**katzekey: Kernel Seviyesinde LSASS'a Giden Gizli Anahtar**

Kırmızı Takım (Red Team) operasyonlarının ve sızma testlerinin en kritik hedeflerinden biri, ağ içinde yatay olarak hareket edebilmek için kimlik bilgilerini ele geçirmektir. Windows sistemlerde bu kimlik bilgilerinin Kalesi olarak bilinen yer **LSASS (Local Security Authority Subsystem Service)** işlemidir. Yıllardır bu kaleyi fethetmek için kullanılan en meşhur araç mimikatz olmuştur. Ancak güvenlik çözümleri (özellikle EDR'lar) geliştikçe, mimikatz'in LSASS'a yaptığı her hareket izlenir ve engellenir hale geldi.

İşte bu kedi-fare oyununda, savunmayı atlatmak için geliştirilen yeni nesil araçlardan biri de **katzekey**'dir. Bu yazıda, katzekey'in ne olduğunu, neden bu kadar önemli olduğunu, nasıl kurulup kullanılacağını ve onu bir siber güvenlik uzmanı gibi etkili bir şekilde nasıl kullanabileceğinizi keşfedeceğiz.

**1. katzekey Aracı Nedir?**

**katzekey**, Windows sistemlerde LSASS işleminin belleğinden kimlik bilgilerini (NTLM hash'leri, Kerberos biletleri, bazen de düz metin parolalar) dökmek (dump) için tasarlanmış, açık kaynak kodlu bir araçtır.

Onu mimikatz gibi geleneksel araçlardan ayıran en temel özelliği, çalışma yöntemidir. katzekey, LSASS işlemine erişmek için standart Windows API'lerini (örneğin OpenProcess) kullanmak yerine, **kernel (çekirdek) seviyesinde çalışan bir sürücü (driver)** kullanır. Bu sürücü sayesinde, kullanıcı modu (user-mode) güvenlik kancalarını ve EDR'ların izleme mekanizmalarını tamamen atlayarak doğrudan LSASS belleğine erişir. Bu, ona muazzam bir gizlilik ve etkinlik avantajı sağlar.

Kısacası, katzekey "kapıdan girmek" yerine, "duvarların içinden geçerek" hedefine ulaşır.

**2. Neden Kullanılır?**

katzekey'in geliştirilme ve kullanılma nedenleri, modern siber savunma mekanizmalarının getirdiği zorlukları aşmaya odaklıdır:

* **EDR ve Antivirüs Atlatma (EDR/AV Evasion):** Bu, en önemli kullanım nedenidir. EDR çözümleri, LSASS'a yapılan şüpheli erişim denemelerini tespit etmek için kullanıcı modundaki API çağrılarını sıkı bir şekilde izler. katzekey, kernel sürücüsü aracılığıyla bu izlemeden kaçar ve operasyonunu gizlice gerçekleştirir.
* **Doğrudan ve Ham Bellek Erişimi:** Kernel sürücüsü, LSASS işleminin belleğine kısıtlamasız ve ham bir erişim sağlar. Bu sayede, güvenlik yazılımlarının koyduğu sanal engeller ortadan kalkar.
* **Etkili Kimlik Bilgisi Toplama:** Başarılı bir şekilde çalıştığında, ağ içinde ilerlemek (yatay hareket) için kullanılabilecek değerli NTLM hash'lerini ve Kerberos biletlerini elde eder. Bu hash'ler, "Pass-the-Hash" gibi tekniklerle diğer sistemlerde oturum açmak için kullanılabilir.
* **Savunma Mekanizmalarını Test Etmek:** Mavi Takımlar (Blue Team) için katzekey, kullandıkları EDR çözümünün kernel seviyesindeki tehditleri tespit edip edemediğini ölçmek için mükemmel bir test aracıdır. Eğer katzekey başarılı olursa, savunmada ciddi bir kör nokta olduğu anlaşılır.

**3. Nasıl Kurulum Yapılır?**

katzekey, bir kullanıcı modu uygulaması (katzekey.exe) ve bir kernel modu sürücüsünden (.sys dosyası) oluşur. Genellikle kaynak koddan derlenmesi gerekir.

**Gereksinimler:**

* Windows İşletim Sistemi
* Visual Studio (C++ Desktop Development paketi ile birlikte)
* Windows SDK ve WDK (Windows Driver Kit)

**Kurulum Adımları:**

1. **Depoyu Klonlama:**  
   Aracın kaynak kodlarını GitHub'dan klonlayın.

git clone https://github.com/bleu-s-oc/katzekey.git

1. **Projeyi Derleme:**
   * Klonladığınız dizine gidin.
   * Visual Studio'da katzekey.sln çözüm dosyasını açın.
   * Derleme yapılandırmasını Release ve x64 olarak ayarlayın.
   * Menüden Build > Build Solution seçeneğine tıklayarak projeyi derleyin.
   * Derleme başarılı olursa, x64/Release klasörü altında katzekey.exe ve driver.sys (veya benzer isimli bir sürücü dosyası) dosyaları oluşacaktır.
2. **Sürücü İmzalama Notu:**  
   Windows, güvenlik nedeniyle varsayılan olarak yalnızca dijital olarak imzalanmış sürücülerin yüklenmesine izin verir. katzekey projesinde kullanılan zam64.sys sürücüsü, bilinen ve sızdırılmış eski bir sertifikaya sahip olduğu için bazı test ortamlarında çalışabilir. Ancak modern ve sıkılaştırılmış sistemlerde, sürücünün geçerli bir sertifika ile imzalanması veya sistemin "Test Modu"na alınması gerekebilir.

**4. Kullanım Parametreleri Nedir?**

katzekey, basit ve anlaşılır komut satırı parametreleri ile kullanılır.

Generated powershell

# Temel yardım menüsünü görmek için:

.\katzekey.exe

# Başlıca parametreler:

/dump # LSASS belleğini döker ve kimlik bilgilerini ekrana basar.

/full # LSASS belleğini ayrıştırmak yerine tam bir minidump dosyası (.dmp) olarak kaydeder.

/load # Sadece sürücüyü sisteme yükler.

/unload # Yüklenmiş olan sürücüyü sistemden kaldırır.

/pid <ID> # LSASS işleminin PID'sini manuel olarak belirtmek için kullanılır (genellikle otomatik bulur).

/driver <yol> # Varsayılan sürücü yerine özel bir sürücü dosyasının yolunu belirtmek için.

**5. Bu Aracı Etkili Kullanmanın 5 Tane Örneği**

Aşağıda, katzekey'in farklı operasyonel senaryolarda nasıl kullanılabileceğini gösteren 5 etkili örnek bulunmaktadır.

**Örnek 1: Standart ve Hızlı LSASS Dökümü**

* **Amaç:** En hızlı şekilde hedef sistemdeki NTLM hash'lerini elde etmek.
* **Açıklama:** Bu, aracın en temel ve en sık kullanılan senaryosudur. Araç, sürücüyü otomatik olarak yükler, LSASS'ı döker, kimlik bilgilerini ekrana basar ve ardından sürücüyü kaldırır.
* **Komut:**

.\katzekey.exe /dump

* **Sonuç:** Ekranda mimikatz çıktısına benzer bir formatta kullanıcı adları, NTLM hash'leri ve diğer kimlik bilgileri listelenir. Bu hash'ler "Pass-the-Hash" saldırıları için hemen kullanılabilir.

**Örnek 2: Çevrimdışı Analiz için Tam Bellek Dökümü (OPSEC Odaklı)**

* **Amaç:** Hedef sistemde en az iz bırakmak ve analizi saldırganın kendi makinesinde yapmak.
* **Açıklama:** LSASS belleğini hedef sistemde ayrıştırmak (parsing), şüpheli bir aktivite olarak görülebilir. /full parametresi, belleği ham bir .dmp dosyası olarak kaydeder. Bu dosya daha sonra saldırganın makinesine çekilip, mimikatz ile güvenli bir ortamda analiz edilebilir. Bu, operasyonel güvenlik (OPSEC) açısından çok daha iyidir.
* **Adımlar:**
  1. **Dökümü Al (Hedef Makinede):**

.\katzekey.exe /full

Bu komut lsass\_xxxxx.dmp adında bir dosya oluşturur.

* 1. **Analiz Et (Saldırgan Makinesinde):**  
     Oluşturulan .dmp dosyasını kendi makinenize aktarın ve mimikatz ile açın.

mimikatz # sekurlsa::minidump lsass\_xxxxx.dmp

mimikatz # sekurlsa::logonpasswords

**Örnek 3: Sürücüyü Manuel Olarak Yönetmek**

* **Amaç:** Aracın adımlarını (sürücü yükleme ve döküm alma) ayırarak daha fazla kontrol sağlamak.
* **Açıklama:** Bazen bir EDR'nin sadece sürücü yükleme anını yakalayıp yakalamadığını test etmek isteyebilirsiniz. Bu senaryoda, işlemleri manuel olarak yaparsınız.
* **Adımlar:**
  1. **Sürücüyü Yükle:**

.\katzekey.exe /load

* 1. **Dökümü Al:**

Generated powershell

.\katzekey.exe /dump

* 1. **İşin bitince Sürücüyü Kaldır:**

.\katzekey.exe /unload

**Örnek 4: Tespit Edilmiş Sürücü Yerine Farklı Bir Sürücü Kullanmak**

* **Amaç:** Güvenlik ürünlerinin imza tabanlı tespitini atlatmak.
* **Açıklama:** katzekey ile gelen varsayılan sürücünün (örneğin zam64.sys) hash değeri, zamanla güvenlik ürünleri tarafından "bilinen kötü" olarak işaretlenebilir. Gelişmiş bir operatör, bilinen başka bir zafiyetli ve imzalı sürücüyü (örneğin, farklı bir oyunun anti-hile sürücüsü) kullanarak bu tespiti atlatabilir.
* **Komut:**

.\katzekey.exe /driver C:\yol\diger\_surucu.sys /dump

* **Sonuç:** katzekey, varsayılan sürücü yerine sizin belirttiğiniz ve potansiyel olarak EDR tarafından tanınmayan bir sürücüyü kullanarak LSASS'a erişir.

**Örnek 5: Bir Komut Dosyası (Script) İçinde Otomatize Etmek**

* **Amaç:** Bir sızma sonrası otomasyon zincirinin parçası olarak kimlik bilgisi toplamak.
* **Açıklama:** Gerçek bir operasyonda, araçlar genellikle manuel çalıştırılmaz. PowerShell veya C2 framework'lerinin (örneğin Cobalt Strike) otomasyon yetenekleri ile kullanılır. katzekey'in çıktısı bir dosyaya yönlendirilip, bu dosya otomatik olarak parse edilerek elde edilen hash'ler ağdaki diğer makinelere karşı anında denenebilir.
* **Örnek PowerShell Komut Dosyası Parçacığı:**

# Katzekey'i çalıştır ve çıktıyı bir dosyaya kaydet

.\katzekey.exe /dump > credentials.txt

# Dosyadaki NTLM hash'lerini çıkar (örnek bir regex ile)

$hashes = Get-Content .\credentials.txt | Select-String -Pattern "NTLM\s+:\s([a-f0-9]{32})"

# Her hash'i kullanarak başka bir makinede komut çalıştırmayı dene

foreach ($hash in $hashes) {

# Pass-the-Hash aracını çağır...

}

**Sonuç**

katzekey, siber güvenlikte devam eden silahlanma yarışının mükemmel bir örneğidir. Savunmacılar mimikatz gibi araçları tespit etmek için sofistike yöntemler geliştirdikçe, saldırganlar da kernel seviyesi gibi daha derin ve karanlık sulara inerek bu savunmaları aşmanın yollarını bulmaktadır. Bu araç, bir EDR'nin yeteneklerinin sınırlarını test etmek ve kimlik bilgisi güvenliğinin ne kadar kritik olduğunu anlamak için paha biçilmezdir. Ancak unutulmamalıdır ki, **katzekey ve benzeri araçlar son derece güçlüdür ve yalnızca siber güvenlik uzmanları tarafından, yasal ve etik kurallar çerçevesinde, izinli sızma testleri ve savunma tatbikatları için kullanılmalıdır.**