Bölüm A: Teori ve Mimari (Research & Logic)

1. Linux Mimarisi: "Her Şey Bir Dosyadır" (Everything is a File)

Kavramsal Analiz:

- Linux'ta /dev/sda (Harddisk), /dev/tty (Terminal) veya /dev/null (Kara delik) gibi donanımların bile dosya sisteminde bir yolunun olması, bir sistem yöneticisine ne gibi bir avantaj sağlar? (İpucu: cat komutuyla bir metin dosyasını okumakla, bir harddiskin imajını almak arasındaki benzerliği düşünün).

Linux'ta donanımların da bir dosya gibi olması bize örneğin : bir diski dosya gibi okunup herhangi bir dosyaya kopyalamamızı sağlar ya da terminalin düzgün çalışıp çalışmadığını kontrol etmek için doğrudan ekrana yazı gönderebiliriz veya koca gereksiz dosya ve logları temizlemek için sistem kaynaklarını tüketmeden direkt kara delik dosyasına yönlendirmek gibi avantajları vardır. Kısaca , özel ve karmaşık sürücü komutlarına ihtiyaç duymadan , standart dosya araçlarını kullanabilmemizi sağlar.

Tehlikeli İzinler ve SUID Biti:

- 777 Tehlikesi: Bir dosyaya chmod 777 (rwxrwxrwx) izni vermek, sistem güvenliği açısından neden bir intihardır? "Others" (Diğerleri) grubunun yazma

yetkisine sahip olması, sıradan bir kullanıcıyı nasıl sisteme zarar verebilecek hale getirir?

777 kodu verildiğinde user, group , others kullanıcı tiplerine okuma ,yazma ve çalıştırma yetkisi vermiş oluruz . Yani bir dosya üzerinde tüm kullanıcı tiplerinin istediği gibi işlem yapma yetkisini vermiş oluruz. Script dosyasıysa hacker sistemi sil komutunu ekledikten sonra ve biz onu yönetici olarak çalıştıdığımız durumda sistem çökebilir. Veri tabanı dosyasıysa hacker verileri değiştirebilir ve kullanıcı şifrelerini okuyabilir. Eğer bir hacker ise bu dosya olduğu durumda zararlı yazılımı bu dosyaya koyması yeterlidir. Biz bu dosyayı silsek de hacker 777 kod yetkisi verdiğimiz için tekrar o dosyayı üretebilir. Bu yüzden siber güvenlikte her zaman kişilere ihtiyacı kadar yetki vermek esastır.

- Hackerların Altın Anahtarı (SUID): SUID (Set User ID) biti nedir? Normalde passwd komutunu çalıştıran bir kullanıcı, nasıl olur da sadece Root'un yazabildiği /etc/shadow dosyasına şifre yazabilir? Saldırganlar neden sistemde find / -perm -4000 komutuyla bu bitin ayarlandığı dosyaları arar?

Normalde bir programı çalıştırdığında, o program bizim yetkilerimizle çalışır. Ancak bir dosyaya SUID biti atanmışsa, o programı kim çalıştırırsa çalıştırsın, program dosyanın sahibinin yetkileriyle çalışır. /etc/shadow dosyasını sadece root okuyup yazabilir, /usr/bin/passwd komutunun sahibi root kullanıcısıdır ve üzerinde SUİD biti ayarlıdır. Bu yüzden passwd komutu da /etc/shadow dosyasına şifre yazabilir. Bir saldırgan sisteme sızdığında düşük yetkili olup root olmak ister bu yüzden sistemdeki tüm SUİD dosyalarını find / -perm -4000 -type f 2>/dev/null komutu ile aratır.

Bash Gücü vs. GUI:

- Bir Siber Güvenlik Analisti olarak önünüze 5 GB boyutunda bir web sunucu log dosyası (access.log) geldi. Bunu Excel veya Notepad ile açmaya çalışırsanız RAM şişer ve bilgisayar donar. Aynı dosyayı Linux terminalinde; grep, awk ve | (Pipe) operatörlerini kullanarak saniyeler içinde analiz etmenin (Örn: Sadece 404 hatası veren IP'leri ayıklamanın) mühendislik açısından önemi nedir?

Excel gibi grafik arayüzlü programlar " Batch processing " mantığıyla çalışır. Yani 5 GB'lik dosyanın tamamını belleğe yüklemeden bize ilk satırı göstermezler. Linux araçları (awk , grep) veriyi satır satır okuyup ekrana basar. Bu sayede çok az RAM kullanarak işlemleri yapabiliriz. Pipe operatörü zincirleme bir bağlama yaptığı için (bir komutun çıktısını diğerinin girdisi yaparak) büyük bir filtreleme fabrikası yapmış oluruz. Örneğin Linux terminalinde SQL injection denemesi ararken bu işlemi bize saniyeler içerisinde sonuçlandırabilir. Başka bir örnek verecek olursam terminal loglarını (binlerce satırdan oluşan bir .sh dosaysı) düzenli olarak otomatik tarayan bir sistem kurabiliriz. Bu işlemler çok hızlı gerçekleşir. Bir mühendis için verimlilik zaman demektir. Burda ise zamandan tasarruf etmiş oluruz.

2. Windows Internals: "Çarklar Nasıl Dönüyor?"

User Mode vs. Kernel Mode (Ring 0 - Ring 3):

- Modern işlemciler ve Windows, kararlılığı sağlamak için neden iki farklı modda çalışır?

- Senaryo: Sizin açtığınız chrome.exe çökerse sadece tarayıcı kapanır, ancak ekran kartı sürücüsü (Driver) hata verirse neden "Mavi Ekran" (BSOD) alırız ve tüm sistem çöker? "Ring 0" ve "Ring 3" kavramlarını kullanarak açıklayın.

Modern işlemciler (x86 mimarisi), donanımı ve belleği korumak için donanımsal bir güvenlik duvarı kullanır. Bu yapı iç içe geçmiş halkalara (Ring) benzer. Toplamda 4 ring (ring 0 "Kernel- İşletim Sistemi" , ring 1 " Device Drivers ", ring 2 " OS Services - İşletim Sistemi Servisleri " , ring 3 " Application - Kullanıcı Uygulamaları ") bulunmaktadır. Windows ve Linux iki tanesini (ring 0 ve ring 3) aktif olarak kullanır ve standart olarak kabul edip buna göre kod yazarlar. Çünkü halkalar arası geçişler performans kaybına sebeb oluyordu. Ring 0 = Tanrı modudur. Sistemdeki her şeye %100 erişiminiz vardır (hem donanım hem de yazılım açısından) Ring 3 = Kullanıcılara ayrılmış donanım ve programlara erişimin kısıtlı olduğu bir moddur. Birnevi karantina bölgesidir. Chrome.exe sadece kendini kapatır. Çünkü İşletim sistemin her uygulama için Ring 3 te belli bir bellek verir. Bu bellek aşılırsa o process i öldürür(kill). Ekran kartı sürücüsü mavi ekran verir . Çünkü bu bir aslında savunma mekanizmasıdır. Neden ? Ekran kartı sürücüsü (Driver), oyunlarda yüksek FPS verebilmek ve donanımla doğrudan yüksek hızda haberleşebilmek için işletim sisteminin tam kalbinde, yani Ring 0 seviyesinde çalışmak zorundadır. Ring 0 da bellek

taşması demek felaket demektir. Eğer ring 0 da bellek taşması yaşasaydı , Windows çekirdeği bunun diğer uygulamarı bozmasını önlemek için zamanı durduruyor. Bu da mavi ekran anlamaına geliyor.

Ayrıca İlgilisine Bilgi !

Ring 3 teki bir program nasıl ring 0 'a mesaj yollar ve bunu hackerlar nasıl ve ne için kullanır ? Ring 3 ' ü bir müşteri Ring 0 ' ı bir kasa olarak ve Syscall' ü ise bir veznadar olarak destek tanımlayalım. syscall komutu çalıştığı zaman ,zaman durması yaşanır yani işmlemci Ring 3' ü durdurur. Tabi Çekirdek yetkinin olup olmadığını kontrol edip Ring 0 moduna geçirir. Yani kasanın sahibi olmadan kasaya erişemezsin. Antivirüs programları syscall'un yanına bir güvenlik (NtWriteFile) koyar ve her syscall çağrıldığında araya girip yazılan şeyin güvenli olup olmadığına bakar. Hackerler ise bu durumu şöyle çözer : Antivirüsün izlemediği kendi güvenlik veznedarını yazarlar. Buna " Direct Syscalls " tekniği denir. Bu sayede Assembly diline syscall komutunu yollayıp güvenlik önlemlerinin altından tünel kazıp geçerler.

Registry (Kayıt Defteri) Anatomisi:

- Windows'un beyni olan Registry'de; HKLM (HKEY_LOCAL_MACHINE) ile HKCU (HKEY_CURRENT_USER) arasındaki kritik fark nedir?
- Zararlı Yazılım Analizi: Bir Trojan bilgisayara bulaştığında, bilgisayar her yeniden başlatıldığında tekrar çalışabilmek (Persistence/Kalıcılık) için kendini genellikle hangi Registry yoluna (Path) yazar? (Örn: .../CurrentVersion/Run).

HKLM ve HKCU arasındaki en büyük farklar : Etki alanı ve Yetki (Privilege) 'dir. Bunu şöyle düşünelim : HKLM apartmanın ana şalteri olsun . Etki alanı tüm kullanıcılardır ve yetki olarak normal bir apartman daire sahibi buraya dokunamaz sadece apartmanın yöneticisi izin verirse dokunulabilir. HKCU ise bir dairenin ışığıdır. Sadece apartmanda yaşayan insanları etkiler. Bizim masaüstü arkaplanımız buradadır. Yetki olarak ise burada yönetici olmaya gerkek yoktur . Yani herhangi bir kullanıcı burayı istediği gibi kullanabilir. Kalıcılık için zararlı bir trojen yazılımı HKLM kısmına yazmalıdır. Çünkü buraya yazılan herhangi bir şey her kullanıcı için geçerlidir. Bilgisayar çalıştığı andan itibaren aktif olur. HKCU path ine yazılan bir trojen yazılımı sadece o yazıldığı kullanıcı için geçerlidir. Yani A kullanıcısında virüs sızdıysa bilgisayara sadece A kullanıcısı bilgisayarı açtığı zaman çalışır. HKLM path ine sızan bir virüs, hangi antivirüs uygulamasını kullanırsak kullanalım sistemin tamamını ele geçirir.

NTFS İzinleri ve ACL (Erişim Kontrol Listesi):

- Linux'taki basit "Okur/Yazar/Çalıştır" mantığı Windows'ta çok daha detaylıdır.
- Bir klasöre sağ tıklayıp "Security" sekmesine geldiğimizde gördüğümüz ACL (Access Control List) yapısı nedir? Bir dosyaya "Bunu Ali sadece okusun, Veli hem okusun hem silsin, ama SYSTEM hesabı dokunamasın" diyebilmek, kurumsal güvenliği nasıl etkiler?

ACL bir dosyanı kapısında bekleyen ve elinde çok detaylı listeler olan güvenlik görevlisidir. Linuxta 3 muhattap vardır : Sahibi, grubu , diğerleri . Windowsta ise ACL sayesinde sınırsız muhattap ve yetki ekleyebiliriz. Bu şirket yönetim sisteminde güvenlik

açısından büyük önem sağlıyor. Mesela bir dosyayı değiştirme yetkisini bir stajer yapamazken müdür yapacak şekilde ayarlayabiliriz. Veya bir dosyayı okuma ve silmeye kapatıp sadece eklemeye açacak bir şekilde yetkilendirebiliriz bu bize güvenlik açısından büyük ayrıcalıklar sunar ve bir hacker'ın sisteme sızıp önemli bilgiler alınmasını zorlar. Hatta diyelim ki bir dosya içeriğini sahip bir insanın sildiği bir içeriği bir üst yöneticiye log düşüp haber vermesini sağlayacak şekilde ayarlayabiliriz. Linux'ta 777 yetkisi verme durumunda mesela bir zararlı yazılım yüklendiği durumunda i kendi kendini dıştan silsek bile geri getirme yetkisi vermiş oluruz. (ki bu büyük güvenlik zafiyetlerine yol açabilir.) Ayrıca eğer ki ACL de verilen bir DENY(reddetme) verilen tüm ALLOW (izin verme) lardan daha üstündür. Bu da sadece yöneticinin sistemi zorla sıfırlamasıyla değiştirilebilir. Onun haricinde mühendislik açısından ACL, kaosu mükemmel bir düzene çeviren ince ayar tablosudur.

Bölüm B: Saha Eğitimi (TryHackMe - Lab)

Görev 1: Linux Cephesi - Terminalle Tanışma

Linux, siber güvenlik dünyasının ana dilidir. Sunucular, saldırı araçları ve bulut sistemleri Linux üzerinde konuşur. Fareyi bırakıp klavyeye hükmetme vakti !

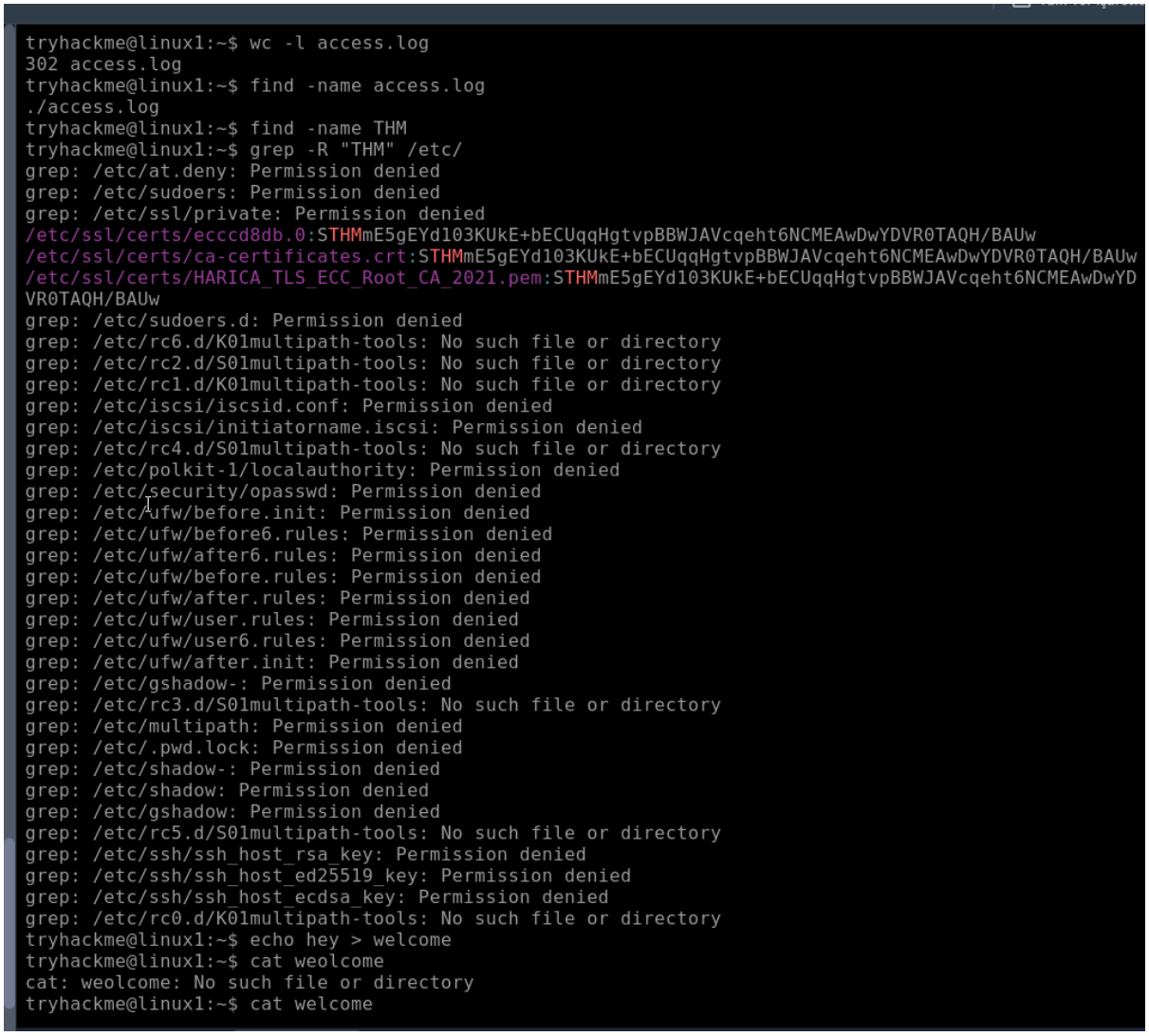
Oda: TryHackMe | Linux Fundamentals Part 1-2-3

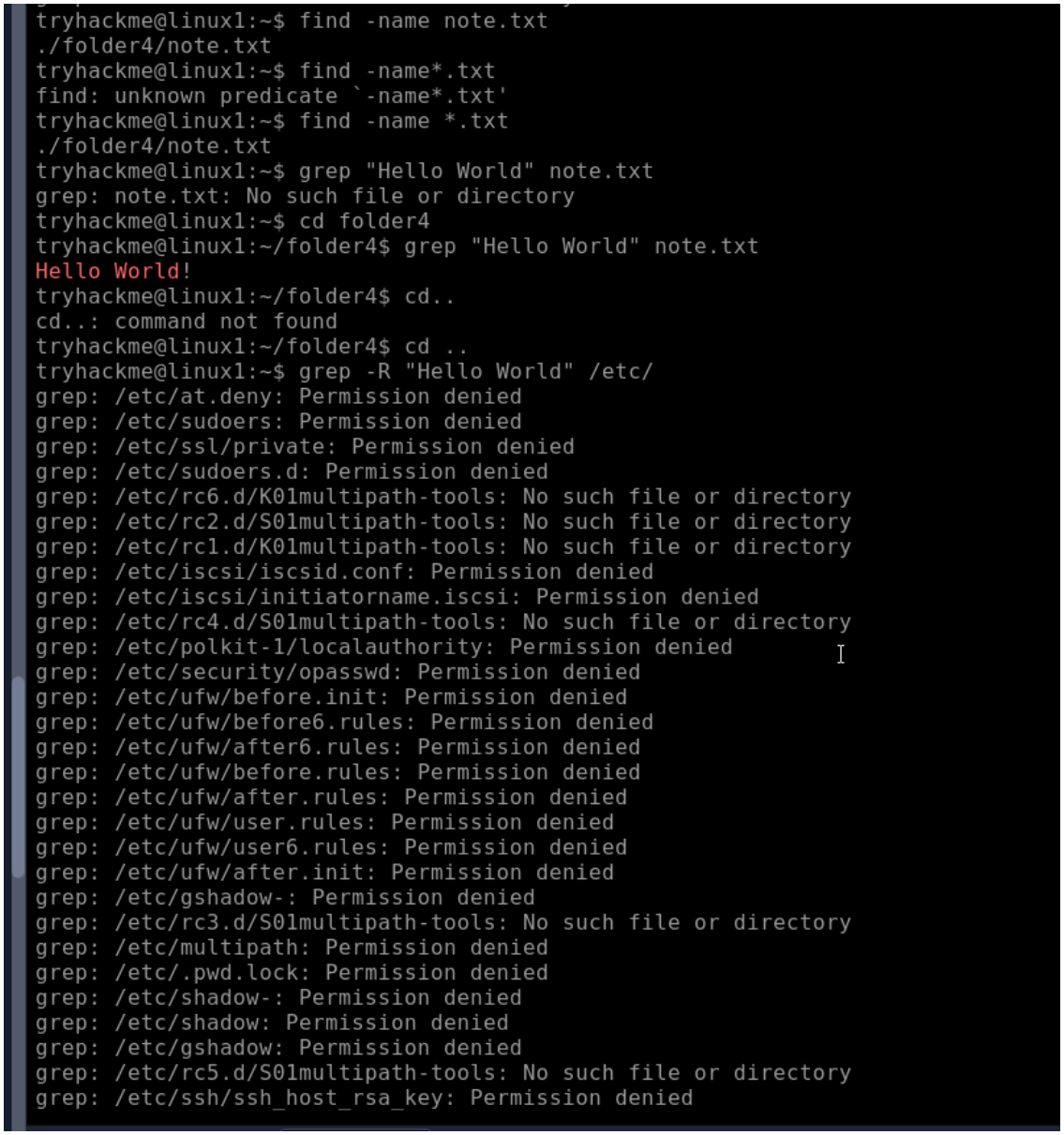
Amaç: Dosya sistemi hiyerarşisini, temel komutları ve izin yapılarını öğrenmek.

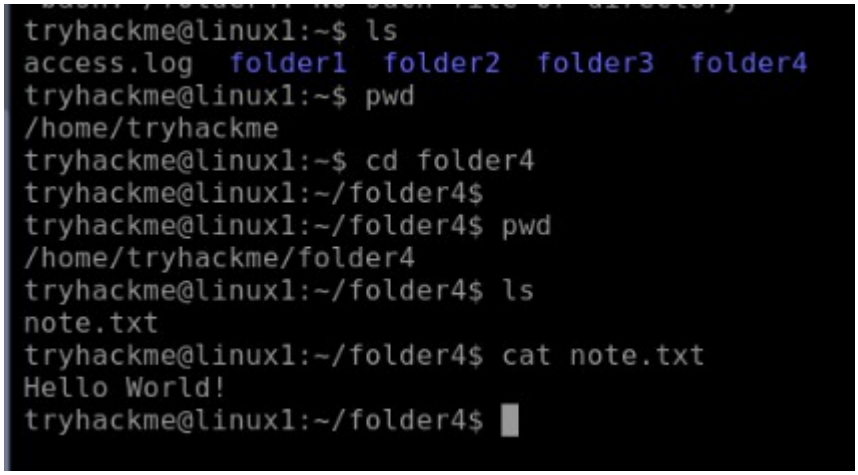
Raporlama İsteği: Modülleri tamamlarken öğrendiğiniz, bir analistin hayatını kurtaracak 3 kritik komutu seçin ve uygulamasını ekran görüntüsüyle gösterin.

Örnek Komutlar: grep (Arama), chmod (İzin değiştirme), pipe | (Komutları bağlama), find (Dosya bulma).

Beklenti: Sadece komutu yazmayın; komutun çıktısını (Output) da gösteren bir ekran görüntüsü ekleyin.


tryhackme@linux1:~$ wc -l access.log
302 access.log
tryhackme@linux1:~$ find -name access.log
./access.log
tryhackme@linux1:~$ find -name THM
tryhackme@linux1:~$ grep -R "THM" /etc/
grep: /etc/at.deny: Permission denied
grep: /etc/sudoers: Permission denied
grep: /etc/ssl/private: Permission denied
/etc/ssl/certs/ecccd8db.0:STHMmE5gEYd103KUkE+bECUqqHgtvpBBWJAVcqeht6NCMEAwDwYDVR0TAQH/BAUw
/etc/ssl/certs/ca-certificates.crt:STHMmE5gEYd103KUkE+bECUqqHgtvpBBWJAVcqeht6NCMEAwDwYDVR0TAQH/BAUw
/etc/ssl/certs/HARICA_TLS_ECC_Root_CA_2021.pem:STHMmE5gEYd103KUkE+bECUqqHgtvpBBWJAVcqeht6NCMEAwDwYD
VR0TAQH/BAUw
grep: /etc/sudoers.d: Permission denied
grep: /etc/rc6.d/K01multipath-tools: No such file or directory
grep: /etc/rc2.d/S01multipath-tools: No such file or directory
grep: /etc/rc1.d/K01multipath-tools: No such file or directory
grep: /etc/iscsi/iscsid.conf: Permission denied
grep: /etc/iscsi/initiatorname.iscsi: Permission denied
grep: /etc/rc4.d/S01multipath-tools: No such file or directory
grep: /etc/polkit-1/localauthority: Permission denied
grep: /etc/security/opasswd: Permission denied
grep: /etc/ufw/before.init: Permission denied
grep: /etc/ufw/before6.rules: Permission denied
grep: /etc/ufw/after6.rules: Permission denied
grep: /etc/ufw/before.rules: Permission denied
grep: /etc/ufw/after.rules: Permission denied
grep: /etc/ufw/user.rules: Permission denied
grep: /etc/ufw/user6.rules: Permission denied
grep: /etc/ufw/after.init: Permission denied
grep: /etc/gshadow-: Permission denied
grep: /etc/rc3.d/S01multipath-tools: No such file or directory
grep: /etc/multipath: Permission denied
grep: /etc/.pwd.lock: Permission denied
grep: /etc/shadow-: Permission denied
grep: /etc/shadow: Permission denied
grep: /etc/gshadow: Permission denied
grep: /etc/rc5.d/S01multipath-tools: No such file or directory
grep: /etc/ssh/ssh_host_rsa_key: Permission denied
grep: /etc/ssh/ssh_host_ed25519_key: Permission denied
grep: /etc/ssh/ssh_host_ecdsa_key: Permission denied
grep: /etc/rc0.d/K01multipath-tools: No such file or directory
tryhackme@linux1:~$ echo hey > welcome
tryhackme@linux1:~$ cat weolcome
cat: weolcome: No such file or directory
tryhackme@linux1:~$ cat welcome





Görev 2: Windows Anatomisi - "Normal"i Tanımak

Siber savunmanın altın kuralı şudur: "Normali bilmeyen, anomaliyi (virüsü) bulamaz." Zararlı yazılımlar genellikle kendilerini Windows'un kendi sistem dosyaları gibi (Örn: svchost.exe) göstermeye çalışır. Bu labda gerçek ile sahteyi ayırt etmeyi öğreneceksiniz.

!

Oda: TryHackMe | Core Windows Processes

Kritik Süreç Analizi: Odayı tamamlarken aşağıdaki kritik süreçlerin ne işe yaradığını birer cümleyle (Türkçe ve kendi yorumunuzla) açıklayın. Bu süreçler ilerde malware analizi yaparken karşınıza en çok çıkacak olanlardır:

- System (PID 4): Neden her zaman PID 4'tür? Kernel (Çekirdek) ile ilişkisi nedir?

Bu süreç aslında bir .exe dosyası değil, işletim sisteminin kalbi olan Kernel'ın (Çekirdek) ta kendisidir; donanım ile yazılım arasındaki tüm trafiği yönettiği için sistemin ilk ve en yetkili süreci olarak her zaman PID 4 numarasını alır.

- smss.exe (Session Manager): Windows açılırken hangi kritik işi yapar? (Örn: Oturum başlatma).

Windows açılırken ilk kullanıcı oturumunu başlatan "ilk düğme" görevini görür; ortam değişkenlerini ayarlar ve diğer kritik sistem süreçlerini (csrss.exe gibi) hayata döndürür.

- csrss.exe (Client/Server Runtime): Neden bu süreci öldürürseniz "Mavi Ekran" (BSOD) alırsınız?

Pencerelerin çizilmesi ve konsol işlemleri gibi temel grafiksel/sistem fonksiyonlarını yürüten bu süreci öldürmek,

Windows'un nefesini kesmek demektir; sistem bu hayati organı kaybettiği için koruma amaçlı anında Mavi Ekran (BSOD) verir.

- lsass.exe (Local Security Authority): Hackerlar neden en çok bu süreci sever? (İpucu: Parolalar ve Hash'ler nerede tutulur?).

Bilgisayardaki tüm giriş (login) işlemlerinden, şifre doğrulamalarından ve yetkilerden sorumlu olduğu için, hackerlar kullanıcı şifrelerini veya Hash bilgilerini (kimlik verilerini) bellekten çalmak amacıyla en çok bu süreci hedef alır.

- svchost.exe (Service Host): Neden Görev Yöneticisi'nde bundan onlarca var?

Windows'un kendi içinde çalışan birçok farklı servisini (Windows Update, Defender vb.) bir araya toplayıp çalıştıran bir "taşıyıcı" olduğu için, her servis grubu ayrı bir kılıf içinde çalışır; bu yüzden Görev Yöneticisi'nde onlarca kopyasını görürüz.

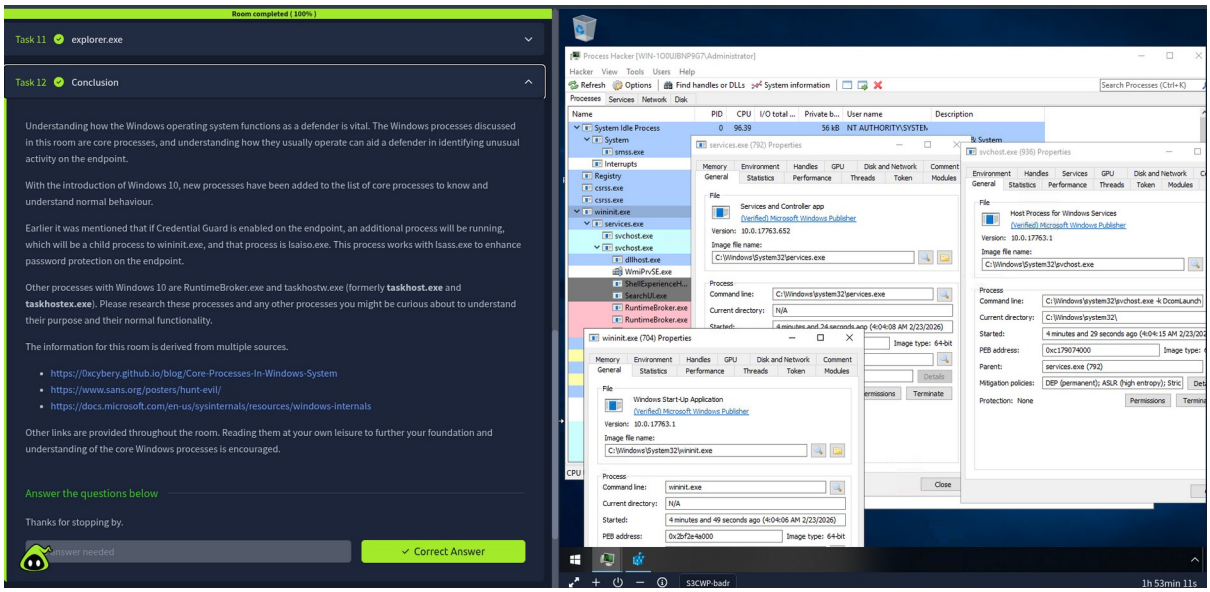
İlgilisine bilgi !

- lsaiso.exe ne işe yarar ?

lsass.exe , ring 3 modunda çalıştığı için yönetici yetkisi alan bir zararlı yazılım belleğe sızıp şifreleri okuyabiliyordu. Microsoft bunu engellemek için Credential Guard özelliğini geliştirdi. lsaiso.exe sanallaştırma tabanlı bir güvenlik katmanıdır. Şifreler bu katmanda tutulur ve çekirdek bile bu katmana doğrudan dokunamaz sadece gelip şu şifre doğru mu deyip doğru ve yanlış bilgisini alıp gider. Yani şifreyi göremez. Birnevi çelik bir kasa var ve en yetkili kişi bile gitse bilginin içeriğini vermiyor. Sadece işlemin (örneğin şifrenin) doğru veya yanlış olduğunu söylüyor.

Kanıt: Odayı bitirdiğinize dair "Room Completed" veya tebrik ekranının görüntüsü.

⚠️ Mühendislik Notu: Bu Windows süreçlerinin "Parent-Child" (Ebeveyn-Çocuk) ilişkisi çok önemlidir. Örneğin; services.exe normalde wininit.exe altından çıkar. Eğer explorer.exe altından çıkan bir services.exe görürseniz, tebrikler! Bir virüs buldunuz.



Cephe 1: OverTheWire - Bandit (Linux)

Burası, terminal korkunuzu yeneceğiniz yer.

Bandit, Linux komut satırını öğretmek için tasarlanmış efsanevi bir savaş oyunudur (Wargame). Burada fare çalışmaz, sadece komutlar konuşur.

Hedef: OverTheWire: Bandit Wargame

Bağlantı Yöntemi: SSH (Secure Shell) kullanarak oyuna bağlanacaksınız.

Komut: ssh bandit0@bandit.labs.overthewire.org -p 2220

Şifre: bandit0

Görev: Level 0'dan başlayıp Level 10'a kadar (Level 10 dahil) gelmeniz gerekmektedir.

Raporlama Formatı:

Sadece şifreyi (Flag) yazıp geçmeyin! Her leveli nasıl geçtiğinizi teknik olarak açıklayın.

Örnek Raporlama:

Level X Çözümü: Dosya adı tire (-) ile başladığı için normal cat -file komutu hata verdi. Bu yüzden cat ./-file komutunu kullanarak dosyayı okudum. Kanıt: [Ekran Görüntüsü]

Kanıt: Her seviyede bir sonraki levelin şifresini bulduğunuz terminal ekranının görüntüsü raporda olmalıdır.

💡 İpucu: Bandit seviyelerinde ilerlerken man, ls -la, file, find, du, grep komutları en iyi dostunuz olacak. Takıldığınızda Google'ı kullanmaktan çekinmeyin, "araştırmak" mühendisliğin %90'ıdır.

Level 0 Çözümü: Sisteme SSH protokolü ile bandit0 kullanıcısı olarak giriş yaptım. İlk adım olarak mevcut dizindeki dosyaları listelemek için ls komutunu kullandım ve dizinde readme adında bir dosya olduğunu tespit ettim. Bu dosyanın bir metin dosyası olduğunu bildiğim için, içeriğini terminale yazdırmak ve şifreyi okumak amacıyla cat readme komutunu kullandım. Komutun çıktısı bana bir sonraki seviyenin parolasını verdi.

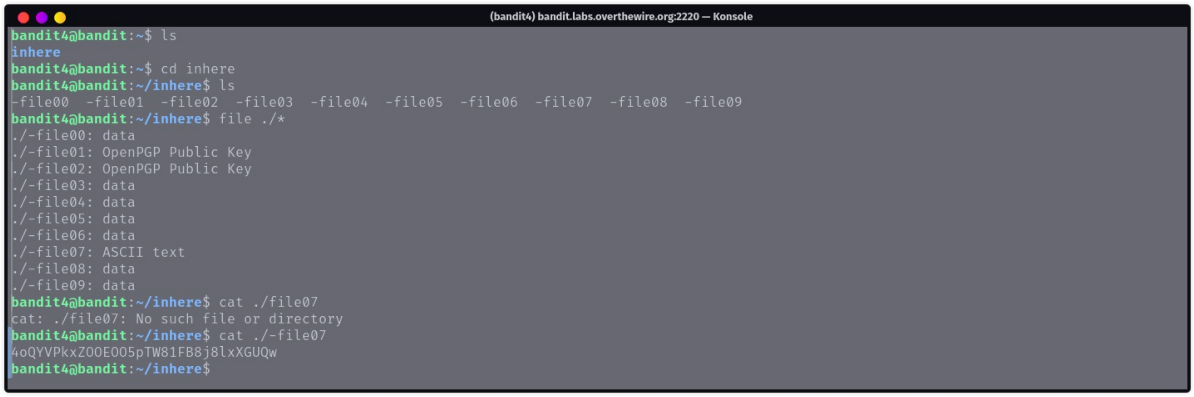
Level 1 Çözümü : ls komutu ile listelediğimde dosya adının sadece tire (-) işaretinden oluştuğunu gördüm. Eğer cat - komutunu denersem terminal hata verir. Çünkü Linux'ta tire işareti genellikle "standart girdi" (stdin) anlamına gelir. Dosyanın tam yolunu belirtmek için ./ (mevcut dizin) önekini kullandım. Böylece Linux'a bunun bir komut parametresi değil, mevcut dizindeki - isimli bir dosya olduğunu anlattım.

Level 2 Çözümü : ls komutu ile listelediğimde dosya adının (spaces in this filename) kelimeler arasında boşluklar (space) içerdiğini gördüm. Eğer doğrudan cat spaces in this filename komutunu denersem terminal hata verir. Çünkü Linux terminalinde boşluk karakteri, komut parametrelerini birbirinden ayırmak için kullanılır ve sistem bunu tek bir dosya yerine 4 farklı dosya sanar. Dosya isminin bütünlüğünü korumak ve boşlukların ismin bir parçası olduğunu sisteme anlatmak için dosya adını çift tırnak (" ") içine aldım. Ayrıca dosya adının karışıklık yaratmaması için ./ (mevcut dizin) ön ekini de ekleyerek tam yolu belirttim.

Level 3 Çözümü : inhere dizinine girdiğimde standart ls komutunu kullandım ancak dizin boş göründü. Linux sistemlerinde ismi nokta (.) ile başlayan dosyaların gizli olduğunu bildiğim için, gizli dosyalar dahil tüm içeriği listelemek amacıyla ls komutuna -a parametresini ekledim (ls -a). Bu komut sonucunda ...Hiding-From-You adında gizli bir dosya tespit ettim. Dosya ismini doğru şekilde terminale girerek (cat ./...Hiding-From-You) içeriğini okudum ve şifreye ulaştım.

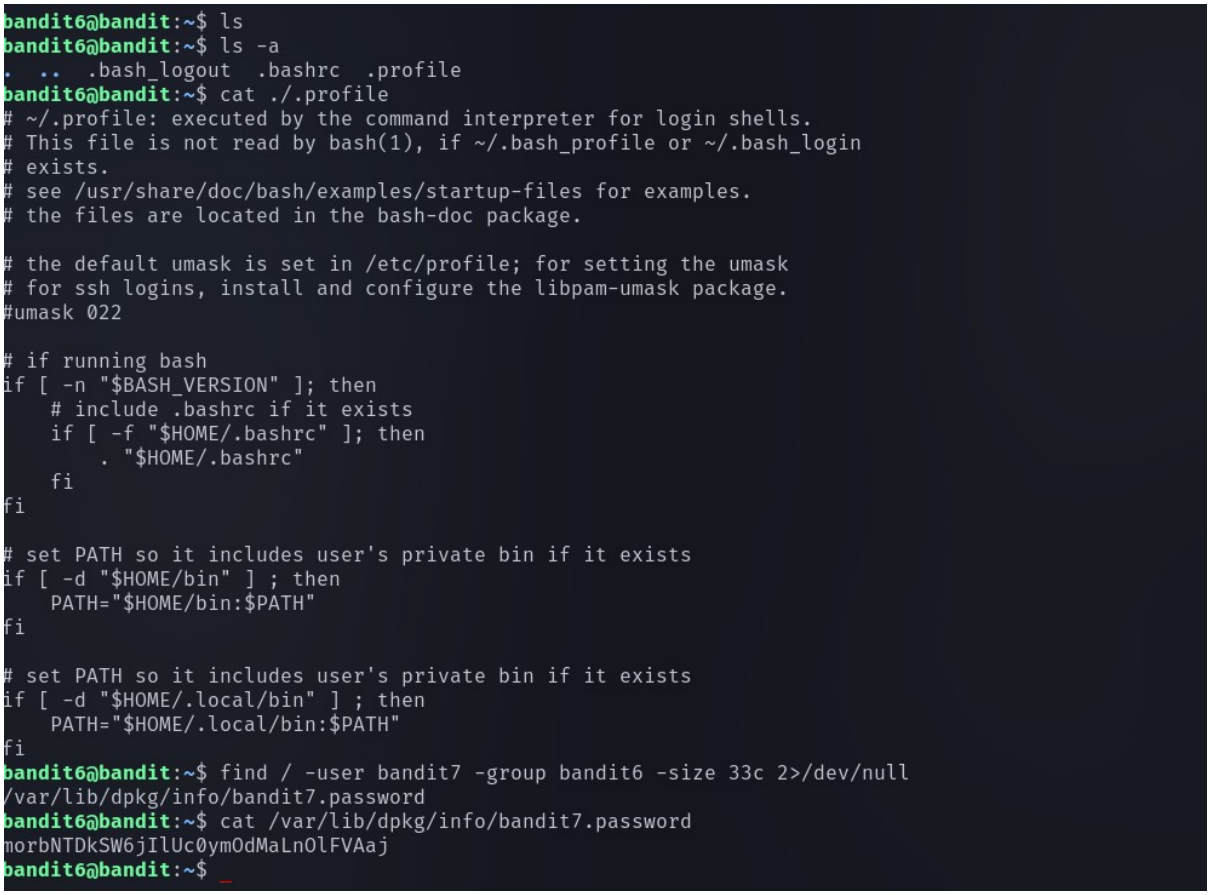
Level 4 Çözümü : inhere dizinine girdiğimde -file00'dan -file09'a kadar isimlendirilmiş çok sayıda dosya ile karşılaştım. Bu dosyaların çoğu okunamaz formatta (binary/data) olduğu için, içlerinden sadece insan tarafından okunabilir olan tek dosyayı bulmam gerekiyordu. Tüm dosyaların içeriğini tek tek okumak yerine, dosya türlerini analiz eden file komutunu kullandım. Eğer sadece file * yazsaydım, dosya isimleri tire (-) ile başladığı için terminal bunları birer dosya adı değil, komut parametresi (flag)

sanabilirdi. ./* yazarak sisteme "Mevcut dizindeki (./) her şeyi (*) al" dedim. Bu sayede dosya yolları ./-file00 şekline dönüştü ve komutun onları güvenli bir şekilde dosya olarak algılamasını sağladım. Bu analiz sonucunda sadece ./-file07 dosyasının "ASCII text" olduğunu tespit ettim ve cat komutuyla şifreyi okudum.



Level 5 Çözümü : inhere dizinine girdiğimde maybehere00'dan başlayarak çok sayıda alt klasör ve bunların içinde yüzlerce dosya ile karşılaştım. Bu kadar dosya arasından doğru olanı manuel olarak bulmak imkansız olduğu için, dosyanın bilinen özellikleri (1033 byte, çalıştırılamaz ve insan okunabilir) üzerinden filtreleme yapmam gerekiyordu. Tüm dosyaları tek tek kontrol etmek yerine, dosya özelliklerine göre detaylı arama yapan find komutunu kullandım. Komuta -type f ekleyerek arama sonucunu sadece dosyalarla sınırladım. -size 1033c parametresi ile dosya boyutunun tam 1033 byte olmasını şart koştum (c eki byte olduğunu belirtir). Ayrıca ! -executable diyerek çalıştırılabilir dosyaları sonuçlardan eledim. Bu analiz sonucunda kriterlere uyan tek dosyanın ./maybehere07/.file2 olduğunu tespit ettim ve cat komutuyla şifreyi okudum. (Dosyanın 1033 byte olduğunu ve çalıştırılamaz olduğunu soruda ipucu olarak vermişti.)

Level 6 Çözümü : Hedef dosyanın sunucunun herhangi bir yerinde (/ dizini altında) olabileceği belirtilmişti. Dosyanın sadece 3 özelliği biliniyordu: bandit7 kullanıcısına ait olması, bandit6 grubuna ait olması ve boyutunun tam 33 byte olması. Tüm dosya sistemini (/) taramak gerektiği için, normal bir arama işlemi sırasında yetkimizin yetmediği binlerce klasörden "Permission Denied" hata mesajları dönecek ve gerçek sonucu görmemizi engelleyecekti. Dosyayı bulmak için find komutunu kök dizinden (/) başlatarak kullandım. -user bandit7, -group bandit6 ve -size 33c parametrelerini kullanarak arama kriterlerini belirledim. Linux'ta hata mesajları "Standart Hata" (stderr, file descriptor 2) kanalından gelir. Bu kirliliği önlemek için 2>/dev/null komutunu ekledim. Bu sayede tüm hata mesajlarını /dev/null (kara delik) aygıtına yönlendirerek terminalde sadece temiz ve başarılı sonucun görünmesini sağladım.



Level 7 Çözümü : bandit7 kullanıcısı ile giriş yaptığımda dizinde data.txt adında bir dosya olduğunu gördüm. Bu dosya, binlerce satır veri içerdiği için manuel olarak okunması ve içerisindeki parolanın bulunması mümkün değildi. Görev açıklamasında parolanın "millionth" kelimesinin yanında olduğu belirtilmişti. Dosyanın tamamını ekrana yazdırmak yerine, sadece ihtiyacım olan anahtar kelimeyi içeren satırı yakalamak için Linux'un en güçlü metin arama araçlarından biri olan grep komutunu kullandım. Bu komut sayesinde dosya içerisindeki binlerce satır filtrelenerek sadece "millionth" kelimesinin geçtiği satır ekrana yazdırıldı. Böylece karmaşık veriler arasından bir sonraki seviyenin parolasına hızlı ve temiz bir şekilde ulaştım.

Level 8 Çözümü :

- Terminalde bağlantı adresini doğru yazıp ve şifreyi doğru yazmama rağmen ulaşamadım . Bu yüzden level 8 –9-10 çözümü yoktur!!!

Cephe 2: Sysinternals Analizi (Windows)

Kendi bilgisayarınızda veya Sanal Makinede (VM) bir "Avcı" olun.

Windows Görev Yöneticisi size yalan söyleyebilir ama Process Explorer söylemez. Bu görevde, Microsoft'un efsanevi mühendisi Mark Russinovich tarafından yazılan Sysinternals araçlarını kullanacağız.

Adım 1: Cephaneyi Hazırla

Microsoft'un resmi Sysinternals Suite paketini indirin.

ZIP dosyasını klasöre çıkarın.

procexp.exe (Process Explorer) uygulamasını Yönetici Olarak (Run as Administrator) çalıştırın.

Adım 2: Analiz ve Raporlama (3 Görev) Aşağıdaki analizleri kendi bilgisayarınızda yapın ve sonuçları raporlayın:

Parent-Child (Ebeveyn-Çocuk) Analizi:

Bir programın kim tarafından başlatıldığı, onun güvenli olup olmadığını anlamanın en iyi yoludur.

Görev: Herhangi bir programa (Örn: Chrome, Word veya Discord) sağ tıklayıp "Properties" deyin. "Image" sekmesindeki "Parent" kısmını bulun.

Rapor: Seçtiğiniz programın Parent süreci nedir? (Örn: explorer.exe -> chrome.exe). Ekran görüntüsünü ekleyin.

Svchost Avı (Service Host):

Listede pembe renkli onlarca svchost.exe göreceksiniz. Bunlar Windows servislerini (Windows Update, Bluetooth, Ses vb.) taşıyan süreçlerdir.

Görev: Bir tane svchost.exe sürecinin üzerine farenizi getirin (Hover). Açılan sarı kutucukta (Tooltip) hangi servislerin çalıştığını görün.

Rapor: İçinde en az 3-4 servisin çalıştığı bir svchost'un ekran görüntüsünü alın.

İmza Kontrolü (Verify Signatures):

Zararlı yazılımlar genellikle Microsoft imzasına sahip değildir.

Görev: Menüden Options > Verify Image Signatures seçeneğini işaretleyin.

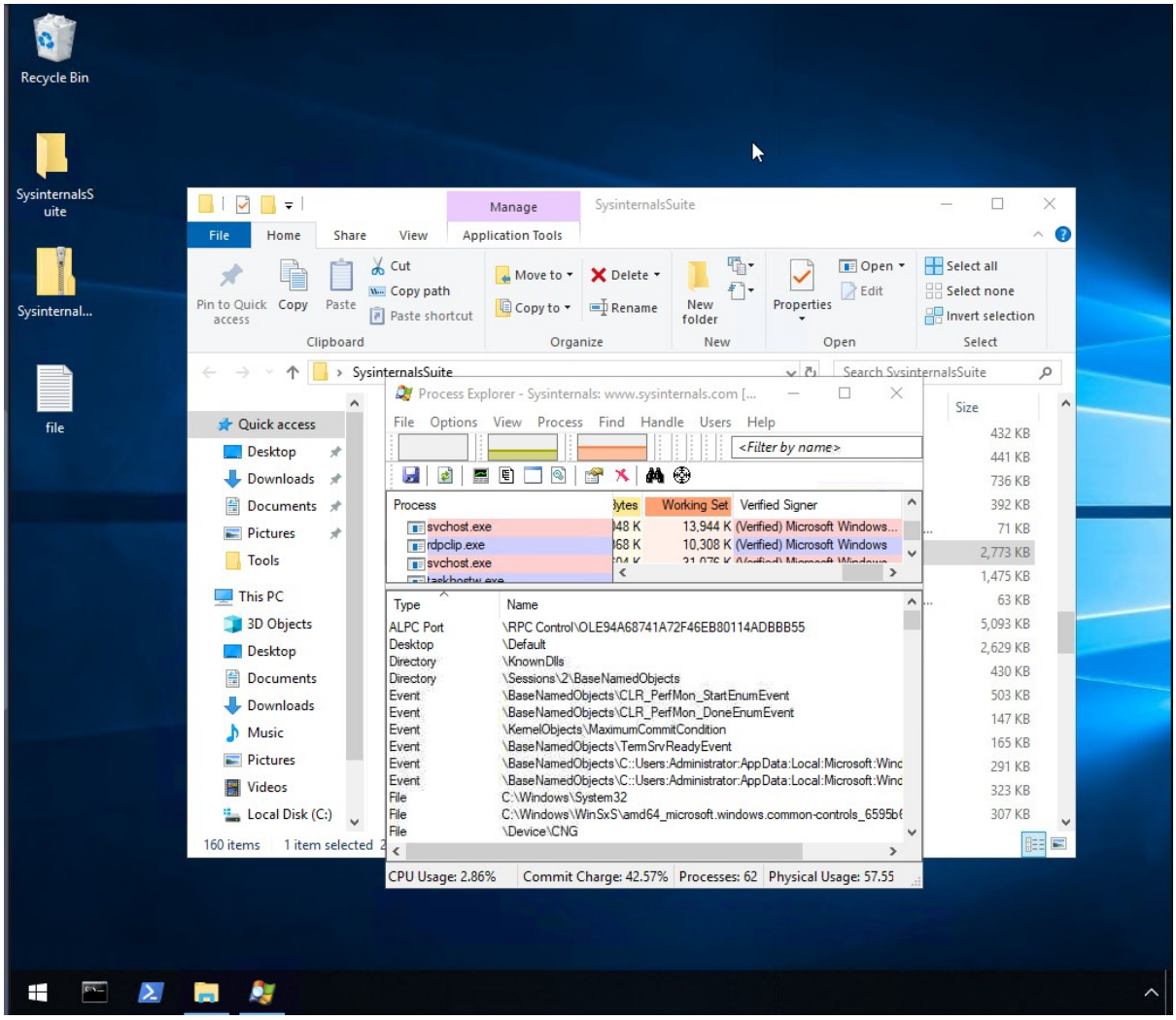
Analiz: Listede "Company Name" kısmında (Verified) Microsoft Windows yazmayan, mor veya kırmızı renkle işaretlenmiş bir süreç var mı?

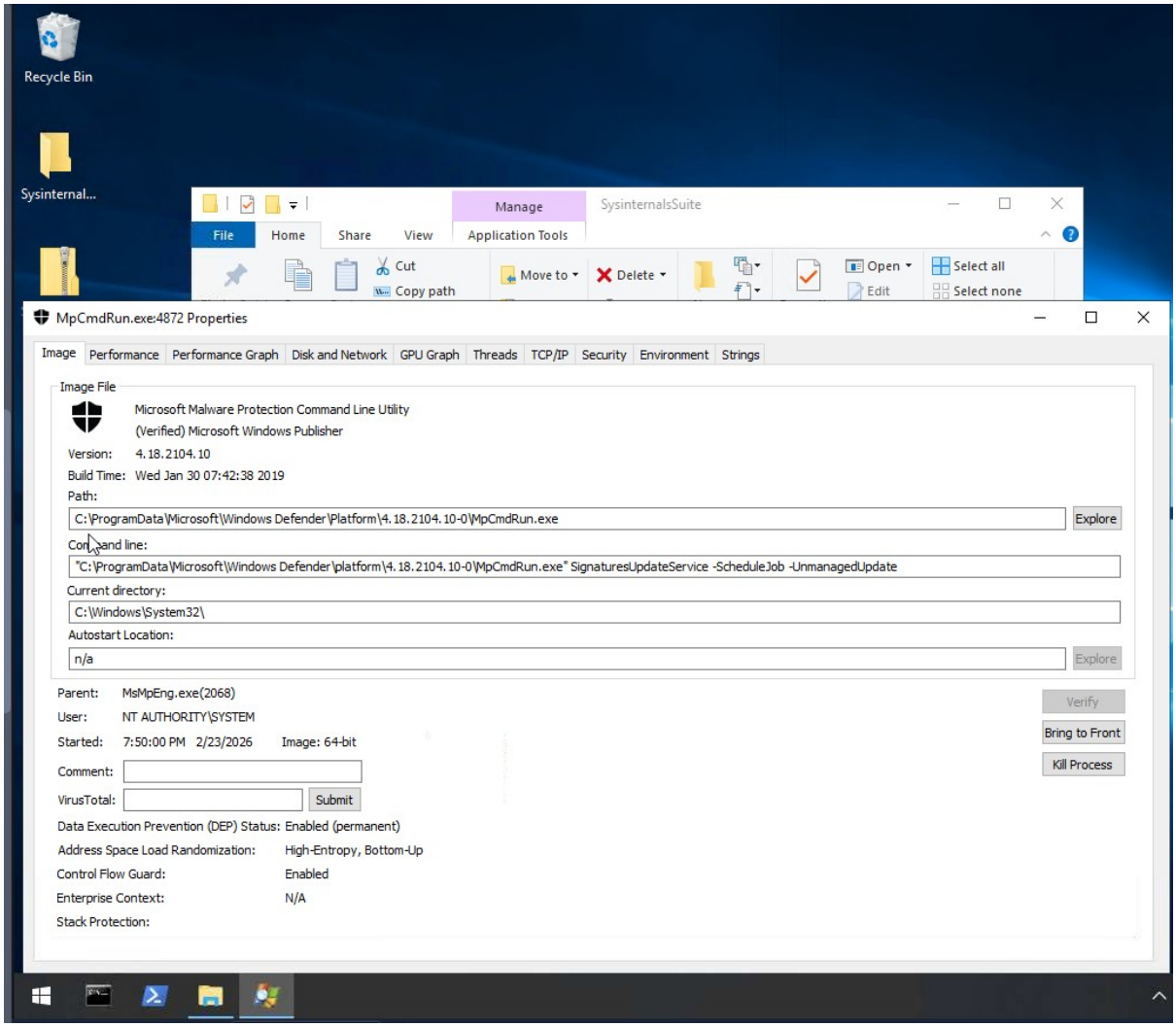
Varsa: Nedir bu program? Güvenli mi? (Google'layın).

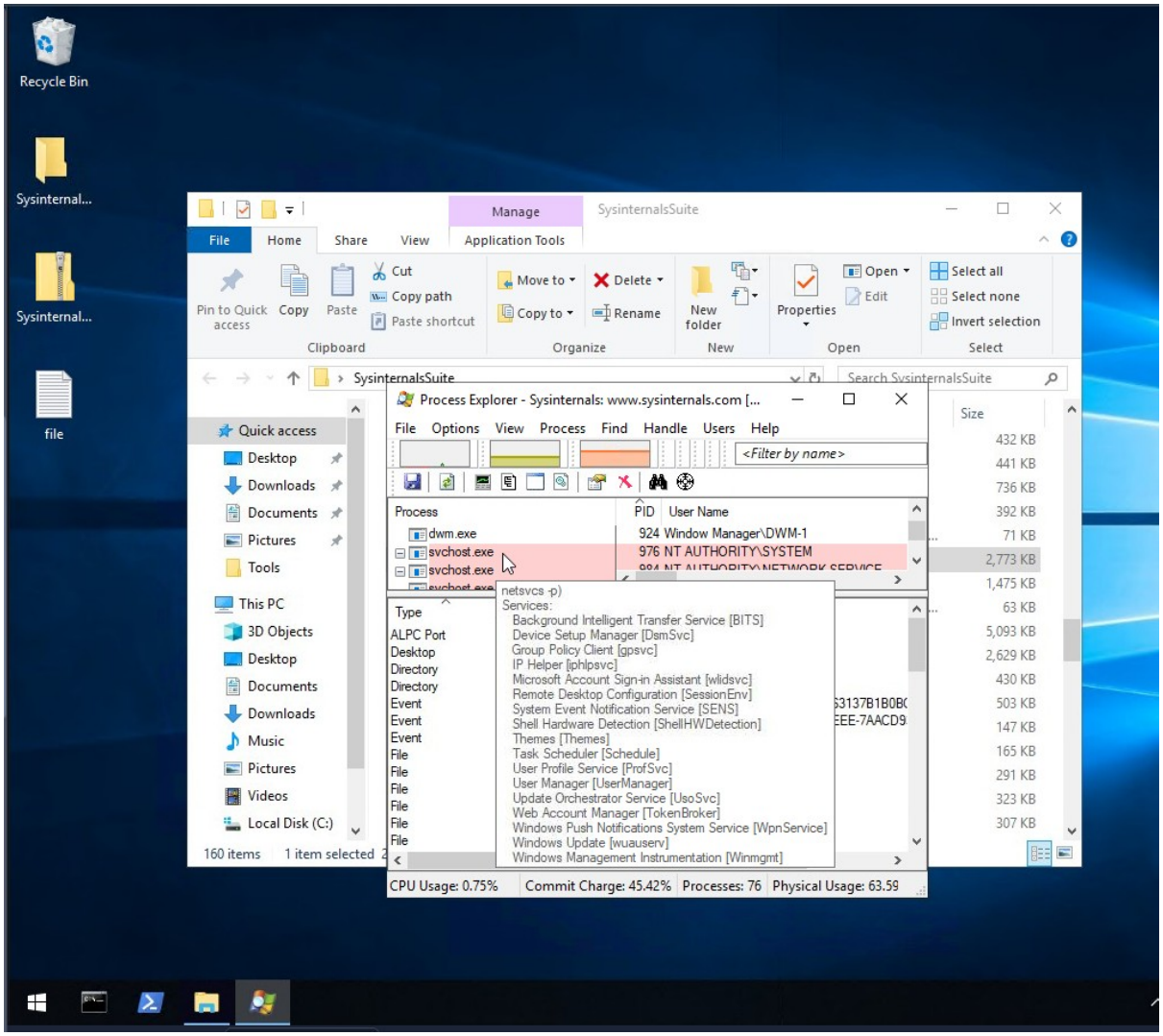
Yoksa: Bu özelliğin bir Malware Analisti için neden hayati olduğunu 1 cümleyle açıklayın.

Bu özellik, bir malware analisti için hayati önem taşır; çünkü saldırganlar zararlı dosyalarına sistem dosyası süsü verse bile, dijital imza kontrolü sayesinde dosyanın orijinal kaynağından gelip gelmediği anında tespit edilebilir. Process Explorer'daki renk kodları, dijital imza doğrulamasıyla birleştiğinde bir erken uyarı sistemi görevi görür. Örneğin, mor renkli (paketlenmiş) bir sürecin

imzasız olması, dosyanın analizden kaçmak için tasarlandığını gösteren en güçlü 'malware' belirtilerinden biridir.




Recycle Bin
Sysinternal...
Manage
SysinternalsSuite
File
Home
Share
View
Application Tools
Cut
Copy path
Move to
Delete
Open
Edit
Select all
Select none
MpCmdRun.exe:4872 Properties
Image
Performance
Performance Graph
Disk and Network
GPU Graph
Threads
TCP/IP
Security
Environment
Strings
Image File
Microsoft Malware Protection Command Line Utility
(Verified) Microsoft Windows Publisher
Version: 4.18.2104.10
Build Time: Wed Jan 30 07:42:38 2019
Path:
C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2104.10-0\MpCmdRun.exe
Explore
Command line:
"C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2104.10-0\MpCmdRun.exe" SignaturesUpdateService -ScheduleJob -UnmanagedUpdate
Current directory:
C:\Windows\System32\
Autostart Location:
n/a
Explore
Parent: MsMpEng.exe(2068)
User: NT AUTHORITY\SYSTEM
Started: 7:50:00 PM 2/23/2026 Image: 64-bit
Comment:
VirusTotal:
Submit
Verify
Bring to Front
Kill Process
Data Execution Prevention (DEP) Status: Enabled (permanent)
Address Space Load Randomization: High-Entropy, Bottom-Up
Control Flow Guard: Enabled
Enterprise Context: N/A
Stack Protection:



BÖLÜM D: Mühendislik Vizyonu (Reflection)

- 1. Yetki Yükseltme (PrivEsc) Mantığı: Senaryo: Linux sistem yöneticisi, işlerini kolaylaştırmak için vim (metin editörü) programına SUID biti atamış (chmod u+s /usr/bin/vim).
- Soru: Sadece metin düzenlemeye yarayan masum bir editör, bu "küçük" izin hatası yüzünden nasıl olur da sıradan bir kullanıcıyı ROOT (Süper Yönetici) yapabilir? /etc/shadow veya /etc/sudoers dosyaları üzerinden mantığı açıklayın.

Bir dosyaya SUID biti atandığında (chmod u+s), o programı kim çalıştırırsa çalıştırsın, program dosya sahibinin (bu senaryoda ROOT) yetkileriyle çalışmaya başlar. Masum görünen vim, root yetkisini eline aldığında artık sistemdeki her dosyayı okuyabilir ve her dosyaya yazabilir hale gelir. Normalde /etc/shadow dosyasını normal bir kullanıcı okuyamaz ama kullanıcı vim /etc/shadow komutunu çalıştırır. Kendi kullanıcısının şifre hash'ini silebilir veya root kullanıcısının hash'ini bilinen bir şifreyle değiştirebilir. /etc/sudoers Bu dosya, hangi kullanıcının sudo yetkisine sahip olduğunu belirler. Saldırgan vim /etc/sudoers komutunu açar. Bu komut ile birlikte sudo komutuyla sistemde tam yetki alır kendisi için ve sudo komutuyla istediği tüm şifreli komutları şifresiz bir şekilde çalıştırabilir.

2. Zararlı Yazılım Kamuflajı (Process Injection): Senaryo: Bir Malware geliştiricisisiniz ve virüsünüzün (Trojan) antivirüsler tarafından fark edilmemesini ve sistemde en yüksek yetkide çalışmasını istiyorsunuz.

Soru: Kodunuzu neden rastgele bir oyun.exe yerine, sistemin kendi parçası olan svchost.exe veya explorer.exe sürecinin içine enjekte edersiniz?

İpucu 1: Görev yöneticisine bakan bir göz ne görür? (Stealth/Gizlilik).

İpucu 2: svchost.exe genellikle hangi kullanıcı yetkisiyle (SYSTEM vs User) çalışır? (Privilege).

oyun.exe yapısı hem yetki olarak hem de güvenirlilik olarak daha zaafiyetlidir. svchost.exe veya explorer.exe içine enjekte edilen virüs kolay kolay kullanıcı tarafından bulunamaz çünkü bunlar hassas süreçlerdir. Kullanıcı tarfından yanlış bir şey yapılması

durumunda Windows çökebilir. Onun haricinde oyun.exe 'yi kullanıcı kurcalarken şüphelenebilir. Ama svchost.exe' den sistemde 50 tane çalışır ve bunu kullanıcı gördüğünde karışmaz . Çünkü yanlış bir hareketi büyük sıkıntılara mal olabilir. Onun haricinde svchost.exe genellikle SYSTEM yetkisiyle çalışır.