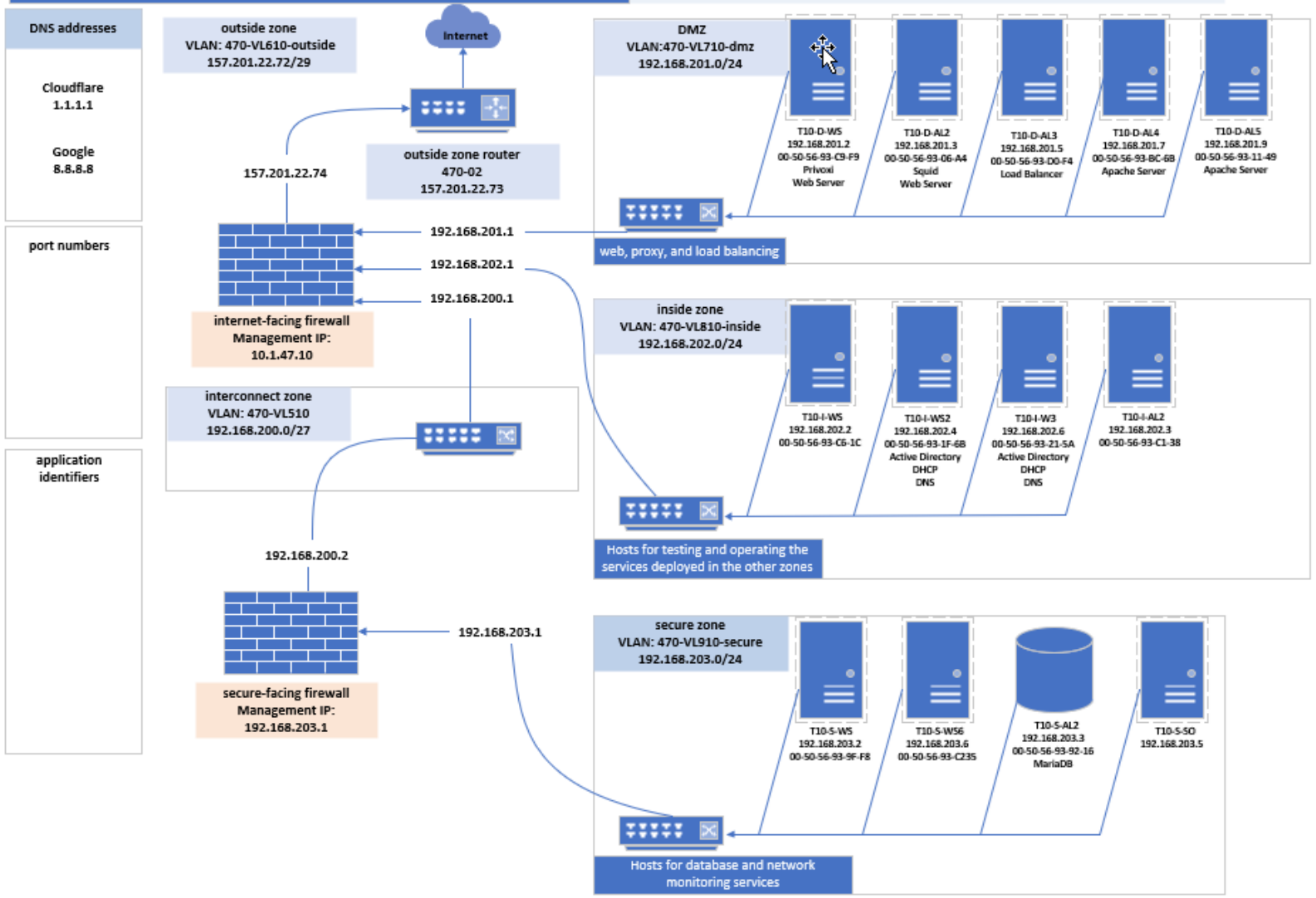
IDS Sensor

cit470
Task: Diagram



Deploy VM

Deploy security Onion VM with at least 4 CPUs and 16 GB of memory, and a 200 GB hard drive.

Add 2 NICs, one connected to the DMZ and on connected to the Secure zone.

For this example we will use securityonion-2.3.210-20230202.iso

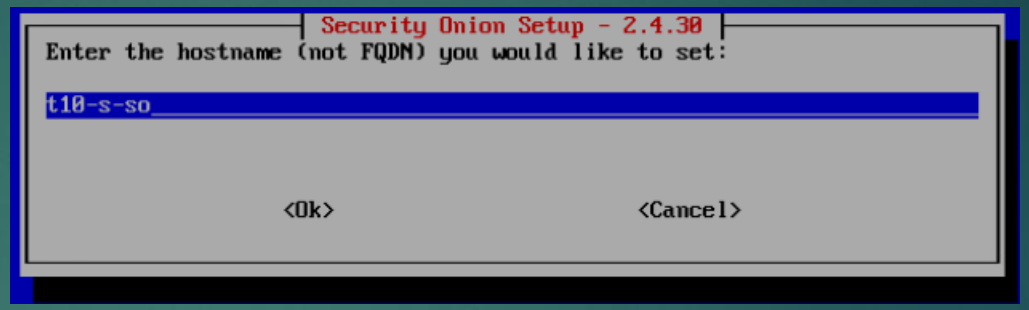
Guest OS name	Red Hat Enterprise Linux 7 (64-bit)
Virtualization Based Security	Disabled
CPUs	4
Memory	16 GB
NICs	2
NIC 1 network	470-VL910-secure (CIT-DSwitch)
NIC 1 type	VMXNET 3
NIC 2 network	470-VL710-dmz (CIT-DSwitch)
NIC 2 type	VMXNET 3
SCSI controller 1	VMware Paravirtual
New hard disk 1	
Capacity	200 GB
Datastore	CIT [UCS ESXi v104 - SMIF700] (Recommended)

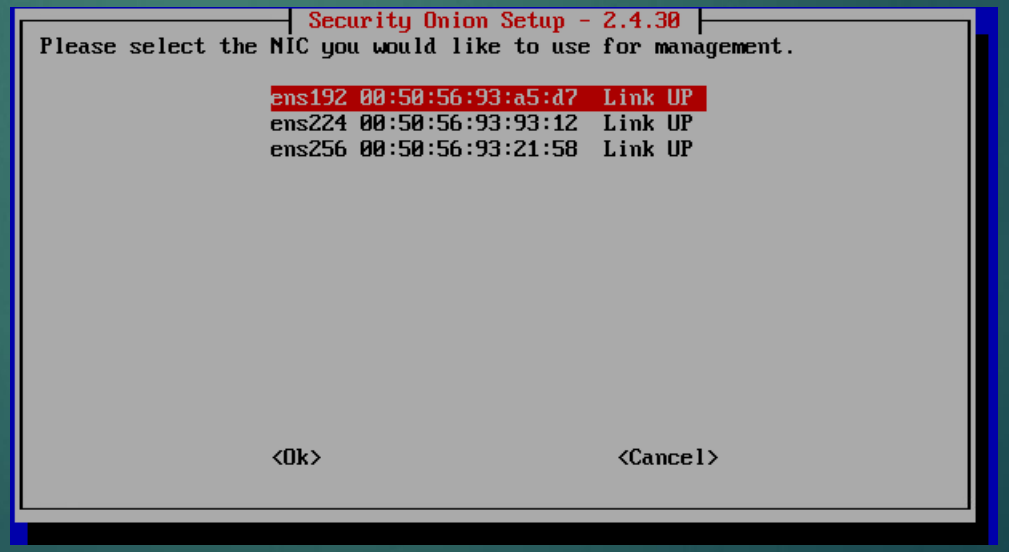
Security Onion Configuration

Launch the new VM and configure Security Onion

Enter the hostname you would like to use (in this example we will use t10-s-so

Pick the NIC you would like to use for management. Use the first choice, which is the NIC connected to the secure zone.



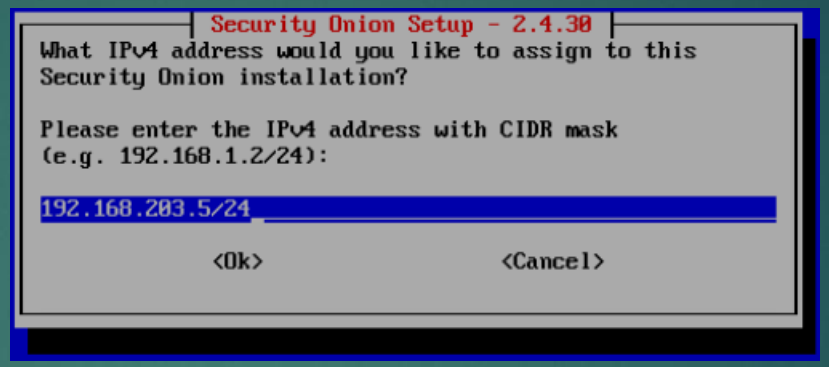


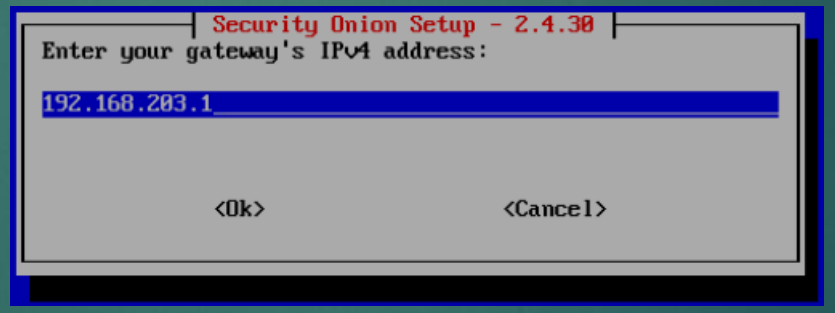
Security Onion Configuration

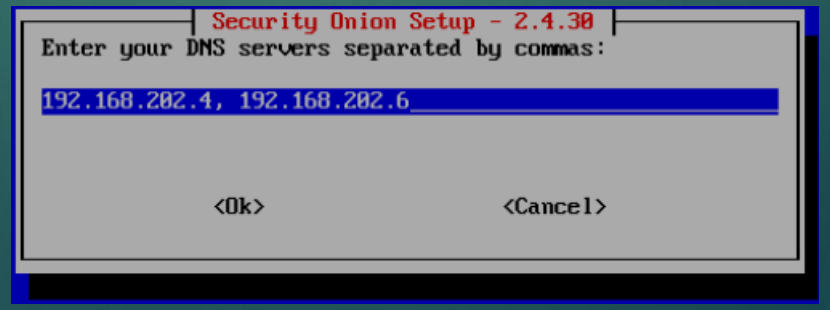
Enter the IP Address of the Security Onion installation

Enter the gateway to the Secure zone where we installed the Security Onion installation

Enter the DNS server address. (in this case we entered the addresses to the DHCP servers we installed in an earlier project





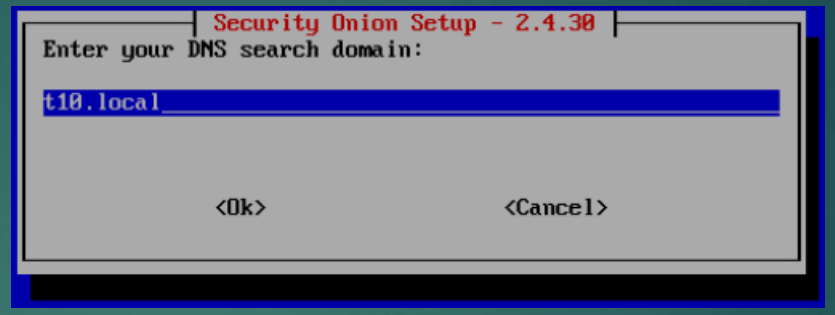


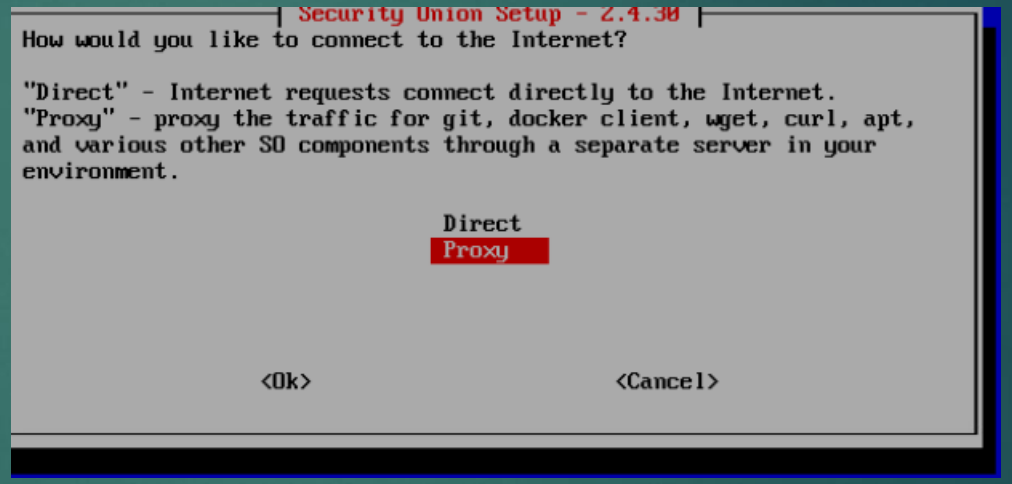
Security Onion Configuration

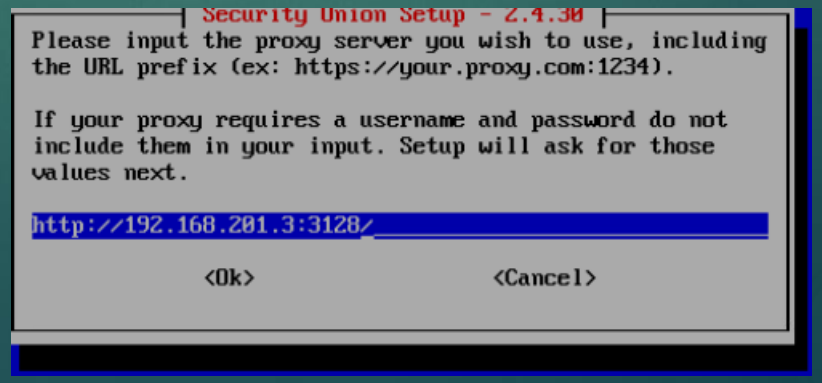
Enter the DNS search domain

In this example our Secure zone is connected by proxy through the DMZ to the internet, when prompted as to how we would like to connect to the internet we will select proxy

Input the proxy server we will use





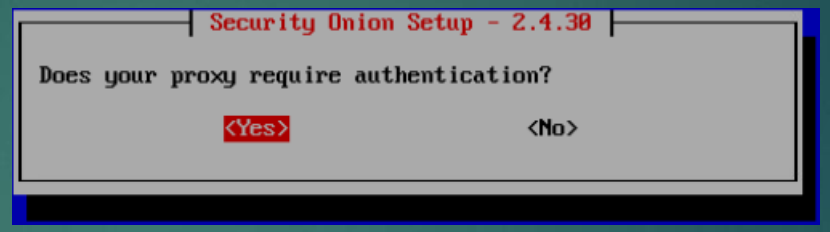


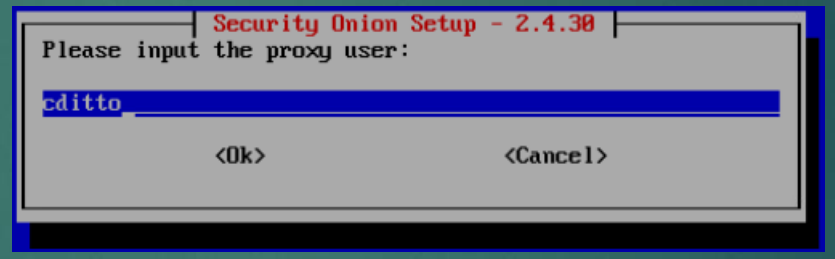
Security Onion Configuration

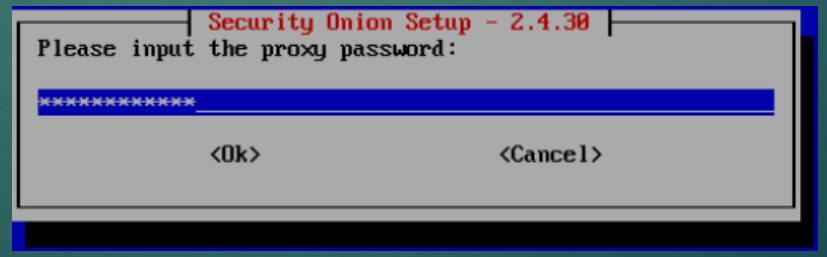
Set proxy authentication

Unlike the example shown, the proxy does not need authentication

Click “No”



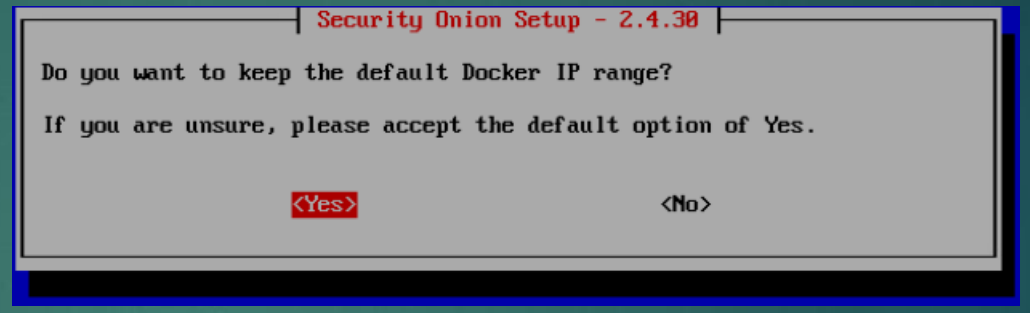


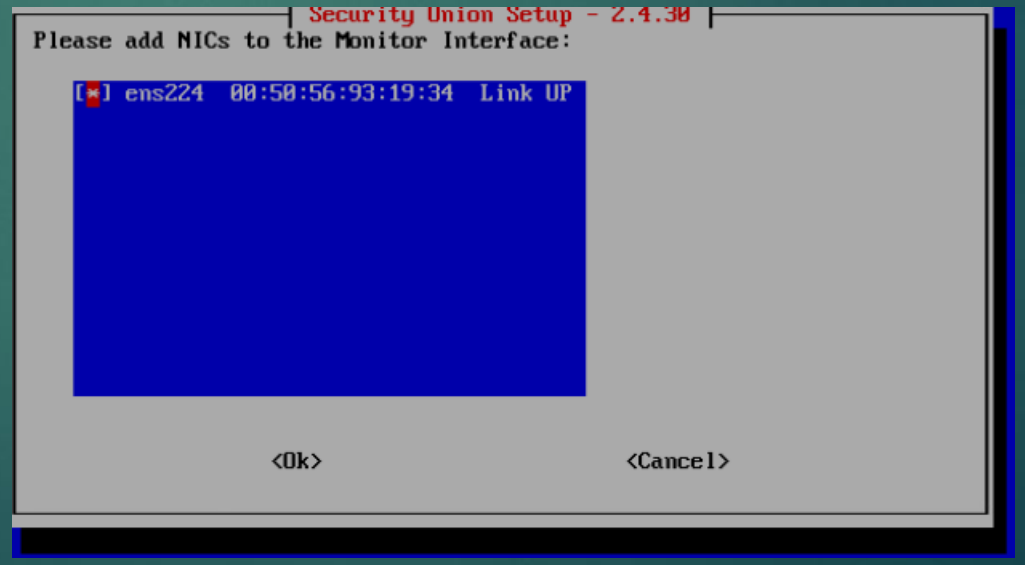


Security Onion Configuration

When asked if you want to keep the default Docker IP range, select the default "Yes"

Select the Monitor interface NIC by pressing the space bar then select "OK"



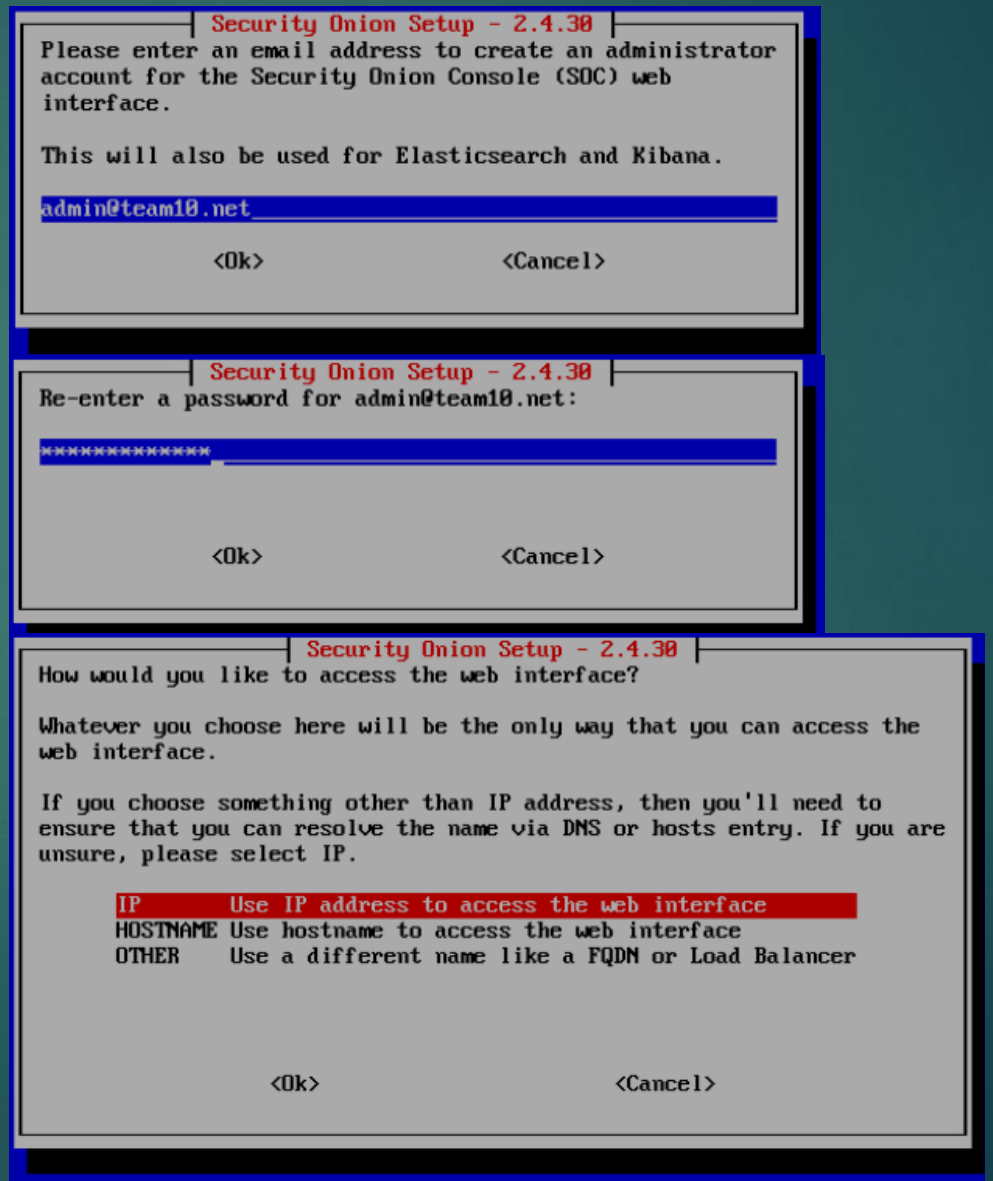


Security Onion Configuration

For this example we will use a ficticious email address to create the administrator account for use in the Security Onion Console web interface

Enter and re-enter a password for the administrator account

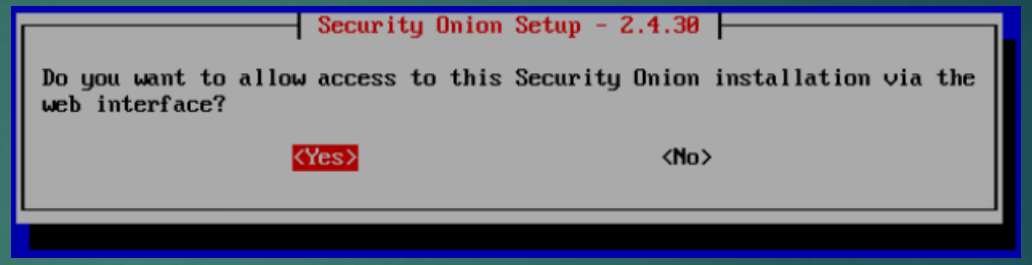
When asked how you would like to access the web interface we will choose IP

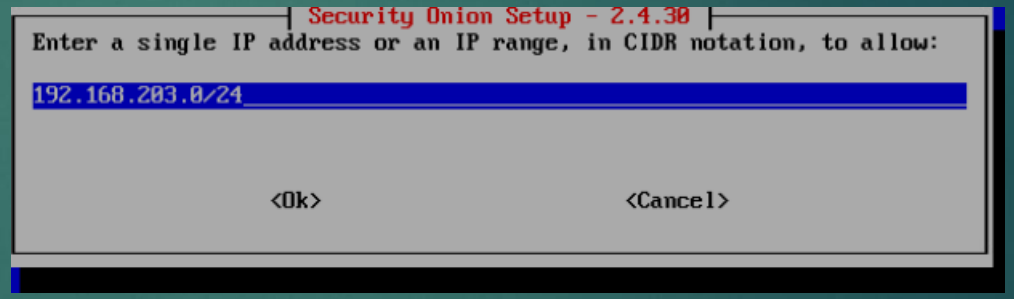


Security Onion Configuration

When asked if you want to allow access to this Security Onion installation via the web interface, select "Yes"

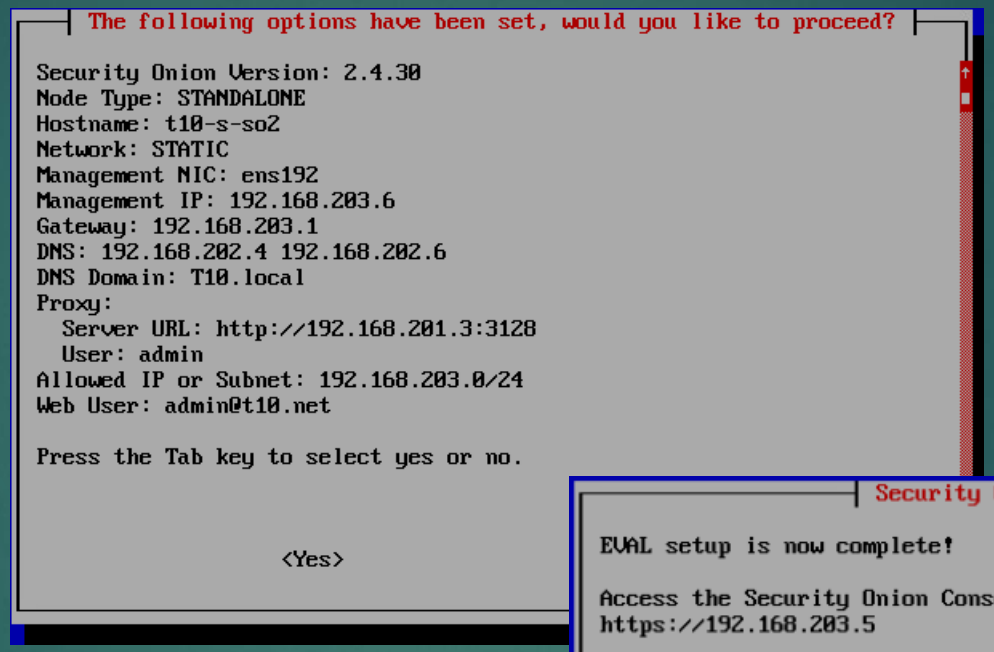
For this example, when asked to enter a single address of IP range, in CIDR notation, we will enter the IP range of the Secure zone since there is little risk of a breach in that zone.

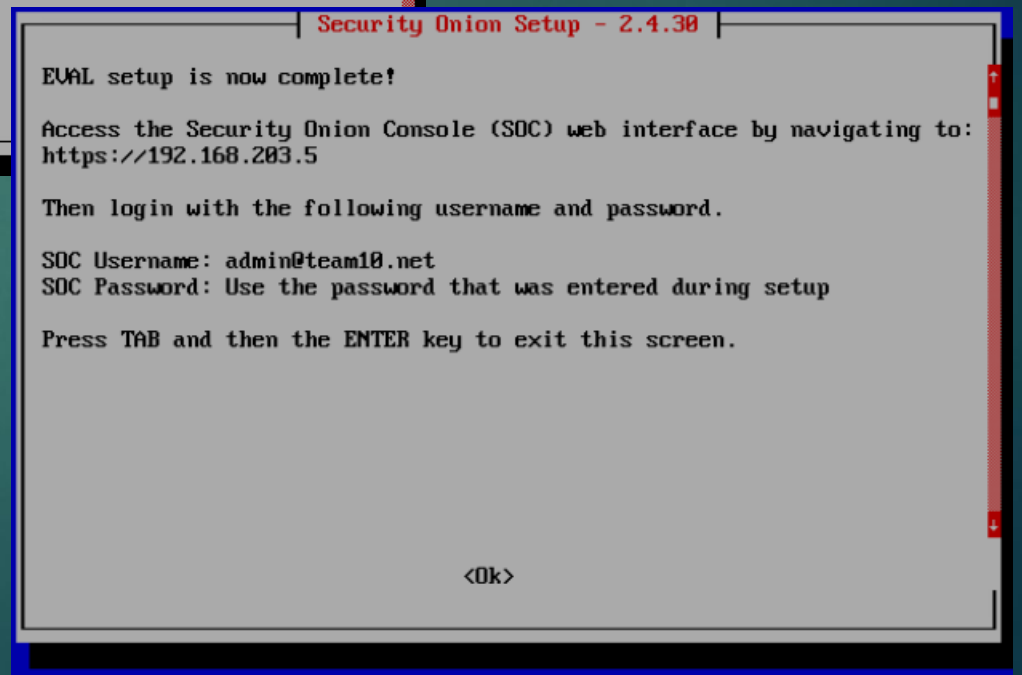




Security Onion Configuration

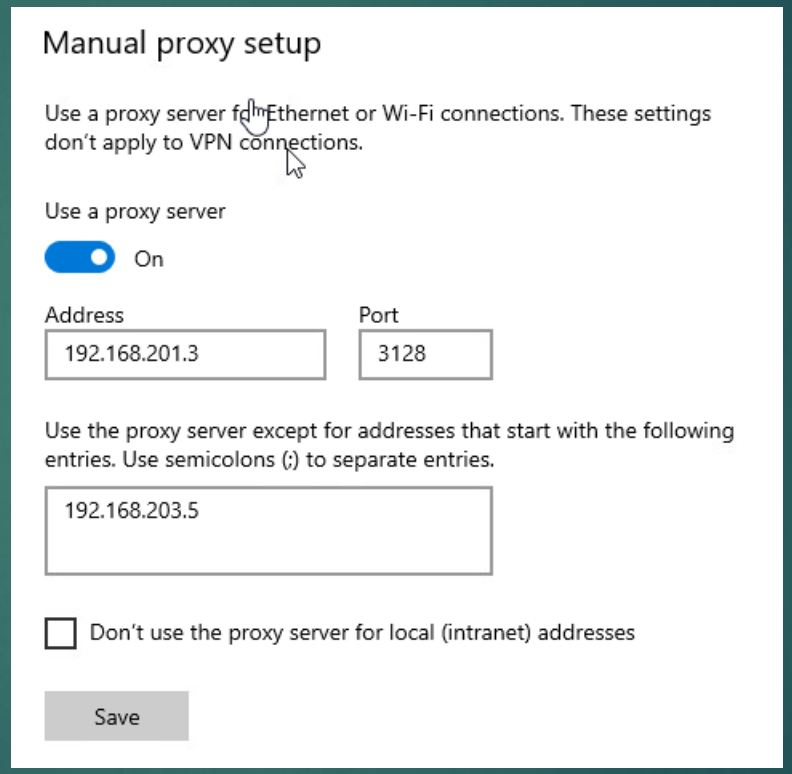
Review the Security Onion configuration and select "Yes" then select "Ok"





Security Onion Configuration

On the machine that we will be using to manage the Security Onion web interface, enter the IP address of the SO machine into the proxy server settings as shown in the following example



Security Onion SSH Login

We will now access the Security Onion console through SSH

Before we move on to this next step it is important that we reach out to our instructor as the nodes we need to connect to and from the Secure zone and the DMZ must be the same and the instructor has permissions to change them

Open CMD through the machine in which we are managing Secure Onion

Enter the command:
ssh –l admin 192.168.203.5

When prompted enter the administrators password

We should now have access to the SOC through SSH

```
admin@t10-s-so:~
admin@team10.net@192.168.203.5's password:
Permission denied, please try again.
admin@team10.net@192.168.203.5's password:
Connection reset by 192.168.203.5 port 22

C:\Users\Administrator.T10>ssh admin@192.168.203.5
##########################################
##########################################
###                                    ###
###   UNAUTHORIZED ACCESS PROHIBITED   ###
###                                    ###
##########################################
##########################################
admin@192.168.203.5's password:


Access the Security Onion web interface at https://192.168.203.5


*****************************************************************************************
* The following nodes in your Security Onion grid may need to be restarted due to package updates. *
* If the node has already been patched, restarted and been up for less than 15 minutes, then it    *
* may not have updated it's restart_needed status yet. This will cause it to be listed below, even *
* if it has already been restarted. This feature will be improved in the future.                  *
*****************************************************************************************

    t10-s-so_eval

Last login: Wed Feb 28 13:47:18 2024
[admin@t10-s-so ~]$
```

Security Onion SSH

Enter the command:
Ip address |less

You should see our NICs ens192 which is our Secure zone and ens224 which is our DMZ

Make sure that our DMZ connection ens224 is set to promiscuous mode

```
admin@t10-s-so:~
noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:93:37:be brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.203.5/24 brd 192.168.203.255 scope global noprefixroute ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe93:37be/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens224: <BROADCAST,NOARP,PROMISC,SLAVE,UP,LOWER_UP> mtu 9000 qdisc mq master bond0 state UP group default qlen 1000
    link/ether 00:50:56:93:19:34 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
4: bond0: <BROADCAST,MULTICAST,PROMISC,MASTER,UP,LOWER_UP> mtu 9000 qdisc noqueue state UP group default qlen 1000
    link/ether 00:50:56:93:19:34 brd ff:ff:ff:ff:ff:ff
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:c1:66:b2:62 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/24 brd 172.17.0.255 scope global docker0
       valid_lft forever preferred_lft forever
6: sobridge: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:62:88:66:54 brd ff:ff:ff:ff:ff:ff
    inet 172.17.1.1/24 brd 172.17.1.255 scope global sobridge
       valid_lft forever preferred_lft forever
8: veth9863580@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether 7a:92:a7:56:1f:0c brd ff:ff:ff:ff:ff:ff link-netnsid 0
10: vethd5bab89@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether 7a:3c:94:d0:38:91 brd ff:ff:ff:ff:ff:ff link-netnsid 1
12: vethd46f6b2@if11: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether c6:5f:09:1e:47:62 brd ff:ff:ff:ff:ff:ff link-netnsid 2
14: veth2483f2c@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether fe:5c:3d:e0:6d:1c brd ff:ff:ff:ff:ff:ff link-netnsid 3
18: veth73fa99a@if17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether 56:6d:24:d2:ca:d9 brd ff:ff:ff:ff:ff:ff link-netnsid 5
20: veth97b3028@if19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether 66:4f:2e:74:9c:0d brd ff:ff:ff:ff:ff:ff link-netnsid 6
24: veth2cf7054@if23: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether 9e:32:92:63:1a:78 brd ff:ff:ff:ff:ff:ff link-netnsid 7
26: veth4e20bb0@if25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether 9a:8e:f3:ba:5f:89 brd ff:ff:ff:ff:ff:ff link-netnsid 8
28: veth0bfc142@if27: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether c6:bd:ed:bc:8e:0e brd ff:ff:ff:ff:ff:ff link-netnsid 9
42: vethc942600@if41: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether 8e:88:1e:1c:e4:e1 brd ff:ff:ff:ff:ff:ff link-netnsid 16
44: vethe649cd7@if43: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether b2:d9:32:7a:bd:83 brd ff:ff:ff:ff:ff:ff link-netnsid 17
50: veth7278c6e@if49: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
    link/ether 6a:9a:54:b4:3d:66 brd ff:ff:ff:ff:ff:ff link-netnsid 10
52: veth01397a6@if51: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master sobridge state UP group default
:
```

Security Onion SSH DMZ Traffic Test

Now enter the command:
Sudo tcpdump –nn –I ens224
the –nn turns off name lookups for IP addresses and port numbers and the –i option specifies the name of the interface we want to capture

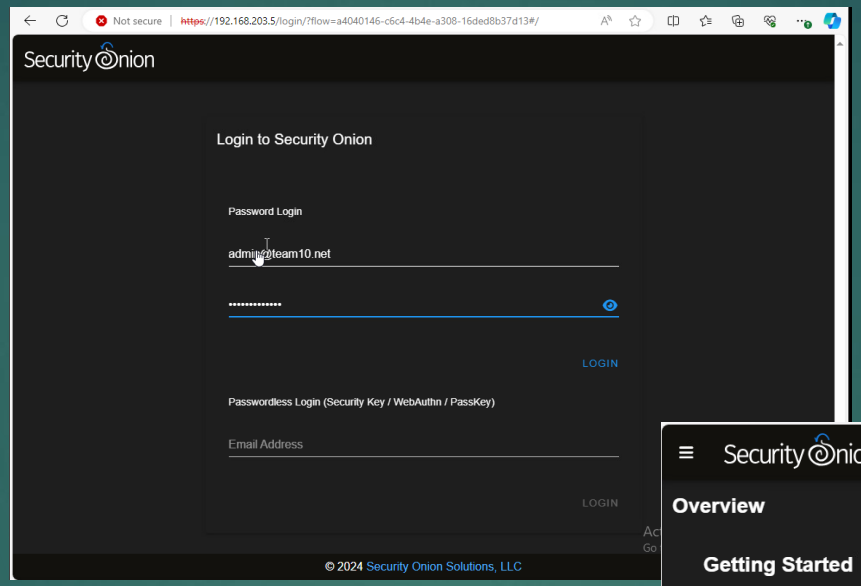
The output of this command should look something like the example shown

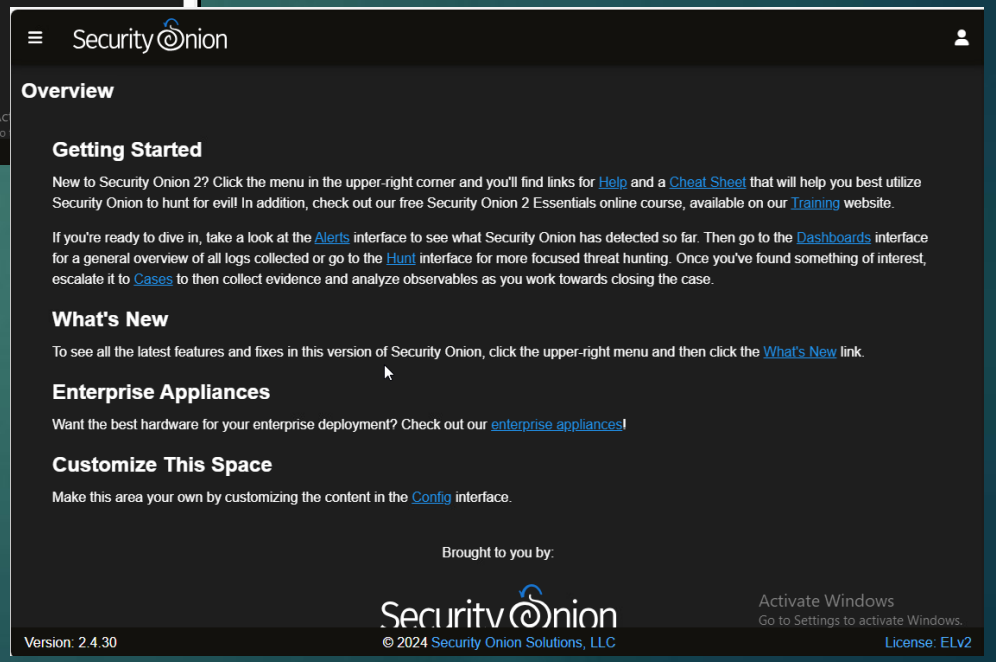
```
16:39:03.912616 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [.], ack 1, win 229, options [nop,nop,TS val 2827147354 ecr 4282267901], length 0
16:39:03.913065 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [P.], seq 1:518, ack 1, win 229, options [nop,nop,TS val 2827147355 ecr 4282267901], length 517
16:39:03.920581 IP 192.168.202.4.53 > 192.168.201.9.47447: 20701 2/0/0 AAAA 2a06:98c1:52::4, AAAA 2803:f800:53::4 (100)
16:39:03.920675 IP 192.168.202.4.53 > 192.168.201.9.47447: 28639 2/0/0 A 172.64.41.4, A 162.159.61.4 (76)
16:39:03.935912 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [.], ack 518, win 261, options [nop,nop,TS val 4282267924 ecr 2827147355], length 0
16:39:03.943140 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [P.], seq 1:157, ack 518, win 261, options [nop,nop,TS val 4282267931 ecr 2827147355], length 156
16:39:03.943191 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [.], ack 157, win 237, options [nop,nop,TS val 2827147385 ecr 4282267931], length 0
16:39:03.943683 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [P.], seq 518:569, ack 157, win 237, options [nop,nop,TS val 2827147385 ecr 4282267931], length 51
16:39:03.948867 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [P.], seq 569:746, ack 157, win 237, options [nop,nop,TS val 2827147391 ecr 4282267931], length 177
16:39:03.948880 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [P.], seq 746:1114, ack 157, win 237, options [nop,nop,TS val 2827147391 ecr 4282267931], length 368
16:39:03.948915 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [.], seq 1114:2514, ack 157, win 237, options [nop,nop,TS val 2827147391 ecr 4282267931], length 1400
16:39:03.948918 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [.], seq 2514:3914, ack 157, win 237, options [nop,nop,TS val 2827147391 ecr 4282267931], length 1400
16:39:03.948945 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [P.], seq 3914:3954, ack 157, win 237, options [nop,nop,TS val 2827147391 ecr 4282267931], length 40
16:39:03.966390 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [P.], seq 157:226, ack 569, win 261, options [nop,nop,TS val 4282267954 ecr 2827147385], length 69
16:39:03.966477 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [P.], seq 3954:3992, ack 226, win 237, options [nop,nop,TS val 2827147408 ecr 4282267954], length 38
16:39:03.970998 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [.], ack 1114, win 269, options [nop,nop,TS val 4282267960 ecr 2827147391], length 0
16:39:03.971238 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [.], ack 3914, win 291, options [nop,nop,TS val 4282267960 ecr 2827147391], length 0
16:39:03.971752 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [P.], seq 226:264, ack 3954, win 291, options [nop,nop,TS val 4282267960 ecr 2827147391], length 38
16:39:03.993820 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [.], ack 3992, win 291, options [nop,nop,TS val 4282267982 ecr 2827147408], length 0
16:39:04.011878 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [.], ack 264, win 237, options [nop,nop,TS val 2827147454 ecr 4282267960], length 0
16:39:04.038714 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [P.], seq 264:720, ack 3992, win 291, options [nop,nop,TS val 4282268027 ecr 2827147454], length 456
16:39:04.038764 IP 192.168.201.9.46810 > 34.120.208.123.443: Flags [.], ack 720, win 245, options [nop,nop,TS val 2827147480 ecr 4282268027], length 0
16:39:04.038924 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [P.], seq 720:1023, ack 3992, win 291, options [nop,nop,TS val 4282268027 ecr 2827147454], length 303
16:39:04.038925 IP 34.120.208.123.443 > 192.168.201.9.46810: Flags [P.], seq 1023:1069, ack 3992, win 291, options [nop,nop,TS val 4282268027 ecr 2827147454], length 46
```

Security Onion Login

Open the Security Onion web interface by entering the IP address of the Security Onion machine into the web browser of the machine we will use to manage the Security Onion installation

Enter the administrator credentials to the web interface





Security Onion Ruleset Change

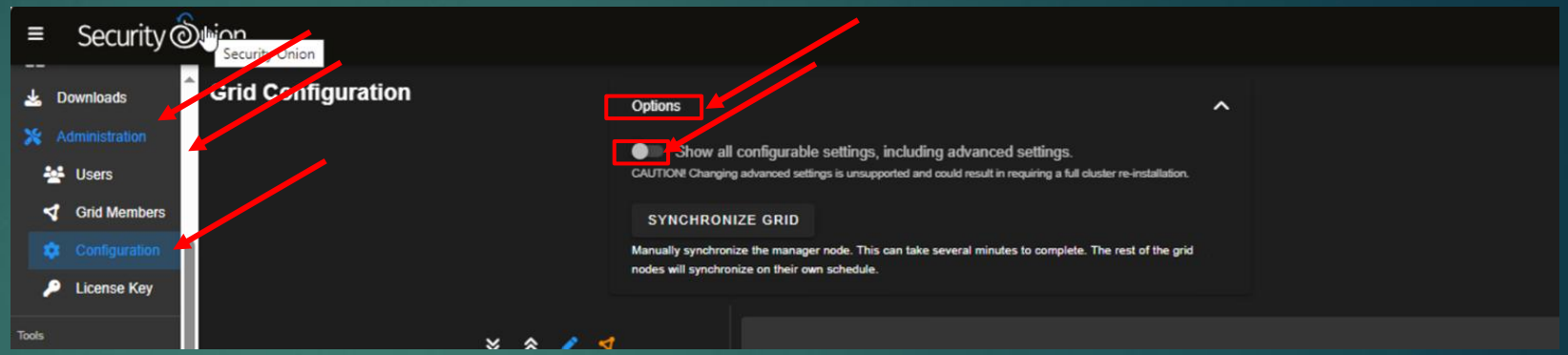
We will now make a rule and test whether our alert system is working for DMZ traffic

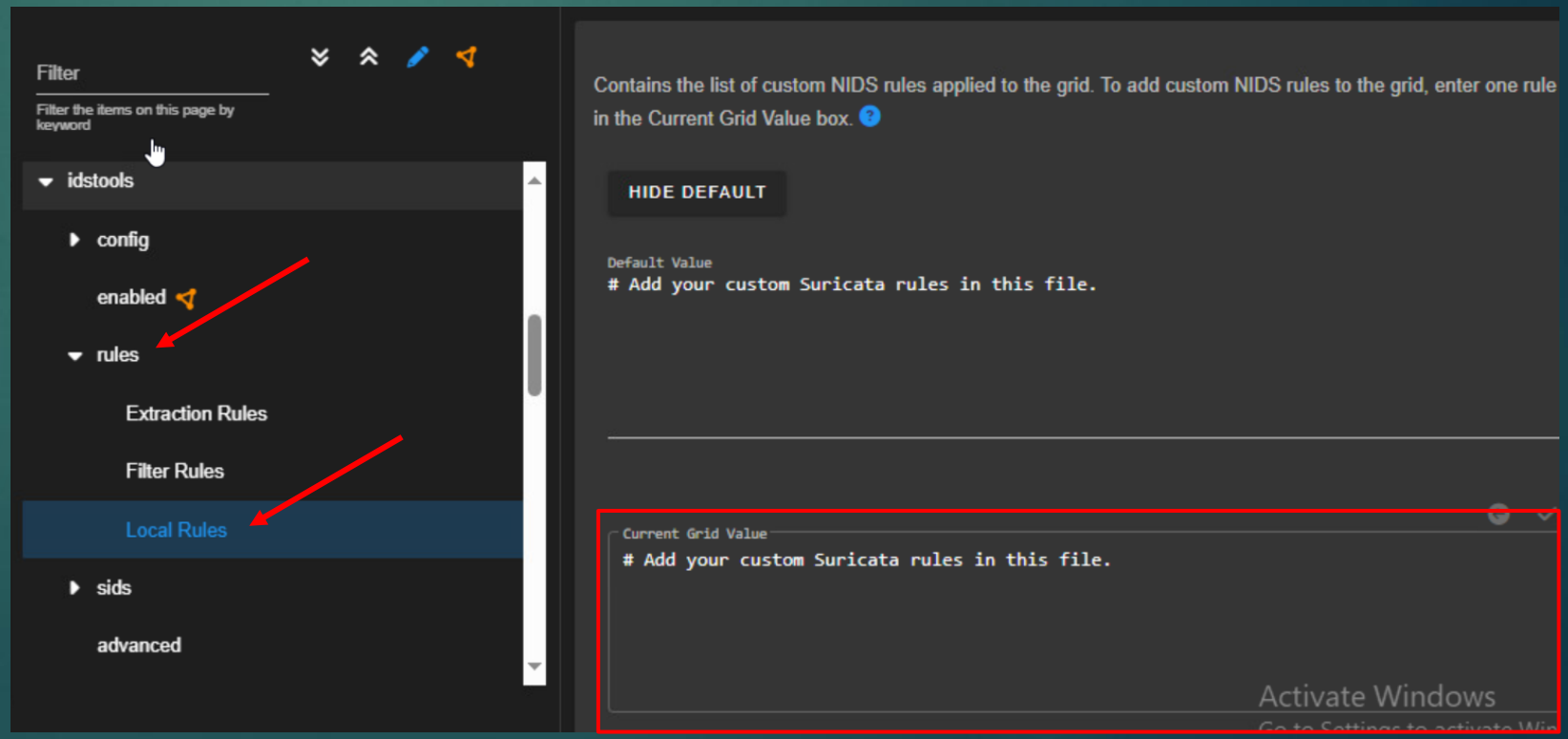
Select “administration” and then “configuration”

Select the “Options” dropdown at the top of the page and move the slider where it says “Show all configurable settings, including advanced settings”

Select “idstools” and then from the dropdown menu select “rules” then “Local Rules

We will now enter our rule into the “Current Grid Value” box at the bottom right of the page





Security Onion Ruleset Change

Add the rule shown in the example

This rule will alert Security Onion whenever it detect the word Viking over the DMZ network

The rule can also be set through the command line using vim or nano to create the rule in the local.rules file. The path to the local rules file is shown in the example

```
Current Grid Value
# Add your custom Suricata rules in this file.
alert icmp any any -> 192.168.201.0/24 any (msg:"viking ping";
itype:8; content:"viking"; sid:7001001; rev:1;)
```

```
[admin@t10-s-so ~]$ vi /opt/so/rules/nids/local.rules
```

```
# Add your custom Suricata rules in this file.
alert icmp any any -> 192.168.201.0/24 any (msg:"viking ping"; itype:8; content:"viking"; sid:7001001; rev:1;)
~
```

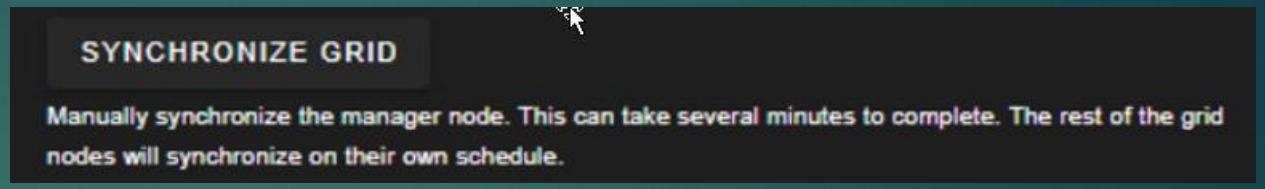
Security Onion Ruleset Change

The new rule will take around 15 minutes to commit but the process can be expedited using either the "SYNCHRONIZE GRID" selection located under the "Options" menu dropdown, or by entering the command:
sudo so-rule-update

Next enter the command:
sudo salt-call state.highstate

These commands took around 4 minutes to commit

To check the rule use the command:
sudo so-status




```
     Result: True
    Comment: Updated times on file /opt/so/log/salt/lasthighstate
    Started: 19:04:37.285506
   Duration: 2.08 ms
    Changes:
             ----------
             touched:
                 /opt/so/log/salt/lasthighstate
----------
          ID: salt_master_service
    Function: service.running
        Name: salt-master
      Result: True
     Comment: The service salt-master is already running
     Started: 19:04:37.294378
    Duration: 39.719 ms
     Changes:
----------
          ID: salt_minion_service
    Function: service.running
        Name: salt-minion
      Result: True
     Comment: The service salt-minion is already running
     Started: 19:04:37.340554
    Duration: 41.711 ms
     Changes:

Summary for local
------------
Succeeded: 759 (changed=41)
Failed:      0
------------
Total states run:     759
Total run time:   120.944 s
[admin@t10-s-so ~]$
```

```
admin@t10-s-so:~
[admin@t10-s-so ~]$ sudo so-status

                    Security Onion Status
                 Container | Status  |                Details
------------------------------------------------------------------
                so-curator | running |          Up About an hour
           so-dockerregistry | running |                Up 2 hours
               so-elastalert | running |          Up About an hour
             so-elastic-fleet | running |          Up About an hour
so-elastic-fleet-package-registry | running | Up About an hour (healthy)
           so-elasticsearch | running |          Up About an hour
                so-idstools | running |          Up About an hour
                so-influxdb | running | Up About an hour (healthy)
                  so-kibana | running |          Up About an hour
                  so-kratos | running |                Up 2 hours
                   so-mysql | running | Up About an hour (healthy)
                   so-nginx | running | Up About an hour (healthy)
                so-playbook | running |          Up About an hour
               so-sensoroni | running |          Up About an hour
                     so-soc | running |          Up About an hour
               so-soctopus | running |          Up About an hour
                  so-steno | running |          Up About an hour
           so-strelka-backend | running |          Up About an hour
       so-strelka-coordinator | running |          Up About an hour
        so-strelka-filestream | running |          Up About an hour
          so-strelka-frontend | running |          Up About an hour
        so-strelka-gatekeeper | running |          Up About an hour
           so-strelka-manager | running |          Up About an hour
                 so-suricata | running |             Up 28 minutes
                 so-telegraf | running |          Up About an hour
                    so-zeek | running | Up About an hour (healthy)

✔This onion is ready to make your adversaries cry!
```

Security Onion Ruleset Change

We will now ping the DMX machine using the command:
Ping –c 4 192.168.201.9 –p 76696b696e67 the last set of digits being the word viking in hexadecimal notation

Now we can go back to our web interface > Alerts and see our rule viking ping has caused Security Onion to alert.

```
cgerez@T10-D-AL3:~
File  Edit  View  Search  Terminal  Help
[cgerez@T10-D-AL3 ~]$ ping -c 4 192.168.201.5 -p 76696b696e67
PATTERN: 0x76696b696e67
PING 192.168.201.5 (192.168.201.5) 56(84) bytes of data.
64 bytes from 192.168.201.5: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from 192.168.201.5: icmp_seq=2 ttl=64 time=0.072 ms
64 bytes from 192.168.201.5: icmp_seq=3 ttl=64 time=0.072 ms
64 bytes from 192.168.201.5: icmp_seq=4 ttl=64 time=0.069 ms

--- 192.168.201.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3090ms
rtt min/avg/max/mdev = 0.056/0.067/0.072/0.008 ms
```