**ProcessWatcher**

**Основные термины:**

1. Жертва – компьютер, на котором отслеживают создание и завершение процессов.
2. Наблюдатель – компьютер, который отслеживает активность жертвы или нескольких жертв.

Проект состоит из 3 основных частей:

1. Драйвер - программа, собирающие данные, на *жертве*.
2. Сервис - программа, слушающая команды *наблюдателя* и выполняющая пересылку данных с *жертвы* на *наблюдателя*.
3. Хозяин - программа, представляющая интерфейс пользователя на *наблюдателе*, позволяющая запросить данные с *жертв* и показать их.

Рассмотрим каждую часть более подробно.

1. **Драйвер.**

Драйвер представляет собой драйвер, использующий функцию ядра PsSetCreateProcessNotifyRoutine. Данный драйвер предоставляет возможность передать ему call-back функцию, в которую передаётся информация о запущенном или завершённом процессе. Возможное решение представлено на <http://www.codeproject.com/Articles/2018/Detecting-Windows-NT-K-process-execution> . После получения данных необходимо их сохранить в файле на *жертве*. Файл будет бинарным и иметь структуру, описанную ниже. Все файлы будут находиться в папке %application\_data%\ProcW. Новый файл будет создаваться при каждом запуске драйвера (запуске компьютера, так как драйвер в автозагрузке). Имя файла будет соответствовать Guid.

1. **Сервис.**

Сервис представляет собой либо просто программу, либо сервис, который слушает команды хозяина. Если приходит команда на получение данных, то она должна отправлять необходимые данные в формате, описанном ниже. Также сервис должен выбирать для чтения только те файлы, в которых хранятся данные, удовлетворяющие временным рамкам, пришедшим в запросе от хозяина.

PID – уникальный идентификатор процесса в системе **во время выполнения процесса**. Так как в файлах лежат последовательно записи о начале и конце процесса, то нужно найти начало и конец процесса и сохранить в 1 структуру для дальнейшей передачи. Создание процесса точно будет, но не надо исключать его отсутствие, так как драйвер может быть запущен не самым первым при загрузке системы. А вот завершения процесса вполне может и не быть, в этом случае можно писать дату начала старта системы или можно не отправлять информацию о таком процессе вообще (то есть считать его не валидным). Можно использовать словарь в виде стека. Помещать в него информацию о старте процесса, а при прочтении информации о конце процесса, находить соответствующий процесс

1. **Хозяин.**

Оконная программа, определяющая список компьютеров-жертв, которые находятся в локальной сети, и позволяющая получить с них данные о запускаемых процессах в указанный период времени. <http://www.cyberforum.ru/csharp-net/thread248538.html>

Формат файла, в котором хранится информация о запущенных процессах:

1. Идентификатор процесса (4 байта)
2. Имя процесса (n байт) (при сохранении с помощью BinaryWriter в начало записывается длина строки, поэтому вручную её записывать не нужно)
3. Дата старта или завершения процесса (8 байт)
4. Идентификатор старта или завершения процесса (1 байт)

Скорее всего, имя будем писать только при старте процесса, а при завершении только идентификатор и дату.

Формат команды запроса данных:

1. Дата from
2. Дата to (может быть равна 01.01.0001 00:00:00.000)

Формат данных для пересылки:

1. Длина имени процесса (4 байта)
2. Имя процесса (n байт)
3. Дата старта процесса (8 байт)
4. Дата завершения процесса (8 байт может быть равна 01.01.0001 00:00:00.000)