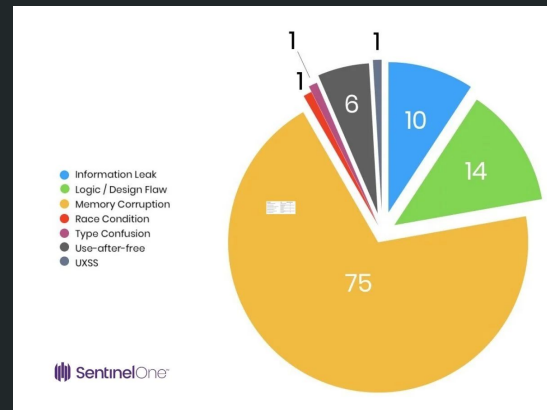


Как несколько ошибок могут предоставить полный доступ к чужому телефону: разбор на примере эксплойта FORCEDENTRY.

Раевский Григорий, P3221

Введение

- Наиболее популярный вид уязвимостей — манипулирование памятью
- FORCEDENTRY (CVE-2021-30860) от NSO GROUP
- Защитные техники: NX, ASLR, DEP, Sandboxing



Обзор уязвимости FORCEDENTRY

- Эксплойт работал на iOS 14
- Уязвимость исправлена в iOS 14.8
- Впервые обнаружен в марте 2021 канадской лабораторией CitizenLab
- Разработан израильской компанией NSO Group.
- Работает благодаря некорректной обработке PDF с кодировкой JBIG2

Небольшое отступление. JBIG2

JBIG2 - формат сжатия изображений, разработанный в 2000 году.

Обеспечивает эффективное сжатие текстовых документов.

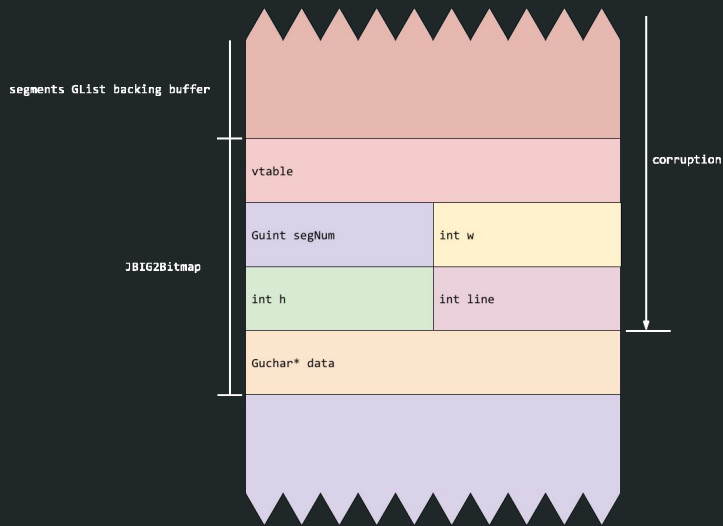
- Распознает символы, сохраняет 1 экземпляр каждого символа и запоминает расположение их на документе
- С помощью логических операторов сравнивает референтный символ и текущий, сохраняя различия. Это позволяет улучшить качество изображения
- В процессе сжатия страдает качество и возможны ошибки из-за перепутанных символов
- Состоит из сегментов, BitMap и SymbolDictionary

Код

Часть кода, в которой срабатывает переполнение:

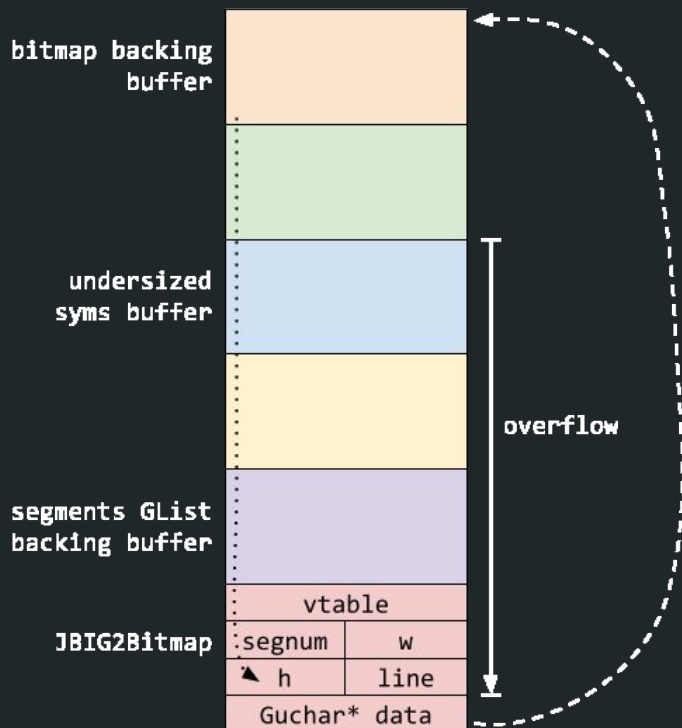
```
Guint numSyms = 0; // - размер всех словарей
for () {
    numSyms += ((JBIG2SymbolDict *)seg)->getSize(); // - получение общего размера всех
словарей
}
syms = (JBIG2Bitmap **)gmallocn(numSyms, sizeof(JBIG2Bitmap *)); // - выделение памяти
для массива указателей на Bitmap
for () {
    syms[kk++] = symbolDict->getBitmap(k); // - запись Bitmap в выделенный буфер
```

Обработка BltMap



- JBIG2Bitmap - оболочки вокруг резервного буфера
- После перезаписи 3 указателей переполнение останавливается
- iOS - little-endian => segNum и h получают большое значение
- Возможность читать и писать данные за пределами исходных границ

Ну как там с ~~деньгами~~ памятью?



Подбирая отступы в *w* и *h* можно писать данные в произвольное место буфера.

Возможность манипулировать памятью все еще ограничена, так как мы находимся в обработке изображений.

А вот и логика

- Однопроходный парсер не дает возможности запуска скриптов
- JBIG2 поддерживает базовые логические операции (AND, OR, XOR, XNOR)
- Логические операторы + неограниченный доступ к памяти = наборы команд
- Из набора команд можно создать архитектуру и запрограммировать



Защитные техники

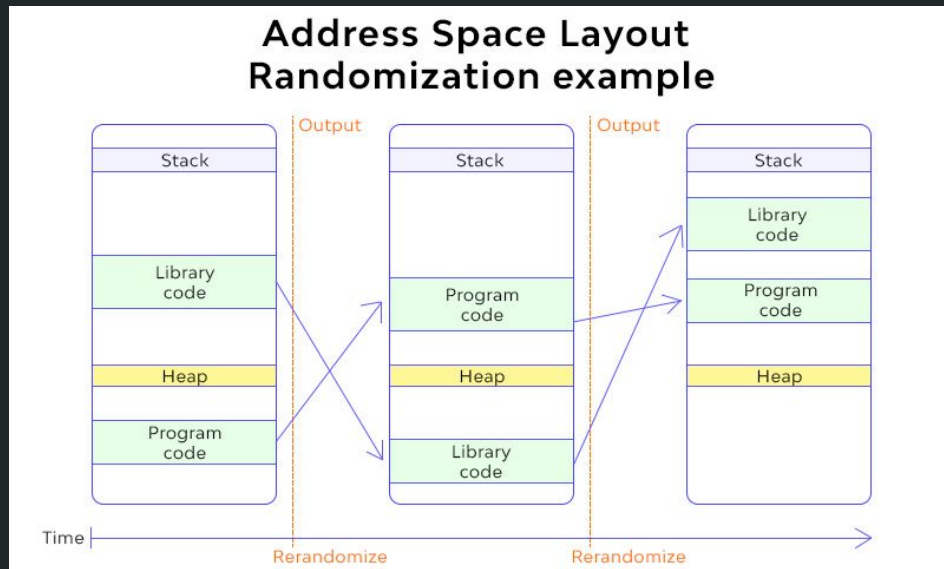
- NX - аппаратный механизм для разметки памяти
- ASLR - механизм рандомизации адресов памяти
- DEP - механизм разметки памяти
- Sandboxing - механизм изоляции процессов и модулей

NX и DEP

- NX (No execute) - аппаратное разделение кода на исполняемый и неисполняемый. Реализуется с помощью управляющего бита, который помечает область памяти. Обеспечивает хорошую защиту от выполнения произвольного кода.
- DEP (Data execution prevention) - схож с NX. Может быть аппаратным или программным. Обладает большей гибкостью, но меньшей безопасностью.

ASLR

ASLR (Address Space Layout Randomization) - рандомизация адресов памяти. Защищает от атак, направленных на составление кода из различных кусков данных (гаджетов).



Методы эксплуатации

- FORCEDENTRY обходит NX и DEP, так как находится в памяти, которая считается безопасной
- Эксплойт получает доступ к управлению памятью, т. е. ASLR так же не способен обеспечить защиту
- Отсутствие защит и проверок в библиотеке для обработки изображений дают фактически неограниченные возможности злоумышленнику

Песочница

Sandbox (Песочница) - механизм защиты для запуска программ или модулей в контролируемой среде.

- Изоляция процессов
- Контроль доступа
- Ограничение прав



Улучшения защиты iOS 14

- Улучшение ASLR - теперь адрес для доступа к основным библиотекам обновляется после каждого запроса
- Crash Throttling - преднамеренное замедление работы приложений при их многократном краше
- BlastDoor - модуль, отвечающий за обработку данных в iMessage

BlastDoor

- Появился в iOS 14
- Создан на Swift
- Ограниченное число доступных модулей
- Серьезные ограничения к файловой системе



Побег из песочницы

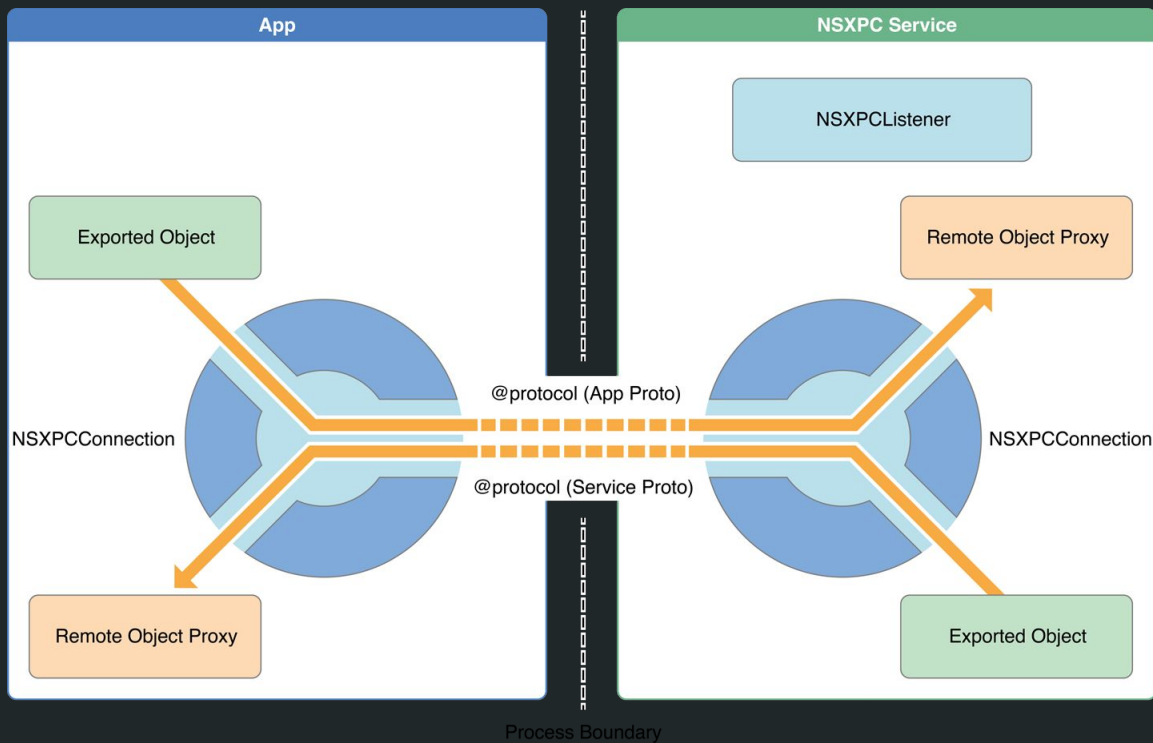
Злоумышленник уже создал архитектуру внутри парсера, но основное шпионское ПО все еще не получено.

Для побега будут использоваться NSExpression и NSPredicate. Пейлоды будут эксплуатироваться в среде CommCenter, доступной благодаря NSXPC.



NSXPCConnection

- Механизм RPC
- Для вызова объектов необходимы протоколы
- Для передачи доступны любые подклассы



Заметание следов

Одно из созданных ранее
NSFunctionExpression с помощью файловой
системы находит и удаляет все полученные
ранее .gif.



Голос и PTRow

Через NSXPC передается два объекта: AVSpeechSynthesisVoice и PTSection:

1. Инициализация AVSpeechSynthesisVoice загружает различные библиотеки в CommCenter, в том числе и библиотеку для работы с PTSection и PTRow.
2. PTRow содержит предикат, причем во время десериализации не будет выполняться никаких проверок

Pegasus к нам приходит

- Создание URL для доступа к серверу
- Загрузка библиотек
- Сохранение параметров устройства
- Параметризация URL
- Установление связи со шпионским сервером и получение информации от него

Заключение. Основные ошибки

- Недостаточная обработка входных данных
- Отсутствие защиты от переполнения
- Ошибки в архитектуре логики модулей
- Недостаточная изоляция