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

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

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Дослідження механізмів захисту в розподілених системах та аналіз проксі-з'єднань

Виконав: студент IV курсу, групи СНзс-42
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис) Гриб І.О. (прізвище та ініціали)

Керівник (підпис) Щербак Л.М. (прізвище та ініціали)

Нормоконтроль (підпис) Шимчук Г.В. (прізвище та ініціали)

Завідувач кафедри (підпис) Боднарчук І.О. (прізвище та ініціали)

Рецензент (підпис) Стадник М.А. (прізвище та ініціали)

Тернопіль
2021

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

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

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«__» _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Гриб Ірині Олегівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження механізмів захисту в розподілених системах та аналіз проксі-з'єднань

Керівник роботи Щербак Леонід Миколайович, д.т.н., професор кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «02» 03 2021 року № 4/7-170

2. Термін подання студентом завершеної роботи 14.06 2021р.

3. Вихідні дані до роботи Наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ, Розділ 1. Аналіз предметної області, 1.1 Архітектури систем, 1.2 Архітектури розподіленої системи, 1.3 Клієнт-серверні системи, 1.4 Приховування розподілу в розподілених системах, 1.5 Інтеграція файлової системи, 1.6 Механізми зберігання інформації в розподілених системах, 1.7 Розподілені операційні системи, 1.8 Безпека в розподілених системах, 1.9 Розподілені бази даних та сховища даних, 1.10 Зв'язок у розподілених системах, 1.11 Протоколи трансляції для розподілених систем, 1.12 Розподілені файлові системи (DFS), 1.13 NFS (мережева файлова система), 1.14 Файлова система Ендрю (AFS), 1.15 Висновок до першого розділу, Розділ 2. Використання Proxy-з'єднань, 2.1 Налаштування NGINX для прийняття протоколу Proxy, 2.2 Авторизація проксі, 2.3 Зміна IP-адреси балансувача навантаження на IP-адресу клієнта, 2.4 HTTP Proxy та SOCKS-Proxy сервера, 2.5 Анонімні Proxy, 2.6 Кібератаки на проксі-сервери, 2.7 Тестування на проникнення: TOR, VPN або проксі, 2.8 Висновок до другого розділу, Розділ 3. Безпека життєдіяльності, основи хорони праці, Висновки, Перелік літературних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Тема, мета, завдання. 2. Передача типового повідомлення. 3. Модель мережевої взаємодії. 4. Логічна багаторівнева організація. 5. Зв'язок proxy та аутентифікації. 6. Протокол авторизації. 7. VPN. 8. Робота SSH-тунелю. 9. Налаштування Proxy зєднання. 10. Створення скану. 11. Сканування портів. 12. Результати сканування. 12. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання 17травня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	17.05.2021	*Виконано*
2.	Аналіз літературних джерел	18.05.2021-19.05.2021	*Виконано*
3.	Виконання дослідження щодо механізмів захисту в розподілених системах	20.05.2021-21.05.2021	*Виконано*
4.	Оформлення розділу «Аналіз предметної області»	22.05.2021-28.05.2021	*Виконано*
5.	Оформлення розділу «Використання Proxy-з'єднань»	29.06.2021-01.06.2021	*Виконано*
6.	Виконання завдання до підрозділу «Безпека життєдіяльності»	2.06.2021	*Виконано*
7.	Виконання завдання до підрозділу «Основи хорони	2.06.2021	*Виконано*
	праці»		
8.	Оформлення кваліфікаційної роботи	6.06.2021	*Виконано*
9.	Нормоконтроль	7.06.2021	*Виконано*
10.	Перевірка на плагіат	7.06.2021	*Виконано*
11.	Попередній захист кваліфікаційної роботи	11.06.2021	*Виконано*
12.	Захист кваліфікаційної роботи	14.06.2021	

Студент _____ Гриб І.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Щербак Л.М..

АНОТАЦІЯ

Дослідження механізмів захисту в розподілених системах та аналіз проксі-з’єднань // Кваліфікаційна робота освітнього рівня «Бакалавр» // Гриб Ірина Олегівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СНзс-42 // Тернопіль, 2021 // С. –65 рисунків – 7 , таблиці – 4 кресл. – , додат. – 0 , бібліогр. – 11 .

Ключові слова: DNS, OSI, DES, KDC, HTTP, SOCKS, NAT, SSH

Провести дослідження механізмів захисту в розподілених системах та аналіз проксі-з’єднань і вирішити такі завдання:

1. Провести дослідження особливостей розподіленої системи.
2. Огляд захисту інформації в розподіленій системі.
3. Провести дослідження особливостей Proxy з'єднань.
4. Провести огляд захисту інформації Proxy з'єднань.

ANNOTATION

Study of security mechanisms in distributed systems and proxy interconnect analysis // Qualification work of educational level «Bachelor» // Hryb Iryna Olehivna // Ternopil’ Ivan Pul’uj National Technical University, Faculty of Computer Information System and Software Engineering, Department of Computer Science, group SNzs-42 // Ternopil', 2021 // P. 65, Fig. – 7, Tables – 4, References –11 Annexes. – 0 .

Keywords: DNS, OSI, DES, KDC, HTTP, SOCKS, NAT, SSH

Carry out research of protection mechanisms in distributed systems and analysis of proxy connections and solve the following tasks:

1. To study the features of a distributed system.
2. Review of information security in a distributed system.
3. Conduct research on the features of Proxy compounds.
4. Review the protection of Proxy connection information.

ЗМІСТ

ВСТУП

Актуальність теми. В інформатиці «розподілені системи» є важливою предметною областю. Розподілена комп'ютерна система складається з безлічі взаємопов'язаних комп'ютерів через загальний канал зв'язку. Цей канал зв'язку полегшує обмін інформацією підключеним комп’ютерам. Отже, це підтримує взаємозв’язок розподіленої системи. Комп’ютерні системи розподілу можуть бути розповсюджені у великій географічній зоні, іноді один комп’ютер знаходиться на одному континенті, а інший – на іншому континенті.

Мета і задачі дослідження. Провести дослідження механізмів захисту в розподілених системах та аналіз проксі-з’єднань і вирішити такі завдання:

1. Провести дослідження особливостей розподіленої системи.
2. Огляд захисту інформації в розподіленій системі.
3. Провести дослідження особливостей Proxy з'єднань.
4. Провести огляд захисту інформації Proxy з'єднань.

Практичне значення одержаних результатів. Основним стимулом впровадження та розвитку розподілених систем є те, що висока доступність (мінімальний час простою) та надійність, обчислювальна потужність та апаратні ресурси (ЦП, пам’ять тощо) перевершують порівняно з автономними комп’ютерами. Такі властивості, як сталість, безпека, відмовостійкість, в основному враховуються при проектуванні розподілених систем. Оскільки розподілена система може розширитися до мільйонів вузлів з декількох вузлів, на прийнятність та масштабованість слід також зосередити увагу. На продуктивність розподілених систем впливають зміни, які пов'язані з носієм, який використовується для взаємозв'язку вузла, тобто мережі. Такі поняття, як балансування навантаження та надмірність, тісно пов'язані з розподіленими системами залежно від цілей системи.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Архітектури систем

Існує дві основні категорії архітектур комп'ютерних систем, тобто централізована архітектура та архітектура розподіленої системи. Централізовані системи мають такі важливі переваги, як простота обслуговування, висока надійність обчислень і забезпечують надійний інтерфейс до системи.

Однак у централізованих систем є основні недоліки. Централізовані системи дуже чутливі до відмов. Коли один компонент виходить з ладу в централізованій системі, вся система зазнає збою. Коли відбувається оновлення або модифікація, вся система повинна бути вимкнена. Централізованим системам важко впоратися з великим навантаженням, оскільки вони мають дуже обмежену пам’ять і обчислювальну потужність. Порівняно вартість централізованих систем дуже висока. Щоб впоратися з усіма цими мінусами в централізованих системах, розподілена системна архітектура була представлена кілька десятиліть тому.

1.2 Архітектури розподіленої системи

Розподілену систему можна розглядати як набір комп'ютерів або робочих станцій та серверів, з'єднаних за допомогою спільного каналу зв'язку [1]. Цей носій зв'язку та вузли, підключені до носія зв'язку, колективно створюють локальну мережу (LAN). Користувачі розподілених систем можуть одночасно споживати ресурси власних робочих станцій та ресурси, що надаються віддаленими робочими станціями, приєднаними до розподілених систем.

У розподілених системах є ключові переваги. Розподілені системи забезпечують неоднорідне обчислювальне середовище. Вузли, приєднані до

розподіленої системи, можуть забезпечити безліч функціональних можливостей, а також вони мають різні функції. Розподілені системи забезпечують високу доступність, коли виходить з ладу один вузол, який можна від'єднати від системи, не впливаючи на інший компонент системи, а також новий компонент може бути введений в систему без повної зупинки всієї системи. Ці системи сприяють паралельному виконанню процесів.

Розподілені системи мають свої недоліки, такі як відсутність спільної пам'яті [1] для обміну інформацією та труднощі в управлінні системою через фізичний розподіл робочих станцій у системі важливо керувати безпекою та авторизацією користувачів [1].

Розподілені системи можна розділити на три основні категорії [1] системи клієнт-сервер, інтегровані системи управління файлами та інтегровані розподілені системи [1]. Однак сучасні розподілені системи – це набори функцій, що належать до всіх трьох категорій архітектур.

1.3 Клієнт-серверні системи

Концепція клієнт-серверних систем була ініційована дослідниками Xerox-parc [2]. Сервер забезпечує чітко визначений набір послуг для клієнта через Інтернет. Клієнт і сервер спілкуються через HTTP.

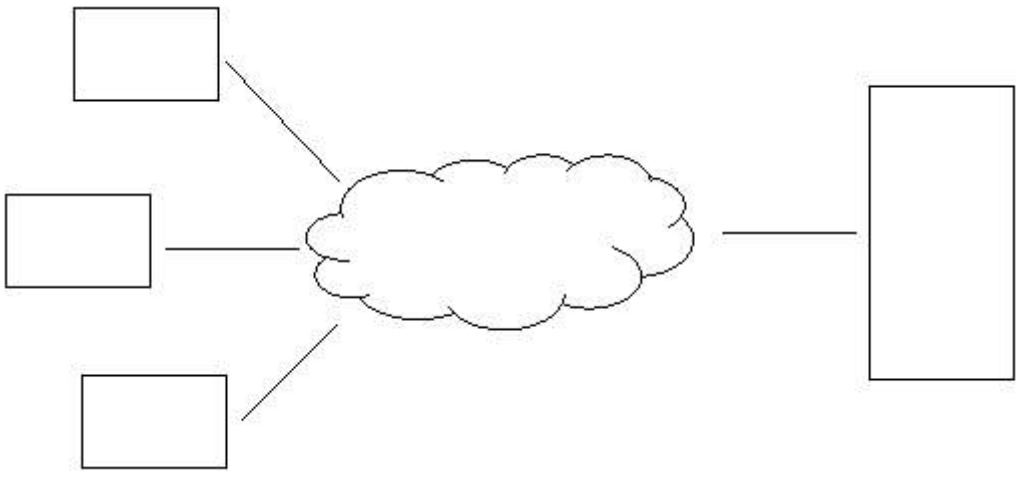

Рисунок 1.1 – Архітектура клієнтського сервера

Сервер надає інтерфейс, який визначає послуги. Клієнти отримують доступ до послуг відповідно до інтерфейсу. Більшість сучасних ядер операційної системи використовують цю модель [1]. Цю архітектуру використовують не тільки веб-системи, але й сучасні ядра операційних систем.

Архітектури клієнтського сервера недостатньо для прозорого доступу до віддалених файлів, оскільки клієнт повинен знати точний сервер, на якому розміщений файл.

Для подолання цих проблем була введена абстракція єдиної файлової системи.

У розподіленій інтегрованій системі функції управління розподілом безпосередньо пов'язані архітектурою системи. Функціональні можливості управління розподілом, такі як управління процесами або віртуальною пам'яттю, іменування об’єктів забезпечуються розподіленим ядром розподіленої системи [1]. На кожному сайті розподіленої системи розташована копія ядра. Ці розподілені ядра спільно працюють над забезпеченням функцій розподіленої системи.

1.4 Приховування розподілу в розподілених системах

Відповідно до визначення розподіленої системи Таненбаумом та Ван Ренссесом [11]. Прозорість – ключове поняття. Незважаючи на те, що розподілені системи складаються з декількох процесорів, користувач системи повинен розглядати його як "віртуальний уніпроцесор". Він не повинен бачити, як у системі діють кілька процесорів. Прозорість розподілених систем досягається різними способами. Автори статті [1] «Приховування розподілу в розподілених системах» описали чотири основних підходи до впровадження прозорості в розподілених системах. Такими підходами є комунікація, інтеграція файлової системи, зберігання інформації та відмовостійкість.

1.4.1 Механізми спілкування

В архітектурі розподіленої системи важливим є зв’язок між взаємопов’язаними вузлами. Механізми зв'язку в сучасних розподілених системах можна розділити на три великі категорії: передача повідомлень, механізм порту та віддалені виклики процедур.

Передача повідомлення

Для синхронізації та зв'язку між процесом усередині розподіленої системи може використовуватися механізм передачі повідомлень. У механізмі передачі повідомлень фізична копія повідомлення передається в процес з іншого процесу. У розподіленому системному середовищі, коли використовується явний механізм передачі повідомлень, програміст повинен знати точну назву та місце розташування процесу, який йому потрібно надіслати повідомлення. Це ускладнює здійснення міграції процесів, оскільки розташування процесу не є прозорим. Як вирішення цієї проблеми може бути використана модель зв'язку порту.

Портовий механізм

Механізм порту може бути використаний для відокремлення управління процесами від комунікацій [4]. Порти діють як поштові скриньки, за допомогою цього механізму повідомлення спрямовуються на ці поштові скриньки. Цей механізм відокремлює ідентифікацію порту від місця, де працює процес. А також це полегшує міграційні стратегії.

Віддалені виклики процедур

Віддалений виклик процедури або RPC є модифікацією механізму виклику процедури. Різниця в RPC полягає лише в тому, що викликана особа, яка телефонує, розташована в різних фізичних місцях. Викликаний – це сервер, а абонент – клієнт. Зазвичай процеси виклику та абонента виконуються у двох різних місцях.

У механізмі RPC параметри запиту та результати передаються за допомогою методів передачі повідомлень. Це синхронізує абонента, щоб чекати результатів абонента [3].

1.5 Інтеграція файлової системи

Для підтримки прозорості розподіленої системи прозорість доступу до файлів є критично важливою. Прозорість розташування файлів, прозорість імен файлів та прозорість міграції файлів важко реалізувати.

Прозорість імен

Ідеальний механізм прозорості імен реалізований у системах підключення Newscale (NC), для реалізації цього дерева імен UNIX поєднуються разом для побудови єдиної структури імен [1].

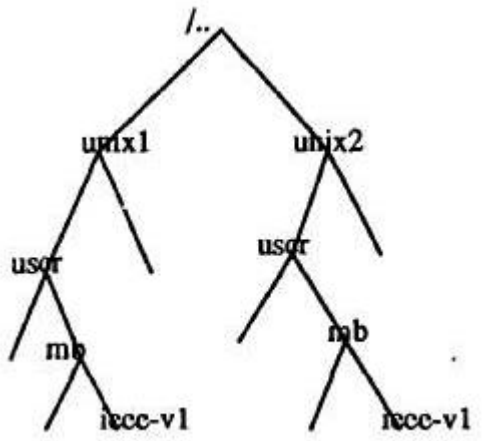

Рисунок 1.2 – Підключення двох файлових систем UNIX [1]

У системі NC вводиться новий додатковий рівень програмного забезпечення між користувацькою програмою та ядром UNIX. Основна відповідальність цього рівня NC полягає у розрізненні системних викликів, які потрібно перенаправити на іншу систему, та прийнятті системних викликів, що надходять з іншої системи.

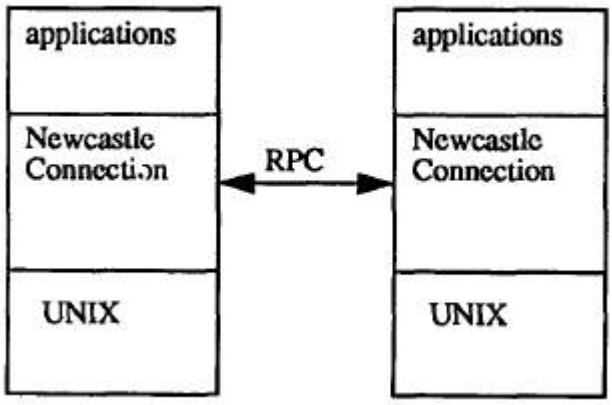

Рисунок 1.3 – Впровадження рівня ЧПУ [1]

Прозорість розташування

Для прозорості розташування внутрішнє ім'я об'єкта повинно бути незалежним від фізичного розташування об'єктів. У системі Apollo [12] як внутрішнє ім'я об'єкта використовується унікальний ідентифікатор (UID). Цей ідентифікатор унікальний для всієї системи. UID складається з двох полів, позначки часу створення та ідентифікатора вузла. UID об'єкта використовується для доступу до відповідних об'єктів [1]. Менеджер підказок зберігає підказки про розташування об’єкта. Спочатку алгоритм визначення об'єкта намагається знайти об'єкт в локальній системі, і якщо об'єкт не вдається знайти в локальній системі, алгоритм отримує допомогу підказки для пошуку об'єкта.

1.6 Механізми зберігання інформації в розподілених системах

У розподіленій системі файли можна отримати за допомогою двох механізмів. Багаторівнева пам’ять і однорівнева пам’ять.

У багаторівневій моделі адресації використовуються два різні режими адресації. Один – адресація даних, що зберігаються в адресному просторі користувача, а другий – доступ до постійних об’єктів або файлів, що зберігаються на диску.

В однорівневій моделі адрес використовується одна адреса. Всі посилання на пам'ять є логічними посиланнями на об'єкти.

Багаторівневе сховище

Для адресування постійних об'єктів використовуються такі файли, які зберігаються в багаторівневих буферах пам'яті. Отже, коли відбувається операція введення / виведення, інформація буде передаватися з диска в буфер і буфер на диск.

Однорівневе сховище

Для доступу до інформації як в основній пам'яті, так і на дисках використовується віртуальна адресація [3]. Режим віртуальної адресації дозволяє користувачам отримати прозорий доступ до інформації. Однорівневий механізм зберігання може бути використаний для забезпечення абстракції спільної пам'яті.

1.7 Розподілені операційні системи

Автори статті [12] представляють розподілену операційну систему під назвою 2K. Ця операційна система вирішує основні проблеми розподілених систем, такі як управління ресурсами, динамічна адаптованість та конфігурація розподілених програм на основі компонентів [12]. У наступному розділі ми опишемо декілька важливих особливостей моделі системи 2K.

Динамічні залежності

2K використовує концепцію об'єкта CORBA для інкапсуляції розподіленого обладнання в системі, а служб CORBA – для інкапсуляції розподілених послуг. Система 2K має сховище компонентів. Коли запитується один компонент, він отримує цей компонент із сховища компонентів, а потім код компонента буде динамічно завантажуватися до часу виконання 2K. Система 2K будує представлення залежностей компонентів під час виконання, використовуючи необхідні специфікації. Для побудови цих подань використовуються об'єкти COBA, названі як "ComponentConfigurators" [12].

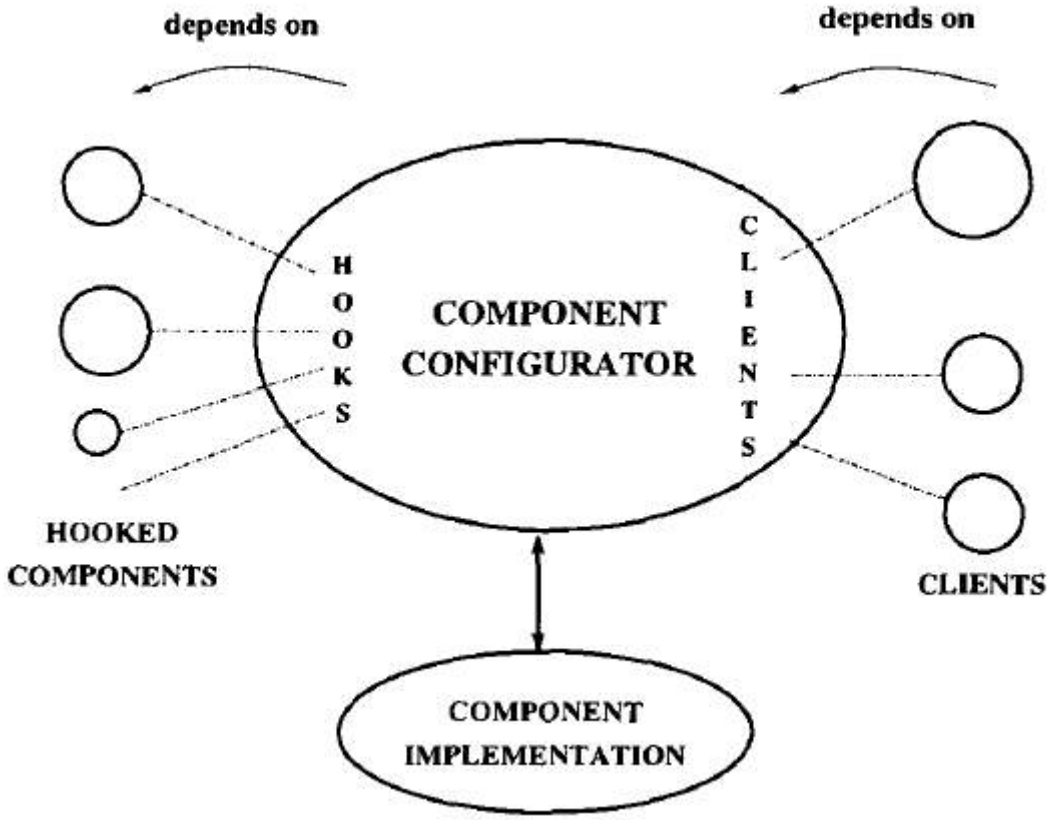

Рисунок 1.4 – Залежності компонентів у системі 2K [12].

Мобільні агенти конфігурації.

Система 2K використовує мобільну реалізацію мобільного агента на основі push для отримання компонентів із сховища компонентів. Ці мобільні агенти складаються з команди для завантаження конфігурацій. За допомогою механізму завантаження конфігурації на основі тяги та натискання система 2K забезпечує більш гнучку інфраструктуру.

Динамічна безпека

Динамічна безпека – ще одна важлива особливість системи 2K. Інтерфейси CORBA використовуються як обмеження для послуг 2K. У прототипі 2K Херувимська система безпеки [13] використовується для підтримки динамічних політик безпеки. Політику безпеки системи 2K можна змінювати залежно від ситуації.

У системі 2K конфігурації безпеки можуть змінюватися динамічно. Це підтримує багато додатків, які працюють поверх архітектури 2K системи.

1.8 Безпека в розподілених системах

Безпека є одним з основних аспектів розподілених систем. Це надзвичайно важливо, оскільки розподілені системи доступні та обмінюються інформацією через мережевий носій. Якщо така система не відповідає надійній політиці контролю доступу, це може спричинити витік інформації. Автори статті Безпека в розподілених системах [3] описують три основні політики контролю доступу щодо розподілених систем. Ці політики контролю доступу є обов’язковим контролем доступу, дискреційним контролем доступу. На додаток до цих політик контролю безпеки, безпека в мережевому середовищі також є критично важливою [3].

Обов’язковий контроль доступу

Обов’язковий контроль доступу або MAC – це механізм контролю доступу, де операційна система контролює доступ до об’єкта або цілі. Як правило, предметом може бути об'єкт або нитка. MAC контролюється адміністратором політики безпеки. Інші користувачі не мають ніяких привілеїв, щоб замінити політику. Вони не можуть видавати доступ до файлів. MAC дозволяє адміністраторам політик застосовувати політики безпеки для всієї організації [3]. У рамках MAC користувачі, крім адміністративних користувачів, не можуть замінити або змінити цю політику випадково чи навмисно. Це дозволяє адміністраторам безпеки визначити центральну політику, яка гарантується (в принципі) для всіх користувачів.

Дискреційний контроль доступу

Дискреційний контроль доступу або ЦАП – це політика контролю доступу, при якій доступ до об’єктів обмежується залежно від суб’єктів або груп, де ці об’єкти належать [3]. Контроль доступу є дискреційним, оскільки суб’єкт, який має певний дозвіл на доступ, може передати дозвіл будь-якому іншому суб’єкту. За допомогою ЦАП ми можемо передавати предмет між собою.

Безпека в мережевих середовищах

Розподілені системи використовують мережі як носій зв'язку. Мережа розподіленої системи також повинна бути захищена. Мережа може знаходитись у фізично захищеному фізично захищеному середовищі або незахищеному відкритому середовищі.

Маркування

Багаторівнева інформація повинна бути встановлена в мережі. Повідомлення, що надсилаються через мережу, повинні бути позначені. Ці ярлики повинні містити рівень безпеки. Необхідно забезпечити цілісність етикеток. Коли спілкування відбувається через відкриту мережу, цілісність ярликів не може бути гарантована. Тому потрібно використовувати такий механізм, як Crypto Sealing

Шифрування

Коли передача даних відбувається через відкриту мережу, шифрування даних має важливе значення для захисту. Це шифрування може виконуватися на рівні зв'язку або пакета. Багаторівневі дані вимагають використання обладнання для шифрування даних, яке відповідає вимогам COMSEC [3].

Аутентифікація ядра

У мережевому середовищі важливо мати аутентифікацію на рівні ядра. Зазвичай розподілені операційні системи функціонують у широкомовних мережах, тому систему відкритих ключів можна використовувати для взаємної автентифікації.

1.9 Розподілені бази даних та сховища даних

Розподілене сховище даних – це механізм зберігання даних, де він зберігає дані у декількох вузлах у реплікаційному режимі. Це може бути або розподілена база даних, яка копіює дані в декількох екземплярах, або однорангові доступні комп'ютерні вузли, які можуть зберігати інформацію про користувача.

Розподілені бази даних зазвичай називаються нереляційними базами даних, які виконують можливості запитів і доступні в кількості розподілених вузлів, які також можуть бути відтворені в декількох екземплярах. Хоча деякі розподілені бази даних надають розширені можливості запитів, деякі бази даних обмежені семантикою сховища ключ-значення. BigTable від Google – це розподілена база даних, яка має більше можливостей, ніж розподілена файлова система або однорангова мережа. DB Dyanamo від Amazone і Windows Azure Storage є ще одним прикладом для розподілених баз даних.

Обробка транзакцій

Обробка розподілених транзакцій відбувається, коли задіяні два або більше мережевих хостів. Менеджер транзакцій та транзакційні ресурси – це дві основні сутності, що стосується розподілених транзакцій. Менеджер транзакцій відповідає за управління та створення глобальних транзакцій, пов’язаних із ресурсами транзакцій. Розподілені транзакції також мають властивості ACID (атомність, послідовність, ізоляція та довговічність), як і будь-які інші транзакції. Різні реалізації розподілених баз даних використовують різний механізм обробки транзакцій. Гайковерт: Глобально розподілена база даних Google [6] використовує техніку на основі часових позначок для управління розподіленими транзакціями.

Послідовність і толерантність до несправностей

Синхронізація кожного вузла в мережі, включаючи кеш, якщо він доступний, з останніми модифікаціями даних, має вирішальне значення для розподілених сховищ даних. Це також можна описати як консистенцію. Здатність системи функціонувати у випадку часткової несправності можна назвати стійкістю до несправностей. Навіть якщо система продовжує функціонувати, загальна продуктивність може вплинути. Відмовостійкість в основному досягається тиражуванням та контрольними пунктами. Основна ідея реплікації – отримати доступ до даних за відсутності вузла через відмову іншого вузла. Збереження стабільного стану системи в стабільному сховищі

називається контрольним наведенням. У разі відмови система відновлюється у попередній стабільний стан, що зберігає корисну інформацію.

Ефективність

Ефективність розподіленої системи в основному залежить від затримки зв'язку та затримки вводу-виводу. Різні системи впровадили багато методів для покращення згаданих точок затримки на користь продуктивності системи. Механізми кеш-пам’яті відіграють важливу роль у продуктивності розподіленого сховища даних.

Кешування

Розподілені сховища даних та бази даних значною мірою покладаються на дані кешу пам’яті так само, як традиційна концепція кешу локальної мови. У розподіленому сховищі даних або кеші бази даних може охоплювати багато серверів, які розташовані віддалено. Основна пам'ять дешевшала, що дозволяє кешувати життєздатне рішення для підвищення продуктивності. Керуючий рівень ТАО забезпечує посередництво між клієнтом та базами даних, забезпечуючи хорошу продуктивність зчитування [5].

Балансування робочого навантаження

Балансування *робочого* навантаження – ще одне важливе завдання, яке виконують розподілені системи баз даних. Балансування навантаження – це спосіб розподілу одиниць навантаження по декількох вузлах на основі порогового значення. За допомогою балансування навантаження очікується, що кожен вузол буде зайнятий однаково, щоб закінчити роботу приблизно в один і той же час. Це максимізує використання ресурсів із більшою пропускною здатністю. Я також дозволяю розширюваність та поступове зростання.

1.10 Зв'язок у розподілених системах

Розподілену систему та уніпроцесорну систему можна диференціювати головним чином за міжпроцесорним зв’язком. Уніпроцесорні системи в

основному спираються на спільну пам’ять, коли йдеться про міжпроцесорний зв’язок. Прикладом цього є проблема виробник-споживач. За відсутності спільної пам'яті розподілені системи покладаються на передачу повідомлень. У розподілених системах не відбувається взаємодії між процесами спільної пам'яті.

Протоколи у зв'язку

Протоколи можна класифікувати за рівнями, визначеними в еталонній моделі OSI

– Протоколи нижчого рівня: Визначення протоколів на фізичному рівні, каналі передачі даних та мережі.

– Транспортні протоколи: передача повідомлень між клієнтами, включаючи їх розрив.

– Входить до пакетів, контролюючи низький рівень (TCP і UDP без підключення).

– Протоколи високого рівня: Визначте протоколи на рівні сеансів, презентацій та додатків.

Види спілкування

– Стійкий або тимчасовий.

– Асинхронний або синхронний.

– Клієнт / сервер.

– Дискретна або потокова передача

Виклик віддаленої процедури

Віддалений виклик процедур (RPC) – це виклик функції або підпрограми, що відбувається між різними комп’ютерами в мережі без необхідності розуміти деталі мережі. Зазвичай RPC вимагає послуги у однієї програми, яка працює на іншому комп’ютері.

Асинхронні RPC

Асинхронна модель RPC надає можливості підтримувати кілька видатних викликів RPC в одному потоковому клієнті, відокремлюючи повернені значення. У традиційних синхронних викликах RPC клієнт

блокується під час виклику віддаленої процедури, поки виклик не повернеться, що є обмеженням синхронних викликів віддаленої процедури.

Надійні RPC

Визначте механізми обробки для помилок викликів віддалених процедур.

– Клієнт не може знайти сервер.
– Клієнтський запит втрачено.
– Збій сервера.
– Відповідь сервера втрачена.
– Клієнт аварійно завершує роботу.

Орієнтоване на повідомлення спілкування

Комунікація між процесами може сприяти спілкуванню, орієнтованому на повідомлення, яке відповідає подіям. Зв’язок між процесами відбувається щодо подій. Дані про події доставляються через повідомлення. Існує дві основні категорії для спілкування, орієнтованого на повідомлення. Це синхронне або асинхронне спілкування та перехідне або постійне спілкування. При синхронному спілкуванні і обмінником повинні бути задіяні як відправник, так і одержувач. Асинхронне спілкування не вимагає активності обох приймачів під час надсилання повідомлення. Вони нещільно зв'язані. Перехідний або постійний характер спілкування, орієнтованого на повідомлення, визначає кількість часу збереження повідомлень. Перехідні зв’язки дозволяють повідомленням бути живими лише тоді, коли обидва процеси виконуються.

1.11 Протоколи трансляції для розподілених систем

Типова локальна мережа може передбачати використання широкомовного зв'язку. Іноді процесори зазвичай вибирають широкомовне спілкування для використання в якості середовища зв'язку серед процесорів. Трансляційні повідомлення можуть отримувати чи не отримувати всі інші

процесори. Передбачається, що широкомовні повідомлення відповідають вимогам протоколів Trans та Total, описаних нижче [7].

Протокол Trans, який гарантує, що кожне повідомлення транслюється або отримується будь-яким робочим процесором, і є ефективним протоколом. Протокол Total, який з великою ймовірністю оперативно розміщує загальний порядок на широкомовних повідомленнях, гарантуючи, що навіть за наявності несправностей всі робочі процесори узгоджують абсолютно однакову послідовність широкомовних повідомлень [7].

1.12 Розподілені файлові системи (DFS)

Розподілені файлові системи (DFS) дозволяють користувачам працювати з файлами, які фізично знаходяться на різних машинах / віддалених дисках. (Користувач зазвичай бачить файли у віддалених системах, як ті, що знаходяться на локальному диску). Оскільки файл зберігається на певній клієнтській машині, це дозволяє декільком користувачам отримати доступ до файлу. Слід керувати кількома клієнтами, які отримують доступ до одного файлу, а також забезпечувати узгодженість та надійність. І однорідність апаратного та програмного забезпечення не потрібна для роботи з DFS.

На додаток до вищезазначених основних цілей, DFS має такі переваги, як покращена надійність (інформація копіюється на декілька серверів, отже, файли доступні, а інформація не втрачається, навіть якщо один або кілька серверів не працюють) та бездискові робочі станції. (DFS дозволяє клієнтам працювати з файлами без диска, що існувало раніше. Попри те, що сьогодні використовуються бездискові робочі станції, концепція доступу до файлів з бездискових робочих станцій знецінюється. [10] Обговорюється питання впровадження бездискових робочих станцій, які підключені до файлових серверів за допомогою локальної мережі)

Найпопулярнішими розподіленими файловими системами є такі, і обговорюється їх поведінка в процесі експлуатації та дизайнерські рішення.

- NFS (мережева файлова система).
- AFS (файлова система Ендрю).

Рішення щодо проектування в розподілених файлових системах

- Як називати файли / прозорість (як відбувається монтування)
- Рішення щодо віддаленого доступу / кешування.
- Сервер без громадянства / повноцінного стану.

Присвоєння імен та прозорість

Прозорість розташування – назва файлу не повинна вказувати на фізичне розташування файлу. Таким чином, користувач бачить, що всі файли зберігаються на диску його комп'ютера, і він може отримати доступ до файлів без відома фізичного розташування файлу.

Незалежність місцезнаходження – якщо місце розташування файлу змінено, чи потрібно змінювати ім'я файлу. Зазвичай DFS не мають властивості незалежності від розташування, оскільки ім'я файлу використовується файловою системою для пошуку фізичного розташування (або розташування, якщо файл розсіяний), а зміна фізичного розташування призведе до зміни імені файл. (наприклад: -NFS)

Абсолютна назва

Ім'я файлу складається з імені сервера та фактичного файлу шляху. При такому підході користувач може розрізняти локальні та віддалені файли. Крім того, це усуває властивість відмовостійкості, оскільки ім'я файлу містить ім'я сервера (ім'я віддаленої машини, де знаходиться файл), і якщо ця машина недоступна через проблему (збій або проблема в мережі), а потім файл не доступний клієнту. Отже, базові вимоги DFS не можуть бути досягнуті за допомогою абсолютного методу іменування.

Збереження між іменем, що використовується для ідентифікації файлу, що використовується клієнтом, та фізичним сервером (та шляхом), де знаходиться файл, зберігається. (Така інформація зіставлення зберігається у певному каталозі, а відображення відомі як точки монтування). Цей механізм має прозорість розташування та незалежність від місцезнаходження

(застосовується після перезавантаження клієнта) (наприклад; – Мережева файлова система)

Глобальний простір імен

Усі клієнти використовують одне і те саме ім'я файлу (на відміну від NFS, де клієнти можуть використовувати різні імена для одного віддаленого файлу). Клієнт отримує структуру імен файлів із сервера. (наприклад: – Ендрю Файлова система)

Віддалений доступ та кешування

У методі віддаленого доступу RPC (процедури віддаленого управління) використовуються для читання та запису даних у віддалений файл. (Отже, файл або частина файлу не зберігається в клієнтській пам'яті або на диску).

У методі кешування копія блоку файлу або цілого файлу зберігається на стороні клієнта. Читання та запис виконується з / до локальної копії в кеш-пам’яті на локальній машині. Модель кешування зручна для продуктивності, оскільки читання та запис виконуються з використанням кешованої копії на локальній машині і не залежать від затримок мережі (у моделі RPC затримка мережі завжди впливає на продуктивність). Крім того, для моделі кешування сервера зручна продуктивність у порівнянні з моделлю RPC, оскільки в RPC сервер повинен виконувати читання та запис відповідно до запитів клієнта.

Сервер без повноважень / повноцінний сервер

На серверах без громадянства сервер не відстежує клієнтів, підключених до файлів читання / запису, а також їх статуси підключення. Тому, коли клієнт видає RPC для операцій читання / запису, а сервер читає з / записує у відповідний блок файлів, і відповідь надсилається клієнту. Для кожного запиту на читання / запис серверу потрібно відкрити відповідний файл.

На повноцінних серверах сервер відстежує відкриті файли від клієнтів. Тому паралельні оновлення одного і того ж файлу від різних клієнтів повинні виявлятися та оброблятися сервером. І після перезапуску сервера (після

аварії), сервер повинен відновити стан (або повідомити клієнтів, щоб клієнти могли повторно застосувати та надіслати зміни, зроблені локально)

1.13 NFS (мережева файлова система)

NFS представлений Sun Micro Systems і широко використовується в розподілених системах UNIX. У попередніх версіях NFS (до версії 3) використовувались протоколи без стану, а у версії 4. У NFS кожна машина може виконувати функції сервера та клієнта.

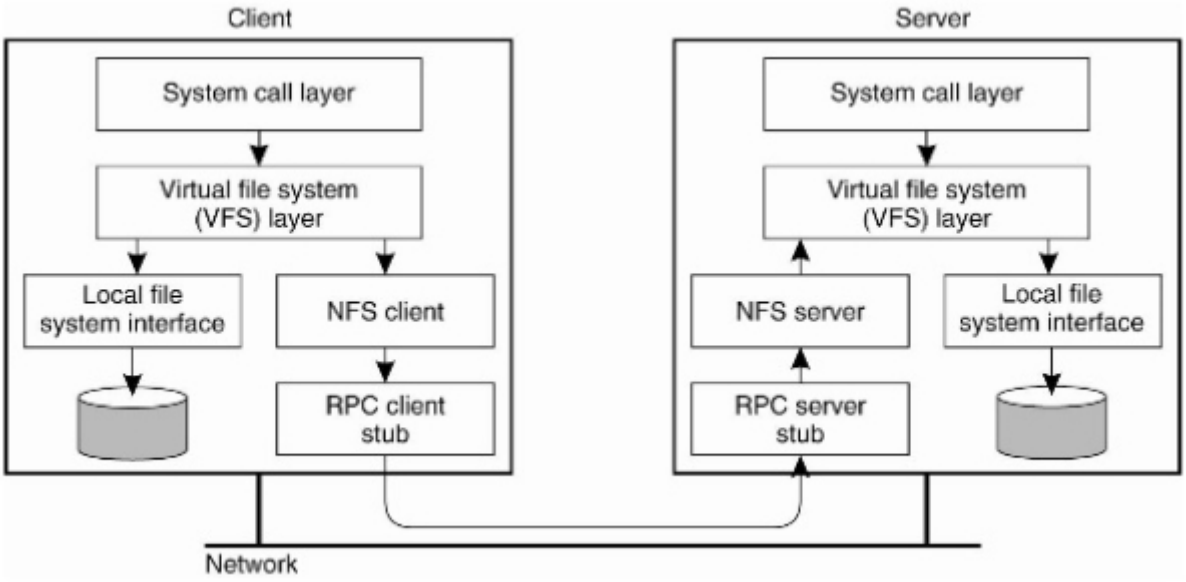

Рисунок 1.5 – Обробка RPC в архітектурі мережевої файлової системи

Системи UNIX, які використовують NFS, використовують інтерфейсний рівень, який називається Віртуальна файлова система (VFS), який забезпечує абстракцію для користувача, щоб користувач не бачив різниці між локальними та віддаленими файлами. Коли клієнт видає та виконує операції введення-виведення через інтерфейс системного виклику, система віртуальних файлів перевіряє, чи призначений виклик для локального файлу або віддаленого файлу. Якщо він призначений для локального файлу, виклик буде перенаправлено до локальної файлової системи (файлова система UNIX).

В іншому випадку виклик перенаправляється на модуль NFS, і модуль NFS видає RPC серверу для виконання фактичної операції вводу-виводу, і сервер надішле відповідь.

Кешування в NFS

Для підвищення продуктивності NFS підтримує локальний кеш в пам'яті в клієнті, де зберігаються нещодавно отримані блоки диска. Це покращує продуктивність розподіленої файлової системи, але додає додаткові накладні витрати, такі як управління невідповідностями кешу. Крім того, якщо клієнт зазнає збою перед надсиланням змін на віддалений сервер, його оновлення в кеші будуть втрачені.

Невідповідність кешу

Невідповідність кешу виникає, коли кілька клієнтів працюють з одним файлом. Деякі конкретні клієнти записують у файл, і сервер застосовує зміни, надіслані цими клієнтами. Оскільки інші клієнти працюють зі своїми локальними копіями файлів або файлових блоків у кеші, ці клієнти можуть посилатися на недійсні / застарілі дані.

Щоб подолати цю проблему, синхронізація клієнта з сервером повинна відбуватися контрольовано. Для мінімізації невідповідності кешу використовуються два методи.

Послідовність, ініційована клієнтом – Клієнт несе відповідальність за перевірку невідповідностей. (Перевірку можна проводити періодично або для кожного доступу до блоку диска)

Послідовність, ініційована сервером – Коли певний клієнт змінює файл, сервер передає зміни клієнтам за допомогою механізму зворотного виклику (наприклад: – Ендрю Файлові Системи)

Моделі оновлення кешу

У цій моделі, як тільки оновлення застосовується до кешу, оновлення надсилається на сервер. Це збільшує мережевий трафік і залежить від затримки мережі, що призводить до зниження продуктивності. Перевага в

продуктивності – це прибутковість лише тоді, коли надсилаються прочитані дзвінки. (Читання здійснюється з кешу)

Запізнілий запис

Оновлення застосовуються до кешу, але не надсилаються на сервер. Зміни передаються на сервер при такій події, як закриття файлу. На процес запису не впливає затримка мережі, оскільки процес продовжується після запису в кеш. Недоліком моделі є те, що якщо клієнт аварійно завершить роботу перед надсиланням оновлення на сервер, зміни будуть втрачені. (У DFS, таких як Andrew File System, кешування здійснюється на рівні диска. Тому цей недолік не застосовується для таких файлових систем)

1.14 Файлова система Ендрю (AFS)

Файлова система Andrew розроблена головним чином з орієнтацією на масштабованість (початкова версія орієнтована на понад 5000 підписників). При розробці цієї моделі робляться припущення / спостереження, такі як «операції зчитування файлів виконуються часто, ніж операції запису» та «більшість файлів мають невеликий розмір у порівнянні з розподілом». (AFS не дає суттєвих переваг, коли вищезазначені припущення недійсні для певної системи).

Також AFS перевершує інші розподілені файлові системи, такі як NFS, з точки зору продуктивності та безпеки. Після первинної реалізації AFS (початкова реалізація була виконана як частина університетської розподіленої файлової системи), на основі цієї моделі – OpenAFS, а розподілені файлові системи IBM-Transarc були розроблені як комерційні продукти.

AFS використовує повноцінний сервер де на сервері відстежується інформація про з’єднання відкритих файлів від кожного клієнта. (У AFS сервер називається «Vice», а клієнта «Venus»). Також AFS використовує локальний кеш диска, щоб зберігати повну копію віддаленого файлу для читання та запису. (У NFS кеш пам'яті використовується для копіювання

файлових блоків). Ця функція дозволяє клієнтові працювати над локальним кешем, коли файл копіюється на локальний диск, не використовуючи механізми зв'язку запиту даних із сервера. Всі зчитування та запис виконуються з / в локальний кеш (накладні витрати на локальний диск / введення все ще є). Як результат, продуктивність AFS краща, ніж інші традиційні розподілені файлові системи. Якщо клієнт виконує лише операції читання, у порівнянні з DFS на основі RPC, AFS обіцяє забезпечити чудову продуктивність.

Оскільки RPC не використовується, сервер має мінімальні переривання від клієнтів, що полегшує масштабованість . (Сервер може обробляти більшу кількість клієнтів, оскільки навантаження операцій на стороні сервера зменшується через кешування локального диска).

Також ця модель дозволяє клієнтові працювати без існування sever, оскільки клієнт працює з локальною копією, і якщо сервер виходить з ладу, сервер повинен відновити стан, завантаживши деталі підключення клієнта відповідно. Крім того, якщо клієнт аварійно завершує роботу перед відправкою файлу оновлення на сервер (або зміни не застосовуються на стороні сервера через перебої в роботі мережі), усі зміни клієнтів будуть втрачені. Клієнт може перезапустити та продовжувати працювати над кешованою копією, збереженою на диску перед збоєм, якщо кеш все ще дійсний. (Запит перевірки кешу надсилається на сервер клієнтом після перезапуску клієнта)

Локальні зміни клієнта застосовуються до сервера, коли клієнт закриває файл. Після застосування цих змін сервер надсилає оновлену копію всім іншим клієнтам через механізм зворотного дзвінка, відомий як обіцянка зворотного дзвінка, і інші клієнти можуть відповідно оновити файл. Але одночасне оновлення від різних клієнтів зазвичай не обробляється успішно. (Оновлення від останнього клієнта зберігається на сервері, замінюючи зміни інших клієнтів). Через цю властивість AFS не рекомендується використовувати для файлів баз даних.

У AFS безпека також розглядається як важливий аспект. AFS використовує протокол Kerberos для автентифікації клієнтів, і клієнти перевіряються за списками контролю доступу (отже, рівні привілеїв можна підтримувати в каталогах (зазвичай в AFS, елементи керування доступом не визначені для файлів)).

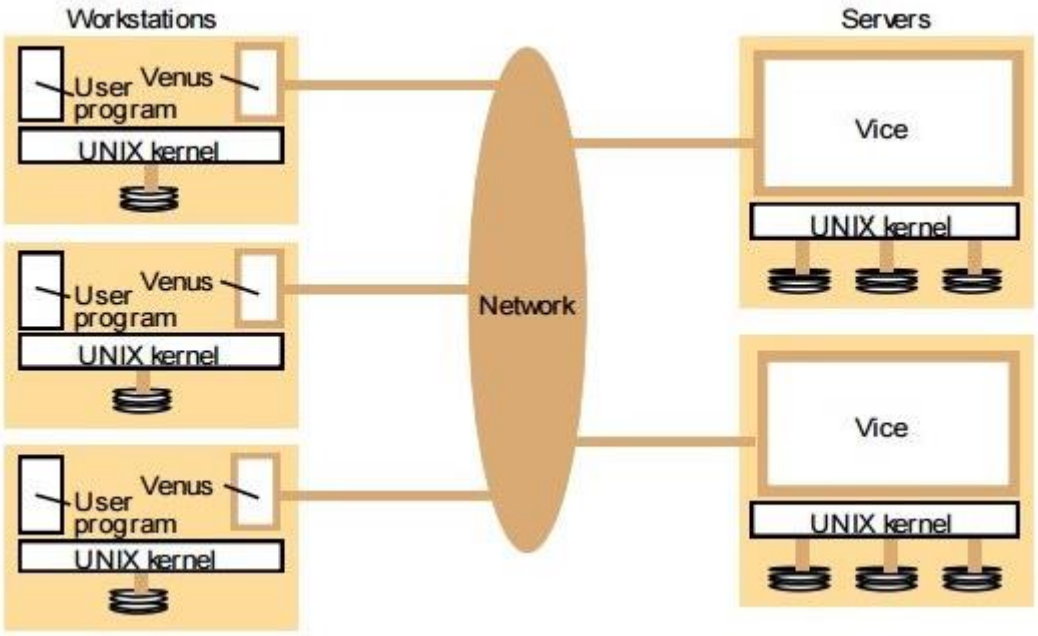

Рисунок 1.6 – Архітектура файлової системи Ендрю

На додаток до згаданих розподілених файлових систем, сьогодні використовується кілька систем. (наприклад: – Файлова система Google (GFS), Розподілена файлова система Hardoop (HDFS)).

1.15 Висновок до першого розділу

Компоненти розподілених систем знаходяться в мережі, в якій спілкування відбувається шляхом передачі повідомлень. Розглядаючи різні аспекти розподілених систем, важливими характеристиками є контроль паралельності, зв'язок та незалежна обробка відмов компонентів.

Архітектури розподілених систем усувають основні недоліки централізованих систем, такі як висока чутливість до збоїв, відсутність

масштабованості. Однак розподіленим системам властиві такі недоліки, як складність та накладні витрати на зв'язок через відсутність спільної пам'яті. Цей звіт також охоплює основні три типи розподілених систем клієнт-серверні системи, інтегровані системи управління файлами та інтегровані розподілені системи.

Спілкування між різними обчислювальними хостами в мережі є важливим аспектом при вивченні розподілених систем. Механізми зв'язку в сучасних розподілених системах можна розділити на три великі категорії: передача повідомлень, механізм порту та віддалені виклики процедур.

Архітектура розподіленої операційної системи під назвою 2K пояснюється в підрозділі розділі «Розподілені операційні системи». Ця операційна система вирішує основні проблеми розподілених систем, такі як управління ресурсами, динамічна адаптованість та конфігурація розподілених додатків на основі компонентів.

РОЗДІЛ 2. ВИКОРИСТАННЯ PROXY-З'ЄДНАНЬ

У донному розділі наводиться, як налаштувати NGINX і NGINX Plus на прийняття протоколу Proxy, як переписати IP-адресу балансира навантаження або проксі-сервер на адресу, отриману в заголовку протоколу Proxy, налаштувати просту реєстрацію IP-адреси клієнта та активувати протокол Proxy між NGINX та сервером TCP вищого рівня.

Протокол Proxy дозволяє NGINX та NGINX Plus отримувати інформацію про підключення клієнта, що передається через проксі-сервери та балансири навантаження, такі як HAproxy та Amazon Elastic Load Balancer (ELB).

За допомогою протоколу Proxy NGINX може дізнатись вихідну IP-адресу з HTTP, SSL, HTTP / 2, SPDY, WebSocket та TCP. Знання похідної IP-адреси клієнта може бути корисним для встановлення певної мови веб-сайту, ведення списку IP-адрес або просто для ведення журналу та статистики.

Інформація, що передається через протокол Proxy, – це IP-адреса клієнта, IP-адреса проксі-сервера та обидва номери портів.

Використовуючи ці дані, NGINX може отримати вихідну IP-адресу клієнта кількома способами:

- За допомогою змінних $proxy_protocol_addrand, $proxy_protocol_portякі фіксують вихідну клієнтську IP-адресу та порт. $remote_addri$remote_port змінні захопити IP – адреса і порт балансування навантаження.
- За допомогою модуля RealIP, який переписує значення у змінних $remote_addrand $remote_port, замінюючи IP-адресу та порт балансувача навантаження на вихідну IP-адресу та порт клієнта. $realip_remote_addrI $realip_remote_portзмінні зберігають адресу і порт балансування навантаження, а також $proxy_protocol_addri $proxy_protocol_portзмінні зберігають вихідний IP – адреса клієнта і номер порту в будь-якому випадку.

Передумови

– Щоб прийняти протокол Proxy v2, NGINX Plus R16 та новіших версій, або NGINX з відкритим кодом 1.13.11 та новіших версій

– Щоб прийняти протокол Proxy для HTTP, NGINX Plus R3 та новіших версій, або NGINX з відкритим кодом 1.5.12 та новіших версій

– Для підтримки протоколу TCP на базі клієнта Proxy, NGINX Plus R7 та новіших версій або NGINX з відкритим кодом 1.9.3 та новіших

– Щоб прийняти протокол Proxy для TCP, NGINX Plus R11 та новіших версій, або NGINX з відкритим кодом 1.11.4 та новіших версій

– Модулі Real-IP для HTTP та Stream (TCP) за замовчуванням не входять у відкритий код NGINX; см Установка NGINX Open Source для деталей. Для NGINX Plus не потрібні додаткові кроки.

2.1 Налаштування NGINX для прийняття протоколу Proxy

Щоб налаштувати NGINX на прийняття заголовків протоколу Proxy, додайте proxy_protocol параметр до listen директиви в server блоці в блоці або .http {}stream {}.

```
http {
    #...
    server {
        listen 80   proxy_protocol;
        listen 443  ssl proxy_protocol;
        #...
    }
}

stream {
    #...
    server {
        listen 12345 proxy_protocol;
        #...
    }
}
```

Рисунок 2.1 – Налаштування NGINX на прийняття заголовків протоколу Proxy

Тепер ви можете використовувати змінні $proxy_protocol_addrand $proxy_protocol_portдля IP-адреси та порту клієнта, а також додатково налаштувати модулі HTTP та Stream Real-IP для заміни IP-адреси балансувача навантаження у змінних $remote_addrand $remote_portз IP-адресою та портом клієнта.

2.2 Авторизація проксі

Авторизація проксі-сервера - це параметр, який ви вмикаєте в політиці, щоб вимагати, щоб проксі-сервер ETP санкціонував підключення від локального проксі-сервера в конфігурації ланцюжка проксі-серверів. Цей параметр додає заголовок Proxy-Authorization до цих підключень. Заголовок проксі-авторизації містить облікові дані проксі, які використовуються для автентифікації локального проксі. Проксі-сервер ETP перевіряє ці облікові дані, перш ніж дозволить підключення від локального проксі-сервера.

Щоб налаштувати авторизацію проксі-сервера, потрібно:

– Налаштуйте облікові дані проксі. Цей процес передбачає створення імені користувача та пароля. Інструкції див. У розділі Створення облікових даних проксі .

– Налаштуйте ці облікові дані проксі в локальному проксі. Інструкції щодо налаштування цих облікових даних у Squid див. У розділі Налаштування Squid для переадресації трафіку на проксі ETP .

– Виберіть, щоб довіряти заголовку X-Forwarded-For у політиці. Інструкції див. У розділі Увімкнення повноцінного веб-проксі .

– Увімкніть параметр Авторизація проксі-сервера в політиці. Інструкції див. У розділі Увімкнення повноцінного веб-проксі або Увімкнення авторизації проксі

Авторизація проксі використовує базову схему автентифікації. Повноваження в заголовку авторизації проксі кодуються base64. HTTPS та TLS додатково захищають ці дані в заголовку.

У випадках, коли проксі-сервер ETP не може перевірити запит, з'являється повідомлення про помилку браузера. Наприклад:

– Якщо автентифікація не вдається, повідомлення про помилку браузера вказує на те, що автентифікація не вдалася.

– Якщо автентифікація проксі-сервера ввімкнена в політиці, а облікові дані проксі-сервера не налаштовано, повідомлення про помилку браузера вказує на необхідність автентифікації проксі.

Якщо авторизація проксі-сервера не ввімкнена в політиці для конфігурації ланцюжка проксі-сервісів, запити приймаються проксі-сервером ETP, якщо вони надходять з відомого місця.

2.3 Зміна IP-адреси балансувача навантаження на IP-адресу клієнта

Ви можете замінити адресу балансувача навантаження або проксі TCP на IP-адресу клієнта, отриману з протоколу Proxy. Це можна зробити за допомогою модулів HTTP та Stream Real-IP. За допомогою цих модулів змінні $remote_addrand $remote_portзберігають реальну IP-адресу та порт клієнта, тоді як $realip_remote_addri$realip_remote_port

Щоб змінити IP-адресу з IP-адреси балансувача навантаження на IP-адресу клієнта:

– Переконайтеся, що ви налаштували NGINX на прийняття заголовків протоколу Proxy. Див. Розділ Налаштування NGINX для прийняття протоколу Proxy .

– Переконайтеся, що ваша інсталяція NGINX включає модулі HTTP та Stream Real-IP:

– Якщо ні, перекомпілюйте NGINX з цими модулями. Детальніше див. У розділі Встановлення відкритого коду NGINX . Для NGINX Plus додаткові кроки не потрібні.

– У set_real_ip_fromдирективі для HTTP , Stream або обох вкажіть IP-адресу або діапазон адрес CIDR адрес проксі-сервера TCP або балансування навантаження:

У контексті змініть IP-адресу балансувача навантаження на IP-адресу клієнта, отриману із заголовка протоколу Proxy, вказавши параметр до директиви:http {}proxy_protocolreal_ip_header

Запис вихідної IP-адреси

Коли ви знаєте оригінальну IP-адресу клієнта, ви можете налаштувати правильний журнал/

Для HTTP налаштуйте NGINX для передачі клієнтської IP-адреси вищим серверам, використовуючи $proxy_protocol_addr змінну з proxy_set_header директивою:

Додайте $proxy_protocol_addrзмінну до log_formatдирективи (HTTP або Stream).

Протокол Proxy для підключення TCP

Для потоку TCP можна ввімкнути протокол Proxy для з'єднань між NGINX та вихідним сервером. Щоб увімкнути протокол Proxy, включіть proxy_protocolдирективу в serverблок на рівні:stream {}

Передбачається, що перед NGINX є балансир навантаження для обробки всього вхідного трафіку HTTPS, наприклад Amazon ELB. NGINX приймає HTTPS-трафік на порту 443 ( ), TCP-трафік на порту 12345, а також приймає IP-адресу клієнта, передану з балансувача навантаження через протокол Proxy (параметр до директиви в ilisten 443 ssl;proxy_protocollistenhttp {}stream {}

NGINX завершує трафік HTTPS (the ssl_certificateта ssl_certificate_keyдирективи) і передає розшифровані дані на серверний сервер:

- Для HTTP: proxy_pass http://backend1;
- Для TCP: proxy_pass backend.example.com:12345

Він включає IP-адресу клієнта та порт із proxy_set_header $proxy_protocol_addrМінлива , зазначена вlog_format.

Крім того, TCP-сервер (блок) надсилає власні дані протоколу Proxy на свої серверні сервери (директива).stream {}proxy_protocol on.

2.4 HTTP Proxy та SOCKS-Proxy сервера

Проксі-сервери – чудовий інструмент, який допоможе вам отримати доступ до будь-яких веб-сайтів або сторінок за допомогою вашого браузера.

Ваша особа залишається прихованою . Наприклад, якщо ви хочете отримати доступ до даних від компанії за кордоном, все, що вам потрібно зробити, це використати проксі-сервер для проведення цієї дії.

Якщо веб-сайт дозволяє користувачам доступ лише з їхньої країни, проксі-сервер приховає ваше місцезнаходження та зробить вигляд, що ви проживаєте з місця проживання цієї компанії. Ніхто не здогадається, що ви, що сидите в Індії, отримуєте доступ до даних компаній за кордоном.

Ви можете легко виконати цю дію, якщо у вас у руках правильні проксі-сервери. Два найкращих називаються HTTPS проксі і SOCKS проксі.

Ці два найпоширеніші, але їх особливості різні. Щоб допомогти вам вирішити, який проксі-сервер вам найбільше підходить, ми перелічимо різницю, використовуючи простішу термінологію.

Під час пошуку у веб-браузері ви знайшли багато посилань, які починаються на "https". Побачивши це, ви повинні зрозуміти, що веб-сторінка використовує протокол HTTP для функціонування.

Проксі Http функціонують через клієнтську та серверну моделі. Наприклад, будучи клієнтом, ви вводите веб-браузер, щоб отримати доступ до певної інформації. Ваш протокол HTTP надішле запит на ваш доступ у вигляді посилання на URL-адресу. Сервер з іншого боку відповість на це посилання разом із інформацією, яку ви запитували назад до вас. Ресурс, який ви отримаєте, також буде у форматі HTTP.

У HTTP ви можете виконувати більш безпечні дії за допомогою декількох проксі-серверів, доступних сьогодні на ринку. Проксі-сервер HTTP функціонуватиме так само, як і раніше, але з проксі-сервером; буде легше захистити свою особу.

Зрозумівши, що таке https-проксі, давайте перейдемо до наступного популярного проксі-сервера.

Розглянемо що таке проксі SOCKS. Lightproxies стверджує: " SOCKS – це тип інтернет-протоколу, який надсилає мережеві пакети від сервера до клієнта і навпаки через проксі-сервер. "

Проксі-сервер SOCKS, який називається Socket Secure, керує інформацією в Інтернеті за допомогою проксі-сервера.

Тут проксі-сервер SOCKS використовує свій протокол SOCKS. На відміну від проксі-серверів https, проксі-сервери SOCKS є більш надійними в порівнянні з https.

Оскільки ви тепер коротко розумієте, які функції цих двох проксі-серверів, ми можемо перейти до вирішальної частини вибору, який проксі-сервер буде ідеальним для вас.

Різниця між проксі-серверами SOCKS та HTTP

Таблиця 2.1 – Функціональність

Проксі HTTPS	**SOCKS**
Функціонує за допомогою протоколу http і є життєво важливим для тих користувачів, які використовують його з метою отримання інформації через веб-браузер	Не використовує проксі-сервер http і вважається проксі-сервером нижчого рівня, оскільки він використовується в основному для загальних цілей

Проксі-сервери HTTPS використовують лише протокол HTTP, що означає, що їх функціональність залежить від того, що вони виконують.

Оскільки вони проводять дію отримання інформації через веб-браузер, вони найкраще підходять для користувачів, які використовуватимуть її з тією ж метою.

На відміну від https, SOCKS функціонує дещо інакше. Він не функціонує за допомогою протоколу HTTP. Це не обмежує дозволу користувачам переглядати інформацію в Інтернеті. В ідеалі, SOCKS використовується для більш загальних цілей і вважається проксі-сервером нижчого рівня.

Однак це не є недоліком, оскільки вони не несуть відповідальності за певний набір мережевих протоколів, що надає своїм користувачам гнучкість використання цього проксі-сервера де завгодно.

Таблиця 2.2 – Безпека

Проксі HTTPS	**SOCKS**
Високі шанси на перегляд ваших даних під час процесу	Низькі шанси на перегляд ваших даних, оскільки SOCKS не може прочитати дані

Весь порядок використання проксі-сервера полягає в тому, щоб залишатися в безпеці та охоплювати під час доступу до інформації в Інтернеті. Як вже згадувалося раніше, проксі-сервери є чудовим середовищем для захисту вашої діяльності в Інтернеті.

Найкраща частина полягає в тому, що проксі-сервери SOCKS використовують його, отже, забезпечуючи хорошу вагу безпеки у порівнянні з HTTP-проксі.

З HTTP-проксі-серверами є ймовірність перегляду та запису ваших даних під час проксі-сервера клієнт-сервер. Якщо це трапляється, ви залишаєтесь під високим ризиком. З проксі-сервером SOCKS ця проблема не може виникнути, оскільки проксі-сервер не може зчитувати дані.

Але якщо ви продовжуєте використовувати HTTP-проксі, ви можете залишатися захищеними в Інтернеті, встановивши тунельне з'єднання. Тунель функціонує як стіна, де це ускладнює відстеження вашої діяльності в Інтернеті і робить вас більш безпечними та надійними для подальшого функціонування.

Таблиця 2.3 – Швидкість

Проксі HTTPS	**SOCKS**
Пропонує приватний та публічний проксі, залежить від вибору (публічний проксі-повільний)	Загальний сервер, швидкість швидка

Проксі-сервери HTTP пропонують вам два типи проксі-серверів, в які можна інвестувати, коли йдеться про швидкість, приватні та публічні проксі-сервери.

За допомогою приватних довірених осіб ви можете самостійно керувати своєю діяльністю в Інтернеті без зайвих клопотів. З державними довіреними особами все навпаки. Оскільки користувачі високі, швидкість залежить від того, де Інтернету знадобиться багато часу, щоб навіть допомогти вам отримати доступ до веб-сторінки, яку ви хочете побачити.

Оскільки проксі-сервер SOCKS є загальним, він простіший і швидший зі своєю швидкістю. Це зменшує необхідність застосування будь-якого коду, а отже, ідеально передавати або завантажувати дані з Інтернету.

Також переконайтеся, що окрім цих проксі-серверів, вам також потрібно шукати добре відомі проксі-компанії, які пропонують цю послугу, оскільки завдяки пропонованим ними серверам Інтернет та швидкість пропускної здатності також можуть допомогти вам краще функціонувати для всіх ваших онлайн-дій.

Таблиця 2.4 – Підключення до інструментів

Проксі HTTPS	SOCKS
Підключається до будь-якого інструменту	Підключається до обмежених інструментів

Кожен із цих двох довірених осіб повинен мати можливість зв’язатись із використовуваними вами інструментами.

Проксі-сервери HTTP здебільшого з'єднуються з усіма інструментами, але з проксі-сервером SOCKS ви, можливо, не зможете підключити занадто багато пристроїв у порівнянні з HTTPS. Отже, важливо відстежувати, які інструменти ви використовуєте найбільше, а потім вирішувати, який проксі-сервер ви хочете застосувати.

Найкращий спосіб зробити це – дослідити різні проксі-сервери, доступні на ринку.

Він має різні варіанти на вибір, а також має спеціальний розділ для користувачів проксі-сервера SOCKS, де ви можете скористатися багатьма перевагами, які вони пропонують під ним. Крім того, ви можете відстежувати інструменти, які підходять для проксі HTTPS та SOCKS, щоб ви могли легко вирішити, який із них можна використовувати.

З появою нових технологій онлайн-платформи стає набагато легше зламати. Перегляд інформації та вилучення важливих даних з неправильною метою закликали компанії вживати суворих заходів у всій своїй діяльності в Інтернеті.

З урахуванням вищезазначених відмінностей, для користувачів, які сильно залежать від завантаження та передачі величезних обсягів даних, SOCKS буде ідеальним варіантом для вас. Для тих, хто хоче завантажити менший обсяг даних, ви можете вибрати проксі HTTPS.

Коли справа доходить до вибору проксі-сервера, щоб гарантувати, що ви інвестуєте в бренд, який має на увазі три речі:

- Зберігає вас анонімним.

– Захищає вашу діяльність в Інтернеті без будь-яких клопотів та ризиків.

– Має хороший варіант швидкості Інтернету.

2.5 Анонімні Proxy

Анонімні проксі-сервери, як правило, використовуються для обходу політик безпеки, дозволяючи користувачам отримувати доступ до заборонених рекреаційних сайтів, сайтів для дорослих та інших некомерційних підприємств шляхом тунелювання цього трафіку через звичайний або зашифрований сеанс HTTP. Анонімні довірені особи також забезпечують анонімність; користувачів, які отримують доступ до веб-сайтів через анонімний проксі, неможливо легко простежити до їх початкової IP-адреси .

Exinda Appliances має вбудовану підтримку для виявлення анонімного проксі. Exinda Appliance отримує щоденні оновлення від www.exinda.com , що містить оновлені анонімних визначення проксі - сервера, так само, як антивірусні програми , отримувати щоденні оновлення загроз.

Анонімний проксі-додаток - це спеціальний об’єкт програми, який використовується для виявлення анонімних проксі-сайтів та служб. Однак послуга анонімного проксі вимкнена за замовчуванням.

Якщо послугу анонімного проксі включено, пристрій Exinda щодня отримує список визначень анонімних проксі з веб-серверів Exinda .

Об'єкт програми під назвою "Анонімний проксі" створюється автоматично. Програма Anonymous Proxy відстежує весь трафік, що надходить через одного з анонімних проксі-серверів у списку. Цей об'єкт програми відображається у звітах моніторингу, як і будь-який інший об'єкт програми, а також може використовуватися в політиках оптимізатора.

2.6 Кібератаки на проксі-сервери

Сьогодні багато організацій усвідомлюють, що захист від DDoS має вирішальне значення для забезпечення виняткової взаємодії з клієнтами. Чому? Оскільки ніщо не зменшує час завантаження та не впливає на досвід кінцевого користувача більше, ніж кібератака.

Як фасилітатор доступу до вмісту та мереж, проксі-сервери стали центральним центром для тих, хто прагне викликати горе в організаціях за допомогою кібератак через наслідки успішного нападу.

Напад на проксі-сервер CDN

Нові вразливості в мережах доставки вмісту (CDN) змушують багатьох замислюватися, чи самі мережі вразливі до найрізноманітніших кібератак. Ось п’ять кібер «сліпих плям», на які часто нападають – і способи зменшення ризиків:

Збільшення кількості динамічних атак вмісту.

Зловмисники виявили, що обробка запитів динамічного вмісту є головною сліпою зоною в CDN. Оскільки динамічний вміст не зберігається на серверах CDN, усі запити на динамічний вміст надсилаються на сервери джерела. Зловмисники використовують цю поведінку, щоб генерувати трафік атак, який містить випадкові параметри в запитах HTTP GET. Сервери CDN негайно перенаправляють цей трафік атаки на джерело – очікуючи, що сервер джерела буде обробляти запити. Однак у багатьох випадках сервери джерела не мають можливості обробляти всі ці запити на атаку та не надають онлайнові послуги законним користувачам. Це створює ситуацію з відмовою у наданні послуги. Багато CDN можуть обмежувати кількість динамічних запитів на сервер, що зазнає атаки.

DDoS-атаки на основі SSL.

DDoS-атаки на основі SSL використовують цей криптографічний протокол для орієнтування на онлайн-послуги жертви. Ці атаки легко запускати і їх важко пом'якшити, що робить їх улюбленими хакерами. Для виявлення та пом'якшення атак на основі SSL сервери CDN повинні спочатку

розшифрувати трафік за допомогою ключів SSL замовника. Якщо клієнт не бажає надавати ключі SSL своєму постачальнику CDN, тоді трафік атаки SSL перенаправляється на джерело клієнта. Це робить клієнта вразливим до атак SSL. Такі атаки, які вражають походження клієнта, можуть легко зруйнувати захищену онлайн-послугу.

Під час DDoS-атак, коли задіяні технології брандмауера веб-додатків (WAF), CDN також мають значну слабкість у масштабованості щодо того, скільки SSL-з'єднань в секунду вони можуть обробити. Можуть виникнути серйозні проблеми із затримкою. PCI та інші проблеми дотримання безпеки також є проблемою, оскільки вони обмежують центри обробки даних, які можна використовувати для обслуговування клієнта. Це може збільшити затримку та спричинити проблеми з аудитом.

Майте на увазі, що ці проблеми посилюються завдяки масовій міграції з алгоритмів RSA на алгоритми ECC та DH.

Атаки на послуги, що не належать до CDN

Послуги CDN часто пропонуються лише для програм HTTP / S та DNS. Інші онлайнові послуги та програми в центрі обробки даних клієнта, такі як VoIP, пошта, FTP та власні протоколи, не обслуговуються CDN. Отже, трафік до цих програм не спрямовується через CDN. Зловмисники користуються перевагами цієї сліпої зони та запускають атаки на такі програми. Вони вражають походження замовника масштабними атаками, які загрожують наситити Інтернет-трубу замовника. Усі програми від джерела замовника стають недоступними для законних користувачів після насичення каналу Інтернету, включаючи програми, що обслуговуються CDN.

Прямі IP-атаки

Навіть на програми, які обслуговуються CDN, можна атакувати, як тільки зловмисники здійснюють пряме потрапляння на IP-адресу веб-серверів у центрі обробки даних клієнта. Це можуть бути мережеві атаки, такі як UDP або ICMP, які не будуть маршрутизовані через служби CDN і безпосередньо потраплять на сервери клієнта. Такі об'ємні мережеві атаки можуть наситити

Інтернет-канал. Це призводить до погіршення якості програмних та онлайнових служб, включаючи послуги, що обслуговуються CDN.

Атаки веб- додатків

Захист CDN від загроз обмежений і піддає веб-програми замовника витоку та викраденню даних та іншим загрозам, спільним із веб-додатками. Більшість можливостей WAF на основі CDN є мінімальними, вони охоплюють лише базовий набір заздалегідь визначених підписів та правил. Багато WAF на основі CDN не вивчають параметри HTTP і не створюють позитивних правил безпеки. Тому ці WAF не можуть захистити від атак нульового дня та відомих загроз. Для компаній, які пропонують налаштування веб-додатків у своїй WAF, ціна надзвичайно висока, щоб отримати такий рівень захисту. На додаток до виявлених значних сліпих зон, більшість служб безпеки CDN просто недостатньо реагують, що призводить до конфігурацій безпеки, на розгортання яких потрібно години. Служби безпеки використовують технології (наприклад,

Атаки ззовні

Багато в чому бічні канали є найбільш незрозумілими та затуманеними векторами атак. Цей прийом атакує цілісність сайту компанії за допомогою різноманітних тактик:

- DDoS – постачальник аналітики компанії.
- Атака грубої сили на всіх користувачів або на всі сторонні компанії.
- Портуйте телефон адміністратора та викрадайте дані для входу.
- Велике навантаження на "розміщення сторінок".
- Великі ботнети для "вивчення" входів і виходів сайту.

2.7 Тестування на проникнення: TOR, VPN або проксі

Часто виникає потреба провести повноцінне тестування на проникнення чорної скриньки. Це форма тестування, при якій фахівці з безпеки повинні мати справу з такими речами, як брандмауери та інші механізми обмеження на стороні замовника. Це втручання, поки пентестери виконують перевірки та періодично блокують їх, наприклад, за допомогою IP-адреси або агента користувача.

Якщо ми не дійшли згоди щодо моделі сірого або білого ящиків, і наші IP-адреси не потрапили до білого списку, що ми можемо зробити, щоб обійти ті обмеження, які регулюються замовником та його брандмауером? Тут ми можемо бачити лише одну можливість уникнути цих обмежень: перемикаючи як нашу IP-адресу, так і користувача-агента. Якщо ми говоримо про user-agent тут, все здається простішим, оскільки достатньо буде лише встановити певний плагін для вашого веб-браузера або, наприклад, змінити агенти у вашому сценарії за допомогою певної функції.

То що ми будемо робити з IP-адресою? Нижче я опишу кілька доступних методів, а також їх плюси і мінуси.

Існує багато відмінностей між проксі, VPN і Tor, але їх загальна мета – приховати нашу IP-адресу, замаскувати нашу діяльність, допомогти уникнути обмежень міжмережевого екрану та, в свою чергу, обходити заборони. TOR та проксі є набагато простішими у використанні у випадках, коли вам дуже часто потрібно міняти вашу IP-адресу (наприклад, проксі-ланцюги). Проксі-сервери здаються більш складними для використання, оскільки вони не можуть проксі-сервером весь трафік, а лише для одного порту або служби. Нижче я надаю більш детальну інформацію про кожен із цих варіантів, а також їх плюси та мінуси.

Довіреність

Проксі-сервери бувають декількох форм:

– HTTP передає запити GET / POST і може додати оригінальну IP-адресу до заголовка запиту, а також зберегти повну історію вашої взаємодії з сайтом.

Плюси:

– Анонімність (при правильному використанні).

– Підтримується майже всіма браузерами.

– DNS-запити від імені сервера.

Мінуси:

– Історія сервера.

– Можливість фільтрації та заміни даних за допомогою проксі-сервера.

– Працює лише для протоколу HTTP.

У випадку проксі SOCKS браузер відкриває всі сокети TCP (а іноді і UDP) від імені сервера. У той же час (залежно від браузера) ви можете використовувати свій локальний DNS-сервер, і сайт зможе відстежувати вас за ним, видаючи унікальне ім'я для кожного запиту в своєму субдомені та запам'ятовуючи, з яких адрес надходять запити DNS з них. .

Використання проксі переважно застосовується для сканування та запитів на рівні додатків, а не на рівні мережі. Візьмемо для прикладу nmap; тут ми можемо зустріти кілька проблем. Nmap може виконувати лише CONNECT і SOCKS4, а ці протоколи – лише TCP.

Окрім цього, використання будь-якого типу проксі означає, що nmap взаємодіє зі стеком IP проксі, а не цільового. Це означає, що ICMP-пінг не можна виконати, щоб перевірити, чи хост живий, оскільки ICMP не є TCP.

Плюси:

– Анонімність клієнта (при правильному використанні).

– Можливість переадресації довільного TCP-з'єднання (наприклад, SSH).

– DNS-запити від імені сервера (Google Chrome).

Мінуси:

– DNS-запити від імені клієнта (Firefox).

– Можливість фільтрації та заміни даних за допомогою проксі-сервера.

– Історія сервера.

Головною перевагою в цьому випадку, якщо говорити про тестування на проникнення, є те, що ми зможемо легко розгорнути багато проксі-серверів на основі контейнерів докерів або зображень AWS / DigitalOcean і швидко переключатися між ними. Також певному сценарію / додатку набагато простіше працювати через проксі, оскільки ви можете запускати різні інструменти на різних IP-адресах, на відміну від VPN.

VPN (OpenVPN)

У разі використання VPN найпопулярнішим рішенням буде OpenVPN. Він має безліч корисних функцій, включаючи можливість проходження NAT за допомогою інкапсуляції SSL / TLS, можливість роботи через UDP та багато іншого. За допомогою VPN ви можете змінити IP-адресу на своєму комп'ютері, і це також допоможе вам з трафіком DNS (не забудьте тут перевірити конфігурацію клієнт-сервер, що трафік DNS буде проходити через сервер).

Ви зможете деякий час приховувати свою справжню IP-адресу від брандмауерів, доки не отримаєте нову заборону, але це рішення також ефективно, оскільки немає необхідності в конкретному проксі-порту, інструменті чи чомусь іншому. VPN замаскує весь ваш трафік для всіх портів.

Використання VPN (OpenVPN) сьогодні дуже популярне для будь-яких інструментів і ідеально підходить як для роботи на рівні програми, так і на рівні мережі. За допомогою VPN ви можете легко виконувати сканування мережі, не турбуючись про те, чи прихований ваш справжній IP. Це популярно для використання при тестуванні моделі сірого вікна, коли ви домовились із замовником виконати тестування на проникнення зі статичної

IP-адреси, але такої немає. Однак, якщо вам потрібно імітувати атаку з певної країни або якщо вам потрібно приховати реальну IP-адресу під час тестування на проникнення в чорну скриньку, настав час також використовувати VPN.

Плюси:

- Краще шифрування.
- Більш надійний.
- Більше можливостей.
- Простіше у використанні з величезним набором інструментів.
- Може використовуватися як “подвійна VPN” або навіть VPN всередині VPN.

Мінуси:

- У вас є 1 IP на все
- Важко змінити IP-адреси під час використання VPN
- Потрібно більше часу для підключення
- У більшості випадків для автоматичного переключення між VPN потрібно більше часу та зусиль

TOP

Tor (мережа Tor) – це дуже популярна на сьогоднішній день технологія анонімізації, яка допомагає вам не тільки приховати свою справжню IP-адресу, але й отримати доступ до мережі Tor. У вас є два варіанти, як його запустити:

1. Ви можете просто встановити браузер Tor (від Mozilla).
2. Ви можете запустити свою службу Tor на віддаленому сервері, який буде використовуватися як проксі.

За допомогою Tor ви можете легко перемикати IP-адреси за допомогою певної опції у вашому браузері або просто перезапустивши службу Tor на своєму сервері. Кожного разу ви отримаєте новий IP-вузол виходу. Ваш справжній IP буде прихований, а з’єднання зашифровано, але тут ми стикаємось з іншою проблемою. Як уже згадувалося вище, ви

підключитеся величезним ланцюгом до вихідного вузла, який має можливість бачити ваш трафік в чистому режимі.

До речі, за допомогою Tor і проксі-ланцюгів ви навіть можете використовувати інструменти сканування на рівні мережі, такі як nmap , zmap, hping та інші. Підхід заснований на використанні служби Tor та запуску проксі-ланцюгів перед використанням будь-якого інструменту чи сценарію. Команда для nmap для використання з проксі-ланцюгами через Tor виглядає так: проксі-ланцюжки nmap –sV 192.168.1.1.

Звичайно, це додає додаткові ризики для нашої діяльності щодо тестування на проникнення через можливість витоку інформації. Тож нам слід врахувати всі ці ризики, перш ніж ми почнемо використовувати Tor для наших проектів пентестування. Цей підхід в основному використовується при тестуванні на проникнення в чорну скриньку, коли нам потрібно приховати справжній IP або уникнути брандмауерів.

Плюси:

- Простий у використанні.
- Безкоштовно.
- Ви можете швидко змінити IP-адреси та розташування.

Мінуси:

- Хтось із вузла виходу може контролювати ваш трафік.
- Ви не можете використовувати Tor так само, як проксі; це ближче до методу VPN.
- Він працює повільніше, оскільки вихідний вузол зазвичай вибирається за замовчуванням у далекій від вас країні.

2.8 Висновок до другого розділу

Якщо говорити про інструменти, автоматизацію та потокову роботу, нам слід наближатись до методу з використанням проксі-сервера, оскільки

за допомогою такого методу ми можемо багато отримати на швидкості та можливості IP-комутації.

Якщо ми говоримо про переважно ручний процес і нам потрібне більш безпечне та стабільне з’єднання, тоді VPN задовольнить наші потреби. Хоча це забезпечить вам кращі швидкості, а також стабільне та зашифроване з'єднання, за замовчуванням він також спрямовуватиме весь ваш трафік через сервер, а не лише через певний порт. Величезним недоліком є те, що якщо нам потрібно багато IP-адрес, ми повинні прив’язати всі їх до нашого одного сервера і переключатися між ними або призначати конкретний IP конкретному користувачеві за NAT.

Нарешті, якщо у нас немає можливостей або грошей для запуску нашого власного сервера для проксі-сервера або VPN, тоді ми можемо спробувати використовувати Tor. Це рішення є менш ефективнішими через втрату швидкості та можливості витоку інформації (додаткові ризики), але принаймні воно може допомогти нам переключити наш IP і дає можливість обійти блокування з сторони брандмауера.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

3.1 Безпека життєдіяльності

3.1.1 Соціальні небезпеки

В основу визначення соціальних небезпек, спричинених низьким духовним рівнем, покладено цінності і компоненти суспільства та людини.

Існують дві ціннісні компоненти, співвідношення між якими характеризує стан суспільного життя.

Перша ціннісна компонента – цінності культури суспільства. Друга ціннісна компонента – ціннісна орієнтація особистості. Зв'язок між цими двома крайніми компонентами культури – найважливіший цементуючий і стимулюючий початок всього суспільного життя. І навпаки – порушення цього зв'язку визначає глибоку духовну кризу. В сучасному суспільстві у поєднанні будь-яких складових суспільного життя і в усвідомленні його цілісності велика роль належить інтелігенції. Вона є духовним й інтелектуальним посередником у системі суспільних зв'язків. Але інтелігенція може впоратися з цією роллю за умов не порушення її світорозуміння. Однією з особливостей сучасної духовної кризи інтелігенції є різкий поворот від атеїстичного світосприйняття до релігійного. І тому криза в світоглядній орієнтації інтелігенції теж є небезпекою соціального стану суспільства.

На тлі змінених орієнтирів суспільство потерпає від соціальних небезпек, які спричиняли зміни і втрати загальнолюдських цінностей і орієнтацій значної кількості населення.

Результатом зміни світу цінностей і орієнтирів частини суспільства є бродяжництво, проституція, пияцтво, алкоголізм, паління, наркоманія, СНІД.

Зазначені соціальні небезпеки формують в людському середовищі «групи ризику». «Групи ризику» впливають на стан суспільства підвищенням чисельності кримінальних злочинів, втягуванням в свої лави нових представників здорової частини суспільства, підриванням здоров'я людей, що

їх оточують, погіршують генофонд нації. Вищезгадані негативні явища в суспільстві (бродяжництво, проституція та ін.) створюють негативне коло, причини якого в більшості випадків пов'язані між собою.

У першу чергу слід відзначити наявність ряду моральних чинників, які поділяють суспільство на працівників комерційних структур, спільних підприємств та працівників державного сектора економіки. Неспіввідношеність платні в обставинах низьких заробітків і нерегулярних виплат спричиняють незадоволення роботою у працівників державного сектора.

Загостреність обставин зумовлюється також незадовільним станом умов праці, проживання і побуту. Вся сукупність обставин збільшує ступінь соціального напруження.

Це може стати передумовою виникнення страйку, а за розвитком відповідних негативних явищ і умов дійти аж до повстання чи перевороту.

Такий розвиток подій завжди, навіть в ретельно організованих суспільних змінах, супроводжується виникненням стихійних угруповань, які здійснюють свої плани наживи за рахунок мародерства і вандалізму серед населення, що згодом закінчується тяжкими наслідками актів тероризму.

Необхідність усвідомлених знань розвитку соціальних небезпечних чинників пов'язана з розумінням – куди може привести сукупний їх розвиток і яких негараздів може зазнати населення від необачливих дій стихійного характеру.

Найбільше, від чого може потерпати людина, – це від помилкового розуміння свого стану в суспільстві. Щоб орієнтуватися в цьому світі, щоб його розуміти, людина повинна визначити, які її дії матимуть підтримку і схвалення, а які, навпаки, спричинять недовіру і непорозуміння з боку суспільства. Визначена для себе система спілкування з суспільством складає передумови подальшого прогнозу під час взаємодій з суспільством. Це вже здійснюється за набутим досвідом і становить систему самооцінки людиною

своїх дій. Розуміння критеріїв самооцінки й оцінки вчинків інших людей дається завдяки усвідомленню ціннісного і нормативного змісту культури.

Найбільш суттєва характеристика системи цінностей складається з того, що саме тут сконцентровані уявлення людей про смисл їх життя. Неадекватність системи оцінок, що застосовує людина, призводить до конфліктної ситуації її з суспільством. Неадекватність оцінок завжди зумовлює відповідну реакцію з боку людини – морального та психологічного походження. Накопичення таких реакцій у людини сприяє відчуттю відповідного дискомфорту і непрогнозованій поведінці.

3.1.2 Наслідки забруднення навколишнього середовища

Унаслідок господарської діяльності людини у природному середовищі нагромаджуються не властиві йому речовини. Серед них тверді відходи (сміття) та хімічні сполуки, які призводять до забруднення довкілля. Забрудненою може бути невелика територія, зазвичай навколо промислового підприємства або населеного пункту. Якщо ж забруднення охоплює всю планету і виявляється у будь-якій точці Землі навіть на значній відстані від джерела забруднення, то говорять про глобальне забруднення.

Основними джерелами забруднення повітря є промислові підприємства (заводи, фабрики, теплові електростанції) і транспорт. Спалюючи паливо або виробляючи продукцію, вони викидають в атмосферу пил, сажу, різні хімічні сполуки. Забруднене повітря стає загрозою для всього живого. Воно подразнює очі, ніс та горло людини, викликає отруєння, вбиває рослини. Забруднене повітря охоплює великі райони і безперешкодно надходить у різні країни, залежно від напрямку вітрів. Наслідками забруднення атмосферного повітря стали такі глобальні проблеми, як потепління клімату, випадання кислотних дощів і утворення озонових дір.

Про глобальне потепління клімату нині часто йдеться на сторінках газет і в теленовинах. Викликане воно збільшенням кількості вуглекислого газу і пилу в атмосфері. Забруднене повітря перешкоджає випромінюванню

тепла від Землі назад у космічний простір. Тепло нагромаджується і викликає порушення звичних кліматичних умов. Підвищення температури повітря на Землі навіть на 1 ^{0}С, призведе до танення криги в Арктиці і Антарктиці. Невдовзі підвищиться рівень Світового океану. Внаслідок підняття рівня води в ньому навіть на 1 м будуть затоплені густонаселені прибережні низовини материків, а мільйони людей – позбавлені своїх місць проживання. Через потепління клімату почастішали стихійні лиха: урагани, смерчі, зливи, повені.

У забрудненому повітрі з'явився новий вид атмосферних опадів – кислотні дощі. Вони виникають внаслідок сполучення з атмосферною вологою хімічних речовин, що викидаються з труб заводів і ТЕС. Кислотні дощі спричиняють захворювання людей, всихання лісів, отруєння води озер і ставків, унаслідок чого в них гине риба. З цієї причини, наприклад, за останні 50 років зникла форель з численних озер Норвегії і Швеції.

Гази, що потрапляють у повітря, руйнують озоновий шар стратосфери. В результаті виникають озонові діри. Найбільша з них виявлена над Антарктидою. Зменшення концентрації озону призводить до збільшення кількості згубного для всього живого ультрафіолетового випромінювання, яке надходить на Землю.

Щоб запобігти забрудненню повітря і екологічним проблемам, що з цим пов'язані, необхідно встановлювати очисні споруди на промислових підприємствах, які б зменшили викиди хімічних речовин.

Основними джерелами забруднення води є промислові підприємства, які скидають у річки і водойми неочищені стоки. Вода забруднюється також змитими з полів мінеральними добривами та отрутохімікатами. До них додаються побутові стоки. Забруднена вода стає непридатною для пиття, купання та поливу рослин. Небезпечно забрудненими є багато річок та озер на різних материках. "Стічними канавами" Європи називають річки Рейн і Дунай. До них можна віднести й наш Дніпро.

Забруднену воду річки несуть у моря й океани. Там додається ще й забруднення нафтою внаслідок аварій трубопроводів і танкерів, що її

транспортують. Нафта утворює на поверхні води нафтову плівку, яка перешкоджає надходженню кисню, а тому небезпечна для всього живого. Нині значна частина поверхні Світового океану вкрита нафтовими плямами і перетворена на безжиттєву пустелю. Сильно забруднені Північне і Балтійське моря, Мексиканська і Перська затоки.

Основними заходами боротьби із забрудненням води є очищення стоків, що надходять у водойми. Велике значення має запровадження сучасних безстічних технологій у промисловому виробництві.

Практично всі забруднюючі речовини, що спочатку потрапили в повітря, згодом опиняються у ґрунтах. Крім того, ґрунти забруднюються і при надмірному внесенні в них отрутохімікатів та мінеральних добрив. Самоочищення землі відбувається дуже повільно, тому отруйні хімічні речовини нагромаджуються там. Їх поглинають рослини, вживання яких викликає захворювання людей і тварин. Щоб запобігти забрудненню ґрунтів, мінеральні добрива потрібно вносити дуже обережно, у науково обґрунтованій кількості.

Забруднення місцевості радіоактивними речовинами може виникнути внаслідок аварії на атомній електростанції. Радіоактивні речовини спричиняють зовнішнє і внутрішнє опромінення людини, що викликає надзвичайно небезпечні для життя захворювання. Унаслідок аварії на Чорнобильській АЕС в Україні у 1986 р. радіоактивно забрудненими виявилися території багатьох областей.

Райони з катастрофічним забрудненням природного середовища називають районами екологічного лиха. Такими, наприклад, є південні райони Африки, надмірно забруднені викидами численних ТЕС і промислових підприємств. В Євразії районами екологічним лиха є Аральське море і Перська затока, в Україні – 30-кіломерова зона навколо Чорнобильської АЕС, частини Донецької і Луганської областей.

3.2 Основи охорони праці

3.2.1 Техніка безпеки при роботі з ПК

До самостійної роботи на комп’ютерах допускаються особи, які пройшли медичний огляд, навчання по професії, вступний інструктаж з охорони праці та первинний інструктаж з охорони праці на робочому місці. В подальшому вони проходять повторні інструктажі з охорони праці на робочому місці один раз на півріччя, періодичні медичні огляди один раз на два роки.

Під час роботи на комп’ютерах можуть діяти такі небезпечні та шкідливі фактори, як:

– фізичні;

– психофізіологічні.

Основним обладнанням робочого місця користувача комп’ютера є монітор, системний блок та клавіатура, мишка.

Робочі місця мають бути розташовані на відстані не менше 1,5 м від стіни з вікнами, від інших стін на відстані 1м, між собою на відстані не менше 1,5 м. Відносно вікон робоче місце доцільно розташовувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

Робочі місця слід розташовувати так, щоб уникнути попадання в очі прямого світла. Джерела освітлення рекомендується розташовувати з обох боків екрану паралельно напрямку погляду. Для уникнення світлових відблисків екрану, клавіатури в напрямку очей користувача, від світильників загального освітлення або сонячних променів, необхідно використовувати анти полискові сітки, спеціальні фільтри для екранів, захисні козирки, на вікнах – жалюзі.

Монітор повинен бути розташований на робочому місці так, щоб поверхня екрана знаходилася в центрі поля зору на відстані 400-700 мм від очей користувача. Рекомендується розміщувати елементи робочого місця так, щоб витримувалася однакова відстань очей від екрана, клавіатури, тексту.

Зручна робоча поза при роботі з комп'ютером забезпечується регулюванням висоти робочого столу, крісла та підставки для ніг. Раціональною робочою позою може вважатися таке положення, при якому ступні працівника розташовані горизонтально на підлозі або підставці для ніг, стегна зорієнтовані у горизонтальній площині, верхні частини рук – вертикальні. Кут ліктєвого суглоба коливається в межах 70-90°, зап'ястя зігнуті під кутом не більше ніж 20°, нахил голови 15-20°.

Для нейтралізації зарядів статичної електрики в приміщенні, де виконується робота на комп'ютерах, в тому числі на лазерних та світлодіодних принтерах, рекомендується збільшувати вологість повітря за допомогою кімнатних зволожувачів. Не рекомендується носити одяг з синтетичних матеріалів.

Згідно статті 18 Закону України "Про охорону праці" працівник зобов'язаний:

– знати і виконувати вимоги нормативних актів про охорону праці, правила поводження з устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту;

– дотримуватись зобов'язань щодо охорони праці, передбачених колективним договором та правилами внутрішнього трудового розпорядку підприємства;

– співробітничати з власником у справі організації безпечних і нешкідливих умов праці, особисто вживати посильних заходів щодо усунення будь-якої виробничої ситуації, яка створює загрозу його життю чи здоров'ю, або людей, які його оточують, повідомляти про небезпеку свого безпосереднього керівника або іншу посадову особу.

Вимоги безпеки перед початком роботи:

– увімкнути систему кондиціювання в приміщенні;

– перевірити надійність встановлення апаратури на робочому столі. Повернути монітор так, щоб було зручно дивитися на екран – під прямим кутом (а не збоку) і трохи зверху вниз, при цьому екран має бути трохи

нахиленим, нижній його край ближче до оператора;

– перевірити загальний стан апаратури, перевірити справність електропроводки, з'єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана;

– відрегулювати освітленість робочого місця;

– відрегулювати та зафіксувати висоту крісла, зручний для користувача нахил його спинки;

– приєднати до системного блоку необхідну апаратуру. Усі кабелі, що з'єднують системний блок з іншими пристроями, слід вставляти та виймати при вимкненому комп'ютері;

– ввімкнути апаратуру комп'ютера вимикачами на корпусах в послідовності: монітор, системний блок, принтер (якщо передбачається друкування);

– відрегулювати яскравість свічення монітора, мінімальний розмір світної точки, фокусування, контрастність. Не слід робити зображення надто яскравим, щоб не втомлювати очей.

Рекомендується:

– яскравість свічення екрана – не менше 100Kg/м^2;

– відношення яскравості монітора до яскравості оточуючих його поверхонь в робочій зоні – не більше 3:1;

– мінімальний розмір точки свічення не більше 0,4 мм для монохромного монітора і не менше 0,6 мм для кольорового, контрастність зображення знаку – не менше 0,8.

При виявленні будь-яких несправностей роботу не розпочинати, повідомити про це керівника.

Вимоги безпеки під час виконання роботи:

– необхідно стійко розташовувати клавіатуру на робочому столі, не опускати її хитання. Під час роботи на клавіатурі сидіти прямо, не напружуватися;

– для забезпечення несприятливого впливу на користувача пристроїв типу “миша” належить забезпечувати вільну велику поверхню столу для переміщення “миші” і зручного упору ліктьового суглоба;

– не дозволяються сторонні розмови, подразнюючі шуми;

– періодично при вимкненому комп’ютері прибирати ледь змоченою мильним розчином бавовняною ганчіркою порох з поверхонь апаратури. Екран протирають ганчіркою, змоченою у спирті. Не дозволяється використовувати рідинні або аерозольні засоби чищення поверхонь комп’ютера.

3.2.2 Інструкція з охорони праці при експлуатації ЕОМ

Ця інструкція розроблена на основі Державних санітарних правил і норм роботи з візуальними дисплейними терміналами ЕОМ (СНіП 3.3.2 007 1998) і Правил охорони праці під час експлуатації ЕОМ, затверджених наказом Держнаглядохоронпраці від 10.02.1999 р. №21.

До самостійної роботи по експлуатації, монтажу, налагодженню і перевірці засобів обчислювальної техніки допускаються особи, що пройшли необхідне теоретичне навчання, вступний та первинний інструктаж і перевірку знань.

Вимоги до профілактичних медоглядів:

– усі працівники, які виконують роботи, пов'язані з експлуатацією ЕОМ, підлягають обов'язковим медичним оглядам: попередньому – під час оформлення на роботу і періодичному – один раз на два роки комісією в складі терапевта, невропатолога та офтальмолога. За наявності медичних показань в комісію по оглядах можуть залучатися лікарі інших спеціальностей (в т.ч. гінеколог – для жінок);

– до роботи безпосередньо на ЕОМ допускаються особи, які не мають медичних протипоказань, обумовлених санітарними нормами;

– жінки з часу встановлення вагітності та в період годування дитини до виконання всіх робіт, пов'язаних з використанням ЕОМ, не допускаються.

Вимоги до виробничих приміщень:

– площу приміщень, в яких розташовують відеотермінали, визначають згідно нормативних документів, з розрахунку на одне робоче місце: площа – не менше 6.0 $м^2$, об'єм – не менше 20,0 $м^3$ з урахуванням максимальної кількості осіб, які одночасно працюють у зміні;

– стіни приміщень для роботи з ЕОМ мають бути пофарбовані чи обклеєні шпалерами пастельних кольорів з коефіцієнтом відбиття 40-60%;

– підлога всієї зони обслуговування, ремонту та налагодження ЕОМ має бути вкрита діелектричними килимками або ізоляційними підстилками для ніг;

– у приміщеннях для роботи необхідно проводити щоденне вологе прибираний та регулярне провітрювання протягом робочого дня. Видалення пилу з екрану необхідно проводити не рідше одного разу на день;

– приміщення з ЕОМ повинні бути оснащені переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 $м^2$ площі приміщення;

– підходи до засобів пожежегасіння повинні бути вільними;

– у приміщеннях з ЕОМ повинні бути медичні аптечки першої допомоги.

Вимоги до освітлення:

– приміщення з ЕОМ повинні мати природне і штучне освітлення відповідно до СНіП П-4-79 “Естественное, искусетвенное освещение”;

– природне світло повинно проникати через бічні світлопрорізи, зорієнтовані, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості не нижче 1,5%;

– вікна приміщень з ЕОМ повинні мати регулювальні пристрої для відкривання, а також жалюзі, штори, зовнішні козирки;

– у виробничих та адміністративно-громадських приміщеннях, де переважають роботи з документами, допускається використовувати систему

комбінованого освітлення (додатково до загального освітлення встановлюють світильники місцевого освітлення);

– для освітлення приміщень з ЕОМ необхідно використовувати люмінесцентні світильники. Освітленість робочих місць у горизонтальній площині на висоті 0,8м від підлоги повинна бути не менше 400 лк. Вертикальна освітленість у площині екрану – не більше 300 лк.;

– необхідно обмежувати прямий блиск від джерел природнього та штучного освітлення. При цьому яскравість поверхонь, що світяться (вікна, джерела штучного освітлення) і перебувають в полі зору, повинна бути не більшою за 200 кд/м^2.

Приміщення з ЕОМ повинні бути обладнані системами опалення, кондиціювання повітря або припливно-витяжною вентиляцією. У приміщеннях, де одночасно експлуатується більше п'яти ЕОМ, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

3.3 Висновок до третього розділу

В даному розділі розглянуто такі питання: соціальні небезпеки, наслідки забруднення навколишнього середовища, техніка безпеки при роботі з ПК та інструкція з охорони праці при експлуатації ЕОМ

ВИСНОВКИ

При виконанні кваліфікаційної роботи «Бакалавр» були виконані поставленні завдання, які дали змогу зробити короткий підсумок виконаної роботи.

Архітектури розподілених систем усувають основні недоліки централізованих систем, такі як висока чутливість до збоїв, відсутність масштабованості. Однак розподіленим системам властиві такі недоліки, як складність та накладні витрати на зв'язок через відсутність спільної пам'яті. Цей звіт також охоплює основні три типи розподілених систем клієнт-серверні системи, інтегровані системи управління файлами та інтегровані розподілені системи.

Спілкування між різними обчислювальними хостами в мережі є важливим аспектом при вивченні розподілених систем. Механізми зв'язку в сучасних розподілених системах можна розділити на три великі категорії: передача повідомлень, механізм порту та віддалені виклики процедур.

Архітектура розподіленої операційної системи під назвою 2K пояснюється в підрозділі «Розподілені операційні системи». Ця операційна система вирішує основні проблеми розподілених систем, такі як управління ресурсами, динамічна адаптованість та конфігурація розподілених додатків на основі компонентів.

Також пояснюється різниця між розподіленими базами даних та сховищами даних з прикладами, а також такими характеристиками, як обробка транзакцій, узгодженість та стійкість до відмов, продуктивність, кешування та балансування робочого навантаження.

Якщо говорити про інструменти, автоматизацію та потокову роботу, нам слід наближатись до методу з використанням проксі-сервера, оскільки за допомогою такого методу ми можемо багато отримати на швидкості та можливості IP-комутації.

Якщо ми говоримо про переважно ручний процес і нам потрібне більш безпечне та стабільне з’єднання, тоді VPN задовольнить наші потреби. Хоча це забезпечить вам кращі швидкості, а також стабільне та зашифроване з'єднання, за замовчуванням він також спрямовуватиме весь ваш трафік через сервер, а не лише через певний порт. Величезним недоліком є те, що якщо нам потрібно багато IP-адрес, ми повинні прив’язати всі їх до нашого одного сервера і переключатися між ними або призначати конкретний IP конкретному користувачеві за NAT.

Нарешті, якщо у нас немає можливостей або грошей для запуску нашого власного сервера для проксі-сервера або VPN, тоді ми можемо спробувати використовувати Tor. Це рішення є менш ефективнішими через втрату швидкості та можливості витоку інформації (додаткові ризики), але принаймні воно може допомогти нам переключити наш IP і дає можливість обійти блокування з сторони брандмауера.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. Banatre, “Hiding distribution in distributed systems,” [1991 Proceedings] 13th International Conference on Software Engineering, Austin, TX, 1991, pp. 189-196.

2. A.D. Birrell and B.J. Nelson, “Implementing Remote Procedure Call,” in *ACM Transactions on Computer Systems*, 1984, pp. 39-59.

3. R. M. Wong, “Issues in secure distributed operating system design,” Digest of Papers. COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, San Francisco, CA, USA, 1989, pp. 338-341.

A. Tevanian and R.F. Rashid, *MACH A Basis for Future UNIX Development. Technical Report CMU-CS-87-139*, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, June 1987.

4. N. Bronson et al, “TAO: Facebook’s Distributed Data Store for the Social Graph,” in USENIX Annu. Technical Conf., 2013, pp. 49-60.

5. J. C. Corbett et al, “Spanner: Google’s Globally-Distributed Database,” in the *Proceedings of OSDI.*, 2012, pp. 1-14.

6. P. M. Melliar-Smith, L. E. Moser and V. Agrawala, “Broadcast protocols for distributed systems,” *in IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 1, pp. 17-25, Jan 1990.doi: 10.1109/71.80121

7. S. J. Mullender, A. S. Tanenbaum, “The Design of a Capability-Based Distributed Operating System”. Comput J 1986; 29 (4): 289-299. doi: 10.1093/comjnl/29.4.289

8. David R. Cheriton, Willy Zwaenepoel, “The Distributed V Kernel and its Performance for Diskless Workstations” in Proc. 9th ACM Symposium on Operating Systems Principles, 1983, pp. 129-140

A. Tanenbaum ct R. Van Re-. Distributcd Operating Systcms. ACM Computing Surveys, 17(4):419-470, Dccembcr 1985.

9. PJ. Leach, P.H. Lcvinc, B.P. Douros, J.A.Hamilton, DL. Nelson, et B.L. Stumpf. Thc architecture of an Integrated Local Network. IEEE Journal on Sclectcd Areas in Communications, 1(5):842–857, November 1983.

10. F. Kon, R. H. Campbell, M. D. Mickunas, K. Nahrstedt and F. J. Ballesteros, "2K: a distributed operating system for dynamic heterogeneous environments," Proceedings the Ninth International Symposium on High-Performance Distributed Computing, Pittsburgh, PA, 2000, pp. 201-208.

11. Dynamic Agent-based Security Architecture for Mobile Computers. In *Proceedings of The Second International Conference on Parallel and Distributed Computing and Networks (PDCN'98)***,** pages 29 1-299, Australia, December 1998.