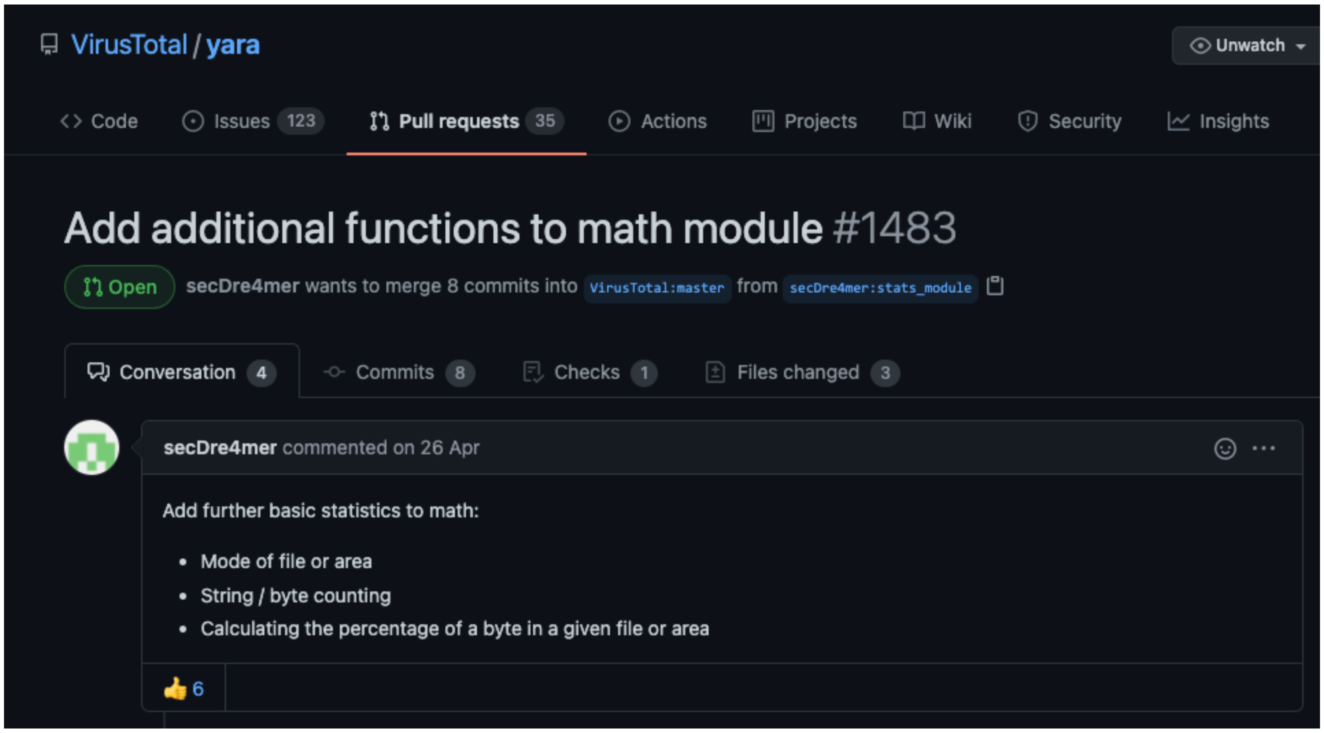
Use YARA math Module Extension in THOR TechPreview and THOR Lite

nextron-systems.com/2021/06/15/use-yara-math-module-extension-in-thor-techpreview-and-thor-lite/

June 15, 2021

Not long ago, we've created a pull request for the official YARA repository on Github, that would introduce new functions in the `math` module to improve the flexibility in cases in which a sample is heavily scrambled or obfuscated. These cases require further statistical evaluations that go beyond the currently available "entropy", "mean" or "deviation" functions.



Add additional functions to math module #1483

Open secDre4mer wants to merge 8 commits into VirusTotal:master from secDre4mer:stats_module

secDre4mer commented on 26 Apr

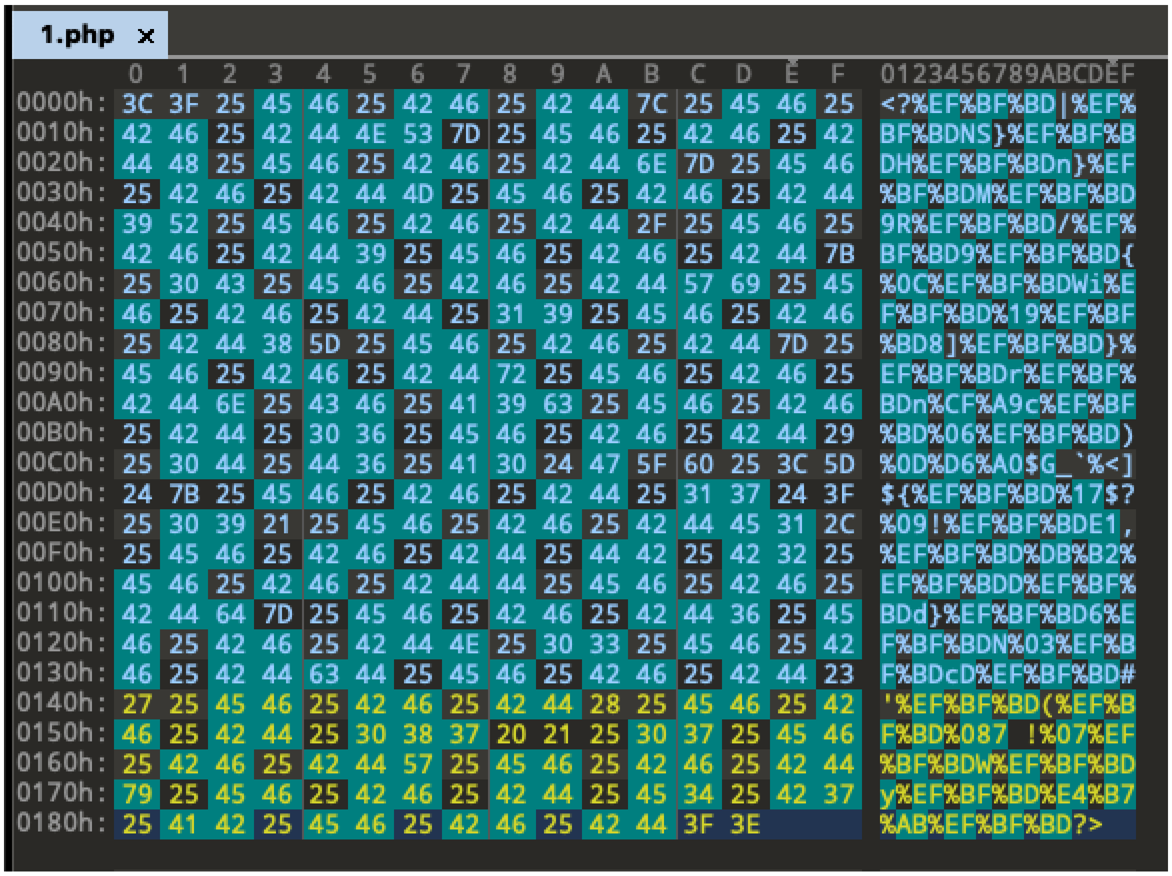
Add further basic statistics to math:

- Mode of file or area
- String / byte counting
- Calculating the percentage of a byte in a given file or area

👍 6

The example on the right shows a heavily obfuscated PHP web shell, as used by a Chinese actor.

You immediately notice the high amount of "%" characters, but since each of them is preceded and followed by different characters, it's difficult to find atoms that are long enough to maintain an acceptable performance / stability of that rule.



```
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 3C 3F 25 45 46 25 42 46 25 42 44 7C 25 45 46 25  <?%EF%BF%BD|%EF%
0010h: 42 46 25 42 44 4E 53 7D 25 45 46 25 42 46 25 42  BF%BDNS}%EF%BF%B
0020h: 44 48 25 45 46 25 42 46 25 42 44 6E 7D 25 45 46  DH%EF%BF%BDn}%EF
0030h: 25 42 46 25 42 44 4D 25 45 46 25 42 46 25 42 44  %BF%BDM%EF%BF%BD
0040h: 39 52 25 45 46 25 42 46 25 42 44 2F 25 45 46 25  9R%EF%BF%BD/%EF%
0050h: 42 46 25 42 44 39 25 45 46 25 42 46 25 42 44 7B  BF%BD9%EF%BF%BD{
0060h: 25 30 43 25 45 46 25 42 46 25 42 44 57 69 25 45  %0C%EF%BF%BDWi%E
0070h: 46 25 42 46 25 42 44 25 31 39 25 45 46 25 42 46  F%BF%BD%19%EF%BF
0080h: 25 42 44 38 5D 25 45 46 25 42 46 25 42 44 7D 25  %BD8]%EF%BF%BD}%
0090h: 45 46 25 42 46 25 42 44 72 25 45 46 25 42 46 25  EF%BF%BDr%EF%BF%
00A0h: 42 44 6E 25 43 46 25 41 39 63 25 45 46 25 42 46  BDn%CF%A9c%EF%BF
00B0h: 25 42 44 25 30 36 25 45 46 25 42 46 25 42 44 29  %BD%06%EF%BF%BD)
00C0h: 25 30 44 25 44 36 25 41 30 24 47 5F 60 25 3C 5D  %0D%D6%A0$G_`%<]
00D0h: 24 7B 25 45 46 25 42 46 25 42 44 25 31 37 24 3F  ${%EF%BF%BD%17$?
00E0h: 25 30 39 21 25 45 46 25 42 46 25 42 44 45 31 2C  %09!%EF%BF%BDE1,
00F0h: 25 45 46 25 42 46 25 42 44 25 44 42 25 42 32 25  %EF%BF%BD%DB%B2%
0100h: 45 46 25 42 46 25 42 44 44 25 45 46 25 42 46 25  EF%BF%BDD%EF%BF%
0110h: 42 44 64 7D 25 45 46 25 42 46 25 42 44 36 25 45  BDd}%EF%BF%BD6%E
0120h: 46 25 42 46 25 42 44 4E 25 30 33 25 45 46 25 42  F%BF%BDN%03%EF%B
0130h: 46 25 42 44 63 44 25 45 46 25 42 46 25 42 44 23  F%BDcD%EF%BF%BD#
0140h: 27 25 45 46 25 42 46 25 42 44 28 25 45 46 25 42  '%EF%BF%BD(%EF%B
0150h: 46 25 42 44 25 30 38 37 20 21 25 30 37 25 45 46  F%BD%087 !%07%EF
0160h: 25 42 46 25 42 44 57 25 45 46 25 42 46 25 42 44  %BF%BDW%EF%BF%BD
0170h: 79 25 45 46 25 42 46 25 42 44 25 45 34 25 42 37  y%EF%BF%BD%E4%B7
0180h: 25 41 42 25 45 46 25 42 46 25 42 44 3F 3E        %AB%EF%BF%BD?>
```

If you could, you would formulate a rule like this: “Detect files smaller 400 bytes, that begin with ‘<?’ and consist of at least 25 percent ‘%’ characters”.

Well, the new module extension allows you to do exactly that.

```
import "math"

rule SUSP_PHP_Percent_OBFUSC_Indicators_Jun21 {
   meta:
      description = "Detects PHP files with a high percentage of % characters"
      author = "Florian Roth"
      reference = "Internal Research"
      date = "2021-06-15"
      score = 70
   condition:
      filesize < 400
      and uint16(0) == 0x3f3c
      and math.percentage(0x25) >= 0.25
}
```

Read the documentation provided with the pull request for details on all three new functions:

- count(byte/string, offset, size)
- percentage(byte, offset, size)
- mode(offset, size)

While the first two functions are self-explanatory, the “mode” function isn’t. It is is a term used in statistics for the most common value.

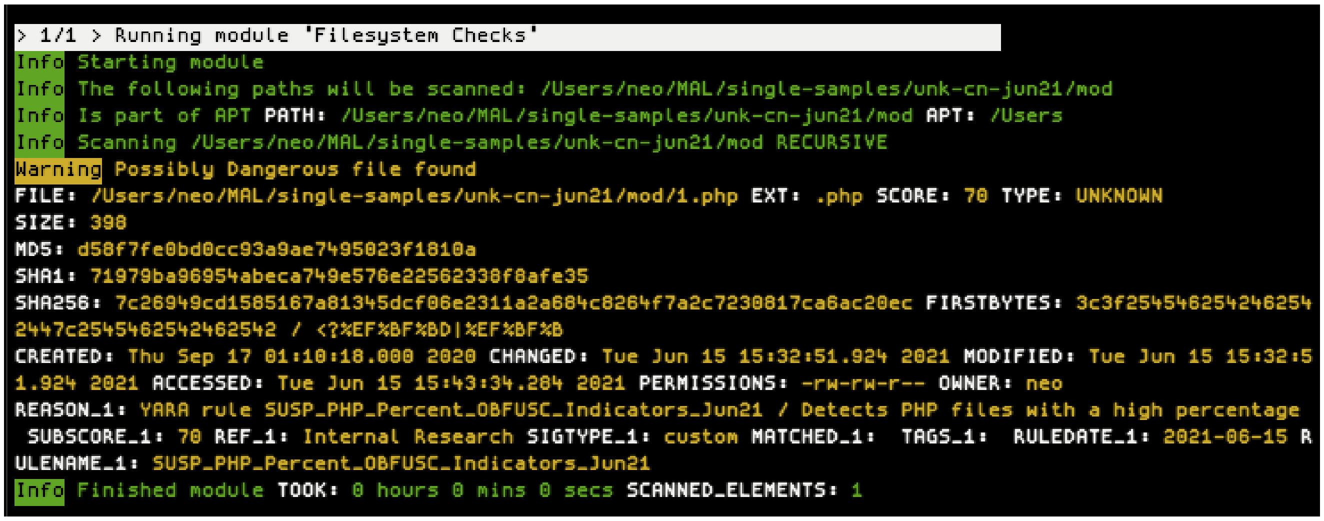
```
.. c:function:: count(byte/string, offset, size)

    .. versionadded:: 4.2.0

    Returns how often a specific byte or substring occurs, starting at *offset*
    and looking at the next *size* bytes. When scanning a
    running process the *offset* argument should be a virtual address within
    the process address space.
    *offset* and *size* are optional; if left empty, the complete file is searched.

    *Example: math.count("$[]", 0, 100) >= 5*

    *Example: math.count(0x4A) >= 10*
```

```
.. c:function:: percentage(byte, offset, size)

    .. versionadded:: 4.2.0

    Returns the occurrence rate of a specific byte, starting at *offset*
    and looking at the next *size* bytes. When scanning a
    running process the *offset* argument should be a virtual address within
    the process address space. The returned value is a float between 0 and 1.
    *offset* and *size* are optional; if left empty, the complete file is searched.


    *Example: math.percentage(0xFF, filesize-1024, filesize) >= 0.9*

    *Example: math.percentage(0x4A) >= 0.4*
```

```
.. c:function:: mode(offset, size)

    .. versionadded:: 4.2.0

    Returns the most common byte, starting at *offset* and looking at the next
    *size* bytes. When scanning a
    running process the *offset* argument should be a virtual address within
    the process address space. The returned value is a float.
    *offset* and *size* are optional; if left empty, the complete file is searched.

    *Example: math.mode(0, filesize) == 0xFF*
```

For your convenience, we’ve already patched our versions of THOR TechPreview and THOR Lite to support these extensions of the “math” module. You need at least v10.6.6 to use the new function in your rules.

We wish you good hunting.



```
> 1/1 > Running module 'Filesystem Checks'
Info Starting module
Info The following paths will be scanned: /Users/neo/MAL/single-samples/unk-cn-jun21/mod
Info Is part of APT PATH: /Users/neo/MAL/single-samples/unk-cn-jun21/mod APT: /Users
Info Scanning /Users/neo/MAL/single-samples/unk-cn-jun21/mod RECURSIVE
Warning Possibly Dangerous file found
FILE: /Users/neo/MAL/single-samples/unk-cn-jun21/mod/1.php EXT: .php SCORE: 70 TYPE: UNKNOWN
SIZE: 398
MD5: d58f7fe0bd0cc93a9ae7495023f1810a
SHA1: 71979ba96954abeca749e576e22562338f8afe35
SHA256: 7c26949cd1585167a81345dcf06e2311a2a684c8264f7a2c7230817ca6ac20ec FIRSTBYTES: 3c3f2545462542462544
2447c25454625424625 42 / <?%EF%BF%BD|%EF%BF%B
CREATED: Thu Sep 17 01:10:18.000 2020 CHANGED: Tue Jun 15 15:32:51.924 2021 MODIFIED: Tue Jun 15 15:32:5
1.924 2021 ACCESSED: Tue Jun 15 15:43:34.284 2021 PERMISSIONS: -rw-rw-r-- OWNER: neo
REASON_1: YARA rule SUSP_PHP_Percent_OBFUSC_Indicators_Jun21 / Detects PHP files with a high percentage
 SUBSCORE_1: 70 REF_1: Internal Research SIGTYPE_1: custom MATCHED_1:  TAGS_1:  RULEDATE_1: 2021-06-15 R
ULENAME_1: SUSP_PHP_Percent_OBFUSC_Indicators_Jun21
Info Finished module TOOK: 0 hours 0 mins 0 secs SCANNED_ELEMENTS: 1
```