**Dan Otieno**

**CPE 449-01**

**Wireshark Assignment.**

**11/11/23.**

# **PART 1:**

# **Connect\_scan.pcap:**

**Write a filter to display all non-IP packets in the file.**

1. **Use the “Expression” button to find a filter that displays only IP packets.**

* **I created a filter button by pressing the “+” symbol on the right hand side of the filter field.**
* **Named the filter button “ip\_pkts”.**
* **Set filter text to “ip.addr”.**
* **Saved the button and tested it to display only IP packets.**

**A screenshot of a computer

Description automatically generated**

1. **Add a “not” operator in front of your IP filter.**

**A screenshot of a computer

Description automatically generated**

1. **How many packets are displayed using your completed filter?**

* **When I apply the view IP packets only filter, 3353 packets are displayed.**

**A screenshot of a computer

Description automatically generated**

* **When I filter out IP packets, only 2 are displayed.**

**A screen shot of a computer

Description automatically generated**

1. **Include your final Wireshark filter in your answer. !(ip.addr)**

**Write a Wireshark filter to display all packets from the scanner.**

1. **Write a Wireshark filter that shows all TCP packets with the syn flag set.**

* **Expression used: tcp.flags.syn==1**

**A screenshot of a computer

Description automatically generated**

1. **Write a second Wireshark filter that shows all TCP packets with the ack flag not set.**

* **Expression used: tcp.flags.ack==0.**

**A screenshot of a computer

Description automatically generated**

1. **Combine the above two filters with an AND statement to see the port scan as a set of SYN packets originating from the source IP.**

**A screenshot of a computer

Description automatically generated**

1. **Include the combined Wireshark filter in your answer.**

* **tcp.flags.syn==1 and tcp.flags.ack==0.**

1. **What is the IP address of the scanner? In the screenshot above, 192.168.1.100 listed as the source IP address, with the syn flag is attributed to it.**

**Write a Wireshark filter to display packets from successful TCP connection (established connections) requests.**

1. **Write a Wireshark filter to display [SYN, ACK] packets returning to the scanner from the victim. This filter should identify packets sent to the scanner IP address with SYN and ACK flags set.**

**A screenshot of a computer

Description automatically generated**

1. **Include this Wireshark filter in your answer.**

* **tcp.flags.syn==1 and tcp.flags.ack==1**

1. **How many open ports were found by the scanner?**

* **6 ports are open.**

**A group of colorful text boxes

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

# **Xmas\_scan.pcp:**

**The xmas\_scan file includes a network capture that observed one node on a network in the 192.168.1.xxx domain scanning other nodes using a Christmas scan. Answer the following questions. For a Christmas scan, the port scanner sends TCP packets with the FIN, PSH, and URG flags asserted. Write a filter to display only the aforementioned packets from the scanner. Use the tcp.flags filter category for this rule.**

* 1. **Try this filter "tcp.flags.push and tcp.flags.fin and tcp.flags.urg." Does this work? Why or why not?**
* **The command doesn’t work, all TCP packets are displayed because each packet has psh, fin and urg bits. The second screenshot shows a better filter expression, where those specific flag bits are set to 1.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* 1. **Calculate an unsigned integer that equals the expected value of the flag bytes in the TCP header when the FIN, PSH, and URG flags asserted. Tip: Type out the 8 flag bits in order in binary and then convert to decimal on a calculator. Try the filter tcp.flags == <your unsigned integer>. Does this work? Why or why not? Include this Wireshark filter in your answer.**
* **Frame 17 Flag details:**

**A screenshot of a computer

Description automatically generated**

**From the above screenshot, the hexadecimal value is 0x029, which is equivalent to 0010\_1001 (we can also determine that binary equivalent by observing the flag bit values). The integer equivalent is 41, so I typed the expression tcp.flags==41 in the filter field, as shown in the screenshot below, that command worked, because we have specified in the filter expression that we want to see the packet with flag bits set such that that all 8 bits add up to a binary value equivalent to unsigned decimal 41.**

**A screenshot of a computer

Description automatically generated**

* 1. **What is the IP address of the scanner? 192.168.1.103.**

**A screenshot of a computer

Description automatically generated**

* 1. **How many packets have the FIN, PSH, and URG flags asserted? With a filter expression of “tcp.flags.push==1 and tcp.flags.fin==1 and tcp.flags.urg==1” set, there are 1,668 packets displayed.**

**A screen shot of a computer

Description automatically generated**

# **General\_datalog.pcapng:**

**The general\_datalog file includes a network capture that observed general traffic on a computer. Answer the following questions. Write a filter to display all DNS packets. For most (or all protocols) typing the acronym for the protocol in the filter box will provide a filter to display only those packets.**

1. **Write a filter to display all DNS packets. Include this Wireshark filter in your answer.**

* **Expression used: dns**

**A screenshot of a computer

Description automatically generated**

1. **How many DNS packets are present?**

* **There are 24 DNS packets present.**

**A screen shot of a computer

Description automatically generated**

1. **Write a filter to display only DNS responses. Include this Wireshark filter in your answer.**

* **Expression used: dns.flags.response==1**

**A screenshot of a computer

Description automatically generated**

1. **How many responses are present?**

* **12 Responses are present.**

**A close up of a computer screen

Description automatically generated**

1. **Write a filter to display only the DNS response for the gstatic.com query. Include this Wireshark filter in your answer.**

* **Expression: dns.flags.response==1 and dns.qry.name matches “gstatic.com”**

**A screenshot of a computer

Description automatically generated**

1. **What is the IP address of the gstatic.com server? You need to look in the packet manually to find the IP address. Look for the “Answers” field.**

* **The IP address for gstatic.com is 172.217.4.3**

**A screenshot of a computer

Description automatically generated**

1. **Use a Who IS registry on the internet to find the street address associated with the above IP address.**

* **Lookup website used:** [**https://lookup.icann.org/en/lookup**](https://lookup.icann.org/en/lookup)
* **Street Address: 1600 Amphitheatre Parkway, Mountain View, CA, 94043, United States.**

**A screenshot of a computer

Description automatically generated**

**The ssl filter type can display many versions of SSL and TLS packets.**

1. **Write a filter to display only TLS v1.2 packets. Use Google to figure this one out. Include this Wireshark filter in your answer.**

* **Expression: tls.record.version==0x0303**

**A screenshot of a computer

Description automatically generated**

1. **How many TLSv1.2 packets are in the general\_datalog pcap file?**

* **There are 68 TLSv1.2 packets in this file.**

**A close up of a computer screen

Description automatically generated**

# **PART 2:**

**I opened the Security-Desk VM and analyzed each pcap file, starting with 10.7.**

**10.7.pcap:**

* + **After opening the file, I clicked on Analyze -->Expert Information. Next, I clicked Statistics-->Protocol Hierarchy, Statistics-->Conversations and Statistics-->Endpoints. I noted that IP 172.16.30.109 was sending mass ARP requests to a wide range of IPs across a network. The first ARP request was at packet 41. IP 172.16.30.109 initiated the TCP-3-way handshake with 172.16.10.7 at packet 3088 by sending a SYN connection establish request over port 80. I scrolled to the very bottom of the packet capture file and last saw the suspect IP in packet #6670. The rogue IP is 172.16.30.109 and the range is packets 41-6670.**

**20.0.pcap:**

* + **For this file the Expert Information details showed more than 2000 SYN connection requests and more than 2,000 RST/ACK connection reset packets. I noticed that IP 172.16.30.109 initiated SYN connection request with 172.16.20.2. Then 172.16.20.4 replied with SYN/ACK. IP 172.16.30.109 proceeded to spam 172.16.20.2 with SYN requests and 172.16.20.4 kept sending RST/ACK. I flagged IP 172.16.30.109 as malicious, with a packet range of 77-8801.**

**30.21.pcap:**

* + **Using the Expert Information menu. I noticed that IP 172.16.30.109 was sending a lot of ARP requests. Then I noticed 172.16.30.109 initiated communication with 172.6.30.21 over SSH and at packet 8186, the connection between them was terminated. I flagged IP 172.16.30.109 as malicious, packet range of 963-8186.**

# **Canvas Questions.**

**IP address for database server: 172.16.20.4**

**A computer screen with a blue screen

Description automatically generated**

# **2. For each pcap file analyzed, provide a 1 sentence description of each major action performed by the attacker.**

* **10.7.pcap:**
  + **The attacker used IP 172.16.30.109 to initiate mass ARP requests and a TCP-3-way handshake with 172.16.10.7 on port 80, between packets 41 and 6670.**
* **20.0.pcap:**
  + **The attacker used IP 172.16.30.109 to spam-flood 172.16.20.2 and 172.16.20.4 with SYN requests in attempts to crash the server.**
* **30.21.pcap:**
  + **Once again, 172.16.30.109 connected with 172.6.30.21 at packet 7601, initiated communication over SSH, and completed connection at packet 8186.**

# **3. How many total packets were sent by the attacker?**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Total packets = 2011 + 2002 + 2261 = 6,274.**

# **4. How many total packets were sent to the attacker?**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Total packets = 70 + 2,122 + 373 = 2,565.**