kaspersky

Стеганография

Материал для учителя

kaspersky

Стеганография

Стеганография (от греч. Στεγανός – скрытый + γράφω – пишу; буквально «тайнопись») – наука, позволяющая спрятать передаваемые данные в некотором контейнере, таким образом скрыв сам факт передачи информации.

Контейнер (стегоконтейнер) – любой объект, используемый для тайного внедрения сообщения.

В отличие от криптографии, которая скрывает содержимое тайного сообщения, стеганография скрывает сам факт его существования. Преимущество стеганографии над чистой криптографией состоит в том, что сообщения не привлекают к себе внимания. Сообщения, факт шифрования которых не скрыт, вызывают подозрение и могут быть сами по себе уличающими в тех странах, в которых запрещена криптография.

Спрятать секретное послание можно практически в любой цифровой объект – текстовый документ, лицензионный ключ, расширение файла. Однако один из самых удобных «контейнеров» – медиафайлы (картинки, аудио, видео и так далее). Они обычно достаточно большие сами по себе, а значит, и скрываемая информация может быть не такой маленькой, как в случае, скажем, с документом Word.

Секретную информацию можно записать в метаданные файла или же прямиком в основное содержимое. Возьмем, например, картинку. С точки зрения компьютера она представляет собой набор из сотен тысяч точек-пикселей. У каждого пикселя есть «описание» – информация о его цвете.

Для формата RGB, который используется в большинстве цветных картинок, это описание занимает в памяти 24 бита. Если в описании некоторых или даже всех точек 1–3 бита будет занято секретной информацией, на картинке в целом изменения будут неразличимы. А за счет огромного числа пикселей всего в изображение вписать можно довольно много данных.

В большинстве случаев прячут информацию в пиксели и извлекают ее оттуда при помощи специальных утилит. Иногда для этой цели пишут собственные скрипты или добавляют нужную функциональность в программы другого назначения. А иногда пользуются готовыми кодами, которых в сети немало.

Исторически впервые понятие стеганографии было введено в 1499 году, но сам метод существовал очень давно. Легенды донесли до нас метод, который использовался в Римской империи: для доставки сообщения выбирали раба, голову которого брили, а затем с помощью татуировки наносили текст. После того как волосы отрастали, раба отправляли в путь. Получатель сообщения снова обривал голову раба и читал сообщение.

В течение всего XX века активно развивалась как стеганография, так и наука об определении факта внедрения информации в контейнер – стегоанализ.

На сегодняшний момент наблюдается новый и опасный тренд: все больше разработчиков вредоносного ПО и средств кибершпионажа прибегает к использованию стеганографии. Большинство антивирусных решений не защищают от стеганографии или защищают слабо, меж тем, нужно понимать, что каждый заполненный контейнер опасен. В нем могут быть скрыты данные, которые эксфильтруются шпионским ПО, или коммуникация вредоносного ПО с командным центром, или новые модули вредоносного ПО.

Качественную стеганографию распознать крайне сложно. Избавиться от нее тоже не так-то просто: есть методы, позволяющие вшить сообщение в картинку настолько глубоко, что оно сохранится даже после того, как ее напечатают и снова отсканируют, уменьшат, увеличат или еще как-то отредактируют.

Однако решения есть, они основаны на комбинировании различных способов анализа, высокоскоростных предетектах, исследовании метаданных потенциально заполненного контейнера и т.п. Такие решения есть у Лаборатории Касперского – платформа защиты от таргетированных атак – KATA. Ее использование позволяет сотруднику службы информационной безопасности своевременно узнать о возможной таргетированной атаке на защищаемый периметр и/или эксфильтрации данных из него.

За недавнее время наблюдается использование стеганографии в следующих вредоносных программах и средствах кибершпионажа: Microcin (AKA six little monkeys); NetTraveler; Zberp; Enfal (its new loader called Zero.T); Shamoon; KinS; ZeusVM; Triton (Fibbit).

На сегодня учеными разработаны и опробованы различные алгоритмы и методы стеганографии

Вот некоторые из них:

- LSB-стеганография – сообщение скрывается в младших битах (возможно использование одного или нескольких младших бит) контейнера. Чем меньше битов задействовано, тем меньше артефактов получает оригинальный контейнер после внедрения.
- Метод сокрытия информации при помощи младших бит палитры – этот метод по сути является вариантом общего метода LSB, но информация встраивается не в наименее значащие биты контейнера, а в наименее значащие биты палитры, очевидный недостаток такого метода – низкая емкость контейнера.
- Метод сокрытия информации в служебных полях формата – метод, основанный на использовании служебных полей заголовка контейнера для хранения сообщения. Очевидные минусы – низкая емкость контейнера и возможность обнаружения внедренных данных при помощи обычных программ для просмотра изображения (которые иногда позволяют видеть содержимое служебных полей).
- Метод встраивания сообщения – заключается в том, что сообщение встраивается в контейнер, затем при помощи схемы, известной обеим сторонам, извлекается. Можно встроить несколько сообщений в один контейнер, при условии, что способы их внедрения ортогональны.
- Широкополосные методы, которые подразделяются на:
    - метод псевдослучайной последовательности, используется секретный сигнал, который моделируется псевдослучайным сигналом.
    - метод прыгающих частот: частота несущего сигнала меняется по определенному псевдослучайному закону.
- Метод оверлея – по сути не является настоящей стеганографией, основан на том, что некоторые форматы содержат в заголовке размер данных, или же обработчик этих форматов будет читать файл до маркера конца данных. Примером такого метода является метод «rar-jpeg», который основан на конкатенации графического файла в формате JREG и RAR-архива. ПО для просмотра JPEG будет считывать информацию до границы, указанной в заголовке файла, а RAR-архиватор откинет все, что находится до сигнатуры «RAR!», которая обозначает начало архива. Таким образом, если такой файл открыть в программе для просмотра графических файлов, то увидим картинку, а если в RAR-архиваторе – содержимое RAR-архива. Минусы такого подхода заключаются в том, что оверлей, добавленный к контейнеру, легко выделяем при визуальном исследовании такого файла.

Почему авторы вредоносного ПО все активней используют стеганографию в своих разработках?

Выделим три главные причины:

- Это позволяет им скрыть сам факт загрузки/выгрузки данных, а не только сами данные;
- Помогает обойти DPI-системы, что актуально в корпоративных сетях;
- Использование стеганографии может позволить обойти проверку в AntiAPT-продуктах, поскольку последние не могут обрабатывать все графические файлы (их слишком много в корпоративных сетях, а алгоритмы анализа довольно дорогие).

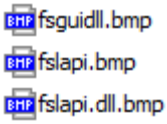
kaspersky

Zero.T

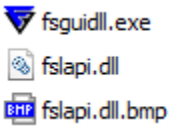
Этот загрузчик был обнаружен Лабораторией Касперского в конце 2016 года, но первое описание было опубликовано компанией Proofpoint.

Угрозу назвали Zero.T, так как такая строка присутствует в его исполняемом коде (в пути к pdb-файлу проекта):

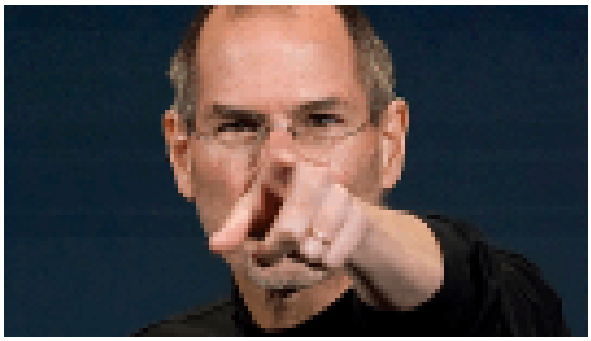
Не останавливаясь на попадании в систему и закреплении в ней, отметим, что Zero.T скачивает полезную нагрузку в виде Bitmap-файлов:



Обрабатывает их особым образом, после чего получает вредоносные модули:



На поверку эти три BMP-файла оказались картинками:



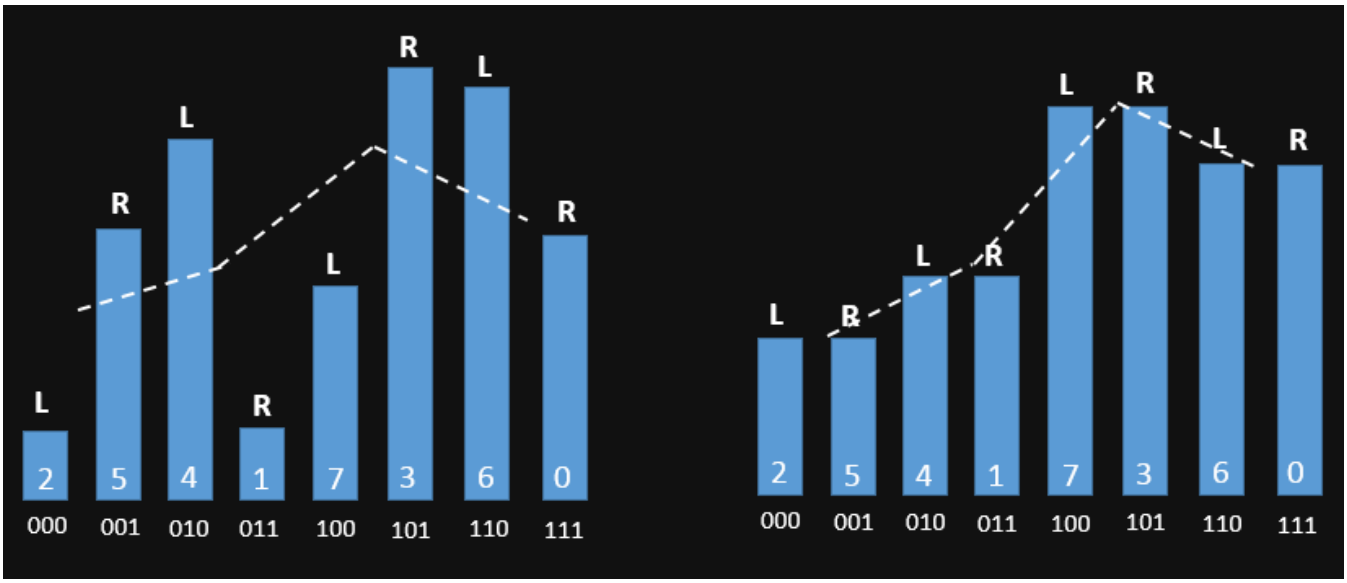
Но картинки эти не совсем обычные – заполненные контейнеры, в каждом из которых несколько (алгоритм допускает вариативность) младших значащих бит заменены на полезную нагрузку.

Как же определить является ли картинка заполненным контейнером или нет?

Рассмотрим статистический метод анализа – гистограммный метод.

Описываемый метод, предложенный в 2000 году Андресом Вестфелдом и Андреасом Пфитцманом, также известен как «хи-квадрат»-метод. Попытаемся изложить его суть.

Весь растр анализируется, для каждого цвета считается количество точек такого цвета в растре (для простоты здесь говорим про изображение, имеющее одну цветовую плоскость).



а – пустой контейнер

б – заполненный контейнер

Метод исходит из предположения, что количество точек двух соседних цветов («соседние» цвета – цвета, которые отличаются только наименее значимым битом) различается существенно для нормального, обычного изображения (пустого контейнера). И количество пикселей таких цветов является примерно одинаковым для заполненного контейнера.

www.kaspersky.com/
www.securelist.com