HY455 Cyber Security
Assignment 4
Chris Papastamos (csd4569)

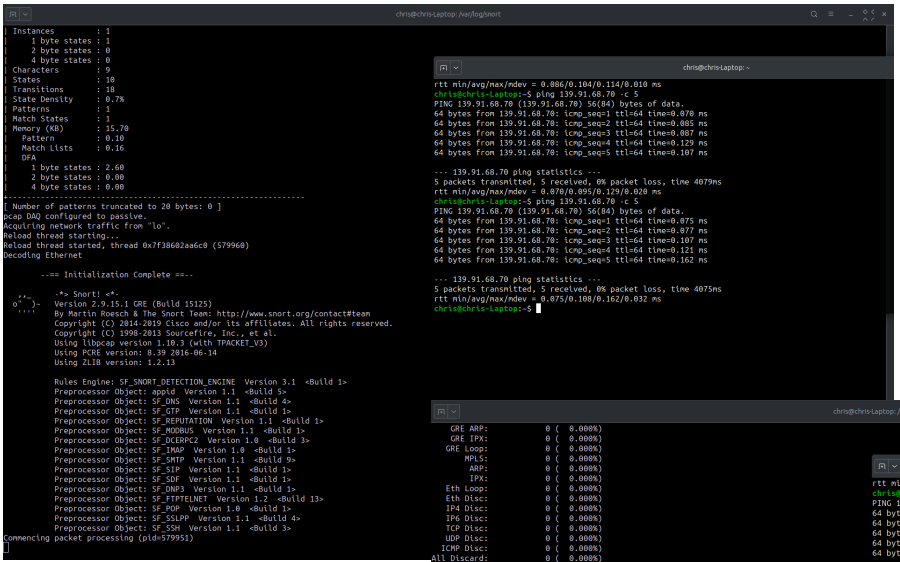
Question 1

For the rules described in the assignment I created the following 5 rules:

- log icmp any any -> 139.91.68.70 any (msg:"Captured ICMP packet destined to the host";itype:8;sid:1000001;rev:1;)
- log tcp any any -> any 7000:7999 (msg:"Captured FIN packet destined to the host";flags:F;sid:1000002;rev:1;)
- alert tcp any any -> any any (msg:"FIN and SYN flags detected at the same time!";flags:FS+;sid:1000003;rev:1;)
- log tcp any any -> any 21 (msg:"Root FTP login detected!";content:"USER root";nocase;sid:1000004;rev:1;)
- log tcp any any -> any 22 (detection_filter:track by_src, count 3, seconds 60;sid:1000005;rev:1;)

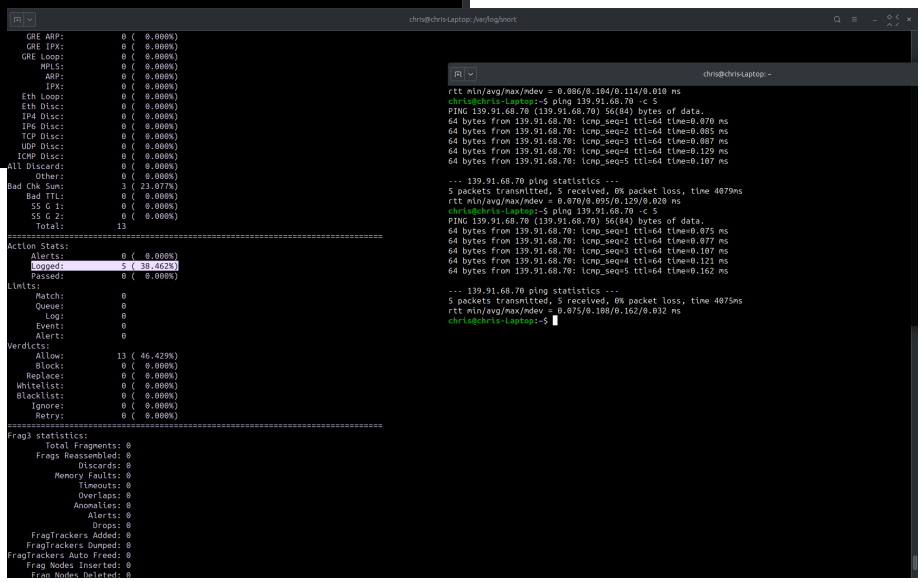
For demonstrating these rules I will run snort default configuration file while including only the local.rules file. In this file I will append my rules mentioned above

For demonstrating the ICMP request rule I will ping my own IP address and will run snort on the lo interface:



First up I fired up snort and pinged my own IP address demanding ping to send only 5 packets

After the ping is finised, we can observe that snort logged 5 packets (wich are the ICMP request packets sent by ping



```
chris@chris-Laptop:/var/log/snort$ sudo -i
root@chris-Laptop:~# cd /var/log/snort/139.91.68.70/
root@chris-Laptop:/var/log/snort/139.91.68.70# cat ICMP_ECHO
[**] Captured ICMP packet destined to the host [**]
04/27-14:14:21.619609 139.91.68.70 -> 139.91.68.70
ICMP TTL:64 TOS:0x0 ID:39029 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:841   Seq:1  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] Captured ICMP packet destined to the host [**]
04/27-14:14:22.651348 139.91.68.70 -> 139.91.68.70
ICMP TTL:64 TOS:0x0 ID:39169 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:841   Seq:2  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] Captured ICMP packet destined to the host [**]
04/27-14:14:23.675361 139.91.68.70 -> 139.91.68.70
ICMP TTL:64 TOS:0x0 ID:39251 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:841   Seq:3  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] Captured ICMP packet destined to the host [**]
04/27-14:14:24.699399 139.91.68.70 -> 139.91.68.70
ICMP TTL:64 TOS:0x0 ID:39468 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:841   Seq:4  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] Captured ICMP packet destined to the host [**]
04/27-14:14:25.723353 139.91.68.70 -> 139.91.68.70
ICMP TTL:64 TOS:0x0 ID:39604 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:841   Seq:5  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Examining the log files (in ASCII as I requested from snort using "-K ascii") we can see the 5 packets logged and the message from the rule I created appearing in before each logged packet

For the demonstration of the second rule, I will use the FIN scan option of nmap in order to send FIN TCP packets to the loopback interface and then capure and log them using snort. I am going to scan ports 6000 through 7999 so that we can check if only the ports 7000-7999 are logged:

```
chris@chris-Laptop: /var/log/snort
   ICMP Disc:           0 (  0.000%)
All Discard:           0 (  0.000%)
      Other:           0 (  0.000%)
Bad Chk Sum:           1 (  0.025%)
    Bad TTL:           0 (  0.000%)
     S5 G 1:           0 (  0.000%)
     S5 G 2:           0 (  0.000%)
      Total:        4001
===============================================================================
Action Stats:
     Alerts:           0 (  0.000%)
     Logged:        1000 ( 24.994%)
     Passed:           0 (  0.000%)
Limits:
      Match:           0
      Queue:           0
        Log:           0
      Event:           0
      Alert:           0
Verdicts:
      Allow:        4001 ( 49.988%)
      Block:           0 (  0.000%)
    Replace:           0 (  0.000%)
  Whitelist:           0 (  0.000%)
  Blacklist:           0 (  0.000%)
     Ignore:           0 (  0.000%)
      Retry:           0 (  0.000%)
===============================================================================
Frag3 statistics:
        Total Fragments: 0
      Frags Reassembled: 0
               Discards: 0
          Memory Faults: 0
               Timeouts: 0
               Overlaps: 0
              Anomalies: 0
                 Alerts: 0
                  Drops: 0
     FragTrackers Added: 0
    FragTrackers Dumped: 0
FragTrackers Auto Freed: 0
    Frag Nodes Inserted: 0
     Frag Nodes Deleted: 0
===============================================================================
===============================================================================
Stream statistics:
            Total sessions: 0
              TCP sessions: 0
              UDP sessions: 0
             ICMP sessions: 0
               IP sessions: 0
                TCP Prunes: 0
                UDP Prunes: 0
               ICMP Prunes: 0
                 IP Prunes: 0
```

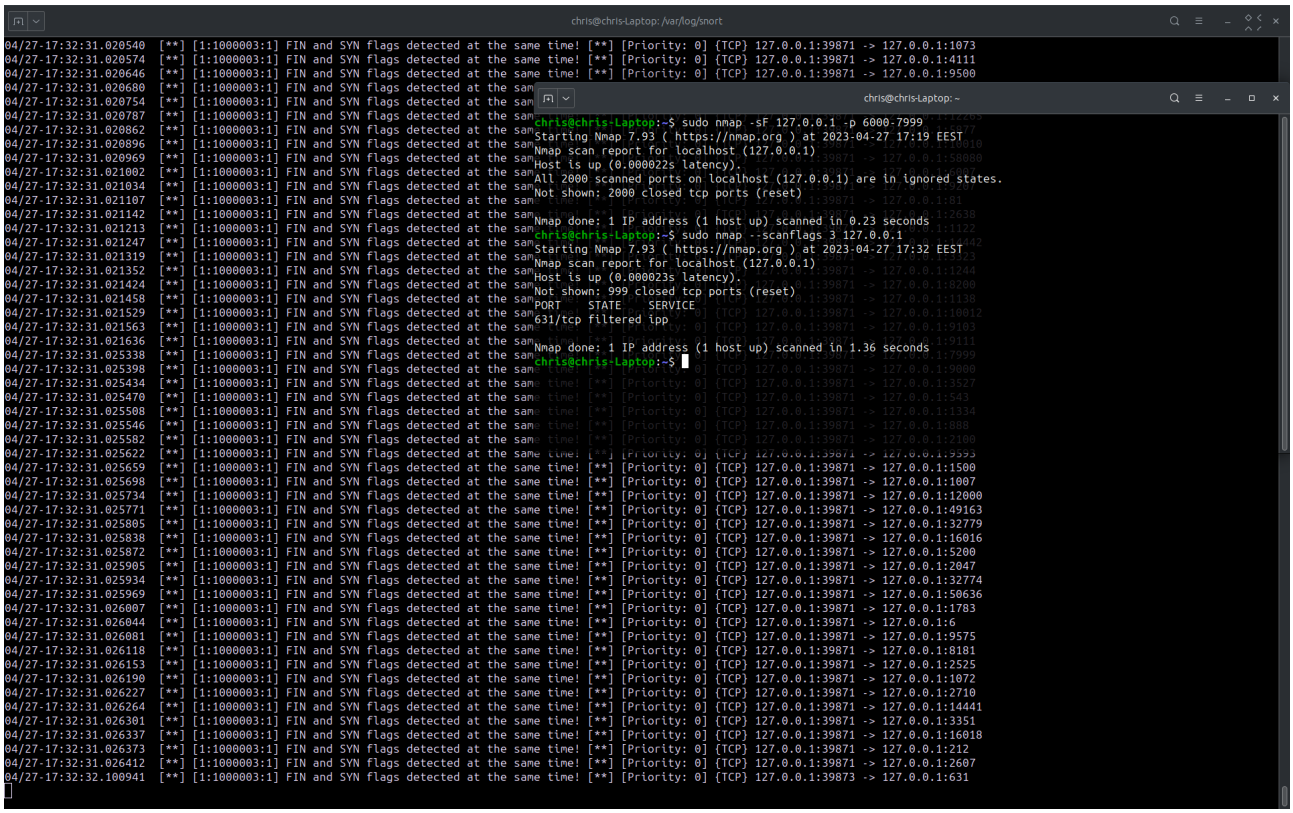
```
chris@chris-Laptop: ~
chris@chris-Laptop:~$ sudo nmap -sF 127.0.0.1 -p 6000-7999
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-27 17:19 EEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000022s latency).
All 2000 scanned ports on localhost (127.0.0.1) are in ignored states.
Not shown: 2000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
chris@chris-Laptop:~$
```

After shutting down snort, we can see that exactly 1000 entries are logged which means that the rule works as its supposed to work

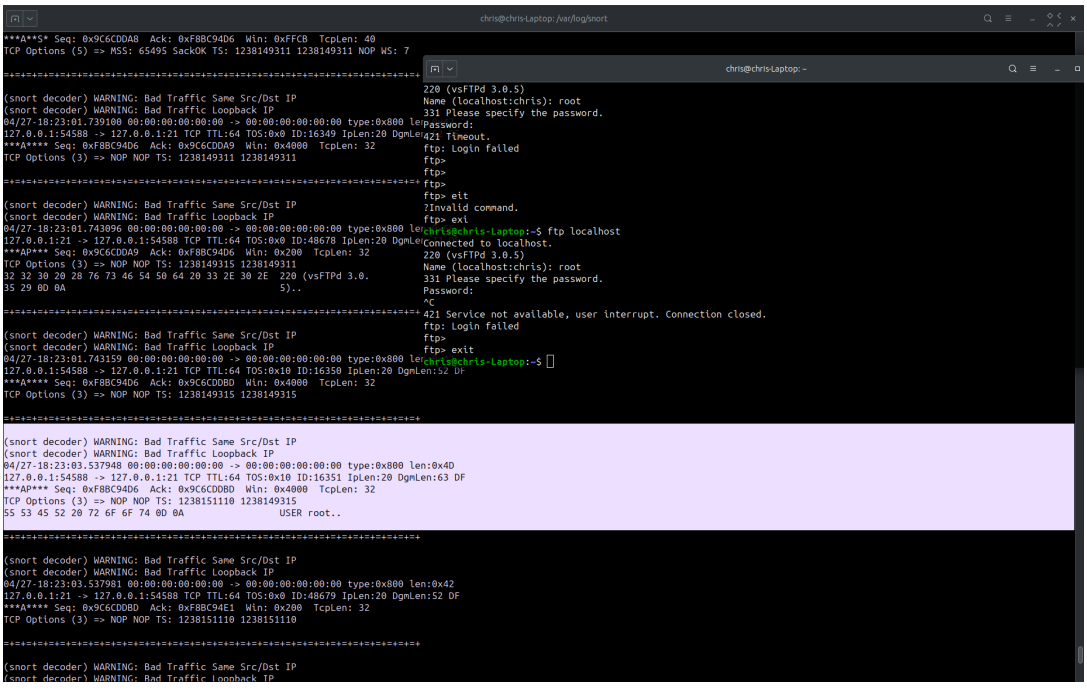
```
Snort exiting
chris@chris-Laptop:/var/log/snort$ sudo su
root@chris-Laptop:/var/log/snort# cat
127.0.0.1/          snort.alert       snort.alert.fast
root@chris-Laptop:/var/log/snort# cat 127.0.0.1/TCP\:54574-7475
[**] Captured FIN packet destined to the host [**]
04/27-17:19:16.182955 127.0.0.1:54574 -> 127.0.0.1:7475
TCP TTL:46 TOS:0x0 ID:24954 IpLen:20 DgmLen:40
*******F Seq: 0xB2352C90  Ack: 0x0  Win: 0x400  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

root@chris-Laptop:/var/log/snort# cat 127.0.0.1/TCP\:54574-7111
[**] Captured FIN packet destined to the host [**]
04/27-17:19:16.215904 127.0.0.1:54574 -> 127.0.0.1:7111
TCP TTL:47 TOS:0x0 ID:34693 IpLen:20 DgmLen:40
*******F Seq: 0xB2352C90  Ack: 0x0  Win: 0x400  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

root@chris-Laptop:/var/log/snort#
```

checking the logs we can observe that a file named TCP\:54574-x is created for every port from 7000 to 7999

For the alert rule, I will run snort using the option “-A console” which is gonna display alerts in the prompt in real time. On the other terminal I will run nmap with the option “--scanflags 3” to craft a SYN FIN packet and send it to every port.



For the last two rules the tests were not successful...
For the ftp login rule I created a dummy FTP server and then tried to login (unsuccesfully) as root user. Snort did not manage to log the packet even though I managed to find it using packet sniffer mode and displaying the packet contents on the screen. Therefore I still don’t know why the highlighted packet doesn’t trigger the log rule



Question 2

Rule	Explanation
alert icmp any any -> any any (msg:”ICMP Source Quench”;itype:4’icode:0;)	This snort rule alerts for any ICMP packets with the following fields on the header: type=4 (Source Quench) code=0 (echo reply)
alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNSEXPLOIT named overflow";flags:A+;content:"thisissometempspacefort hesockinaddrinyeahyeahiknowthisislamebutanyway whocareshorizongotitworkingsoalliscool";reference:cve,CVE-1999-0833;)	This snort rule alerts for any incoming tcp connections from the EXTERNAL_NET variable (default=! HOME_NET=any), destined to the HOME_NET variable (default = any) and to port 53 (Which is used by DNS servers), is an acknowledgment (flags:A+;) and includes the string *"thisissometempspaceforthesockinaddrinyeahyeahiknowthisislamebutanyway whocareshorizongotitworkingsoalliscool"*, When this alert is executed, the message that is displayed is "DNSEXPLOIT named overflow" and contains a CVE reference.
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"NETBIOS SMB ADMIN$access"; flow:to_server,established; content:"\\ADMIN$|00 41 3a 00|"; reference:arachnids,340; classtype:attempted-admin; sid:532; rev:4;)	This snort rule alerts any tcp from the EXTERNAL_NET to HOME_NET port 139 (used by SMB dialects that communicate over NetBIOS), triggered when the client sends packet to the server, only for an established TCP connection. The content should contain the following string “\ADMIN$|00 41 3a 00|” (the hex numbers contained in the two columns is treated as hex). The rule also has the classtype attempted-admin which gives the rule high priority as it describes an Attempted Administrator Privilege Gain. Finally it has a message to display, a referance to arachnids an sid and a revision number.
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE sparc NOOP"; content:"| a61c c013 a61c c013 a61c c013 a61c c013|"; reference:arachnids,355; classtype:shellcode-detect; sid:646; rev:4;)	This snort rule alerts for any EXTERNAL_NET connections from the SHELLCODE_PORTS (!80) to any HOME_NET port with content as described in the rule (contains hex code). The rule has a classtype of shellcode-detect which is for executable code detected and has high priority. This rule also has a message to display, a reference to arachnids, a sid and a revision number

Question 4

For the installation of Pulled Pork all I had to do is follow the instructions on the github README file. When I was thrrough with the instructions I had to make some changes to the configuration file, like inputing my oinkcode and describing which rulesets I want to download. I also had to change some path variables. After all the configuration, I ran pulledpork and the rules file was created in the rules directory of snort:

```
chris@chris-Laptop:~/pulledpork3$ sudo python3 pulledpork.py -c /usr/local/etc/pulledpork/pulledpork.conf

    https://github.com/shirkdog/pulledpork3

      _____ ____
     `----,\    )   PulledPork v3.0.0.5
      `--==\\  /    Lowcountry yellow mustard bbq sauce is the best bbq sauce. Fight me.
       `--==\\/
     .-~~~~~-.Y|\\_  Copyright (C) 2021 Noah Dietrich, Colin Grady, Michael Shirk
  @_/        /  66\_  and the PulledPork Team!
    |    \   \   _(")
     \   /-| ||'--'   Rules give me wings!
      \_\  \_\\
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Loading configuration file:  /usr/local/etc/pulledpork/pulledpork.conf
Processing Community ruleset
Preparing to modify rules by sid file
Completed processing all rulesets and local rules:
 - Collected Rules:  Rules(loaded:4031, enabled:4031, disabled:0)
 - Collected Policies:
    - Policy(name:balanced, rules:4031)
Writing rules to:  /etc/snort/rules/pulledpork.rules
Writing blocklist file to:  /usr/local/etc/lists/default.blocklist
WARNING: Unable to write blocklist:  [Errno 2] No such file or directory: '/usr/local/etc/lists/default.blocklist'
```

Due to some compatability issues (pulledpork3 is for snort3 while I ran snort 2) I didn't manage to run snort with the pulledpork.rules rules that beeing because some rules have options not compatitable with snort2 (like service). The generated pulledpork.rules file can be seen below

```
chris@chris-Laptop:~/pulledpork3$ cat /etc/snort/rules/pulledpork.rules
#--------------------------------------------------
#  Rules file created by PulledPork  at 2023.04.27-21.52.50
#
#  To Use this file: in your snort.lua, you need the following settings:
#  ips =
#  {{
#      include = "/etc/snort/rules/pulledpork.rules",
#      ...
#  }}
#
# You have chosen to enable so rules.
# To prevent errors when running snort, make sure to include
# the following command-line option:
#    --plugin-path "/usr/local/etc/so_rules/"
#
#--------------------------------------------------


alert tcp $HOME_NET 2589 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR - Dagger_1.4.0"; flow:to_client,established; content:"2|00 00 00 06 00 00 00|Drives|24 00|",depth 16; metadata:ruleset community; classtype:misc-activity; sid:105; rev
:14; )
alert tcp $EXTERNAL_NET any -> $HOME_NET 7597 ( msg:"MALWARE-BACKDOOR QAZ Worm Client Login access"; flow:to_server,established; content:"qazwsx.hsq"; metadata:ruleset community; classtype:misc-activity; sid:108; rev:12; )
alert tcp $EXTERNAL_NET any -> $HOME_NET 12345:12346 ( msg:"MALWARE-BACKDOOR netbus getinfo"; flow:to_server,established; content:"GetInfo|0D|"; metadata:ruleset community; classtype:trojan-activity; sid:110; rev:10; )
alert tcp $HOME_NET 20034 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR NetBus Pro 2.0 connection established"; flow:to_client,established; flowbits:isset,backdoor.netbus_2.connect; content:"BN|10 00 02 00|",depth 6; content:"|05 00|",dep
th 2,offset 8; metadata:ruleset community; classtype:trojan-activity; sid:115; rev:15; )
alert tcp $HOME_NET any -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR Infector.1.x"; flow:to_client,established; content:"WHATISIT",depth 9; metadata:impact_flag red,ruleset community; reference:nessus,11157; classtype:misc-activity; sid:
117; rev:17; )
alert tcp $HOME_NET 666 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR SatansBackdoor.2.0.Beta"; flow:to_client,established; content:"Remote|3A| ",depth 11,nocase; content:"You are connected to me.|0D 0A|Remote|3A| Ready for commands",dist
ance 0,nocase; metadata:ruleset community; reference:url,www.megasecurity.org/trojans/s/satanzbackdoor/SBD2.0b.html; reference:url,www3.ca.com/securityadvisor/pest/pest.aspx?id=5260; classtype:trojan-activity; sid:118; rev:12; )
alert tcp $HOME_NET 6789 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR Doly 2.0 access"; flow:to_client,established; content:"Wtzup Use",depth 32; metadata:ruleset community; classtype:misc-activity; sid:119; rev:11; )
alert tcp $EXTERNAL_NET 1000:1300 -> $HOME_NET 146 ( msg:"MALWARE-BACKDOOR Infector 1.6 Client to Server Connection Request"; flow:to_server,established; content:"FC "; metadata:ruleset community; reference:nessus,11157; classtype:misc-a
ctivity; sid:121; rev:14; )
alert tcp $HOME_NET 31785 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR HackAttack 1.20 Connect"; flow:to_client,established; content:"host"; metadata:ruleset community; classtype:misc-activity; sid:141; rev:10; )
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 ( msg:"PROTOCOL-FTP ADMw0rm ftp login attempt"; flow:to_server,established; content:"USER",nocase; content:"W0rm",distance 1,nocase; pcre:"/^USER\s+w0rm/ims"; metadata:ruleset community; servic
e:ftp; classtype:suspicious-login; sid:144; rev:16; )
alert tcp $HOME_NET 30100:30102 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR NetSphere access"; flow:to_client,established; content:"NetSphere"; metadata:ruleset community; classtype:trojan-activity; sid:146; rev:13; )
alert tcp $HOME_NET 6969 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR GateCrasher"; flow:to_client,established; content:"GateCrasher",depth 11,nocase; content:"Server",distance 0,nocase; content:"On-Line...",distance 0,nocase; pcre:"/^Ga
teCrasher\s+v\d+\.\d+\.\d+\s+Server\s+On-line\x2E\x2E\x2E/ims"; metadata:policy max-detect-ips drop,ruleset community; reference:url,www.spywareguide.com/product_show.php?id=973; classtype:trojan-activity; sid:147; rev:12; )
alert tcp $HOME_NET 5401:5402 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR BackConstruction 2.1 Connection"; flow:to_client,established; content:"C|3A 5C|"; metadata:ruleset community; classtype:misc-activity; sid:152; rev:11; )
alert tcp $EXTERNAL_NET any -> $HOME_NET 666 ( msg:"MALWARE-BACKDOOR BackConstruction 2.1 Client FTP Open Request"; flow:to_server,established; content:"FTPON"; metadata:ruleset community; classtype:misc-activity; sid:157; rev:9; )
alert tcp $HOME_NET 666 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR BackConstruction 2.1 Server FTP Open Reply"; flow:to_client,established; content:"FTP Port open"; metadata:ruleset community; classtype:misc-activity; sid:158; rev:10;
)
alert udp $EXTERNAL_NET 3344 -> $HOME_NET 3345 ( msg:"MALWARE-BACKDOOR Matrix 2.0 Client connect"; flow:to_server; content:"activate"; metadata:ruleset community; classtype:misc-activity; sid:161; rev:10; )
alert udp $EXTERNAL_NET 3345 -> $HOME_NET 3344 ( msg:"MALWARE-BACKDOOR Matrix 2.0 Server access"; flow:to_server; content:"logged in"; metadata:ruleset community; classtype:misc-activity; sid:162; rev:10; )
alert tcp $HOME_NET 5714 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR WinCrash 1.0 Server Active"; flow:stateless; flags:AS,12; content:"|84 84|"; metadata:ruleset community; classtype:misc-activity; sid:163; rev:14; )
alert tcp $EXTERNAL_NET any -> $HOME_NET 79 ( msg:"MALWARE-BACKDOOR CDK"; flow:to_server,established; content:"ypi0ca",depth 15,nocase; metadata:ruleset community; classtype:misc-activity; sid:185; rev:10; )
alert udp $HOME_NET 2140 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR DeepThroat 3.1 Server Response"; flow:to_client; content:"Ahhhh My Mouth Is Open"; metadata:ruleset community; reference:nessus,10053; classtype:trojan-activity; sid:1
95; rev:15; )
alert tcp $HOME_NET 555 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR PhaseZero Server Active on Network"; flow:to_client,established; content:"phAse zero server",depth 17,nocase; metadata:policy max-detect-ips drop,ruleset community; ref
erence:url,www.megasecurity.org/trojans/p/phasezero/PhaseZero1.0b.html; reference:url,www3.ca.com/securityadvisor/pest/pest.aspx?id=4539; classtype:trojan-activity; sid:208; rev:13; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR w00w00 attempt"; flow:to_server,established; content:"w00w00"; metadata:ruleset community; classtype:attempted-admin; sid:209; rev:9; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR attempt"; flow:to_server,established; content:"backdoor",nocase; metadata:ruleset community; classtype:attempted-admin; sid:210; rev:7; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC r00t attempt"; flow:to_server,established; content:"r00t"; metadata:ruleset community; classtype:attempted-admin; sid:211; rev:7; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC rewt attempt"; flow:to_server,established; content:"rewt"; metadata:ruleset community; classtype:attempted-admin; sid:212; rev:7; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC Linux rootkit attempt"; flow:to_server,established; content:"wh00t!"; metadata:ruleset community; reference:url,attack.mitre.org/techniques/T1014; classtype:a
ttempted-admin; sid:213; rev:9; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC Linux rootkit attempt lrkr0x"; flow:to_server,established; content:"lrkr0x"; metadata:ruleset community; reference:url,attack.mitre.org/techniques/T1014; clas
stype:attempted-admin; sid:214; rev:9; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC Linux rootkit attempt"; flow:to_server,established; content:"d13hh["nocase; metadata:ruleset community; reference:url,attack.mitre.org/techniques/T1014; clas
stype:attempted-admin; sid:215; rev:9; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC Linux rootkit satori attempt"; flow:to_server,established; content:"satori"; metadata:ruleset community; reference:url,attack.mitre.org/techniques/T1014; clas
stype:attempted-admin; sid:216; rev:12; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC sm4ck attempt"; flow:to_server,established; content:"hax0r"; metadata:ruleset community; classtype:attempted-admin; sid:217; rev:7; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC Solaris 2.5 attempt"; flow:to_server,established; content:"friday"; metadata:ruleset community; classtype:attempted-user; sid:218; rev:8; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR HidePak backdoor attempt"; flow:to_server,established; content:"StoogR"; metadata:ruleset community; classtype:misc-activity; sid:219; rev:10; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR HideSource backdoor attempt"; flow:to_server,established; content:"wank"; metadata:ruleset community; classtype:misc-activity; sid:220; rev:10; )
alert icmp $EXTERNAL_NET any -> $HOME_NET any ( msg:"PROTOCOL-ICMP TFN Probe"; icmp_id:678; itype:8; content:"1234",fast_pattern,nocase; metadata:ruleset community; reference:cve,2000-0138; classtype:attempted-dos; sid:221; rev:12; )
```