Detecting Mimikatz with Sysmon

Sysmon

Sysmon, Microsoft tarafından geliştirilen ve cihazın aktivitelerinin kaydedilmesini sağlayan bir araçtır. Süreçler ve ağ bağlantıları gibi faaliyetler için ayrıntılı bilgi sağlar ve anormal durumların tespit edilmesini sağlar. Kurulum ve konfigürasyon için ayrıntılı bilgi Microsoft'un web sitesinde bulunabilir.

https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Mimikatz

Windows sistemlerinde hafızadan şifre alınmasını sağlayan bir araçtır.

https://github.com/gentilkiwi/mimikatz

Sysmon kullanarak sistemde mimikatz tespit etmenin 3 farklı yolundan bahsedeceğiz:

-Monitoring files named Mimikatz Mimikatz isimli dosyaların izlenmesi
-Monitoring hash
-"lsass.exe"

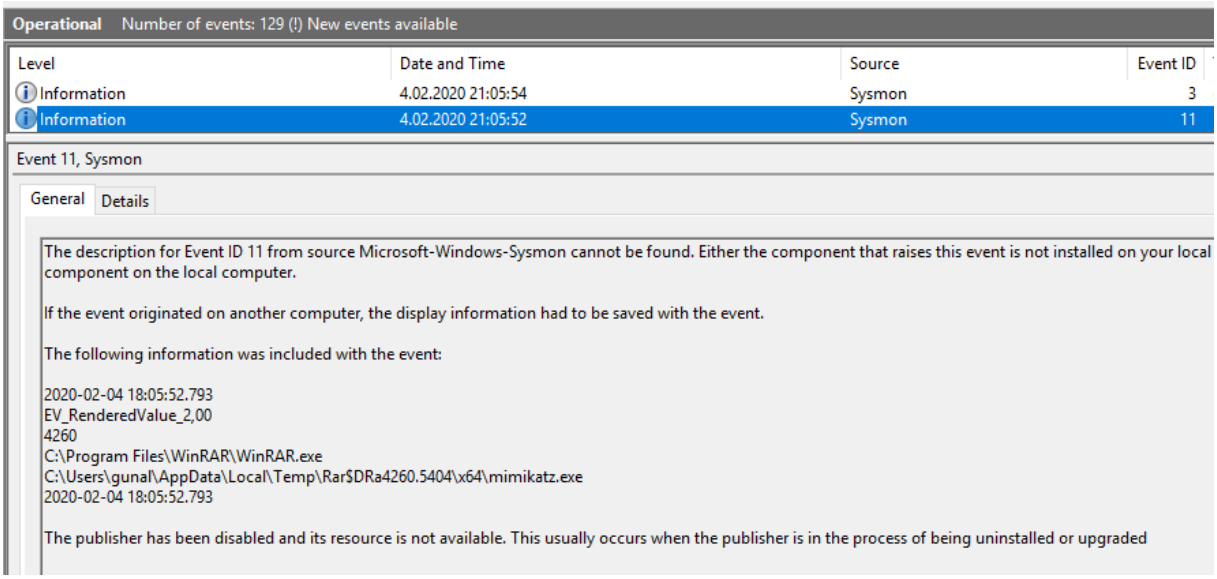
Mimikatz adlı dosyaları izleme

Sistemde oluşturulan "mimikatz" isimli dosyaların izlenmesi bir tespit seçeneğidir. Ancak, dosya adı kolayca değiştirilebilir, bu nedenle atlanması kolaydır.

Sysmon yapılandırması:

```
<FileCreate onmatch="include">
        <TargetFilename condition="contains">mimikatz</TargetFilename>
</FileCreate>
```

Sysmon çıktısı:



Çıktıya bakıldığında "mimikatz.exe" dosyasının sıkıştırılmış dosyadan çıkarıldığı anlaşılıyor.

Monitoring hash

Mimikatz'a ait hash değerlerine sahip bir işlem başlatıldığında, Sysmon uyarı oluşturacak şekilde yapılabilir. Dosyada ufak bir değişiklikle hash değeri yenileneceği için bu yöntem de pek sağlıklı değil.

"mimikatz.exe"nin hash değerine bakıldığında "010D11288BAF561F633D674E715A2016" olduğu görülmektedir.

```
C:\Users\gunal\Desktop\mimikatz-master\x64>certutil -hashfile mimikatz.exe MD5
MD5 hash of file mimikatz.exe:
01 0d 11 28 8b af 56 1f 63 3d 67 4e 71 5a 20 16
CertUtil: -hashfile command completed successfully.
```

Dosyaya küçük bir ekleme yapıldığında hash değeri de değişecektir.

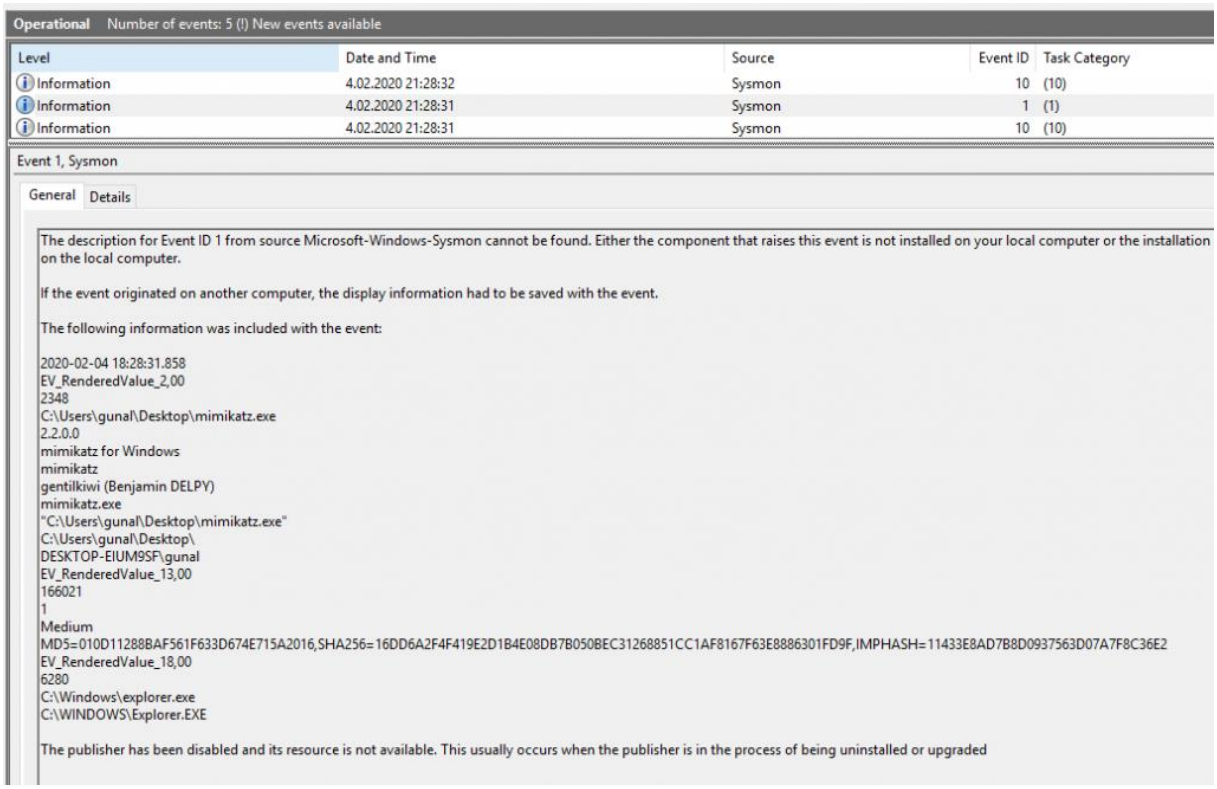
```
C:\Users\gunal\Desktop\mimikatz-master\x64>echo "gunal" >> mimikatz.exe

C:\Users\gunal\Desktop\mimikatz-master\x64>certutil -hashfile mimikatz.exe MD5
MD5 hash of file mimikatz.exe:
1c 43 04 f2 c4 f0 fe 42 b3 7b f1 a5 0d 1a 01 6d
CertUtil: -hashfile command completed successfully.
```

"010D11288BAF561F633D674E715A2016" karma değerine sahip dosyanın yürütülüp yürütülmediğini görmek için gereken yapılandırma:

```
<ProcessCreate onmatch="include">
        <Hashes condition="contains">010D11288BAF561F633D674E715A2016</Hashes>
</ProcessCreate>
```

sistem çıktısı:



”lsass.exe”

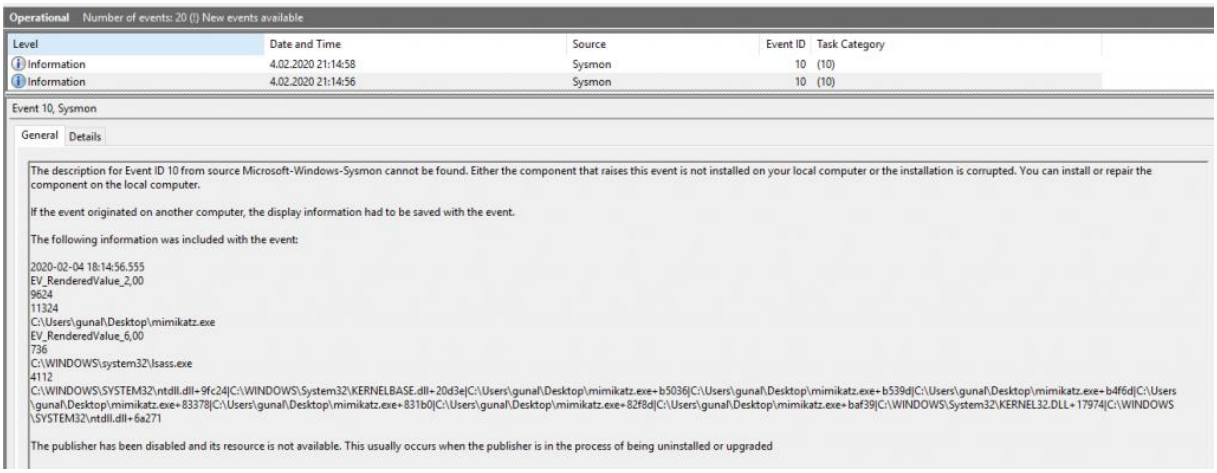
Mimikatz, parolaları yakalamak için lsass.exe'yi kullanır. "lsass.exe"nin izlenmesi ile onu kullanan işlemler de kayıt altına alınır. Bu sayede sadece mimikatz değil, lsass.exe kullanan tüm şüpheli işlemler kayıt altına alınır.

Yapılandırma:

```
<ProcessAccess onmatch="include">
        <TargetImage condition="contains">lsass.exe</TargetImage>
</ProcessAccess>
```

Hukuki faaliyetler için "lsass.exe" olarak adlandırılan işlemler, daha etkin sonuçlara ulaşmak için hariç tutulabilir.

Sysmon çıktısı:



Python ile İtibara Dayalı Tespit

İtibar tabanlı tespit sistemlerinin amacı, düşük itibarlı davranışları tespit etmektir (Düşük itibarlı dosya açma, IP adresi isteme). Bu sistemi kullanan bir ağ içinde düşük itibarlı IP adresi talep edildiğinde şüpheli durum fark edilecektir.

Temel olarak itibar tabanlı algılama sisteminin nasıl oluşturulacağını göstereceğim. İtibar verileri için 3. taraf kaynakları kullanacağım

Bazı veri kaynakları:

malwaredomainlist.com

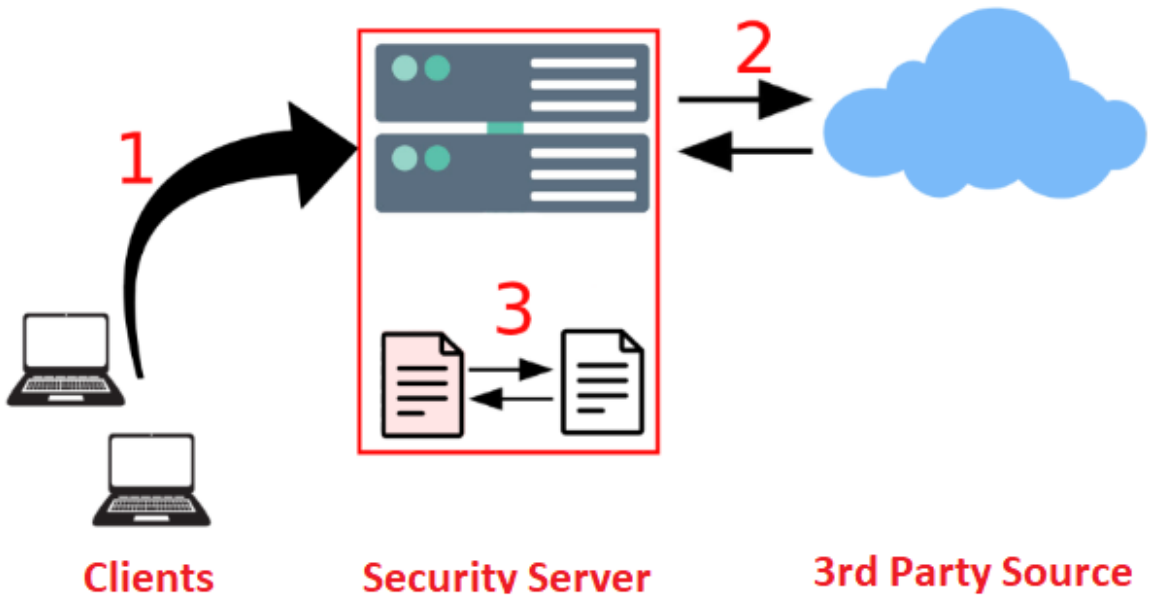
SANS

suistimal.ch

Sistem temel olarak 3 aşamadan oluşmaktadır:

1. Ağ trafiğini izleme

2. İtibar verileri için 3. taraf kaynaklardan veri toplama
3. Cihazların iletişim kurduğu adreslerin ve kaynaklardan gelen verilerin karşılaştırılması



Letsdefend.io ile gerçek bir SOC ortamında itibara dayalı vakaları inceleyebilirsiniz.

1- Ağ Trafiğini İzleme

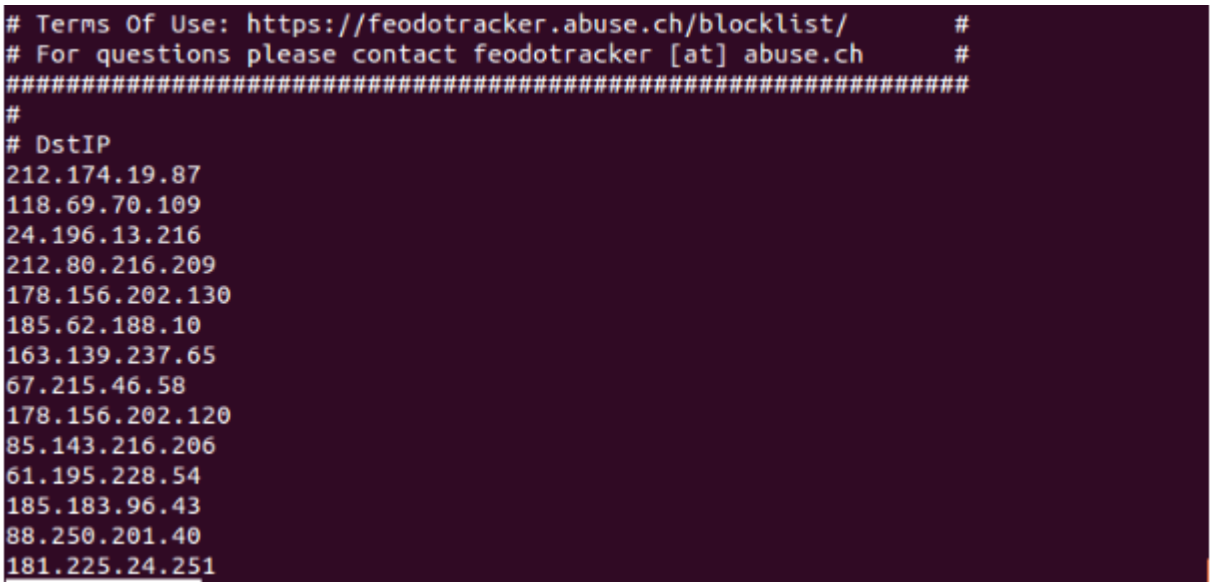
Verileri araştırmak için istemcilerin ağ günlüklerini güvenlik sunucusuna iletmeleri gerekir. Bu blogda loglar "tcpdump" ile kaydedilmekte ve ".pcap" formatında yönlendirilmektedir.

2- İtibar verileri için 3. taraf kaynaklardan veri toplama

Geçmişte kötü amaçlı yazılımlarla ilişkilendirilen bazı IP adresleri “abuse.sh” tarafından paylaşılanlar listesinden alınabilir. Veriler aşağıdaki resimdeki gibi basit bir script ile “IP_list.txt” dosyasına yazılır.

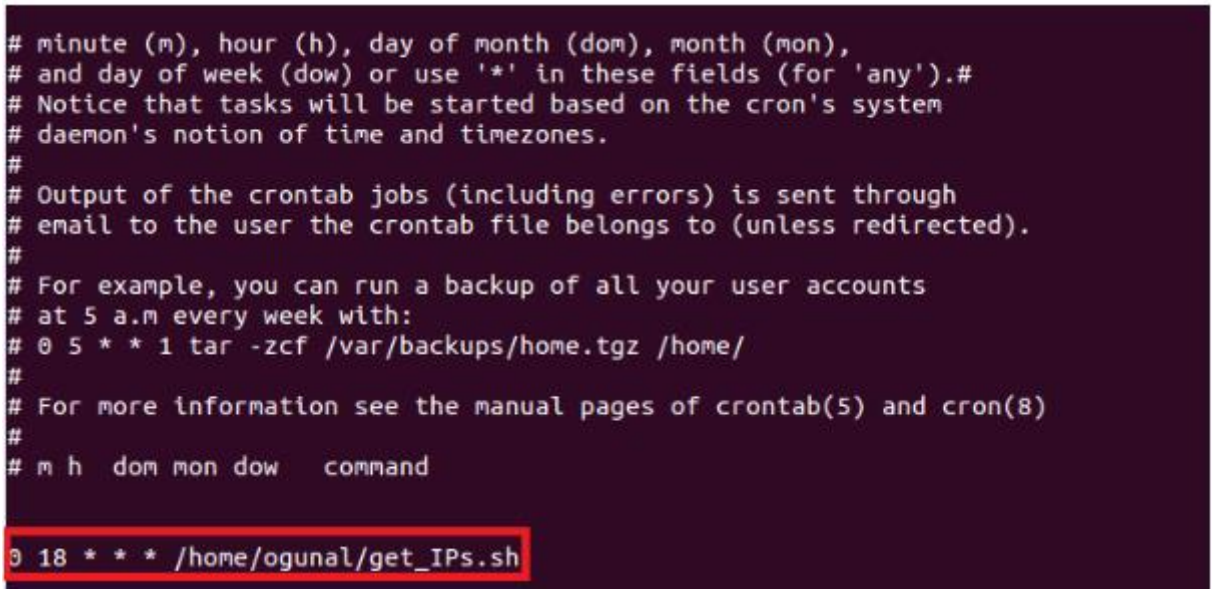
```
ogunal@ubuntu:~$ cat get_IPs.sh
curl https://feodotracker.abuse.ch/downloads/ipblocklist.txt >> IP_list.txt
```

"IP_list.txt" içeriği:



"IP_list.txt" dosyasına yeni veri ekleyebilmemiz için 3.parti servislerden günlük veri toplamamız gerekmektedir. Linux sistemlerinde "Crontab" yardımı ile "get_IPs.sh" betiği her gün otomatik olarak çalışabilmektedir.

Yeni işi crontab'da tanımlamak için "crontab -e" komutunu kullanıyorum ve betiğimin her gün saat 18:00'de çalışmasını sağlamak için alt satıra "0 18 * * * /home/ogunal/get_IPs.sh" ekliyorum.



Bu aşamada. 3. parti kaynaktan gelen verilerin günlük olarak yenilenmesini sağladık.

3-Cihazların iletişim kurduğu adreslerin ve kaynaklardan gelen verilerin karşılaştırılması

Bu aşamada, günlüklerden IP adreslerini çıkarır ve 3. taraf kaynak verilerini karşılaştırırız. Bir eşleşme olursa, bir uyarı oluştururuz.

İlk olarak, bu fonksiyonu loglardan IP adreslerini çıkarmak için hazırladım.

```
def pcap_parser():
    IP_list = []
    for p in PcapReader('log.pcap'):
        if IP in p:
            if p[IP].src not in IP_list:
                IP_list.append(p[IP].src)
            if p[IP].dst not in IP_list:
                IP_list.append(p[IP].dst)

    return(IP_list)
```

2. aşamada, IP_list.txt dosyasını okuyup verileri bir diziye ileten başka bir fonksiyon hazırladı.

```
def read_IPs():
    f = open('IP_list.txt', "r")
    lines = f.read().splitlines()
    f.close()
    return(lines)
```

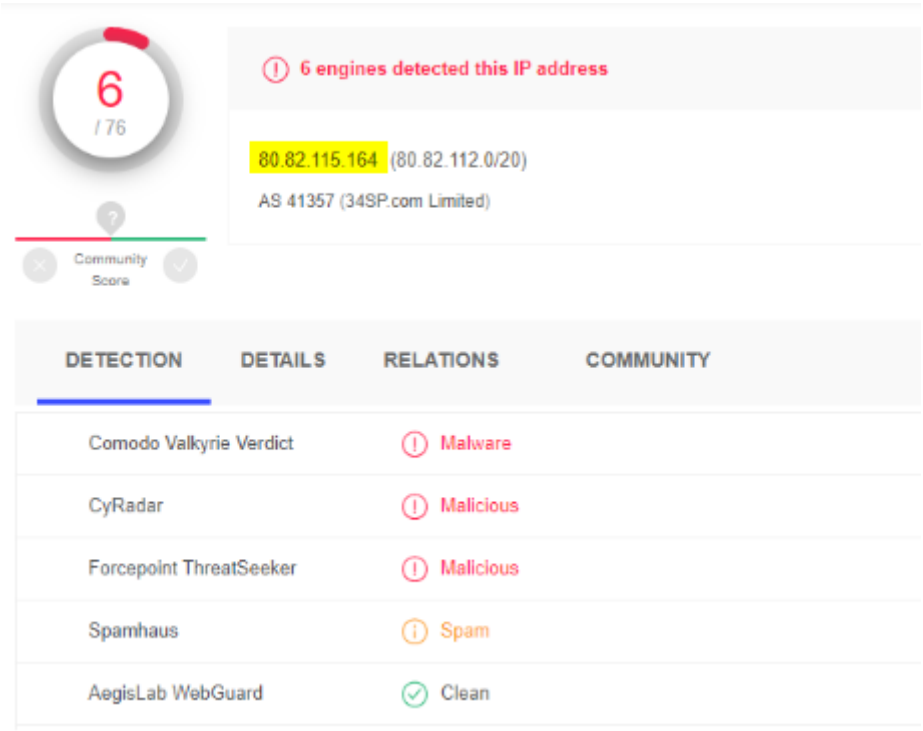
Mevcut 2 listenin kesişimini alarak itibarı düşük IP adresleri ile herhangi bir iletişim olup olmadığını belirleyen kodu hazırladım.

```
def compare(IP_addresses, bad_IPs):
    int = list(set(IP_addresses).intersection(bad_IPs))
    for i in int:
        print(i)
```

Uygulamayı çalıştırdığımda log kayıtlarında listelenen IP adreslerinden herhangi biri varsa uyarı veriyor.

```
ogunal@ubuntu:~$ python3 main.py
80.82.115.164
ogunal@ubuntu:~$
```

```
177.99.167.185
52.4.64.240
178.62.253.139
186.103.199.252
80.82.115.164
84.200.208.98
198.61.207.174
69.43.168.200
```



İtibara dayalı algılama sisteminin sorunları

- IP adresi birden fazla alana bağlıysa:
    - Bir yanda zararlı amaçlar için kullanılan bir domain var, diğer yanda sıradan bir blog bulunabiliyor.

- Bu durumda zararsız bloga yapılan istekler de IP adresi nedeniyle sistem tarafından şüpheli görünecektir.

- IP adresi sahibinin değişikliği
    - IP adresinin sahibi değiştiyse ve IP hala kara listeden biri. Yeni içerik talepleri de şüpheli olarak değerlendirilecektir.

Sysmon ile Proses Enjeksiyon Tespiti

Bu yazımızda proses enjeksiyon tekniğinin ne olduğunu ve Sysmon ile nasıl tespit edilebileceğini anlatacağız.

Proses Enjeksiyonu Nedir?

Basitçe söylemek gerekirse, başka bir işlemin adres alanında kod çalıştıran bir işlem, işlem enjeksiyonu olarak adlandırılır. Saldırganlar ve kötü amaçlı yazılımlar genellikle "Process Injection" tekniğini kullanır. Bu teknik sayesinde tespitin önüne geçerek saldırının başarı oranını arttırabilirler.

Process Injection tekniğini nasıl tespit edebileceğimizi açıklamadan önce bu tekniğin uygulanabileceği yöntemlerden bahsedelim.

Proses enjeksiyonu için birçok yöntem vardır. Bunlar arasında sıklıkla kullanılanlar aşağıdaki gibidir.

1. DLL Enjeksiyonu
2. PE Enjeksiyon
3. İşlem Boşluğu
4. Kanca Enjeksiyon
5. AppInit_DLL'ler

DLL Injection

P.E. Injection

Process Hollowing

Hook Injection

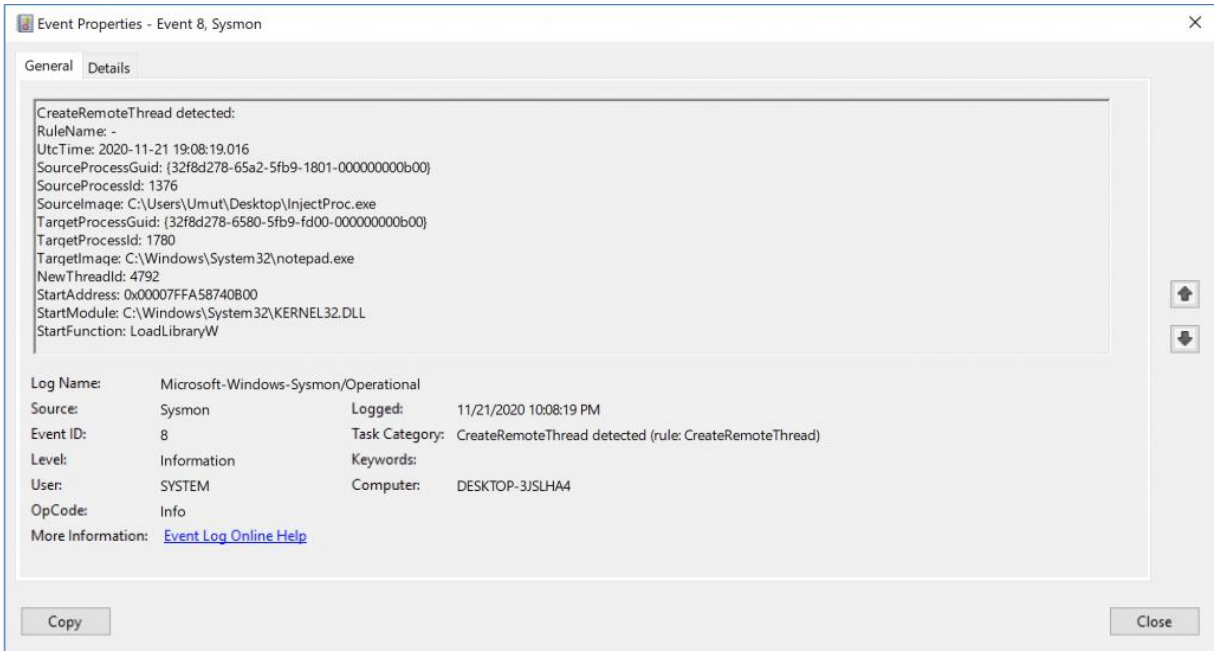
AppInit_DLLs

Sysmon

Sistem Monitörü **(** ***Sysmon*** **), bir sisteme yüklendikten sonra sistem etkinliğini izlemek ve Windows olay günlüğüne kaydetmek için sistem yeniden başlatmalarında yerleşik kalan bir Windows sistem hizmeti ve aygıt sürücüsüdür. İşlem oluşturma, ağ bağlantıları ve dosya oluşturma süresindeki değişiklikler hakkında ayrıntılı bilgi sağlar.**

https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Sysmon birçok farklı olayı kaydetmektir. Sysmon'un EventID:8 ile Process Injection tekniğini tespit edebiliyoruz.



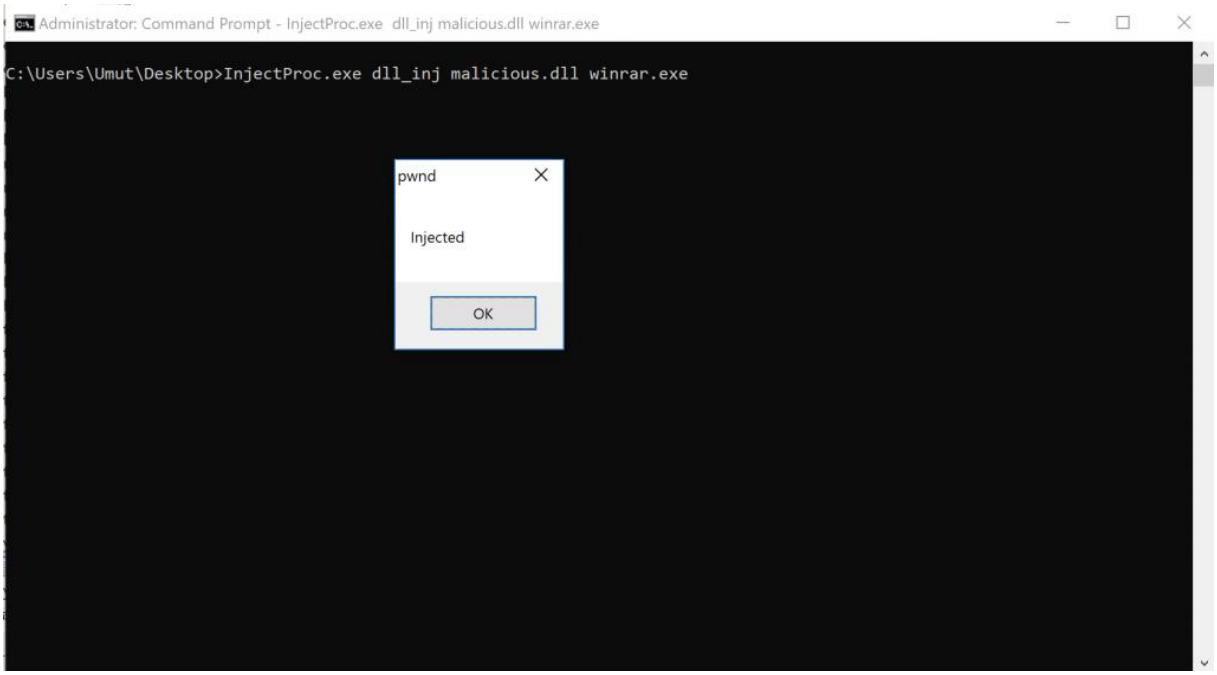
Misalen örneğin:

Sysmon Events ile Process Injection tekniğini nasıl tespit edebileceğimizi inceleyelim.

Process Injection tekniğini simüle etmek için InjectProc'u kullanabiliriz . InjectProc, Process Injection tekniğini simüle etmek için oluşturulmuş açık kaynaklı bir projedir. Ayrıca projede test etmeniz için oluşturulmuş bir dll dosyası bulunmaktadır.

InjectProc'un yürütülebilir dosyasını buradan indirebilirsiniz . Aşağıdaki komutla "winrar.exe" işlemine bir DLL enjekte edelim.

InjectProc.exe dll_inj kötü amaçlı.dll winrar.exe

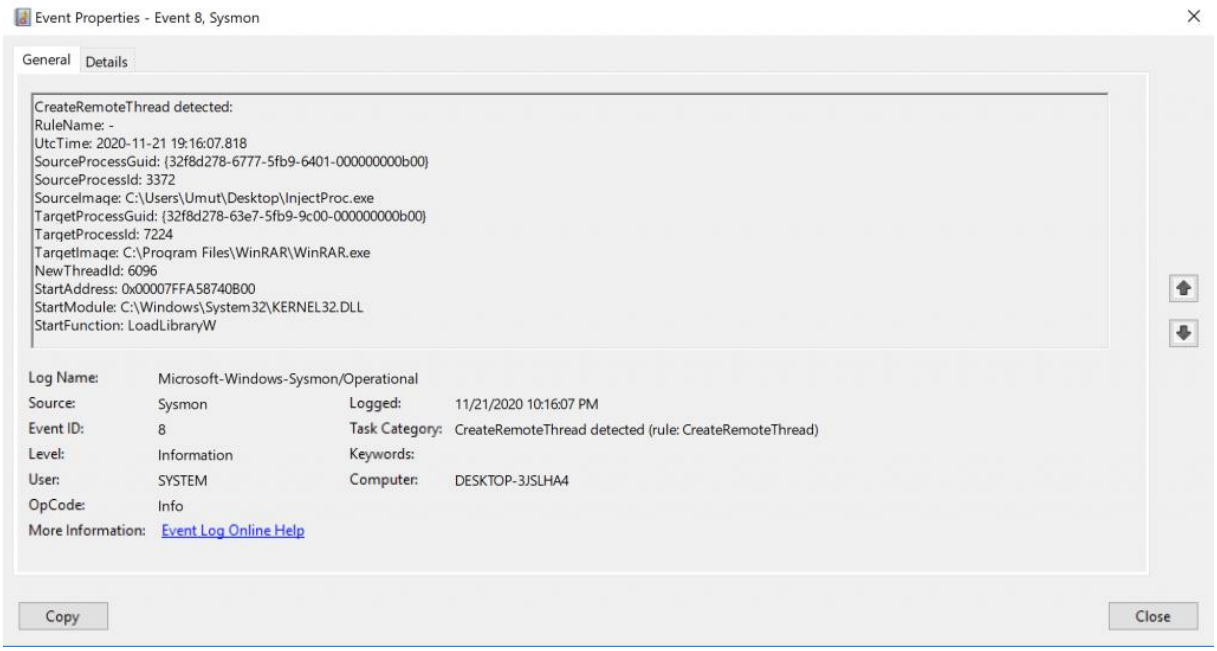


Dll'yi başarılı bir şekilde enjekte ettiğimizde, yukarıdaki resimdeki gibi enjeksiyonun başarılı olduğunu belirten mesaj kutusu çıkıyor.

Şimdi simüle ettiğimiz Sysmon ile Process Injection'ı tespit edelim. "Olay Görüntüleyici" aracı ile Sysmon Events'i görebiliriz. Sysmon günlükleri aşağıdaki dizinde bulunur.

"Uygulamalar ve Hizmet Günlükleri/Microsoft/Windows/Sysmon/Operational"

8 numaralı olaylara baktığımızda Process Injection'ın yakalandığı görülmektedir.



SIEM ürünlerinizde bu olayı takip ederek Process Injection tekniğini tespit edebilirsiniz.