

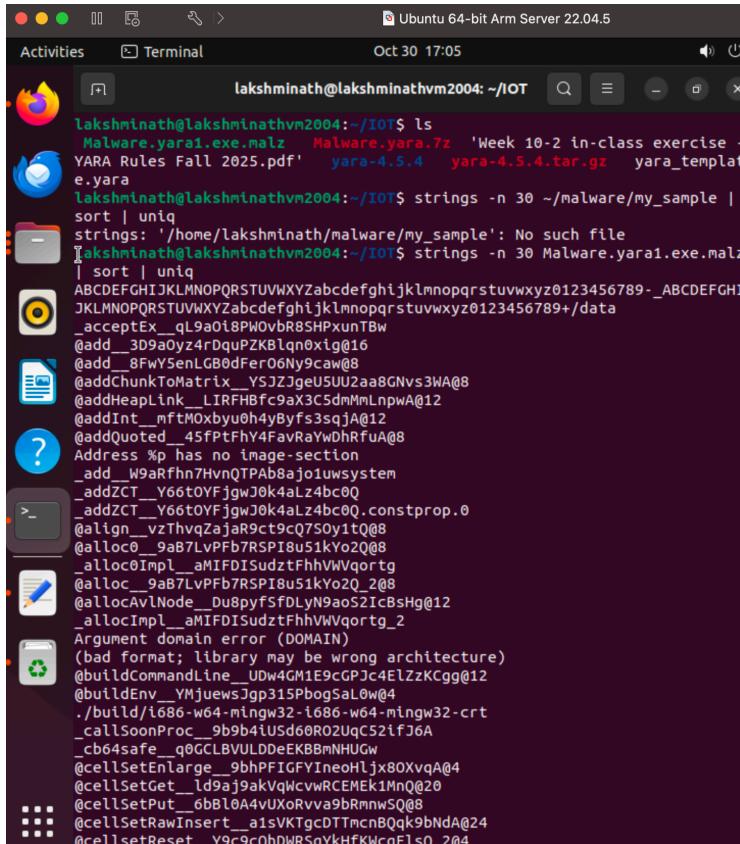
IOT Inclass Exercise Week10_2 YARA

Lakshminath Reddy Alamuru

SUID: 367982169

Task-1

Extracted the strings from malware sample file given in the task.
I showed the strings which are at least 30 characters.



```
lakshminath@lakshminathvm2004:~/IOT$ ls
Malware.yara1.exe.malz  Malware.yara.7z  'Week 10-2 in-class exercise -
YARA Rules Fall 2025.pdf'  yara-4.5.4  yara-4.5.4.tar.gz  yara_template.yara
lakshminath@lakshminathvm2004:~/IOT$ strings -n 30 ~/malware/my_sample |
sort | uniq
strings: '/home/lakshminath/malware/my_sample': No such file
lakshminath@lakshminathvm2004:~/IOT$ strings -n 30 Malware.yara1.exe.malz |
sort | uniq
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-_ABCDEFH
JKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+_data
_acceptedEx_qL9ao18Pwv0r8SHPxunTBw
_addr__3D9a0yz4rDquPZKBlqn0ig@16
_addr__8FwY5enLG80dFer06Ny9caw@8
@addChunkToMatrix__YSJZjge5UU2aa8GNvs3WA@8
@addHeapLink__LIRFHBFc9aXc35dmMnLnpwA@12
@addInt__mftMOxbyu0h4yByfs3sqjA@12
@addQuoted__45fPtFhy4FavRaywDhRfuA@8
Address %p has no image-section
_addr_W9aRfhn7HvnqTPAb8ajoiuwsystem
_addrZCT_Y66t0YFjgwJ0k4aLz4bc0Q.constprop.0
@align_vzThvqzajar9ctc9Q7Soy1t@0@8
@alloc0_9aB7LvPFb7RSPI8u51kYo2Q@8
@alloc0Impl_aMFDISudztFhhVVqortg
@alloc_9aB7LvPFb7RSPI8u51kYo2Q_@8
@allocAvlNode__Du8pyffsf0LyN9aoS2Ic8sHg@12
@allocImpl_aMFDISudztFhhVVqortg._2
Argument domain error (DOMAIN)
(bad format; library may be wrong architecture)
@buildCommandLine__UDW4GM1E9cGPjC4ElZzKCgg@12
@buildEnv__YMjuewsJgp315Pbog5aL0w@4
./build/l686-w64-mingw32-l686-w64-mingw32-crt
_callSoonProc__9b9b4iUsd60R02Uqc52lfJ6A
_cb64safe_q0GCLBVLDDeEKBBmNHUGw
@cellSetEnlarge__9bhPFIgy1neoHljx80Xvqa@4
@cellSetGet__ld9aj9akVqlCvvrCEMEk1MnQ@20
@cellSetPut__6B8l0A4VUxOrvva9bRnnwS@8
@cellSetRawInsert__aisVKTgcDTTmcnBQqkbsNdA@24
@cellsetReset__Y9c9c0hDRSgYkHfKwcaEls0_@24
```

The total number of strings are 572 with length greater than 30 in this case.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><assembly xmlns="urn:schemas-microsoft-com:asm.v1"
.0.0" processorArchitecture="*" name="winini" type="win32"/><dependency><dependentAssembly><assemblyIdentity
version="6.0.0.0" processorArchitecture="*" publicKeyToken="6595b64144ccfidf" language="" /></dependentAssembly>
lakshminath@lakshminathvm2004:~/IOT$ strings -n 30 Malware.yara1.exe.malz| sort | uniq | wc -l
572
lakshminath@lakshminathvm2004:~/IOT$
```

Task-2

For the task-2, I have created a myrule1.yar file with two extracted string from the task-1

```
1 rule myrule1 {
2     meta:
3         description = "Detect malware strings from sample Malware.yara1.exe.malz"
4         author      = "Lakshminath Alamuru"
5         date        = "10-30-2025"
6         hash1       = "e128283461b14224459e966abf317c070d50aae7a531d64"
7
8     strings:
9         // Example unique strings extracted from your malware sample
10        $s1 = "_acceptEx__qL9a0i8PW0vbR8SHPxunTBw" ascii
11        $s2 = "@add__3D9a0yz4rDquPZKBln0xig@16" ascii
12        $mz  = {4D 5A}    // MZ magic number for PE files
13
14     condition:
15         // Check if at least 2 of the defined strings are present
16         (uint16(0) == 0x5A4D) and          // MZ at offset 0 (little-endian)
17         (#s1 + #s2 >= 2)              // at least 2 of the defined strings
18
19 }
```

Task-3

Test the yara rule against the malware file that has given in exercise

```
rule myrule1 {
    meta:
        description = "Detect malware strings from sample Malware.yara1.exe.malz"
        author      = "Lakshminath Alamuru"
        date        = "10-30-2025"
        hash1       = "e128283461b14224459e966abf317c070d50aae7a531d64"

    strings:
        // Example unique strings extracted from your malware sample
        $s1 = "_acceptEx__qL9a0i8PW0vbR8SHPxunTBw" ascii
        $s2 = "@add__3D9a0yz4rDquPZKBln0xig@16" ascii
        $mz  = {4D 5A}    // MZ magic number for PE files

    condition:
        $mz at 0 and (#s1 + #s2 >= 2) // at least 2 of the defined strings and $mz at 0
}
```

```
Lakshminath@lakshminathvm2004:~/IOT$ yara myrule1.yar ~/IOT/Malware.yara1.exe.malz
error: rule "myrule1" in myrule1.yar(18): unreferenced string "$mz"
Lakshminath@lakshminathvm2004:~/IOT$ yara myrule1.yar ~/IOT/Malware.yara1.exe.malz
myrule1 /home/lakshminath/IOT/Malware.yara1.exe.malz
Lakshminath@lakshminathvm2004:~/IOT$
```

myrule1.yar file is has run against malware file successfully here. It seems no condition strings are found here.

Task-4



```
myrule1.yar Task4_yara_rule.yar
Save × × × ×

1 rule my_entropy_rule {
2   meta:
3     description = "Detect malware sample based on string, header, size"
4     author      = "Lakshminath Alamuru"
5     date        = "10-30-2025"
6
7   strings:
8     $nd = "ExampleStringFromMalware" ascii
9
10  condition:
11    $nd and           // string must exist
12    uint16(0) == 0x457f and // first 2 bytes in little-endian = 0x457f
13    filesize <= 20000 * 1024 // file size ≤ 20,000 KB
14 }
15
```

Run the Task4_yara_rule.yar file again malware file didn't get any output. Seems the conditions are not met.

```
Lakshminath@lakshminathvm2004:~/IOT$ yara -r Task4_yara_rule.yar ~/IOT/Malware.yara1.exe.malz
error: rule "my_entropy_rule" in Task4_yara_rule.yar(14): undefined identifier "entropy"
Lakshminath@lakshminathvm2004:~/IOT$ yara -r Task4_yara_rule.yar ~/IOT/Malware.yara1.exe.malz
Lakshminath@lakshminathvm2004:~/IOT$
```

Observations:

I got to know how to write war file and can check for the conditions met for string matching and file size checking as well. At least in the conditions that I have taken, those are not met. But it is successful learning for me about yara rules in detail.