



efrei

PARIS PANTHÉON - ASSAS UNIVERSITÉ



Architecture Sécurisé – Analyse d'un ransomware  
Par David TEJEDA, Vincent LAGOGUE et Thomas PEUGNET



I3 – 2024-2025

## Table des matières

### 1. Installation de theZoo

Clonage du repos github :

```
(kali@kali)-[~]
$ git clone https://www.github.com/ytisf/theZoo
Cloning into 'theZoo' ...
warning: redirecting to https://github.com/ytisf/theZoo.git/
remote: Enumerating objects: 3090, done.
remote: Counting objects: 100% (127/127), done.
remote: Compressing objects: 100% (100/100), done.
remote: Total 3090 (delta 25), reused 107 (delta 15), pack-reused 2963 (from 1)
Receiving objects: 100% (3090/3090), 1.06 GiB | 42.17 MiB/s, done.
Resolving deltas: 100% (668/668), done.
Updating files: 100% (1439/1439), done.
```

Installation des requirements :

```
(kali@kali)-[~/theZoo]
$ pip install --user -r requirements.txt
Requirement already satisfied: urllib3 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1)) (2.0.7)
Collecting pyminizip (from -r requirements.txt (line 2))
  Downloading pyminizip-0.2.6.tar.gz (261 kB)
  Preparing metadata (setup.py) ... done
Collecting pyzipper (from -r requirements.txt (line 3))
  Downloading pyzipper-0.3.6-py2.py3-none-any.whl.metadata (3.5 kB)
Requirement already satisfied: pycryptodomex in /usr/lib/python3/dist-packages (from pyzipper→-r requirements.txt (line 3)) (3.11.0)
Downloading pyzipper-0.3.6-py2.py3-none-any.whl (67 kB)
Building wheels for collected packages: pyminizip
  Building wheel for pyminizip (setup.py) ... done
  Created wheel for pyminizip: filename=pyminizip-0.2.6-cp311-cp311-linux_x86_64.whl size=203781 sha256=bc6c69053da5989cff527f7b0533ba6a0d0f27cdf6dd34ec167583291dddb55e
  Stored in directory: /home/kali/.cache/pip/wheels/50/c4/3c/6fb797c8b35d61411c595e7b2074dc657e4395a7ff525bbace
Successfully built pyminizip
Installing collected packages: pyminizip, pyzipper
Successfully installed pyminizip-0.2.6 pyzipper-0.3.6
```

Nous avons bien installé theZoo :

```

sMMs          oMMY
:ooooo/      /ooooo:
--+MMd-----hMMo+
oNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
/oodMMdooyMMMMMMMMMyoodMMdoo/   theZoo 0.6.0 'Moat'
`..dMMMMMY.:MMMMMMMM/ sMMMMMM..` DB ver. 1712294860000
dmmMMMMMMNNmmNNNNNNNNNNmmNNNNNNNNNN
NMMyoodMMMMMMMMMMMMMMMMMMMMMMdoosMMM   https://github.com/ytisf/theZoo
NMM- sMMMMNNNNNNNNNNNNNNNNNNNNMMY .MMM
NMM- sMMY`-----sMMY .MMM
ooo. :ooooooo+ +ooooooo/`ooo
      /MMMMN mMMMM+

authors: Yuval Nativ, Lahad Ludar, 5fingers
maintained by: Shahak Shalev, Yuval Nativ
github: https://github.com/ytisf/theZoo

mdb #> 
```

Malheureusement, nous ne parvenons pas à télécharger le code via theZoo :

```

mdb #> search rex
+-----+-----+-----+-----+-----+
| # | Type | Language | Architecture | Platform | Name |
+-----+-----+-----+-----+-----+
| 160 | ransomware | bin | x86 | linux | Rex |
+-----+-----+-----+-----+-----+
[+] Total records found: 1

mdb #> use 160
mdb Rex#> info
+-----+-----+-----+-----+-----+-----+-----+
| % | Name | Ver. | Author | Lang | Date | Arch. | Plat. | Tags |
+-----+-----+-----+-----+-----+-----+-----+
| ransomware | Rex | NA | NA | bin | NA | x86 | linux | Drupal |
+-----+-----+-----+-----+-----+-----+-----+
[+] Total records found: 1

mdb Rex#> get
[-] Error getting malware.
mdb Rex#> update-db
Traceback (most recent call last):
  File "/home/kali/theZoo/theZoo.py", line 109, in <module>
    main()
  File "/home/kali/theZoo/theZoo.py", line 104, in main
    terminalHandler.MainMenu()
  File "/home/kali/theZoo/imports/terminal_handler.py", line 91, in MainMenu
    self.actOnCommand(cmd)
  File "/home/kali/theZoo/imports/terminal_handler.py", line 121, in actOnCommand
    update_handler.update_db(db_ver)
  File "/home/kali/theZoo/imports/update_handler.py", line 67, in update_db
    print(red('[+]') + " A newer version is available: " + new_maldb_ver + "!")
    ~~~~~^~~~~~
TypeError: can only concatenate str (not "bytes") to str

```

Nous avons utilisé python 2.7 pour lancer theZoo et cette fois ça fonctionne :

```

mdb Rex#> get
Downloading: Ransomware.Rex.zip Bytes: 2843585
2843585 [100.00%]

Downloading: Ransomware.Rex.pass Bytes: 10
10 [100.00%]

Downloading: Ransomware.Rex.md5 Bytes: 53
53 [100.00%]

Downloading: Ransomware.Rex.sha256 Bytes: 85
85 [100.00%]

[+] Successfully downloaded a new friend.

```

Nous avons extrait le fichier :

```

(root@kali)-[/home/kali/theZoo]
# unzip Ransomware.Rex.zip
Archive: Ransomware.Rex.zip
[Ransomware.Rex.zip] WTEpZSFwgb password:
inflating: WTEpZSFwgb

```

### 3. Installation de clamav

Nous avons bien installé clamav :

```
(root@kali)-[/home/kali]
# sudo apt install clamav
The following packages were automatically installed and are no longer required:
ibverbs-providers libcephfs2 libglusterfs0 libpython3.11-dev python3-lib2to3 samba-vfs-modules
libassuan0 libgfsapi0 libgphoto2-l10n librados2 python3.11
libboost-iostreams1.83.0 libgfrpc0 libibverbs1 librdmacm1t64 python3.11-dev
libboost-thread1.83.0 libgfsxdr0 libperl5.38t64 perl-modules-5.38 python3.11-minimal
Use 'sudo apt autoremove' to remove them.

Upgrading:
blueman libldb2 libss2 onboard-common python3-ldb samba-common zstd
dpkg-dev libminizip1t64 libtalloc2 onboard-data python3-minimal samba-common-bin
gnutls-bin libpcr2-16-0 libtdb1 p11-kit python3-nassl samba-libs
ldap-utils libpython3-dev libtevent0t64 python3 python3-samba samba-vfs-modules
libdpkg-perl libpython3-stdlib libwbclient0 python3-arc4 python3-talloc smbclient
libjs-sphinxdoc libsass2-modules libxml2-utils python3-brotli python3-tdb tdb-tools
libldap-common libsmbclient0 onboard python3-dev samba xz-utils

Installing:
clamav

Installing dependencies:
clamav-base libnss-winbind python3.12 samba-ad-dc winbind
clamav-freshclam libpam-winbind python3.12-dev samba-ad-provision
libclamav12 libpython3.12-dev python3.12-minimal samba-dsdb-modules
```

## 4. Analyse Statique

### File

```
(root@kali)-[/home/kali/theZoo]
# file WTF25FWgb
WTF25FWgb: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, Go BuildID=fc5a3d09dbaf04f6
ec0587eae8c207fe211c5530, stripped
```

Le fichier analysé, nommé WTF25Payload, est un binaire exécutable au format ELF conçu pour s'exécuter sur des systèmes Linux. Il est destiné à une architecture x86 (32 bits), compatible avec les processeurs Intel 80386, et utilise un ordre des octets Little Endian (LSB). Ce fichier est statiquement lié, ce qui signifie qu'il contient toutes les bibliothèques nécessaires à son exécution, le rendant autonome. Par sa nature, il ne s'agit pas d'une librairie, mais d'un programme exécutable, probablement conçu pour cibler des systèmes Linux spécifiques.

### Strings

On a utilisé string pour extraire toutes les chaînes de caractères du binaire et ensuite on a utilisé grep pour rechercher des mots clés.

### Contenu du mail

```
09: failed to parse EC private key: x509: trailing data after X.509 key-idzip: unsupported compr
cause it doesn't contain any IP SANs%q is an incomplete or empty template%$2006-01-02 15:04:05.9
77555756156289135105907917022705078125?n=admin/config/system/site-informationGobDecoder: length
rMSpan_Sweep: bad span state after sweepSubject: ATTENTION: Ransom request!!!
EDNS: version can't represent recursive pointer type crypto/rsa: invalid options for Decryptgob
pointer of type http: putIdleConn: keep alive disabledinvalid indexed representation index %dm
n in MHeap_SysAllocmissing argument to repetition operatormultipart: can't write to finished par
titioned on addition of new data to field and, not defined as shared with no shared re
```

Le sujet est : ATTENTION : Ransom request !!!

```
response to channel open. tlts: ECDSA signature contained zero or negative values. bad signature type for client
s ECDSA certificatetls: failed to create cipher while encrypting ticket: tls: found unknown private key type in PKC
S#8 wrappingtls: server resumed a session with a different versiontls: unsupported signature type for client certif
icatex509: cannot verify signature: algorithm unimplementedx509: trailing data after X.509 CRL distribution point$S
From: Armada Collective <armada.collective@gmail.com>
-----END PUBLIC KEY----- FFFFFFFF00FDAA22168C234C6628B80DC1CD129024E088A67CC74020BBEA63B139B22514A08798E3
404DDEF9519B3CD3A431B302B0A6DF25F14374FE1356D0D51C245E485B76625E7EC6F44C42E9A637ED6B0BFF5C86F406B7EDEE386BFB5A899F
A5AE9F24117C4B1FE649286651ECE4583DC2007C88A163BF0598DA8361C5D39A69163FA8FD24CF5F83655D23DCA3AD961C62F356208552BB9
ED529077096966D670C354E4ABC9804F1746C08CA18217C32905E462E36C3BE39E772C180E8603982783A2EC07A28FB5C55DF06F4C52C9DE2B
CBF6955817183995497CEA956AE515D2261898FA051015728E5A8AACAA68FFFFFFFFFFFFFFFFFORWARD THIS MAIL TO WHOEVER IS IMPORTA
NT IN YOUR COMPANY AND CAN MAKE DECISION!
rex/scanner.(*Drupal).SetEmail
rex/scanner.(*RansomScanner).extractMailto
```

On a l'expéditeur du mail et la mention « Forward to whoever is important in your company and can make decision! ».



762a4f2bf5ea4ff72fce674da1adf29f0b9357be18de4cd992d79198c56bb514

42

/ 64

Community Score

-178

42/64 security vendors flagged this file as malicious

ReanalyzeSimilarMore

762a4f2bf5ea4ff72fce674da1adf29f0b9357be18de4cd992d79198c56bb514

Size7.28 MB

Last Analysis Date1 month ago

ELF

elfssh-communicationexploit

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY17

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat labeltrojan.elfrexxdos/mjss

Threat categoriestrojanransomware

Family labelselfrexxdosmjss

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3

Linux/Rex

ALYac

Trojan.Ransom.Drupal

Antiy-AVL

Trojan/Linux.Rex.a

Arcabit

Trojan.Generic.D2220601

Avast

ELF:Rex-A [Trj]

AVG

ELF:Rex-A [Trj]

Avira (no cloud)

LINUX/Rex.mjss

BitDefender

Trojan.Generic.35784193

ClamAV

Unix.Malware.Agent-1628853

CTX

Elf.trojan.generic

SentinelOne (Static ML)

Static AI - Malicious ELF

Skyhigh (SWG)

Generic Trojan.hf

Sophos

Mal/Generic-S

Symantec

Trojan Horse

Tencent

Linux.Trojan.Rex.Qqil

Trellix (ENS)

Generic Trojan.hf

Trellix (HX)

Trojan.Generic.35784193

TrendMicro

Ransom\_ElfRexDDOS.A

TrendMicro-HouseCall

Ransom\_ElfRexDDOS.A

Varist

E32/DCRex.WLOS-32

VIPRE

Trojan.Generic.35784193

WithSecure

Malware.LINUX/Rex.mjss

Xcitium

Malware@#2m1kdg0st2b61

Zillya

Trojan.Rex.Linux.19

Acronis (Static ML)

Undetected

AliCloud

Undetected

Avast-Mobile

Undetected

Baidu

Undetected

Bkav Pro

Undetected

CMC

Undetected

CrowdStrike Falcon

Undetected

Gridinsoft (no cloud)

Undetected

Jiangmin

Undetected

K7AntiVirus

Undetected

K7GW

Undetected

Malwarebytes

Undetected

MaxSecure

Undetected

Panda

Undetected

## Détection des menaces sur VirusTotal : comprendre les angles morts

Vous avez peut-être remarqué que certains antivirus ne signalent pas un fichier suspect lors d'un scan. Plusieurs mécanismes expliquent ces faux négatifs. D'abord, le ransomware pourrait être une toute nouvelle version (*zero-day*), non répertoriée dans les bases de signatures. Certains éditeurs mettent parfois plusieurs jours à mettre à jour leurs définitions, surtout face à des codes polymorphes qui mutent à chaque infection.

Autre piste : les techniques d'obfuscation. En compressant ou chiffrant son code (via des packers comme UPX ou des outils maison), le malware devient une énigme pour les analyseurs statiques. Même l'analyse heuristique peut échouer si le moteur antivirus n'intègre pas les dernières règles de détection comportementale.

## Cas concret avec Ransomware.Rex

Lors d'un test en sandbox, après avoir exécuté :

```
chmod u+x Ransomware.Rex
./Ransomware.Rex -debug -log.http -log.dht # (sans connexion
réseau)
```

```
(kali@kali)-[~/theZoo]
└─$ ping 7.7.7.7
ping: connect: Network is unreachable
```

```
(kali@kali)-[~/theZoo]
└─$ chmod u+x WTEpZSFwgb

(kali@kali)-[~/theZoo]
└─$ ./WTEpZSFwgb -debug -log.http -log.dht
```

On observe un comportement révélateur : dès le lancement, le malware tente de contacter un serveur externe via HTTP ou le réseau DHT (comme un nœud BitTorrent). Mais sans internet, cette tentative échoue silencieusement.

```
(kali@kali)-[~/theZoo]
└─$ ./WTEpZSFwgb -debug -log.http -log.dht

*HTTP.Do GET https+verify://ipinfo.io/ip: Get https+verify://ipinfo.io/ip: dial tcp: lookup ipinfo.io on [::1]:53: read udp [::1]:48109->[::1]:53: read: connection refused
*HTTP.Do GET https+verify://ip4.icanhazip.com: Get https+verify://ip4.icanhazip.com: dial tcp: lookup ip4.icanhazip.com on [::1]:53: read udp [::1]:58123->[::1]:53: read: connection refused
*HTTP.Do GET http://www.trackip.net/ip7json: Get http://www.trackip.net/ip7json: dial tcp: lookup www.trackip.net on [::1]:53: read udp [::1]:57829->[::1]:53: read: connection refused
^C
```

Ce qui est intrigant : le programme refuse de poursuivre son exécution sans avoir reçu une réponse spécifique, probablement une clé de géolocalisation ou un ordre du C&C (serveur de commandement). Cette dépendance explique pourquoi l'infection peut rester dormante dans certains environnements de test - une faille que les solutions antivirus traditionnelles ne captent pas toujours.

## 5. Analyse Dynamique

### • Rapport d'Analyse Dynamique

#### Rapport d'Analyse Dynamique

### 1. Introduction

L'analyse dynamique, par opposition à l'analyse statique, **observe un programme en cours d'exécution** afin de comprendre son comportement réel. L'outil strace permet de tracer tous les appels système que le programme effectue (ouverture de fichiers, connexions réseau, etc.).

- **Intérêt** : Voir concrètement ce que le programme fait au niveau du système (création/lecture de fichiers, connexions à des sockets, etc.).
- **Exemple** : Sur un binaire inoffensif comme whoami, strace montre clairement le cheminement du programme. Sur un malware comme REX, c'est vital pour repérer ses actions malveillantes.

## 2. Observations de strace sur la commande whoami

### 2.1. Processus général

En lançant :



```

scatpet          spa-resample
scanimage        sptoppm
scapy             spd-conf
└─(kali@kali)-[~/theZoo]
$ strace whoami
execve("/usr/bin/whoami", ["whoami"], 0x7ffc40e5c400 /* 56 vars */) = 0
brk(NULL)                = 0x5605bbf1f000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x
7f7a4595d000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or director
y)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=91951, ...}, AT_EMPTY_PATH)
= 0
mmap(NULL, 91951, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f7a45946000
close(3)                 = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P-\2\0\0\0\0" ... , 83
2) = 832
pread64(3, "\6\0\0\0\4\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0" ...
, 784, 64) = 784
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1933688, ...}, AT_EMPTY_PATH
) = 0
pread64(3, "\6\0\0\0\4\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0" ...
, 784, 64) = 784
mmap(NULL, 1985936, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f7a45761
000
mmap(0x7f7a45787000, 1404928, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_
DENYWRITE, 3, 0x26000) = 0x7f7a45787000
mmap(0x7f7a458de000, 348160, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x17d000) = 0x7f7a458de000
mmap(0x7f7a45933000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_D
ENYWRITE, 3, 0x1d1000) = 0x7f7a45933000
mmap(0x7f7a45939000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_A
NONYMOUS, -1, 0) = 0x7f7a45939000
close(3)                 = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0
x7f7a4575e000
arch_prctl(ARCH_SET_FS, 0x7f7a4575e740) = 0
set_tid_address(0x7f7a4575ea10) = 29300
set_robust_list(0x7f7a4575ea20, 24) = 0
rseq(0x7f7a4575f060, 0x20, 0, 0x53053053) = 0
mprotect(0x7f7a45933000, 16384, PROT_READ) = 0
mprotect(0x5605bb9fc000, 4096, PROT_READ) = 0
mprotect(0x7f7a4598f000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINIT
Y}) = 0
munmap(0x7f7a45946000, 91951) = 0

```

1. **execve(...)** : Démarrage du binaire `/usr/bin/whoami`.
2. **openat(...)** : Lecture des bibliothèques dynamiques (ex. `/lib/x86_64-linux-gnu/libc.so.6`).
3. **getrandom(...)** et **seteuid(...)** : Configuration d'éléments de sécurité et récupération de l'UID effectif.
4. **openat("/etc/passwd") + read(...)** : Recherche du nom d'utilisateur associé à l'UID.
5. **write(1, "kali\n", 5)** : Écriture finale du résultat sur la sortie standard.

- whoami se contente de vérifier l'utilisateur courant via /etc/passwd.
- Les multiples openat, mmap, etc. relèvent surtout de la résolution dynamique des librairies et de la configuration mémoire.
- Le programme n'initie aucune action « malveillante » : il se limite à renvoyer le nom de l'utilisateur en cours.

### 3.1. Contexte et fichier strace.txt

Pour un ransomware comme REX, on **capture** également les appels système avec strace. Dans le TP, on dispose déjà d'un **fichier strace.txt** qui montre ce que REX fait au démarrage :

- **Ouverture de ressources réseau :**
  - Par exemple, `openat(AT_FDCWD, "/proc/sys/net/core/somaxconn", ...)`, qui suggère une vérification du paramètre système sur le nombre maximal de connexions.
- **Tentative de connexion à un C&C :**
  - `connect(8, {sa_family=AF_INET, sin_port=htons(5099), sin_addr=inet_addr("83.241.220.100")}, 16) = -1 EINPROGRESS.`
  - Cela indique que REX essaie de se connecter à un serveur distant.

### 3.2. Interprétation

1. **Paramétrage réseau :** REX souhaite peut-être évaluer la capacité du système à accepter de multiples connexions (utile pour DDoS ou propagation).
2. **Communication avec un serveur de commande et de contrôle :** Le ransomware pourrait y récupérer des instructions ou signaler son activité.
3. **Éventuelles actions locales :** Lecture ou écriture de fichiers spécifiques, comme une étape de préparation au chiffrement.

En analysant la trace, on comprend mieux **quand** et **comment** le malware déclenche son côté malveillant (par exemple, seulement si la géolocalisation renvoie une certaine réponse, comme mentionné dans l'énoncé).

## 4. Conclusion

Grâce à strace, nous constatons :

- **Sur whoami :** Lecture de `/etc/passwd` pour déterminer l'utilisateur. Processus simple et transparent.
- **Sur REX :** Des ouvertures de fichiers système spécifiques, des connexions à un serveur externe, et potentiellement d'autres appels signalant des fonctions de chiffrement ou d'exfiltration.

**Bilan :** L'analyse dynamique donne **une vision très concrète** des opérations effectuées par un binaire, bien plus parlante que la simple observation du code (analyse statique). Dans un contexte de sécurité, cela aide à confirmer et documenter les activités réelles d'un malware, et donc à mieux s'en prémunir ou l'endiguer.