НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Комп'ютерний практикум №4 з курсу
Методи Реалізації Криптографічних Механізмів

Дослідження особливостей реалізації існуючих програмних систем, які використовують криптографічні механізми захисту інформації.

Виконав студент
групи ФІ-22мн
Кушнір О.С.

Перевірив:
Селюх П.В.

Київ — 2022

1 МЕТА КОМП'ЮТЕРНОГО ПРАКТИКУМУ

Отримання практичних навичок побудови гібридних криптосистем.

2 ПОСТАНОВКА ЗАДАЧІ

Дослідити стійкість стандартних криптопровайдерів до атак, що використовують недосконалість механізмів захисту операційної системи Windows 10. Особливості реалізації стандартних крипто провайдерів ОС Windows 10. Вразливості реалізації прозорого шифрування та протоколів автентифікації. Вразливості системи збереження ключової інформації.

3 ХІД РОБОТИ

Cryptographic Service Provider (CSP) — це незалежний модуль, що дозволяє здійснювати криптографічні операції в операційних системах Microsoft Windows, управління яким відбувається за допомогою функцій Microsoft CryptoAPI. Модуль реалізує функції розшифрування і зашифрування які потім прикладна програма може використати для створення системи аутентифікації, захищеної електронної пошти чи чогось подібного. Простими словами криптопровайдер виступає посередником між операційною системою та виконавцем криптографічних опрецій (програма чи апаратний комплекс). У всі операційні системи Microsoft, починаючи з Windows 2000, вбудований криптопровайдер Microsoft Base Cryptographic Provider, який володіє набором основних криптографічних функцій. У Microsoft Base Cryptographic Provider довжина ключів шифрування не перевищує 40 біт. Так як до січня 2000 року в США існувала заборона на експорт програмного забезпечення для шифрування з використанням ключів довжиною більше 40 біт, то в Windows 98 і ранніх версіях Windows 2000 існувала підтримка тільки цього криптопровайдера. Microsoft Base Cryptographic Provider по суті є урізаним варіантом Microsoft Enhanced Cryptographic Provider. Але після скасування заборони на експорт стало безглуздо мати 2 криптопровайдера, тому програмісти Microsoft ввели ще одну назву — Microsoft Strong Cryptographic Provider, який нічим не відрізняється від Microsoft Enhanced Cryptographic Provider. Цей криптопровайдер є криптопровайдером за замовчуванням типу PROV_RSA_FULL Windows 2000, Windows XP, Windows 2003.

Всі криптопровайдери Microsoft можуть бути завантажені з сайту Microsoft.

Одним з основних об'єктів є ключовий контейнер. Контейнер має своє ім'я, створюється (або запитується, якщо вже був створений) функцією CryptAcquireContext(...). У контейнері може існувати не більше однієї пари ключів підпису, однієї пари ключів обміну і одного симетричного ключа. Якщо підтримується декілька алгоритмів симетричного шифрування, то симетричних ключів може бути кілька, по одному ключу кожного алгоритму.

Пари ключів і симетричні ключі можуть знаходитися тільки в контейнері. Тільки відкритий ключ пари може перебувати поза контейнером.

Закриті (private) ключі пар ключів експортуються тільки в зашифрованому вигляді. Деякі криптопровайдери принципово не дозволяють експортувати закриті ключі, навіть у зашифрованому вигляді. Симетричні ключі при експорті також обов'язково шифруються на відкритому ключі одержувача або ключі узгодження. Для обчислення хеш-функцій

Microsoft Base Cryptographic Provider	MS_DEF_PROV	PROV_RSA_FULL	Має широкий набір основних криптографічних функцій. Довжина ключів шифрування не перевищує 40 біт.
Microsoft Strong Cryptographic Provider	MS_STRONG_PROV	PROV_RSA_FULL	Відрізняється від Microsoft Base Cryptographic Provider підтримкою великої довжини ключів.
Microsoft Enhanced Cryptographic Provider	MS_ENHANCED_PROV	PROV_RSA_FULL	Нічим не відрізняється від Microsoft Strong Cryptographic Provider. Є криптопровайдером за замовчуванням.
Microsoft AES Cryptographic Provider	MS_ENH_RSA_AES_PROV	PROV_RSA_AES	= Microsoft Enhanced Cryptographic Provider з підтримкою AES
Microsoft DSS Cryptographic Provider	MS_DEF_DSS_PROV	PROV_DSS	Хешування, підпис, перевірка підпису з підтримкою алгоритму DSS.
Microsoft Base DSS and Diffie-Hellman Cryptographic Provider	MS_DEF_DSS_DH_PROV	PROV_DSS_DH	Хешування, підпис DSS, генерація та обмін ключами Діффі-Геллмана. Підтримує генерацію ключів для протоколів SSL3 і TLS1.
Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider	MS_ENH_DSS_DH_PROV	PROV_DSS_DH	Те ж, що і Microsoft Base DSS and Diffie-Hellman Cryptographic Provider з підтримкою ключів великої довжини.
Microsoft DSS and Diffie-Hellman/Schannel Cryptographic Provider	MS_DEF_DH_SCHANNEL _PROV	PROV_DH_SCHANNEL	Хешування, підпис DSS, генерація та обмін ключами Діффі-Геллмана. Підтримує генерацію ключів для протоколів SSL3 і TLS1.
Microsoft RSA/Schannel Cryptographic Provider	MS_DEF_RSA_SCHANNEL _PROV	PROV_RSA_SCHANNEL	Хешування, підпис, перевірка підпису. Використовується для автентифікації протоколу SSL 3.0 and TLS 1.0.
Microsoft RSA Signature Cryptographic Provider	MS_DEF_RSA_SIG_PROV	PROV_RSA_SIG	Мінімальна функціональність, необхідна для електронного підпису та перевірки ЕЦП.

створюються об'єкти хешування. Для створення об'єктів хешування створювати контейнер не потрібно.

У Windows всі активні сутності, які можуть бути автентифіковані операційною системою (користувачі або групи), називаються учасниками безпеки (security principals).

Всі учасники безпеки мають унікальний ідентифікатор змінної довжини Security ID (SID). Структура SID наступна - S-R-IA-SA-RID, наприклад S-1-5-21-1687231434-1254558764-1544283289-1004, де:

S — літеральний префікс, що вказує на те, що ідентифікатор є SID (це просто конвенція найменування); R – однобайтне значення версії або ревізії (revision) SID. Поки що існує лише версія 1; IA – джерело видачі (issuing authority), шестибайтне значення. Вказує, у чиїй області відповідальності було видано SID (буквально authority означає «орган влади»). Майже завжди має значення 5 (SECURITY_NT_AUTHORITY), за винятком well-known SID, про які ми поговоримо трохи згодом. Наприклад, 1 означає SECURITY_WORLD_SID_AUTHORITY і відноситься до well-known-групи Everybody; SA – уповноважений центр (sub-authority). Унікальне (в рамках IA) значення складається з чотирьох частин: 4-байтного числа, що вказує, ким був виданий ідентифікатор (контролером домену або локальним комп'ютером), та 12-байтного значення, яке ділиться на три частини та ідентифікує конкретний об'єкт, що видав ідентифікатор . Сенс цього поля у тому, що за наявності кількох доменів у лісі об'єкти у різних доменах матимуть унікальний SA; RID - відносний ідентифікатор (Relative-ID), 4-байтне значення, служить для поділу об'єктів усередині домену. Для вбудованих облікових записів RID завжди буде той самий (наприклад, для облікового запису адміністратора RID = 500). Якщо бути точнішим, існує SID машини (machine SID) і SID домену (domain SID). А сам SID є базовим ідентифікатором (S, R, IA, SA) + RID.

Також є стандартні так звані well-known SID для користувачів та груп. Вони мають той самий SID на будь-яких системах (наприклад, група Everyone або користувач System).

Також при адмініструванні можна припустити невеликий недолік, пов'язаний з дублікацією сидів. Іноді впливає на безпеку або функціональність (наприклад, коли ОС розгортають, просто копіюючи диск).

SID, у свою чергу, входить до так званого маркеру доступу — програмного об'єкта (структура в ядрі Windows), який закріплюється за сесією (logon session) учасників безпеки після авторизації. За видачу маркера, як і автентифікацію, відповідає LSASS (local security authority subsystem).

Крім усього іншого, в маркер включені SID користувача та його груп, а також механізм привілеїв на здійснення будь-яких дій (наприклад, привілей на налагодження debug, який, до речі, використовується в mimikatz для отримання доступу до системних процесів).

Після того, як видаляється останній асоційований із сесією токен, LSASS видаляє і саму сесію - таким чином завершується сеанс користувача. Докладніше можна вивчити структуру маркера доступу у відладчику ядра (kernel debugger) за допомогою команди

dt_TOKEN. Взагалі, якщо не вдається нагуглити подробиці про внутрішній пристрій Windows, можна вивчити структуру самому за допомогою засобів налагодження. Подробиці наведено в документації Microsoft. З привілеями може бути, наприклад, пов'язана така річ, як bypass traverse checking.

У свою чергу, для «пасивних» об'єктів, які призначені для надання доступу до них ззовні (їх називають securable objects), використовується SD — security descriptor. Це дескриптор для керування доступом до даних об'єктів (наприклад, процес може мати SD або SD може бути прив'язаний до файлу NTFS). У SD теж, до речі, бувають проблеми з безпекою.

У security descriptor включені ACL (access control list), які бувають двох видів - SACL (необхідний ведення журналу доступу до об'єкта) і DACL (потрібний безпосередньо управління доступом). У свою чергу, ACL включають ACE (access control entry) - кожна ACE призначена для конкретного суб'єкта, який отримує доступ, і містить тип (заборона або дозвіл) і маску доступу. Правильно налаштовані ACL дозволяють перекрити несанкціонований доступ до об'єктів усередині системи.

Однак, якщо хтось завантажиться на такому комп'ютері в Linux або витягне жорсткий диск з машини з Windows і примонтує його в тому ж Linux, він зможе прочитати ці дані. У Windows вдасться отримати доступ до даних із «чужого» носія NTFS, тільки якщо користувач або його група на іншій системі має той самий SID — наприклад, якщо в ACL виставлено надто багато прав на well-known-групи.

Від цього можна захиститися лише шифруванням (FDE), наприклад BitLocker.

У ядрі Windows перевірки ACL виконуються за допомогою security reference monitor та objectmanager. Надання доступу виглядає так: суб'єкт (користувач) після авторизації отримує маркер доступу, потім суб'єкт звертається до файлу, система порівнює необхідні дані з маркера доступу з відповідними ACE в об'єкті DACL, і в залежності від дозволів суб'єкт отримує доступ або відмову.

Як ми бачимо FDE є невід'ємною частиною побудови безпечної системи на базі Windows, бо воно захищає контейнери ключів. Тому було розглянуто BitLocker.

BitLocker використовується симетричне шифрування диска. За замовчуванням алгоритм AES з ключем довжиною 128 біт. У нових збірках Windows 10 використовується режим XTS; Старіші версії ОС і перший реліз Windows 10 використовують режим CBC.

Як саме виконується шифрування: засобами центрального процесора (з використанням команд AES-NI) або контролера диска, питання складне. Донедавна Windows віддавала перевагу механізмам шифрування, вбудованим у контролер диска або SSD, проте виявлена дослідниками в цілому ряді моделей вразливість у реалізації

подібних механізмів змусила Microsoft відмовитися від цієї практики та реалізувати шифрування силами центрального процесора. Втім, це стосується тільки новостворених томів; вже зашифровані диски продовжуватимуть використовувати той механізм, який був використаний під час їх створення.

На наш погляд, обидва механізми еквівалентні. Для розшифрування диска в будь-якому випадку знадобиться майстер-ключ, який можна отримати одним із кількох способів:

1. Розшифрувати паролем, якщо розділ зашифрований саме в такий спосіб.
2. Розшифрувати ключем відновлення (Recovery Key).
3. Витягти безпосередньо з модуля TPM.

Також варто зазначити, що в ідеальній системі BitLocker з TPM використовується разом з паролем на BIOS, що може вберегти нас від деяких атак. Також варто вберегтися від апаратних атак як мінімум обмеживши доступ до usb-портів.

ВИСНОВКИ

Під час виконання лабораторної работи було досліджено стандартні CSP операційних систем родини Windows як видно з наведених фактів: можливість копіювання контейнера ключів, а також вразливість операційної системи в її дефолтному налаштуванні робить використання стандартних провайдерів недоцільним (також варто зауважити що не всі стандартні провайдери мають достатню довжину ключа, для того щоб вважатись досить безпечними для деяких задач). Однак дуже велика кількість інструментів, можливість додаткового захисту у вигляді шифрування дискової інформації, використання програмно-аппаратного комплексу TPM (наприклад у продукції Surface), пароль на BIOS, PIN, коректні налаштування системи, апаратні методи захисту тощо - роблять можливим побудову надійних криптосистем в рамках Windows.