

YARA Report and Demonstration

1. Introduction to YARA

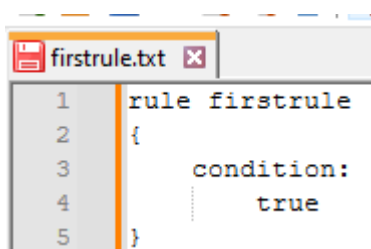
YARA is “the pattern matching swiss knife for malware researchers (and everyone else)” as stated on the YARA website/documentation. Simply put, YARA is a tool primarily used in malware research that allows researchers to create descriptions of malware families based on textual or binary patterns. These descriptions, known as YARA rules, help identify and classify malware by specifying patterns to look for in files. YARA rules are often integrated into various tools and platforms to automatically scan file to identify potential malware.

2. Important Note

Before we start creating some basic YARA rule, it is important to note that if you follow along, we are dealing with live malware that can damage your host system. Therefore, you should take basic precautions such as using a virtual machine (VM) like FLARE VM, disconnected from your local network or a sandboxed environment.

3. Creating a Basic YARA Rule

Let's start by creating a very simply YARA rule that always returns true, meaning if you use this rule against any file, it will always match said file. To start off, open a text editor, such as Notepad++, and enter the following text seen in the screenshot below:

A screenshot of a text editor window titled 'firstrule.txt'. The window contains a YARA rule definition. The text is as follows:

```
1 rule firstrule
2 {
3     condition:
4         true
5 }
```

The text after 'rule' denotes the name of the rule, and 'condition: true' means that this will match any file it is checked against. Save this file in the same directory as your downloaded YARA binary, and open a terminal or command prompt in that directory. To run the rule, enter:

```
C:\Users\vboxuser\Desktop\Tools\Forensic
λ yara firstrule.txt .
firstrule .\Autopsy.lnk
firstrule .\firstrule.txt
firstrule .\yara.lnk
firstrule .\yarak.lnk
```

The output indicates that this rule matched all files in the current directory.

4. Practical Example: Detecting WannaCry

Let's create a more practical YARA rule to detect WannaCry, an infamous piece of ransomware from 2017. You can find WannaCry samples on several malware sharing platforms such as vxunderground. First off, I am going to use floss, which is a tool created by Mandiant that extracts strings from malware. The purpose being that we want to identify interesting strings in the WannaCry sample:

```
C:\Users\vboxuser\Desktop
λ floss Ransomware.wannacry.exe.malz
```

Or:

```
C:\Users\vboxuser\Desktop
λ floss -n 8 Ransomware.wannacry.exe.malz > wannacry_floss.txt
```

The syntax is simply floss followed by the malware binary. The output is very long, making it likely beneficial to use the second command example to only extract strings of a larger length. The output indicates several strings unique to this ransomware which are interesting, these include (but not limited to):

- WanaCrypt0r
- <http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com> (kill switch for the malware)
- C:\%s\qeriuwjhrf
- cmd.exe /c "%s"
- WNcry@2017

Using some of these strings, we can write a YARA rule to identify WannaCry:

```
wannacry.txt x
1 rule WannaCry : ransomware
2 {
3     meta:
4         author = "Me"
5         description = "This basic YARA rule can be used to identify WannaCry ransomware"
6
7     strings:
8         $killswitch = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"
9         $wannacry1 = "WanaCrypt0r"
10        $wannacry2 = "WNCry@2017"
11
12    condition:
13        1 of them
14 }
```

This rule is more detailed than the previous one, it includes the following sections:

- **Meta:** Contains metadata about the rule, such as the author and description.
- **Strings:** Lists the strings that are unique to WannaCry.
- **Condition:** Specifies that the rule will match a file if it contains 1 or more of the three strings.

To test this rule out, save the rule file and run:

```
C:\Users\vboxuser\Desktop\Tools\Forensic
λ yara wannacry.txt C:\Users\vboxuser\Desktop\Ransomware.wannacry.exe.malz -s
WannaCry C:\Users\vboxuser\Desktop\Ransomware.wannacry.exe.malz
0x313d0:$killswitch: http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x415d0:$wannacry2: WNCry@2017
```

As you can see, we have appended the -s switch which tells us that strings the rule has matched against.

5. “Advanced Example”: Checking for PE Files

It is important to understand that you can include multiple rules in one file. To demonstrate this, let's create a rule to check if a file is a portable executable (PE) along with if it contains strings found in the WannaCry sample:

```

rule IsPE
{
    condition:
        // MZ signature at offset 0 and ...
        uint16(0) == 0x5A4D and
        // ... PE signature at offset stored in MZ header at 0x3C
        uint32(uint32(0x3C)) == 0x00004550
}

rule WannaCry : ransomware
{
    meta:
        author = "Me"
        description = "This basic YARA rule can be used to identify WannaCry ransomware"

    strings:
        $skillswitch = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"
        $wannacry1 = "WanaCrypt0r"
        $wannacry2 = "WNCry@2017"

    condition:
        1 of them
}

```

The 'IsPe' rule checks if the PE signature found at offset '0x3C' matches the MZ signature (0x5A4D which is hex for MZ). Run the rule file against the WannaCry sample to see if it matches both:

```

C:\Users\vboxuser\Desktop\Tools\Forensic
λ yara wannacry.txt C:\Users\vboxuser\Desktop\Ransomware.wannacry.exe.malz -s
IsPE C:\Users\vboxuser\Desktop\Ransomware.wannacry.exe.malz
WannaCry C:\Users\vboxuser\Desktop\Ransomware.wannacry.exe.malz
0x313d0:$skillswitch: http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x415d0:$wannacry2: WNCry@2017

```

The output shows us that it matched against the IsPe and WannaCry rule, indicating that it is a portable executable and WannaCry ransomware. The IsPe rule is beneficial for 99% of malware samples, as the majority of malware is PE. The IsPe rule can also help prevent false positives. To explain this, take for example a report on WannaCry, it would likely contain the strings we have used in the rule. If the IsPe rule/check was not used, it would match the file, even though it is not actually malware.

6. Conclusion

This short report demonstrates YARA in a simple and understandable way. I created this to explain and demonstrate YARA in a manner which would have greatly benefited me weeks ago. While these examples provide a basic introduction to YARA, real-world YARA rules are typically much more complex and require a deep understanding of the patterns you want to

search for. To gain a comprehensive understanding of YARA, please check out the documentation found [here](#).

YARA rules are easy to pick up but challenging to master. The effectiveness of your rules depends on your understanding of the malware patterns you are targeting.