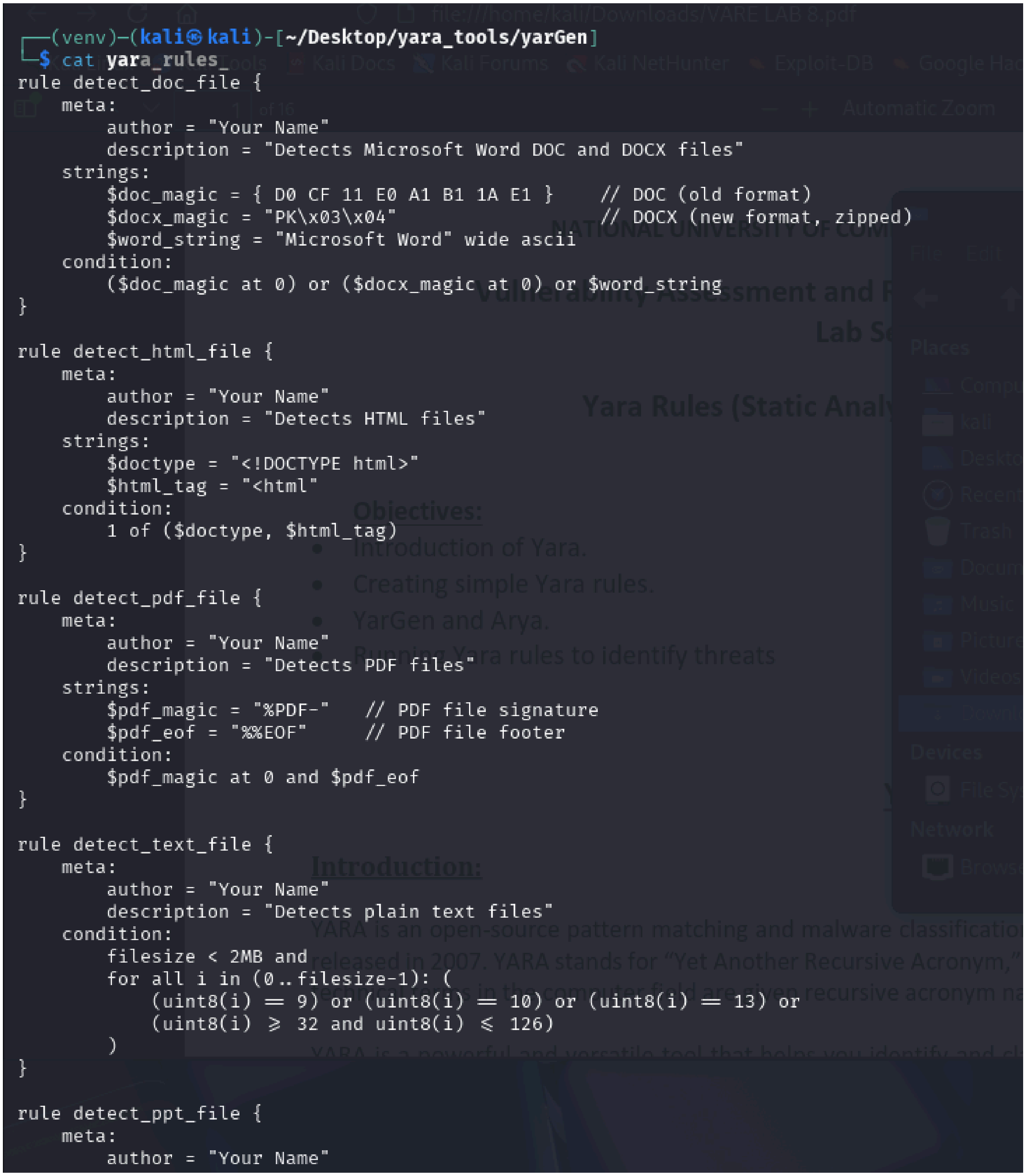
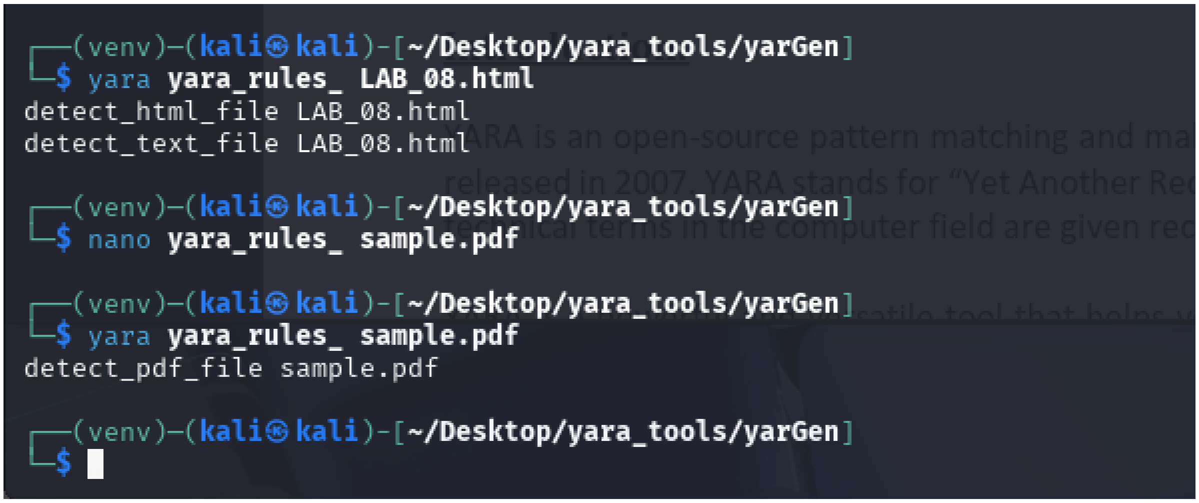
VARE LAB 08

LAB TASK 1
Yara Rules



```
┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen]
└─$ cat yara_rules_
rule detect_doc_file {
    meta:
        author = "Your Name"
        description = "Detects Microsoft Word DOC and DOCX files"
    strings:
        $doc_magic = { D0 CF 11 E0 A1 B1 1A E1 }    // DOC (old format)
        $docx_magic = "PK\x03\x04"                  // DOCX (new format, zipped)
        $word_string = "Microsoft Word" wide ascii
    condition:
        ($doc_magic at 0) or ($docx_magic at 0) or $word_string
}

rule detect_html_file {
    meta:
        author = "Your Name"
        description = "Detects HTML files"
    strings:
        $doctype = "<!DOCTYPE html>"
        $html_tag = "<html"
    condition:
        1 of ($doctype, $html_tag)
}

rule detect_pdf_file {
    meta:
        author = "Your Name"
        description = "Detects PDF files"
    strings:
        $pdf_magic = "%PDF-"    // PDF file signature
        $pdf_eof = "%%EOF"      // PDF file footer
    condition:
        $pdf_magic at 0 and $pdf_eof
}

rule detect_text_file {
    meta:
        author = "Your Name"
        description = "Detects plain text files"
    condition:
        filesize < 2MB and
        for all i in (0..filesize-1): (
            (uint8(i) == 9) or (uint8(i) == 10) or (uint8(i) == 13) or
            (uint8(i) >= 32 and uint8(i) <= 126)
        )
}

rule detect_ppt_file {
    meta:
        author = "Your Name"
```

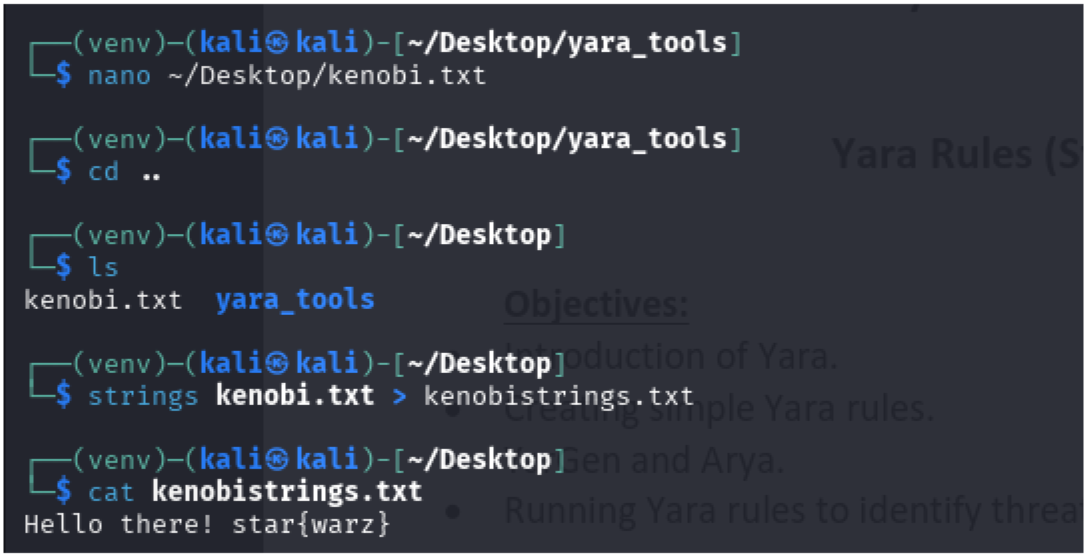
Now Running the yara rules script with sample files such as html and pdf .



```
┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen]
└─$ yara yara_rules_ LAB_08.html
detect_html_file LAB_08.html
detect_text_file LAB_08.html

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen]
└─$ nano yara_rules_ sample.pdf

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen]
└─$ yara yara_rules_ sample.pdf
detect_pdf_file sample.pdf

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen]
└─$ 
```

LAB TASK 02

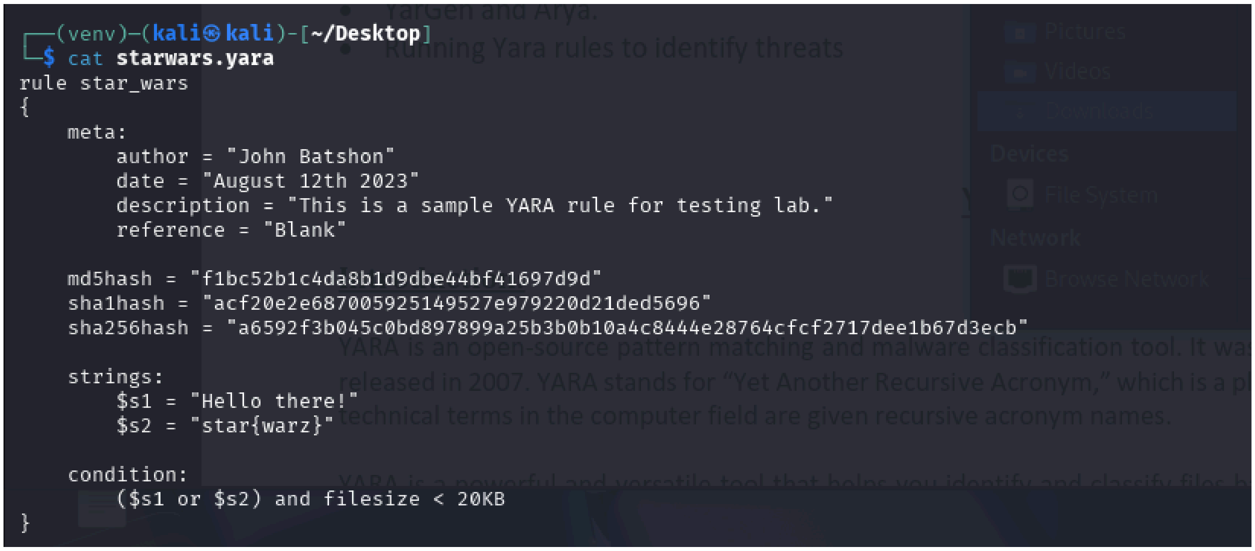
Make file using the pattern star{warz}



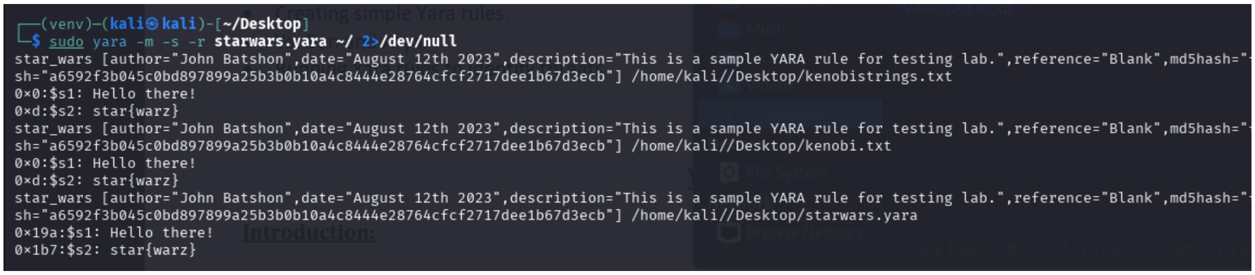
```
┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools]
└─$ nano ~/Desktop/kenobi.txt

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools]
└─$ cd ..

┌──(venv)─(kali㉿kali)-[~/Desktop]
└─$ ls
kenobi.txt  yara_tools

┌──(venv)─(kali㉿kali)-[~/Desktop]
└─$ strings kenobi.txt > kenobistrings.txt

┌──(venv)─(kali㉿kali)-[~/Desktop]
└─$ cat kenobistrings.txt
Hello there! star{warz}
```

Now testing the file for starwar.yara

This is the script

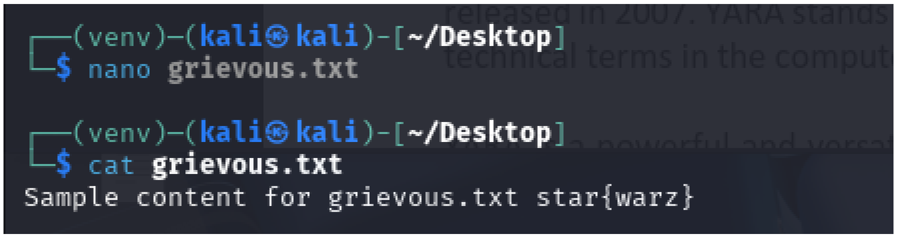


```
┌──(venv)─(kali㉿kali)-[~/Desktop]
└─$ cat starwars.yara
rule star_wars
{
    meta:
        author = "John Batshon"
        date = "August 12th 2023"
        description = "This is a sample YARA rule for testing lab."
        reference = "Blank"

    md5hash = "f1bc52b1c4da8b1d9dbe44bf41697d9d"
    sha1hash = "acf20e2e687005925149527e979220d21ded5696"
    sha256hash = "a6592f3b045c0bd897899a25b3b0b10a4c8444e28764cfcf2717dee1b67d3ecb"

    strings:
        $s1 = "Hello there!"
        $s2 = "star{warz}"

    condition:
        ($s1 or $s2) and filesize < 20KB
}
```



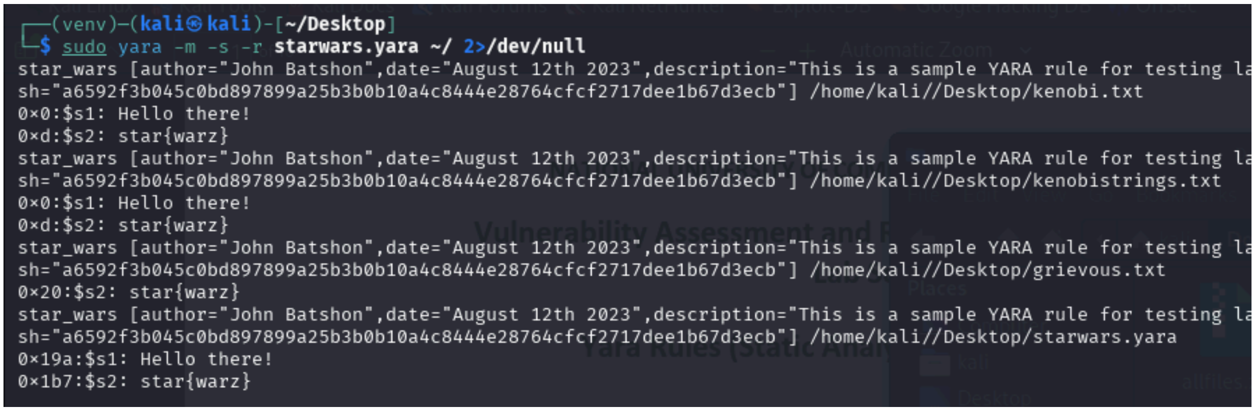
```
┌──(venv)─(kali㉿kali)-[~/Desktop]
└─$ sudo yara -m -s -r starwars.yara ~/ 2>/dev/null
star_wars [author="John Batshon",date="August 12th 2023",description="This is a sample YARA rule for testing lab.",reference="Blank",md5hash="
sh="a6592f3b045c0bd897899a25b3b0b10a4c8444e28764cfcf2717dee1b67d3ecb"] /home/kali//Desktop/kenobistrings.txt
0×0:$s1: Hello there!
0×d:$s2: star{warz}
star_wars [author="John Batshon",date="August 12th 2023",description="This is a sample YARA rule for testing lab.",reference="Blank",md5hash="
sh="a6592f3b045c0bd897899a25b3b0b10a4c8444e28764cfcf2717dee1b67d3ecb"] /home/kali//Desktop/kenobi.txt
0×0:$s1: Hello there!
0×d:$s2: star{warz}
star_wars [author="John Batshon",date="August 12th 2023",description="This is a sample YARA rule for testing lab.",reference="Blank",md5hash="
sh="a6592f3b045c0bd897899a25b3b0b10a4c8444e28764cfcf2717dee1b67d3ecb"] /home/kali//Desktop/starwars.yara
0×19a:$s1: Hello there!
0×1b7:$s2: star{warz}
```

Now making another file which also trigger this



```
┌──(venv)─(kali㉿kali)-[~/Desktop]
└─$ nano grievous.txt

┌──(venv)─(kali㉿kali)-[~/Desktop]
└─$ cat grievous.txt
Sample content for grievous.txt star{warz}
```

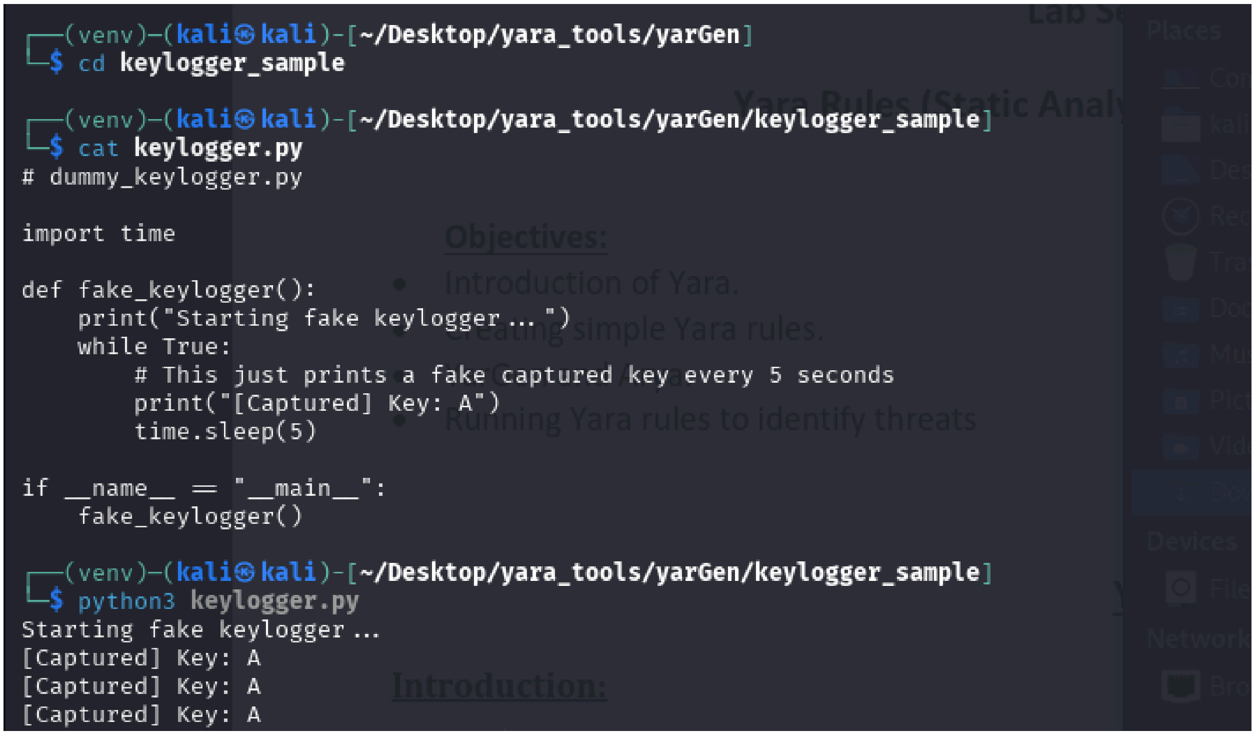
Now run the scripts and new grievous.txt triggers



```
┌──(venv)─(kali㉿kali)-[~/Desktop]
└─$ sudo yara -m -s -r starwars.yara ~/ 2>/dev/null
star_wars [author="John Batshon",date="August 12th 2023",description="This is a sample YARA rule for testing la
sh="a6592f3b045c0bd897899a25b3b0b10a4c8444e28764cfcf2717dee1b67d3ecb"] /home/kali//Desktop/kenobi.txt
0×0:$s1: Hello there!
0×d:$s2: star{warz}
star_wars [author="John Batshon",date="August 12th 2023",description="This is a sample YARA rule for testing la
sh="a6592f3b045c0bd897899a25b3b0b10a4c8444e28764cfcf2717dee1b67d3ecb"] /home/kali//Desktop/kenobistrings.txt
0×0:$s1: Hello there!
0×d:$s2: star{warz}
star_wars [author="John Batshon",date="August 12th 2023",description="This is a sample YARA rule for testing la
sh="a6592f3b045c0bd897899a25b3b0b10a4c8444e28764cfcf2717dee1b67d3ecb"] /home/kali//Desktop/grievous.txt
0×20:$s2: star{warz}
star_wars [author="John Batshon",date="August 12th 2023",description="This is a sample YARA rule for testing la
sh="a6592f3b045c0bd897899a25b3b0b10a4c8444e28764cfcf2717dee1b67d3ecb"] /home/kali//Desktop/starwars.yara
0×19a:$s1: Hello there!
0×1b7:$s2: star{warz}
```

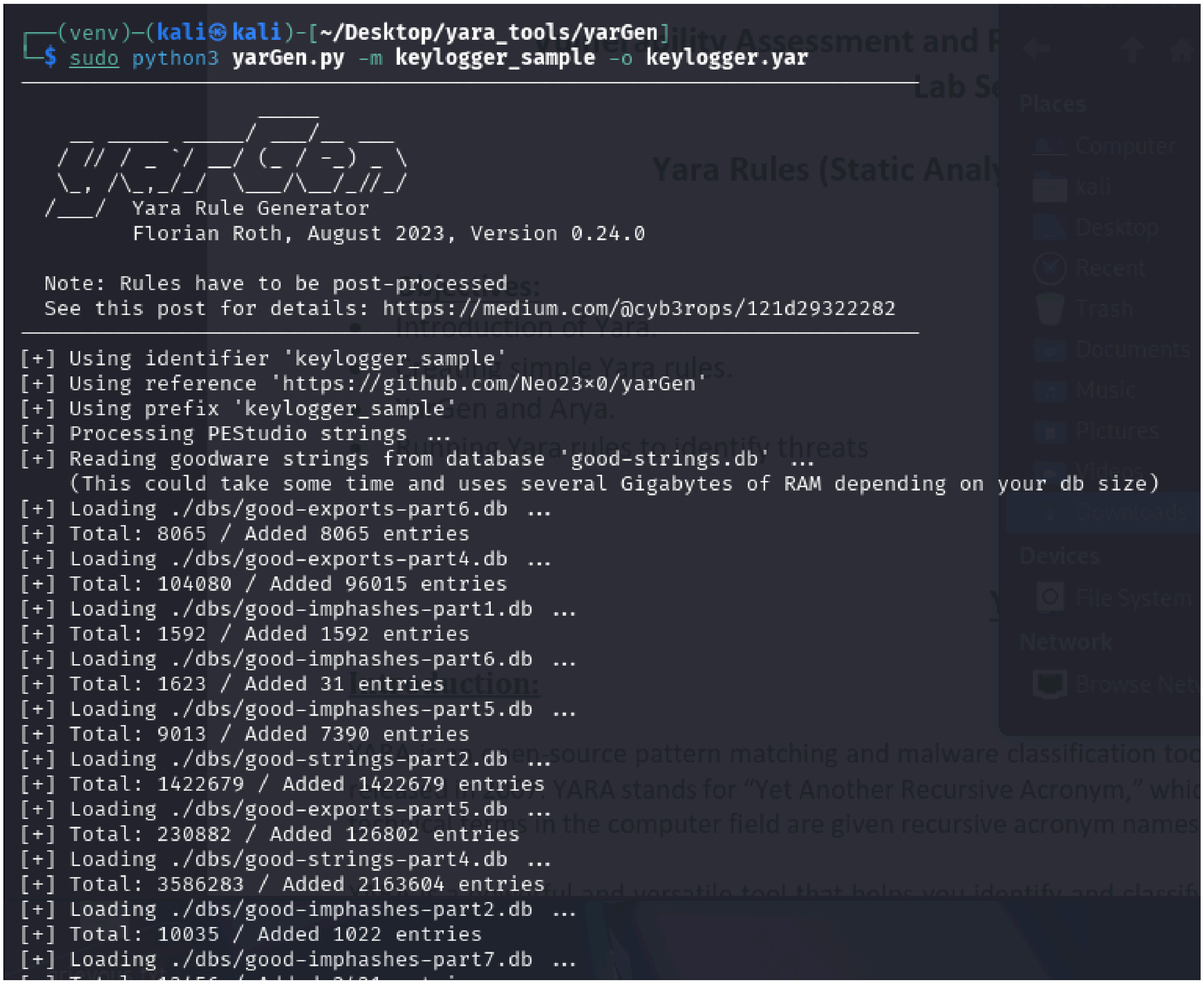
LAB 03

Sample key logger



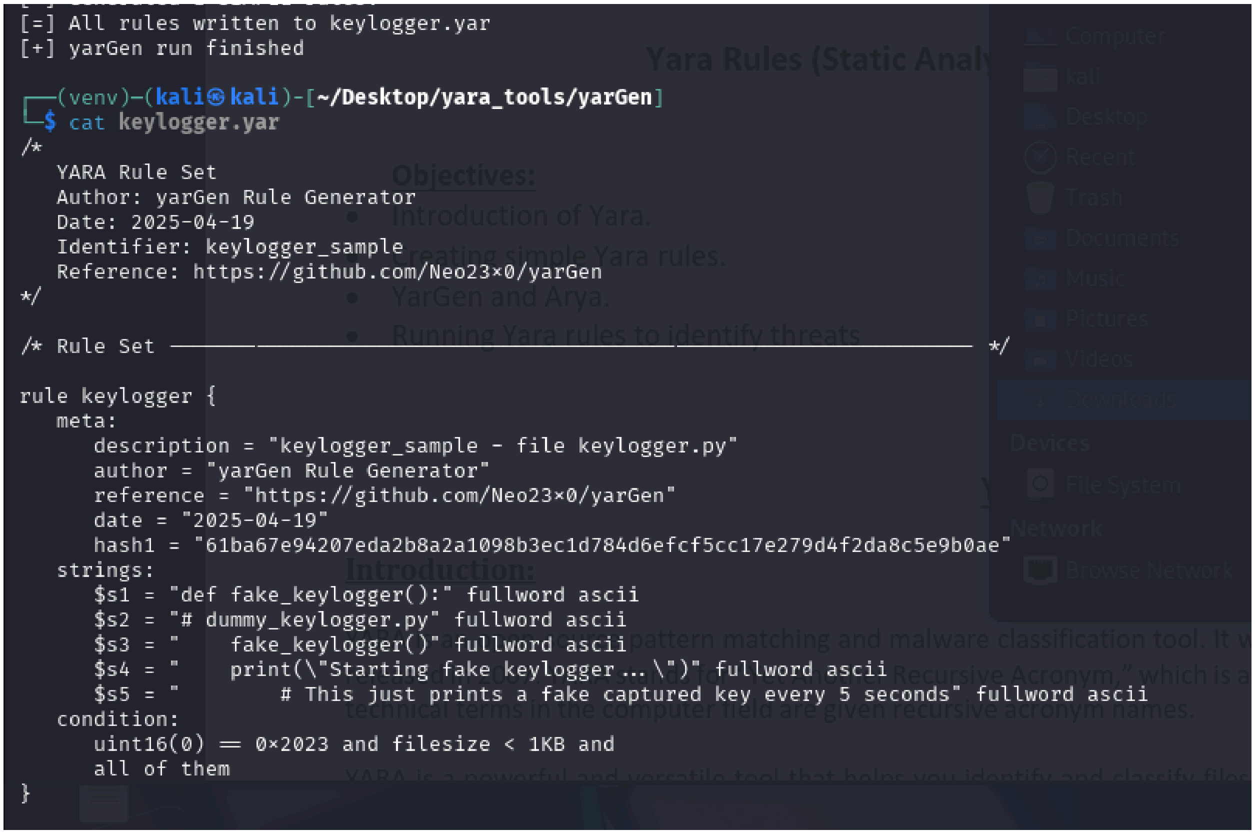
```
┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen]
└─$ cd keylogger_sample

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen/keylogger_sample]
└─$ cat keylogger.py
# dummy_keylogger.py

import time

def fake_keylogger():
    print("Starting fake keylogger...")
    while True:
        # This just prints a fake captured key every 5 seconds
        print("[Captured] Key: A")
        time.sleep(5)

if __name__ == "__main__":
    fake_keylogger()

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen/keylogger_sample]
└─$ python3 keylogger.py
Starting fake keylogger...
[Captured] Key: A
[Captured] Key: A
[Captured] Key: A
```

now making rule with the help of the yaragen



```
┌──(venv)-(kali㉿kali)-[~/Desktop/yara_tools/yarGen]
└─$ sudo python3 yarGen.py -m keylogger_sample -o keylogger.yar
────────────────────────────────────────────────────────────────────────
                   ______
   __  ______ ______/ ____/__  ____
  / / / / __ `/ ___/ / __/ _ \/ __ \
 / /_/ / /_/ / /  / /_/ /  __/ / / /
 \__, /\__,_/_/   \____/\___/_/ /_/
/____/  Yara Rule Generator
        Florian Roth, August 2023, Version 0.24.0

  Note: Rules have to be post-processed
  See this post for details: https://medium.com/@cyb3rops/121d29322282
────────────────────────────────────────────────────────────────────────
[+] Using identifier 'keylogger_sample'
[+] Using reference 'https://github.com/Neo23x0/yarGen'
[+] Using prefix 'keylogger_sample'
[+] Processing PEStudio strings ...
[+] Reading goodware strings from database 'good-strings.db' ...
    (This could take some time and uses several Gigabytes of RAM depending on your db size)
[+] Loading ./dbs/good-exports-part6.db ...
[+] Total: 8065 / Added 8065 entries
[+] Loading ./dbs/good-exports-part4.db ...
[+] Total: 104080 / Added 96015 entries
[+] Loading ./dbs/good-imphashes-part1.db ...
[+] Total: 1592 / Added 1592 entries
[+] Loading ./dbs/good-imphashes-part6.db ...
[+] Total: 1623 / Added 31 entries
[+] Loading ./dbs/good-imphashes-part5.db ...
[+] Total: 9013 / Added 7390 entries
[+] Loading ./dbs/good-strings-part2.db ...
[+] Total: 1422679 / Added 1422679 entries
[+] Loading ./dbs/good-exports-part5.db ...
[+] Total: 230882 / Added 126802 entries
[+] Loading ./dbs/good-strings-part4.db ...
[+] Total: 3586283 / Added 2163604 entries
[+] Loading ./dbs/good-imphashes-part2.db ...
[+] Total: 10035 / Added 1022 entries
[+] Loading ./dbs/good-imphashes-part7.db ...
```

This is the rule



```
[=] All rules written to keylogger.yar
[+] yarGen run finished

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/yarGen]
└─$ cat keylogger.yar
/*
   YARA Rule Set
   Author: yarGen Rule Generator
   Date: 2025-04-19
   Identifier: keylogger_sample
   Reference: https://github.com/Neo23x0/yarGen
*/

/* Rule Set ----------------------------------------------------------------- */

rule keylogger {
   meta:
      description = "keylogger_sample - file keylogger.py"
      author = "yarGen Rule Generator"
      reference = "https://github.com/Neo23x0/yarGen"
      date = "2025-04-19"
      hash1 = "61ba67e94207eda2b8a2a1098b3ec1d784d6efcf5cc17e279d4f2da8c5e9b0ae"
   strings:
      $s1 = "def fake_keylogger():" fullword ascii
      $s2 = "# dummy_keylogger.py" fullword ascii
      $s3 = "    fake_keylogger()" fullword ascii
      $s4 = "    print(\"Starting fake keylogger...\")" fullword ascii
      $s5 = "        # This just prints a fake captured key every 5 seconds" fullword ascii
   condition:
      uint16(0) == 0x2023 and filesize < 1KB and
      all of them
}
```

LAB Task 4

Arya is a YARA rule formatting and generation tool developed by Florian Roth, the same creator of yarGen. Its primary purpose is to clean up, standardize, and optimize YARA rules for better readability and maintenance.

Key features of Arya:

1. Rule Formatting: Automatically adjusts indentation and organizes rule sections (meta, strings, condition) in a consistent structure
2. Syntax Correction: Fixes common syntax issues in YARA rules
3. Standardization: Ensures all rules follow the same formatting conventions
4. Rule Generation: Can help create properly formatted rules from scratch

Typical workflow:

1. Create a YARA rule (manually or using tools like yarGen)
2. Process the rule through Arya
3. Receive a cleaned-up, standardized version of the rule

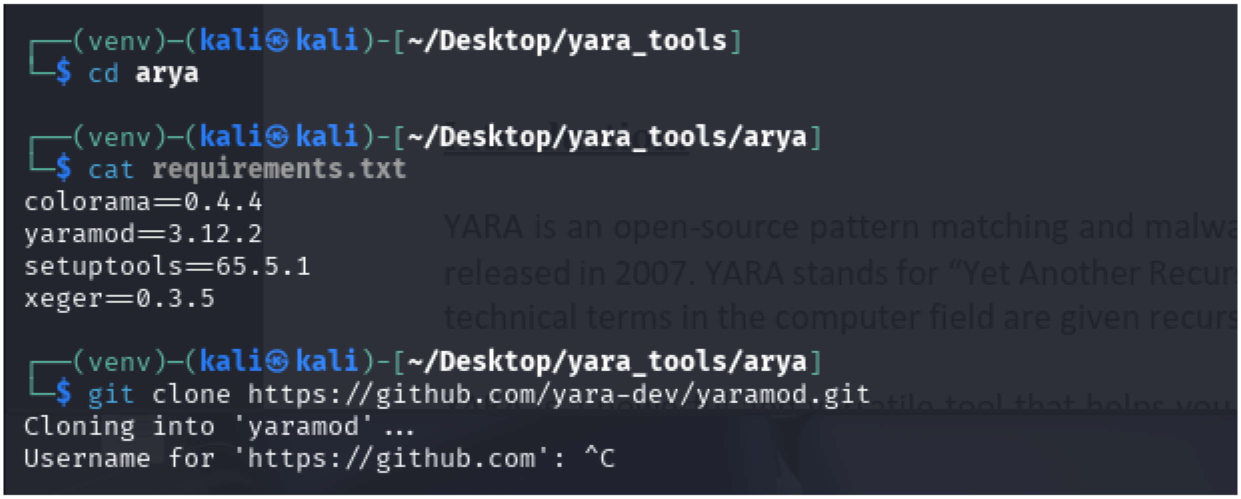
Use cases:

- When collaborating with teams to maintain consistent rule formatting
- When preparing rules for production deployment
- When cleaning up rules generated by automated tools
- When reviewing or refactoring existing rule sets

Arya is particularly useful for security teams that work with large collections of YARA rules, helping maintain consistency across the entire rulebase.

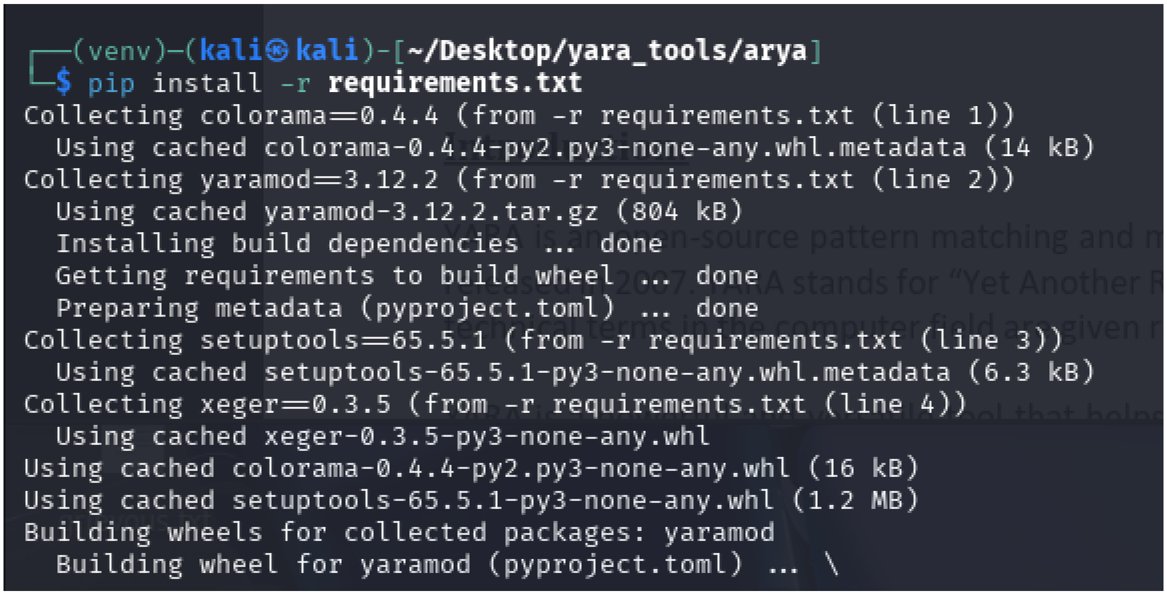
Arya is not working properly there is issue in the yaramod

I tried to install manually from



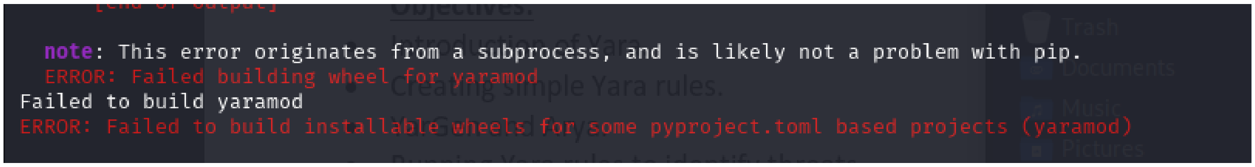
```
┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools]
└─$ cd arya

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/arya]
└─$ cat requirements.txt
colorama==0.4.4
yaramod==3.12.2
setuptools==65.5.1
xeger==0.3.5

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/arya]
└─$ git clone https://github.com/yara-dev/yaramod.git
Cloning into 'yaramod'...
Username for 'https://github.com': ^C
```

If i simple installing the all the requirement it stuck here



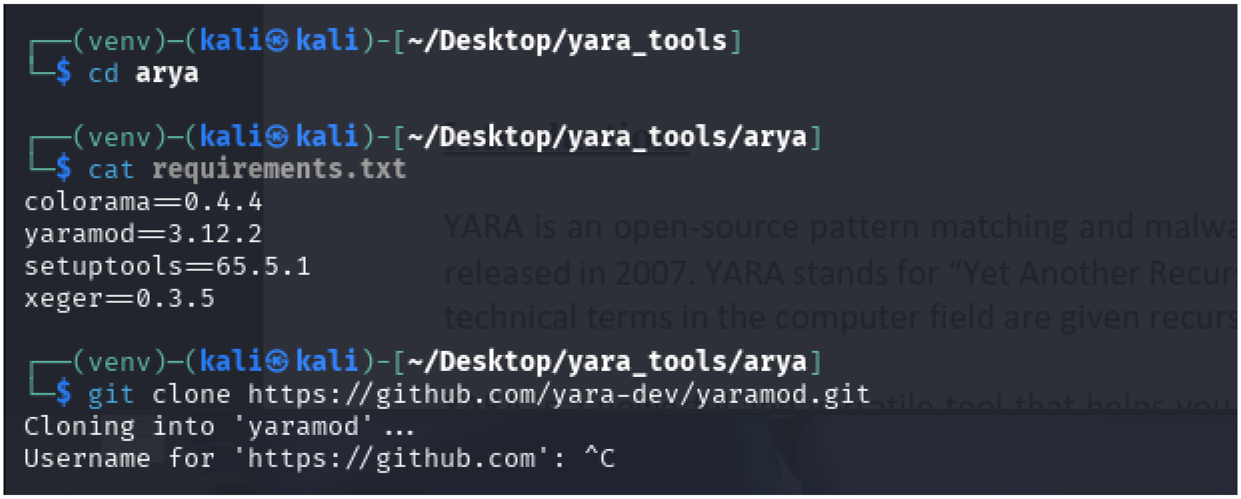
```
┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/arya]
└─$ pip install -r requirements.txt
Collecting colorama==0.4.4 (from -r requirements.txt (line 1))
  Using cached colorama-0.4.4-py2.py3-none-any.whl.metadata (14 kB)
Collecting yaramod==3.12.2 (from -r requirements.txt (line 2))
  Using cached yaramod-3.12.2.tar.gz (804 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting setuptools==65.5.1 (from -r requirements.txt (line 3))
  Using cached setuptools-65.5.1-py3-none-any.whl.metadata (6.3 kB)
Collecting xeger==0.3.5 (from -r requirements.txt (line 4))
  Using cached xeger-0.3.5-py3-none-any.whl
Using cached colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Using cached setuptools-65.5.1-py3-none-any.whl (1.2 MB)
Building wheels for collected packages: yaramod
  Building wheel for yaramod (pyproject.toml) ... \
```

Then it shows the error message like



```
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
  ERROR: Failed building wheel for yaramod
Failed to build yaramod
ERROR: Failed to build installable wheels for some pyproject.toml based projects (yaramod)
```

so for this i manually clone the git but it also causing issues
As i attached ss above



```
┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools]
└─$ cd arya

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/arya]
└─$ cat requirements.txt
colorama==0.4.4
yaramod==3.12.2
setuptools==65.5.1
xeger==0.3.5

┌──(venv)─(kali㉿kali)-[~/Desktop/yara_tools/arya]
└─$ git clone https://github.com/yara-dev/yaramod.git
Cloning into 'yaramod'...
Username for 'https://github.com': ^C
```