Intrusion Detection Systems and Working with Snort

4.1 Briefly answer the following questions.

- Outline the components of an Intrusion Detection System (IDS).
- Describe network-based and host-based IDSs and their differences.
- Discuss the difference between passive and reactive systems.

4.2 Let Home_Net set to 10.130.4.25 and External_Net set to! Home_Net

- Create a snort rule that alerts for FTP connection from any IP address different from Home_Net.
- Create a snort rule that alerts for "worm" in content outgoing from Home_Net.
- Does the rule you have written at the previous question raise an alert when a google search for "Internet Worm" is executed from Home_Net? Explain your answer.
- Create a snort rule that alerts for pings from External_Net.
- Explain the following rule:

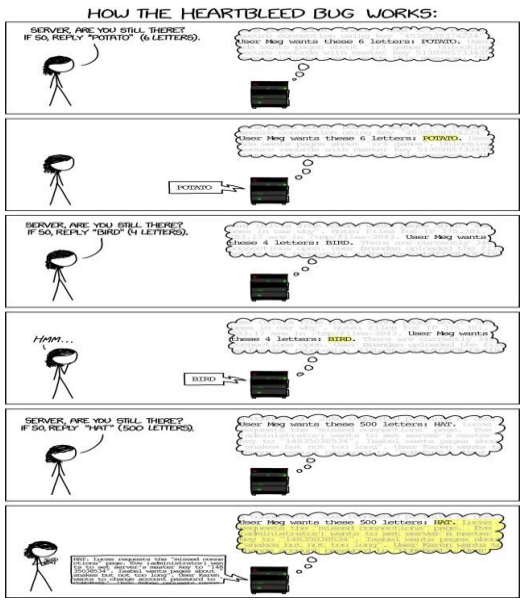
alert tcp any any -> 10.1.1.0/24 6000:6010 (msg: "X Windows Service traffic";)

4.3 State how each of the following rules work:

- alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN"; flags: F; reference: arachnids,27;)
- alert tcp $HOME_NET 23 -> $EXTERNAL_NET any (msg:"TELNET login incorrect"; content:"Login incorrect"; flags: A+; reference: arachnids,127;)
- alert icmp any any -> any any (msg:"ICMP Source Quench"; itype: 4; icode: 0;)

4.4 OpenSSL Heartbleed Vulnerability

The OpenSSL Heartbleed vulnerability is a serious weakness in OpenSSL that can lead to information disclosure, in some cases even to private key leakage. The OpenSSL Heartbleed vulnerability has been assigned the Common Vulnerabilities and Exposure (CVE) ID CVE-2014-0160. This issue occurs because the vulnerable software packages do not properly handle Heartbeat Extension packets (RFC6520). For more information, see the Sourcefire Vulnerability Research Team (VRT) analysis at http://vrt-blog.snort.org/2014/04/heartbleed-memory-disclosureupgrade.html. The following comic (from http://xkcd.com/1354/) nicely explains the Heartbleed bug.



Describe and analyze the following snort rules for Heartbleed released by Sourcefire

- alert tcp $EXTERNAL_NET any -> $HOME_NET 443 (msg:"SERVEROTHER OpenSSL TLSv1.2 heartbeat read overrun attempt"; flow:to_server,established; content:"|18 03 03|"; depth:3; dsize:>40; detection_filter:track by_src, count 3, seconds 1; metadata:policy balanced-ips drop, policy security-ips drop, service ssl; reference:cve,2014-0160; classtype:attempted-recon; sid:30513; rev:2;)

- alert tcp $HOME_NET 443 -> $EXTERNAL_NET any
- (msg:"SERVER-OTHER `TLSv1 large heartbeat response – possible ssl heartbleed attempt"; flow:to_client,established; content:"|18 03 01|"; depth:3; byte_test:2,>,128,0,relative; detection_filter:track by_dst, count 5, seconds 60; metadata:policy balanced-ips drop, policy security-ips drop, service ssl; reference:cve,2014-0160; classtype:attempted-recon; sid:30515; rev:3;)

4.6 A Computer Worm

Suppose you need to detect a computer worm that aims at causing a denial of service on some Internet hosts. The following is the tcp dump output that resulted by running the worm script against a machine on a test network:

```
0x0000  4500 0194 56ea 0000 8011 5e24 c0a8 0164      E...V.....^$...d  0x0010 ..........K....
        c0a8 0196  07b5 059a 0180 a94b  07 01 0101
0x0020 0101 0101 0101 0101 0101 0101 0101 0101      ................
0x0030 0101 0101 0101 0101 0101 0101 0101 0101      ................
0x0040 0101 0101 0101 0101 0101 0101 0101 0101      ................
0x0050 0101 0101 0101 0101 0101 0101 0101 0101      ................ ................
0x0060 0101 0101 0101 0101 0101 0101 0101 0101      ................
0x0070 0101 0101 0101 0101 0101 0101 01dc c9b0      B.........p.B.p.
0x0080 42eb 0e01 0101 0101 0101 70ae 4201 70ae      B........h...B..
0x0090 4290 9090 9090 9090 9068 dcc9 b042 b801      ...1...P..5....P
0x00a0 0101 0131 c9b1 1850 e2fd 3501 0101 0550      ..Qh.dllhel32hke
0x00b0 89e5 5168 2e64 6c6c 6865 6c33 3268 6b65      rnQhounthickChGe
0x00c0 726e 5168 6f75 6e74 6869 636b 4368 4765      tTf.llQh32.dhws2
0x00d0 7454 66b9 6c6c 5168 3332 2e64 6877 7332      _f.etQhtiref.toQ
0x00e0 5f66 b965 7451 6873 6f63 6b66 b974 6f51      hsend....B.E.P..
0x00f0 6873 656e 64be 1810 ae42 8d45 d450 ff16      P.E.P.E.P.. P....
0x0100 508d 45e0 508d 45f0 50ff 1650 be10 10ae      B....=U..Qt.....
0x0110 428b 1e8b 033d 558b ec51 7405 be1c 10ae      B....1.QQP......
0x0120 42ff 16ff d031 c951 5150 71f2 0301 049b      ......Q.E.P.E.P.
0x0130 71f2 0101 0101 518d 45cc 508b 45c0 50ff      .j.j.j...P.E.P.E
0x0140 166a 116a 026a 02ff d050 8d45 c450 8b45      .P........<a...E
0x0150 c050 ff16 89c6 09db 81f3 3c61 d9ff 8b45      ...@...........)
0x0160 b48d 0c40 8d14 88c1 e204 01c2 c1e2 0829      .......E.j..E.P1
0x0170 c28d 0490 01d8 8945 b46a 108d 45b0 5031      .Qf..x.Q.E.P.E.P
0x0180 c951 6681 f178 0151 8d45 0350 8b45 ac50 0x0190 ffd6 ebca
```

Write a snort rule to detect this worm. The rule must include the following:

1. The head of the snort signature must alert on attempts to port tcp/1045 from an external network to the “home network”.
2. The message placed in the alert must specify “Internet Worm to be stopped”.
3. The rule must search for the binary string ‘07’ (highlighted in green) in the first byte of the payload. The section highlighted in yellow is the IP header, followed by the next 8 bytes of TCP header, followed by the start of the payload and the match highlighted in green.
4. If the ‘07’ match succeeds the rule must search for the binary string “71 f2 03 01 04 9b 71 f2 01”. This match is highlighted in pink above.
5. If the rule matches the previous check, the next part of the rule must search for the text “tire” (highlighted in grey).