



南开大学

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月异

Yara病毒检测引擎

王志

zwang@nankai.edu.cn

南开大学 网络空间安全学院



允公允能 日新月异

2021年国家网络安全宣传周

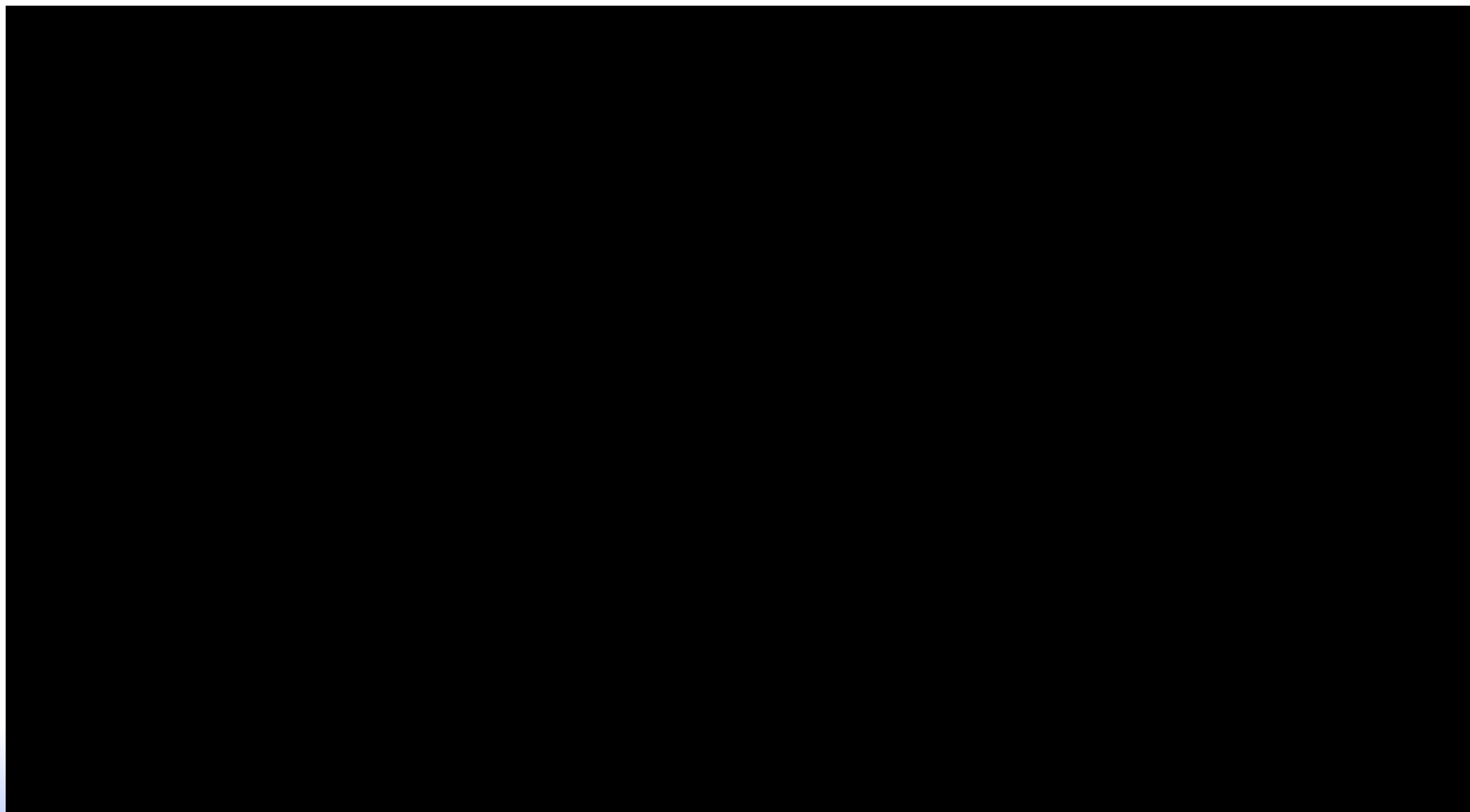


“网络安全为人民，
网络安全靠人民”
为主题的2021年国家网络安全宣传周
将于10月11日至17
日在全国范围内展
开。



允公允能 日新月异

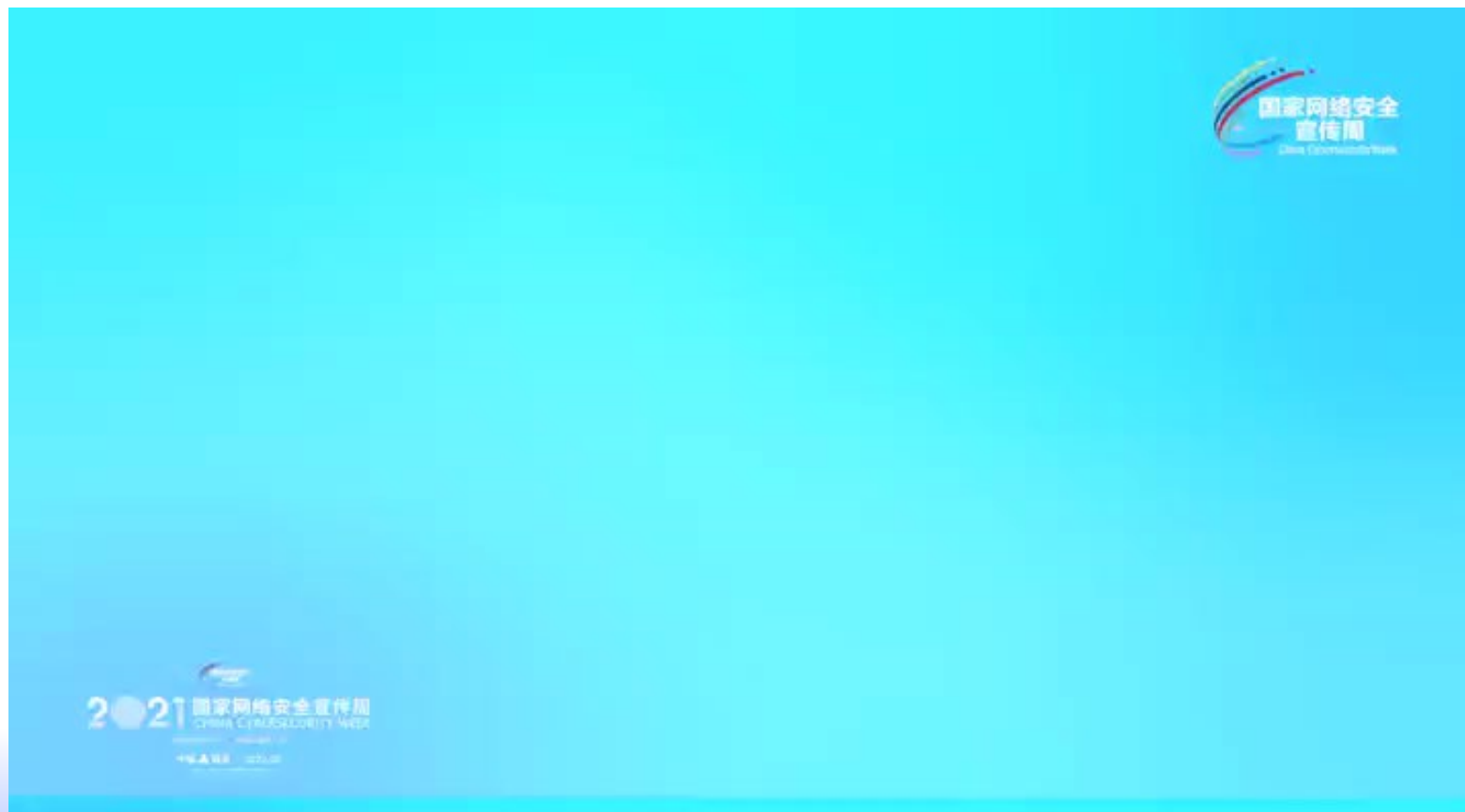
2021国家网络安全宣传周





允公允能 日新月异

2021国家网络安全宣传周





允公允能 日新月异

2021年国家网络安全宣传周





允公允能 日新月异

网络空间安全

没有网络安全
就没有国家安全，
就没有经济社会稳定运行，广大人民群众利益也难以得到保障。

——2018年4月20日，在全国网络安全和信息化工作会议上的讲话



我多次说过，
没有网络安全就没有国家安全；过不了互联网这一关，就过不了长期执政这一关。

——2019年1月25日，在十九届中央政治局第十二次集体学习时的讲话





允公允能 日新月异

本章知识点

- Yara引擎
- Yara引擎安装
- Yara规则
- Yara字符串
- Yara条件表达式



南開大學

NANKAI UNIVERSITY, P.R.CHINA 1919

允公允能 日新月異

1. Yara引擎



允公允能 日新月异

Yara引擎

- Yara 是VirusTotal发布的一个开源恶意代码查杀引擎
 - 识别恶意代码
 - 分类恶意代码
- Yara 引擎是跨平台的，可在 Windows、Linux 和 Mac OS X 上运行。



允公允能 日新月异

Yara引擎

- Yara 本身不提供杀毒功能
 - 没有特征库
 - 需要编写Yara规则，以此来识别和分类恶意软件或者程序。



允公允能 日新月异

Yara规则

- Yara规则是由一系列字符串和一个布尔型表达式构成
- 支持与或非等多种条件



```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"

    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 or $c
}
```


此题未设置答案，请点击右侧设置按钮

以下对Yara引擎的描述哪些是正确的？

- ☐ A 可以跨平台
- ☐ B 可以用来识别和分类恶意代码
- ☐ C Yara引擎本身不提供杀毒功能
- ☐ D Yara规则由1列字符串和1个布尔表达式构成

提交



南開大學

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月異

2. Yara引擎的安裝



允公允能 日新月异

Yara引擎

- Yara引擎的github地址：
 - <https://github.com/VirusTotal/yara/releases>
- Yara引擎文档说明：
 - <https://yara.readthedocs.io/en/stable>



允公允能 日新月异

Windows安装Yara引擎

VirusTotal / yara Public

Notifications

Star

5k

Fork

1.1k

Code Issues 128 Pull requests 30 Actions Projects Wiki Security Insights

Releases

Tags

Latest release

v4.1.2

f844881

Compare

YARA v4.1.2

plusvic released this on 23 Aug

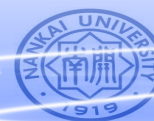
BUGFIX: `TOO_MANY_MATCHES` warning was causing strings to be globally disabled (#1532).

BUGFIX: `fullworld` modifier not working as expected in Mac OS due to locale issue (#1544, [VirusTotal/yara-python#184](#)).

BUGFIX: Default value for `pe.number_of_imported_function` not set to 0 (#1546).

4

4 people reacted



南开大学





Nankai University



允公允能 日新月异

Yara引擎安装包

▼ Assets 4



 yara-v4.1.2-1693-win32.zip	1.35 MB
 yara-v4.1.2-1693-win64.zip	1.98 MB
 Source code (zip)	
 Source code (tar.gz)	



Yara 引擎

📁 > yara-v4.1.2-1693-win64

🔍 搜索"yara-v4.1.2-1693-win64"

名称	修改日期	类型	大小
 yara64.exe	2021/8/23 19:37	应用程序	2,142 KB
 yarac64.exe	2021/8/23 19:37	应用程序	2,096 KB



Windows安装Yara引擎

```
C:\Users\nkamg\Desktop\yara-v4.1.2-1693-win64>yara64.exe
yara: wrong number of arguments
Usage: yara [OPTION]... [NAMESPACE:]RULES_FILE... FILE | DIR | PID

Try `--help` for more options

C:\Users\nkamg\Desktop\yara-v4.1.2-1693-win64>yara64.exe --help
YARA 4.1.0, the pattern matching swiss army knife.
Usage: yara [OPTION]... [NAMESPACE:]RULES_FILE... FILE | DIR | PID

Mandatory arguments to long options are mandatory for short options too.

      --atom-quality-table=FILE      path to a file with the atom quality table
-C,  --compiled-rules               load compiled rules
-c,  --count                         print only number of matches
-d,  --define=VAR=VALUE             define external variable
      --fail-on-warnings             fail on warnings
-f,  --fast-scan                     fast matching mode
-h,  --help                         show this help and exit
-i,  --identifier=IDENTIFIER        print only rules named IDENTIFIER
```



允公允能 日新月异

安装yara-python

- 直接编写python程序来调用Yara引擎
- pip自动安装
 - `pip install yara-python`



允公允能 日新月异

运行Yara引擎

```
C:\Users\nkamg\Desktop\yara-v4.1.2-1693-win64>echo rule dummy { condition: true } > my_first_rule
```

```
C:\Users\nkamg\Desktop\yara-v4.1.2-1693-win64>yara64 my_first_rule my_first_rule  
dummy my_first_rule
```



南开大学

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月异

3. Yara规则



Yara规则

规则开始的关键字



rule dummy



规则标识符Identifier，第一个字符不能是数字，长度不超过128字符，区分大小写

{

condition:

false

}

Yara规则类似于C语言，每个规则都以关键字“rule”开头





Yara规则中的关键字

all	and	any	ascii	at	base64	base64wide	condition
contains	entrypoint	false	filesize	for	fullword	global	import
in	include	int16	int16be	int32	int32be	int8	int8be
matches	meta	nocase	not	of	or	private	rule
strings	them	true	uint16	uint16be	uint32	uint32be	uint8
uint8be	wide	xor					



rule **silent_banker** : banker tag字段

{ 规则名

meta: 描述信息

description = "This is just an example"

strings: 规则字段

\$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 or \$c 条件判断字段

}



允公允能 日新月异

注释

- 可以像编写C语言一样，在Yara规则中添加注释：
 - `//` 单行注释
 - `/*` 多行注释 `*/`



南開大學

NANKAI UNIVERSITY, P.R.CHINA 1919

允公允能 日新月異

4. Yara字符串



字符串

- Yara中有三种类型的字符串：

- 十六进制串：定义原始字节序列

`$a = {6A 40 68 00 30 00 00 6A 14 8D 91}`

- 文本字符串：定义可读文本的部分

`"UVODFRYSIHLNWPEJXQZAKCBGMT"`

- 正则表达式：定义可读文本的部分



通配符

```
rule WildcardExample
```

```
{
```

```
strings: //使用 ‘?’ 作为通配符
```

```
    $hex_string = { 00 11 ?? 33 4? 55 }
```

```
condition:
```

```
    $hex_string
```

```
}
```



跳转

```
rule JumpExample
{
strings:
//使用 ‘[]’ 作为跳转，与任何长度为0-2字节的内容匹配
    $hex_string1 = { 00 11 [2] 44 55 }
    $hex_string2 = { 00 11 [0-2] 44 55 }
    //该写法与string1作用完全相同
    $hex_string3 = { 00 11 ?? ?? 44 55 }
condition:
    $hex_string1 and $hex_string2
}
```




正则表达式

```
rule AlternativesExample1
```

```
{
```

```
strings:
```

```
    $hex_string = { 00 11 ( 22 | 33 44 ) 55 }
```

```
/* 匹配 00 11 22 55 或者 00 11 33 44 55 */
```

```
condition:
```

```
    $hex_string
```

```
}
```



正则表达式

```
rule AlternativesExample2
```

```
{
```

```
strings:
```

```
    $hex_string = { 00 11 ( 33 44 | 55 | 66 ?? 88 ) 99 }
```

```
condition:
```

```
    $hex_string
```

```
}
```

\$hex_string = { 00 11 (33 44 | 55 | 66 ?? 88) 99 }

该规则可以匹配以下哪几个十六进制串

☐ A

00 11 33 44

☒ B

00 11 55 99

☐ C

00 11 66 77 88

☒ D

00 11 66 56 88 99

提交



允公允能 日新月异

文本字符串-转义符

- \ " 双引号
- \\ 反斜杠
- \t 制表符
- \n 换行符
- \xdd 十六进制的任何字节



修饰符

- nocase: 不区分大小写
- wide: 匹配2字节的宽字符
- ascii: 匹配1字节的ascii字符
- xor: 匹配异或后的字符串
- fullword: 匹配完整单词
- private: 定义私有字符串



文本字符串

- 不区分大小写

```
$text_string = "foobar" nocase
```

- 匹配宽字符串

```
$wide_string = "Borland" wide
```

- 同时匹配2种类型的字符串

```
$wide_and_ascii_string = "Borland" wide ascii
```




文本字符串

- 匹配所有可能的异或后字符串

`$xor_string = "This program cannot" xor`

- 匹配所有可能的异或后wide和ascii字符串

`$xor_string = "This program cannot" xor wide ascii`

- 限定异或范围

`$xor_string = "This program cannot" xor (0x01-0xff)`



文本字符串

- 全词匹配

`$wide_string = "domain" fullword`

匹配: `www.domain.com`, `www.my-domain.com`

不匹配: `www.mydomain.com`

- 私有字符串: 正常匹配规则, 不会在输出中显示

`$text_string = "foobar" private`

\$wide_string = "nankai" fullword, 以下哪些字符串会匹配
\$wide_string

- ☒ A www.nankai.edu.cn
- ☐ B ilovenankai
- ☒ C i-nankai
- ☒ D nankai university



南开大学

NANKAI UNIVERSITY, P.R. CHINA 1919

允公允能 日新月异

5. Yara条件表达式



正则表达式-元字符（metacharacters）

\	Quote the next metacharacter
^	Match the beginning of the file
\$	Match the end of the file
	Alternation
()	Grouping
[]	Bracketed character class



数量匹配

*	Match 0 or more times
+	Match 1 or more times
?	Match 0 or 1 times
{n}	Match exactly n times
{n,}	Match at least n times
{,m}	Match at most m times
{n,m}	Match n to m times



字符类型定义

<code>\w</code>	Match a <i>word</i> character (alphanumeric plus “_”)
<code>\W</code>	Match a <i>non-word</i> character
<code>\s</code>	Match a <i>whitespace</i> character
<code>\S</code>	Match a <i>non-whitespace</i> character
<code>\d</code>	Match a <i>decimal digit</i> character
<code>\D</code>	Match a <i>non-digit</i> character



条件表达式

- 布尔表达式
- 布尔操作符: and、or、not
- 关系操作符: \geq 、 \leq 、 $<$ 、 \geq 、 $!=$
- 位操作符: $\&$ 、 $|$ 、 \ll 、 \gg 、 \sim 、 \wedge

$\$a = \text{"text1"}$, $\$b = \text{"text2"}$, $\$c = \text{"text3"}$, $\$d = \text{"text4"}$
condition:

$(\$a \text{ or } \$b) \text{ and } (\$c \text{ or } \$d)$

下面哪个选项可以被规则匹配上？

☐ A text1 text2

☒ B text1 text3

☒ C text1 text4

☐ D text3 text4

提交



Counting strings

strings:

```
$a = "dummy1"
```

```
$b = "dummy2"
```

condition:

//a字符串出现6次，b字符串大于10次

```
#a == 6 and #b > 10
```



@ 获取字符串出现位置

- 可以使用@a[i]，获取字符串\$a在文件或内存中，第*i*次出现的偏移或虚拟地址。
- 索引从1开始，不是从0开始。如果*i*大于字符串出现的次数，结果为NaN(not a number 非数值)。



! 获取字符串长度

- 可以使用!`a[i]`，获取字符串\$a在文件或内存中，第*i*次出现时的字符串长度。
- 索引同@一样都是从1开始，不是从0开始。
- !a 是 !a[1]的简写。



at 指定字符串匹配的位置

- at 匹配字符串在文件或内存中的偏移

strings:

```
$a = "dummy1"
```

```
$b = "dummy2"
```

condition: //a和b字符串出现在文件或内存的100和200偏移处

```
$a at 100 and $b at 200
```



in 指定字符串匹配的范围

- **in** 在文件或内存的某个地址范围内匹配字符串

strings:

```
$a = "dummy1"
```

```
$b = "dummy2"
```

condition:

```
$a in (0..100) and $b in (100..filesize)
```





filesize 文件大小匹配

- filesize 匹配文件大小

condition:

//filesize 只在文件时才有用，对进程无效

//KB MB 后缀只能与十进制大小一起使用

filesize > 200KB



entrypoint 入口点匹配

- 匹配PE或ELF文件入口点(高版本使用PE模块的
pe.entry_point代替)

strings:

```
$a = { E8 00 00 00 00 }
```

condition:

```
$a at entrypoint
```



entrypoint

strings:

```
$a = { 9C 50 66 A1 ?? ?? ?? 00 66 A9 ?? ?? 58  
0F 85 }
```

condition:

```
$a in (entrypoint..entrypoint + 10)
```



读文件或内存数据

- `intxxx` 读取 **小端** 有符号整数
- `int8(<offset or virtual address>)`
- `int16(<offset or virtual address>)`
- `int32(<offset or virtual address>)`



读文件或内存数据

- `uintxxx` 读取 **小端** 无符号整数
- `uint8(<offset or virtual address>)`
- `uint16(<offset or virtual address>)`
- `uint32(<offset or virtual address>)`



读文件或内存数据

- `intxxxbe` 读取大端有符号整数
- `int8be(<offset or virtual address>)`
- `int16be(<offset or virtual address>)`
- `int32be(<offset or virtual address>)`



允公允能 日新月异

读内存或文件数据

- `uintxxxbe` 读取大端无符号整数
- `uint8be(<offset or virtual address>)`
- `uint16be(<offset or virtual address>)`
- `uint32be(<offset or virtual address>)`

编写判断文件是否是PE文件的Yara规则。

IMAGE_DOS_SIGNATURE = 0x5A4D

e_lfanew 的偏移地址是0x3C

IMAGE_NT_SIGNATURE = 0x00004550

正常使用主观题需2.0以上版本雨课堂

作答



IsPE

```
rule IsPE
```

```
{
```

```
condition:
```

```
//判断是否PE文件
```

```
uint16(0) == 0x5A4D and // “MZ” 头
```

```
uint32(uint32(0x3C)) == 0x00004550 // “PE” 头
```

```
}
```



of 匹配部分字符串

- 匹配多个字符串中的某几个

strings:

`$a = "dummy1"`

`$b = "dummy2"`

`$c = "dummy3"`

condition: //3个字符串只需匹配任意2个

`2 of ($a, $b, $c)`



for 多字符串匹配

- for AAA of BBB : (CCC)
- 在BBB字符串集合中，至少有AAA个字符串，满足了CCC的条件表达式，才算匹配成功。

for 1 of (\$a, \$b, \$c) : (# > 3)

//至少1个字符串在文件或内存中出现的次数大于3



any、all、them 多字符串匹配

- 在条件表达式中，可以使用\$依次代替字符串集合中的每一个字符串，#表示字符串的出现次数
- for 1 of (\$a, \$b, \$c) : (\$ at entrypoint)
- for any of (\$a, \$b, \$c) : (\$ at entrypoint)
- for all of them : (# > 3)

$\$a = \text{"55 8B EC"}$

for all of $(\$a^*) : (@[2] < 0x4000000)$, 写出该表达式什么情况下匹配成功?

正常使用主观题需2.0以上版本雨课堂

作答



for-in 多字符串匹配

- `for AAA BBB in (CCC) : (DDD)`

- 作用与for of类似，增加了下标变量与下标范围

```
for all i in (1, 2, 3) : ( @a[i] + 10 == @b[i] )
```

\$a每次在文件或内存中出现位置，都必须都小于100。
(使用for-in表达方式来描述)

正常使用主观题需2.0以上版本雨课堂

作答



允公允能 日新月异

引用其它规则

strings:

$\$a = \text{"dummy2"}$

condition:

$\$a$ and **IsPE**



全局规则

- 全局规则（global rule）可以在匹配其他规则前优先筛选

global rule SizeLimit

{

condition:

filesize < 2MB

}

比如在匹配目标文件之前需要先筛选出小于2MB的文件，再匹配其它规则。





私有规则

- 私有规则（private rule）避免规则匹配结果的混乱，YARA不会输出任何匹配到的私有规则信息。

```
private rule PrivateRuleExample  
{  
...  
}
```



导入模块

- import 导入模块，可以使用第三方模块导出的变量或函数

```
import "pe"
```

```
import "cuckoo"
```

```
pe.entry_point == 0x1000
```

```
cuckoo.http_request(/someregexp/)
```



允公允能 日新月异

外部变量

- 外部变量允许使用YARA -d命令时指定一个自定义数据
- 该数据可以是整数、字符串、布尔变量



文件包含

- include包含其它规则文件的内容到当前文件中
- 相对路径
 - include `"/includes/other.yar"`
 - include `"/../includes/other.yar"`
- 全路径
 - include `"/home/plusvic/yara/includes/other.yar"`





实验课：参与学堂在线讨论

- 对Lab1和Lab3的样本编写Yara规则
- 使用自己编写的规则对自己电脑的C盘进行Yara引擎的扫描，记录扫描所用时间。
- **学堂在线提交报告**：讨论哪些Yara条件执行效率高，哪些Yara条件执行效率低。如何改进那些执行效率低的Yara条件？