LPE @ Windows

A chaotic introduction

Спасибо Дмитрию Коробейникову за помощь в изучении windows

Why...

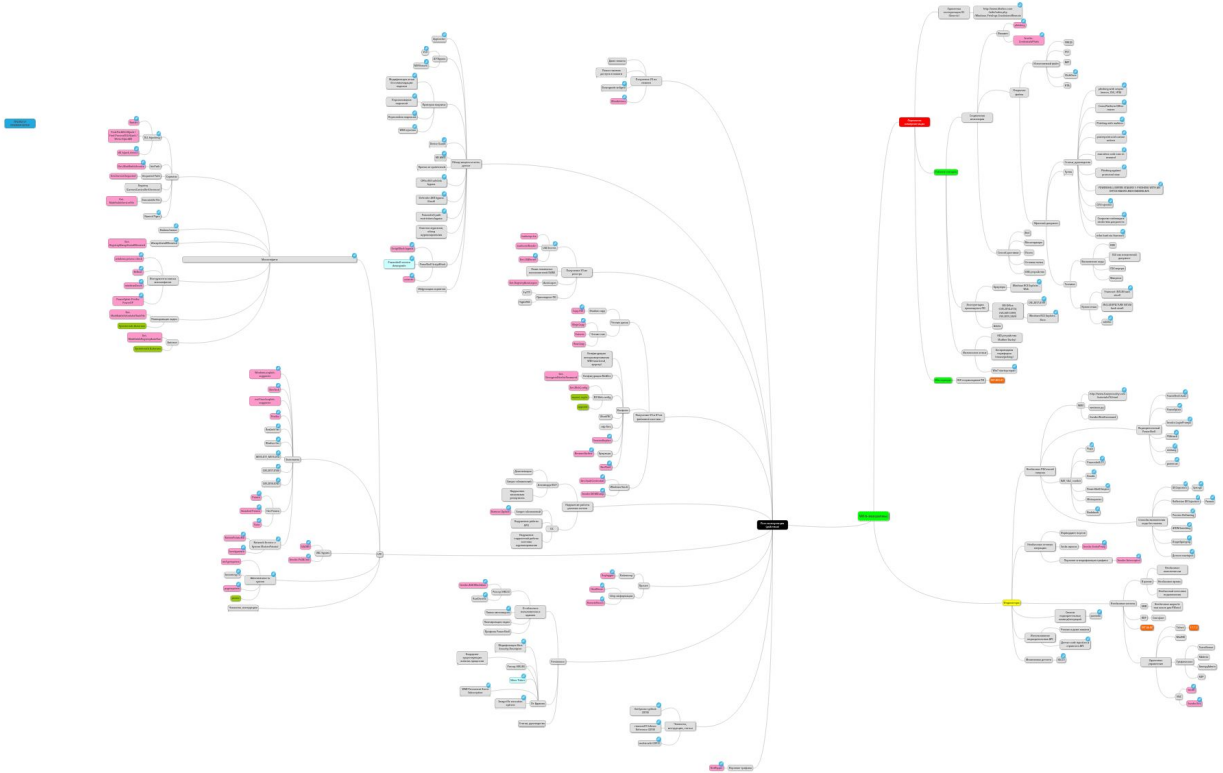
...Windows ?

- Актуально для любых инфраструктур;
- Никуда от нас не денется;
- Еще не охвачено нашими семинарами.

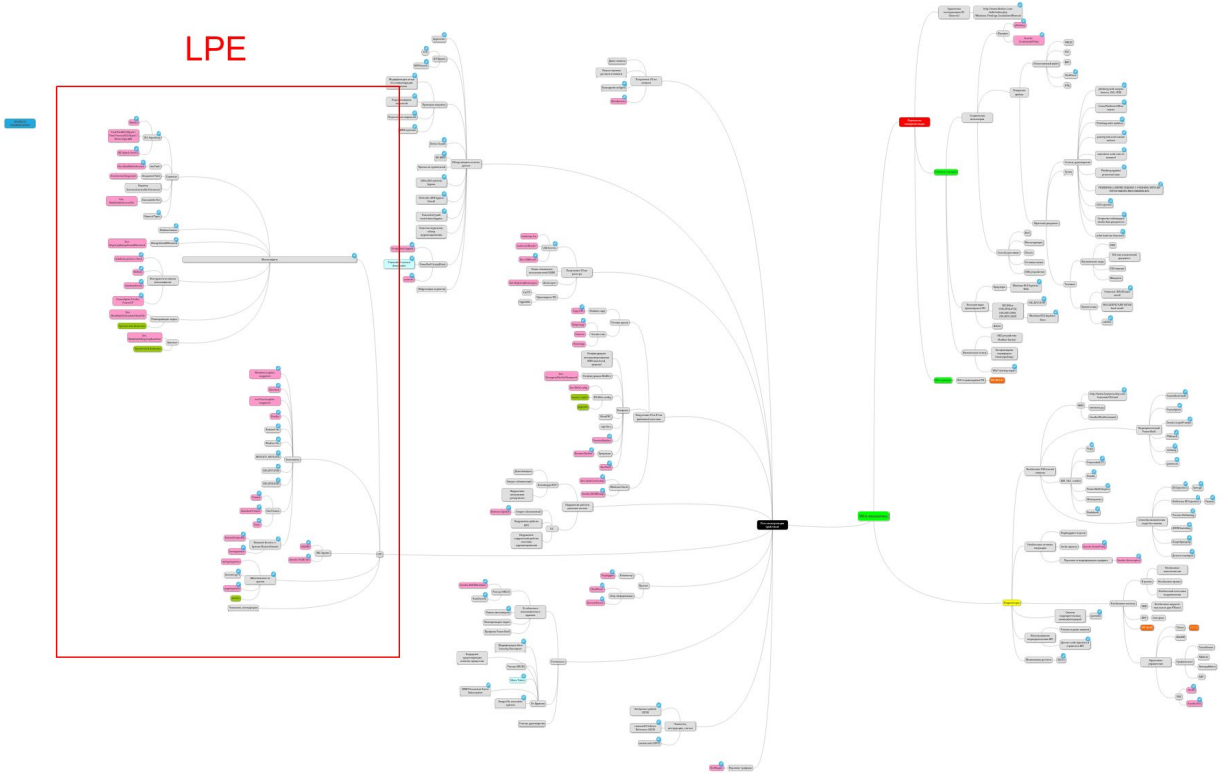
...LPE ?

- Помогает посмотреть, как все устроено;
- Актуально:
  - общие терминальные серверы,
  - рабочие станции обычных пользователей.

Контекст



Контекст



WTF LPE

- Можем выполнять произвольный код от имени какого-то пользователя;

- Хотим выполнить код от имени более привилегированного пользователя.

WTF (Windows) LPE

- Можем выполнять произвольный код от имени какого-то пользователя
- … возможно, с некоторыми ограничениями (см Integrity Level).
- Хотим выполнить код от имени более привилегированного пользователя
- … + обойти какие-то дополнительные механизмы безопасности (например, Protected Process, TrustedInstaller и др).

LPE - зоны ответственности

Disclaimer: “Ребята, не стоит вскрывать эту тему” (С)

- Много материала, сложная архитектура;
- Закрытая система, не везде есть документация;
- Я не эксперт, у меня просто есть немного практического опыта;
- Рассматривайте семинар как пищу для размышлений, чтобы понять, насколько интересно погружаться в win security лично вам.

Agenda

1. Demo - CVE-2019-1388 (UAC LPE via Certificate Properties)
2. Теория - Security Tokens, ACLs, Intergrity Levels
3. CVE-2018-8440 (Task Scheduler ALPC LPE) + Demo
4. LPE via Misconfigs
5. Homework

1. Demo: CVE-2019-1388

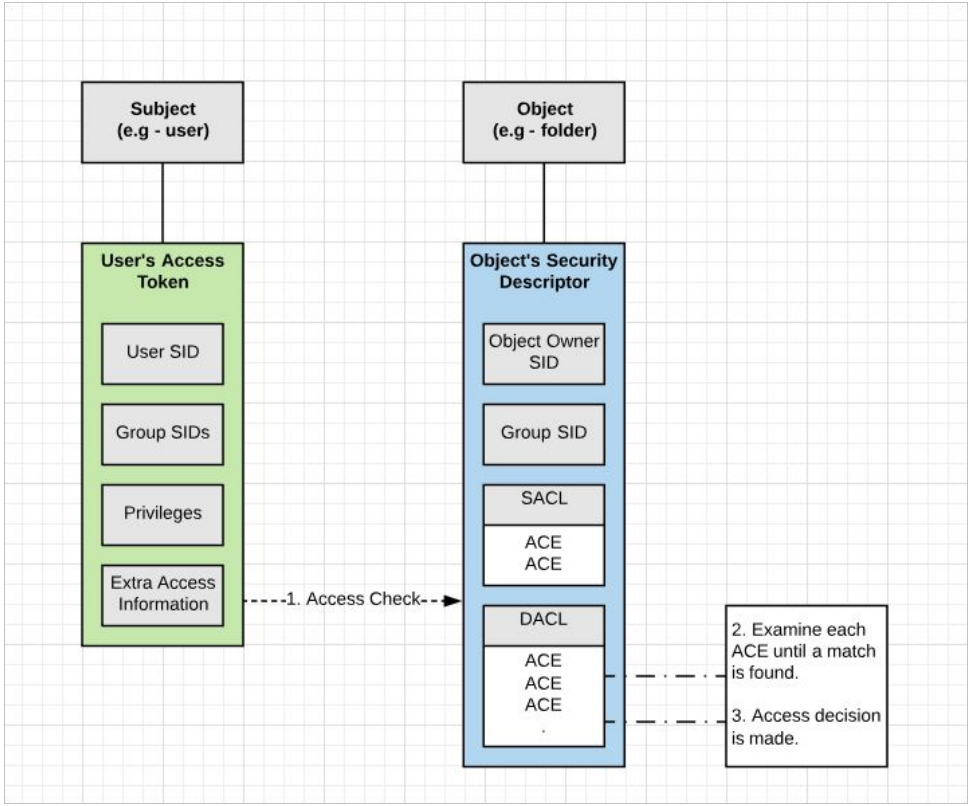
https://www.youtube.com/watch?v=3BQKpPNlTSo

2. Теория

Security Tokens and ACLs

https://docs.microsoft.com/en-us/windows/win32/secauthz/access-control-model

Windows Access checks process



Маркеры доступа (Access Tokens)

- Access Token - структура в ядре Windows (_TOKEN), описывающая **security context** процесса;
- Каждый раз когда процесс взаимодействует с объектом, имеющим **security descriptor,** система использует access token процесса, чтобы понять, разрешать или запрещать;
- Токены наследуются при порождении дочерних процессов;
- Помимо обычных (**primary**) есть еще **impersonation tokens***;
- Токены порождаются в процессе логона (входа в систему);
- Новые токены также можно получить в результате вызова DuplicateToken(Ex) и CreateRestrictedToken.

*Есть еще LowBox Tokens, которые используются в песочнице AppContainer

Access Token содержит:

- Пользователя и группы (в виде их SID*), в том числе primary group;
- Список привилегий**, доступных в описываемом security context;
- Дефолтный DACL и SID владельца для создаваемых securable objects;
- Идентификатор Logon Session, в рамках которой создан токен;
- Тип токена (primary/impersonation);
- Уровень имперсонации: anonymous / identification / impersonation / delegation;
- Restricting SIDs - список групп для ограничения прав доступа у токена.

*SID (Security Identifier) - строка специального вида, идентифицирующая субъект - группу, пользователя, УЗ компьютера.
Например: S-1-5-21-3543244142-3217374098-3050147399-1001 или S-1-5-18
** Про привилегии и их security-имапкт см https://github.com/gtworek/Priv2Admin

Demo: Access Tokens

- whoami /all
- Process Explorer
- TokenViewer

Windows ACLs

- Любой **securable object** имеет **security descriptor**;
- **Security descriptor** содержит: SID владельца и primary group, списки доступа (ACLs), регулирующие доступ к securable object;
- ACL бывает двух видов: **Discretionary** (DACL) регулирует доступ, **System** (SACL) регулирует журналирование событий доступа;
- ACL это набор **Access Control Entries** (ACE);
- ACE бывают разных типов, но у всех есть общие поля:
  - trustee: SID **субъекта,** чей доступ надо разрешить/запретить или журналировать;
  - access mask: какие именно права регулирует данные ACE;
  - флаги, регулирующие наследование ACE.

SDDL - текстовый вид security descriptor

https://habr.com/ru/company/pm/blog/442662/

SDDL - текстовый вид security descriptor*

- «A» — правило **разрешает** действия субъекту;
- «CC», «LC», «SW», «LO», «CR», «RC» — список access rights;
- «IU» — означает группу Interactive Logged-on Users.

NB: “access right” может означать разное в зависимости от типа объекта. Для сервисов WP — это «остановка сервиса», для файлов - «исполнение», для папок - «траверс» (доступ к файлам в папке по имени, при отсутствии возможности перечислить содержимое).

https://docs.microsoft.com/en-us/windows/win32/secauthz/ace-strings

Demo: ACLs and Access Check

- Системный диалог
- icacls.exe
- accesschk.exe
- SDDLViewer.exe
- Process Monitor

Integrity levels

- Механизм мандатного контроля доступа, проверяется до DACL;
- Субъекты и объекты доступа integrity levels:

Untrusted < Low < Medium < High < System < Installer

- Для доступа необходимо, чтобы integrity level субъекта был как минимум таким же, как у объекта;
- Для большинства объектов: No-Write-Up, для процессов еще и No-Read-Up;

Integrity levels - UAC

- Административный процесс с medium integrity может запросить повышение до high, для этого производится RPC-вызов к службе Application Information Service (applmgmt.dll), что приводит к оповещению UAC;
- В некоторых случаях оповещения не происходит (autoelevate executables);
- Несмотря на то, что MS считает, что “UAC is not a security boundary”, обход оповещения UAC бывает необходим:
  - Чтобы не спалиться пользователю;
  - Когда нет интерактивной сессии, чтобы отобразить оповещение.

CVE-2018-8440*

Task Scheduler SchRpcSetSecurity ALPC LPE

https://www.exploit-db.com/exploits/45280

Task Scheduler ALPC LPE: вводная информация (1/2)

- В Windows есть интерфейс удаленного вызова процедур (RPC);
- Его локальная разновидность использует в качестве транспорта ALPC;
- Task Scheduler - служба планировщика задач, запущена с правами системы;
- Эта служба реализует общедоступный ALPC-эндпоинт под названием SchRpcSetSecurity, который меняет DACL задачи:

```
long _SchRpcSetSecurity(

            [in][string] wchar_t* arg_1, //Task name

            [in][string] wchar_t* arg_2, //Security Descriptor string

            [in]long arg_3

);
```

https://habr.com/ru/post/421593/

Task Scheduler ALPC LPE: вводная информация (2/2)

- При обработке ALPC-вызовов службы часто используют **имперсонацию** токена вызывающего процесса, чтобы избежать EOP;
- Функция ищет файл задачи в C:\windows\system32\tasks и меняет DACL, **используя** имперсонацию корректно;
- Если там нет файла, происходит поиск в C:\windows\tasks и там уже DACL устанавливается **без имперсонации** (с правами системы);
- В C:\Windows\tasks может писать любой пользователь;
- Благодаря механизму хардлинков*, мы можем “навесить” любой DACL на любой файл в системе от имени SYSTEM.

*https://googleprojectzero.blogspot.com/2015/12/between-rock-and-hard-link.html

Task Scheduler ALPC LPE: особенности эксплуатации

Проблемы:

- Файлы в C:\Windows\System32 доступны на запись только для TrustedInstaller (недоступны даже для SYSTEM);
- Нельзя писать в файлы, которые сейчас используются (например в открытые DLL).

Способ автора (https://www.exploit-db.com/exploits/45280):

- Служба печати spoolsv.exe грузит DLL при использовании XPS принтера.

Demo: Task Scheduler ALPC LPE

4. LPE via misconfigs

aka the easy way

Основная идея

- Ищем что-то, к чему мы имеем доступ на запись, и что как-то влияет на код, выполняющийся от имени более привилегированного пользователя;
- Результат небезопасной настройки системы администраторами или небезопасно сконфигурированного ПО (стороннего или самой ОС);
- Ниже мы перечислим основные места, где возможны такие ошибки;
- Для поиска мисконфигов есть автоматические средства, такие как winPEAS.

4.1 DLL Hijacking

- Основная идея: имеем право записи куда-то, откуда целевая программа (например, служба или что-то, что запускает администратор) загрузит DLL;
- Такое возможно если наше контролируемое место находится выше чем настоящая библиотека в перечне мест, где программа ищет библиотеки:
  - директория исполняемого файла;
  - текущая директория;
  - системные DLL-каталоги
  - директории из PATH.

4.2 Мисконфиги, связанные с сервисами (1/2)

- Сервис - это securable object, а значит права на него могут быть настроены некорректно (и позволить внаглую заменить строку запуска сервиса на запуск my-backdoor.exe);
- Может быть подвержен DLL Hijacking или исполняемый файл сервиса может быть доступен на запись;
- Unquoted Path: Если в конфигурации сервиса путь к исполняемому файлу содержит пробелы и не заключен в кавычки, система сначала попробует альтернативные варианты при запуске.

4.2 Мисконфиги, связанные с сервисами (2/2)

binPath=C:\program files\sub dir\program name.exe:

- C:\program.exe
- C:\program files\sub.exe
- C:\program files\sub dir\program.exe
- C:\program files\sub dir\program name.exe

Если можем писать в любое из этих мест - PROFIT.

4.3 Планировщик задач или файлы автозапуска

- Sysinternals Autoruns - ищет все места, из которых может происходить автоматический запуск программ (включая и сервисы);
- Проверяем эти места на возможность модификации нашим пользователем;
- На практике регулярно встречаются всякие C:\Stuff\CheckServiceIsWorking.ps1 в задачах от системы.

4.4 AlwaysInstallElevated

- Это такой способ разрешить неадминам устанавливать программы;
- Все установщики запускаются с повышенными правами;
- LPE by design, нужно только собрать MSI, который делает то, что нам нужно (например, регистрирует службу);
- По умолчанию, разумеется, отключено.

Homework / Further Reading

CVE-2019-1405 + CVE-2019-1322

- CVE-2019-1405 - повышение привилегий до LOCAL SERVICE через использование COM-интерфейса (задача со звёздочкой);
- CVE-2019-1322 - повышение привилегий от LOCAL SERVICE до SYSTEM через небезопасный ACL сервиса UsoSvc (должно быть понятно).

Райтап:

https://web.archive.org/web/20200329100247/https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2019/november/cve-2019-1405-and-cve-2019-1322-elevation-to-system-via-the-upnp-device-host-service-and-the-update-orchestrator-service/

Эксплоит: https://github.com/apt69/COMahawk

Старые ISO Win10 можно взять тут (для экспериментов подойдет версия 1803):
https://www.heidoc.net/joomla/technology-science/microsoft/67-microsoft-windows-and-office-iso-download-tool

Источники

https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20Escalation.md

https://book.hacktricks.xyz/windows/checklist-windows-privilege-escalation

https://book.hacktricks.xyz/windows/windows-local-privilege-escalation

https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/winPEAS

https://github.com/bitsadmin/wesng

https://github.com/googleprojectzero/sandbox-attacksurface-analysis-tools/