

Latest Version: 6.0

Question: 1

A security team is discussing lessons learned and suggesting process changes after a security breach incident. During the incident, members of the security team failed to report the abnormal system activity due to a high project workload. Additionally, when the incident was identified, the response took six hours due to management being unavailable to provide the approvals needed. Which two steps will prevent these issues from occurring in the future? (Choose two.)

- A. Introduce a priority rating for incident response workloads.
- B. Provide phishing awareness training for the full security team.
- C. Conduct a risk audit of the incident response workflow.
- D. Create an executive team delegation plan.
- E. Automate security alert timeframes with escalation triggers.

Answer: AE

Question: 2

An engineer is investigating a ticket from the accounting department in which a user discovered an unexpected application on their workstation. Several alerts are seen from the intrusion detection system of unknown outgoing internet traffic from this workstation. The engineer also notices a degraded processing capability, which complicates the analysis process. Which two actions should the engineer take? (Choose two.)

- A. Restore to a system recovery point.
- B. Replace the faulty CPU.
- C. Disconnect from the network.
- D. Format the workstation drives.
- E. Take an image of the workstation.

Answer: AE

Question: 3

Refer to the exhibit.

| No. | Time | Source | Destination | Protocol | Length | Info |
|---------|------------|-----------------|---------------|----------|--------|--|
| 2708... | 351.613329 | 167.203.102.117 | 192.168.1.159 | TCP | 174 | 15120 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2708... | 351.614781 | 52.27.161.215 | 192.168.1.159 | TCP | 174 | 15409 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2708... | 351.615356 | 209.92.25.229 | 192.168.1.159 | TCP | 174 | 15701 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2708... | 351.615473 | 149.221.46.147 | 192.168.1.159 | TCP | 174 | 15969 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2708... | 351.616366 | 192.183.44.102 | 192.168.1.159 | TCP | 174 | 16247 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2708... | 351.617248 | 152.178.159.141 | 192.168.1.159 | TCP | 174 | 16532 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.618094 | 203.98.141.133 | 192.168.1.159 | TCP | 174 | 16533 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.618857 | 115.48.48.185 | 192.168.1.159 | TCP | 174 | 16718 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.619789 | 147.29.251.74 | 192.168.1.159 | TCP | 174 | 17009 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.620622 | 29.158.7.85 | 192.168.1.159 | TCP | 174 | 17304 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.621398 | 133.119.25.131 | 192.168.1.159 | TCP | 174 | 17599 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.622245 | 89.99.115.209 | 192.168.1.159 | TCP | 174 | 17874 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.623161 | 221.19.65.45 | 192.168.1.159 | TCP | 174 | 18160 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.624003 | 124.97.107.209 | 192.168.1.159 | TCP | 174 | 18448 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |
| 2709... | 351.624765 | 140.147.97.13 | 192.168.1.159 | TCP | 174 | 18740 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment |

What should an engineer determine from this Wireshark capture of suspicious network traffic?

- A. There are signs of SYN flood attack, and the engineer should increase the backlog and recycle the oldest half-open TCP connections.
- B. There are signs of a malformed packet attack, and the engineer should limit the packet size and set a threshold of bytes as a countermeasure.
- C. There are signs of a DNS attack, and the engineer should hide the BIND version and restrict zone transfers as a countermeasure.
- D. There are signs of ARP spoofing, and the engineer should use Static ARP entries and IP address-to-MAC address mappings as a countermeasure.

Answer: A

Question: 4

Refer to the exhibit.

| Time | | Dst | port | Host | Info |
|------------|----------|----------------|------|----------------------|--|
| 2019-12-04 | 18:44... | 185.188.182.76 | 80 | ghinatronx.com | GET /edgtron/siloft.php?l=yourght6.cab |
| 2019-12-04 | 18:46... | 45.143.93.81 | 80 | bjanicki.com | GET /images/18hvXdm_2F40bg3onEOH_2/ |
| 2019-12-04 | 18:46... | 45.143.93.81 | 80 | bjanicki.com | GET /favicon.ico HTTP/1.1 |
| 2019-12-04 | 18:46... | 45.143.93.81 | 80 | bjanicki.com | GET /images/6a7GzE2PowJhysjaQ/HULhLB |
| 2019-12-04 | 18:46... | 45.143.93.81 | 80 | bjanicki.com | GET /images/aiXla28QV6duatPF_2BY9stc |
| 2019-12-04 | 18:47... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 18:48... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 18:52... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 18:57... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 19:02... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 19:07... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 19:08... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 19:13... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 19:18... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |
| 2019-12-04 | 19:19... | 194.61.1.178 | 443 | prodrigo29ibkd20.com | Client Hello |

| |
|--|
| > Frame 6: 386 bytes on wire (3088 bits), 386 bytes captured (3088 bits) |
| > Ethernet II, Src: HewlettP_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1) |
| > Internet Protocol Version 4, Src: 160.192.4.101, Dst: 185.188.182.76 |
| 0000 20 e5 2a b6 93 f1 00 08 02 1c 47 ae 08 00 45 00 * . . . G . E |

A network engineer is analyzing a Wireshark file to determine the HTTP request that caused the initial Ursnif banking Trojan binary to download. Which filter did the engineer apply to sort the Wireshark traffic logs?

- A. http.request.un matches
- B. tls.handshake.type ==1
- C. tcp.port eq 25
- D. tcp.window_size ==0

Answer: B

Reference:

<https://www.malware-traffic-analysis.net/2018/11/08/index.html>

<https://unit42.paloaltonetworks.com/wireshark-tutorial-examining-ursnif-infections/>

Question: 5

What is a concern for gathering forensics evidence in public cloud environments?

- A. High Cost: Cloud service providers typically charge high fees for allowing cloud forensics.
- B. Configuration: Implementing security zones and proper network segmentation.
- C. Timeliness: Gathering forensics evidence from cloud service providers typically requires substantial time.
- D. Multitenancy: Evidence gathering must avoid exposure of data from other tenants.

Answer: D

Reference:
publication/307871954_About_Cloud_Forensics_Challenges_and_Solutions

[https://www.researchgate.net/](https://www.researchgate.net/publication/307871954_About_Cloud_Forensics_Challenges_and_Solutions)

Question: 6

Which scripts will search a log file for the IP address of 192.168.100.100 and create an output file named parsed_host.log while printing results to the console?

- A.
- ```
import os
import re
line_regex = re.compile(r".*fwd=\"192.168.100.100\".*$")
output_filename = os.path.normpath("output/parsed_host.log")
with open(output_filename, "w") as out_file:
 out_file.write("")
with open(output_filename, "a") as out_file:
 with open("parsed_host.log", "r") as in_file:
 for line in in_file:
 if (line_regex.search(line)):
 print line
 out_file.write(line)
```
- B.
- ```
import os
import re
line_regex = re.compile(r".*fwd=\"192.168.100.100\".*$")
output_filename = os.path.normpath("output/parsed_hosts.log")
with open(output_filename, "w") as out_file:
    out_file.write("")
with open(output_filename, "a") as out_file:
    with open("test_log.log", "r") as in_file:
        for line in in_file:
            if (line_regex.search(line)):
                print line
                out_file.write(line)
```

- C.

```
import os
import re
line_regex = re.compile(r".*fwd=\\"192.168.100.10\\".*$")
output_filename = os.path.normpath( "output/parsed_host.log")
with open(output_filename, "w") as out_file:
    out_file.write("")
with open(output_filename, "a") as out_file:
    with open( "parsed_host.log", "r") as in_file:
        for line in in_file:
            if (line_regex.search(line)):
                print line
                out_file.write(line)
```
- D.

```
import os
import re
line_regex = re.compile(r".*fwd=\\"192.168.100.100\\".*$")
output_filename = os.path.normpath( "output/parsed_host.log")
with open(output_filename, "w") as out_file:
    out_file.write("")
with open(output_filename, "a") as out_file:
    with open( "test_log.log", "r") as in_file:
        for line in in_file:
            if (line_regex.search(line)):
                print line
                out_file.write(line)
```

- A. Option A
B. Option B
C. Option C
D. Option D

| |
|------------------|
| Answer: A |
|------------------|