**Обязательный контроль целостности** (Mandatory Integrity Control) – функция ОС Windows (начиная с Vista), предоставляющая механизм управления доступом к охраняемым объектам. Для определения прав доступа к объектам используются уровни целостности: системный, высокий, средний и низкий.

У каждого файла/директории NTFS есть соответствующая запись в Master File Table, в которой содержится дескриптор безопасности. Каждый такой дескриптор содержит **SACL** (System access-control list) и **DACL** (Discretionary access-control list) списки контроля доступа. Первым списком управляет, как понятно из названия, система, второй можно настраивать, указывая настройки доступа к файлам/директориям. Также списки контроля доступа есть и у записей реестра.

Списки ACL – это наборы записей **ACE** (Access control entries), в которых указывается:

* **SID** (Security ID) пользователя либо группы, к которым применяется правило доступа
* Вид доступа
* Тип ACE (разрешающий/запрещающий)

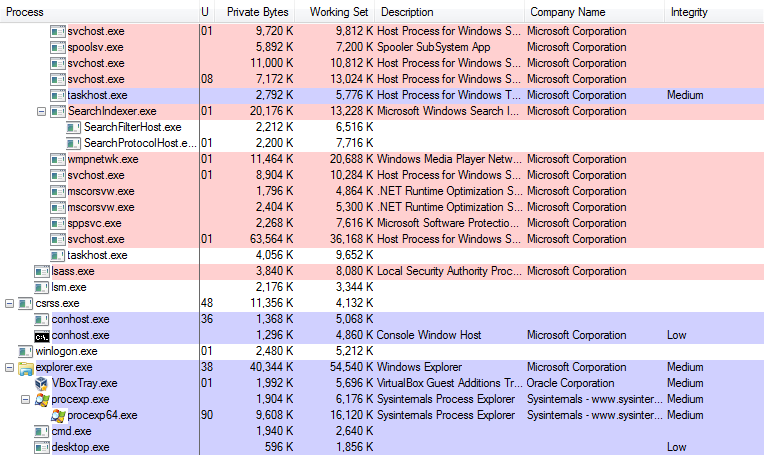
Наиболее интересен в смысле уровней целостности SID, поскольку часть его – это и есть уровень целостности, который указывается в ACE для пользователей/групп:

* 0x1000 – low integrity
* 0x2000 – medium integrity
* 0x3000 – high integrity
* 0x4000 – system

Процессы также имеют уровень целостности – более того, в отличие от файла или директории, у процесса он обязательно должен быть. Каждому выполняющемуся процессу соответствует **Access token**, в котором содержится информация о контексте безопасности процесса – в том числе, данные о пользователе, который создал процесс.

Если у исполняемого файла не указан уровень целостности, то процесс запустится со средним уровнем. Большинство пользовательских процессов также по умолчанию запускается со средним уровнем, даже если исполняемый файл имеет высокий уровень целостности.

Процессы с низким уровнем целостности заметно ограничены во взаимодействии с файловой системой и с другими процессами – запись возможна лишь в обозначенные файлы (процессу может быть выделен файл или директория). Запись в память процесса с более высоким уровнем целостности недопустима, иногда запрещается даже чтение из памяти других процессов.



**Эскалация уровня целостности процесса.**

Одним из ключевых методов, которым можно добиться повышения уровня целостности процессов – внедрение DLL в другой процесс и выполнение действий, предполагающих автоматическую эскалацию привилегий.

**Component Object Model** – стандарт для создания ПО, разработанный Microsoft – позволяет создавать такие объекты, которые автоматически увеличивают свой уровень целостности, если запускаются из программ, подписанных сертификатом Microsoft. Объекты, созданные таким образом, позволяют взаимодействовать с процессами или файлами, используя высокие привилегии.

**Bypass UAC (User account control) attack** при помощи внедрения DLL в приложение notepad позволяет создать COM-объект iFileOperation, у которого есть методы для копирования и удаления файлов. Использование такого объекта с высоким уровнем целостности делает возможным запись файла в любую директорию.

Используя другую библиотеку DLL, содержащую в себе вредоносную программу, можно осуществить подмену, принцип действия которой основывается на порядке, в котором осуществляется поиск библиотек процессами встроенных в Windows программ (эти программы также запрашивают более высокий уровень целостности). Когда библиотека будет загружена, вредоносный код можно будет выполнить.

Однако данный метод требует наличия у процесса среднего уровня целостности – интегрировать DLL в процесс notepad из процесса низкого уровня не получится. Также требуется, чтобы UAC не был установлен в “Always notify”, когда программы пытаются внести изменения.

Альтернативный метод использует Access tokens для повышения уровня целостности процесса, поскольку они используются для запуска процессов текущим пользователем. Существует два типа AT: primary token и impersonate token. Последние позволяют потоку приложения взаимодействовать с защищаемыми объектами, используя его контекст безопасности.

Повышение уровня целостности основывается на том, что, осуществив олицетворение на нужном уровне, можно взаимодействовать с объектами, уровень целостности которых выше, чем у исходного процесса.

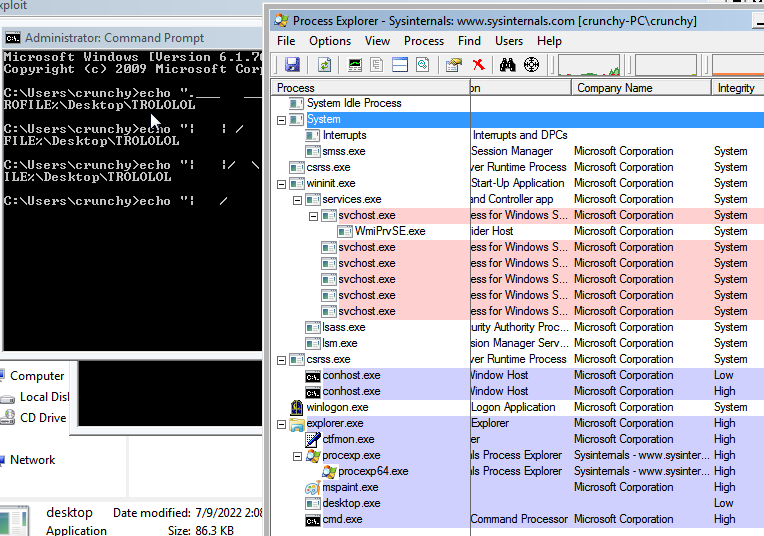
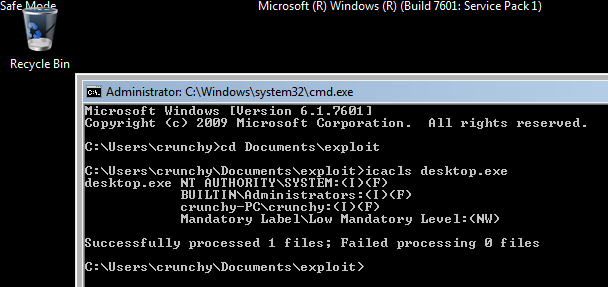
Третий метод базируется на поиске уязвимостей в компонентах системы, таких как драйверы или службы Windows. Использование таких уязвимостей может затрагивать даже процессы, работающие в режиме ядра (как, например, драйвер win32k.sys – уязвимость CVE-2013-0008). Используя такие уязвимости, можно запускать процессы с более высоким уровнем целостности, либо имеющие доступ к управлению процессами более высокого уровня (leaked handles).

**Эксплойт CVE-2013-0008**

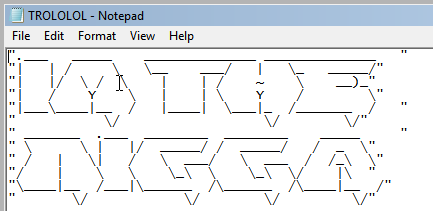
Данный эксплойт использует симуляцию нажатия клавиш для запуска процесса с более высоким уровнем целостности (изначально запускается командная строка с низким уровнем, затем проводнику даётся команда создать процесс, идентичный присутствующему в панели задач). Суть эксплойта – в использовании функции SendMessage, осуществляющей передачу сообщений между процессами. В связи с неправильной обработкой сообщений широкого вещания (HWND\_BROADCAST) запущенный с низким уровнем процесс может передавать сообщения не только самостоятельным процессам (не имеющим родительского процесса), но и запущенной по его инициативе командной строке со средним уровнем целостности. Таким образом в командную строку передаются команды, производящие запись в файл на рабочем столе (если его не существует, он создаётся) – по крайней мере, в демонстрационной версии эксплойта.

Эксплойт протестирован в Windows 7 SP1 (Build 7601) 64bit. Судя по данным из бюллетеня [MS13-005](https://docs.microsoft.com/ru-ru/security-updates/securitybulletins/2013/ms13-005), уязвимости подвержены также 32bit версии Windows 7 и версии без Service Pack 1. User account control не реагирует на запуск программы, если не установлен параметр «Always notify».

Даже в безопасном режиме эксплойт выполняется успешно.



Получен следующий результат:



**Эксплойт CVE-2019-1184**

Данная уязвимость позволяет процессу с низким уровнем целостности исполнять код со средним уровнем целостности. Класс COM с именем CoreShellCOMServerRegistrar при попытке его запуска в качестве локального сервера хранится как экземпляр внутри процесса sihost.exe (Shell Infrastructure Host). Данный процесс запускается системой, когда вход в систему осуществляется с участием пользователя («обычный» вход в систему – interactive user logon), со средним уровнем целостности.

sihost.exe позволяет процессам «интерактивного» пользователя запускать CoreShellCOMServerRegistrar даже при наличии у них низкого уровня целостности. Поскольку этот класс имеет методы OpenProcess и DuplicateHandle, можно получить доступ к управлению sihost.exe (передать в вызывающий процесс HANDLE), после чего осуществлять запись в память процесса sihost.exe и, запуская отдельный поток, выполняющий записанный код, добиваться выполнения этого кода со средним уровнем целостности.

Эксплойт, представленный 0vercl0k, запускает калькулятор как дочерний процесс svhost.exe, который имеет метку «App container», но запуск его осуществляется при помощи sihost.exe, доступ к которому получает процесс с низким уровнем целостности.

Эксплойт протестирован в Windows 10 Pro (v1809) 64bit (версии до 08.2019 имели эту уязвимость).

