



Gotta catch 'em all!

This page intentionally left blank.

because if books can do it I can do it too lol



source: <https://imgur.com/gallery/U0gbwIn>




source: <https://i.imgur.com/siHEVRy.gif>

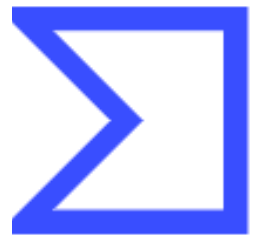
What is YARA?

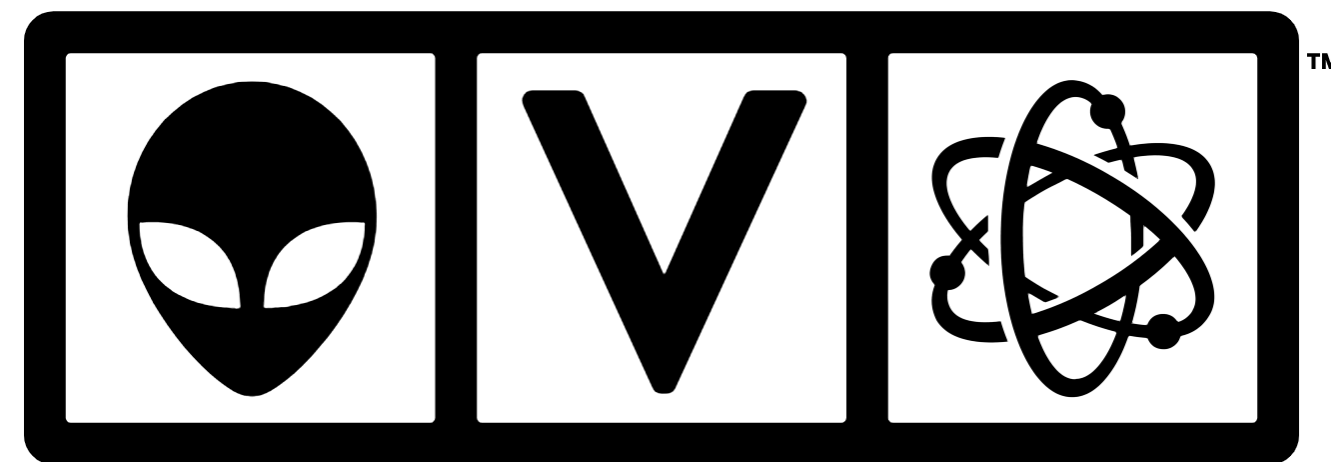
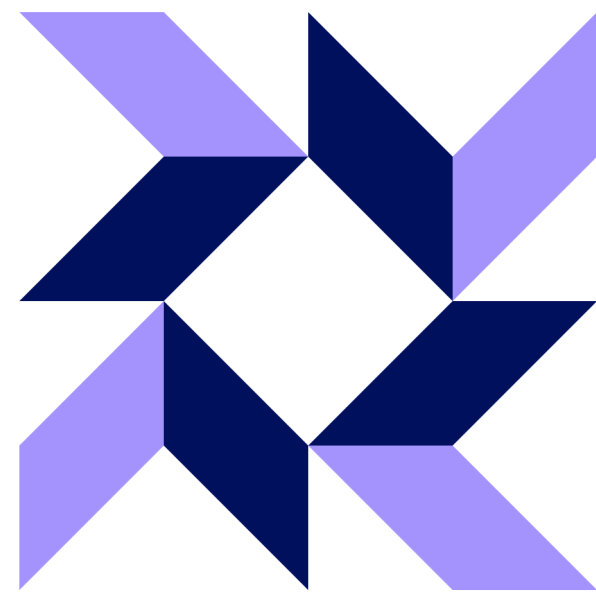
YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples. With YARA you can create descriptions of malware families (or whatever you want to describe) based on textual or binary patterns. Each description, a.k.a. rule, consists of a set of strings and a boolean expression which determine its logic.

<https://yara.readthedocs.io/en/stable/>

OR the pattern matching swiss knife for malware researchers (and everyone else) developed by Victor M. Alvarez (@plusvic) along with VirusTotal.

Used by...  VIRUSTOTAL

Used by...  VIRUSTOTAL



... almost everyone

You can use it too.

- Command line tool: `/usr/local/bin/yara`
 - `yarac` if you want to compile the rule beforehand
- C API: `libyara`
 - Python: `yara-python`
 - Perl: `Parse::YARA`
 - Go: `go-yara`
 - Ruby: `Yara::Rules`
 - Rust: `yara-rust`
 - etc... *you get the idea.*

Not only for malware research

- YARA rules can be utilised for any kind of identification and classification tasks.
- For example: list all images in the Pictures folder that are JPEG and PNG but don't carry any metadata (/rules/img.yar).

```
rule images : my_tools search_tool {  
  meta:  
    description = "Get all JPEG and PNG images, avoiding EXIF metadata"  
  strings:  
    $jpg = { FF D8 FF (DB|E0|EE|E1) }  
    $str = "Exif" nocase  
  condition:  
    ( $jpg and not $str ) or uint32be(0) == 0x89504E47 /* png */  
}
```

Example rule

```
import "pe"
rule silent_banker : banker
{
    // c++ comment
    /* c style comment */
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a      = {6A 40 68 00 30 00 [01-04] 6A 14 8D 91}
        $b      = {8D 4D B0 2B C1 ?? C? 27 99 6A 4E 59 F7 F9}
        $c01    = "ABC" fullword wide
        $c02    = /DEF/ ascii nocase
    condition:
        pe.characteristics & pe.EXECUTABLE_IMAGE and uint16be(0) == 0x4d5a and (
            $a or $b or any of ($c*)
        )
}
```

Example rule

```
import "pe"
rule silent_banker : banker
{
    // c++ comment
    /* c style comment */
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a      = {6A 40 68 00 30 00 [01-04] 6A 14 8D 91}
        $b      = {8D 4D B0 2B C1 ?? C? 27 99 6A 4E 59 F7 F9}
        $c01    = "ABC" fullword wide
        $c02    = /DEF/ ascii nocase
    condition:
        pe.characteristics & pe.EXECUTABLE_IMAGE and uint16be(0) == 0x4d5a and (
            $a or $b or any of ($c*)
        )
}
```

Example rule

```
import "pe"
rule silent_banker : banker
{
    // c++ comment
    /* c style comment */
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a      = {6A 40 68 00 30 00 [01-04] 6A 14 8D 91}
        $b      = {8D 4D B0 2B C1 ?? C? 27 99 6A 4E 59 F7 F9}
        $c01    = "ABC" fullword wide
        $c02    = /DEF/ ascii nocase
    condition:
        pe.characteristics & pe.EXECUTABLE_IMAGE and uint16be(0) == 0x4d5a and (
            $a or $b or any of ($c*)
        )
}
```

Rule setup

Example rule

```
import "pe"
rule silent_banker : banker
{
    // c++ comment
    /* c style comment */
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a      = {6A 40 68 00 30 00 [01-04] 6A 14 8D 91}
        $b      = {8D 4D B0 2B C1 ?? C? 27 99 6A 4E 59 F7 F9}
        $c01    = "ABC" fullword wide
        $c02    = /DEF/ ascii nocase
    condition:
        pe.characteristics & pe.EXECUTABLE_IMAGE and uint16be(0) == 0x4d5a and (
            $a or $b or any of ($c*)
        )
}
```

Metadata

Example rule

```
import "pe"
rule silent_banker : banker
{
    // c++ comment
    /* c style comment */
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a      = {6A 40 68 00 30 00 [01-04] 6A 14 8D 91}
        $b      = {8D 4D B0 2B C1 ?? C? 27 99 6A 4E 59 F7 F9}
        $c01    = "ABC" fullword wide
        $c02    = /DEF/ ascii nocase
    condition:
        pe.characteristics & pe.EXECUTABLE_IMAGE and uint16be(0) == 0x4d5a and (
            $a or $b or any of ($c*)
        )
}
```

Strings Definition

Example rule

```
import "pe"
rule silent_banker : banker
{
    // c++ comment
    /* c style comment */
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a      = {6A 40 68 00 30 00 [01-04] 6A 14 8D 91}
        $b      = {8D 4D B0 2B C1 ?? C? 27 99 6A 4E 59 F7 F9}
        $c01    = "ABC" fullword wide
        $c02    = /DEF/ ascii nocase
    condition:
        pe.characteristics & pe.EXECUTABLE_IMAGE and uint16be(0) == 0x4d5a and (
            $a or $b or any of ($c*)
        )
}
```

Conditionals



WARNING

Case study: WannaCry

- Rule location: `/rules/RANSOM_MS17-010_Wannacrypt.yar`
- Malware location: `/malware/
aee20f9188a5c3954623583c6b0e6623ec90d5cd3fdec4e1001646e27664002
c`
- Source: *[https://github.com/Yara-Rules/rules/blob/master/malware/
RANSOM_MS17-010_Wannacrypt.yar](https://github.com/Yara-Rules/rules/blob/master/malware/RANSOM_MS17-010_Wannacrypt.yar)*

Case study: Dridex

- Rule location: `/rules/Maldoc_Dridex.yar`
- Malware location: `/malware/
db788d6d3a8ed1a6dc9626852587f475e7671e12fa9c9faa73b7277886f1e21
0`
- Source: *[https://github.com/Yara-Rules/rules/blob/master/maldocs/
Maldoc_Dridex.yar](https://github.com/Yara-Rules/rules/blob/master/maldocs/Maldoc_Dridex.yar)*



HANDS-ON

Q&A

