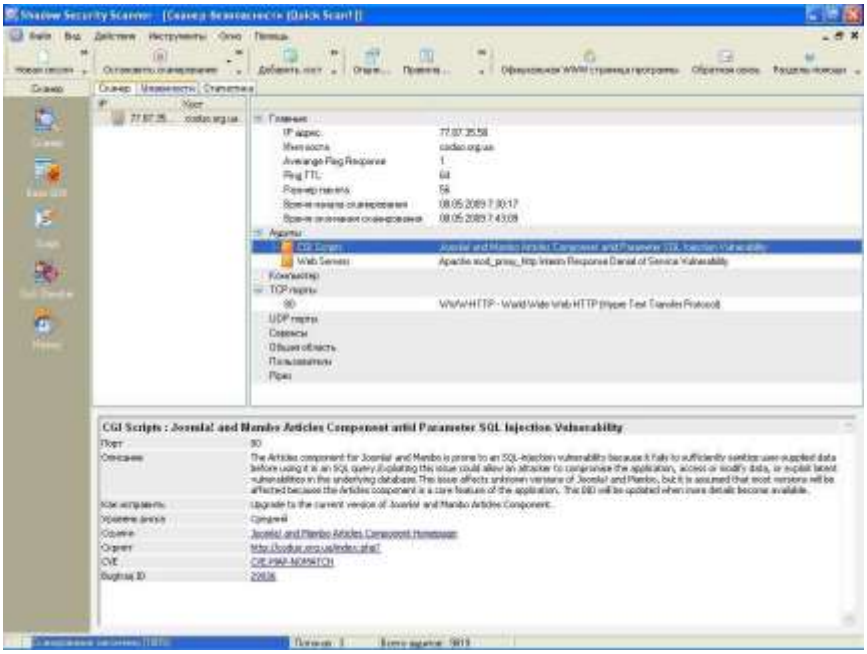

Рисунок 5.1 – Приклади використання Shadow Security Scanner

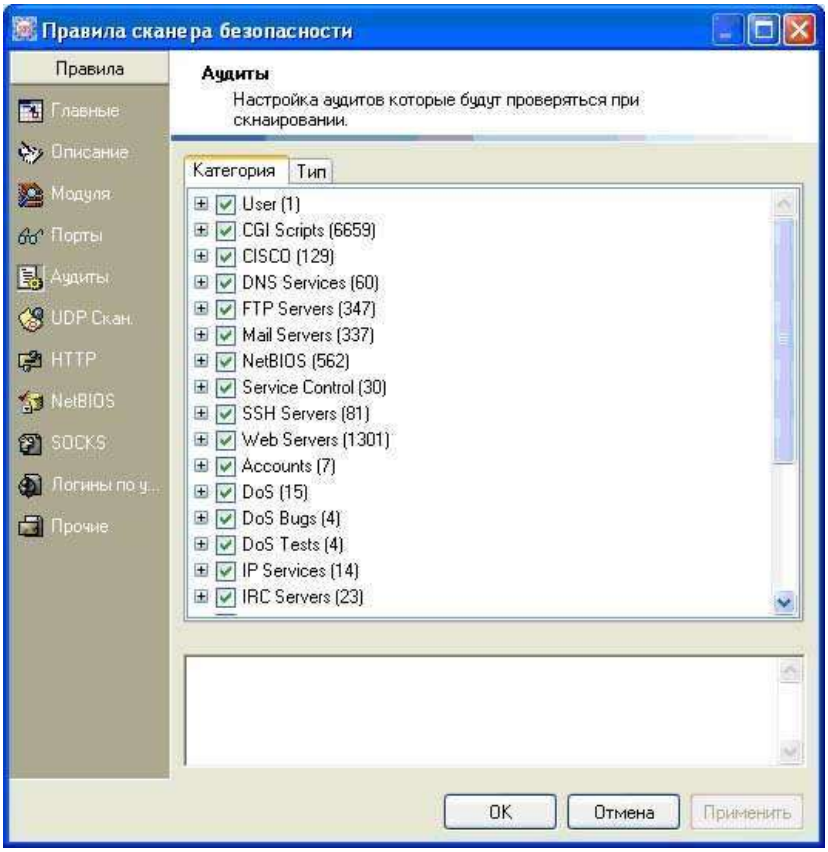

Рисунок 5.2 – Налаштування правил сканування

5.2 Acunetix Web Vulnerability Scanner

Компанія Acunetix – всесвітній лідер захисту веб-додатків. Компанія стала першовідкривачем у сфері дослідження технологій захисту веб-додатків. Ще з 1997 року компанія сконцентрувала свою діяльність на захисті веб-сайтів і поступово вдосконалила технічні переваги у тестуванні веб-сайтів та виявленні вразливих місць у системі їх захисту. Acunetix Web Vulnerability Scanner має такі інноваційні характеристики:

– автоматичний аналізатор JavaScript, що робить безпечне тестування додатків Ajax і Web 2.0;

– найбільш удосконалене тестування SQL ін'єкцій та Cross site scripting;

– visual macro recorder, що дозволяє побачити, в яких саме місцях були знайдені помилки полегшує роботу над веб-бланками і сторінками, захищеними паролями;

– програми докладних звітів, включаючи звіти відповідають стандартам;

– VISA PCI;

– багатопотоковий сканер, здатний з блискавичною швидкістю ретельно перевірити сотні тисяч сторінок;

– пошуковий агент, який визначає тип веб-сервера і мову програми;

– дослідження та аналіз вмісту веб-сайтів, включаючи flash content протокол SOAP і AJAX.

Acunetix робить перевірку всіх вразливих місць веб-сайту, включаючи перевірку

SQL ін'єкцій, Cross site scripting та інших вразливих місць у системі захисту веб-сайтів.

Для виявлення даних вразливостей, необхідний комплексний двигун. Основним у процесі перевірки сайта є не тільки кількість вразливостей, які сканер здатний виявити, а також комплексність і ретельність запуску різноманітних SQL ін'єкцій, Cross site scripting та інших хакерських атак. Acunetix має вдосконалений програмний код, що дозволяє з високою швидкістю і з найменшою вірогідністю помилкового результату знайти вразливі місця в системі захисту веб-сайту. Також він дозволяє виявити CRLF ін'єкції, Code execution, Directory Traversal, File inclusion та вразливості при аутентифікації.

Перевірка AJAX та технологій Web 2.0 на уразливості. Сучасний аналізатор JavaScript дозволяє всебічно перевірити новітні та найбільш складні AJAX та Web 2.0 Веб-додатки та знайти вразливі місця.

Детальні звіти, що дозволяють знайти відповідності прийнятим стандартам. Acunetix Web Vulnerability Scanner має схему докладних звітів,

які крім усього також показують, чи відповідають Ваші Веб-додатки новим вимогам і стандартам VISA PCI.

Перевірка Вашого веб-сайту на захищеність від The Google Hacking Database. The Google Hacking Database (GHDB) - база запитів, якими користуються хакери для виявлення незахищених даних на Вашому веб-сайті. До них відносяться сторінки реєстрацій на порталах, файли подій і так далі. Acunetix запускає GHDB запити на Ваш веб-сайт і розпізнає незахищені дані або дані, схильні до використання до того, як це зробить пошукова програма хакера.

Можливість налаштування програми тестування. Окрім програми автоматичного сканування, Acunetix також має вдосконалені сервісні модулі, що дозволяють тестувальником з точністю налаштувати перевірку безпеки Веб-додатків:

– HTTP Editor – за допомогою цієї сервісної програми Ви зможете без особливих зусиль зробити HTTP/HTTPS запити і проаналізувати реакцію Інтернет-сервера;

– HTTP Sniffer – зупиняти, розпізнавати і модифікувати весь мережевий трафік HTTP/HTTPS і виявляти всю інформацію, надіслану Вебдодатком

– HTTP Fuzzer – виконує всебічну перевірку переповнення буфера і правильності введення даних. Легко налаштовуються правила http Fuzzer дозволяють тестувати тисячі вхідних змінних. Тестування, які зазвичай займають цілі дні, зараз можуть бути виконані протягом декількох хвилин.

– також можна створити схему індивідуальної атаки або ж модифіковані вже існуючу за допомогою Web Vulnerability Editor.

Тестування сторінок, захищених паролями, а також веб-форм за допомогою програми автоматичного заповнення веб-форм HTML. Acunetix Web Vulnerability Scanner автоматично заповнює веб-форми і поля вебреєстрації. Більшість інших сканерів нездатні цього зробити або вимагають написання складних програм для тестування таких сторінок. Але з Acunetix все

набагато простіше: за допомогою макро записуючого пристрою Ви зможете записати реєстрацію або процес заповнення форм і зберегти послідовність. Під час процесу тестування сканер знову відтворить цю послідовність, а також автоматично заповнить веб-форму або увійде на сторінку, захищену паролем.

Вдосконалені характеристики:

– тестуючі профілі без особливих зусиль перевіряють веб-сайти з різними скануючими опціями і особливостями;

– розвинена система звітності;

– порівняння і пошук відмінностей від результатів сканування попередніми версіями;

– програма без особливих зусиль здійснює додаткові перевірки змін вебсайту;

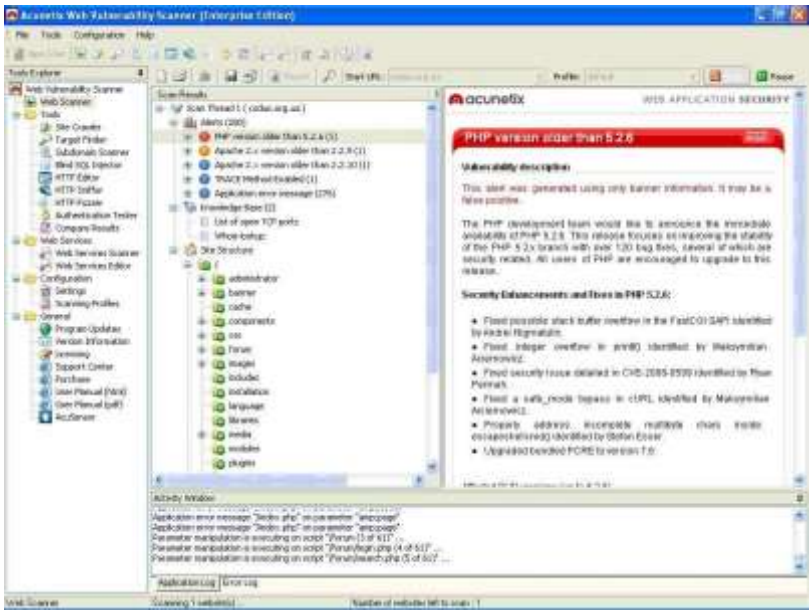

Рисунок 6.3 – Інтерфейс Acunetix Web Vulnerability Scanner

– виявлення каталогів (директорій), доступ до яких є небажаним;

– програма знаходить популярні Веб-додатки (наприклад, форуми, кошики для віртуальних покупок) і виявляє уразливі версії;

– визначення, небезпечних HTTP методів, активовані на веб-сервері.

5.3 Сканер Nessus

Сканер представляє з себе клієнт-серверну архітектуру. Відповідно до проведеного порталом securitylab.ru опитуванням, nessus використовують 17% респондентів.

Рисунок 5.4 – Інтерфейс сканера Nessus

Перш за все використовується для сканування портів і визначає сервіси, що використовують їх. Так само проводиться перевірка сервісів по базі вразливостей. Для тестування вразливостей використовуються спеціальні плагини, написані на мові NASL (Nessus Attack Scripting Language).

База вразливостей оновлюється щотижня, проте для комерційних абонентів є можливість завантажувати нові плагіни без семиденної затримки.

При відключеній опції "safe checks" деякі тести на уразливості, що використовуються nessus можуть привести до порушень в роботі сканованих систем. Ранні версії поширюються за ліцензією GPL. Нові версії є комерційним продуктом.

5.4 Сканер вразливостей XSpider 7

XSpider може в повністю автоматичному режимі перевіряти комп'ютери

і сервіси в мережі на предмет виявлення вразливостей. База вразливостей постійно поповнюється фахівцями Positive Technologies, що в сумі з автоматичним оновленням баз і модулів програми постійно підтримує актуальність версії XSpider.

XSpider може виконувати перевірки за розкладом. Таким чином, налаштувавши планувальник XSpider, автоматичне оновлення і надсилати звіти про результати перевірки поштою або їхнє збереження на мережному диску, можна значно полегшити процес виявлення вразливостей. Це дозволить сконцентрувати свою увагу на боротьбі з уже виявленими уразливими місцями та оновленні програмного забезпечення. У цьому XSpider надає так само неоціненну допомогу, виводячи до звіту про результати перевірки не тільки інформацію про знайдену уразливості, але і посилання, наприклад, на статті на сайті Microsoft, які описують виявлену XSpider вразливість і дають рекомендації по її усуненню.

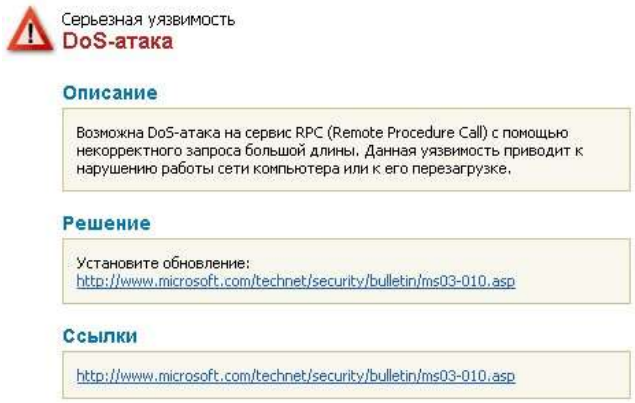

Рисунок 5.5 – Повідомлення про DoS–атаку

Кожне налаштоване завдання зберігається у файлі. Якщо запланований запуск завдання з Планувальника, то результат її виконання буде збережений у файлі .tsk, який може бути відкритий у будь-який час за допомогою XSpider. У файлі зберігається результат не тільки останньої перевірки хоста або хостів, а вся історія перевірок. Таким чином, можна контролювати зміни рівня безпеки після ліквідації виявлених раніше вразливостей і поновлення операційних

систем і сервісів. Приклад завантаженого завдання показаний на рисунку нижче.

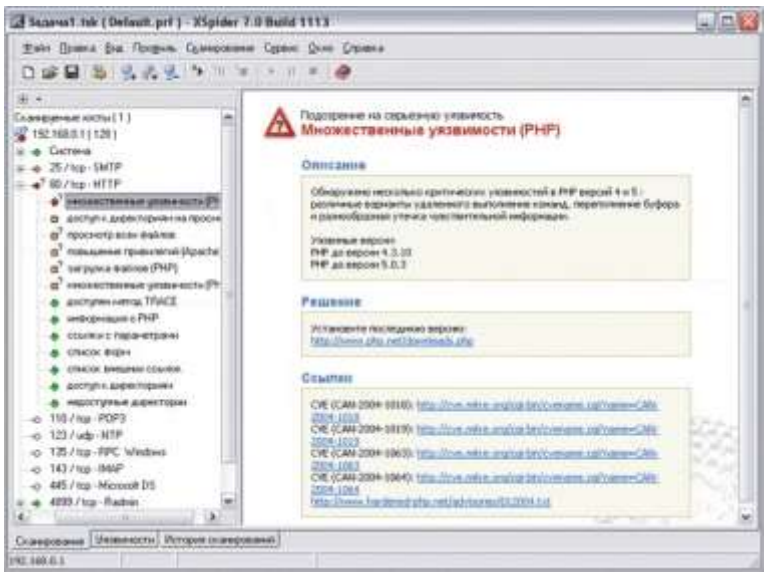

Рисунок 5.6 – Приклад завантаженого завдання

У даному прикладі на Windows XP SP3 з увімкненим файрволом був встановлений Apache, PHP, mySQL, безкоштовний скрипт для підписки на новини, демоверсія поштового сервера MERAK Mail Server (SMTP/ESMTP/POP3/IMAP4) з модулем віддаленого управління і налаштуваннями за замовчуванням, Remote Administrator 2.2 від компанії FamaTech. Для цієї перевірки був створений профіль, в якому було включено сканування всіх портів хоста і пошук всіх можливих вразливостей. В результаті аудиту було виявлено багато нарікань до скрипту підписки на новини і був виявлений паблікрелей на поштовому сервері (так як настроювання сервера після його встановлення не виконувалася). Всі інші сервіси, які працюють на тестовому комп'ютері, були виявлені і безпомилково ідентифіковані.

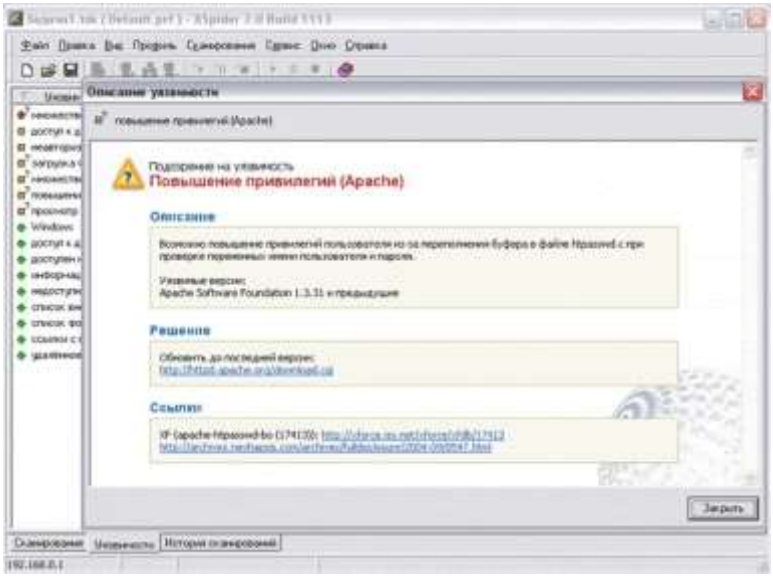

Рисунок 5.7 – Результати сканування

У результатах перевірки, крім інформації про виявлені вразливості, були наведені посилання на опис уразливості на сайтах, що спеціалізуються на безпеці і дані посилання на завантаження оновлених версій програмного забезпечення.

У другому тесті аудиту був підданий хост з ОС Windows XP без сервіс-пака з відключеним брандмауером. Вкладка Вразливості головного вікна XSpider показана на рисунку нижче.

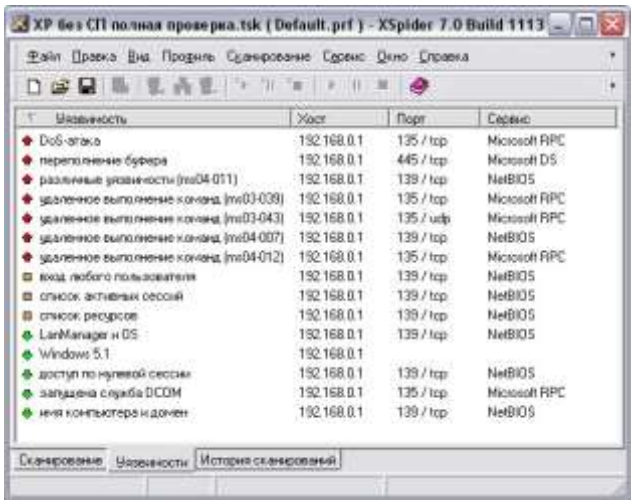

Рисунок 5.8 – Вкладка «Уязвимости»

У ході сканування було виявлено кілька критичних вразливостей. У

результатах роботи були дані посилання статті в Базі знань Microsoft, які описують виявлену уразливість і посилання для завантаження виправлень, що усувають ці уразливості.

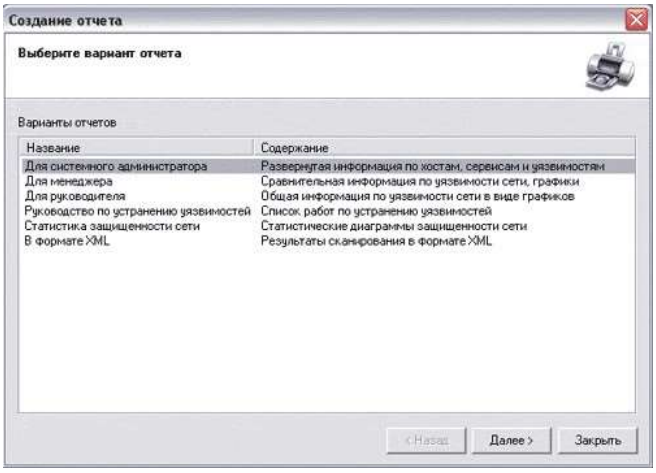

Рисунок 5.9 – Вибір варіантів представлення звіту

XSpider пропонує на вибір декілька стандартних типів звітів про результати перевірки.

5.5 Висновки до шостого розділу

В розділі проведено аналіз альтернативних сканерів вразливостей веб-сайтів.

6 ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Метою цього розділу дипломної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності від дослідження систем захисту веб-сайтів, а також прийняття рішення щодо подальшого розвитку і впровадження або ж недоцільність впровадження відповідної розробки.

Для здійснення оцінки потрібно зробити розрахунки трудомісткості кожної операції, що мала місце при проведенні наукових досліджень.

6.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

Витрати часу по окремих операціях технологічного процесу відображені в таблиці 6.1.

Таблиця 6.1 – Операції технологічного процесу та час їх виконання

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1.	Витрати праці на підготовку опису задачі	інженер	15
2.	Витрати праці на розробку проекту	інженер	35
3.	Витрати праці на розробку структури системи	інженер	25
4.	Витрати праці на створення системи по вибраному проекту та структурі	інженер	105
5.	Витрати праці на підготовку документації	інженер	25

Продовження таблиці 6.1

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
6.	Витрати праці на відлагодження роботи спроектованої системи при комплексній відладці	інженер	20
Разом			**225**

Сумарний час на проведення науково-дослідної роботи становить 225 годин.

6.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України “Про оплату праці” заробітна плата – це “винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу”.

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

При розрахунку заробітної плати кількість робочих днів у місяці слід в середньому приймати – 24,5 дні/міс., або ж 196 год./міс. (тривалість робочого дня – 8 год.).

Місячний оклад кожного працівника слід враховувати згідно існуючих на даний час тарифних окладів. Рекомендовані тарифні ставки: керівник проекту – 42,5...74,0 грн./год., інженер – 30,0...50,0 грн./год., консультант – 33,5...51,5 грн./год., технік – 30,0...43,5 грн./год., лаборант – 22,0...35 грн./год.

Основна заробітна плата розраховується за формулою:

$$З_{осн.} = Т_с \cdot К_г, \tag{6.1}$$

де $Т_с$ – тарифна ставка, грн.;

$К_г$ – кількість відпрацьованих годин.

Оскільки всі види робіт в даному дослідженні виконує інженер, то основна заробітна плата буде розраховуватись тільки за однією формулою

$$З_{осн.} = 50 \cdot 225 = 11250 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати.

$$З_{дод.} = З_{осн.} \cdot К_{допл.}, \tag{6.2}$$

де $К_{допл.}$ – коефіцієнт додаткових виплат працівникам, 0,1–0,15 (візьмемо його рівним 0,15).

$$З_{дод.} = 11250 \cdot 0{,}15 = 1687{,}5 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{о.п.}$) визначаються за формулою:

$$B_{о.п.} = З_{осн.} + З_{дод.}. \quad (6.3)$$

$$B_{о.п.} = 11250 + 1687,5 = 12937,5 \text{ грн.}$$

Крім того, слід визначити відрахування на соціальні заходи:

1) фонд страхування на випадок безробіття – 1,3 %;
2) фонд по тимчасовій втраті працездатності – 2,9 %;
3) пенсійний фонд – 32,3 %.

У сумі зазначені відрахування становлять 37,5 %.

Отже, сума відрахувань на соціальні заходи буде становити:

$$B_{с.з.} = ФОП \cdot 0,375, \quad (6.4)$$

де $ФОП$ – фонд оплати праці, грн.

$$B_{с.з.} = 12937,5 \cdot 0,375 = 4851,56 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці зведемо у таблицю 6.2.

Таблиця 6.2 – Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівник-ків	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахув. на ФОП, грн.	Всього витрати на оплату праці, грн. *6=3+4+5*
		Тарифна ставка, грн.	К-сть відпра-цьов. год.	Фактично нарах. з/пл., грн.			
А	*Б*	*1*	*2*	*3*	*4*	*5*	*6*
1	інженер	50	225	11250	1687,5	4851,56	17789,06

Сумарні затрати на оплату праці становлять 17789,06 грн.

6.3 Розрахунок матеріальних витрат

Матеріальні витрати визначаються як добуток кількості витрачених матеріалів та їх ціни:

$$M_{B\,i} = q_i \cdot p_i, \tag{6.5}$$

де: q_i – кількість витраченого матеріалу *i*-го виду;

p_i – ціна матеріалу *i*-го виду.

Звідси, загальні матеріальні витрати можна визначити:

$$З_{м.в.} = \sum M_{B\,i}. \tag{6.6}$$

Проведені розрахунки занесемо у таблицю 6.3.

Таблиця 6.3 – Зведені розрахунки матеріальних витрат

Найменування матеріальних ресурсів	**Одиниця виміру**	**Норма витрат**	**Ціна за одиницю, грн**	**Затрати матеріалів, грн**	**Транспортно-заготівельні витрати, грн**	**Загальна сума витрат на матеріали, грн**
1. Основні матеріали						
Програмне забезпечення різного рівня	комплект	1	21459	–	–	21459
Разом:						21459

Сумарні матеріальні затрати становлять 21459 грн.

6.4 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$З_е = W \cdot T \cdot S \,, \tag{6.7}$$

де W – необхідна потужність, кВт;

Т – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Вартість кіловат-години електроенергії слід приймати згідно існуючих на даний час тарифів (0,714 грн. з ПДВ за 1 кВт).

Потужність комп'ютера для проведення дослідження – 750 Вт, кількість годин роботи обладнання згідно таблиці 6.1 – 225 годин.

Тоді, $З_е = 0{,}75 \cdot 225 \cdot 0{,}714 = 120{,}49$ грн.

6.5 Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %).

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$А = \frac{Б_В \cdot Н_А}{100\%}, \tag{6.8}$$

де A – амортизаційні відрахування за звітний період, грн.;

$Б_В$ – балансова вартість групи основних фондів на початок звітного періоду, грн.;

$Н_А$ – норма амортизації, %.

Для даного дослідження засобом роботи є комп'ютер. Його сума становить 10255 грн. Отже, амортизаційні відрахування будуть рівні:

$$A = \frac{10255 \cdot 5\%}{100\%} = 512{,}75 \text{ грн.}$$

Оскільки робота виконувалась 225 годин, то амортизаційні відрахування будуть становити:

$$A = \frac{512{,}75 \cdot 225}{150} = 769{,}13 \text{ грн.}$$

6.6 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління спілкою та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_в = B_{о.п.} \cdot 0{,}2...0{,}6, \qquad (6.9)$$

де $Н_В$ – накладні витрати.

Отже, накладні витрати:

$$H_в = 12937,5 \cdot 0,3 = 3881,25 \text{ грн.}$$

6.7 Складання кошторису витрат та визначення собівартості НДР

Результати проведених вище розрахунків зведемо у таблицю 6.4.

Таблиця 6.4 – Кошторис витрат на НДР

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	12937,5	29,39
Відрахування на соціальні заходи	4851,56	11,02
Матеріальні витрати	21459	48,75
Витрати на електроенергію	120,49	0,27
Амортизаційні відрахування	769,13	1,75
Накладні витрати	3881,25	8,82
Собівартість	**44018,93**	**100**

Собівартість ($C_В$) НДР розрахуємо за формулою:

$$C_В = B_{о.п.} + B_{с.з.} + З_{м.в.} + З_е + A + H_в. \quad (6.10)$$

Отже, собівартість дослідження дорівнює:

$$C_В = 12937,5 + 4851,56 + 21459 + 120,49 + 769,13 + 3881,25 = 44018,93 \text{ грн.}$$

В результаті проведених розрахунків собівартість науково-дослідної роботи становить 44018,93 грн.

6.8 Розрахунок ціни науково-дослідної роботи

Ціну НДР можна визначити за формулою:

$$Ц = \frac{C_B \cdot (1 + P_{рен}) + K \cdot B_{н.і.}}{K} \cdot (1 + ПДВ), \qquad (6.11)$$

де $P_{рен.}$ – рівень рентабельності, 30 %;

K – кількість замовлень, од. (встановлюється лише при розробці програмного продукту та мікропроцесорних систем);

$B_{н.і.}$ – вартість носія інформації, грн. (встановлюється лише при розробці програмного продукту);

ПДВ – ставка податку на додану вартість, (20 %).

Оскільки розробка є прикладною, і використовуватиметься тільки для одного підприємства, то для розрахунку ціни не потрібно вказувати коефіцієнти K та $B_{і.н}$, оскільки їх в даному випадку не потрібно.

Тоді, формула для обчислення ціни НДР буде мати вигляд:

$$Ц = C_B \cdot (1 + P_{рен}) \cdot (1 + ПДВ). \qquad (6.12)$$

Звідси ціна на НДР складе:

$$Ц = 44018{,}93 \cdot (1 + 0{,}3) \cdot (1 + 0{,}2) = 68669{,}53 \text{ грн.}$$

6.9 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність ($E_р$) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_р = \frac{П}{C_В}, \tag{6.13}$$

де $П$ – прибуток;

$C_В$ – собівартість.

Плановий прибуток ($П_{пл}$) знаходимо за формулою:

$$П_{пл} = Ц - C_В. \tag{6.14}$$

Розраховуємо плановий прибуток:

$$П_{пл} = 68669{,}53 - 44018{,}93 = 24650{,}6 \text{ грн.}$$

Отже, формула для визначення економічної ефективності набуде вигляду:

$$E_р = \frac{П_{пл}}{Св}. \tag{6.15}$$

Тоді, $E_p = \frac{24650,6}{44018,93} = 0,559$.

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_p}, \tag{6.16}$$

Термін окупності дорівнює:

$$T_p = \frac{1}{0,559} = 1,78 \text{ року.}$$

6.10 Висновки до сьомого розділу

В цьому розділі дипломної роботи було розраховано основні техніко-економічні показники дослідження (див. таблицю 6.5).

Розраховане значення економічної ефективності становить 0,559, що є високим значенням.

Так само нормальним є термін окупності. Для даного дослідження він становить 1.78 року.

Таблиця 6.5 – Техніко-економічні показники НДР

№ п/п	Показник	Значення
1.	Собівартість, грн.	44018,93
2.	Плановий прибуток, грн.	24650,6
3.	Ціна, грн.	68669,53
4.	Економічна ефективність	0,559
5.	Термін окупності, рік	1,78

Отже, дане дослідження може бути впроваджене та мати подальший розвиток, оскільки воно є економічно вигідним за всіма основними техніко-економічними показниками.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Охорона праці

7.1.1 Дії роботодавця за результатами атестації робочих місць за умовами праці

Поняття «атестація робочих місць за умовами праці» – це комплексна оцінка всіх факторів виробничого середовища і трудового процесу, супутніх соціально-економічних факторів, що впливають на здоров'я і працездатність працівників в процесі трудової діяльності. Тому в разі перевищення норм шкідливих факторів працівники повинні отримувати пільги за перебування в цих умовах.

Атестація проводиться на підприємствах, в установах та організаціях незалежно від форм власності й господарювання, де технологічний процес, використовуване обладнання, сировина та матеріали є потенційними джерелами шкідливих і небезпечних виробничих факторів, що можуть несприятливо впливати на стан здоров'я, а також на їхніх нащадків, як тепер, так і в майбутньому [25].

Для того, щоб підприємство змогло надати, а працівники, які працюють в несприятливих умовах праці змогли отримати пільги і компенсації за роботу у цих умовах, насамперед, необхідно перевірити наявність професій (посад) у Списках виробництв, робіт, професій, посад і показників, зайнятість в яких дає право на пенсію за віком на пільгових умовах (остання редакція затверджена постановою Кабінету Міністрів України від 24 червня 2016 року № 461), Списки виробництв, цехів, професій і посад із шкідливими і важкими умовами праці, зайнятість працівників на роботах в яких дає право на щорічну додаткову відпустку, затверджений постановою Кабінету Міністрів України від 17 листопада 1997 року № 1290 (зі змінами), Перелік виробництв, цехів, професій і посад із шкідливими умовами праці, робота в яких дає право на скорочену тривалість робочого

тижня, затверджений постановою Кабінету Міністрів України від 21 лютого 2001 року № 163, Перелік робіт із важкими, шкідливими та особливо шкідливими умовами праці у будівництві, на яких встановлюється підвищена оплата праці затверджений постановою Кабінету Міністрів України від 12.07.2005 р. № 576[26].

Атестацію має проводити атестаційна комісія, склад і повноваження якої визначаються наказом по підприємству, організації в строки, передбачені колективним договором, але не рідше як один раз на п’ять років (п. 4 Порядку проведення атестації). Проте, в разі докорінної зміни умов і характеру праці з ініціативи власника або уповноваженого ним органу, профспілкового комітету, трудового колективу або його виборного органу проводиться позачергова атестація.

Дата і термін проведення чергової атестації визначаються з урахуванням того, що вона має бути завершена до закінчення терміну попередньої атестації.

Недотримання термінів проведення чергової атестації так як і не проведення атестації робочих місць призводить до порушення прав працівників – фактично, працюючи в шкідливих і важких умовах праці, наймана особа не одержить передбачених законодавством пільг, компенсацій і соціальних гарантій. Керівник підприємства несе повну відповідальність за своєчасне та якісне проведення атестації робочих місць за умовами праці (п. 4 Порядку проведення атестації робочих місць за умовами праці, затвердженого постановою Кабінету Міністрів України від 01.08.1992 № 442).

Статтею 13 ЗУ «Про охорону праці» передбачено, що роботодавець зобов’язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів та організувати, зокрема, проведення лабораторних досліджень умов праці, атестацій робочих місць на відповідність нормативно-правовим актам з охорони праці в порядку і строки, що визначаються законодавством. За їх підсумками вжити заходів

щодо усунення небезпечних і шкідливих для здоров'я виробничих факторів [25].

Несвоєчасне виконання вимог цієї статті тягне за собою притягнення роботодавця чи інших посадових осіб до адміністративної відповідальності, що виражається накладенням штрафу згідно ч. 5 ст. 41 Кодексу Законів про адміністративні порушення у розмірах передбачених статтею.

Проте, варто нагадати, що сплата штрафу не звільняє суб'єкта господарювання від проведення атестації. В будь-якому випадку атестацію обов'язково потрібно провести.

За результатами проведеної атестації атестаційна комісія складає відповідні переліки. По-перше, робочих місць, виробництв, робіт, професій і посад, працівникам яких підтверджено право на пільги і компенсації, що визначаються законодавством. По-друге, робочих місць, виробництв, робіт, професій і посад, працівникам яких пропонується встановити пільги і компенсації за рахунок коштів підприємства. Це регламентовано статтею 26 ЗУ «Про підприємства» і статтею 13 ЗУ «Про пенсійне забезпечення». По-третє, робочих місць з несприятливими умовами праці, на яких необхідно здійснити першочергові заходи щодо їх поліпшення.

Перелік робочих місць, виробництв, робіт, професій та посад підписує голова комісії, відповідно погоджуючи з профспілковим комітетом. На підприємстві видається наказ, яким затверджуються вищевказані переліки.

Матеріали атестації робочих місць за умовами праці є документами суворої звітності і повинні зберігатися на підприємстві протягом 50 років. Звертаємо увагу, що сьогодні існує проблема, коли навіть діючі підприємства не зберегли матеріали атестацій робочих місць за умовами праці і це створює проблему для працівників яким підтверджено право на пільгову пенсію за Списками № 1 і № 2. Тому, в разі виникнення такої ситуації працівники можуть звертатися до суду з метою захисту свої прав[25].

7.1.2 Організація робочого місця: природне та штучне освітлення

Освітлення – потік світла, який падає на певну горизонтальну площину з джерела світла.

В нормативному документі ДБН В.2.5-28-2006 [27] описано вимоги до природного та штучне освітлення, яким в свою чергу має керуватись керівник і довести до відома своїм працівникам.

Очищення скла світлових прорізів має проводитися не рідше 2 разів на рік у приміщеннях з незначним виділенням пилу і не рідше 4 разів на рік при значному виділенні пилу; для світильників – 4-12 разів на рік (залежно від характеру запиленості виробничого приміщення).

Мити вікна треба 4 рази на рік ззовні, 2 рази на місяць з середини.

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

Основні вимоги до освітлення [28]:

– освітлення повинно відповідати санітарним нормам;

– повинно бути стійким, рівномірним, не повинно бути перепадів світла, різких тіней;

– створювати на робочій поверхні освітленість, що відповідає характеру зорової роботи і не є нижчою за встановлені норми;

– не повинно чинити засліплюючої дії як від самих джерел освітлення, так і від інших предметів, що знаходяться в полі зору;

– забезпечити достатню рівномірність та постійність рівня освітленості у виробничих приміщеннях, щоб уникнути частої переадаптації органів зору;

– не створювати на робочій поверхні різких та глибоких тіней (особливо рухомих);

– повинен бути достатній для розрізнення деталей контраст поверхонь, що освітлюються;

– не створювати небезпечних та шкідливих виробничих факторів (шум, теплові випромінювання, небезпечне ураження струмом, пожежо- та вибухонебезпека світильників);

– повинно бути надійним і простим в експлуатації, економічним та естетичним;

– спектральний склад джерел штучного світла повинен наближатися до спектрального складу джерела природного світла;

– повинно бути таким, щоб людина, яка знаходиться в цьому приміщенні чітко розрізняла найменші об’єкти розгляду, розмір яких 0,5-1,0 мм;

– щоб чітко розрізняти найменші об’єкти розгляду повинен бути чіткий контраст між фоном і об’єктом розгляду.

Природне освітлення залежить від:

– метеорологічних умов;

– пори року;

– пори доби;

– орієнтації вікон відносно сонця;

– висоти і глибини приміщень;

– площі світового пройому;

– висоти вікна;

– чистоти вікна;

– якості віконного скла;

– від поверху;

– від предметів на вікнах.

Освітлення природне:

– бокове одностороннє (надходить у приміщення через світлові прорізи в зовнішніх стінах будинку);

– бокове двостороннє;

– верхнє (проникає крізь ліхтарі та світлові прорізи в покритті);

– комбіноване (бокове і верхнє освітлення).

Штучне освітлення здійснюється за допомогою ламп розжарювання і газорозрядних ламп (люмінісцентні).

За функціональним призначенням штучне освітлення поділяється на:

– робоче призначене для освітлення всього приміщення або окремого робочого місця: - робоче загальне;

– робоче місцеве;

– комбіноване.

Робоче місцеве освітлення окремо використовувати заборонено, воно не нормується і не регулюється.

– евакуаційне;

– аварійне;

– охоронне освітлення передбачається для освітлення небезпечних ділянок в нічний час.

Зір дає людині майже 90% інформації про довкілля. Недостатність освітлення призводить до втоми не тільки органів зору, а й організму людини в цілому, підвищується травмонебезпека. Надто яскраве світло осліпляє. Залежно від джерела світла виробниче освітлення може бути трьох видів: природне, що створюється безпосередньо сонцем; штучне, що здійснюється електричними лампами; змішане, що створюється одночасно природним та штучним освітленням.

Вимоги до освітлення для візуального сприймання користувачами інформації з двох різних носіїв (з екрана ПК та паперового носія) різні. Надто низький рівень освітленості погіршує сприймання інформації при читанні документів, а надто високий призводить до зменшення контрасту зображення знаків на екрані. При 10% зменшенні освітленості працездатність знижується на 1% [28].

Освітлення робочого місця повинно бути змішаним (природним і штучним). Доцільно, щоб орієнтація світлових пройомів для приміщення з ВДТ була на північ.

Слід передбачати наявність сонцезахисних засобів, що знижують перепади яскравостей між природним світлом та свіченням екрана ЕОМ. Необхідно використовувати жалюзі з вертикальними ламелями, що регулюються.

Правила експлуатації освітлення:

– періодичне фарбування (побілка) у світлі тони колон, простінок, стелі для збільшення освітленості на робочих місцях (не рідше одного разу на рік);

– організація при необхідності контролю за світловим потоком на робочі місця;

– обслуговування систем освітлення працівниками неелектричних спеціальностей не допускається;

– ремонт мережі, заміна перегорілих ламп загального і місцевого освітлення тощо – обов'язок електротехнічного персоналу.

Під час експлуатації освітлювальної установки необхідно періодично перевіряти:

– стан ізоляції проводів (не менше 1 разу на 6 місяців);

– рівень освітленості в контрольних точках виробничого приміщення (не менше 1 разу на рік після чергової чистки світильників і заміни згорілих ламп).

7.2 Безпека в надзвичайних ситуаціях

7.2.1 Забезпечення захисту працівників суб'єктів господарювання та населення від впливу іонізуючих випромінювань

Захист від іонізуючих випромінювань може здійснюватись шляхом використання наступних принципів:

– використання джерел з мінімальним випромінюванням шляхом переходу на менш активні джерела, зменшення кількості ізотопа;

– скорочення часу роботи з джерелом іонізуючого випромінювання;

– віддалення робочого місця від джерела іонізуючого випромінювання;

– екранування джерела іонізуючого випромінювання.

Екрани можуть бути пересувні або стаціонарні, призначені для поглинання або послаблення іонізуючого випромінювання. Екранами можуть бути стінки контейнерів для перевезення радіоактивних ізотопів, стінки сейфів для їх зберігання

Альфа-частинки екрануються шаром повітря товщиною декілька сантиметрів, шаром скла товщиною декілька міліметрів. Однак, працюючи з альфа-активними ізотопами, необхідно також захищатись і від бета- або гамма-випромінювання.

З метою захисту від бета-випромінювання використовуються матеріали з малою атомною масою. Для цього використовують комбіновані екрани, у котрих з боку джерела розташовується матеріал з малою атомною масою товщиною, що дорівнює довжині пробігу бета-частинок, а за ним — з великою масою.

З метою захисту від рентгенівського та гамма-випромінювання застосовуються матеріали з великою атомною масою та з високою щільністю (свинець, вольфрам).

Для захисту від нейтронного випромінювання використовують матеріали, котрі містять водень (вода, парафін), а також бор, берилій, кадмій, графіт. Враховуючи те, що нейтронні потоки супроводжуються гамма-випромінюванням, слід використовувати комбінований захист у вигляді шаруватих екранів з важких та легких_матеріалів (свинець-поліетилен).

Дієвим захисним засобом є використання дистанційного керування, маніпуляторів, роботизованих комплексів.

В залежності від характеру виконуваних робіт вибирають засоби індивідуального захисту: халати та шапочки з бавовняної тканини захисні фартухи, гумові рукавиці, щитки, засоби захисту органів дихання (респіратор „Лепесток"), комбінезони, пневмокостюми, гумові чоботи.

Дієвим чинником забезпечення радіаційної безпеки є дозиметричний контроль за рівнями опромінення персоналу та за рівнем радіації в навколишньому середовищі.

Оцінка радіаційного стану здійснюється за допомогою приладів, принцип дії котрих базується на наступних методах:

– іонізаційний (вимірювання ступеня іонізації середовища);

– сцинтиляційний (вимірювання інтенсивності світлових спалахів, котрі виникають в речовинах, що люмінесціюють при проходженні через них іоні: «чих випромінювань);

– фотографічний (вимірювання оптичної щільності почорніння фотопластинки під дією випромінювання);

– калориметричні методи (вимірювання кількості тепла, що виділяється в поглинальній речовині).

8.2.2 Оцінка стійкості роботи промислового підприємства до впливу вторинних вражаючих факторів

При надзвичайних ситуаціях всілякі підприємства, що потрапили в їх зону, часто повністю або частково втрачають здатність виробляти продукцію, виконувати інші свої функції. У цьому випадку говорять про втрату даними виробничим об'єктом стійкості функціонування.

Будь-інженерові-виробничнику вході своєї діяльності деколи доводиться мати справу з виникаючими на підприємстві аваріями, з техногенними впливами ззовні і з впливами на об'єкт природної стихії. Тому для інженера актуальні знання, які можуть бути використані для підтримки і підвищення стійкості функціонування виробництва в цих умовах.

У загальному випадку під стійкістю функціонування промислового об'єкта в надзвичайних ситуаціях розуміється здатність об'єкта випускати встановлені види продукції в заданих обсягах і номенклатурі, передбачених відповідними планами в умовах цих ситуацій, а також пристосованість цього об'єкта до відновлення у разі пошкодження. Для об'єктів, не пов'язаних з виробництвом матеріальних предметів (транспорт, зв'язок, електроенергетика, наука, освіта тощо), стійкість функціонування визначається здатність об'єкта виконувати свої функції і відновлювати їх.

Оскільки об'єкти поряд з персоналом, будівлями, спорудами, паливно-енергетичними ресурсами включають в якості базової складової технологічні (технічні) системи, доцільно визначити і їх стійкість.

Під стійкістю технологічної (технічної) системи розуміється можливість збереження її працездатності при надзвичайній ситуації.

Процес структурної перебудови в галузях промисловості на фоні роздержавлення і приватизації підприємств проходив без належного врахування необхідності забезпечення технічної безпеки та протиаварійної стійкості промислових виробництв. Багато підприємці та керівники підприємств розглядали і розглядають витрати на безпеку і противоаварийную стійкість в якості свого роду резерву для зниження витрат і забезпечення швидкого прибутку.

Аналіз стану безпеки промислових об'єктів показує, що її низький рівень пов'язаний, насамперед, з незадовільним станом основних фондів, повільними темпами реконструкції виробництв, відставанням термінів ремонтів та заміни застарілого обладнання, несправностями або відсутністю надійних систем попередження та локалізації аварій, приладів контролю і засобів захисту.

На працездатність промислового об'єкта можуть чинити негативний вплив умови району його розташування, які визначають рівень і ймовірність впливу небезпечних факторів природного походження: сейсмічного впливу, селів, зсувів, тайфунів, цунамі, зливи тощо Важливі також метеорологічні та інші природні умови.

На стійкість функціонування об'єкта також впливають характер забудови території (структура, тип та щільність забудови), що оточують об'єкт суміжні та інші виробництва, транспортні комунікації.

Стійкість функціонування, крім цього, залежить від деяких особливостей виробництва, пов'язаних із станом персоналу, в тому числі від рівня кваліфікації, підготовки персоналу та фахівців з безпеки, технологічної і виробничої дисципліни, впливу керівників та інженерно-технічних працівників на виконавців робіт.

Рівень стійкості обумовлюють також темпи і результати науково-дослідних і конструкторських розробок та стан їх впровадження, що, в кінцевому рахунку, позначається на вдосконаленні та оновленні техніки і технологій виробництва.

При конкретної надзвичайної ситуації ступінь і характер ураження об'єктів економіки, що ведуть до втрати стійкості функціонування, залежать від параметрів вражаючих чинників джерела надзвичайної ситуації (стихійне лихо, аварія техногенного характеру, застосування противником сучасних засобів ураження), відстань від об'єкта до епіцентру формування вражаючих факторів, технічних характеристик будівель, споруд і обладнання, планування об'єкта, метеорологічних та інших умов, а також від уміння персоналу протистояти лиху.

Вторинними вражаючими факторами є пожежі, вибухи, затоплення, забруднення атмосфери та місцевості і т. ін. Втрати від вторинних вражаючих факторів у ряді випадків можуть значно перебільшувати втрати, які одержує господарство в результаті дії первинних факторів, притаманних більшості надзвичайних ситуацій.

Джерела вторинних вражаючих факторів на об'єкті й в небезпечному віддаленні від нього повинні виявлятися заздалегідь з метою завчасного прийняття заходів, що направлені на виключення чи зменшення вражаючої дії.

Оцінка стійкості об'єктів до дії вторинних вражаючих факторів проводиться в такій послідовності:

– виявляють всі можливі джерела вражаючих факторів, як внутрішні, так і зовнішні;

– визначають найкоротшу відстань від об'єкта до кожного джерела вторинного ураження (на місцевості або на мапі чи плані);

– визначають характер вражаючої дії вторинного фактора (пожежа, затоплення, загазованість т. ін.);

– встановлюють чи вираховують час від моменту появи до моменту початку дії на об'єкт вторинного вражаючого фактора;

– визначають тривалість дії вражаючого фактора й можливі розміри втрат.

Одержані результати аналізують і роблять конкретні висновки для розробки організаційних, інженерно-технічних та технологічних заходів щодо виключення або обмеження дії на роботу об'єкта вторинних вражаючих факторів.

8 ЕКОЛОГІЯ

8.1 Етапи та техніка збору та опрацювання екологічної інформації

Екологічні дослідження вимагають систематичного дотримання чотирьох послідовних етапів: спостереження; формулювання на основі спостережень теорії про закономірність досліджуваного явища; перевірка теорії наступними спостереженнями і експериментами; спостереження за тим, чи є правдивими передбачення, основані на цій теорії.

Факти базуються на прямих або непрямих спостереженнях, що виконані за допомогою органів відчуття або приладів. Всі факти, які належать до конкретної проблеми, називають даними. Спостереження можуть бути якісними (тобто описувати колір, форму, смак, зовнішній вигляд тощо) або кількісними. Кількісні спостереження є точнішими. Вони включають вимірювання величини або кількості, наочним виразом яких можуть бути якісні ознаки. Внаслідок спостережень отримують так званий “сирий матеріал”, на основі якого формулюється гіпотеза.

Для оцінки гіпотези проводять серію експериментів з метою отримання нових результатів, які б підтверджували або ж заперечували гіпотезу. В більшості гіпотез обговорюється ряд факторів, які могли б вплинути на результати спостережень.

Методологічною основою екологічної статистики як науки про екологічний стан оточуючого середовища є системний підхід.

Техніка збору інформації. В екології найбільше поширені польові біометричні методи і експерименти: перші дають змогу одержати інформацію методом безпосередніх спостережень, другі - забезпечують інформацією в процесі лабораторних досліджень. Збирається інформація за допомогою різних методів.

Метод безпосередніх спостережень екосистеми або її окремих компонентів в природних умовах передбачає невтручання (або ж мінімально

можливе втручання) спостерігача в природні процеси, стосунки чи стани. Цей метод ще називають порівняльним еколого- географічним, або ж методом порівняльної екології.

Методи збору інформації. Існує багато методів збору інформації: польовий метод, метод безпосередніх спостережень, ландшафтно- екологічний підхід, ландшафтно-індикаційні, гідрохімічні, біохімічні, грунтовогазові, гідрогеологічні, радіоекологічні спостереження, геохімічні спостереження ландшафтів, дистанційні спостереження, експериментальні дослідження.

Польовий метод – один із основних методів, який проводиться в природних умовах. Його широко використовують в агрохімії, фізіології рослин, землеробстві, рослинництві, лісівництві, селекції. При цьому здійснюють фенологічні спостереження, агрофізичні, агрохімічні, мікробіологічні дослідження ґрунтів, ботанічні, фізіологічні та біохімічні дослідження рослин. Все це дає змогу виявити біоекологічні можливості виду чи сорту рослин, з'ясувати природу відмінності у врожаї та його якості тощо.

Ландшафтно-екологічний підхід дає змогу виділити екосистеми ландшафтну, місцевості, урочища і, нарешті, фацій або асоціацій. Межі цих утворень і є межами біогеоценозу або екосистеми нижчого базового рівня. Вони легко картуються, описуються, досліджуються. Такий підхід дає змогу виділяти як природні, так і штучні біогеоценози, досліджувати їх генезис, прогнозувати сукцесії, здійснювати екологічний моніторинг.

Ландшафтно-індикаційні спостереження виконуються з метою виявлення характерних зовнішніх (наочних) особливостей місцевості, що дає можливість більш цілеспрямовано проводити екологічні роботи, раціонально розташовувати мережу місць спостережень з урахуванням направленості змін рівня забруднення навколишнього середовища.

Гідрохімічні спостереження проводять з метою вивчення підземних вод, здійснюються пробо відбором з природних джерел, криниць і гідрогеологічних свердловин. В кожному конкретному випадку вони повинні обгрунтовуватись,

виходячи з існуючої можливості відбору, природної захищеності водоносних горизонтів і рівня техногенних порушень дослідницької території.

Біохімічні спостереження проводяться з метою вивчення речовинного складу рослинності, насамперед її мікро компонентного складу. Однак при вивченні впливу на навколишнє середовище будь- якого специфічного забруднення, доцільно вивчення біоти саме за цим показником.

Грунтово-газові спостереження використовуються для вивчення активних зон тектонічних порушень; для вивчення техногенних забруднень вуглеводами підземних вод чи порід у випадку, якщо забруднення не проявляється на поверхні; вивчення летючих забруднювачів.

Гідрогеологічні спостереження спрямовані на вивчення гідрохімічних, гідродинамічних і гідрофізичних особливостей стану підземних вод за допомогою природних джерел, криниць і гідрогеологічних свердловин. При цьому встановлюються зміни гідрохімічних і гідродинамічних параметрів підземних вод в просторі і часі. Схема розташування гідрогеологічних пунктів спостережень, обсяги і режими досліджень визначаються конкретною природнотехногенною обстановкою.

Геохімічні спостереження ландшафтів включають в себе роботи з вивчення геохімічних характеристик різних компонентів природного середовища, що дозволяє виконувати балансові розрахунки і, таким чином, оцінювати кількісні характеристики міграції забруднюючих речовин. В найбільш повному виді геохімічні дослідження ландшафтів включають в себе комплекс робіт з вивчення: геохімії грунтів і порід зони аерації, гідро-геохімії підземних вод, геохімії донних осаджень водотоків і водойм, біогеохімії представницьких рослинних спільнот, гідрохімії атмосферних опадів і поверхневих вод.

Дистанційні спостереження дозволяють одержувати інформацію про стан окремих компонентів природного середовища і його перетворення під впливом техногенезу, активності прояву екзогенних геологічних процесів тощо. За допомогою одержаних дистанційним зондуванням спектральних

характеристик рослинного покрову, грунтів і водоймищ можна вирішувати наступні задачі:

– оцінки біомаси і вологовмісту рослин, впливу на них метеоумов, агрохімікатів і важких металів;

– ідентифікації мінерального складу грунтів і гірських порід, в тому числі мінеральних включень агрохімікатів;

– оцінки вмісту завислих речовин і нафтопродуктів у водоймах.

Експериментальні дослідження використовують методи прямого втручання в будову і життя ценоекосистеми або культурекосистеми, їх фрагментів, синузій, популяцій. Деякі з цих об’єктів досліджуються і в умовах лабораторій методом моделей.

Різниця між польовим і лабораторним експериментом полягає в тому, що перший є практично неконтрольованим через безмежну кількість природних факторів, які діють на об’єкт, другий є життєво контрольований.. Екологічний експеримент, одночасно як і спостереження над екосистемами, є ефективним лише в поєднанні з третім, дуже важливим, методом екології - методом моделювання.

8.2 Індексний метод в екології

Індекс англійський термін “*index number”* означає число-показник. Статистичні індекси — це відносні величини, які одержують внаслідок порівняння складних екологічних явищ, утворених з різнорідних елементів, що не підлягають безпосередньому підсумовуванню.

Індекс у статистиці – узагальнюючий відносний показник, який характеризує співвідношення в часі чи просторі соціально-екологічних явищ і процесів.

За своєю суттю статистичний індекс характеризує зміну рівня будь-якого суспільного явища в часі, просторі чи порівняно з планом, нормою, стандартом. У цих випадках зіставляються між собою числові значення однойменних

показників, що мають однаковий екологічний зміст. Отже, індексом можна назвати відносну величину динаміки, виконання плану, порівняння.

За допомогою індексів можна характеризувати зміну в часі і просторі найрізноманітніших показників: обсяги викидів в атмосферу, скидів шкідливих речовин у водне середовище, інтенсивність забруднень і т. д. Їх поділяють на дві групи: до першої належать об'ємні (сумарні) показники (наприклад, обсяг викидів та скидів кількість забруднювачів, площа забрудненої території та ін.), які виражаються абсолютними величинами; до другої – показники, розраховані на певну одиницю (наприклад, викиди в розрахунку на одиницю земельної площі або на одного жителя, працівника і т.д.). Останні умовно можна назвати якісними показниками, і виражаються вони у вигляді середніх величин. Ця особливість зумовлює поділ індексів на індекси кількісних та індекси якісних показників.

За допомогою статистичних індексів можна відображувати зміну в часі і просторі як окремих простих показників (наприклад, обсяг викидів вуглецю, окислів азоту, сірки і т.д.), так і однойменних показників за складними сукупностями (наприклад, зміна обсягу викидів по місту, району, області в цілому і т.д.).

За допомогою індексного методу вирішуються такі завдання: характеризують загальну зміну складного економічного явища чи окремих його елементів (складових), виділяють вплив одного з факторів через елімінування впливу інших, відокремлюють впливу зміни структури явища на зміну індексованої величини.

При цьому сама міра впливу може бути визначена як у відносних вимірниках, так і в абсолютних.

Класифікація індексів. Класифікують індекси за різними ознаками:

– за змістом досліджуваних об'єктів, явищ і процесів - індекси обсягу, індекси якісних показників;

– за повнотою охоплення елементів сукупності - індивідуальні індекси, зведені (групові, загальні) індекси;

– за формою зображення - агрегатні індекси, середні зважені індекси (арифметичні, гармонійні);

– за базою порівняння - індекси динаміки (базові, ланцюгові), індекси виконання плану, територіальні індекси;

– за характером впливу на зміну складного явища - індекси сталого складу, індекси структурних зрушень;

– за коефіцієнтами співвимірювання - індекси зі змінними вагами, індекси зі сталими вагами.

Обчислення загальних індексів, що дають змогу співвіднести між собою показники за складними сукупностями, являє собою особливий прийом дослідження, який називається індексним методом. За його допомогою можна не тільки вивчати динаміку показників, а й вимірювати вплив окремих факторів на динаміку складного показника. При цьому залежно від завдань аналізу можна фактори вивчати ізольовано, абстрагуючись від дії інших, або розглядати їх взаємопов'язано.

Методологічні принципи побудови індексів. Індексний метод має свою термінологію та символіку. Її дотримання є обов'язковою умовою в індексному аналізі.

Для побудови статистичного індексу необхідно мати вихідну інформацію, як мінімум, за два періоди. Один з таких періодів називається базисним, другий – поточним. Базисний – це період, з яким порівнюють досліджувані явища, поточний – період, що порівнюється. Так. в індексах динаміки базисним є показник попереднього періоду (моменту) часу, в індексах порівняння з нормативною базою – нормативний рівень, а в індексах порівняння (в просторі) базисним може бути показник, що належить до якоїсь з територій. Якщо досліджуються дані за кілька періодів, то один з них (як правило, початковий) буде базисним, а решта — поточними, або звітними.

У теорії індексів показник, зміну якого характеризує індекс, називають індексованою величиною, а пов'язану з нею величину, що використовують як постійну, – елімінованою величиною, або вагою. Остання відіграє роль

сумірннка. Використання цих двох видів величин вважається особливістю індексного методу аналізу. При побудові статистичних індексів насамперед необхідно вирішити такі питання: який набір різнорідних елементів досліджуватиметься, які показники виступатимуть індексованими величинами, які величини виступатимуть сумірниками (вагами).

При цьому встановлюють, які досліджувані показники при побудові індексів вважаються базисними, а які — поточними.

8.3 Висновки до сьомого розділу

В даному розділі розглянуто наступні питання: етапи та техніка збору та опрацювання екологічної інформації та індексний метод в екології.

ВИСНОВКИ

Веб-додатки можуть мати декілька вразливих моментів, які важко виявити вручну. Проведемо огляд найпоширеніших проблем безпеки, як перевірити стан своєї програми та отримати кращі поради щодо безпеки сайту.

Кожен веб-додаток може мати вразливі місця, будь то пов’язаний з основною системою ядра або одним веб-сайтом на основі популярного рішення CMS.

Багато додатків розробляються протягом тривалого періоду часу за участі різних середовищ та людей. Якщо ви самі програмували, то знаєте, що метою є, як правило, виправити помилку або змусити щось працювати. Коли все вирішено після багатогодинної роботи, просто запустити додаток таким, яким він є, і не замислюватися над питаннями безпеки.

«Найбільша помилка безпеки в усіх сайтах – пропуск роботи з виявлення вразливості. Інша проблема – це усвідомлення вразливих місць, щоб нічого не робити. Це частіше, ніж ви можете собі уявити. Зазвичай основні вразливості безпеки виправляються, тоді як залишаються менші. Кілька питань із низьким рівнем ризику швидко додаються до головного питання безпеки», – каже Лінус Саруд, дослідник безпеки компанії Detectify.

На даний момент в компаніях майже не замислюються про безпеку інформаційних сторінок в мережі Інтернет, і практично не виділяють цьому питанню уваги, зокрема часто починають частіше вживати заходів тоді коли відбувся витік важливої інформації.

Щоб усунути цю проблему, яка пов’язана з захистом інформації і захистити веб-сайт від зловмисників, потрібно розглянути захист бази даних всередині мережі, а також провести моніторинг мережі. Моніторинг мережі і діагностику можна провести за допомогою сканерів мережі, або за допомогою спеціалізованого обладнання.Поряд з тим було розглянуто ряд додаткових розділів.

В розділі “Спеціальна частина” проведено аналіз альтернативних сканерів вразливостей веб-сайтів.

В розділі “Обґрунтування економічної ефективності” проведено економічні розрахунки, спрямовані на визначення економічної ефективності від дослідження систем захисту веб-сайтів, а також прийнято рішення щодо подальшого розвитку розробки. Розраховане значення економічної ефективності становить 0,559, що є високим значенням. Так само нормальним є термін окупності. Для даного дослідження він становить 1.78 року.

В розділі “Охорона праці та безпека в надзвичайних ситуаціях” розглянуто такі питання: дії роботодавця за результатами атестації робочих місць за умовами праці, організація робочого місця: природне та штучне освітлення, забезпечення захисту працівників суб’єктів господарювання та населення від впливу іонізуючих випромінювань та оцінка стійкості роботи промислового підприємства до впливу вторинних вражаючих факторів.

В розділі “Екологія” наведено етапи та техніка збору та опрацювання екологічної інформації та індексний метод в екології.

ПЕРЕЛІК ПОСИЛАНЬ

1. Black, E. IBM and the Holocaust: The Strategic Alliance between Nazi Germany and America's Most Powerful Corporation. Crown Pub, 2001.

2. Common Criteria for Information Technology Security Evaluation-Part 1: Introduction and general model. ISO/IEC SC27 N2161, 1998.

3. Information Processing Systems — Open Systems Interconnection (ISO), Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules

4. ISO/IEC 17799:2000. Информационные технологии. Свод правил по управлению защитой информации. Международный стандарт [Текст] / ISO/IEC, 2000.

5. Алферов А. П. Основы криптографии [Текст]: учеб. пособие / А. П. Алфе-ров, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. – М.: Гелиос АРВ, 2001. – 480 с. : ил.

6. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий [Текст] : РД: утв. Гостехкомиссией России. – М., 2002.

7. Гайдамакин Н. А. Автоматизированные системы, базы и банки данных. Вводный курс [Текст]: учеб. пособие / Н. А. Гайдамакин. – М.: Гелиос АРВ, 2002. – 368 с. : ил.

8. Методичні вказівки до виконання дипломної роботи ОКР “Магістр” для студентів спеціальності 8.05010101– Інформаційні управляючі системи та технології / Укладачі: О. В. Маєвський, О.В. Мацюк, М.В. Приймак, Г.В. Шимчук – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2014. – 196 с.

9. Гайдамакин Н. А. Разграничение доступа к информации в компьютерных системах [Текст] / Н. А. Гайдамакин. – Екатеринбург: Изд-во Урал. Ун-та, 2003. – 328 с. : ил.

10. ГОСТ Р 15408–02. Критерии оценки безопасности информационных технологий [Текст]. – Введ. 2004–01–01 – М.: Изд-во стандартов, 2002.

11. ГОСТ Р 51275–99. Защита информации. Объект информатизации. Факторы, воздействующие на информацию [Текст]. – Введ. 2000–01–01 – М.: Изд-во стандартов, 1999. – 8 с.

12. Девянин П. Н. Теоретические основы компьютерной безопасности [Текст]: учеб. пособие для вузов / П. Н. Девянин, О. О. Михальский, Д. И. Правиков, А. Ю. Щербаков. – М.: Радио и связь, 2000. – 192 с. : ил. ; 21 см.

13. Защита от несанкционированного доступа к информации. Термины и определения [Текст] : РД : утв. Гостехкомиссией России. – М.: Изд-во стандартов, 1992.

14. Зегжда Д. П. Как построить защищенную информационную систему. Технология создания безопасных систем [Текст] / Д. П. Зегжда, А. М. Ивашко ; под научн. ред. П. Д. Зегжды, В. В. Платонова. – СПб.: Мир и Семья-95, Интерлайн, 1998. – 256 с. : ил. ; 20 см. – 500 экз.

15. Молдовян А.А. Криптография [Текст] / А. А. Молдовян, Н. А. Молдовян, Б. Я. Советов. СПб.: Лань, 2000. – 224 с. : ил.

16. Петров А. А. Компьютерная безопасность. Криптографические методы защиты [Текст] / А. А. Петров – М.: ДМК, 2000. – 448 с. : ил.

17. Проскурин В. Г. Программно-аппаратные средства обеспечения информационной безопасности. Защита в операционных системах [Текст]: учеб. пособие для вузов / В. Г. Проскурин, С. В. Крутов, И. В. Мацкевич. – М.: Радио и связь, 2000. – 168 с. : ил.

18. Разработка политики безопасности организации в свете новейшей нормативной базы / А. С. Марков, С. В. Миронов, В. Л. Цирлов // Защита информации. Конфидент. – 2004. – № 2 – С. 20–28.

19. Синадский Н. И. Угрозы безопасности компьютерной информации [Текст]: учеб. пособие / Н. И. Синадский, О. Н. Соболев – Екатеринбург: Изд-во Урал. ун-та, 2000. – 85 с. : ил.

20. Средства вычислительной техники. Защита от НСД к информации. Показатели защищенности от НСД к информации [Текст] : РД : утв. Гостехкомиссией России. – М.: Изд-во стандартов, 1992.

21. Хорев П. Б. Методы и средства защиты информации в компьютерных системах [Текст]: учеб. пособие для вузов / П. Б. Хорев. – М.: Академия, 2005. – 256 с. : ил.

22. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа [Текст] / А. Ю. Щеглов ; под ред. М. В. Финкова. – СПб: Наука и Техника, 2004. – 384 с. : ил.

23. Жидецький В. Ц. Основи охорони праці: Підручник. / В.Ц. Жидецький - 4-те вид., перероб. і доп. - К.: Знання, 2010. - 375 с. + компакт-диск. – ISBN 978-966-346-601-9.

24. Запорожець О. І. Основи охорони праці. Підручник / О. І. Запорожець, О. С. Протоєрейський, Г. М. Франчук, І. М. Боровик – К.: Центр учбової літератури, 2009. – 264 с. – ISBN 978-966-364-934-4.

25. Цапко В.Г. Безпека життєдіяльності: Навч. посіб. / За ред. В.Г. Цапка. - 3-тє вид., стер. - К.: Знання, 2004. - 397 с. – ISBN 966-8148-39-8.

26. Як проводиться атестація робочих місць за умовами праці [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: https://hrliga.com/index.php?module=news&op=view&id=20009

27. ДБН В.2.5-28-2006 [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: http://kbu.org.ua/assets/app/documents/dbn2/95.1.%20ДБН%20В.2.5-28-2006.% 20 Природне%20і%20штучне%20освітлення.pdf

28. Установлюємо доплату за роботу з важкими та шкідливими умовами праці [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: https://interbuh.com.ua/ua/documents/ib/8902/123191

29. Вимоги до освітлення робочого місця [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: https://pidruchniki.com/10560412/bzhd/osvitlennya_primischen_robochih_mists