YET ANOTHER TALK ABOUT YARA

FERNANDA BILLORDO ZINI & AYELEN TORELLO

```
le Winnti_malware_Nsiproxy {

 meta:
     description = "Detects a Winnt
     author = "Florian Roth"
     date = "2015-10-10"
     score = 75
     hash1 = "9001572983d5b1f997872
     hash2 = "cf1e006694b33f27d7c74
     hash3 = "326e2cabddb641777d489
     hash4 = "aff7c7478fe33c57954b6
     hash5 = "ba7ccd027fd2c826bbe8f

 strings:
     $x1 = "\\Driver\\nsiproxy" ful
     $a1 = "\\Device\\StreamPortal"
     $a2 = "\\Device\\PNTFILTER" fu
     $s1 = "Cookie: SN=" fullword a
     $s2 = "\\BaseNamedObjects\\_tr
     $s3 = "Winqual.sys" fullword w
     $s4 = "\\Registry\\Machine\\SY
     $s5 = "http://www.wasabii.com.

 condition:
     uint16(0) == 0x5a4d and $x1 an
```

BLUESPACE EDITION

SOBRE NOSOTRAS

Fernanda Billordo Zini

Consultora y desarrolladora en el área de Cyber Threat Intelligence en Deloitte.

Ayelen Torello

Senior Security Researcher en el área de Cyber Threat Intelligence en Deloitte

Las opiniones expresadas son únicamente nuestras y no expresan los puntos de vista u opiniones de nuestro empleador

¿QUÉ ES YARA?

Definición

YARA es una herramienta que tiene como objetivo ayudar a los investigadores de malware a **identificar** y **clasificar** muestras de archivos maliciosos. Con YARA puedes crear descripciones de familias de malware basados en textos o patrones del binario y así descubrir nuevas muestras.

- Es **gratis y Open-Source**
- Tiene un gran soporte tanto de la comunidad como de la industria
- Desarrollado por VirusTotal

ESCRIBIENDO REGLAS DE YARA

ESTRUCTURA DE UNA REGLA DE YARA

```
rule <rule_name> {
        meta:
                description= "Sample YARA rule"

        strings:
                $a = "example"
                $b = "example2"

        condition:
                ($a or $b)
}
```

EJEMPLO DE UNA REGLA DE YARA

```
rule silent_banker : banker {
        meta:
                description = "This is just an example"
                threat_level = 3
                in_the_wild = true
        strings:
                $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
                $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
                $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"
        condition:
                ($a or $b)
}
```

¡¡ETIQUETAS!!

CÓMO REALMENTE SE VE UNA REGLA DE YARA

```
rule MAL_Emotet_Jan20_1 {
    meta:
        description = "Detects Emotet malware"
        author = "Florian Roth"
        reference = "https://app.any.run/tasks/5e81638e-df2e-
        4a5b-9e45-b07c38d53929/"
        date = "2020-01-29"
        hash1 =
        "e7c22ccdb1103ee6bd15c528270f56913bb2f47345b360802
        b74084563f1b73d"
    strings:
        $op0 = { 74 60 8d 34 18 eb 54 03 c3 50 ff 15 18 08 41 00 }
        $op1 = { 03 fe 66 39 07 0f 85 2a ff ff ff 8b 4d f0 6a 20 }
        $op2 = { 8b 7d fc 0f 85 49 ff ff ff 85 db 0f 84 d1 }
    condition:
        uint16(0) == 0x5a4d and filesize < = 200KB and (
         pe.imphash() == "009889c73bd2e55113bf6dfa5f395e0d"
        or 1 of them      )
}
```

```
rule MAL_Emotet_Jan20_1 {
    meta:
        description = "Detects Emotet malware"
        author = "Florian Roth"
        reference = "https://app.any.run/tasks/5e81638e-df2e-
        4a5b-9e45-b07c38d53929/"
        date = "2020-01-29"
        hash1 =
        "e7c22ccdb1103ee6bd15c528270f56913bb2f47345b360802
        b74084563f1b73d"
    strings:
        $op0 = { 74 60 8d 34 18 eb 54 03 c3 50 ff 15 18 08 41 00 }
        $op1 = { 03 fe 66 39 07 0f 85 2a ff ff ff 8b 4d f0 6a 20 }
        $op2 = { 8b 7d fc 0f 85 49 ff ff ff 85 db 0f 84 d1 }
    condition:
        uint16(0) == 0x5a4d and filesize < = 200KB and (
         pe.imphash() == "009889c73bd2e55113bf6dfa5f395e0d"
        or 1 of them      )
}
```

Información y contexto sobre la regla de YARA

Patrones interesantes para buscar

Condiciones para encontrar muestras

TIPOS DE STRINGS

Strings hexadecimales

```
$hex1 = {E2 34 ?? C? [4-6] B4}
$hex2 = { 8b 7d fc 0f 85 49 ff ff ff 85 db 0f 84 d1 }
```

TIPOS DE STRINGS

Strings hexadecimales

Wilcards

Jumps

Alternatives

$hex1 = {E2 34 ?? C? [4-6] B4}
$hex2 = { 8b 7d fc 0f 85 49 ff ff ff 85 db 0f 84 d1 }

Representados con ?
Un ? reemplaza un caracter hexadecimal

Para longitudes desconocidas. Sintaxis: [X-Y]
[4-6] -> secuencia arbitraria desde 4 a 6 bytes

Para proporcionar una alternativa diferente
$hex_string = { F4 23 (62 B4 | 56) 45 }

TIPOS DE STRINGS

Strings hexadecimales

```
$hex1 = {E2 34 ?? C? [4-6] B4}
$hex2 = { 8b 7d fc 0f 85 49 ff ff ff 85 db 0f 84 d1 }
```

Strings de textos

```
$text = "Texto de ejemplo"
```

TIPOS DE STRINGS

Strings hexadecimales

```
$hex1 = {E2 34 ?? C? [4-6] B4}
$hex2 = { 8b 7d fc 0f 85 49 ff ff ff 85 db 0f 84 d1 }
```

Strings de textos

```
$text = "Texto de ejemplo"
```

Expresiones regulares

```
$re1 = /md5: [0-9a-fA-F]{32}/
```

ESCRIBIENDO REGLAS DE YARA

KEYWORDS

- fullword — $full = "This program cannot" fullword
- ascii — $ascii= "BlueSpace" wide ascii
- wide — $wide = "BlueSpace" wide
- nocase — $text = "bluespace" nocase
- xor — $xor = "This program cannot" xor
- base64 — $b64 = "This program cannot" base64
- base64wide — $b64wide= "BlueSpace" base64wide

ESCRIBIENDO REGLAS DE YARA

KEYWORDS

• fullword	Busca sólo fullwords, no substrings.
• ascii	Busca por strings del tipo ASCII (default).
• wide	Busca por strings del tipo Unicode.
• nocase	No distinguir entre mayúsculas y minúsculas.
• xor	Strings con un XOR de un solo byte.
• base64	Strings que han sido codificados en base64.
• base64wide	Igual que base64 pero convertido en wide.

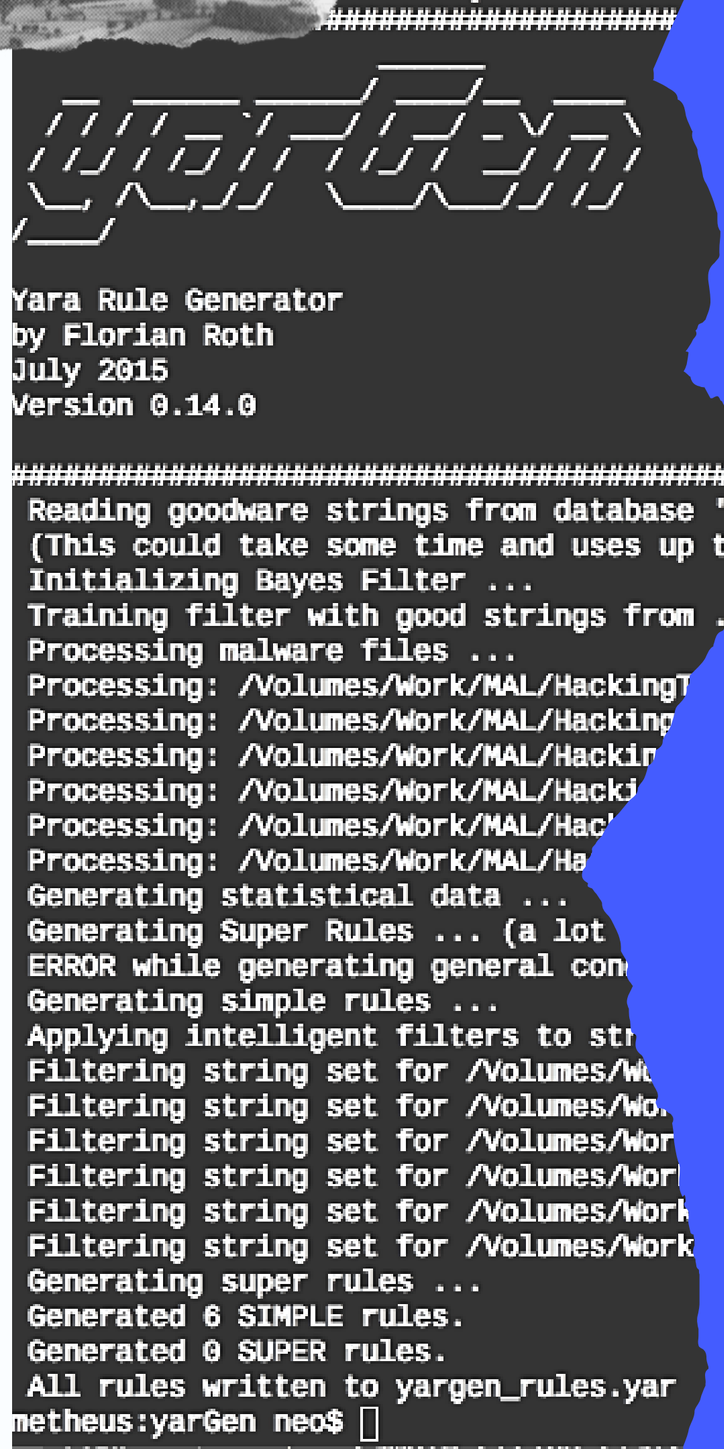
Herramientas e implementaciones

YARGEN

YarGen es una herramienta que se utiliza para generar reglas de YARA de forma automatizada. Permite generar reglas a partir de un archivo malicioso.

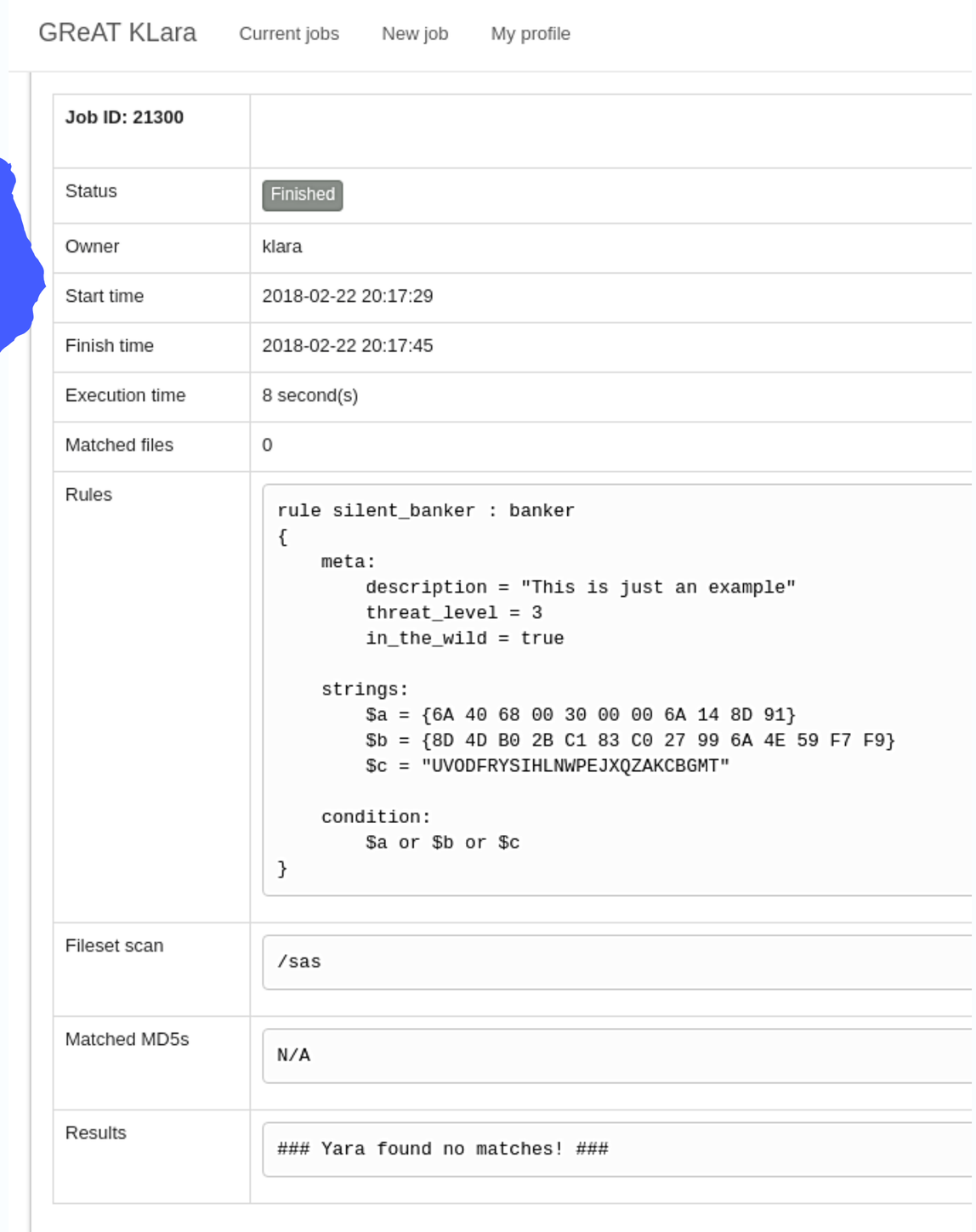
- Incluye una gran base de datos de strings y opcodes que también aparecen de forma recurrente en archivos no maliciosos.

https://github.com/Neo23x0/yarGen



Herramientas e implementaciones

Herramientas e implementaciones



KLARA

KLara es un sistema distribuido desarrollado en Python que permite escanear una o más reglas de YARA en colecciones de muestras. También se pueden obtener notificaciones vía email o en su interfaz web cuando los resultados del escaneo esten listos.

https://github.com/KasperskyLab/klara

YAYA

YET ANOTHER YARA AUTOMATON

- Manejo de conjuntos de reglas OpenSource
- Ignora las reglas problemáticas
- Permite agregar tus propias reglas
- Corre escaneos

https://github.com/cooperq/yaya

```
cooperq@lucyparsons:~$ yaya
YAYA - Yet Another Yara Automaton
Usage:
yaya [-h] <command> <path>
        -h        print this help screen
Commands:
        update - update rulesets
        edit - ban or remove rulesets
        add - add a custom ruleset, located at <path>
        scan - perform a yara scan on the directory at <path>
```

```
cooperq@lucyparsons:~$ yaya edit
 Enabled ID                                        Name   Description
 enabled 1                          AlienVault Labs Rules       - Collection of tools, signatures, and ru
 enabled 2                               bamfdetect rules       - Custom rules from Brian Wallace used fo
 enabled 3                         BinaryAlert YARA Rules       - A couple dozen rules written and releas
 enabled 4                                Burp YARA Rules       - Collection of YARA rules intended to be
 enabled 5                                   BinSequencer       - Find a common pattern of bytes within a
disabled 6                              Brian Carter Rules       - Collection of personal rules written by
 enabled 7                                    CAPE Rules       - Rules from various authors bundled with
 enabled 8                                     CDI Rules       - Collection of YARA rules released by [C
 enabled 9                  Citizen Lab Malware Signatures       - YARA signatures developed by Citizen La
 enabled 10                        ConventionEngine Rules       - A collection of Yara rules looking for
 enabled 11                               Deadbits Rules       - A collection of YARA rules made public
 enabled 12                         Didier Stevens Rules       - Collection of rules from Didier Stevens
 enabled 13                                    ESET IOCs       - Collection of YARA and Snort rules from
 enabled 14                               Fidelis Rules       - You can find a half dozen YARA rules in
 enabled 15                            Florian Roth Rules       - Florian Roth's signature base is a freq
 enabled 16                          Franke Boldewin Rules      - A collection of YARA Rules from [@r3c0n
 enabled 17                                    FSF Rules       - Mostly filetype detection rules, from t
 enabled 18                       GoDaddy ProcFilter Rules      - A couple dozen rules written and releas
 enabled 19                                  h3x2b Rules       - Collection of signatures from h3x2b whi
```

VIRUSTOTAL

Help

Stuxnet_Malware

```
rule Stuxnet_Malware_3
{
meta:
      description = "Stuxnet Sample - file ~WTR4141.tmp"
      author = "Florian Roth"
      reference = "Internal Research"
      date = "2016-07-09"
      hash1 = "6bcf88251c876ef00b2f32cf97456a3e306c2a263d487b0a50216c6e3cc07c6a"
      hash2 = "70f8789b03e38d07584f57581363afa848dd5c3a197f2483c6dfa4f3e7f78b9b"

   strings:
      $x1 = "SHELL32.DLL.ASLR." fullword wide
      $s1 = "~WTR4141.tmp" fullword wide
      $s2 = "~WTR4132.tmp" fullword wide
      $s3 = "totalcmd.exe" fullword wide
      $s4 = "wincmd.exe" fullword wide
      $s5 = "http://www.realtek.com0" fullword ascii
      $s6 = "{%08x-%08x-%08x-%08x}" fullword wide

   condition:
      ( uint16(0) == 0x5a4d and filesize < 150KB and ( $x1 or 3 of ($s*) ) ) or ( 5 of them )
}
```

Agregá tu conjunto de reglas en VirusTotal y recibí notificaciones cuando alguna de tus reglas haya sido activada por algún archivo al ser subido a la plataforma.

DEMO TIME

RECURSOS

DOCUMENTACIÓN

YARA Official documentation = https://yara.readthedocs.io/

CONJUNTO DE REGLAS

YaraRules Project - Official Repo = https://github.com/Yara-Rules/rules
Florian Roth YARA Rules = https://github.com/Neo23x0/signature-base
ReversingLabs YARA Rules = https://github.com/reversinglabs/reversinglabs-yara-rules
McAfee YARA Rules = https://github.com/advanced-threat-research/Yara-Rules
CitizenLab YARA Rules = https://github.com/citizenlab/malware-signatures

Y muchos mas ...

GRACIAS

Fernanda Billordo Zini

@Cyb3rR4v3n

¡Sigamos en contacto!
Seguinos en Twitter

Ayelen Torello

@Cyb3rhuskys