guardant

конференция РусКрипто

Защита от копирования программной лицензии

Михаил Бакаляров,
Руководитель департамента разработки Guardant

Решения для защиты и монетизации программного обеспечения

Что мы умеем:

- Защита и лицензирование программного обеспечения
- Управление лицензиями
- Защита встраиваемых систем
- Безопасность программного кода

Ключевые преимущества:

- Стойкие защитные технологии, проверенные временем
- Гибкая система лицензирования
- Высокая степень автоматизации для эффективного применения
- Интеграция со сторонними системами (ERP, CRM и т.д.)

Программная лицензия

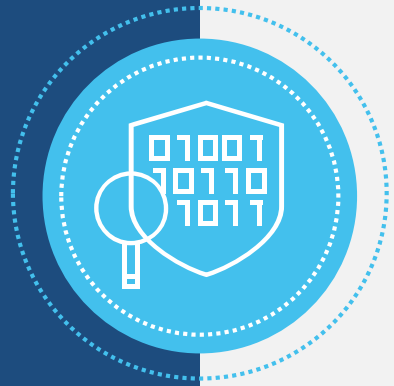
Вечная лицензия

- Лицензия хранится в файле на диске
- Файл зашифрован и подписан приватным ключом
- Лицензия привязана к аппаратным компонентам ПК

Ограниченная лицензия

- Защита файла лицензии от копирования
- Блокировка лицензии

Защита лицензии
от копирования



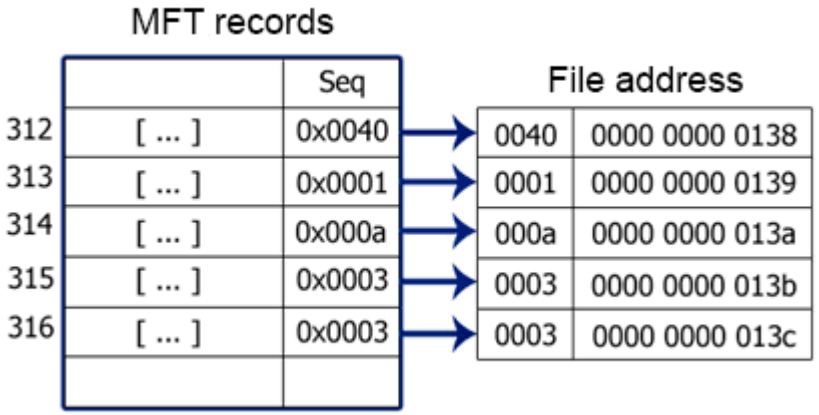
Исходные данные для задачи:

- Лицензия хранится в файле на диске
- Лицензия может меняться в процессе работы программы
- Файл лицензии разрешается перемещать
- Восстановление файла лицензии из резервной копии не допускается
- Алгоритм защиты должен быть одинаковым для Windows, Linux и MacOS
- Все данные должны храниться внутри файла лицензии

Master File Table (MFT)

- Файловая система NTFS хранит информацию о файлах и каталогах в главной таблице файлов (MFT).
- Таблица содержит информацию о каждом файле и каталоге в файловой системе.
- Каждый файл или каталог имеет по крайней мере одну запись в MFT.

MFT Records



- Записи MFT адресуются в 48-битной системе.
- Все записи MFT нумеруются.
- Каждая запись содержит 16-битное поле Sequence number, которое означает сколько раз MFT запись использовалась повторно.
- Поле Sequence number увеличивается каждый раз, когда файл удаляется и запись используется повторно для нового файла.
- Уникальный идентификатор файла это сумма порядкового номера записи в MFT таблице и поля Sequence number внутри самой записи.
- Уникальный идентификатор файла меняется при копировании

Уникальный идентификатор файла внутри тома

```
BOOL GetFileInformationByHandle(
  [in]  HANDLE                        hFile,
  [out] LPBY_HANDLE_FILE_INFORMATION lpFileInformation
);
```

Имя поля структуры	Значение
fileInfo.nFileIndexLow	порядковый номер записи в MFT таблице
fiIeInfo.nFileIndexHigh	поле Sequence number внутри самой записи

- Функция GetFileInformationByHandle возвращает необходимые данные в структуре BY_HANDLE_FILE_INFORMATION.
- Последние два поля в структуре позволяют вычислить уникальный идентификатор файла внутри NTFS тома.
- FileId = ((FileInfo.nFileIndexHigh) <<32) + FileInfo. nFileIndexLow) & 0x0000FFFFFFFFFFFF

Как меняется идентификатор файла в реальности

```
for ( ;; )
{
  HANDLE h = CreateFileW(_T("C:\\tmp\\test.mft"), GENERIC_WRITE, 0, NULL, CREATE_ALWAYS, 0, NULL);

  BY_HANDLE_FILE_INFORMATION fi;
  GetFileInformationByHandle(h, &fi);

  printf("h=0x%08X l=0x%08X\n", fi.nFileIndexHigh, fi.nFileIndexLow);
  CloseHandle(h);
  DeleteFileW((const wchar_t*)_T("C:\\tmp\\test.mft"));
}
```

Порядковый номер записи	Поле Sequence number
0x0000C72B	0xFFFC0000
0x0000C72B	0xFFFD0000
0x0000C72B	0xFFFE0000
0x0000C72B	0xFFFF0000
0x0000C72B	0x00010000
0x0000C72B	0x00020000
0x0000C72B	0x00030000

Порядковый номер записи не меняется при удалении и повторном создании файла, т.к. запись в MFT таблице используется повторно. Это плохо для защиты.

Алгоритм защиты

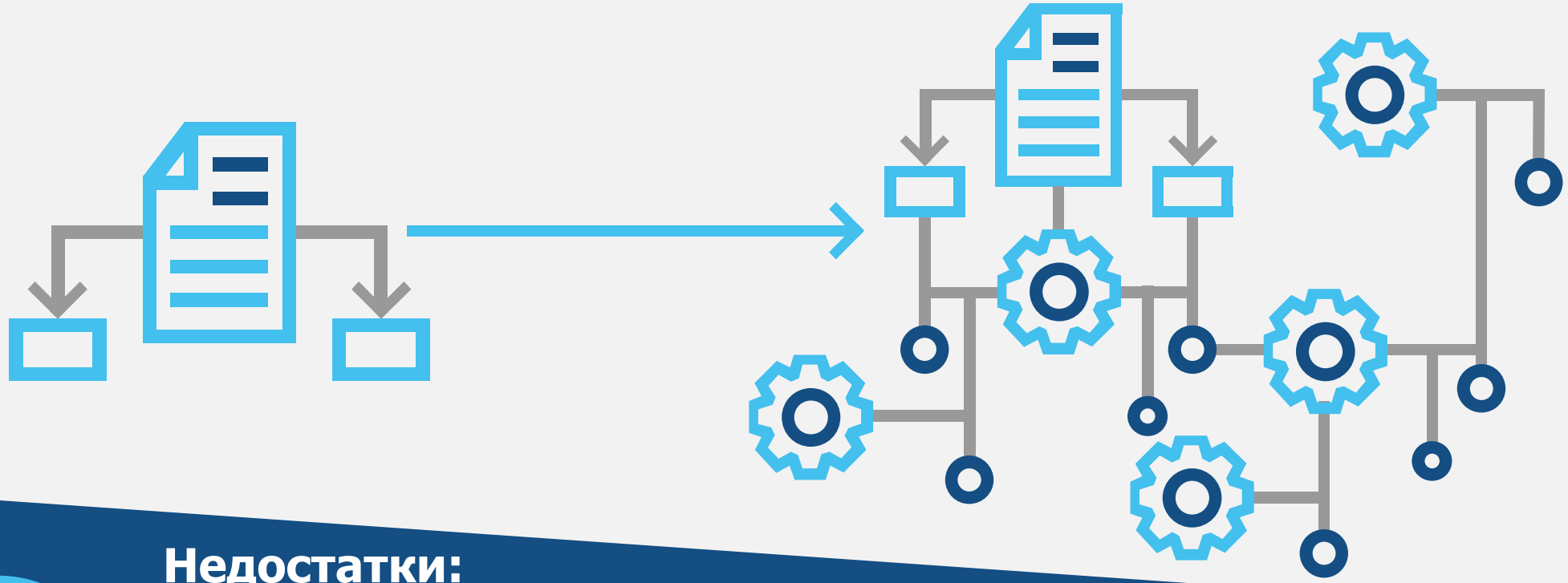
Проверка лицензии

1. Прочитать идентификатор файла лицензии
2. Сравнить идентификатор с ранее сохранённым в файле лицензии
3. Если значения не совпадают, то файл был скопирован

Вычитание счётчика запусков

1. Проверить счётчик запусков на нулевое значение
2. Удалить и повторно создать файл лицензии с уменьшенным значением счётчика
3. Прочитать новый идентификатор файла лицензии
4. Сохранить новое значение идентификатора в файле лицензии

Алгоритм защиты

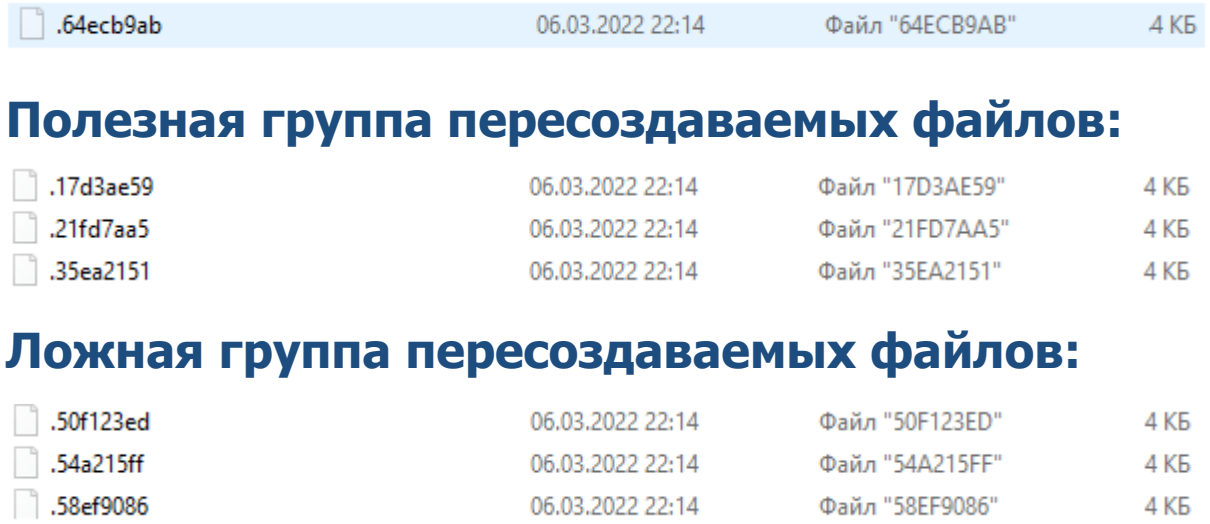


Недостатки:

- Имя файла лицензии не меняется
- Порядковый номер записи в MFT не меняется
- FileMon позволят легко фильтровать обращения к файлу по имени и записи в MFT
- Легко подменить идентификатор файла в момент обращения к WinAPI
- Целостность может быть нарушена в момент повторного создания файла

Усложняем алгоритм привязки к идентификатору файла

Основной файл лицензии:



- Лицензия состоит из множества файлов
- Данные хранятся в основном файле
- Отдельные группы файлов для привязки к уникальным идентификаторам
- Имена всех файлов в группах случайны
- Группа может быть как полезной так и ложной
- Уникальные идентификаторы из MFT считываются для всех файлов
- Несколько групп пересоздаваемых файлов для обеспечения отказоустойчивости

Полиморфные функции и SHA2 для вычисления хеш-суммы

Пример

```
ADD EAX, 87eb0976
SUB EAX, a365f123
ROL EAX, 92847adf
XOR EAX, 1a372047
```

Хеш-сумма уникальных идентификаторов файлов

- Генерируем случайное количество полиморфных функций
- Вычисляем хеш-сумму от идентификаторов полезной группы пересоздаваемых файлов
- Используем полиморфную функцию и SHA2

Контрольные суммы файлов

- Используем полиморфную функцию для вычисления контрольных сумм файлов
- Вычисляем контрольные суммы для всех пересоздаваемых файлов

Защита алгоритма протектором

- Защитить все функции алгоритма проверки лицензии протектором
- Желательно также защитить другие функции приложения, которые будут напрямую работать с алгоритмом
- Заставить реверсировать алгоритм защиты полностью

Выводы

- Данный способ использует документированный функционал файловой системы
- Алгоритм защиты работает на практике
- Рассмотрен универсальный подход, который применяется в большинстве случаев при построении защит

Вопросы

guardant

guardant

bma@guardant.ru

+7 903 198-23-39

www.guardant.ru