



```

USAGE: Snorter.sh -i INTERFACE
USAGE: Snorter.sh -o OINKCODE -i INTERFACE
Example: Snorter.sh -o 123456abcdefg -i eth0

```

@joan_bono

What do you need?

- A computer running:
 - **Debian**
 - **Kali Linux**
 - **Raspbian Jessie**
- Oinkcode:
 - It's **FREE!** 😊
 - Highly recommended!
 - Get yours [here](#).
- Identified Network Interface:
 - `ip link show`
- Previous dependencies:
 - `sudo apt-get install git`
- Patience.

First steps

- Cloning the repository:

```
git clone https://github.com/joanbono/Snorter.git`  
cd Snorter/src  
bash Snorter.sh -h
```

- **Recommended:** Execute the program using an **oinkcode**

```
bash Snorter.sh -o <oinkcode> -i <interface>  
Ex: bash Snorter.sh -o XXXXXXXXXXXXXXXX -i eth0
```

- **Not Recommended:** Execute the program without an **oinkcode**

```
bash Snorter.sh -i interface  
bash Snorter.sh -i eth0
```

Snort installation

- Superuser password, and wait...

```
debian@debian:~/Snorter/src$ bash Snorter.sh -o [REDACTED] -i eth0
```

```

,--
\(\(_--
<\--\>/(/(_
/._. \
('') , @
\_._, /
)_-)/--( >
,,,,,

```

A complex, multi-colored circuit diagram. It features several vertical columns of components. The first column on the left has red components. The second column has grey components. The third column has yellow components. The fourth column has green components. The fifth column has cyan components. The sixth column has blue components. The seventh column has magenta components. The components are interconnected by a network of lines of corresponding colors, forming a dense, interconnected web. The overall layout is symmetrical and highly structured, suggesting a complex digital or analog circuit design.

```
[+] OINKCODE: XXXXXXXXXX
[+] INTERFACE: eth0
[+] DAQ: daq-2.0.6
[+] SNORT: snort-2.9.9.0
[+] ARCH: x86_64
```

```
[i] INFO: Updating and Upgrading repositories...
```

```
[sudo] password for debian:
```

- Snort and daq are installed.

```
[+] INFO: snort-2.9.9.0 installed successfully.
```

```
[i] INFO: Adding user and group SNORT.
```

```
[i] INFO: /var/log/snort and /etc/snort created and configured.
```

```
,,-      -*> Snort! <*-  
o"  )~   Version 2.9.9.0 GRE (Build 56)  
' ' '   By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
        Copyright (C) 2014-2016 Cisco and/or its affiliates. All rights reserved.  
        Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
        Using libpcap version 1.6.2  
        Using PCRE version: 8.35 2014-04-04  
        Using ZLIB version: 1.2.8
```

```
[+] INFO: SNORT is successfully installed and configured!
```

- Now it's time to add the `HOME_NET` and the `EXTERNAL_NET` .

```
[!] INFO: Now it's time to edit the SNORT configuration file.
```

```
[i] INFO: Add your HOME_NET address [Ex: 192.168.1.0/24]
```

```
[!] WARNING: Press ENTER to continue. █
```

- Press `Enter` to continue. It will open `vim` :
 - Press `A` to go to the end of the line.
 - Add the address and the mask you want to protect.
 - Press `Esc` and then `:wq!` to save the changes.

```
39
40 #####
41 # Step #1: Set the network variables. For more information, see README.variables
42 #####
43
44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 172.16.77.0/24
46
```

- Do the same for the `EXTERNAL_NET` :

```
[i] INFO: Add your EXTERNAL_NET address [Ex: !$HOME_NET]  
[!] WARNING: Press ENTER to continue. █
```

- Press `Enter` to continue. It will open `vim` :
 - Press `A` to go to the end of the line.
 - Add the *attacker* address. **Recommended:** `!$HOME_NET` .
 - Press `Esc` and then `:wq!` to save the changes.

```
46  
47 # Set up the external network addresses. Leave as "any" in most situations  
48 ipvar EXTERNAL_NET !$HOME_NET  
49
```

- Now the **output**. By default, `unified2` output is enabled, but you can enable more than one output. I'm going to enable both **CSV** and **TCPdump** output.

```
[i] INFO: Enabling local.rules and adding a PING detection rule...  
[!] WARNING: Unified2 output configured. Configure another output?  
    1 - CSV output  
    2 - TCPdump output  
    3 - CSV and TCPdump output  
    4 - None  
  
Option [1-4]: █
```


- Now **SNORT** will start in **console** mode. Send a **PING** from another machine.

```
[!] WARNING: Attempting to test ICMP rule in eth0. Send a PING to your SNORT machine. Press Ctrl+C once and wait few seconds to stop the process...

[!] WARNING: Press ENTER to continue.
01/09-12:39:29.229291  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.1 -> 172.16.77.137
01/09-12:39:29.229320  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.137 -> 172.16.77.1
01/09-12:39:30.229230  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.1 -> 172.16.77.137
01/09-12:39:30.229294  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.137 -> 172.16.77.1
01/09-12:39:31.230473  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.1 -> 172.16.77.137
01/09-12:39:31.230526  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.137 -> 172.16.77.1
01/09-12:39:32.231436  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.1 -> 172.16.77.137
01/09-12:39:32.231553  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.137 -> 172.16.77.1
01/09-12:39:33.236303  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.1 -> 172.16.77.137
01/09-12:39:33.236387  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.137 -> 172.16.77.1
01/09-12:39:34.241661  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.1 -> 172.16.77.137
01/09-12:39:34.241796  [**] [1:10000001:1] Atac per PINGS [**] [Priority: 0] {ICMP} 172.16.77.137 -> 172.16.77.1
^C*** Caught Int-Signal
snort: no process found
```

- It will show a **PING** alert. Press **Ctrl+C** once, and continue the installation.

Barnyard2 installation

- Now it's time to install `BARNYARD2` if you want.
- You will be asked to insert a password for the `SNORT` database which is going to be created. In the example, I've used `SNORTSQL`

```
[!] IMPORTANT: Would you like to install BARNYARD2? [Y/n] Y
```

```
[!] WARNING: Insert new SNORT Database Password: SNORTSQL
```

- Now the program will install dependencies.
- It's going to install `MySQL` , so if it's not installed, you will insert a password for this service too. In the example, I've used `ROOTSQL` .

```
[i] INFO: Installing dependencies.  
[!] WARNING: You will be asked for a password for MySQL service if it isn't installed in the system.  
[!] WARNING: Press ENTER to continue. █
```

- And the MySQL password.

Configuring mysql-server-5.5

Repeat password for the MySQL "root" user:

<Ok>

- Now you are going to be asked for the `MySQL` password **3 times**
- Please keep in mind: `MySQL` `root` password **3 times**.

```
[+] INFO: BARNYARD2 installed successfully.
```

```
[i] INFO: The SNORT database is going to be created. You will be asked for MySQL password 3 times
```

```
[!] WARNING: Press ENTER to continue.
```

```
Enter password:
```

```
Enter password:
```

```
Enter password: █
```

PuLledPork installation

- Now it's time to install PuLledPork if you want.

```
[!] IMPORTANT: Would you like to install PULLEDPORK? [Y/n] Y
```

```
[i] INFO: Downloading PULLEDPORK.
```

```
Cloning into 'pulledpork'...
```

```
remote: Counting objects: 1207, done.
```

```
remote: Total 1207 (delta 0), reused 0 (delta 0), pack-reused 1207
```

```
Receiving objects: 100% (1207/1207), 249.49 KiB | 0 bytes/s, done.
```

```
Resolving deltas: 100% (814/814), done.
```

```
Checking connectivity... done.
```

```
[i] INFO: Adding PULLEDPORK to crontab. [Everyday at 4:15 AM].
```

```
PulledPork v0.7.3 - Making signature updates great again!
```

```
[+] INFO: PULLEDPORK is successfully installed and configured!
```

Emerging Threats ruleset

- Do you want to add the **Emerging Threats** rules and the **community** ones (enabled by default)?

```
[!] IMPORTANT: Would you like to enable Emerging Threats rules? [Y/n] Y
[+] INFO: Emerging Threats rules enabled!
[i] INFO: Editing pulledpork.conf settings...

[!] IMPORTANT: Now edit your /etc/snort/snort.conf and enable the rules you need by uncomment the lines
[!] EXAMPLE: If you want to enable the Exploit rules, remove the #:
    #include $RULE_PATH/exploit.rules --> include $RULE_PATH/exploit.rules
```

- Remember to edit your `/etc/snort/snort.conf` and remove the `#` to enable the rules you want or the ones you need.

service creation

- Create a system service :

```
[!] IMPORTANT: Would you like to create a service snort? [Y/n] Y
```

```
[i] INFO: Now you can run sudo service snort {start|stop|status}.
```


Download and install new rules

- You can download rules when everything is installed and configured.

```
[!] IMPORTANT: Would you like to download new rules using PULLEDPORK? [Y/n] Y
https://github.com/shirkdog/pulledpork
  -----
  \-----,\      )
  \--==\\  /      PulledPork v0.7.3 - Making signature updates great again!
  \--==\\ /
  .-~~~~-.Y|\\_   Copyright (C) 2009-2016 JJ Cummings
@_/_      / 66\_   cummingsj@gmail.com
  |      \  \ _(")
  \      /-| ||'--' Rules give me wings!
  \_ \  \_ \ \
~~~~~
```

Reboot

- Reboot the system.

```
[!] IMPORTANT: Would you like to REBOOT now? [Y/n] Y
```

```
[i] INFO: Rebooting...
```

Enjoy!

Please, feel free to open **issues** if you have any problem with the program.