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

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

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розробка модуля двофакторної аутентифікації для Unix подібних операційних систем

Виконав(ла): студент(ка) IV курсу, групи СБс 42

спеціальності 125 Кібербезпека
(шифр і назва спеціальності)

(підпис) Марко Р.Б. (прізвище та ініціали)

Керівник (підпис) Гладьо Ю.Б. (прізвище та ініціали)

Нормоконтроль (підпис) (прізвище та ініціали)

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

Рецензент (підпис) (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 кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В
(підпис) (прізвище та ініціали)
«22 » червня 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня БАКАЛАВР
(НАЗВА ОСВІТНЬОГО СТУПЕНЯ)

за спеціальністю 125 Кібербезпеа
(шифр і назва спеціальності)

студенту Марку Роману Богдановичу
(ПРІЗВИЩЕ, ІМ'Я, ПО БАТЬКОВІ)

1. Тема роботи Розробка модуля двофакторної аутентифікації для Unix подібних операційних систем

Керівник роботи Гладьо Ю.Б., к.т.н., доц. каф. ММ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «__» _____ 2021 року №_____

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи технічна документація, інтернет-джерела.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз Методів Аутентифікації. 1.1 Передумови створення аутентифікації як механізму контролю доступу. 1.2. Види та способи аутентифікація. 1.3 Двофакторна аутентифікація. 1.5 Висновок до розділу. 2. Аналіз процесу аутентифікації в ОС Linux. 2.1. Основні компоненти системи аутентифікації. 2.2 Конфігураційні файли. 2.3 Висновок до розділу. 3. Розробка Модуля Аутентифікації. 3.1 Вибір способу аутентифікації. 3.2 Реалізація. 3.3 Інтеграція та тестування модуля. 3.4 Висновок до розділу. 4. Безпека Життєдіяльності, Основи Охорони праці. 4.1 Роль центральної нервової системи в трудовій діяльності людини. 4.2 Долікарська допомога при пораненнях. 4.3 Висновок до розділу. Висновок.

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

1. Назва Роботу. 2. Вступ. 3. Види аутентифікації. 4. Способи аутентифікації. 5. Компоненти PAM системи. 6. Типи модулів в PAM системі. 7. Алгоритм модуля двофакторної аутентифікації. 8.Дані які отримує скрипт від PAM системи. 9. Висновок. 10. Кінець.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці			

7. Дата видачі завдання 16.02.2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	**Примітка**
1.	Ознайомлення з завданням до кваліфікаційної роботи	16.02 – 19.02	*Виконано*
2.	Підбір джерел про види аутентифікації	16.02 – 19.02	*Виконано*
3.	Опрацювання джерел про види аутентифікації	19.02 – 02.03	*Виконано*
4.	Підбір джерел про системи аутентифікації	02.03 – 10.03	*Виконано*
5.	Опрацювання джерел про системи аутентифікації	10.03 – 16.03	*Виконано*
6.	Дослідження видів та способів аутентифікації	16.03 – 01.04	*Виконано*
7.	Дослідження систем аутентифікації в Unix подібних ОС	01.04 – 10.04	*Виконано*
8.	Розробка та реалізація модуля аутентифікації	10.04 – 16.04	*Виконано*
9.	Оформлення розділу «Аналіз методів аутентифікації»	16.04 – 25.04	*Виконано*
10.	Оформлення розділу «Аніліз процесу аутентифікії в ОС Linux»	25.04 – 05.05	*Виконано*
11.	Оформлення розділу «Розробка модуля аутентифікації»	05.05 – 16.05	*Виконано*
12.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці»	16.05 – 22.05	*Виконано*
13.	Оформлення кваліфікаційної роботи	22.05 – 08.06	*Виконано*
14.	Нормоконтроль	08.06 – 10.06	*Виконано*
15.	Перевірка на плагіат	10.06 – 16.06	*Виконано*
16.	Попередній захист кваліфікаційної роботи	16.06 – 19.06	*Виконано*
17.	Захист кваліфікаційної роботи	24.06	

Студент _____ Марко Р.Б.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Гладьо Ю.Б.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка модуля двофакторної аутентифікації для Unix подібних операційних систем // Марко Роман Богданович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра кібербезпеки, група СБс–42 // Тернопіль, 2021 // С.–50, рис.–10. Табл.–2, слайдів–10, бібліогр.–8.

Ключові слова: АУТЕНТИФІКАЦІЯ, СИСТЕМА АУТЕНТИФІКАЦІЇ, МОДУЛІ АУТЕНТИФІКАЦІЇ, ДВОФАКТОРНА АУТЕНТИФІКАЦІЯ

Кваліфікаційна робота присвячена аналізу системи аутентифікації Unix подібних операційних систем, та реалізація модуля двофакторної аутентифікації. В роботі проаналізовано існуючи види та способи аутентифікації, досліджено та описано їх переваги та недоліки. Також в роботі описано принцип роботи системи аутентифікації на прикладі операційної системи Linux.

Було розроблено модуль двофакторної аутентифікації, який здатний інтегруватися в систему аутентифікації. Розроблений модуль було перевірено та успішно інтегровано в систему аутентифікації Linux. Крім цього було запропоновано ряд рекомендацій щодо використання розробленого модуля, також було описано способи покращення модуля в майбутньому.

ANNOTATION

Two-factor authentification module development for Unix-like operation systems // Marko Roman // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cyber Security // Ternopil, 2021 // P.-50, Fig.-10, Table-2, Slides-10, References-8.

Keywords: AUTHENTICATION, AUTHENTICATION SYSTEM, AUTHENTICATION MODULES, TWO-FACTOR AUTHENTICATION

Qualification work on the analysis of the Unix authentication system of similar operating systems and the implementation of the two-factor authentication module. The paper analyzes the existing sees and methods of authentication, explores and describes their advantages and disadvantages.

The paper also describes the principle of authentication systems on the example of the Linux operating system. A two-factor authentication module has been developed that can be integrated into the authentication system. The developed module was tested and successfully integrated into the Linux authentication system. In addition, a number of recommendations for the use of the developed module were offered, as well as ways to improve the module in the future.

ЗМІСТ

СПИСОК УМОВНИХ СКОРОЧЕНЬ

ADB	(Android Debug Bridge) протокол, та засіб налагоджування
ОС	Операційна Система
ПЗ	Програмне Забезпечення
PAM	(Plugin Authenticate Module) Розширюваний модуль Аутентифікація
SS7	Signalling System No. 7, це набір сигнальних телефонних протоколів, використовуваних для роботи більшості телефонних станцій

ВСТУП

В сучасному житті комп’ютери та комп’ютерні системи використовуються для великого спектру задач у різних галузях людського життя. Ми все частіше використовуємо їх для проведення юридичних, та фінансових операцій. На наших персональних комп’ютерах зберігається важлива інформація, компрометація якої може завдати шкоди як нам так і іншим. Тому однією з важливих задач є обмеження доступу до інформації стороннім особам.

Одним з механізмів обмеження доступу є механізм аутентифікації. Його роль у системі — перевірка належності користувача в системі. Цей механізм є надзвичайно важливим, адже несправності в його роботі можуть дозволити зловмиснику отримати доступ в систему, що в свою чергу може призвести до розкриття важливих даних, їх модифікації, або втрати.

На даний момент існує багато різноманітних методів аутентифікації для різних ОС. Якщо на мобільних пристроях використовується, системи на основі відбитку пальців, або розпізнавання обличчя, що надає досить високу ступінь захисту, так як пройти таку аутентифікацію зловмиснику набагато важче, ніж на ПК, на яких зазвичай використовується аутентифікація по паролю.

Основним методом аутентифікації на ПК вважається аутентифікація на основі парольної фрази. Що є досить поганим методом, зважаючи що людина схильна використовувати паролі, які легко запам’ятати, або ж використовує один і той самий пароль для багатьох сервісів. В такому випадку зловмиснику достатньо отримати один пароль. Щоб мати доступ до багатьох сервісів жертви.

Щоб запобігти вище згаданим проблемам, можна використовувати двофакторну аутентифікацію. Що дозволяє збільшити рівень захисту системи, так як для успішного входу в систему потрібно володіти двома факторами. Володіння лише одним з двох факторів не зможе забезпечити доступ в систему.

Дана роботу зосереджена саме на проблемах в процесах аутентифікації, та їх рішенні для звичайних користувачів. Тому що в ІКС (Інформаційно комп’ютерних системах) процесу аутентифікації приділяють увагу, ще при розробці такий систем, та створюють рішення під конкретні задачі.

Об’єктом дослідження в даній роботі є система аутентифікації в Unix подібних операційних система. Хоча доля ринку таких ОС серед користувачів є дуже маліє, та на даний момент ОС цього сімейства активно розвиваються, і в майбутньому можуть замінити ОС сімейства Windows в деяких сферах.

Предметом дослідження є система аутентифікації в Unix подібних ОС.

Метою роботи є розробка модуля двофакторної аутентифікації.

Для виконання поставленої мети було розроблено список завдання які потрібно виконати щоб досягнуть мети дипломної робити.

Завдання дипломної роботи:

— аналіз методів аутентифікації;
— огляд існуючих способів аутентифікації, їх переваги та недоліки;
— огляд основним методів аутентифікації;
— аналіз механізмів аутентифікації в Unix подібні ОС
— аналіз способів розробки модуля аутентифікації;
— розробка модуля аутентифікації, його тестування та інтегрування в систему аутентифікації.

1 АНАЛІЗ МЕТОДІВ АУТЕНТИФІКАЦІЇ

1.1 Передумови створення аутентифікації як механізму контролю доступу

На початку, коли люди тільки зрозуміли потенціал комп’ютерів, комп’ютери використовувалися одним користувачем за раз. Це означає що всі ресурси використовував тільки один користувач. В ті часи не було потреби розділювати повноваження користувачів між собою, ще означає що не було потреби в ідентифікації, аутентифікації та в авторизація, так як не було ресурсів, доступ до яких потрібно обмежувати.

Однак після появи ОС, а саме багатозадачних ОС. З’явилася потреба обмежити доступ до ресурсів інших задач та користувачів. В ці часи почали з’являтися і комп’ютерні мережі, в який комп’ютери можуть взаємодіють між собою, тому потрібно було обмежити вплив одного комп’ютера на інші.

Для того щоб обмежити доступ до комп’ютерів, але водночас мати можливість виконувати певні дії, з’явилася потреба керувати правами користувачів та контролю над ними.

Саме тоді з’явилися такі терміни як ідентифікація, аутентифікація, та авторизація. Потрібно чітко розуміти значення цих термінів, щоб не плутати їх функції та призначення в комп’ютерах та комп’ютерних системах.

Ідентифікація — процес отримання даних (відомостей), що чітко визначають об’єкт.

Прикладами ідентифікації є:

— ім’я користувача, цей спосіб ідентифікація часто використовують в комп’ютерних системах, та персональних комп’ютерах для доступу до системи;

— номер рахунку, дуже часто використовується для отримання доступу до банківських рахунків.

Аутентифікація — процес підтвердження того, що ідентифікованих об’єкт, дійсно є тим за кого себе видає.

Основними способами аутентифікації є:

— аутентифікація по паролю, часто використовується для підтвердження ідентифікованого об’єкта;
— аутентифікація на основі відомостей якими володіє об’єкт, наприклад для доступу банківського рахунку може використовуватися інформація що є розміщення на банківських картках. Наприклад дата доки дійсно картка, CSV код, звичайно в банківських системах цього не достатньо для підтвердження, тому використовуються інші методи в зв’язці з цими.

Авторизація — процес підтвердження права на виконання певної дії, або групи дій.

Прикладами роботи механізму авторизації є:

— файлова система з правами доступу до файлів. У такій файловій системі у кожного файлу є власник, та є список прав які можуть робити інші групи користувачів. В такій файловій системі механізм авторизація забезпечує драйвер файлової системи, який слідкує за кожною дією, та перевіряє чи може певний об’єкт виконати операції над файлом;
— операційна система, кожна сучасна ОС є багатокористувацькою. Що змушує мати хорошу та надійну системи прав доступу до різних ресурсів. Як приклад можна привести ситуацію коли декілька користувачів намагаються отримати доступ до одного ресурсу, в такі ситуацій ОС зобов’язана контролювати та обмежувати деякі дії користувачів, щоб запобігти пошкодженню об’єктів, або їх знищенню.

Серед ідентифікації аутентифікації так авторизація, я б відзначив що аутентифікація є найважливішим з них. Тому що ідентифікація на даних час процес який можна легко подолати в більшості систем. Наприклад:

— в банківській системі ідентифікація відбувається по номеру рахунку, а сам номер рахунку є відомий тому, що використовується для банківських переведень;
— в більшості ПК в ОС з графічним режимом при вході в систему користувачу виводитись список користувачів в системі. Це означає що процес ідентифікації є надзвичайно простим.

— В більшості Unix подібних ОС є користувач ім’я якого наперед відоме. Це root. Такі особливості дозволяють легко проходити процедуру ідентифікації.
— Зважаючи на всі чинники є не можу назвати ідентифікацію важливішою за інші процеси.

Як щодо авторизації, цей процес вже є важливішим, за процес ідентифікації, тому що він забезпечує доступу аутентифікованим об’єктом.

Але все ж якщо процедура аутентифікація була успішно пройдена зловмисником та він отримає доступ до системи як легітимний користувач, і в такому випадку він буде мати право виконувати дії які є доступні скомпрометованому користувачеві.

Отже я можу зробити висновок що аутентифікація є найважливішим етапом з трьох.

1.2. Види та способи аутентифікація

Так як аутентифікація відіграє важливу роль в процесі отримання доступу до ресурсів, або даних. Тому є багато способів для аутентифікації. Кожен з придуманих способів вирішує певні проблеми та задачі які виникають. Також існують специфічні потреби які вимагають досить робити досить цікаві технічні та програмні рішення.

Основні види аутентифікація які можна виділити, і які є досить популярні це:

— слабка аутентифікація, зазвичай це однофакторна аутентифікація на основі паролю, слабка вона тому що пароль можна підібрати, а його складність може бути не велика;
— багатофакторна аутентифікація, для аутентифікація використовуються два або більше факторів, фактори це інформація, властивість або річ якою володіє користувач;
— сувора аутентифікація, полягає в використані асиметричних криптографічних алгоритмів, під час такої аутентифікації не може бути розкритий секретний ключ, який є необхідний для аутентифікації.

Також існують багато способів аутентифікації, серед основних можна виділити:

— парольна аутентифікація, для аутентифікація використовується конфіденційна інформація, якою володіє користувач;
— біометрична аутентифікація, для аутентифікація використовується фізіологічні особливості людини які є індивідуальними для кожного. Наприклад відбиток пальців, карта сітківки ока, голосові параметри;
— за допомогою унікального предмета, для аутентифікація використовуються фізичні(апаратні) ключі, токени. Популярним на даний час є Yubikey. Це Апаратний токен, де зберігаються асиметричні ключі, та який може генерувати одноразовий пароль.

1.3 Двофакторна аутентифікація

З вище згаданих способів аутентифікація можна зробити висновок що деякі з них не можна використовувати на ПК, через ряд причин. Першою і головною причиною, є не доцільність та вартість впровадження, так наприклад, сканер відбитків пальців коштує близько 3 тис. грн. (див. Рис. 1.3.1).

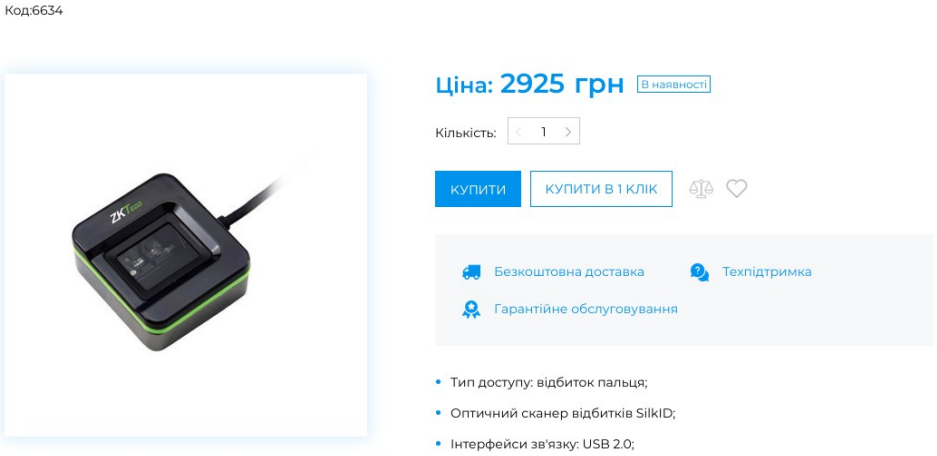

Рисунок 1.3.1 — Ціна на сканер відбитків пальців

Крім того апаратні токени також є дорогими (Рис 1.3.2) що є не доцільним та не є виправданим для більшості користувачів.

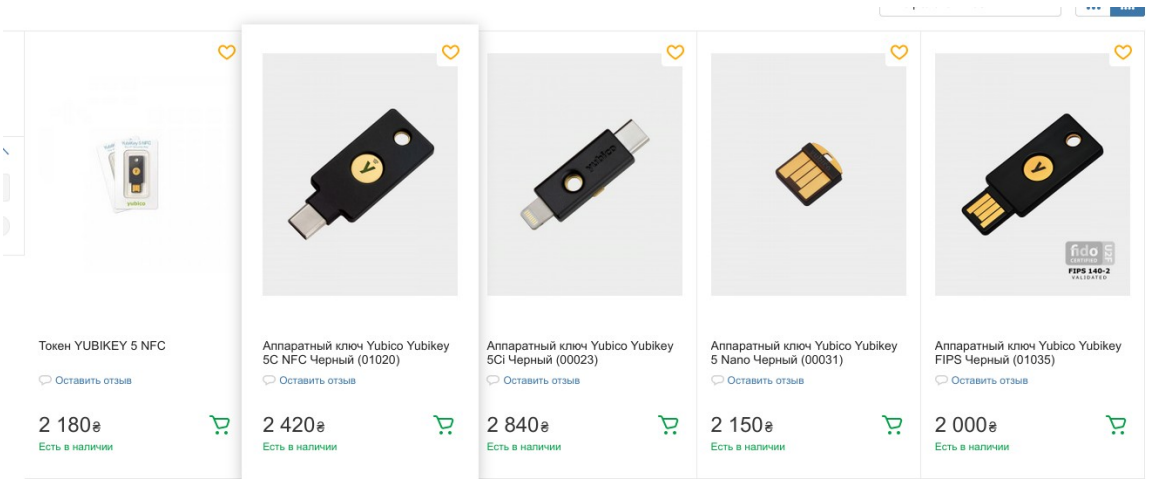

Рисунок 1.3.2 — Ціни на апаратні токени

Тому основним варіантом для користувачів є використання двофакторної аутентифікація на основі програмних рішень.

Серед програмних рішень можна використати такі програмні рішення, для аутентифікації як:

— google auhtentificator;

— microsoft authentificator;

— open otp та інші (див. Рис 1.3.3).

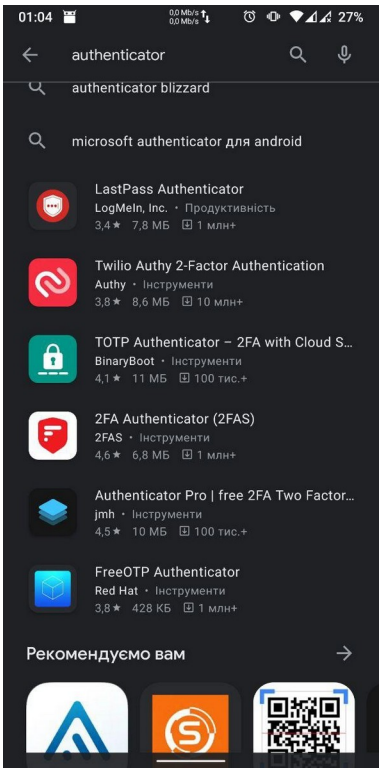

Рисунок 1.4.3 — Програми аутентифікація для ОС android

Як можна побачити на даний момент існує дуже велика кількість програм для ОС android, більшість з них використовують one time код для аутентифікації.

Ці рішення базуються на одноразовий паролях. Які в свою чергу базуються на математичних алгоритмах. Це одною перевагою є те що для кожної сесії використовуються різні паролі, що не дозволяє використати повторно пароль, якщо його отримає зловмисник.

Такі рішення чудово підходять для аутентифікації так як вони використовують мобільний телефон, які чудово для цього підходить. Основною проблемою цього рішення є те що втрата мобільного телефона, призведе до не можливості пройти аутентифікацію, що в свою чергу призведе до неможливості доступу до даних.

Отже серед способів аутентифікації не можна виокремити ідеальний, тому що кожен із способів, має свої переваги та недоліки. Їх вибір залежить від потреб та можливостей.

Що один спосіб для двофакторної аутентифікації є смс, в цьому випадку для аутентифікації потрібно вести код який приходить на номер мобільного телефону.

Однак з розповсюдженням мобільної мережі, інтерес до стільникових систем зв'язку зросла, окремі люди та компанія почали досліджувати системи стільникового зв'язку, а саме SS7 систему яка лежить в основі мобільних мереж.

Останнім часом було багато публікацій де дослідники показали багато проблем в SS7 мережах. Під час дослідження дослідники виявили, та використали багато проблем в SS7 системах, що дало можливість отримати доступ до телефонних розмов смс, та до телефонів.

Як пише у своїй роботі [2] для перехоплення повідомлення в системі SS7 потрібно знати тільки номер телефона та доступ до SS7 системи. В даркнет мережі, доступ до деякий SS7 систем можна придбати за невелику для зловмисників суму. Також проблеми в SS7 описані в роботі [3]. Він пише, що доступ до будь якої системи SS7 автоматично надає доступ і до інших систем SS7 інших операторів. Автор у своїй роботі описує варіант при якому зловмисник може отримати доступ до SS7 мережі оператора в Африці, і через SS7 систему цього оператора отримати доступ до інших операторів. При цьому зловмисника

буду складно знайти так як отримати доступ до інформації компаній іноземний країнах, буде не просто зважаючи на те, що деякі країни можуть буди проти того щоб розголошувати дані іншим країнам. Це робить SS7 систему досить небезпечною для передавання по ній важливих даних.

Хоча з розповсюдженням 4G мереж які використовують систему «Diameter» це система, яка прийшла на зміну SS7 системи. Проте після появи даної системи дослідники почали активно перевіряти їх на предмет вразливостей які були виявлені у SS7 система, я через деякий час, в новій системі виявили дуже подібні з SS7 проблеми.

Всі вище згадані проблеми в SS7 мережах та новому аналогу DIameter є причиною, щоб не рекомендувати аутентифікацію на основі смс, як основний спосіб аутентифікації для двофакторної аутентифікації.

Дуже популярним на сьогодні є апаратні токени, це факт також підтверджує дослідження у роботі [2] «Найпоширенішим різновидом токенів з пам’яттю є картки з магнітною смугою. Для використання цих токенів необхідно також мати пристрій читання. Головною перевагою застосування апаратної ідентифікації є досить висока надійність. У пам’яті токенів можуть зберігатися ключі, підібрати які хакерам не вдасться. Крім того, у них реалізовано чимало різних захисних механізмів. А вбудований мікропроцесор дозволяє електронному ключу не тільки брати участь у процесі ідентифікації користувача, але й виконувати деякі інші корисні функції. Недоліком апаратної ідентифікації є висока ціна. Взагалі ж останнім часом вартість як самих токенів, так і програмного забезпечення, що може працювати з ними, помітно знизилася.»

Як можна бачити з цього фрагменту роботи. Зараз апаратні токени стають дуже популярними тому, що люди стають більш грамотними в питаннях захисту власних даних, це зв’язано також з тим що важливість даних які зберігаються на ПК є більшою ніж колись. На даний момент на ПК зберігають фінансові та юридичні дані. Але крім цього варто розуміти, що зі зростанням важливості захисту ПК, збільшується також способи захисту, знаходяться недоліки в існуючих методах. Це звичайний процес розвитку цього напрямку в IT. Проте деякі апаратні токени на так і прості у використанні, деякі з них використовуються

для зберігання секретних ключів. А генерація таких ключів не є простою дією для звичайних користувачів. Тому апаратні токени не є хорошим варіантом для усіх людей.

1.5 Висновок до розділу

В цьому розділі було проаналізовано основні види та способи аутентифікації, було описано їхні переваги та недоліки. На жаль не виявлено способу який підходив би для усіх ситуацій, я вирішував усі існуючі проблеми, кожен з методів має свої недоліки, біометричні системи які надто дорогими для звичайний ПК та потребують спеціального ПО для функціонування. Апаратні токени також мають свої недоліки, до яких можна віднести ціну та необхідність розуміти деякі аспекти при роботі з токенами. Проте на даний момент апаратні токени досить часто використання в якості другого фактору для аутентифікації.

Останнім часом досить популярними стає ПО для android, яке базується на one time base code алгоритмі. Таку популярність цього ПО можна пояснити тим фактом що мобільні телефони стали дуже популярними та є практично у кожної людини, крім того телефон моє власні системи захисту, що підвищує загальну безпеку при використання спеціалізованого ПО для аутентифікації.

Але усі методи аутентифікації є неефективними, якщо зловмисник отримає фізичний доступ до ПК, тому перед тим як впроваджувати багатофакторну аутентифікацію, слід забезпечити базовий захист. Основним методом захисту ПК, є шифрування диску, що не дозволяє отримати доступ до даних навіть якшо зловмисник отримає фізичний доступ до ПК.

2 АНАЛІЗ ПРОЦЕСУ АУТЕНТИФІКАЦІЇ В ОС LINUX

Для аналізу системи аутентифікації в Unix подібних операційних системах було обрано систему ОС Linux, а саме дистрибутив Arch Linux з ядром версії 5.12.10. Так як Linux є однією з Unix подібних ОС.

Вибір конкретного дистрибутиву в даному випадку не є важливим, так як усі сучасні дистрибутиви використовують однакову систему аутентифікації. Яка буде описана в даному розділі. Усі описані у цьому розділ компоненти присутні та працюють в усіх сучасних Unix подібних ОС.

В даному розділі буде тільки поверхневий огляд компонентів системи аутентифікації та їх роботи. Тому, що дана тема є занадто великою для того щоб описати усі механізми в одній роботі.

На початку коли Unix подібні операційні системи тільки но почали розвиватися, програми які виконували аутентифікації, використовували власні механізми, але з часом стало зрозуміло, що механізми в багатьох системах однакові, отже їх можна вивести в окремий компонент, таким чином і з’явилася PAM система, яка зуміла закріпитися в ролі основної та надійної системи аутентифікації.

Варто зазначити що сама PAM система функціонує як основна система аутентифікації вже дуже багато часу, для неї написано багато модулів аутентифікації такий як аутентифікація на основі Access Directory сервера, який застосовується для інтеграції в Windows домен. Для kerberos та raduis вже давно існують модулі що контролюють доступ на основі цих протоколів.

Зараз PAM система використовується для побудови ІКС зі складними схемами аутентифікації та моніторингу, для цього чудові підходить PAM система адже дає можливість використати можливість розширювання системи на основі модулів, що сприяє створенню новий модулів, які призначенні вирішувати проблеми, які з’являються під час розробки впровадження та аутентифікації системи.

2.1 Основні компоненти системи аутентифікації

На даний момент основною системою аутентифікації є PAM. В перше цей механізм з’явився ще у 1996 році в Red Hed Linux 3.0.4, з того часу механізм зазнав деяких змін, але основний функціонал та компоненти залишися такими як і були. На даний момент PAM є основним механізмом аутентифікації в усіх популярних дистрибутивах Linux.

За весь період механізм удосконалився і тепер використовується майже у всіх Unix подібних ОС включаючи Mac OS, Free BSD, Solaris, Net BSD та інших. Цей факт говорить про надійність та ефективність даного механізму.

PAM використовує три основні компоненти для своєї роботи:

— клієнти, серед таких клієнтів можна виділити: sshd, login, ftpd;

— libpam.so, бібліотека яку використовують клієнти для доступу до PAM модулів;

— модулі, бібліотеки які реалізують один з чотирьох інтерфейсів, що будуть отримувати управління, на їх основі PAM система буде приймати рішення про успішну чи неуспішну аутентифікацію.

Саме розробка модулів, є основною задачею при створенні системи аутентифікації.

Розберемо детально як працює система аутентифікації на рівні ОС.

Клієнт, програма що використовує бібліотеку libpam.so, яка реалізовує доступ до системи аутентифікації PAM, та до усіх модулів аутентифікації. Для кожного клієнта, тобто для кожної програми яка використовує PAM систему, існує окремий файл конфігурацій, в ньому описано правила по який PAM система буде проводити процес ідентифікації та аутентифікації. Розберемо детально структуру конфігураційного файлу.

Усі конфігураційні файли системи PAM розміщені в директорії /etc/pam.d/<client_name>, де <client_name> назва клієнта(програми). На рисунку 2.1.1 зображено усі конфігураційні файли в даній системі.

В PAM системі існує чотири типи модулів, кожен з який відповідає, за окремі функції в процесах ідентифікації та аутентифікації, а саме:

— auth, основні типи модулів, сами вони перевіряють паролі, ім’я, та інші параметри які можуть використовуватися для двофакторної аутентифікації. Ці модулі використовуються для процедури аутентифікації;
— account, модулі даного типу виконують розподілення ресурсів між користувачами та групами користувачів;
— session, модулі даного типу виконують дії перед тим як користувач увійде в систему, найкращим прикладом таких дій є логування події входу користувачів в систему;
— password, дані типи модулів виконують перевірку паролів та інших факторів аутентифікації, наприклад перевіряють стійкість паролів, терміни їх дії, та інші важливі властивості.

У конфігураційному файлі для кожного з вище згаданих типів модулів можна використати маркер контролю, маркер контролю являє собою команду, яку PAM система використає, щоб прийняти рішення на основі результату модуля, всього існує 4 маркери контролю:

— required, якщо модуль з цим маркером верне помилку, в такому випадку процес аутентифікації продовжиться, але аутентифікація в цілому буде неуспішна, незалежно від того який результат видадуть інші модуль в конфігурації;
— requisite, дуже схожий з required, але у випадку якщо модуль до якого використано цей маркер поверне неуспішний результат, в цьому випадку процес аутентифікації відразу буде скасовано;
— sufficient, даний маркер означає, що у випадку коли модуль до якого застосованих цей маркер буде пройдений успішно, то процес аутентифікації буде вважатися успішним, та усі наступні модулі будуть пропущені. Але процес аутентифікації буде успішним, лише тоді і тільки тоді коли попередні модулі з маркерами required та requisite, будуть успішно пройдені;
— optional, модулі до який буде застосовано даний маркер, не зможуть вплинути на процес аутентифікації, незважаючи на їх результат, прикладами коли можна використовувати цей маркер можуть бути модулі які перевіряють складність паролів, та інші речі під час аутентифікації, в

такому випадку результат цих модулів не є важливим під час процесу аутентифікації. Найчастіше такі маркери використовуються для модулів типу session та password тому, що маркер optional ніяким чином не може вплинути на процес аутентифікації.

Крім маркерів контролю, існують також опції які заміняють маркери але виконують таку саму роль як і маркери контролю. Дані опції утворюються з констант (кодів повернення від pam модулів) які описані в файлі /usr/include/security/_pam_types.h та дій які можна виконати. Усі можливі варіанти таких констант є в Додатку А таблиця А.1.

До кожної з такий констант можна виконати певні дії, а саме:

— ignore, при використанні даної дії код повернення модуля буде змінено на 0. що означає успішне проходження модуля;
— bad, ця дія означає якщо отриманий код повернення до якого застосували дану дію, то модуль вважається не пройденим;
— die, якщо до коду повернення застосували дану дію, то при його отримані процес аутентифікації вважається проваленим, без виконання наступних модулів у списку;
— ok, ця дія означає що коли модуль поверне код, до якого застосували дану дію, то процес аутентифікації буде продовжуватися, з урахування пройденого модуля;
— done, практично ідентична дія до дії «ок» проте процес аутентифікації буде вважатися пройденим, і не буде виконуватися перевірка наступних модулів по списку;
— N(ціле беззнакове число), це означає що у якості отримання коду повернення до якого застосовується це число, PAM система пропустить перевірку наступних N-1 модулів та виконає модулі які йдуть після них, якщо N=0, то у цьому випадку дія буде еквівалентна до дії «ignore»;
— reset, у випадку якщо до отриманий коду повернення застосовуються дана дія, то стан PAM системи буде очищено, це означає що успішність аутентифікації не буде залежати від усіх попередніх модулів, а процес аутентифікації буде продовжуватися з наступного модуля у списку.

Варто зазначити що усі вищезгадані маркери це набір дій, застосованих до кодів повернення:

— required — [success=ok new_authtok_reqd=ok ignore=ignore default=bad];
— requisite — [success=ok new_authtok_reqd=ok ignore=ignore default=die];
— sufficient — [success=done new_authtok_reqd=done default=ignore];
— optional — [success=ok new_authtok_reqd=ok default=ignore].

У цьому списку є код повернення «default», насправді це не є код повернення, це означає що до усіх кодів повернення, для який не було застосовано дій буде присвоєно дію яка застосована до «default».

З цього стає зрозуміло поведінку таких маркерів як «sufficient», та «requisite». Sufficient до коду повернення «succcess» використовує дію done, що одразу успішно завершує процес аутентифікації всієї PAM системи, у випадку якщо модуль повертає код success (див. Додаток А. таблиця А.1). Requisite в свою чергу до «default» застосовує дію «die», що призведе до провалу усього процесу аутентифікації у випадку отримання кодів повернення який немає у списку дій.

Одними з важливих компонентів системи є модулі аутентифікації, саме на основі цих методів PAM система приймає рішення про успіх аутентифікації.

За весь час існування PAM системи було розроблено багато методів аутентифікації, які зараз входять в стандартний пакет модулів. У додатку Б таблиця Б.1 наведено увесь список стандартних модулів та їх призначення.

Серед цікавих модулів можна визначити модуль «pam_exec», так як цей модуль може запускати програми та скрипти на основі який проводити аутентифікацію, це дозволяє за досить короткий час написати програму, або скрипт на будь-якій мові програмування, не углиблюючись на низькорівневі інтерфейси PAM системи, а також не високорівневий мовах програмування є механізми контролю пам’яті, що не дозволить зробити помилок у програмах, які могли б бути якщо б програма були би написана на C чи C++. Це дозволяє в короткі терміни та без знання низькорівневих інтерфейсів PAM системи реалізувати модуль аутентифікації. Звісно при такому підході, не буде можливості використати усі можливості PAM системи.

2.2 Конфігураційні файли

Конфігураційні файли — це основні компоненти PAM системи, саме в конфігураційних файлах, описано правила та порядок модулів які будуть виконуватися та на основі який буде прийняти рішення про успішну, або провалену аутентифікацію.

Конфігураційні файли розміщенні в каталогу /etc/pam.d/<назва_клієнта>. Де клієнтом виступає програма яка виконує запит на аутентифікацію

Зазвичай контриб'ютор надає файли конфігурації разом із пакетом ПО. Проте цей файл можна модифікувати, якщо потрібно змінити схему аутентифікації. На рисунку 2.2.1 зображено вмісти каталогу /etc/pam.d, де видно які конфігураційні файли існують, варто зазначити що зазвичай назва клієнта відповідає назві ПО.

```
ZER0 :: ~ 130 » ls /etc/pam.d/
chage                  login           su
chfn                   lxdm            su-l
chgpasswd              newusers        sudo
chpasswd               other           system-auth
chsh                   passwd          system-local-login
cinnamon-screensaver   polkit-1        system-login
cups                   rlogin          system-remote-login
deepin-auth-keyboard   rsh             system-services
groupadd               runuser         systemd-user
groupdel               runuser-l       tigervnc
groupmems              samba           useradd
groupmod               sddm            userdel
i3lock                 sddm-autologin  usermod
lightdm                sddm-greeter    vlock
lightdm-autologin      shadow          xscreensaver
lightdm-greeter        sshd
ZER0 :: ~ »
```

Рисунок 2.2.1 Список усіх конфігураційних файлів

Як видно на цьому рисунку, конфігураційні файли існують не тільки для консольних утиліт, але й для графічних, що дає можливість графічним програмам блокувати екран, та проходити процес аутентифікації заново зберігаючи стан системи.

Кожен конфігураційний файл має мати чітко визначену структуру. А саме набір правил по яких PAM система буде орієнтуватися при виконанні аутентифікації. Давайте розглянемо з чого складаються ці правила.

Правила можуть мати наступний вигляд:

— <тип модуля> <маркери контролю> <модуль> <параметри модуля>;

— <тип модуля> [код_повернення=дія код_повернення=дія ...] <модуль> <параметри модуля>;

— @include <назва конфігураційного файлу>, в місці виклику ;

— # коментар, все що йде в рядку після «#» вважаються коментарями.

Давайте розглянути файл що було зрозуміло про шо йде мова. У лістингу 2.2.1 Показано вміст конфігураційного файлу для ssh сервера.

Лістинг 2.2.1 — Конфігураційний файл для sshd в PAM системі

```
#%PAM-1.0
#auth     required  pam_securetty.so     #disable remote root
auth      include   system-remote-login
account   include   system-remote-login
password  include   system-remote-login
session   include   system-remote-login
```

У лістингу 2.2.1 видно як використовуються усі типи модулів. Проте тут замість звичайних маркерів контролю, використовується маркер контролю «include» що означає що з файлу конфігурації будуть імпортовані усі модулі з даним типом, так як тут є імпорт 4 різних типів то, це означає що весь конфігураційний файл буде скопійовано.

Для того щоб розуміти які модулі будуть застосовані для sshd, треба подивитися вміст файлу system-remote-login. У лістингу 2.2.2 показано вміст файлу конфігурацій system-remote-login.

Лістинг 2.2.2 — Конфігураційний файл system-remote-login

```
#%PAM-1.0
auth      include   system-login
account   include   system-login
password  include   system-login
session   include   system-login
```

Як видно і лістингу 2.2.2 файл конфігурації system-remote-login також тільки імпортує конфігурацію з system-login. Ще означає що для аутентифікації локальна та віддалено, не має практично ніяких відмінностей, так чому створений файл

конфігурації system-remote-login, якщо можна було імпортувати усе з system-login. На це я одне пояснення, якщо вам потрібно змінити поведінку аутентифікації під для віддалений з'єднань, вам достатньо змінити файл конфігурації system-remote-login. Така структура конфігурації я особливістю дистрибутиву, кожен дистриб'ютор по різному формує такі файли, але їх суть не міняється. Що мати повну картину у лістингу 2.2.3 показано конфігураційний файли system-login.

Лістинг 2.2.3 — Конфігураційний файл system-login

```
#%PAM-1.0

auth      required   pam_shells.so
auth      requisite  pam_nologin.so
auth      include    system-auth

account   required   pam_access.so
account   required   pam_nologin.so
account   include    system-auth

password  include    system-auth

session   optional   pam_loginuid.so
session   optional   pam_keyinit.so      force revoke
session   include    system-auth
session   optional   pam_motd.so         motd=/etc/motd
session   optional   pam_mail.so         dir=/var/spool/mail standard quiet
-session  optional   pam_systemd.so
session   required   pam_env.so          user_readenv=1
```

Давайте розглянемо основні команди. Додаткові відомості про модулів можна знайти у додатку Б таблиця Б.1. Серед цікавих команд можна визначити наступні команди:

— «auth required pam_shells.so» -- перевіряє що командний інтерпретатор користувача є в списку дозволених;

— «auth requisite pam_nologin.so» -- перевіряє чи дозволений доступ в системи простих користувачів, якщо існую файл /etc/nologin то вхід в систему дозволено тільки root користувачу;

— «auth include system-auth» -- імпортує опції з файлу system-auth, саме в даному файлі відбуваються стандартні перевірки на стійкість паролю та перевіряється інші важливі параметри.

Інші команди виконують додаткові перевірки та дії, які потрібні перед тим як користувач увійде в систему. Серед таких дій можна зазначити встановлення змінних середовища для користувачів. Та монтування директорій.

2.3 Висновок до розділу

У цьому розділі було розглянуто систему аутентифікації PAM. Дана система має дуже багато можливостей за рахунок написання модулів, так як модуль аутентифікації це програма, або бібліотека яка реалізує інтерфейс, то на основі цього можна сказати, що така система має майже необмежені можливості.

Крім того система може бути налаштована таким чином, щоб забезпечити максимальний захист, якщо це буде потрібно, також PAM система може легко реалізовувати аутентифікації, як однофакторну, так і багатофакторну, багатофакторна аутентифікація реалізується додаванням модуля аутентифікації який реалізує другий фактор, до конфігураційного файлу.

Крім того PAM система може бути сконфігурована таким чином, щоб зменшити повторення коду у файлах конфігурації.

Крім того існує велика кількість модулів, практично усі додаткові можливості які можуть бути потрібно вже є в складі стандартних модулів, якщо цього не достатньо то у репозиторіях також є багато додатковий модулів аутентифікації, практично ніколи не буде потрібно реалізовувати власний модуль, так я скоріше за вже те що вам потрібно вже є реалізовано.

Основною і вагомою перевагою PAM системи є те що механізм аутентифікації є реалізовано в окремому компоненті системи, це робить таку систему дуже надійною з точки зору безпеки, так як не потрібно реалізовувати власні способи аутентифікації у власних програмах, досить просто скористатися інтерфейсом який надає PAM система в якості динамічної бібліотеки.

3 РОЗРОБКА МОДУЛЯ АУТЕНТИФІКАЦІЇ

Перед тим як починати розробку модуля потрібно визначитися, який другий фактор буде використовуватися для аутентифікації, та як реалізовувати модуль.

Існує два способи реалізувати модуль аутентифікації. Перший спосіб передбачає написання бібліотеки на C або C++, проте цей спосіб має декілька недоліків, які потрібно враховувати про розробці модуля аутентифікації. А саме в C та C++ потрібно дивитися щоб не було проблем з використання пам’яті, так як це може призвести до того що модуль може відкрити вразливість, яка може бути використана для отримання доступу до системи та обійти процес аутентифікації, тому цей варіант може бути використаний лише якщо мати досвід розробки на C, або C++.

Другий варіант це розробка програми або скрипту на високорівневій мові програмування, яка не дозволяє зробити критичні помилки, такий варіант передбачає використання програми(скрипту) з модулем аутентифікації pam_exec, деталі щодо цього модуля можна побачити у додатку Б таблиця Б.1.

Але цей спосіб має ряд недоліків, серед яких можна відзначити те що програма(скрипт) не може отримати доступу до інтерфейсу системи, усі дані які скрипт може отримати, він отримує через змінні середовища. Проте цього досить для реалізації деякого корисного функціоналу.

Не менш важливим питання перед розробкою модуля аутентифікації, який спосіб аутентифікації вибрати, адже існує велика кількість способів, частина з них описана у першому розділі даної роботи.

Проте вибір способу аутентифікації, не є простою задачею. Адже від цього буде залежати, час який буде потрібно на його реалізацію, та рівень захисту.

3.1 Вибір способу аутентифікації

Вибір способу аутентифікації є важливим етапом при розробці модуля аутентифікації тому, що від того який спосіб буде обрано для аутентифікації буде залежати термін виконання роботи, та ефективність аутентифікації.

В даній роботі я вирішив що спосіб аутентифікації буде простим, так як цілю даної роботи не є написати найефективніший алгоритм для двофакторної аутентифікації, що являється якщо не можливим, то дуже складним завданням. Тому було обрано досить простий алгоритм аутентифікації.

Під час аналізу способів аутентифікації, було оглянути дуже різні способи, проте було опущено досить вагомий спосіб аутентифікації, який на даний момент є доступний для усіх. Мова йде про спосіб аутентифікації, на основі мобільного телефона, способів аутентифікації на основі мобільного телефону є багато, наприклад за допомогою bluethooth, проте це є не найкращий спосіб, тому що будь хто може підробити ідентифікатори, якщо отримає їх з радіоефіру.

Після аналізу можливостей мобільного телефону, можна відзначити один інтерфейс, який можна використати при розробці методу аутентифікації.

Після аналізу можливостей було вирішено, що модуль аутентифікації буде заснований на ADB інтерфейсі мобільного телефона.

На основі можливостей ADB інтерфейсу розроблено алгоритм роботи модуля двофакторної аутентифікації:

— для кожного користувача який буде використовувати модуль двофакторної аутентифікації створюється файл довільно розміру з випадковими даними;
— створений на першому кроці файл зберігається на комп’ютері на якому буде впроваджуватися аутентифікація та шлях до файлу зберігається в конфігураційному файлу, який повинен розташовуватися в домашньому каталозі користувача та називатися .2fa.env;
— створений на першому кроці файл зберігається на телефоні, шлях до цього файлу зберігається в конфігураційному файлі;
— за допомогою ADB, отримується ідентифікатор телефону, за допомогою якого можна проводити аутентифікацію та зберігається його у конфігураційний файл;
— під час аутентифікації програма читає конфігураційній файл, отримує файл з мобільного телефона, за допомогою ADB;
— перевіряє отриманий файл з тим який є збережений на ПК;

— якщо файли однакові, то аутентифікація пройшла успішно, інакше неуспішно.

Я вважаю що даний спосіб аутентифікації, є достатньо надійним так як сам мобільний телефон має методи захисту, такі як, відбиток пальця, розблокування за обличчям та інші, тому щоб отримати файл який потрібен для аутентифікації, зловмиснику потрібно буде отримати доступ до мобільного телефана, що на даний час є дуже складним, через вище згадані механізми захисту мобільних телефонів.

3.2 Реалізація

Для реалізації вище описаного алгоритму я вирішив використати bash скрипт, для інтеграції його з PAM системою, використовується модуль pam_exec, даний модуль дозволяє виконувати програми або скрипти, і на основі коду повернення, вирішує про успішність аутентифікації.

Bash було обрано в якості інтерпретатора для скриту тому, що для отримання інформації про мобільний пристрій та робота з ним використовується програма «adb», так як це програма в ОС, то bash являється набором таких команд, що чудово підходить для цієї задачі.

Настав час розібратися з деталями роботи алгоритму. Щоб передати інформацію в програму модуль pam_exec передає параметри через змінні середовища. Список параметрів які pam_exec передає в програму включає:

— PAM_RHOST — IP адреса, або доменне ім’я комп’ютера з якого виконується вхід;
— PAM_RUSER — ім’я користувача якщо вхід виконується віддалено;
— PAM_SERVICE — назва програми, яка робить запит в PAM систему;
— PAM_TTY — термінал з якого виконується вхід;
— PAM_USER — ім’я користувача якщо вхід виконується локально;
— PAM_TYPE — тип PAM запиту (auth, account, session, password. Деталі в розділі 2).

Так як скрипт буде тільки працювати при локальному вході в систему то будуть використовуватися тільки PAM_USER та PAM_TYPE.

Також перед тим як писати крипт, потрібно зрозуміти як скрипт буде визначати чи користувач використовує двофакторну аутентифікацію на основі цього скрипту.

Після короткого аналізу варіантів, я зупинився на варіанті при якому скрипт перевіряє існування конфігураційного файлу «.2fa.env» у домашній директорії користувача, якщо цього файлу не існує, це означає що користувач не налаштував двофакторної аутентифікації, в цьому варіанті скрипт не буде впливати на загальний результат аутентифікації.

Якщо файл конфігурації існує в ному має бути структура, яка показана у лістингу 3.2.1.

Лістинг 3.2.1 — Структура конфігураційного файлу

```
DEVICE_ID=97faa44
ANDROID_PATH=/sdcart/.2fa.data
LOCAL_PATH=/home/roman/.2fa.data
```

У лістингу 3.2.1, показано обов’язкові параметри які повинні бути у конфігураційному файлу. Опція «DEVICE_ID» це ідентифікатор мобільного телефана отриманий за допомогою команди adb. Приклад такої команди показано на рисунку 3.2.1. На цьому рисунку деякі дані були закриті, тому що це реальні дані мого мобільного телефона.

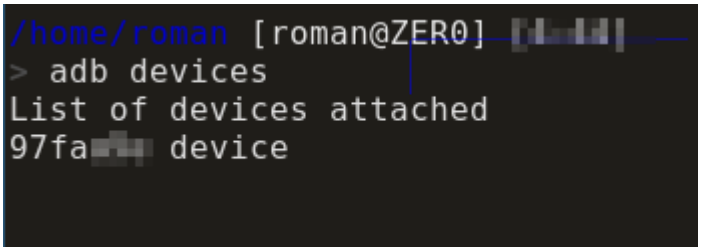

Рисунок 3.2.1 — Приклад команди для отримання ідентифікатора мобільного телефона

Опції «ANDROID_PATH» та «LOCAL_PATH» це шляхи розташування фалів на телефоні та ПК відповідно.

Крім налаштування на ПК потрібно виконати деякі налаштування на мобільному телефоні, а саме: увімкнути режим розробника, на різних моделях телефонів, це робиться по різному, тому тут цей процес описувати не має сенсу. Далі потрібно увімкнути налагодження по ADB рис 3.2.2.

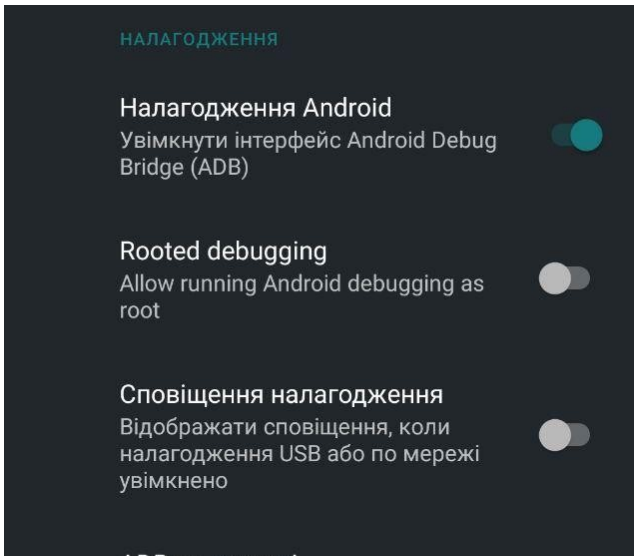

Рисунок 3.2.2 — Увімкнення налагодження по ADB

Після етапу підготовки можна писати код скрипту, зважаючи на усі вищезгадані моменти.

Увесь код скрипта показано у додатку В лістинг В.1. Розберемо основні елементи скрипта. У лістингу 3.2.2 показано код який відповідає за те щоб скрипт реагував тільки на аутентифікацію. Крім цього у лістингу 3.2.2 чітко видко що скрипт використовує bash інтерпретатор для виконання команд.

Лістинг 3.2.2 — Код перевірки тупи модуля який запусти скрипт

```
#!/usr/bin/env bash
if ! test $PAM_TYPE == "auth"; then
        exit 0
fi
```

Далі необхідно визначити чи використовує користувач двофакторну аутентифікацію за допомогою, написаного нами скрипта. На це вказує існування конфігураційного файлу, якщо його не існує скрипт повинен повернути код

успіху 0, що не впливати на процес аутентифікації. Все вищеописане у цьому абзаці робить код який показано у лістингу 3.2.3.

Лістинг 3.2.3 — Код перевірки конфігураційного файлу

```
CONF_FILE="$(getent passwd $PAM_USER | cut -d: -f6)/.2fa.env"
if ! test -f $CONF_FILE; then
        exit 0
fi
```

Якщо конфігураційний файл існує, ще означає що користувач використовує двофакторну аутентифікацію. І тому слід перевірити чи усі опції у конфігураційному файлі існують. За цю частину відповідає код показаний у лістингу 3.2.4.

Лістинг 3.2.4 — Перевірка опцій у конфігураційному файлі

```
source $CONF_FILE
if test -z $DEVICE_ID; then
        echo "No setup required option (DEVICE_ID). Check config file $CONF_FILE"
        exit 1
fi
if test -z $ANDROID_PATH; then
        echo "No setup required option (ANDROID_PATH). Check config file $CONF_FILE"
        exit 1
fi
if test -z $LOCAL_PATH; then
        echo "No setup required option (LOCAL_PATH). Check config file $CONF_FILE"
        exit 1
fi
```

Так як конфігураційних файл має формат «ключ=значення» то це можна легко перевести у змінні середовище, що і робить команда «source» на початку цього коду. Далі відбувається перевірка усіх наявних опцій.

Увесь вище показаний код був тільки підготовчим етапом, який перевіряв правильність усіх складових. Наступний код показаний у лістингу 3.2.5 вже виконує опитування на предмет того, що телефон підключений до ПК, та готовий взаємодіяти з ним через ADB протокол. Крім того для того щоб користувач розумів що відбувається на даному етапі, та розумів що він повинен зробити, користувачеві виводиться повідомлення. Для того щоб виявити потрібний телефон, в коді формується регулярний вираз на основі ідентифікатора, який є

збережений у конфігураційному файлі. Далі регулярний вираз застосовується до списку підключених телефонів які є в системі.

Лістинг 3.2.5 — Код очікування на підключення телефона

```
DEVICE_DETECT=0
GREP_REGEX="$DEVICE_ID\sdevice"
adb start-server >> /dev/null 2>&1
for second in `seq 1 10`
do
	adb reconnect >> /dev/null 2>&1
	echo "Waiting android device $second.."
	adb devices | grep $GREP_REGEX >> /dev/null 2>&1
	if  test $? == 0; then
		DEVICE_DETECT=1
		break
	fi
	sleep 4
done
```

Після перевірки наявності доступу до мобільно телефона, потрібно завантажити файл, який знаходиться за шляхом який вказаний у конфігураційному файлу як «ANDROID_PATH», після того як файл буде завантажено, потрібно переконатися що цей файл є правильним, так як на ПК за шляхом який вказаний у конфігурації як «LOCAL_PATH» є файл ідентичний тому який повинен бути на мобільному телефоні, нам достатньо перевірити, чи є файли ідентичними, і на основі цього можна робити висновок про успішну чи не успішну аутентифікацію. Про успішну аутентифікації буде стверджувати код повернення 0. Про неуспішну 1. Крім того потрібно видалити завантажений з телефона файл, так як не має сенсу зберігати цей файл в файловій системі. Також слід опрацювати варіант при якому виконання програми піде не по плану, для цього випадку в кінці скрипта після усіх перевірок йде команда «exit 1» що вказує про неуспішну аутентифікацію. Це зроблено з тією метою щоб аутентифікація не пройшла, якщо станеться проблема під час виконання скрипта.n Крім того потрібно опрацювати ситуацію, коли телефон не було виявлено під час попереднього етапу. Увесь цей функціонал виконує код який показано у лістингу 3.2.6.

Лістинг 3.2.6 — Код завантаження та перевірки файлу на ідентичність тому який є в системі

```
if test $DEVICE_DETECT == 0; then
	exit 1
else
	TARGET_DOWNLOAD="/tmp/.2fa_$PAM_USER"
	if ! adb -s $DEVICE_ID pull "$ANDROID_PATH" "$TARGET_DOWNLOAD" >> /dev/null 2>&1; then
		echo "Not found token on android device"
		exit 2
	fi
	if ! test -f "$LOCAL_PATH"; then
		echo "Not found token ok PC"
		exit 1
	fi
	if diff "$LOCAL_PATH" "$TARGET_DOWNLOAD"; then
		rm "$TARGET_DOWNLOAD"
		exit 0
	else
		rm "$TARGET_DOWNLOAD"
		exit 1
	fi
fi
exit 1
```

На даному етапі скрипт повністю готовий до тестування та інтеграції в PAM систему. Уже після цього етапу можна буде робити висновки про даний спосіб двофакторної аутентифікації.

3.3 Інтеграція та тестування модуля

Обов’язковий етап перед тим як впроваджувати модуль аутентифікації в ОС, його потрібно перевірити на предмет логічних та синтаксичних помилок, а для перевірки його роботоздатності потрібно перевірити його в середовищі для якого він був розроблений.

Інтеграція будь якого модуля в PAM систему відбувається, через додавання опису цього модуля в один або декілька файлів конфігурації.

Так як даний модуль був розроблений для локальної аутентифікації, за яку відповідає програма «login», «sudo», тому для тестування модуля було додано до цих конфігурацій опцію яка вмикає розроблений модуль. Дану опцію показано на рисунку 3.3.1.

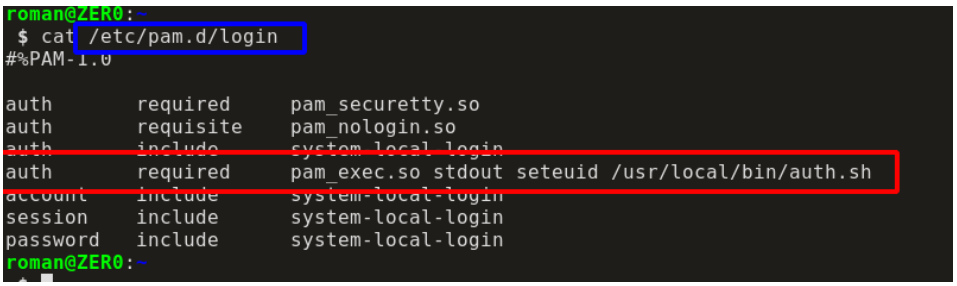

Рисунок 3.3.1 — Додавання модуля аутентифікації до PAM системи

Як видно на рисунку 3.3.1 було додано опцію з типом «auth», що вказує що модуль відповідає за аутентифікацію, з маркером контролю «required», що означає що цей модуль обов'язково успішно пройти щоб пройти процес аутентифікації. Далі іде внутрішній модуль pam_exec, який з'єднує наш скрипт з PAM системою. Параметр stdout вказує на те що в консолі скрипт може виводити повідомлення, а параметр seteuid встановлює uid скрипта як uid користувача, що дозволить скрипту читати файли користувача який і входить в систему.

Таким чином інтеграція модуля в систему відбулася. Наступний етап, це етап тестування, для цього можна, виконати процес входу в систему, декілька разів щоб перевірити, чи працює скрипт, проте перед цим необхідно підготувати файл конфігурації для користувача, тому що якщо не існує конфігурації то скрипт буде пропускати перевірку, так як ще означає що користувач не використовує двофакторну аутентифікацію.

Отже підготуємо конфігурацію. Для цього необхідно заповнити усі опції які згадувалися в попередньому підрозділі цього розділу. А саме, потрібно заповнити три опції: «DEVICE_ID», «ANDROID_PATH» та «LOCALE_PATH». Що вони означають, де їх взяти, та на що вони впивають говорилося в попередньому підрозділі. На рисунку 3.3.2 показано вміст конфігураційного файлу.

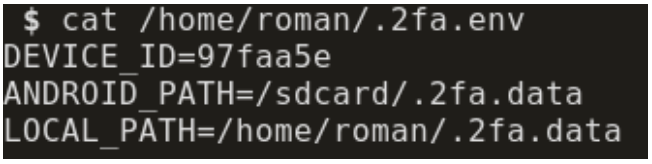

Рисунок 3.2.2 — Вмісти конфігураційного файлу для користувача «roman»

Далі потрібно створити файл /home/roman/.2fa.data. Так як бажано щоб вміст цього файлу був як найбільш випадковим, скористаємося командою «dd if=/dev/urandom of=/home/roman/.2fa.data bs=1M count=1» таким чином у нас буде створений файл, з випадковими даними, ці дані генеруються на основі апаратних характеристик, таких як тактова частота, рух мишки, та набір на клавіатурі, тому ці дані мають високу ступінь випадковості.

Після того як ми створили файл нам потрібно зберегти цей файл на телефон за вказаним у конфігурації шляхом. Для цього використаємо наступну команду «adb -s 97faa53 push /home/roman/.2fa.data/ sdcard/.2fa.data». Крім того важливо зберегти цей файл в надійне місця «не на ПК чи телефоні», щоб можна було відновити його, якщо ви втратити його на телефоні.

Після усіх цих дій можна починати тестування модуля. Для цього можна скористатися програмою «pamtester» яка робить запит до PAM системи імітуючи іншу програму. На рисунках 3.2.3 та 3.2.4 протестовано процес аутентифікації для програм «login» та «sudo» відповідно.

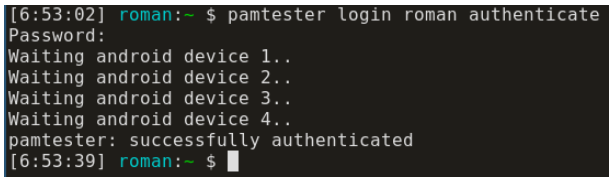

Рисунок 3.2.3 — Тестування модуля для програми «login»

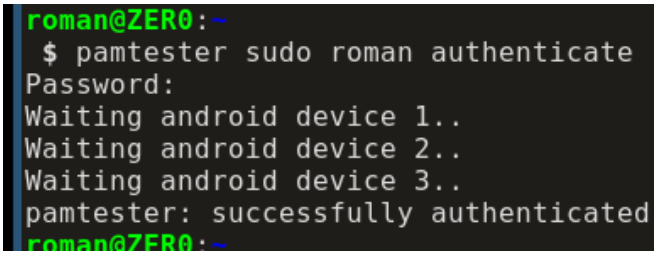

Рисунок 3.2.4 — Тестування модуля для програми «sudo»

3.4 Висновок до розділу

У третьому розділі було описано та реалізовано алгоритм аутентифікації, який пропонується використовувати мобільний телефон в якості другого фактору для аутентифікації, адже на даний час мобільні телефони є дуже популярними то

розповсюдженими, крім того телефон, має високий рівень захисту, що не дозволить зловмиснику отримати файл який використовується як додаткова перевірка перевірка, крім ідентифікатора телефона.

Реалізований у цьому розділі алгоритм, є протестований та успішно пройшов тест, вже на моєму ПК, я зараз його використовую, я за час використання проблем не було виявлено.

Для того щоб не було проблеми при використанні цього методу, я рекомендую виконати ряд наступних дії:

— заборонити змінення та видалення конфігураційного файлу та файлу для перевірки, використовуючи засоби операційної системи;
— зробити резервну копію файлу перевірки, щоб можна було відновити його у випадку втрати;
— налаштувати PAM систему таким чином щоб у випадку втрати мобільного телефона можна було отримати доступ до ПК.
— Проте даний алгоритм потрібно розвивати, та шукати способи покращення, але для поставленої на цю роботу задачі він цілком підходить.

В якості модернізація даного алгоритму, можна запропонувати переписати його на мову програмування C. Тобто реалізувати його як повноцінний модуль, така реалізація може відкрити додаткові можливості, та позбавити недоліків. Проте це потрібно робити тільки якщо ви дуже впевнено володієте мовою програмування C. Тому що помилка в такій програмі може призвести до отримання зловмисником повного доступу до системи, що практично неможливо зробити при помилці у скипті.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Роль центральної нервової системи в трудовій діяльності людини

Нервова система має найголовніше значення в організмі людини. Вона координує, регулює роботу всіх внутрішніх органів і здійснює зв'язок організму із зовнішнім середовищем. Нервова система людини складається із центральної (ЦНС), яка включає головний і спинний мозок і периферійної (ПНС), яка складається з нервових волокон, що відходять від головного і спинного мозку. За функціями нервову систему поділяють на соматичну і вегетативну. Соматична нервова система регулює опорно-руховий апарат і всі органи чуття, а вегетативна - процес обміну речовин та роботу всіх внутрішніх органів (серця, нирок, легенів). Найпростіші рухи регулює спинний мозок. Довгастий мозок керує процесами травлення, дихання, кровообігу та іншими життєво важливими функціями. Підкіркова і кіркова частини головного мозку керують усією психічною діяльністю людини.

Центральна нервова система виконує рефлекторну, інтегративну та координаційну функції.

Рефлекторна діяльність мозку зумовлена безумовними та умовними рефлексами. Безумовні рефлекси є вродженими, мають велику стійкість і забезпечують пристосування організму до зовнішнього середовища. Умовні рефлекси набуваються залежно від обставин, розширюють діапазон пристосувальницьких можливостей організму і згасають, якщо потреби в них немає.

Стійка і злагоджена система умовних рефлексів формується у процесі навчання і забезпечує виконання певного виробничого завдання. Стійкість системи умовних рефлексів може бути порушена при відхиленні трудової діяльності від програми, а надійність - під впливом несприятливих виробничих чинників. Такі порушення, якщо не вжити належних заходів, можуть призвести до зниження працездатності, травм або нещасних випадків.

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

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

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

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

Анатомічною основою другої сигнальної системи, яка властива тільки людині, є мовно-руховий апарат, тісно пов'язаний із зоровим та слуховим аналізаторами, а її подразником є слово. Мова, в усіх її видах, являє собою найбагатше джерело подразників. За допомогою слова передаються сигнали про конкретні подразники, і в цьому випадку слово служить принциповим подразником - сигналом сигналів, є пусковим механізмом дій і вчинків людей. Мова підвищує здатність мозку відображати дійсність, забезпечує аналіз і синтез, абстрактне мислення, створює можливість для спілкування, використання і передачі життєвого досвіду, досягнень культури і мистецтва. Але в деяких випадках слово може бути негативним подразником і може призвести до розладів нервової системи, порушень функціонування всіх систем організму і, таким чином, стати небезпечним виробничим фактором.

Центральна нервова система бере участь у прийманні, обробці та аналізі будь-якої інформації, що надходить із зовнішнього і внутрішнього середовищ. При

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

4.2 Долікарська допомога при пораненнях

При будь-якому порушенні цілісності шкіри і глибоко розташованих тканин необхідно обробити шкіру навколо рани розчином йоду, спиртом тощо. Не рекомендується промивати рану водою або дезінфікуючим розчином. Після обробки рани необхідно накласти асептичну пов'язку. Пов'язка захищає рану від забруднення, інфікування, зменшує біль, а вигляд перев'язаної рани заспокоює хворого.

Обробка рани потребує додержання таких правил: перед обробкою рани необхідно помити руки (якщо поряд немає води, їх слід протерти спиртом або бензином):

- невеликі поранення, садна після обробки шкіри навколо них настойкою йоду або перекисом водню заклеюють лейкопластиром чи медичним клеєм БФ-6;
- не можна видаляти із ран сторонні тіла або бруд, тому, що можна пошкодити судини і викликати кровотечу;
- шкіру навколо рани протирають від країв до периферії шматочком марлі, бинта або вати, яка змочена спиртом, спиртовим розчином йоду чи бензином. (не можна заливати рану йодом!)
- із бинта або індивідуального пакета зробити салфетку такого розміру, щоб вона закривала усю рану, накласти її на ранову поверхню, забинтувати або приклеїти смужками лейкопластиру;
- якщо в рані видно внутрішні органи, мозок або сухожилля, потрібно акуратно накласти стерильну пов'язку, щоб у рану не потрапила інфекція, або краще накрити рану стерильним матеріалом.

При проникаючих пораненнях перша допомога спрямована на запобігання інфікування рани. Проникаючі поранення грудної клітки можуть супроводжуватись кровохарканням, проникненням повітря до підшкірної клітковини, яке виглядає як набряк, але викликає хруст при прощупуванні,

накопиченням повітря у плевральній порожнині (пневмоторакс), накопиченням крові у плевральній порожнині (гемоторакс).

Пневмоторакс - це скупчення атмосферного повітря в плевральній порожнині, яке потрапляє через відкриту рану грудної клітки (відкритий пневмоторакс), або при ушкодженні легені чи бронха (закритий пневмоторакс). При його виникненні можливе балотування органів середостіння, що супроводжується розладом кровообігу і дихання. Перша допомога при проникаючих пораненнях грудної клітки має бути спрямована на ліквідацію пневмотораксу, попередження шоку, захист рани від інфікування. Необхідно надати потерпілому напівсидяче положення; накласти герметичну пов'язку, щоб зробити перепону для попадання повітря до плевральної порожнини. Для цього після обробки рани, її закривають смужками лейкопластиру, який накладається у вигляді черепиці. Можна використовувати обгортку від індивідуального перев'язочного пакету, клейонку, целофановий пакет, серветки, що оброблені вазеліном, які потім фіксуються бинтовою пов'язкою до грудної клітки.

Перша допомога при проникаючих пораненнях органів черевної порожнини. Частіше за все це вогнепальні і колото-різані рани. Для всіх поранень черевної порожнини характерним є різкий біль у животі, напруження м`язів черевної стінки (живіт, як "дошка") і симптоми внутрішньої кровотечі, шоку й колапсу. Значна кровотеча спостерігається при ушкодженнях паренхіматозних органів (печінка, селезінка, нирки). Розвиваються характерні симптоми кровотечі: наростає слабкість, настає загальна блідість, нудота, блювання, похолодіння кінцівок. Пульс частий і слабкий, артеріальний тиск падає. Поранення органів шлунково-кишкового тракту супроводжується виходженням у вільну черевну порожнину кишкового вмісту та інфікуванням її, внаслідок чого швидко розвивається запалення очеревини (перитоніт).

У всіх випадках проникаючих поранень черевної порожнини необхідно обробити рану і накласти асептичну пов'язку. Петлі кишечнику і сальник, які випали в рану, у черевну порожнину не вправляти. Якщо не має абсолютних симптомів проникаючого поранення у живіт, знеболюючих засобів не призначати. Прийом води і їжі категорично забороняється. Пораненого у живіт необхідно

негайно госпіталізувати, транспортувати його у положенні лежачи на носилках. При проникаючих пораненнях черевної порожнини показана екстрена операція в перші години, поки не розвинулась інфекція і хворий не втратив багато крові.

4.3 Висновок до розділу

У першому питанні цього розділі описано роль центральної нервової системи в трудовій діяльності людини, було проаналізовано плив різних факторів на центральну нервову систему. Це було взято за основу при розроблені алгоритму, алгоритм був розроблений таким чином, що максимально унеможливити поганий вплив на центральну нервову систему.

У другому питанні проаналізовано методи долікарської допомоги при пораненнях, описано які типи поранень існують, і яку допомогу слід надавати при кожному з них щоб не завдати більшої шкоди. Тому що головне в такій справі не нашкодити більше.

ВИСНОВКИ

Під час виконання даної роботи було опрацьовано доволі багато інформації щодо процесів аутентифікації, їх видів та способів. Також було проаналізовано основні переваги та недоліки кожного з способів аутентифікації.

Так як дана робота було націленою на розробку модуля аутентифікації для Unix подібних ОС, то велику частину роботи було присвячено тому, як працює система аутентифікації, які компоненти присутні в цій системі, та як вони взаємодіють між собою. Було описано методи управління системою аутентифікації.

Було проаналізовано існуючі модулі в системі аутентифікації, описано їх призначення, та ситуації коли їх доцільно використовувати.

Наряду з цим усім, було виконано основне завдання, яке було поставлення для даної роботи, а саме було описано та розроблено модуль для двофакторної аутентифікації. Його роботу було протестовано, в різних середовищах, описано основні переваги та рекомендації щодо нього.

Отже спираючись на усе вище написане, можна зробити висновок, що поставлене на дипломну роботу завдання виконано успішно. Звичайно можуть бути деякі нюанси, які не були враховані під час написання роботи, але основну ціль було досягнуто.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Афанасьєв А.А., Веденєв Л.Т., Воронцов А. А. Аутентифікація. Теорія і практика забезпечення безпечного доступу до інформаційних ресурсів / А.А. Афанасьєв, Л.Т. Веденєв, А.А. Воронцов // Навчальний посібник для вузів. –2009. –No1. –552 с
2. Сарбуков А. Аутентификация в компьютерных системах / А.СарбуковА. Грушо.// Системы безопасности. –2003. -No5 (53).–С. 25-29
3. ЧепиковО. Особенности применения двухфакторной аутентификации/ О.Чепиков // Информационная безопасность. –2005. –No3.–С.35-41.
4. Анна Чунарьова, Руслана Зюбіна Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2 (26), 2013 р., ISSN2074-9481
5. Malan. Классификация механизмов аутентификации пользователей и их обзор [Електронний ресурс] / Malan. –2013. –Режим доступу до ресурсу:https://habr.com/ru/post/177551/
6. Выростков Д. Обзор способов и протоколов аутентификации в веб-приложениях [Електронний ресурс] / Дмитро Выростков. –2015. –Режим доступу до ресурсу:https://habr.com/ru/company/dataart/blog/262817/.
7. Бедрій І.Я., Нечай В.Я. Безпека життєдіяльності. Навчальний посібник. – Львів: Манголія 2006, 2007. 499 с.
8. Гандзюк М.П., Желібо Є.П., Халімовський М.О. Основи охорони праці. – К.: Каравела, 2007. 408 с.

ДОДАТКИ

ДОДАТОК А

Коди повернення для модулів у PAM системі

Таблиця А.1 — Коди повернення для модулів у PAM системі.

Символьний Код повернення	Скорочення	Значення	Деталі
PAM_SUCCESS	success	0	Модуль успішно пройдено
PAM_OPEN_ERR	open_err	1	Не можливо завантажити модуль
PAM_SYMBOL_ERR	symbol_err	2	Не знайдено функії в аутентифікації модулі
PAM_SERVICE_ERR	service_err	3	Помилка в коді модуля
PAM_SYSTEM_ERR	system_err	4	Помилка в PAM системі
PAM_BUF_ERR	buf_err	5	Помилка «переповнення буфера» в модулі
PAM_PERM_DENIED	perm_denied	6	Помилка доступу до ресурсів
PAM_AUTH_ERR	auth_err	7	Неуспішна аутентифікація
PAM_CRED_INSUFFICIEBT	cred_insufficient	8	Не можна отримати доступ до даних аутентифікації
PAM_AUTHINFO_UNAVAIL	authinfo_unavail	9	Не вдалося отримати інформацію для аутентифікації
PAM_USER_UNKNOWN	user_unknown	10	Не відомий користувач
PAM_MAXTRIES	maxtries	11	Перевищено кількість спроб для аутентифікації
PAM_NEW_AUTHTOK_REQD	nex_authtok_reqd	12	Потрібно оновити токен, зазвичай цей код використовується щоб повідомити що термін дії паролю вийшов
PAM_ACCT_EXPIRED	acct_expired	13	Термін дії користувача в системі минув
PAM_SESSION_ERR	session_err	14	Не можна створити сесії для користувача
PAM_CRED_UNAVAIL	cred_unavail	15	Не можливо отримати дані користувача
PAM_CRED_EXPIRED	cred_expired	16	Термін дії облікових даних користувача закінчується

Продовження таблиці А.1

Символьний Код повернення	Скорочення	Значення	Деталі
PAM_CRED_ERR	cred_err	17	Помилка при додавані або оновленні даних користувача
PAM_NO_MODULE_DAT	no_module_data	18	Відсутні додаткові дані для модуля, наприклад конфігураційні файли
PAM_CONV_ERR	conv_err	19	Помилка в інтерфейсі між модулем та користувачем
PAM_AUTHTOK_ERR	authtok_err	20	Помилка під час дій над токеном аутентифікації
PAM_AUTHROK_RECOVERY_ERR	authtok_recovery_err	21	Неможливо відновити інформацію щодо аутентифікації
PAM_AUTHTOK_LOCK_BUSY	authtok_lock_busy	22	Токен для аутентифікації є заблокований
PAM_AUTHTOK_DISABLE_AGING	authtok_disable_aging	23	Вимкнено можливість змів в токені аутентифікації
PAM_TRY_AGAIN	try_again	24	Виконується попередня перевірка паролю
PAM_ignore	ignore	25	Модуль проігнорував перевірку
PAM_ABORT	abort	26	Критична помилка всередині модуля
PAM_AUTHTOK_EXPIRED	authtok_expired	27	Термін дії токена аутентифікації минув
PAM_MODULE_UNKNOWN	module_unknown	28	Не знайдено модуля аутентифікації
PAM_BAD_ITEM	bad_item	29	В модулю передано неправильний тип даних
PAM_CONV_AGAIN	conv_again	30	Недоступний інтерфейс між користувачем та PAM системою
PAM_INCOMPLETE	incomplete	31	Модуль необхідно запустити ще раз

ДОДАТОК Б

Стандартні модулі які входять в пакет PAM систем

Таблиця Б.1 — Стандартні модуль які входять в пакет PAM систем

Назва модуля	**Тип Модуля**				**Конфігураційні файли**	**Інформація про модуль**
	account	**session**	**auth**	**password**		
pam_access	+	-	-	-	/etc/security/access.conf	Забезпечує контроль доступу на основі IP адрес, доменів, та імен користувачів
pam_chroot	+	+	+	-		Виконує зміну кореневого каталогу для користувачів під час аутентифікації
pam_console	-	+	+	-	etc/security/console.perms, /etc/security/console.apps/*	Забезпечує спеціальні права на файли пристроїв
pam_cracklib	-	-	-	+		Перевіряє пароль на надійність, для цього використовує вбудовані типи перевірок
pam_deny	+	+	+	+		Просто повертає коди: «PAM_AUTH_ERR», «PAM_CRED_ERR», «PAM_AUTHTOK_ER», «PAM_SESSION_ERR». Для кожного з типів відповідно.
pam_env	-	-	+	+	etc/security/pam_env.conf	Встановлює, або очищує зміні середовища
pam_filter	+	+	+	+		Застосовує фільтри, просто зараз це майже не використовується
pam_ftp	-	-	+	+		Дозволяє анонімний доступ, використовується для ftp серверів
pam_group	-	-	+	-	/etc/security/group.conf	Додає користувачів до різних груп на основі правил в файлі конфігурації
pam_krb4	-	+	+	+		Аутентифікація на основі kerberos
pam_lastlog	-	-	+	-		Виконує запис про аутентифікацію в лог файл /var/log/lastlog
pam_limits	-	+	-	-	/etc/security/limits.conf	Контролює за обмеження спроб доступу та блокує спроби при перевищені ліміту
pam_nologin	-	-	+	-		Блокує вхід в систему усіх для усіх користувачів крім «root» користувача, якщо існує файл /etc/nologin

Продовження таблиці Б.1

Назва модуля	**Тип Модуля**				**Конфігураційні файли**	**Інформація про модуль**
	account	**session**	**auth**	**password**		
pam_time	-	-	+	-	/etc/security/time.conf	Контролює аутентифікацію користувачів на основі часу, та конфігурації, можна використовувати для блокування доступу в неробочий час
pam_radius	-	+	-	-		Забезпечує аутентифікацію на основі RADIUS
pam_unix	+	+	+	+		Стандартний модуля для unix, який перевіряє паролі у файлі /etc/passwd та /etc/shadow із введеним користувачем на основі цього надає доступ
pam_userdb	-	-	+			Для перевірки використовується спеціальна база даних, часто використовується для доступу до веб додатків
pam_rootok	-	-	+	-		Дозволяє аутентифікацію тільки якщо UID дорівнює 0
pam_tally	+	-	+	-		Контролює щоб не було багато спроб аутентифікації за короткий час, та блокує доступ якщо спроб є багато
pam_shells	+	-	+	-	/etc/shells	Дозволяє аутентифікацію, якщо інтерпретатор користувача є у файлі конфігурації
pam_exec	+	+	+	+		Дозволяє запускати стороні програми(скрипти) і на основі коду повернення вирішує чи пройшла аутентифікація, цей модуль дозволяє досить легко написати просто перевірку, на скриптових мовах наприклад python, ruby, javascrit

ДОДАТОК В

Лістинг В.1 — Код скрипту для двофакторної аутентифікації

```
#!/usr/bin/env bash
if ! test $PAM_TYPE == "auth"; then
	exit 0
fi
CONF_FILE="$(getent passwd $PAM_USER | cut -d: -f6)/.2fa.env"
if ! test -f $CONF_FILE; then
	exit 0
fi
source $CONF_FILE
if test -z $DEVICE_ID; then
	echo "No setup required option (DEVICE_ID). Check config file $CONF_FILE"
	exit 1
fi
if test -z $ANDROID_PATH; then
	echo "No setup required option (ANDROID_PATH). Check config file $CONF_FILE"
	exit 1
fi
if test -z $LOCAL_PATH; then
	echo "No setup required option (LOCAL_PATH). Check config file $CONF_FILE"
	exit 1
fi

DEVICE_DETECT=0
GREP_REGEX="$DEVICE_ID\sdevice"
adb kill-server >> /dev/null 2>&1
adb start-server >> /dev/null 2>&1
for second in `seq 1 10`
do
	echo "Waiting android device $second.."
	adb devices | grep $GREP_REGEX >> /dev/null 2>&1
	if  test $? == 0; then
		DEVICE_DETECT=1
		break
	fi
	sleep 4
done

if test $DEVICE_DETECT == 0; then
	exit 1
else
	TARGET_DOWNLOAD="/tmp/.2fa_$PAM_USER"
	if ! adb -s $DEVICE_ID pull "$ANDROID_PATH" "$TARGET_DOWNLOAD" >> /dev/null 2>&1;
then
		echo "Not found token on android device"
		exit 2
	fi
```

Продовження лістингу В.1

```
        if ! test -f "$LOCAL_PATH"; then
                echo "Not found token ok PC"
                exit 1
        fi
        if diff "$LOCAL_PATH" "$TARGET_DOWNLOAD"; then
                rm "$TARGET_DOWNLOAD"
                exit 0
        else
                rm "$TARGET_DOWNLOAD"
                exit 1
        fi
fi
exit 1
```