Министерство образования и науки Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
—
Институт компьютерных наук и технологий
Кафедра «Информационная безопасность компьютерных систем»

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

«Изучение механизмов безопасности ОС Windows»

по дисциплине «Модели безопасности компьютерных систем»

_____ Семёнов П.О.

Выполнили _____ Проценко Е.Г.
студенты гр. 33508/3

_____ Латышев М.А.

Руководитель _____ Жуковский Е.В.

Санкт-Петербург
2016

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ

Данная лабораторная работа заключается в разработке программы с графическим интерфейсом, которая должная осуществлять вывод и изменение различных прав субъектов по отношению к различным объектам ОС Windows, а также другой информации о процессах, функционирующих в ОС. Работа выполняется студентами в группах по 2-3 человека. При разработке программы стоит ориентироваться на подходы, заложенные в программах со схожим функционалом, таких как ProcessExplorer и ProcessHacker.

В ходе выполнения лабораторной работы необходимо выполнить следующие действия:

1. Изучить механизмы контроля доступа в ОС Windows (списки контроля доступа, политики целостности, привилегии субъектов, уровни целостности, роли, права владельца и другие).
2. Разработать программу с графическим интерфейсом, собирающую информацию об объектах и субъектах ОС Windows.
3. Разработанная программа должна выводить список процессов, функционирующих в операционной системе и следующую информацию о каждом из процессов;
   a. имя процесса;
   b. идентификатор процесса (PID);
   c. путь до исполняемого файла;
   d. имя и идентификатор родительского процесса;
   e. имя пользователя владельца процесса, SID;
   f. тип процесса (32-bit / 64-bit);
   g. использование DEP, ASLR;
   h. список используемых динамических библиотек (DLL);
   i. дополнительная информация.
4. Разработанная программа должна выводить и изменять следующую информацию о безопасности процесса:

a. уровень целостности (integrity level);

b. привилегии (privileges);

5. Разработанная программа должна выводить и изменять следующую информацию о любом заданном объекте файловой системы:

a. список контроля доступа (ACL);

b. владелец файла;

c. уровень целостности;

2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1 Handle and Object

Объект – это структура данных, описывающая такие системные ресурсы, как файлы, потоки, изображения и т.д. Приложение не может напрямую получить данные процесса или системные ресурсы, которые объект описывает. Вместо этого, приложение должно получить handle процесса, который тот может использовать для изучения или модификации системного ресурса. Каждый handle имеет запись в таблице, которая хранится в системе. Эти записи содержат адреса этих ресурсов и определяют тип ресурса.

Система использует объекты и handles, чтобы регулировать доступ к системным ресурсам по двум главным причинам. Во-первых, использование объектов обеспечивает то, что Microsoft может обновлять функционал системы, пока используется такой тип доступа к объектам. При релизе последующих версий системы, вы можете использовать обновленные объекты с минимальными дополнительными затратами или вовсе без них.

Во-вторых, использование объектов предоставляет возможность пользоваться преимуществами системы безопасности Windows. Каждый объект имеет собственный ACL, который определяет какие действия процесс может совершать над объектом. Система обращается к ACL объекта каждый раз, когда приложение создает handle объекта.

2.2 Контроль доступа

Контроль доступа относится к функционалу системы безопасности, которая контролирует, кто может получить доступ к ресурсам в операционной системе. Приложение вызывает функции контроля доступа, чтобы обозначить, кто может получить доступ к определенным ресурсам или урегулировать доступ к ресурсам, которые предоставляет приложение.

Модель контроля доступа позволяет вам контролировать способность процессов получать доступ к (защищенным) объекта или выполнять различные задачи системного администрирования.

Существует две основные части модели контроля доступа:

- **access token (маркер доступа),** который содержит информацию об авторизированном пользователе;
- **security descriptor (дескриптор безопасности),** которой содержит security information которая защищает (защищаемый) объект

Когда пользователь заходит в систему, система аутентифицирует имя аккаунта пользователя и пароль. Если вход в систему выполнен успешно, система создает маркер доступа. Каждый процесс запускаемый от имени этого пользователя будет иметь копию этого маркер доступа. **Маркер доступа содержит идентификаторы безопасности,** которые идентифицируют аккаунт пользователя и все группы аккаунтов, в которых состоит пользователь. **Token также содержит список привилегий,** принадлежащих пользователю или группам пользователя. Система использует этот token для идентификации ассоциированного пользователя, когда процесс пытается получить доступ к защищаемому объекту или выполнить задачу системного администрирования, которая требует привилегий.

Когда защищаемый объект создается, система назначает ему дескриптор безопасности**, который содержит информацию безопасности**

его создателя. Приложения могут использовать функции для получения и назначения security information для существующего объекта.

Дескриптор безопасности идентифицирует владельца файла и может также содержать следующие access control lists:

- discretionary access control list (DACL), который идентифицирует пользователей и группы, которым разрешен или запрещен доступ к объекту;
- system access control list (SACL), который контролирует как система регулирует попытки получения доступа к объекту.

ACL содержит список **access control entries (ACEs).** Каждый ACE определяет набор **access rights** и содержит SID, который идентифицирует **trustee,** которому rights are allowed, denied, or audited. **Trustee** может быть пользовательским аккаунтом, групповым аккаунтом или logon session.

Рекомендуется использовать функции для манипуляции над содержимым security descriptors, SIDs и ACLs, а не получать доступ к ним напрямую. Это помогает убедиться, что эти структуры останутся синтаксически точными и предотвратят будущие усовершенствования системы безопасности от поломки существующего кода.

2.3 Маркеры доступа

Access token описывает **security context** процесса или потока. Информация в token включает в себя идентификатор и привилегии пользовательского аккаунта ассоциированного с процессом или потоком. Когда пользователь заходит в систему, последняя подтверждает пароль сравнивая его с информацией, хранящейся в security database. Если пароль аутентифицирован, система создает access token. Каждый процесс, запущенный от имени этого пользователя имеет копию такого access token.

Система использует access token, чтобы идентифицировать пользователя, когда поток взаимодействует с securable object или пытается выполнить системные задачи, которые требуют привилегий. Access token содержит следующую информацию:

- **Security identifier (SID)** аккаунты пользователя
- SIDs для групп, членом которой является пользователь;
- **Logon SID**, который идентифицирует текущую **logon session**;
- Список **privileges**, которые имеет пользователь или группы пользователя;
- SID владельца;
- SID основной группы;
- Default **DACL,** который система использует, когда пользователь создает securable object;
- Источник access token;
- **Primary** или **Impersonation** token;
- Возможный список запрещенных **SID**;
- Текущий уровень имперсонации;
- Другая статистика;

Каждый процесс имеет **primary token,** который описывает **security context** аккаунта пользователя, который ассоциирован с процессом. По умолчанию система использует primary token, когда поток процесса

взаимодействует с securable object. Более того, поток может имперсонировать аккаунт клиента. Имперсонация позволяет потоку взаимодействовать с securable objects, используя **security context** клиента. Поток, который имперсонирует клиента или оба токена: и primary token, и impersonation token.

2.4 Дескрипторы безопасности

Security descriptor содержит security information, ассоциированную с securable object. Security descriptor состоит из SECURITY_DESCRIPTOR structure, которая ассоциирована с security information. Security descriptor может включать следующую security information:

- Security identifier (SIDs) для владельца и primary group of an object.
- DACL, которая определяет права предоставленный или запрещенные у конкретного пользователя или группы.
- SACL, которая определяет тип попыток получения доступа и регулирует обращения к объекту.
- Набор управляющих битов, определяющих дексриптор безопасности или его части.

Приложение не может напрямую манипулировать содержанием security descriptor. WinAPI предоставляет функции для настройки и получения security information of security descriptor. Так же, существуют функции для создания и инициализации security descriptor для нового объекта.

Приложение, работающее с security descriptor на **Active Directory** objects может использовать Windows security functions или security interfaces, предоставляемые by **Active Directory Service Interface (ADSI).**

2.5 Access Control Lists

Access control list (ACL) – это список **access control entries (ACE)**. Каждый ACE определяет набор **access rights** и содержит SID, который идентифицирует **trustee** и определяет права, которые разрешены, запрещены или отслеживаются для trustee. Дескриптор безопасности для объекта может содержать два типа ACLs: DACL и SACL.

A discretionary access control list (DACL) идентифицирует trustee, которым разрешается или запрещается доступ к securable object. Когда процесс пытается получить доступ к securable object, система проверяет ACEs в DACL объекта, чтобы определить разрешить ли доступ процессу к объекту. Если объект не имеет DACL, система предоставляет full access всем. Если DACL не содержит ACEs, система запрещает все попытки доступа к объекту, потому что DACL не предоставляет каких-либо access rights. Система проверяет ACEs по порядку, пока не найдет одну или более ACEs, которые предоставят все запрошенные access rights, или пока любые из запрашиваемых прав не будут denied.

A **system access control list (SACL),** позволяет администраторам регистрировать попытки доступа к защищенному объекту. Каждый ACE определяет тип попыток доступа с помощью указанного trustee, которые заставляют систем генерировать запись в security event log. ACE в SACL может генерировать записи, когда попытка доступа fails, when succeeds, или при обоих событиях.

2.5.1Access Control Entries

ACE – элемент ACL. ACL может иметь 0 или более ACEs. Каждый ACE контролирует доступ к объекту для определенных trustee.

Существуют 6 типов ACEs, три из которых поддерживаются всеми securable objects. Другие 3 – **object-specific ACEs**, которые поддерживаются directory service objects.

Все типы ACEs содержат следующую информацию контроля доступа:

- Security identifier (SID), который определяет trustee к которому ACE относится.
- **Access mask**, которая определяет **access rights,** контролируемые ACE.
- Флаг, обозначающий тип ACE.
- Набор битовых флагов, которые определяют, могут ли дочерние контейнеры или объекты наследовать ACE от основного object, к которому ACL прикреплено.

ACE types:

- Access-denied ACE
- Access-allowed ACE
- System-audit ACE

2.6 Механизм уровней целостности

Механизм целостности в Windows является очень важным компонентом архитектуры безопасности, который ограничивает и регулирует права доступа приложений, которые запущены под одной учетной запись, но имеет разные уровни доверия к ним.

Таким образом механизм целостности расширяет механизм защиты, назначая уровень целостности как процессам, так и объектам.

Уровень целостности показывает насколько сильно система доверяет работающим процессам и объектам.

Уровень целостности предоставляет возможность, например, файловому менеджеру, использовать политики, которые запрещают процессам с низким уровнем целостности читать или модифицировать объекты с более высоким уровнем целостности.

Монитор безопасности обеспечивает контроль доступа сравнивая SID’s пользователя и групп в маркере доступа с mandatory label в SACL объекта.

2.6.1 Уровни целостности процесса

Windows использует SID для определения уровня целостности. S-1-16-xxxx. xxxx – поле относительного идентификатора (RID). RID – шестнадцатеричное значение, которое определяет уровень целостности.

Существует 4 основных уровня целостности: низкий, средний, высоки, системный, а также untrusted.

2.6.2 Политики уровня целостности

Политика уровней целостности определяет какие действия процессу запрещается выполнять над объектом. Для такого механизма существуют специальные флаги, которые запрещают процессам с низкими приоритетами читать, писать, исполнять код объекта.

2.6.3ACE уровня целостности

Механизм целостности определяет новый тип ACE уровня целостности. ACE уровня целостности используется для определения уровня целостности объекта и используется только в SACL.

Security Descriptor Definition Language (SDDL) позволяет работать с уровнями целости в строковом формате. Таким образом можно легко переводить как SID, так и ACE в строку и работать с ними. В SID за уровень целостности отвечают число RID. ACE переводится в строку следующего формата: “S:(ML;;NW;;;LW)”, где ML – сигнатура уровня целостности, NW – флаг политики доступа, LW – уровень целостности.

3 ВЫВОД

В этой работы мы ознакомились с механизмами сбора информации о системе, безопасности Windows. Познакомились с очень важной привилегией SE_DEBUG_NAME, которая позволяет получать handle от любого процесса с PROCESS_ALL_ACCESS правами, что очень важно для дальнейшего сбора информации. Экспериментировали с изменениями прав и привилегий объектов и субъектов для получения различного поведения при их взаимодействии.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. https://msdn.microsoft.com
2. http://stackoverflow.com/
3. Руссинович М.,Соломон Д.,Ионеску А. Внутреннее устройство Microsoft Windows