OverTheWire Writeups:

Level 1

flag = CV1DtqXWVFXTvM2F0k09SHz0YwRINYA9

Level 3

flag = pIwrPrtPN36QITSp3EQaw936yaFoFgAB

Level 5

flag = DXjZPULLxYr17uwoI01bNLQbtFemEgo7

Level 7

flag = cvX2JJa4CFALtqS87jk27qwqGhBM9plV

Level 9

mit grep einfach die '=' suchen.

flag = truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk

Level 11

der Text ist mit ROT13 codiert - um zu decoden einfach tr (translate) verwenden: tr 'A-Za-z' 'N-ZA-Mn-za-m' dadurch, werden jeweils die einzelnen Zeichen (egal ob groß oder klein geschrieben » deshalb der erste Part 'A-Za-z' - um 13 Stellen verrückt - das würde dann dem nächsten Part entsprechen 'N-ZA-Mn-za-m').

flag = 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu

Level 13

Gegeben ist ein Private Key RSA für SSH. Auch gegeben ist, dass die flag im Verzeichnis /etc/bandit_pass/bandit14 liegt und nur vom user bandit14 gelesen werden kann. Versuchen mit dem privateKey per ssh anzumelden.

`ssh -i >filename< bandit14@host -p 2220 >> hat nicht funktioniert!`
`Lösung war, dass man statt des Hostnamen von extern localhost nimmt:`

```
ssh -i >file< bandit14@localhost
```

flag = wurde mit privatekey gelöst - das PW kann aber aus /etc/bandit_pass/bandit14 ausgelesen werden = 4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e

Level 15

man muss sich per SSL mit dem Service auf Port localhost:30001 verbinden.

openssl s_client -connect localhost:30001

dann muss man ein Password eingeben. Das Password ist das von Level 14.

flag = cluFn7wTiGryunymYOu4RcffSxQluehd

Level 17

es sind zwei Dateien gegeben:

`passwords.old und passwords.new`

gegeben ist, dass die nächste flag die einzige Zeile ist, die sich in den beiden Dateien geändert hat.

`diff passwords.old passwords.new`

gibt genau eine Zeile aus.

flag = kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd

Level 19

Mit der Datei bandit20-do kann man mit anderen Privilegien Befehle ausführen.

ls -al gibt statt dem x ein s zurück - so kann man setuid files erkennen.

mit ./bandit20-do whoami kann man sehen, welche userrechte man jetzt hat. Dann kann man das Passwort von Bandit20 einfach auslesen - ./bandit20-do cat /etc/bandit_pass/bandit20

flag = GbKksEFF4yrVs6il55v6gwY5aVje5f0j

Level 21

Gegeben ist, dass ein Programm in regelmäßigen Abständen ausgeführt wird. Dieses Programm liegt in der directory - /etc/cron.d/

Darin waren 3 Dateien abgelegt. Diese verwiesen wiederum auf drei bash Skripte in /usr/bin/cronjob_bandit22(23,24). Und diese wiederum speichern das Passwort des jewiligen Bandits im Ordner /tmp/ in einer bestimmten directory ab.

flag1 = Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI

Level 23

Auch hier geht man wieder in /etc/cron.d/ und schaut sich den Code von cronjob_bandit24 an. Dieser führt das dort angegebene Skript aus, welches immer alle Dateien in dem Verzeichnis /var/spool/bandit24 ausführt und anschließend löscht.

Wenn man sich weiter umschaut ist in dem Verzeichnis /var/spool/bandit24 ein test Ordner, in welchem sich ein bash skript befindet. In diesem ist das Password im Klartext gespeichert.

flag = UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ

Level 25

Hier wird angegeben, dass bandit26 nicht die normale bash Shell verwendet. Im home Verzeichnis von bandit24 sieht man einen sshkey. Man kann sich damit auch mit ssh bandit26@localhost -i file verbinden, jedoch wird die Verbindung sofort unterbrochen. Man kann in /etc/passwd alle vorhandenen Nutzer und

deren verwendete Shell anschauen. Dort ist eine text Datei angegeben als Shell. Wenn man diese mit cat aufruft, sieht man folgendes:

```
#!/bin/sh

export TERM=linux

more ~/text.txt
exit 0
```

Wenn man jetzt das Terminal so klein macht, dass die Textdatei nicht ganz angezeigt werden kann – bei der SSH Verbindung, dann wird die Verbindung nicht sofort beendet. Nun öffnet man mit 'v' den vim Editor. Dort gibt man dann :set shell=/bin/bash ein, damit die bash Shell verwendet wird. Dann :sh damit ruft man die Shell auf. Schon ist man angemeldet.

flag = 5czgV9L3Xx8JPOyRbXh6lQbmIOWvPT6Z

Level 28

so hier ist gegeben, dass es ein git repository gibt. Man kann aber nur im tmp Ordner schreiben. Also einen neuen Ordner machen und mit git clone ssh://bandit27-git@localhost/home/bandit27-git/repo alles herunterladen. Man bekommt einen Ordner mit einer README Datei. Darin steht das neue Passwort.

flag = 0ef186ac70e04ea33b4c1853d2526fa2

28 –> 29 man muss wieder die angegebene git repo clonen. Darin befindet sich eine Datei. Dort ist das Passwort jedoch unkenntlich gemacht. Wenn man sich in dem heruntergeladenen Ordner repo befindet, kann man mit `git log -p` kann man jedoch vorher commitete Versionen der repo sehen. Und dort ist dann auch das Passwort zu sehen.

flag = bbc96594b4e001778eee9975372716b2

Level 30

Die nächste repo enthält eine Datei. `+just an epmty file... muahaha` Mehr bekommt man mit `git show` auch nicht. Dann aber mit `git tag` findet man einen tag `secret`. Dann mit `git show secret` findet man dann das Passwort.

flag = 47e603bb428404d265f59c42920d81e5

Level 32

Hier kommt man in eine Shell, die nicht viele Befehle kennt. Mit `$0` wechselt man in die normale bash shell. Dann mit `cat /etc/bandit_pass/bandit33` einfach das Passwort abfragen.

flag = c9c3199ddf4121b10cf581a98d51caee

Level 33

kein weitere Level bis jetzt.