ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА ТЕХНІЧНИХ СИСТЕМ КІБЕРЗАХИСТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Система багатофакторної ідентифікації та автентифікації користувача при роботі з інформацією з обмеженим доступом»

на здобуття освітнього ступеня магістра
зі спеціальності __125 - _Кібербезпека та захист інформації»
(код, найменування спеціальності)
освітньо-професійної програми Технічні системи інформаційного та кібернетичного захисту

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Олександр ПАНАСЮК

Виконав: здобувач вищої освіти групи СЗДМ 61
Олександр ПАНАСЮК

Керівник: _____ТУРОВСЬКИЙ Олександр___
д.т.н. , професор (ПРІЗВИЩЕ, Ім'я)

Рецензент: _____
(ПРІЗВИЩЕ, Ім'я)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

Кафедра ___Технічних систем кіберзахисту_____
Ступінь вищої освіти ____магістр_____
Спеціальність ___125-Кібербезпека та захист інформації_____
Освітньо-професійна програма Технічні системи інформаційного та кібернетичного захисту

ЗАТВЕРДЖУЮ
Завідувач кафедри ТСКБ
_____Олександр ТУРОВСЬКИЙ

«____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Панасюку Олександру Олексіювичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи:
«Система багатофакторної ідентифікації та автентифікації користувача при роботі з інформацією з обмеженим доступом»
керівник кваліфікаційної роботи:ТУРОВСЬКИЙ Олександр, д.т.н., професор

затверджені наказом Державного університету інформаційно-комунікаційних технологій
від « 15 » жовтня 2024 р. № 320

2. Строк подання кваліфікаційної роботи 14.12.2024 р.

3. Вихідні дані до кваліфікаційної роботи: опорний об'єкт інформаційної діяльності, технічні канали витоку інформації, вимоги до захисту інформації з обмеженим доступом, сучасні засоби багатофакторної ідентифікації та автентифікації користувача при роботі з інформацією з обмеженим доступом.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
4.1. Захист інформації на об'єктах інформаційної діяльності.
4.2. Огляд методів і засобів ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.
4.3. Розробка та вдосконалення механізмів багатофакторної ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом

5. Перелік графічного матеріалу: ___Презентаційний матеріал на слайдах.

6. Дата видачі завдання _____20.10.2024 ___ _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури		
2	Написання першого розділу роботи		
3	Написання другого розділу роботи		
4	Написання третього розділу роботи		
5	Написання висновків по роботі		
6	Підготовка демонстраційних матеріалів		
7	Підготовка доповіді		

Здобувач вищої освіти _____ Олександр ПАНАСЮК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник роботи _____ Олександр ТУРОВСЬКИЙ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Кваліфікаційна робота містить вступ, три розділи, загальні висновки, список використаних джерел і додаток. Орієнтовний обсяг роботи становить 72 сторінки; у тексті збережено 19 рисунків і 5 таблиць. Бібліографічний список налічує 31 позицію.

Метою кваліфікаційної роботи є підвищення надійності доступу до конфіденційної інформації шляхом удосконалення механізмів багатофакторної ідентифікації та автентифікації користувачів.

У роботі узагальнено вимоги до захисту інформації на об'єкті інформаційної діяльності, досліджено загрози та технічні канали витоку, систематизовано сучасні методи ідентифікації й автентифікації, а також обґрунтовано підходи до побудови багатофакторного механізму доступу до інформації з обмеженим доступом.

Апробація:

О.О. Панасюк, В.О. Марченко, Р.В. Скоробагатько, Технології ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом*: Актуальні проблеми безпеки інформаційно-телекомунікаційних систем*. м. Київ, 03 листопада 2024 року, Київ: РВЦ ДУІКТ. – 024. С.83 - 84. https://duikt.edu.ua/uploads/p_2661_93669247.pdf

Ключові слова: ЗАХИСТ ІНФОРМАЦІЇ, ОБ'ЄКТ ІНФОРМАЦІЙНОЇ ДІЯЛЬНОСТІ, НЕСАНКЦІОНОВАНИЙ ДОСТУП, ПАРОЛЬ, QR-КОД, ТОКЕН, БІОМЕТРІЯ, БАГАТОФАКТОРНА АВТЕНТИФІКАЦІЯ.

ABSTRACT

The qualification work contains an introduction, three sections, general conclusions, a list of sources used and an appendix. The approximate volume of the work is 72 pages; 19 figures and 5 tables are preserved in the text. The bibliographic list has 31 items.

The purpose of the qualification work is to increase the reliability of access to confidential information by improving the mechanisms of multi-factor identification and authentication of users.

The work summarizes the requirements for information protection at the object of information activity, investigates threats and technical leakage channels, systematizes modern methods of identification and authentication, and also substantiates approaches to building a multi-factor mechanism for access to information with limited access.

Approbation:

O.O. Panasyuk, V.O. Marchenko, R.V. Skorobagatko, Technologies for identification and authentication of users when accessing information with restricted access: Current security problems of information and telecommunication systems. Kyiv, November 3, 2024, Kyiv: DUIKT Research and Development Center. – 024. P.83 - 84. https://duikt.edu.ua/uploads/p_2661_93669247.pdf

Keywords: INFORMATION PROTECTION, OBJECT OF INFORMATION ACTIVITY, UNAUTHORIZED ACCESS, PASSWORD, QR-CODE, TOKEN, BIOMETRICS, MULTI-FACTOR AUTHENTICATION.

ЗМІСТ

СПИСОК УМОВНИХ СКОРОЧЕНЬ ……. ..7
ВСТУП……..…….. ..8
РОЗДІЛ 1 ЗАХИСТ ІНФОРМАЦІЇ НА ОБ'ЄКТАХ ІНФОРМАЦІЙНОЇ ДІЯЛЬНОСТІ ……..…….. 9
1.1. Аналіз загроз інформаційній безпеці на об'єкті інформаційної діяльності ……..……..……... 9
1.2. Аналіз каналів витоку інформації на об'єкті інформаційної діяльності . 10
1.3 Висновки по розділу……..……..…… 13

РОЗДІЛ 2 ТЕХНОЛОГІЇ ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ ПРИ ДОСТУПІ ДО РОБОТИ З ІНФОРМАЦІЄЮ З ОБМЕЖЕНИМ ДОСТУПОМ ……...……..……. 14
2.1 Процес ідентифікації та автентифікації користувачів в інформаційно-комунікаційних мережах ……..…….. 14
2.2 Класифікація та характеристики методів ідентифікації та автентифікації користувача ……. 16
2.3 Вимоги та порядок організації процесів ідентифікації та автентифікації 35
2.4 Висновки по розділу ……... 40

РОЗДІЛ 3 РОЗРОБКА МЕХАНІЗМІВ БАГАТОФАКТОРНОЇ ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ ПРИ ДОСТУПІ ДО РОБОТИ З ІНФОРМАЦІЄЮ З ОБМЕЖЕНИМ ДОСТУПОМ 41
3.1 Зміст, характеристики та вимоги до механізму ідентифікації та автентифікації користувачів в інформаційно-комунікаційних мережах ….. 41
3.2 Моделі оцінки захищеності механізмів багатофакторної автентифікації користувачів від несанкціонованого доступу …… 47
3.3 Висновки по розділу ……..……..……..... 51

ВИСНОВКИ……..……. 52
ПЕРЕЛІК ПОСИЛАНЬ……..……... 54
ДОДАТОК ……..……... 58

СПИСОК УМОВНИХ СКОРОЧЕНЬ

ОІД	–	Об’єкт інформаційної діяльності
ІзОД	–	Інформація з обмеженим доступом
ТЗІ	–	Технічний захист інформації
ТКВІ	–	Технічні канали витоку інформації
ОТЗС	–	Об’єкт технічного захисту систем
ТЗО	–	Технічні засоби охорони
ТСО	–	Технічна система охорони
ТЗОС	–	Технічні засоби охоронної сигналізації
ІКМ		Інформаційно-комунікаційна мережа

ВСТУП

На сучасному етапі розвитку суспільства, однією з найактуальніших проблем, яка є не тільки в Україні, а й в світі, полягає в захисту інформації на об’єктах інформаційної діяльності. Одним з аспектів вирішення вказаного питання є організація та здійснення багатофакторної ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.

Це, в свою чергу, вимагає проведення відповідних досліджень та на їх основі удосконалення методів та механізмів вказаної багатофакторної ідентифікації та автентифікації користувачів.

Метою кваліфікаційної роботи удосконалення механізмів багатофакторної ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.

У процесі підготовки кваліфікаційної роботи були поставлені наступні задачі:

– Визначено порядок розподілу інформації на об’єкті інформаційної діяльності, проведено аналіз загроз інформаційній безпеці та аналіз технічних каналів витоку інформації на об’єкті інформаційної діяльності.

– Проведено методів і засобів ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.

– Розроблено та вдосконалено механізми ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.

Об’єкт дослідження. Процес ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.

Предмет дослідження. Механізми ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.

РОЗДІЛ 1
ЗАХИСТ ІНФОРМАЦІЇ НА ОБ'ЄКТАХ ІНФОРМАЦІЙНОЇ ДІЯЛЬНОСТІ

1.1. Аналіз загроз інформаційній безпеці на об'єкті інформаційної діяльності

Модуль 2-8: Криптографічні атаки

Цей модуль охоплює наступні цілі CompTIA Security+:

- 1.2 Порівнювати і протиставляти типи атак

- 2.2 Використовувати відповідні програмні засоби для оцінки стану безпеки в організації за певним сценарієм

- 6.1 Порівняйте та порівняйте основні концепції криптографії

Криптоаналіз, наука про злам шифрування, така ж давня, як і саме шифрування. Від Людовика XVI в його палаці у Версалі до останніх атак, спонсорованих державою, шифрування успішно атакували багато разів. У цьому модулі розглядаються деякі з найпоширеніших криптографічних атак та інструменти, що використовуються для їх здійснення. Розуміння цих атак та їхніх інструментів дає змогу краще зрозуміти, як захистити дані. (До того ж, зламувати щось дуже весело!)

Модуль починається із загального обговорення стратегій атак, а потім розглядаються дані, що піддаються атакам. У другому розділі розглядаються атаки, орієнтовані на паролі, та інструменти, які зазвичай використовуються в таких атаках. Модуль закінчується розглядом кількох атак, які не пов'язані з паролями.

Стратегії атак

Давайте почнемо з розгляду того, де можна атакувати будь-яку криптосистему. Кожна криптосистема має базовий алгоритм, ключ (або у випадку з хешами - дайджест), або певну реалізацію самої криптосистеми, і все це зрозуміло зловмиснику. Всі криптографічні алгоритми відомі і рідко змінюються, відповідно до принципу Керкхоффа. Кожна криптографічна атака починається з ретельної перевірки криптосистеми і агресивного зондування та

експериментів, щоб знайти якусь форму слабкості в одному з цих трьох аспектів.

Атака на алгоритм

Кожен криптографічний алгоритм можна зламати, якщо мати достатньо часу на дослідження. Саме тому організації та агентства, такі як Агентство національної безпеки (АНБ), оновлюють стандарти шифрування кожні кілька років. На момент написання цієї статті єдиним симетричним алгоритмом шифрування, який не виявив жодної суттєвої вразливості, є AES. Небезпека полягає в тому, що багато сучасних криптосистем можуть використовувати шифрування до AES. Transport Layer Security (TLS), криптосистема, яка використовується на кожному захищеному (HTTPS) веб-сайті, може додатково (і нерозумно) використовувати попередні AES-шифрування, такі як RC4 або DES. У цьому випадку досвідчені криптографи можуть зламати TLS, атакуючи алгоритм, що лежить в його основі.

Атака на ключ

Враховуючи, що практично кожна криптосистема використовує якусь форму ключа, очевидно, що криптоаналіз буде шукати в ньому слабкі місця або, що більш важливо, намагатися визначити ключ, просто атакуючи зашифрований текст. Однією з найбільших слабкостей є довжина ключа. Сучасне зростання обчислювальних потужностей полегшує злам ключів, які колись вважалися занадто довгими для злому.

Розглянемо велосипедний замок з чотирма колесами проти велосипедного замка, у якого їх набагато більше, ніж чотири (Мал. 2-31). Який з них швидше зламати? Та ж концепція застосовується і до ключів. Якщо взяти шматок зашифрованого тексту і спробувати всі можливі ключі, поки не з'явиться щось схоже на відкритий текст - це атака грубою силою. (Про неї ми поговоримо пізніше).

DES має ефективну довжину ключа всього 56 біт. Це означає, що він має ключовий простір (загальна кількість можливих ключів) 256 або 72 057 594 037 927 936 різних комбінацій. Це число здається неймовірно великою кількістю комбінацій, але спеціалізовані обчислювальні системи, такі як Deep Crack

Федерації електронної свободи, зламали ключ DES за 22 години... в далекому 1998 році. Сьогодні обчислювальна потужність настільного комп'ютера високого класу не надто далека від такої ж потужності.

Атака на реалізацію

Деякі реалізації нових криптосистем мають певний вроджений недолік, який призводить до непередбачуваної вразливості, яка не є виною алгоритму, що лежить в їх основі. Дротовий еквівалент конфіденційності (WEP) був повноцінною криптосистемою шифрування 802.11, представленою в першому поколінні 802.11 WAP в 1999 році. Отримавши широке розповсюдження, вже через рік було виявлено, що WEP погано реалізує спосіб використання базового шифрування RC4. Цей недолік дозволяв за допомогою простого криптоаналізу отримати ключ, зібравши достатньо зашифрованих даних.

Дані, які можна атакувати

Жодна атака на шифр не спрацює, якщо у вас немає доступу до чогось, що можна зламати. Ці дані, які можна зламати, можна розділити на два різних типи: зашифрований текст або хеші. Перш ніж ми почнемо атакувати, давайте спочатку створимо два приклади сценаріїв: один із зашифрованим текстом, а інший з хешами. А потім почнемо атакувати.

Сценарій із зашифрованим текстом

Візьмемо один приклад мережі 802.11, зашифрованої за допомогою Wi-Fi Protected Access Pre-Shared Key (WPA-PSK). Коли клієнт підключається до точки доступу, він проходить через просту процедуру рукостискання, яка визначає, що кожна сторона зберігає однаковий PSK. Цей процес не передає ключ, але містить достатньо інформації, щоб надати ключ досвідченому хакеру. Програмісти розробили програми, які можуть перехоплювати рукостискання WPA. На рисунку 2-32 показано популярну програму Aircrack-ng, яка перехоплює рукостискання.

Після перехоплення цього рукостискання зловмисник отримує значення, до якого він може застосувати 128-бітний ключ. Якщо зловмисник застосує правильний 128-бітний ключ, він зможе визначити PSK.

Сценарій хешування

Багато криптографічних атак спрямовані на злам паролів. Давайте почнемо цей сценарій з огляду того, де, як і навіщо потрібні паролі.

Комп'ютерна система або програма, яка вимагає парольної автентифікації, повинна містити базу даних або таблицю паролів. Оскільки ці таблиці паролів є потенційно вразливими для будь-кого, хто має технічні знання, щоб знайти їх і спробувати зламати, зберігання паролів у вигляді відкритого тексту є небезпечним. Тому дуже поширеною практикою є не зберігання пароля, а зберігання криптографічного хешу пароля користувача в базі даних.

Нижче наведено приклад гіпотетичного списку паролів, що складається з чотирьох імен користувачів з відповідними паролями, збереженими у вигляді хешів MD5. Як ви можете визначити, що це MD5? Просто підрахувавши кількість символів! Цей приклад можна використовувати для багатьох застосувань. У цьому прикладі використовується текстовий файл; список паролів може бути текстовим файлом, двійковим файлом, файлом cookie або параметрами реєстру, залежно від програми або операційної системи, яка використовує паролі.

Mike, bd4c4ea1b44a8ff2afa18dfd261ec2c8

Steve, 4015e9ce43edfb0668ddaa973ebc7e87

Janet, 8cbad96aced40b3838dd9f07f6ef5772

Penny, 48cccca3bab2ad18832233ee8dff1b0b

Тепер, коли у вас є хеші, ви можете хешувати всі можливі концептуальні паролі, щоб побачити, чи зможете ви підібрати хеш.

Сценарії атак

Представлені сценарії атаки на зашифрований текст і хеш мають спільні риси. Обидва покладаються на єдину точку даних: або шифрований текст, або хеш. Обидва мають чітко зрозумілий алгоритм: WPA і MD5. Нарешті, ви можете використовувати атаку, щоб застосувати ключ для отримання відкритого тексту. Але як зловмисники здійснюють цю атаку? Які інструменти вони використовують, щоб прочитати зашифрований текст або хеш і спробувати

витягти відкритий ключ або пароль? Ці атаки можуть здійснювати програми, які називаються зломщиками паролів або інструментами для відновлення паролів.

1.3 Висновки по розділу

1. Подано порядок розподілу інформації на об’єкті інформаційної діяльності. Визначено класифікацію інформації за змістом та порядком доступу. Подано способи та режими доступу до інформації. Визначено інформацію, яка підлягає захисту та подано критерії інформаційної безпеки.

2. Проведено аналіз загроз та визначено канали витоку технічної інформації. Показана структура технічного каналу витоку інформації.

3. Здійснено аналіз технічних каналів витоку інформації та подана їх класифікація. Яка включає розподіл за видом інформаційної діяльності, принципом формування небезпечного сигналу, за середовищем поширення небезпечного сигналу, за способом перехоплення (зняття) небезпечного сигналу засобами технічної розвідки противника.

РОЗДІЛ 2
ТЕХНОЛОГІЇ ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ ПРИ ДОСТУПІ ДО РОБОТИ З ІНФОРМАЦІЄЮ З ОБМЕЖЕНИМ ДОСТУПОМ

2.1 Процес ідентифікації та автентифікації користувачів в інформаційно-комунікаційних мережах

Модуль 2-8: Криптографічні атаки

Цей модуль охоплює наступні цілі CompTIA Security+:

- 1.2 Порівнювати і протиставляти типи атак

- 2.2 Використовувати відповідні програмні засоби для оцінки стану безпеки в організації за певним сценарієм

- 6.1 Порівняйте та порівняйте основні концепції криптографії

Криптоаналіз, наука про злам шифрування, така ж давня, як і саме шифрування. Від Людовика XVI в його палаці у Версалі до останніх атак, спонсорованих державою, шифрування успішно атакували багато разів. У цьому модулі розглядаються деякі з найпоширеніших криптографічних атак та інструменти, що використовуються для їх здійснення. Розуміння цих атак та їхніх інструментів дає змогу краще зрозуміти, як захистити дані. (До того ж, зламувати щось дуже весело!)

Модуль починається із загального обговорення стратегій атак, а потім розглядаються дані, що піддаються атакам. У другому розділі розглядаються атаки, орієнтовані на паролі, та інструменти, які зазвичай використовуються в таких атаках. Модуль закінчується розглядом кількох атак, які не пов'язані з паролями.

Стратегії атак

Давайте почнемо з розгляду того, де можна атакувати будь-яку криптосистему. Кожна криптосистема має базовий алгоритм, ключ (або у випадку з хешами - дайджест), або певну реалізацію самої криптосистеми, і все

це зрозуміло зловмиснику. Всі криптографічні алгоритми відомі і рідко змінюються, відповідно до принципу Керкхоффа. Кожна криптографічна атака починається з ретельної перевірки криптосистеми і агресивного зондування та експериментів, щоб знайти якусь форму слабкості в одному з цих трьох аспектів.

Атака на алгоритм

Кожен криптографічний алгоритм можна зламати, якщо мати достатньо часу на дослідження. Саме тому організації та агентства, такі як Агентство національної безпеки (АНБ), оновлюють стандарти шифрування кожні кілька років. На момент написання цієї статті єдиним симетричним алгоритмом шифрування, який не виявив жодної суттєвої вразливості, є AES. Небезпека полягає в тому, що багато сучасних криптосистем можуть використовувати шифрування до AES. Transport Layer Security (TLS), криптосистема, яка використовується на кожному захищеному (HTTPS) веб-сайті, може додатково (і нерозумно) використовувати попередні AES-шифрування, такі як RC4 або DES. У цьому випадку досвідчені криптографи можуть зламати TLS, атакуючи алгоритм, що лежить в його основі.

Атака на ключ

Враховуючи, що практично кожна криптосистема використовує якусь форму ключа, очевидно, що криптоаналіз буде шукати в ньому слабкі місця або, що більш важливо, намагатися визначити ключ, просто атакуючи зашифрований текст. Однією з найбільших слабкостей є довжина ключа. Сучасне зростання обчислювальних потужностей полегшує злам ключів, які колись вважалися занадто довгими для злому.

Розглянемо велосипедний замок з чотирма колесами проти велосипедного замка, у якого їх набагато більше, ніж чотири (Мал. 2-31). Який з них швидше зламати? Та ж концепція застосовується і до ключів. Якщо взяти шматок зашифрованого тексту і спробувати всі можливі ключі, поки не з'явиться щось схоже на відкритий текст - це атака грубою силою. (Про неї ми поговоримо пізніше).

DES має ефективну довжину ключа всього 56 біт. Це означає, що він має

ключовий простір (загальна кількість можливих ключів) 256 або 72 057 594 037 927 936 різних комбінацій. Це число здається неймовірно великою кількістю комбінацій, але спеціалізовані обчислювальні системи, такі як Deep Crack Федерації електронної свободи, зламали ключ DES за 22 години... в далекому 1998 році. Сьогодні обчислювальна потужність настільного комп'ютера високого класу не надто далека від такої ж потужності.

Атака на реалізацію

Деякі реалізації нових криптосистем мають певний вроджений недолік, який призводить до непередбачуваної вразливості, яка не є виною алгоритму, що лежить в їх основі. Дротовий еквівалент конфіденційності (WEP) був повноцінною криптосистемою шифрування 802.11, представленою в першому поколінні 802.11 WAP в 1999 році. Отримавши широке розповсюдження, вже через рік було виявлено, що WEP погано реалізує спосіб використання базового шифрування RC4. Цей недолік дозволяв за допомогою простого криптоаналізу отримати ключ, зібравши достатньо зашифрованих даних.

Дані, які можна атакувати

Жодна атака на шифр не спрацює, якщо у вас немає доступу до чогось, що можна зламати. Ці дані, які можна зламати, можна розділити на два різних типи: зашифрований текст або хеші. Перш ніж ми почнемо атакувати, давайте спочатку створимо два приклади сценаріїв: один із зашифрованим текстом, а інший з хешами. А потім почнемо атакувати.

Сценарій із зашифрованим текстом

Візьмемо один приклад мережі 802.11, зашифрованої за допомогою Wi-Fi Protected Access Pre-Shared Key (WPA-PSK). Коли клієнт підключається до точки доступу, він проходить через просту процедуру рукостискання, яка визначає, що кожна сторона зберігає однаковий PSK. Цей процес не передає ключ, але містить достатньо інформації, щоб надати ключ досвідченому хакеру. Програмісти розробили програми, які можуть перехоплювати рукостискання WPA. На рисунку 2-32 показано популярну програму Aircrack-ng, яка перехоплює рукостискання.

Після перехоплення цього рукостискання зловмисник отримує значення, до якого він може застосувати 128-бітний ключ. Якщо зловмисник застосує правильний 128-бітний ключ, він зможе визначити PSK.

Сценарій хешування

Багато криптографічних атак спрямовані на злам паролів. Давайте почнемо цей сценарій з огляду того, де, як і навіщо потрібні паролі.

Комп'ютерна система або програма, яка вимагає парольної автентифікації, повинна містити базу даних або таблицю паролів. Оскільки ці таблиці паролів є потенційно вразливими для будь-кого, хто має технічні знання, щоб знайти їх і спробувати зламати, зберігання паролів у вигляді відкритого тексту є небезпечним. Тому дуже поширеною практикою є не зберігання пароля, а зберігання криптографічного хешу пароля користувача в базі даних.

Нижче наведено приклад гіпотетичного списку паролів, що складається з чотирьох імен користувачів з відповідними паролями, збереженими у вигляді хешів MD5. Як ви можете визначити, що це MD5? Просто підрахувавши кількість символів! Цей приклад можна використовувати для багатьох застосувань. У цьому прикладі використовується текстовий файл; список паролів може бути текстовим файлом, двійковим файлом, файлом cookie або параметрами реєстру, залежно від програми або операційної системи, яка використовує паролі.

Mike, bd4c4ea1b44a8ff2afa18dfd261ec2c8

Steve, 4015e9ce43edfb0668ddaa973ebc7e87

Janet, 8cbad96aced40b3838dd9f07f6ef5772

Penny, 48cccca3bab2ad18832233ee8dff1b0b

Тепер, коли у вас є хеші, ви можете хешувати всі можливі концептуальні паролі, щоб побачити, чи зможете ви підібрати хеш.

Сценарії атак

Представлені сценарії атаки на зашифрований текст і хеш мають спільні риси. Обидва покладаються на єдину точку даних: або шифрований текст, або хеш. Обидва мають чітко зрозумілий алгоритм: WPA і MD5. Нарешті, ви можете

використовувати атаку, щоб застосувати ключ для отримання відкритого тексту. Але як зловмисники здійснюють цю атаку? Які інструменти вони використовують, щоб прочитати зашифрований текст або хеш і спробувати витягти відкритий ключ або пароль? Ці атаки можуть здійснювати програми, які називаються зломщиками паролів або інструментами для відновлення паролів. Програми для злому паролів мають широкий спектр функцій та інтерфейсів. Наприклад, зловмисник на базі Linux використовує Jack the Ripper або Hashcat (Мал. 2-33). Зловмисник на базі Windows може використовувати поважний Cain & Abel (Рисунок 2-34).

Загалом, всі ці інструменти працюють за схожими схемами. По-перше, вони перехоплюють зашифрований текст або хеші, які ви хочете атакувати. По-друге, вони аналізують дані, щоб визначити тип шифрування (або вони знають шифрування від джерела). Нарешті, вони дають вам можливість атакувати більшістю, якщо не всіма, способами, перерахованими в цьому модулі.

Давайте розглянемо три найпоширеніші типи атак, які застосовуються до цих сценаріїв на основі паролів:

- Атака грубою силою
- Словникова атака
- Атака веселкових таблиць

Атака грубої сили

Атака грубим перебором - це найпростіший і найменш ефективний тип атаки на відомі зашифровані тексти/хеші. При переборі методом грубого перебору намагаються отримати пароль або ключ шляхом перевірки зашифрованого тексту або хешу, а потім перебирають всі можливі комбінації ключа або хешу до тих пір, поки не вдасться розшифрувати відкритий текст або згенерувати збіг. Складність методу грубої сили полягає в загальній кількості ітерацій, необхідних для перебору всіх можливих комбінацій ключів. Розглянемо перехоплене рукостискання WPA. Ключ 128-бітний, тому він може містити від 128 нулів до 128 одиниць, плюс всі комбінації між ними.

Отже, скільки часу потрібно, щоб перебрати 2128 комбінацій? Чим довше,

тим більше комбінацій і, відповідно, складніше, тому 2128 дорівнює

3.402823669209384634633746074317682114 56 × 1038

Це величезне число, але комп'ютери потужні. Один дуже потужний процесор Intel Core i7 може обробляти півтрильйона операцій з плаваючою комою в секунду (FLOPS), і ви можете легко створити комп'ютер з великою кількістю таких процесорів. Крім того, зломщики паролів

ретельно оптимізовані для використання всіх цих FLOPS. Насправді, кращі зломщики можуть навіть скористатися перевагами потужних графічних процесорів (GPU) на відеокартах, а також спеціальних пристроїв на інтегральних схемах (ASIC), які працюють в сотні або навіть тисячі разів швидше, ніж центральний процесор (Мал. 2-35).

Отже, не витрачаючи обурливо багато тексту на тонкощі процесорів, FLOPS, графічних процесорів тощо, ми можемо стверджувати, що дуже потужна система злому може обробляти один трильйон спроб злому за секунду. Щоб підібрати всі можливі ключі, знадобиться приблизно 10 790 283 070 806 014 188 років. Якщо це здається вам неймовірно довгим часом, то так воно і є. Майте на увазі, що ключ може бути повністю нульовим, що означає, що ви отримаєте ключ WPA з першої спроби. Або ж ключ може складатися з одних одиниць, що означає, що вам доведеться спробувати всі 2128 комбінацій.

Це може наштовхнути вас на думку, що грубий перебір марний, і це для відносно великих ключів. Однак, коротші ключі можна легко зламати. Наприклад, 56-бітний ключ DES можна зламати (використовуючи попередні припущення) приблизно за 20 годин. Зі збільшенням обчислювальної потужності те, що колись було довгим ключем або хешем, стає коротким. Ось чому ключі і хеші з роками стають все довшими і довшими.

Але навіть якщо ми можемо виконати атаку грубої сили за кілька годин, чому б не придумати атаки, які прискорюють цей процес? Як ми можемо зменшити можливу кількість ключів для перебору?

Словникова атака

Найслабшою стороною паролів є те, що їх створюють люди. Люди

створюють такі паролі, як цей:

Password123

Це чудовий пароль, чи не так? Він поєднує великі та малі літери з буквами та цифрами.

Хм... не дуже, коли мова йде про сучасні криптографічні атаки. Люди рідко створюють паролі, які виглядають так:

1x3^^7vEssde6)

Це набагато крутіший пароль, але це не те, що створюють звичайні люди.

Пам'ятаючи про це, ми можемо наказати нашій програмі-зломщику прочитати текстовий файл, заповнений усіма загальновживаними словами (в даному випадку) англійської мови - файл-словник. Потім програма-зломщик буде використовувати цей файл словника замість грубої сили в словниковій атаці. Ось крихітний, крихітний шматочок файлу словника, завантажений з Інтернету:

A

AA

AAA AAAA

a aa aa aaa aaa aaaa

Як бачите, більшість файлів словників містять багато не-слів. Це тому, що не-слова, такі як "AAAA", "QWERTY" або "NASA", часто використовуються для створення паролів. Невеликий файл словника може зберігати лише кілька мільйонів паролів. Найбільші файли словників містять багато мільйонів паролів.

Хороші інструменти для атаки за словниками не перебирають лише паролі зі списку словників. Подивіться налаштування словникової атаки для Cain & Abel на Рисунку 2-36. Зверніть увагу на опції, які дозволяють модифікувати словник. Наприклад, якщо у вашому файлі словника є слово cheese,

Cain & Abel дозволить вам

- Перевернути слово: eseehc

- Подвійний прохід: cheesecheese

- З великої літери на малу: не застосовується, оскільки слово "сир"

пишеться з малої літери

- Малі літери на великі: СИР

- Заміна числа: ch33s3

- Перестановки регістру: Cheese, cHeese і т.д.

- Додавання однієї або двох цифр в кінці: cheese1, cheese44

Ви можете стверджувати, що ці варіанти додають до атаки по словнику аспект грубої сили. Атака по словнику в поєднанні з такими опціями, як перебір, називається гібридною атакою.

Майте на увазі, що існує величезна різниця в якості файлів словників. Я зберігаю набір файлів-словників, зібраних роками з багатьох джерел, відсортованих за довжиною пароля, малими/великими літерами, спеціальними символами, цифрами перед паролем і так далі. Маючи хороші файли-словники, добре розуміючи інструменти та вірячи в лінь людей використовувати погані паролі, я можу зламати майже будь-який офлайн-пароль за кілька годин.

Що робити, якщо у вас немає ніякого зашифрованого тексту для атаки? Що, якщо ви просто хочете спробувати кілька мільйонів паролів, щоб увійти на веб-сайт? Це класична ситуація для онлайн-атаки, і це не проблема! Існує безліч онлайнових програм для злому паролів за допомогою словникової атаки, таких як популярна Hydra (Мал. 2-37). Онлайн-атаки відбуваються повільно, але їх легко розпочати і залишити на кілька днів або тижнів.

Атака за допомогою веселкових таблиць

Складність атаки за допомогою словників, особливо коли ви намагаєтеся зламати хешовані паролі, полягає в тому, що вам потрібно взяти слово зі словника, хешувати його, а потім порівняти хеш з хешем у файлі пароля. Ви можете включити хеш-значення для всіх Рис. 2-37 Гідра в дії записів у файл словника, використовуючи поширені алгоритми хешування, щоб підвищити ефективність атаки по словнику. Хеш-функція SHA-256 для "password" завжди буде такою

5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8

Отже, що, якщо ми відредагуємо наші файли словників, щоб додати хеш?

Звичайно, це може зайняти тижні, щоб згенерувати мільярди хешів і зберегти їх у нашому файлі словника, але коли це буде зроблено, ми матимемо файл, який міститиме слово зі словника, а також хеш. Такий файл словника буде виглядати так:

a, ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb aa, 961b6dd3ede3cb8ecbaacbd68de040cd78eb2ed5889130cceb4c49268ea4d506 aaa, 9834876dcfb05cb167a5c24953eba58c4ac89b1adf57f28f2f9d09af107ee8f0 aaaa, 61be55a8e2f6b4e172338bddf184d6dbee29c98853e0a0485ecee7f27b9af0b4

Використання такого інклюзивного файлу словника може зробити словникову атаку надзвичайно ефективною. Але є одна величезна проблема. Такі файли дуже великі, потенційно в діапазоні сотень петабайт для файлів-словників, які включають дійсно довгі паролі. Вони просто занадто великі для практичного використання (Мал. 2-38).

На допомогу приходить математика! За межами іспиту CompTIA Security+ складна математика може створити райдужну таблицю - значно зменшений за розміром файл словника, який містить хешовані записи.

Веселкові таблиці - це бінарні, а не текстові файли. На Рисунку 2-39 показано вміст одного набору таблиць-веселок, призначених для злому старих файлів паролів диспетчера списків у стилі Windows XP. Веселкові таблиці для більш досконалих операційних систем можуть бути набагато більшими, хоча завжди набагато меншими, ніж традиційний файл словника з хешами.

Веселкові таблиці довгий час були першим вибором для злому паролів; такі атаки швидкі і, як правило, дуже успішні. Веселкові таблиці були настільки популярними, що ви можете знайти онлайн-сервіси, які дозволяють завантажувати хеші для злому, а потужні системи зламають пароль всього за кілька доларів (Рисунок 2-40). (Або ж ви можете використати грубий перебір на графічному процесорі, як це роблять сучасні хакери).

Захист сховища паролів

Традиційні сховища хеш-паролів постійно піддаються атакам грубого перебору, словниковим атакам і атакам "веселки", тому криптографам важливо

придумати методи, які зроблять сховища паролів більш стійкими до цих атак. Солі та розтягування ключів пропонують такі способи захисту.

Модуль 2-8: Криптографічні атаки

Цей модуль охоплює наступні цілі CompTIA Security+:

- 1.2 Порівнювати і протиставляти типи атак

- 2.2 Використовувати відповідні програмні засоби для оцінки стану безпеки в організації за певним сценарієм

- 6.1 Порівняйте та порівняйте основні концепції криптографії

Криптоаналіз, наука про злам шифрування, така ж давня, як і саме шифрування. Від Людовика XVI в його палаці у Версалі до останніх атак, спонсорованих державою, шифрування успішно атакували багато разів. У цьому модулі розглядаються деякі з найпоширеніших криптографічних атак та інструменти, що використовуються для їх здійснення. Розуміння цих атак та їхніх інструментів дає змогу краще зрозуміти, як захистити дані. (До того ж, зламувати щось дуже весело!)

Модуль починається із загального обговорення стратегій атак, а потім розглядаються дані, що піддаються атакам. У другому розділі розглядаються атаки, орієнтовані на паролі, та інструменти, які зазвичай використовуються в таких атаках. Модуль закінчується розглядом кількох атак, які не пов'язані з паролями.

Стратегії атак

Давайте почнемо з розгляду того, де можна атакувати будь-яку криптосистему. Кожна криптосистема має базовий алгоритм, ключ (або у випадку з хешами - дайджест), або певну реалізацію самої криптосистеми, і все це зрозуміло зловмиснику. Всі криптографічні алгоритми відомі і рідко змінюються, відповідно до принципу Керкхоффа. Кожна криптографічна атака починається з ретельної перевірки криптосистеми і агресивного зондування та експериментів, щоб знайти якусь форму слабкості в одному з цих трьох аспектів.

Атака на алгоритм

Кожен криптографічний алгоритм можна зламати, якщо мати достатньо

часу на дослідження. Саме тому організації та агентства, такі як Агентство національної безпеки (АНБ), оновлюють стандарти шифрування кожні кілька років. На момент написання цієї статті єдиним симетричним алгоритмом шифрування, який не виявив жодної суттєвої вразливості, є AES. Небезпека полягає в тому, що багато сучасних криптосистем можуть використовувати шифрування до AES. Transport Layer Security (TLS), криптосистема, яка використовується на кожному захищеному (HTTPS) веб-сайті, може додатково (і нерозумно) використовувати попередні AES-шифрування, такі як RC4 або DES. У цьому випадку досвідчені криптографи можуть зламати TLS, атакуючи алгоритм, що лежить в його основі.

Атака на ключ

Враховуючи, що практично кожна криптосистема використовує якусь форму ключа, очевидно, що криптоаналіз буде шукати в ньому слабкі місця або, що більш важливо, намагатися визначити ключ, просто атакуючи зашифрований текст. Однією з найбільших слабкостей є довжина ключа. Сучасне зростання обчислювальних потужностей полегшує злам ключів, які колись вважалися занадто довгими для злому.

Розглянемо велосипедний замок з чотирма колесами проти велосипедного замка, у якого їх набагато більше, ніж чотири (Мал. 2-31). Який з них швидше зламати? Та ж концепція застосовується і до ключів. Якщо взяти шматок зашифрованого тексту і спробувати всі можливі ключі, поки не з'явиться щось схоже на відкритий текст - це атака грубою силою. (Про неї ми поговоримо пізніше).

DES має ефективну довжину ключа всього 56 біт. Це означає, що він має ключовий простір (загальна кількість можливих ключів) 256 або 72 057 594 037 927 936 різних комбінацій. Це число здається неймовірно великою кількістю комбінацій, але спеціалізовані обчислювальні системи, такі як Deep Crack Федерації електронної свободи, зламали ключ DES за 22 години... в далекому 1998 році. Сьогодні обчислювальна потужність настільного комп'ютера високого класу не надто далека від такої ж потужності.

Атака на реалізацію

Деякі реалізації нових криптосистем мають певний вроджений недолік, який призводить до непередбачуваної вразливості, яка не є виною алгоритму, що лежить в їх основі. Дротовий еквівалент конфіденційності (WEP) був повноцінною криптосистемою шифрування 802.11, представленою в першому поколінні 802.11 WAP в 1999 році. Отримавши широке розповсюдження, вже через рік було виявлено, що WEP погано реалізує спосіб використання базового шифрування RC4. Цей недолік дозволяв за допомогою простого криптоаналізу отримати ключ, зібравши достатньо зашифрованих даних.

Дані, які можна атакувати

Жодна атака на шифр не спрацює, якщо у вас немає доступу до чогось, що можна зламати. Ці дані, які можна зламати, можна розділити на два різних типи: зашифрований текст або хеші. Перш ніж ми почнемо атакувати, давайте спочатку створимо два приклади сценаріїв: один із зашифрованим текстом, а інший з хешами. А потім почнемо атакувати.

Сценарій із зашифрованим текстом

Візьмемо один приклад мережі 802.11, зашифрованої за допомогою Wi-Fi Protected Access Pre-Shared Key (WPA-PSK). Коли клієнт підключається до точки доступу, він проходить через просту процедуру рукостискання, яка визначає, що кожна сторона зберігає однаковий PSK. Цей процес не передає ключ, але містить достатньо інформації, щоб надати ключ досвідченому хакеру. Програмісти розробили програми, які можуть перехоплювати рукостискання WPA. На рисунку 2-32 показано популярну програму Aircrack-ng, яка перехоплює рукостискання.

Після перехоплення цього рукостискання зловмисник отримує значення, до якого він може застосувати 128-бітний ключ. Якщо зловмисник застосує правильний 128-бітний ключ, він зможе визначити PSK.

Сценарій хешування

Багато криптографічних атак спрямовані на злам паролів. Давайте почнемо цей сценарій з огляду того, де, як і навіщо потрібні паролі.

Комп'ютерна система або програма, яка вимагає парольної автентифікації, повинна містити базу даних або таблицю паролів. Оскільки ці таблиці паролів є потенційно вразливими для будь-кого, хто має технічні знання, щоб знайти їх і спробувати зламати, зберігання паролів у вигляді відкритого тексту є небезпечним. Тому дуже поширеною практикою є не зберігання пароля, а зберігання криптографічного хешу пароля користувача в базі даних.

Нижче наведено приклад гіпотетичного списку паролів, що складається з чотирьох імен користувачів з відповідними паролями, збереженими у вигляді хешів MD5. Як ви можете визначити, що це MD5? Просто підрахувавши кількість символів! Цей приклад можна використовувати для багатьох застосувань. У цьому прикладі використовується текстовий файл; список паролів може бути текстовим файлом, двійковим файлом, файлом cookie або параметрами реєстру, залежно від програми або операційної системи, яка використовує паролі.

Mike, bd4c4ea1b44a8ff2afa18dfd261ec2c8

Steve, 4015e9ce43edfb0668ddaa973ebc7e87

Janet, 8cbad96aced40b3838dd9f07f6ef5772

Penny, 48cccca3bab2ad18832233ee8dff1b0b

Тепер, коли у вас є хеші, ви можете хешувати всі можливі концептуальні паролі, щоб побачити, чи зможете ви підібрати хеш.

Сценарії атак

Представлені сценарії атаки на зашифрований текст і хеш мають спільні риси. Обидва покладаються на єдину точку даних: або шифрований текст, або хеш. Обидва мають чітко зрозумілий алгоритм: WPA і MD5. Нарешті, ви можете використовувати атаку, щоб застосувати ключ для отримання відкритого тексту. Але як зловмисники здійснюють цю атаку? Які інструменти вони використовують, щоб прочитати зашифрований текст або хеш і спробувати витягти відкритий ключ або пароль? Ці атаки можуть здійснювати програми, які називаються зломщиками паролів або інструментами для відновлення паролів. Програми для злому паролів мають широкий спектр функцій та інтерфейсів.

Наприклад, зловмисник на базі Linux використовує Jack the Ripper або Hashcat (Мал. 2-33). Зловмисник на базі Windows може використовувати поважний Cain & Abel (Рисунок 2-34).

Загалом, всі ці інструменти працюють за схожими схемами. По-перше, вони перехоплюють зашифрований текст або хеші, які ви хочете атакувати. По-друге, вони аналізують дані, щоб визначити тип шифрування (або вони знають шифрування від джерела). Нарешті, вони дають вам можливість атакувати більшістю, якщо не всіма, способами, перерахованими в цьому модулі.

Давайте розглянемо три найпоширеніші типи атак, які застосовуються до цих сценаріїв на основі паролів:

- Атака грубою силою
- Словникова атака
- Атака веселкових таблиць

Атака грубої сили

Атака грубим перебором - це найпростіший і найменш ефективний тип атаки на відомі зашифровані тексти/хеші. При переборі методом грубого перебору намагаються отримати пароль або ключ шляхом перевірки зашифрованого тексту або хешу, а потім перебирають всі можливі комбінації ключа або хешу до тих пір, поки не вдасться розшифрувати відкритий текст або згенерувати збіг. Складність методу грубої сили полягає в загальній кількості ітерацій, необхідних для перебору всіх можливих комбінацій ключів. Розглянемо перехоплене рукостискання WPA. Ключ 128-бітний, тому він може містити від 128 нулів до 128 одиниць, плюс всі комбінації між ними.

Отже, скільки часу потрібно, щоб перебрати 2128 комбінацій? Чим довше, тим більше комбінацій і, відповідно, складніше, тому 2128 дорівнює

3.402823669209384634633746074317682114 56 × 1038

Це величезне число, але комп'ютери потужні. Один дуже потужний процесор Intel Core i7 може обробляти півтрильйона операцій з плаваючою комою в секунду (FLOPS), і ви можете легко створити комп'ютер з великою кількістю таких процесорів. Крім того, зломщики паролів

ретельно оптимізовані для використання всіх цих FLOPS. Насправді, кращі зломщики можуть навіть скористатися перевагами потужних графічних процесорів (GPU) на відеокартах, а також спеціальних пристроїв на інтегральних схемах (ASIC), які працюють в сотні або навіть тисячі разів швидше, ніж центральний процесор (Мал. 2-35).

Отже, не витрачаючи обурливо багато тексту на тонкощі процесорів, FLOPS, графічних процесорів тощо, ми можемо стверджувати, що дуже потужна система злому може обробляти один трильйон спроб злому за секунду. Щоб підібрати всі можливі ключі, знадобиться приблизно 10 790 283 070 806 014 188 років. Якщо це здається вам неймовірно довгим часом, то так воно і є. Майте на увазі, що ключ може бути повністю нульовим, що означає, що ви отримаєте ключ WPA з першої спроби. Або ж ключ може складатися з одних одиниць, що означає, що вам доведеться спробувати всі 2128 комбінацій.

Це може наштовхнути вас на думку, що грубий перебір марний, і це для відносно великих ключів. Однак, коротші ключі можна легко зламати. Наприклад, 56-бітний ключ DES можна зламати (використовуючи попередні припущення) приблизно за 20 годин. Зі збільшенням обчислювальної потужності те, що колись було довгим ключем або хешем, стає коротким. Ось чому ключі і хеші з роками стають все довшими і довшими.

Але навіть якщо ми можемо виконати атаку грубої сили за кілька годин, чому б не придумати атаки, які прискорюють цей процес? Як ми можемо зменшити можливу кількість ключів для перебору?

Словникова атака

Найслабшою стороною паролів є те, що їх створюють люди. Люди створюють такі паролі, як цей:

Password123

Це чудовий пароль, чи не так? Він поєднує великі та малі літери з буквами та цифрами.

Хм... не дуже, коли мова йде про сучасні криптографічні атаки. Люди рідко створюють паролі, які виглядають так:

1x3^^7vEssde6)

Це набагато крутіший пароль, але це не те, що створюють звичайні люди.

Пам'ятаючи про це, ми можемо наказати нашій програмі-зломщику прочитати текстовий файл, заповнений усіма загальновживаними словами (в даному випадку) англійської мови - файл-словник. Потім програма-зломщик буде використовувати цей файл словника замість грубої сили в словниковій атаці. Ось крихітний, крихітний шматочок файлу словника, завантажений з Інтернету:

A

AA

AAA AAAA

a aa aa aaa aaa aaaa

Як бачите, більшість файлів словників містять багато не-слів. Це тому, що не-слова, такі як "AAAA", "QWERTY" або "NASA", часто використовуються для створення паролів. Невеликий файл словника може зберігати лише кілька мільйонів паролів. Найбільші файли словників містять багато мільйонів паролів.

Хороші інструменти для атаки за словниками не перебирають лише паролі зі списку словників. Подивіться налаштування словникової атаки для Cain & Abel на Рисунку 2-36. Зверніть увагу на опції, які дозволяють модифікувати словник. Наприклад, якщо у вашому файлі словника є слово cheese,

Cain & Abel дозволить вам

- Перевернути слово: eseehc

- Подвійний прохід: cheesecheese

- З великої літери на малу: не застосовується, оскільки слово "сир" пишеться з малої літери

- Малі літери на великі: СИР

- Заміна числа: ch33s3

- Перестановки регістру: Cheese, cHeese і т.д.

- Додавання однієї або двох цифр в кінці: cheese1, cheese44

Ви можете стверджувати, що ці варіанти додають до атаки по словнику

аспект грубої сили. Атака по словнику в поєднанні з такими опціями, як перебір, називається гібридною атакою.

Майте на увазі, що існує величезна різниця в якості файлів словників. Я зберігаю набір файлів-словників, зібраних роками з багатьох джерел, відсортованих за довжиною пароля, малими/великими літерами, спеціальними символами, цифрами перед паролем і так далі. Маючи хороші файли-словники, добре розуміючи інструменти та вірячи в лінь людей використовувати погані паролі, я можу зламати майже будь-який офлайн-пароль за кілька годин.

Що робити, якщо у вас немає ніякого зашифрованого тексту для атаки? Що, якщо ви просто хочете спробувати кілька мільйонів паролів, щоб увійти на веб-сайт? Це класична ситуація для онлайн-атаки, і це не проблема! Існує безліч онлайнових програм для злому паролів за допомогою словникової атаки, таких як популярна Hydra (Мал. 2-37). Онлайн-атаки відбуваються повільно, але їх легко розпочати і залишити на кілька днів або тижнів.

Атака за допомогою веселкових таблиць

Складність атаки за допомогою словників, особливо коли ви намагаєтеся зламати хешовані паролі, полягає в тому, що вам потрібно взяти слово зі словника, хешувати його, а потім порівняти хеш з хешем у файлі пароля. Ви можете включити хеш-значення для всіх Рис. 2-37 Гідра в дії записів у файл словника, використовуючи поширені алгоритми хешування, щоб підвищити ефективність атаки по словнику. Хеш-функція SHA-256 для "password" завжди буде такою

5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8

Отже, що, якщо ми відредагуємо наші файли словників, щоб додати хеш? Звичайно, це може зайняти тижні, щоб згенерувати мільярди хешів і зберегти їх у нашому файлі словника, але коли це буде зроблено, ми матимемо файл, який міститиме слово зі словника, а також хеш. Такий файл словника буде виглядати так:

a, ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb aa, 961b6dd3ede3cb8ecbaacbd68de040cd78eb2ed5889130cceb4c49268ea4d506 aaa,

9834876dcfb05cb167a5c24953eba58c4ac89b1adf57f28f2f9d09af107ee8f0 aaaa, 61be55a8e2f6b4e172338bddf184d6dbee29c98853e0a0485ecee7f27b9af0b4

Використання такого інклюзивного файлу словника може зробити словникову атаку надзвичайно ефективною. Але є одна величезна проблема. Такі файли дуже великі, потенційно в діапазоні сотень петабайт для файлів-словників, які включають дійсно довгі паролі. Вони просто занадто великі для практичного використання (Мал. 2-38).

На допомогу приходить математика! За межами іспиту CompTIA Security+ складна математика може створити райдужну таблицю - значно зменшений за розміром файл словника, який містить хешовані записи.

Веселкові таблиці - це бінарні, а не текстові файли. На Рисунку 2-39 показано вміст одного набору таблиць-веселок, призначених для злому старих файлів паролів диспетчера списків у стилі Windows XP. Веселкові таблиці для більш досконалих операційних систем можуть бути набагато більшими, хоча завжди набагато меншими, ніж традиційний файл словника з хешами.

Веселкові таблиці довгий час були першим вибором для злому паролів; такі атаки швидкі і, як правило, дуже успішні. Веселкові таблиці були настільки популярними, що ви можете знайти онлайн-сервіси, які дозволяють завантажувати хеші для злому, а потужні системи зламають пароль всього за кілька доларів (Рисунок 2-40). (Або ж ви можете використати грубий перебір на графічному процесорі, як це роблять сучасні хакери).

Захист сховища паролів

Традиційні сховища хеш-паролів постійно піддаються атакам грубого перебору, словниковим атакам і атакам "веселки", тому криптографам важливо придумати методи, які зроблять сховища паролів більш стійкими до цих атак. Солі та розтягування ключів пропонують такі способи захисту.

Модуль 2-8: Криптографічні атаки

Цей модуль охоплює наступні цілі CompTIA Security+:

- 1.2 Порівнювати і протиставляти типи атак
- 2.2 Використовувати відповідні програмні засоби для оцінки стану

безпеки в організації за певним сценарієм

- 6.1 Порівняйте та порівняйте основні концепції криптографії

Криптоаналіз, наука про злам шифрування, така ж давня, як і саме шифрування. Від Людовика XVI в його палаці у Версалі до останніх атак, спонсорованих державою, шифрування успішно атакували багато разів. У цьому модулі розглядаються деякі з найпоширеніших криптографічних атак та інструменти, що використовуються для їх здійснення. Розуміння цих атак та їхніх інструментів дає змогу краще зрозуміти, як захистити дані. (До того ж, зламувати щось дуже весело!)

Модуль починається із загального обговорення стратегій атак, а потім розглядаються дані, що піддаються атакам. У другому розділі розглядаються атаки, орієнтовані на паролі, та інструменти, які зазвичай використовуються в таких атаках. Модуль закінчується розглядом кількох атак, які не пов'язані з паролями.

Стратегії атак

Давайте почнемо з розгляду того, де можна атакувати будь-яку криптосистему. Кожна криптосистема має базовий алгоритм, ключ (або у випадку з хешами - дайджест), або певну реалізацію самої криптосистеми, і все це зрозуміло зловмиснику. Всі криптографічні алгоритми відомі і рідко змінюються, відповідно до принципу Керкхоффа. Кожна криптографічна атака починається з ретельної перевірки криптосистеми і агресивного зондування та експериментів, щоб знайти якусь форму слабкості в одному з цих трьох аспектів.

Атака на алгоритм

Кожен криптографічний алгоритм можна зламати, якщо мати достатньо часу на дослідження. Саме тому організації та агентства, такі як Агентство національної безпеки (АНБ), оновлюють стандарти шифрування кожні кілька років. На момент написання цієї статті єдиним симетричним алгоритмом шифрування, який не виявив жодної суттєвої вразливості, є AES. Небезпека полягає в тому, що багато сучасних криптосистем можуть використовувати шифрування до AES. Transport Layer Security (TLS), криптосистема, яка

використовується на кожному захищеному (HTTPS) веб-сайті, може додатково (і нерозумно) використовувати попередні AES-шифрування, такі як RC4 або DES. У цьому випадку досвідчені криптографи можуть зламати TLS, атакуючи алгоритм, що лежить в його основі.

Атака на ключ

Враховуючи, що практично кожна криптосистема використовує якусь форму ключа, очевидно, що криптоаналіз буде шукати в ньому слабкі місця або, що більш важливо, намагатися визначити ключ, просто атакуючи зашифрований текст. Однією з найбільших слабкостей є довжина ключа. Сучасне зростання обчислювальних потужностей полегшує злам ключів, які колись вважалися занадто довгими для злому.

Розглянемо велосипедний замок з чотирма колесами проти велосипедного замка, у якого їх набагато більше, ніж чотири (Мал. 2-31). Який з них швидше зламати? Та ж концепція застосовується і до ключів. Якщо взяти шматок зашифрованого тексту і спробувати всі можливі ключі, поки не з'явиться щось схоже на відкритий текст - це атака грубою силою. (Про неї ми поговоримо пізніше).

DES має ефективну довжину ключа всього 56 біт. Це означає, що він має ключовий простір (загальна кількість можливих ключів) 256 або 72 057 594 037 927 936 різних комбінацій. Це число здається неймовірно великою кількістю комбінацій, але спеціалізовані обчислювальні системи, такі як Deep Crack Федерації електронної свободи, зламали ключ DES за 22 години... в далекому 1998 році. Сьогодні обчислювальна потужність настільного комп'ютера високого класу не надто далека від такої ж потужності.

Атака на реалізацію

Деякі реалізації нових криптосистем мають певний вроджений недолік, який призводить до непередбачуваної вразливості, яка не є виною алгоритму, що лежить в їх основі. Дротовий еквівалент конфіденційності (WEP) був повноцінною криптосистемою шифрування 802.11, представленою в першому поколінні 802.11 WAP в 1999 році. Отримавши широке розповсюдження, вже

через рік було виявлено, що WEP погано реалізує спосіб використання базового шифрування RC4. Цей недолік дозволяв за допомогою простого криптоаналізу отримати ключ, зібравши достатньо зашифрованих даних.

Дані, які можна атакувати

Жодна атака на шифр не спрацює, якщо у вас немає доступу до чогось, що можна зламати. Ці дані, які можна зламати, можна розділити на два різних типи: зашифрований текст або хеші. Перш ніж ми почнемо атакувати, давайте спочатку створимо два приклади сценаріїв: один із зашифрованим текстом, а інший з хешами. А потім почнемо атакувати.

Сценарій із зашифрованим текстом

Візьмемо один приклад мережі 802.11, зашифрованої за допомогою Wi-Fi Protected Access Pre-Shared Key (WPA-PSK). Коли клієнт підключається до точки доступу, він проходить через просту процедуру рукостискання, яка визначає, що кожна сторона зберігає однаковий PSK. Цей процес не передає ключ, але містить достатньо інформації, щоб надати ключ досвідченому хакеру. Програмісти розробили програми, які можуть перехоплювати рукостискання WPA. На рисунку 2-32 показано популярну програму Aircrack-ng, яка перехоплює рукостискання.

Після перехоплення цього рукостискання зловмисник отримує значення, до якого він може застосувати 128-бітний ключ. Якщо зловмисник застосує правильний 128-бітний ключ, він зможе визначити PSK.

Сценарій хешування

Багато криптографічних атак спрямовані на злам паролів. Давайте почнемо цей сценарій з огляду того, де, як і навіщо потрібні паролі.

Комп'ютерна система або програма, яка вимагає парольної автентифікації, повинна містити базу даних або таблицю паролів. Оскільки ці таблиці паролів є потенційно вразливими для будь-кого, хто має технічні знання, щоб знайти їх і спробувати зламати, зберігання паролів у вигляді відкритого тексту є небезпечним. Тому дуже поширеною практикою є не зберігання пароля, а зберігання криптографічного хешу пароля користувача в базі даних.

2.3 Вимоги та порядок організації процесів ідентифікації та автентифікації

Модуль 2-8: Криптографічні атаки

Цей модуль охоплює наступні цілі CompTIA Security+:

- 1.2 Порівнювати і протиставляти типи атак

- 2.2 Використовувати відповідні програмні засоби для оцінки стану безпеки в організації за певним сценарієм

- 6.1 Порівняйте та порівняйте основні концепції криптографії

Криптоаналіз, наука про злам шифрування, така ж давня, як і саме шифрування. Від Людовика XVI в його палаці у Версалі до останніх атак, спонсорованих державою, шифрування успішно атакували багато разів. У цьому модулі розглядаються деякі з найпоширеніших криптографічних атак та інструменти, що використовуються для їх здійснення. Розуміння цих атак та їхніх інструментів дає змогу краще зрозуміти, як захистити дані. (До того ж, зламувати щось дуже весело!)

Модуль починається із загального обговорення стратегій атак, а потім розглядаються дані, що піддаються атакам. У другому розділі розглядаються атаки, орієнтовані на паролі, та інструменти, які зазвичай використовуються в таких атаках. Модуль закінчується розглядом кількох атак, які не пов'язані з паролями.

Стратегії атак

Давайте почнемо з розгляду того, де можна атакувати будь-яку криптосистему. Кожна криптосистема має базовий алгоритм, ключ (або у випадку з хешами - дайджест), або певну реалізацію самої криптосистеми, і все це зрозуміло зловмиснику. Всі криптографічні алгоритми відомі і рідко змінюються, відповідно до принципу Керкхоффа. Кожна криптографічна атака починається з ретельної перевірки криптосистеми і агресивного зондування та експериментів, щоб знайти якусь форму слабкості в одному з цих трьох аспектів.

Атака на алгоритм

Кожен криптографічний алгоритм можна зламати, якщо мати достатньо часу на дослідження. Саме тому організації та агентства, такі як Агентство національної безпеки (АНБ), оновлюють стандарти шифрування кожні кілька років. На момент написання цієї статті єдиним симетричним алгоритмом шифрування, який не виявив жодної суттєвої вразливості, є AES. Небезпека полягає в тому, що багато сучасних криптосистем можуть використовувати шифрування до AES. Transport Layer Security (TLS), криптосистема, яка використовується на кожному захищеному (HTTPS) веб-сайті, може додатково (і нерозумно) використовувати попередні AES-шифрування, такі як RC4 або DES. У цьому випадку досвідчені криптографи можуть зламати TLS, атакуючи алгоритм, що лежить в його основі.

Атака на ключ

Враховуючи, що практично кожна криптосистема використовує якусь форму ключа, очевидно, що криптоаналіз буде шукати в ньому слабкі місця або, що більш важливо, намагатися визначити ключ, просто атакуючи зашифрований текст. Однією з найбільших слабкостей є довжина ключа. Сучасне зростання обчислювальних потужностей полегшує злам ключів, які колись вважалися занадто довгими для злому.

Розглянемо велосипедний замок з чотирма колесами проти велосипедного замка, у якого їх набагато більше, ніж чотири (Мал. 2-31). Який з них швидше зламати? Та ж концепція застосовується і до ключів. Якщо взяти шматок зашифрованого тексту і спробувати всі можливі ключі, поки не з'явиться щось схоже на відкритий текст - це атака грубою силою. (Про неї ми поговоримо пізніше).

DES має ефективну довжину ключа всього 56 біт. Це означає, що він має ключовий простір (загальна кількість можливих ключів) 256 або 72 057 594 037 927 936 різних комбінацій. Це число здається неймовірно великою кількістю комбінацій, але спеціалізовані обчислювальні системи, такі як Deep Crack Федерації електронної свободи, зламали ключ DES за 22 години... в далекому

1998 році. Сьогодні обчислювальна потужність настільного комп'ютера високого класу не надто далека від такої ж потужності.

Атака на реалізацію

Деякі реалізації нових криптосистем мають певний вроджений недолік, який призводить до непередбачуваної вразливості, яка не є виною алгоритму, що лежить в їх основі. Дротовий еквівалент конфіденційності (WEP) був повноцінною криптосистемою шифрування 802.11, представленою в першому поколінні 802.11 WAP в 1999 році. Отримавши широке розповсюдження, вже через рік було виявлено, що WEP погано реалізує спосіб використання базового шифрування RC4. Цей недолік дозволяв за допомогою простого криптоаналізу отримати ключ, зібравши достатньо зашифрованих даних.

Дані, які можна атакувати

Жодна атака на шифр не спрацює, якщо у вас немає доступу до чогось, що можна зламати. Ці дані, які можна зламати, можна розділити на два різних типи: зашифрований текст або хеші. Перш ніж ми почнемо атакувати, давайте спочатку створимо два приклади сценаріїв: один із зашифрованим текстом, а інший з хешами. А потім почнемо атакувати.

Сценарій із зашифрованим текстом

Візьмемо один приклад мережі 802.11, зашифрованої за допомогою Wi-Fi Protected Access Pre-Shared Key (WPA-PSK). Коли клієнт підключається до точки доступу, він проходить через просту процедуру рукостискання, яка визначає, що кожна сторона зберігає однаковий PSK. Цей процес не передає ключ, але містить достатньо інформації, щоб надати ключ досвідченому хакеру. Програмісти розробили програми, які можуть перехоплювати рукостискання WPA. На рисунку 2-32 показано популярну програму Aircrack-ng, яка перехоплює рукостискання.

Після перехоплення цього рукостискання зловмисник отримує значення, до якого він може застосувати 128-бітний ключ. Якщо зловмисник застосує правильний 128-бітний ключ, він зможе визначити PSK.

Сценарій хешування

Багато криптографічних атак спрямовані на злам паролів. Давайте почнемо цей сценарій з огляду того, де, як і навіщо потрібні паролі.

Комп'ютерна система або програма, яка вимагає парольної автентифікації, повинна містити базу даних або таблицю паролів. Оскільки ці таблиці паролів є потенційно вразливими для будь-кого, хто має технічні знання, щоб знайти їх і спробувати зламати, зберігання паролів у вигляді відкритого тексту є небезпечним. Тому дуже поширеною практикою є не зберігання пароля, а зберігання криптографічного хешу пароля користувача в базі даних.

Нижче наведено приклад гіпотетичного списку паролів, що складається з чотирьох імен користувачів з відповідними паролями, збереженими у вигляді хешів MD5. Як ви можете визначити, що це MD5? Просто підрахувавши кількість символів! Цей приклад можна використовувати для багатьох застосувань. У цьому прикладі використовується текстовий файл; список паролів може бути текстовим файлом, двійковим файлом, файлом cookie або параметрами реєстру, залежно від програми або операційної системи, яка використовує паролі.

Mike, bd4c4ea1b44a8ff2afa18dfd261ec2c8

Steve, 4015e9ce43edfb0668ddaa973ebc7e87

Janet, 8cbad96aced40b3838dd9f07f6ef5772

Penny, 48cccca3bab2ad18832233ee8dff1b0b

Тепер, коли у вас є хеші, ви можете хешувати всі можливі концептуальні паролі, щоб побачити, чи зможете ви підібрати хеш.

Сценарії атак

Представлені сценарії атаки на зашифрований текст і хеш мають спільні риси. Обидва покладаються на єдину точку даних: або шифрований текст, або хеш. Обидва мають чітко зрозумілий алгоритм: WPA і MD5. Нарешті, ви можете використовувати атаку, щоб застосувати ключ для отримання відкритого тексту. Але як зловмисники здійснюють цю атаку? Які інструменти вони використовують, щоб прочитати зашифрований текст або хеш і спробувати витягти відкритий ключ або пароль? Ці атаки можуть здійснювати програми, які

називаються зломщиками паролів або інструментами для відновлення паролів. Програми для злому паролів мають широкий спектр функцій та інтерфейсів. Наприклад, зловмисник на базі Linux використовує Jack the Ripper або Hashcat (Мал. 2-33). Зловмисник на базі Windows може використовувати поважний Cain & Abel (Рисунок 2-34).

Загалом, всі ці інструменти працюють за схожими схемами. По-перше, вони перехоплюють зашифрований текст або хеші, які ви хочете атакувати. По-друге, вони аналізують дані, щоб визначити тип шифрування (або вони знають шифрування від джерела). Нарешті, вони дають вам можливість атакувати більшістю, якщо не всіма, способами, перерахованими в цьому модулі.

Давайте розглянемо три найпоширеніші типи атак, які застосовуються до цих сценаріїв на основі паролів:

- Атака грубою силою
- Словникова атака
- Атака веселкових таблиць

Атака грубої сили

Атака грубим перебором - це найпростіший і найменш ефективний тип атаки на відомі зашифровані тексти/хеші. При переборі методом грубого перебору намагаються отримати пароль або ключ шляхом перевірки зашифрованого тексту або хешу, а потім перебирають всі можливі комбінації ключа або хешу до тих пір, поки не вдасться розшифрувати відкритий текст або згенерувати збіг. Складність методу грубої сили полягає в загальній кількості ітерацій, необхідних для перебору всіх можливих комбінацій ключів. Розглянемо перехоплене рукостискання WPA. Ключ 128-бітний, тому він може містити від 128 нулів до 128 одиниць, плюс всі комбінації між ними.

Отже, скільки часу потрібно, щоб перебрати 2128 комбінацій? Чим довше, тим більше комбінацій і, відповідно, складніше, тому 2128 дорівнює

3.402823669209384634633746074317682114 56 × 1038

Це величезне число, але комп'ютери потужні. Один дуже потужний процесор Intel Core i7 може обробляти півтрильйона операцій з плаваючою

2.4 Висновки по розділу

1. Проведено огляд існуючих технологій ідентифікації та автентифікації користувача в інформаційно-комунікаційних системах об’єкту інформаційної діяльності. Визначені основні поняття та функції доступу до інформації з обмеженим доступом. Подана система технологій ідентифікації та автентифікації користувачів.

2. Розглянуто зміст, особливості застосування, переваги та недоліки парольної, апаратної, біометричної ідентифікації та автентифікації користувачів в інформаційно - комунікаційних системах.

4. Подано вимоги та порядок організації процесів ідентифікації та автентифікації.

РОЗДІЛ 3
РОЗРОБКА МЕХАНІЗМІВ БАГАТОФАКТОРНОЇ ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ ПРИ ДОСТУПІ ДО РОБОТИ З ІНФОРМАЦІЄЮ З ОБМЕЖЕНИМ ДОСТУПОМ

3.1 Зміст, характеристики та вимоги до механізму ідентифікації та автентифікації користувачів в інформаційно-комунікаційних мережах

Модуль 2-8: Криптографічні атаки

Цей модуль охоплює наступні цілі CompTIA Security+:

- 1.2 Порівнювати і протиставляти типи атак

- 2.2 Використовувати відповідні програмні засоби для оцінки стану безпеки в організації за певним сценарієм

- 6.1 Порівняйте та порівняйте основні концепції криптографії

Криптоаналіз, наука про злам шифрування, така ж давня, як і саме шифрування. Від Людовика XVI в його палаці у Версалі до останніх атак, спонсорованих державою, шифрування успішно атакували багато разів. У цьому модулі розглядаються деякі з найпоширеніших криптографічних атак та інструменти, що використовуються для їх здійснення. Розуміння цих атак та їхніх інструментів дає змогу краще зрозуміти, як захистити дані. (До того ж, зламувати щось дуже весело!)

Модуль починається із загального обговорення стратегій атак, а потім розглядаються дані, що піддаються атакам. У другому розділі розглядаються атаки, орієнтовані на паролі, та інструменти, які зазвичай використовуються в таких атаках. Модуль закінчується розглядом кількох атак, які не пов'язані з паролями.

Стратегії атак

Давайте почнемо з розгляду того, де можна атакувати будь-яку криптосистему. Кожна криптосистема має базовий алгоритм, ключ (або у випадку з хешами - дайджест), або певну реалізацію самої криптосистеми, і все це зрозуміло зловмиснику. Всі криптографічні алгоритми відомі і рідко

змінюються, відповідно до принципу Керкхоффа. Кожна криптографічна атака починається з ретельної перевірки криптосистеми і агресивного зондування та експериментів, щоб знайти якусь форму слабкості в одному з цих трьох аспектів.

Атака на алгоритм

Кожен криптографічний алгоритм можна зламати, якщо мати достатньо часу на дослідження. Саме тому організації та агентства, такі як Агентство національної безпеки (АНБ), оновлюють стандарти шифрування кожні кілька років. На момент написання цієї статті єдиним симетричним алгоритмом шифрування, який не виявив жодної суттєвої вразливості, є AES. Небезпека полягає в тому, що багато сучасних криптосистем можуть використовувати шифрування до AES. Transport Layer Security (TLS), криптосистема, яка використовується на кожному захищеному (HTTPS) веб-сайті, може додатково (і нерозумно) використовувати попередні AES-шифрування, такі як RC4 або DES. У цьому випадку досвідчені криптографи можуть зламати TLS, атакуючи алгоритм, що лежить в його основі.

Атака на ключ

Враховуючи, що практично кожна криптосистема використовує якусь форму ключа, очевидно, що криптоаналіз буде шукати в ньому слабкі місця або, що більш важливо, намагатися визначити ключ, просто атакуючи зашифрований текст. Однією з найбільших слабкостей є довжина ключа. Сучасне зростання обчислювальних потужностей полегшує злам ключів, які колись вважалися занадто довгими для злому.

Розглянемо велосипедний замок з чотирма колесами проти велосипедного замка, у якого їх набагато більше, ніж чотири (Мал. 2-31). Який з них швидше зламати? Та ж концепція застосовується і до ключів. Якщо взяти шматок зашифрованого тексту і спробувати всі можливі ключі, поки не з'явиться щось схоже на відкритий текст - це атака грубою силою. (Про неї ми поговоримо пізніше).

DES має ефективну довжину ключа всього 56 біт. Це означає, що він має ключовий простір (загальна кількість можливих ключів) 256 або 72 057 594 037

927 936 різних комбінацій. Це число здається неймовірно великою кількістю комбінацій, але спеціалізовані обчислювальні системи, такі як Deep Crack Федерації електронної свободи, зламали ключ DES за 22 години... в далекому 1998 році. Сьогодні обчислювальна потужність настільного комп'ютера високого класу не надто далека від такої ж потужності.

Атака на реалізацію

Деякі реалізації нових криптосистем мають певний вроджений недолік, який призводить до непередбачуваної вразливості, яка не є виною алгоритму, що лежить в їх основі. Дротовий еквівалент конфіденційності (WEP) був повноцінною криптосистемою шифрування 802.11, представленою в першому поколінні 802.11 WAP в 1999 році. Отримавши широке розповсюдження, вже через рік було виявлено, що WEP погано реалізує спосіб використання базового шифрування RC4. Цей недолік дозволяв за допомогою простого криптоаналізу отримати ключ, зібравши достатньо зашифрованих даних.

Дані, які можна атакувати

Жодна атака на шифр не спрацює, якщо у вас немає доступу до чогось, що можна зламати. Ці дані, які можна зламати, можна розділити на два різних типи: зашифрований текст або хеші. Перш ніж ми почнемо атакувати, давайте спочатку створимо два приклади сценаріїв: один із зашифрованим текстом, а інший з хешами. А потім почнемо атакувати.

Сценарій із зашифрованим текстом

Візьмемо один приклад мережі 802.11, зашифрованої за допомогою Wi-Fi Protected Access Pre-Shared Key (WPA-PSK). Коли клієнт підключається до точки доступу, він проходить через просту процедуру рукостискання, яка визначає, що кожна сторона зберігає однаковий PSK. Цей процес не передає ключ, але містить достатньо інформації, щоб надати ключ досвідченому хакеру. Програмісти розробили програми, які можуть перехоплювати рукостискання WPA. На рисунку 2-32 показано популярну програму Aircrack-ng, яка перехоплює рукостискання.

Після перехоплення цього рукостискання зловмисник отримує значення,

до якого він може застосувати 128-бітний ключ. Якщо зловмисник застосує правильний 128-бітний ключ, він зможе визначити PSK.

Сценарій хешування

Багато криптографічних атак спрямовані на злам паролів. Давайте почнемо цей сценарій з огляду того, де, як і навіщо потрібні паролі.

Комп'ютерна система або програма, яка вимагає парольної автентифікації, повинна містити базу даних або таблицю паролів. Оскільки ці таблиці паролів є потенційно вразливими для будь-кого, хто має технічні знання, щоб знайти їх і спробувати зламати, зберігання паролів у вигляді відкритого тексту є небезпечним. Тому дуже поширеною практикою є не зберігання пароля, а зберігання криптографічного хешу пароля користувача в базі даних.

Нижче наведено приклад гіпотетичного списку паролів, що складається з чотирьох імен користувачів з відповідними паролями, збереженими у вигляді хешів MD5. Як ви можете визначити, що це MD5? Просто підрахувавши кількість символів! Цей приклад можна використовувати для багатьох застосувань. У цьому прикладі використовується текстовий файл; список паролів може бути текстовим файлом, двійковим файлом, файлом cookie або параметрами реєстру, залежно від програми або операційної системи, яка використовує паролі.

Mike, bd4c4ea1b44a8ff2afa18dfd261ec2c8

Steve, 4015e9ce43edfb0668ddaa973ebc7e87

Janet, 8cbad96aced40b3838dd9f07f6ef5772

Penny, 48cccca3bab2ad18832233ee8dff1b0b

Тепер, коли у вас є хеші, ви можете хешувати всі можливі концептуальні паролі, щоб побачити, чи зможете ви підібрати хеш.

Сценарії атак

Представлені сценарії атаки на зашифрований текст і хеш мають спільні риси. Обидва покладаються на єдину точку даних: або шифрований текст, або хеш. Обидва мають чітко зрозумілий алгоритм: WPA і MD5. Нарешті, ви можете використовувати атаку, щоб застосувати ключ для отримання відкритого тексту.

Але як зловмисники здійснюють цю атаку? Які інструменти вони використовують, щоб прочитати зашифрований текст або хеш і спробувати витягти відкритий ключ або пароль? Ці атаки можуть здійснювати програми, які називаються зломщиками паролів або інструментами для відновлення паролів. Програми для злому паролів мають широкий спектр функцій та інтерфейсів. Наприклад, зловмисник на базі Linux використовує Jack the Ripper або Hashcat (Мал. 2-33). Зловмисник на базі Windows може використовувати поважний Cain & Abel (Рисунок 2-34).

Загалом, всі ці інструменти працюють за схожими схемами. По-перше, вони перехоплюють зашифрований текст або хеші, які ви хочете атакувати. По-друге, вони аналізують дані, щоб визначити тип шифрування (або вони знають шифрування від джерела). Нарешті, вони дають вам можливість атакувати більшістю, якщо не всіма, способами, перерахованими в цьому модулі.

Давайте розглянемо три найпоширеніші типи атак, які застосовуються до цих сценаріїв на основі паролів:

- Атака грубою силою
- Словникова атака
- Атака веселкових таблиць

Атака грубої сили

Атака грубим перебором - це найпростіший і найменш ефективний тип атаки на відомі зашифровані тексти/хеші. При переборі методом грубого перебору намагаються отримати пароль або ключ шляхом перевірки зашифрованого тексту або хешу, а потім перебирають всі можливі комбінації ключа або хешу до тих пір, поки не вдасться розшифрувати відкритий текст або згенерувати збіг. Складність методу грубої сили полягає в загальній кількості ітерацій, необхідних для перебору всіх можливих комбінацій ключів. Розглянемо перехоплене рукостискання WPA. Ключ 128-бітний, тому він може містити від 128 нулів до 128 одиниць, плюс всі комбінації між ними.

Отже, скільки часу потрібно, щоб перебрати 2128 комбінацій? Чим довше, тим більше комбінацій і, відповідно, складніше, тому 2128 дорівнює

3.402823669209384634633746074317682114 56 × 1038

Це величезне число, але комп'ютери потужні. Один дуже потужний процесор Intel Core i7 може обробляти півтрильйона операцій з плаваючою комою в секунду (FLOPS), і ви можете легко створити комп'ютер з великою кількістю таких процесорів. Крім того, зломщики паролів

ретельно оптимізовані для використання всіх цих FLOPS. Насправді, кращі зломщики можуть навіть скористатися перевагами потужних графічних процесорів (GPU) на відеокартах, а також спеціальних пристроїв на інтегральних схемах (ASIC), які працюють в сотні або навіть тисячі разів швидше, ніж центральний процесор (Мал. 2-35).

Отже, не витрачаючи обурливо багато тексту на тонкощі процесорів, FLOPS, графічних процесорів тощо, ми можемо стверджувати, що дуже потужна система злому може обробляти один трильйон спроб злому за секунду. Щоб підібрати всі можливі ключі, знадобиться приблизно 10 790 283 070 806 014 188 років. Якщо це здається вам неймовірно довгим часом, то так воно і є. Майте на увазі, що ключ може бути повністю нульовим, що означає, що ви отримаєте ключ WPA з першої спроби. Або ж ключ може складатися з одних одиниць, що означає, що вам доведеться спробувати всі 2128 комбінацій.

Це може наштовхнути вас на думку, що грубий перебір марний, і це для відносно великих ключів. Однак, коротші ключі можна легко зламати. Наприклад, 56-бітний ключ DES можна зламати (використовуючи попередні припущення) приблизно за 20 годин. Зі збільшенням обчислювальної потужності те, що колись було довгим ключем або хешем, стає коротким. Ось чому ключі і хеші з роками стають все довшими і довшими.

Але навіть якщо ми можемо виконати атаку грубої сили за кілька годин, чому б не придумати атаки, які прискорюють цей процес? Як ми можемо зменшити можливу кількість ключів для перебору?

Словникова атака

Найслабшою стороною паролів є те, що їх створюють люди. Люди створюють такі паролі, як цей:

Password123

Це чудовий пароль, чи не так? Він поєднує великі та малі літери з буквами та цифрами.

Хм... не дуже, коли мова йде про сучасні криптографічні атаки. Люди рідко створюють паролі, які виглядають так:

1x3^^7vEssde6)

Це набагато крутіший пароль, але це не те, що створюють звичайні люди.

Пам'ятаючи про це, ми можемо наказати нашій програмі-зломщику прочитати текстовий файл, заповнений усіма загальновживаними словами (в даному випадку) англійської мови - файл-словник. Потім програма-зломщик буде використовувати цей файл словника замість грубої сили в словниковій атаці. Ось крихітний, крихітний шматочок файлу словника, завантажений з Інтернету:

3.2 Моделі оцінки захищеності механізмів багатофакторної автентифікації користувачів від несанкціонованого доступу

Модуль 2-8: Криптографічні атаки

Цей модуль охоплює наступні цілі CompTIA Security+:

- 1.2 Порівнювати і протиставляти типи атак

- 2.2 Використовувати відповідні програмні засоби для оцінки стану безпеки в організації за певним сценарієм

- 6.1 Порівняйте та порівняйте основні концепції криптографії

Криптоаналіз, наука про злам шифрування, така ж давня, як і саме шифрування. Від Людовика XVI в його палаці у Версалі до останніх атак, спонсорованих державою, шифрування успішно атакували багато разів. У цьому модулі розглядаються деякі з найпоширеніших криптографічних атак та інструменти, що використовуються для їх здійснення. Розуміння цих атак та їхніх інструментів дає змогу краще зрозуміти, як захистити дані. (До того ж, зламувати щось дуже весело!)

Модуль починається із загального обговорення стратегій атак, а потім

розглядаються дані, що піддаються атакам. У другому розділі розглядаються атаки, орієнтовані на паролі, та інструменти, які зазвичай використовуються в таких атаках. Модуль закінчується розглядом кількох атак, які не пов'язані з паролями.

Стратегії атак

Давайте почнемо з розгляду того, де можна атакувати будь-яку криптосистему. Кожна криптосистема має базовий алгоритм, ключ (або у випадку з хешами - дайджест), або певну реалізацію самої криптосистеми, і все це зрозуміло зловмиснику. Всі криптографічні алгоритми відомі і рідко змінюються, відповідно до принципу Керкхоффа. Кожна криптографічна атака починається з ретельної перевірки криптосистеми і агресивного зондування та експериментів, щоб знайти якусь форму слабкості в одному з цих трьох аспектів.

Атака на алгоритм

Кожен криптографічний алгоритм можна зламати, якщо мати достатньо часу на дослідження. Саме тому організації та агентства, такі як Агентство національної безпеки (АНБ), оновлюють стандарти шифрування кожні кілька років. На момент написання цієї статті єдиним симетричним алгоритмом шифрування, який не виявив жодної суттєвої вразливості, є AES. Небезпека полягає в тому, що багато сучасних криптосистем можуть використовувати шифрування до AES. Transport Layer Security (TLS), криптосистема, яка використовується на кожному захищеному (HTTPS) веб-сайті, може додатково (і нерозумно) використовувати попередні AES-шифрування, такі як RC4 або DES. У цьому випадку досвідчені криптографи можуть зламати TLS, атакуючи алгоритм, що лежить в його основі.

Атака на ключ

Враховуючи, що практично кожна криптосистема використовує якусь форму ключа, очевидно, що криптоаналіз буде шукати в ньому слабкі місця або, що більш важливо, намагатися визначити ключ, просто атакуючи зашифрований текст. Однією з найбільших слабкостей є довжина ключа. Сучасне зростання обчислювальних потужностей полегшує злам ключів, які колись вважалися

занадто довгими для злому.

Розглянемо велосипедний замок з чотирма колесами проти велосипедного замка, у якого їх набагато більше, ніж чотири (Мал. 2-31). Який з них швидше зламати? Та ж концепція застосовується і до ключів. Якщо взяти шматок зашифрованого тексту і спробувати всі можливі ключі, поки не з'явиться щось схоже на відкритий текст - це атака грубою силою. (Про неї ми поговоримо пізніше).

DES має ефективну довжину ключа всього 56 біт. Це означає, що він має ключовий простір (загальна кількість можливих ключів) 256 або 72 057 594 037 927 936 різних комбінацій. Це число здається неймовірно великою кількістю комбінацій, але спеціалізовані обчислювальні системи, такі як Deep Crack Федерації електронної свободи, зламали ключ DES за 22 години... в далекому 1998 році. Сьогодні обчислювальна потужність настільного комп'ютера високого класу не надто далека від такої ж потужності.

Атака на реалізацію

Деякі реалізації нових криптосистем мають певний вроджений недолік, який призводить до непередбачуваної вразливості, яка не є виною алгоритму, що лежить в їх основі. Дротовий еквівалент конфіденційності (WEP) був повноцінною криптосистемою шифрування 802.11, представленою в першому поколінні 802.11 WAP в 1999 році. Отримавши широке розповсюдження, вже через рік було виявлено, що WEP погано реалізує спосіб використання базового шифрування RC4. Цей недолік дозволяв за допомогою простого криптоаналізу отримати ключ, зібравши достатньо зашифрованих даних.

Дані, які можна атакувати

Жодна атака на шифр не спрацює, якщо у вас немає доступу до чогось, що можна зламати. Ці дані, які можна зламати, можна розділити на два різних типи: зашифрований текст або хеші. Перш ніж ми почнемо атакувати, давайте спочатку створимо два приклади сценаріїв: один із зашифрованим текстом, а інший з хешами. А потім почнемо атакувати.

Сценарій із зашифрованим текстом

Візьмемо один приклад мережі 802.11, зашифрованої за допомогою Wi-Fi Protected Access Pre-Shared Key (WPA-PSK). Коли клієнт підключається до точки доступу, він проходить через просту процедуру рукостискання, яка визначає, що кожна сторона зберігає однаковий PSK. Цей процес не передає ключ, але містить достатньо інформації, щоб надати ключ досвідченому хакеру. Програмісти розробили програми, які можуть перехоплювати рукостискання WPA. На рисунку 2-32 показано популярну програму Aircrack-ng, яка перехоплює рукостискання.

Після перехоплення цього рукостискання зловмисник отримує значення, до якого він може застосувати 128-бітний ключ. Якщо зловмисник застосує правильний 128-бітний ключ, він зможе визначити PSK.

Сценарій хешування

Багато криптографічних атак спрямовані на злам паролів. Давайте почнемо цей сценарій з огляду того, де, як і навіщо потрібні паролі.

Комп'ютерна система або програма, яка вимагає парольної автентифікації, повинна містити базу даних або таблицю паролів. Оскільки ці таблиці паролів є потенційно вразливими для будь-кого, хто має технічні знання, щоб знайти їх і спробувати зламати, зберігання паролів у вигляді відкритого тексту є небезпечним. Тому дуже поширеною практикою є не зберігання пароля, а зберігання криптографічного хешу пароля користувача в базі даних.

Нижче наведено приклад гіпотетичного списку паролів, що складається з чотирьох імен користувачів з відповідними паролями, збереженими у вигляді хешів MD5. Як ви можете визначити, що це MD5? Просто підрахувавши кількість символів! Цей приклад можна використовувати для багатьох застосувань. У цьому прикладі використовується текстовий файл; список паролів може бути текстовим файлом, двійковим файлом, файлом cookie або параметрами реєстру, залежно від програми або операційної системи, яка використовує паролі.

Mike, bd4c4ea1b44a8ff2afa18dfd261ec2c8

Steve, 4015e9ce43edfb0668ddaa973ebc7e87

Janet, 8cbad96aced40b3838dd9f07f6ef5772

Penny, 48cccca3bab2ad18832233ee8dff1b0b

Тепер, коли у вас є хеші, ви можете хешувати всі можливі концептуальні паролі, щоб побачити, чи зможете ви підібрати хеш.

Сценарії атак

Представлені сценарії атаки на зашифрований текст і хеш мають спільні риси. Обидва покладаються на єдину точку даних: або шифрований текст, або хеш. Обидва мають чітко зрозумілий алгоритм: WPA і MD5. Нарешті, ви можете використовувати атаку, щоб застосувати ключ для отримання відкритого тексту. Але як зловмисники здійснюють цю атаку? Які інструменти вони використовують, щоб прочитати зашифрований текст або хеш і спробувати витягти відкритий ключ або пароль? Ці атаки можуть здійснювати програми, які називаються зломщиками паролів або інструментами для відновлення паролів.

3.3 Висновки по розділу

1. Узагальнено порядок реалізації процедури ідентифікації та автентифікації користувачів в інформаційно-комунікаційних мережах. Виділені елементи вказаного процесу, його фактори: пароль, пристрій автентифікації, біометрика користувача. Описано алгоритм автентифікації, та порядок користування паролем в різних операційних системах.

Наведено порівняння методів автентифікації по критерію ризику.

2. Запропоновано математичні моделі оцінки захищеності механізмів багатофакторної автентифікації користувачів від несанкціонованого доступу.

Вказані моделі враховують ймовірність виникнення випадку несанкціонованого доступу в мережу та ймовірність захисту мережі від вказаного випадку та дозволяють оцінити загальну захищеність з врахуванням часткових ймовірностей захищеності, які забезпечуються різними факторами автентифікації на основі комбінованих та послідовних механізмів побудови.

ВИСНОВКИ

У процесі підготовки кваліфікаційної роботи були виконані наступні задачі:

– Визначено порядок розподілу інформації на об’єкті інформаційної діяльності, проведено аналіз загроз інформаційній безпеці та аналіз технічних каналів витоку інформації на об’єкті інформаційної діяльності.

– Проведено огляд методів і засобів ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.

– Розроблено та вдосконалено механізми ідентифікації та автентифікації користувачів при доступі до роботи з інформацією з обмеженим доступом.

1. У першому розділі кваліфікаційної роботи проведено аналіз загроз та визначено канали витоку технічної інформації. Показана структура технічного каналу витоку інформації. Здійснено аналіз технічних каналів витоку інформації та подана їх класифікація. А саме: розподіл за видом інформаційної діяльності, принципом формування небезпечного сигналу, за середовищем поширення небезпечного сигналу, за способом перехоплення (зняття) небезпечного сигналу засобами технічної розвідки противника.

2. У другому розділі кваліфікаційної роботи проведено огляд існуючих технологій ідентифікації та автентифікації користувача в інформаційно-комунікаційних системах об’єкту інформаційної діяльності. Розглянуто зміст, особливості застосування, переваги та недоліки парольної, апаратної, біометричної ідентифікації та автентифікації користувачів в інформаційно - комунікаційних системах. Подано вимоги до процесів ідентифікації та автентифікації та зміст атак на них.

3. У третьому розділі кваліфікаційної роботи узагальнено порядок реалізації процедури ідентифікації та автентифікації користувачів в інформаційно-комунікаційних мережах. Виділені елементи вказаного процесу, його фактори: пароль, пристрій автентифікації, біометрика користувача. Описано алгоритм автентифікації, та порядок користування паролем в різних

операційних системах. Наведено порівняння методів автентифікації по критерію ризику. Запропоновано математичні моделі оцінки захищеності механізмів багатофакторної автентифікації користувачів від несанкціонованого доступу. Вказані моделі враховують ймовірність виникнення випадку несанкціонованого доступу в мережу та ймовірність захисту мережі від вказаного випадку та дозволяють оцінити загальну захищеність з врахуванням часткових ймовірностей захищеності, які забезпечуються різними факторами автентифікації на основі комбінованих та послідовних механізмів побудови.

ПЕРЕЛІК ПОСИЛАНЬ

ДОДАТОК

Оформлення слайдів та роздаткового матеріалу

ДОДАТКИ
Презентаційний матеріал

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА ТЕХНІЧНИХ СИСТЕМ КІБЕРЗАХИСТУ

ПАНАСЮК ОЛЕКСАНДР ОЛЕКСІЙОВИЧ

Кваліфікаційна робота
здобувача освітнього ступеня «МАГІСТР»

СИСТЕМА БАГАТОФАКТОРНОЇ ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ КОРИСТУВАЧА ПРИ РОБОТІ З ІНФОРМАЦІЄЮ З ОБМЕЖЕНИМ ДОСТУПОМ

Спеціальність: 125 – Кібербезпека та захист інформації
Керівник: д.т.н., професор Олександр ТУРОВСЬКИЙ

Київ – 2025

Слайд 1

АКТУАЛЬНІСТЬ РОБОТИ

• зростання ризиків несанкціонованого доступу до інформації з обмеженим доступом
• необхідність посилення ідентифікації та автентифікації користувачів
• доцільність поєднання паролів, токенів, QR-кодів і біометричних ознак

Мета роботи:
підвищення надійності доступу до конфіденційної інформації шляхом удосконалення механізмів багатофакторної ідентифікації та автентифікації користувачів.

Об’єкт дослідження:
процес ідентифікації та автентифікації користувачів при доступі до інформації з обмеженим доступом.

Предмет дослідження:
механізми ідентифікації та автентифікації користувачів у захищених інформаційно-комунікаційних мережах.

Слайд 2

ЗДОБУТІ РЕЗУЛЬТАТИ РОБОТИ

1. РОЗДІЛ 1
Визначено порядок розподілу інформації на ОІД, проаналізовано загрози та технічні канали витоку інформації.

2. РОЗДІЛ 2
Проведено огляд методів і засобів ідентифікації та автентифікації користувачів при роботі з ІзОД.

3. РОЗДІЛ 3
Розроблено та вдосконалено механізми багатофакторної ідентифікації та автентифікації користувачів.

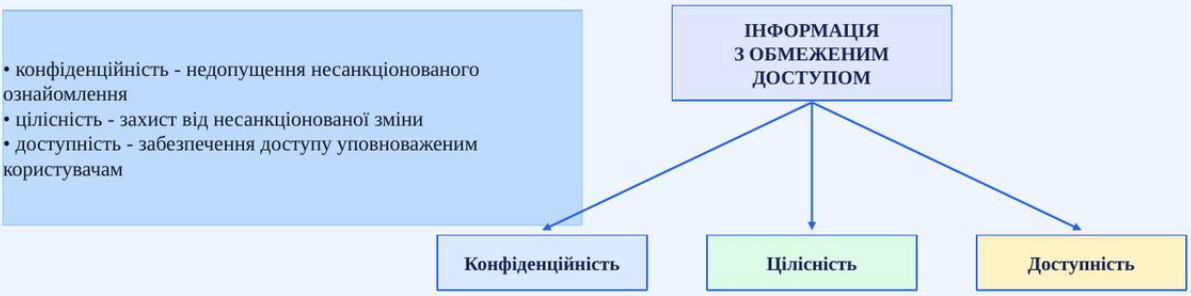
4. ВИСНОВКИ
Сформовано практичні рекомендації щодо застосування комбінованих механізмів доступу.

Слайд 3

ПЕРШИЙ РЕЗУЛЬТАТ

ЗАХИСТ ІНФОРМАЦІЇ НА ОБ’ЄКТАХ ІНФОРМАЦІЙНОЇ ДІЯЛЬНОСТІ

• конфіденційність - недопущення несанкціонованого ознайомлення
• цілісність - захист від несанкціонованої зміни
• доступність - забезпечення доступу уповноваженим користувачам



Комплексний захист ОІД потребує одночасного врахування організаційних, технічних і програмних механізмів контролю доступу.

Слайд 4

ПЕРШИЙ РЕЗУЛЬТАТ

ЗАГРОЗИ ТА ТЕХНІЧНІ КАНАЛИ ВИТОКУ ІНФОРМАЦІЇ

Результат аналізу: найбільш суттєві ризики пов'язані з несанкціонованим доступом, слабкою автентифікацією, атакою на паролі та помилками користувачів.

Слайд 5

ДРУГИЙ РЕЗУЛЬТАТ

ПРОЦЕС ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ

- встановлення унікального ідентифікатора користувача
- перевірка достовірності за одним або кількома факторами
- надання прав доступу відповідно до регламенту

Слайд 6

ДРУГИЙ РЕЗУЛЬТАТ

МЕТОДИ І ЗАСОБИ АВТЕНТИФІКАЦІЇ

Пароль / PIN
«що знає користувач»

Токен / ключ
«що має користувач»

QR-код
динамічний ідентифікатор

Біометрія
«чим є користувач»

- однофакторний доступ має нижчу стійкість до компрометації
- комбінування факторів зменшує ймовірність НСД
- доцільним є використання послідовних і комбінованих схем перевірки

Принцип MFA: доступ надається лише після підтвердження кількох незалежних ознак користувача.

Слайд 7

ДРУГИЙ РЕЗУЛЬТАТ

АТАКИ НА МЕХАНІЗМИ АВТЕНТИФІКАЦІЇ

Атака грубої сили
перебір усіх можливих паролів або ключів

Словникова атака
перевірка поширених слів, шаблонів і комбінацій

Веселкові таблиці
використання підготовлених хеш-таблиць

Захисні заходи

довгі паролі

сіль + хешування

обмеження спроб

MFA

Висновок: парольний фактор не повинен застосовуватися ізольовано для доступу до інформації з обмеженим доступом.

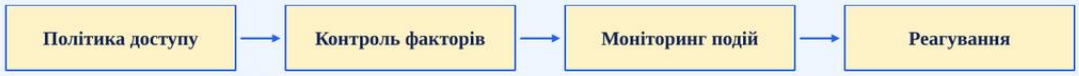
Слайд 8

ДРУГИЙ РЕЗУЛЬТАТ

ВИМОГИ ДО ОРГАНІЗАЦІЇ ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ

- унікальність облікових записів та заборона спільного використання ідентифікаторів
- регламентована зміна паролів і контроль складності облікових даних
- журналювання спроб входу, відмов та підозрілих дій

- розмежування прав доступу за ролями та повноваженнями
- додаткова перевірка для доступу до критичних ресурсів
- аудит механізмів автентифікації та реагування на інциденти



Слайд 9

ТРЕТІЙ РЕЗУЛЬТАТ

РОЗРОБКА МЕХАНІЗМІВ БАГАТОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ

Запропонований підхід передбачає поєднання незалежних факторів автентифікації для зниження ймовірності несанкціонованого доступу та підвищення достовірності перевірки користувача.

- послідовні схеми - кожен фактор перевіряється окремим етапом
- комбіновані схеми - фактори об’єднуються в єдиний механізм рішення
- ймовірнісна оцінка дає змогу порівнювати стійкість варіантів

Слайд 10

ТРЕТІЙ РЕЗУЛЬТАТ

КОМБІНОВАНИЙ МЕХАНІЗМ ДВОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ

Сутність механізму: навіть у разі компрометації пароля доступ блокується без другого незалежного фактора.

Застосування: робота з документами, базами даних, корпоративними ресурсами та інформацією з обмеженим доступом.

Слайд 11

ТРЕТІЙ РЕЗУЛЬТАТ

ПОСЛІДОВНО-ПАРАЛЕЛЬНИЙ МЕХАНІЗМ ТРИФАКТОРНОЇ АВТЕНТИФІКАЦІЇ

- підвищується стійкість до підбору пароля та крадіжки токена
- біометрична ознака забезпечує персональну прив’язку користувача
- послідовність перевірок зменшує ризик помилкового допуску

Слайд 12

ТРЕТІЙ РЕЗУЛЬТАТ

ОЦІНКА ЙМОВІРНОСТІ НЕСАНКЦІОНОВАНОГО ДОСТУПУ

Для незалежних факторів автентифікації загальна ймовірність НСД може оцінюватися через добуток імовірностей компрометації окремих факторів.

РНСД = Рпароль · Ртокен · Рбіометрія

Практичний ефект: додавання кожного незалежного фактора суттєво знижує інтегральний ризик несанкціонованого доступу.

Слайд 13

ВИСНОВКИ

- визначено основні загрози інформаційній безпеці ОІД та технічні канали витоку інформації
- узагальнено технології ідентифікації та автентифікації користувачів при доступі до ІзОД
- запропоновано механізми багатофакторної автентифікації з використанням пароля, QR-коду, токена та біометрії
- обґрунтовано зниження ймовірності несанкціонованого доступу за рахунок поєднання незалежних факторів

ДЯКУЮ ЗА УВАГУ!

Слайд 14