

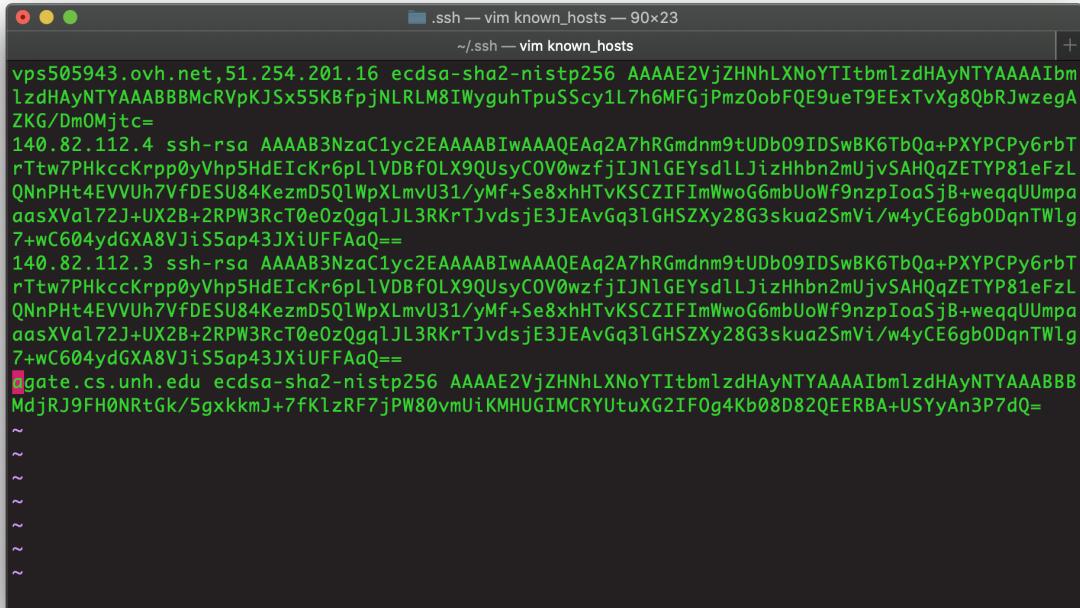
CS 827: Lab 2

Due on Wednesday, April 8, 2020

Shayan Amani (sa1149)

1 Protocol

1. As I have OpenSSH pre-installed on my machine, I have logged in to Agate through SSH. I have Agate's public key stored in the list of my known hosts already. As you it is shown below, the Agate's public key (hashed in SHA256) is existing in my local machine's `known_hosts` file:



```
.ssh — vim known_hosts — 90x23
~/.ssh — vim known_hosts +
```

```
vps505943.ovh.net,51.254.201.16 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIBm
lzdHAyNTYAAABBBMcRVpKJSx55KBfpjNLRLM8IWyguhTpuSScy1L7h6MFGjPmzOobFQE9ueT9EEExTvXg8QbRJwzegA
ZKG/Dm0Mjtc=
140.82.112.4 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq2A7hRGmdnm9tUDb09IDSwbk6TbQa+PXYPCPy6rbT
rTtw7PHkccKrpp0yVhp5HdEIcKr6pLlVDBf0LX9QUsyCOV0wzfjIJNLGEYsdllJizHhb2mUjvSAHQqZETYP81eFzL
QNnPht4EVVUh7VfDESU84KezmD5Q1WpXLmvU31/yMf+Se8xhHTvKSCZIFImWwoG6mbUoWf9nzpIoaSjB+weqqUUmPa
aasXval72J+UX2B+2RPW3RcT0e0zQgqlJL3RKrTJvdsjE3JEAvGq3lGHSZXy28G3skua2SmVi/w4yCE6gb0DqnTWlg
7+wC604ydGXA8VJiS5ap43JXiUFFAaQ==
140.82.112.3 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq2A7hRGmdnm9tUDb09IDSwbk6TbQa+PXYPCPy6rbT
rTtw7PHkccKrpp0yVhp5HdEIcKr6pLlVDBf0LX9QUsyCOV0wzfjIJNLGEYsdllJizHhb2mUjvSAHQqZETYP81eFzL
QNnPht4EVVUh7VfDESU84KezmD5Q1WpXLmvU31/yMf+Se8xhHTvKSCZIFImWwoG6mbUoWf9nzpIoaSjB+weqqUUmPa
aasXval72J+UX2B+2RPW3RcT0e0zQgqlJL3RKrTJvdsjE3JEAvGq3lGHSZXy28G3skua2SmVi/w4yCE6gb0DqnTWlg
7+wC604ydGXA8VJiS5ap43JXiUFFAaQ==
agate.cs.unh.edu ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIBmlzdHAyNTYAAABBB
MdjRJ9FH0NRtGk/5gxkkmJ+7fKlzRF7jPW80vmUiKMHUGIMCRYUtuXG2IF0g4Kb08D82QEERBA+USYyAn3P7dQ=
```

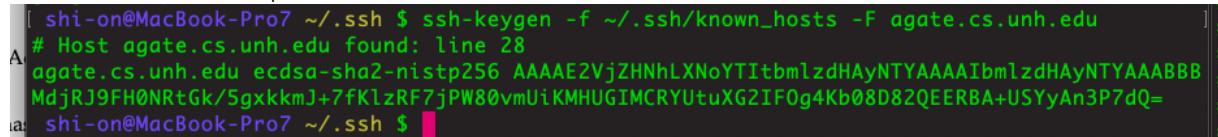
~
~
~
~
~
~

2. The following command confirms that that the Agate's public key is stored on my machine:

`ssh-keygen -f ~/.ssh/known_hosts -F agate.cs.unh.edu`

-f: file name to look up in.

-F: the host to look up for.



```
[ shi-on@MacBook-Pro7: ~/.ssh ]$ ssh-keygen -f ~/.ssh/known_hosts -F agate.cs.unh.edu
# Host agate.cs.unh.edu found: line 28
agate.cs.unh.edu ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIBmlzdHAyNTYAAABBB
MdjRJ9FH0NRtGk/5gxkkmJ+7fKlzRF7jPW80vmUiKMHUGIMCRYUtuXG2IF0g4Kb08D82QEERBA+USYyAn3P7dQ=
```

3. My SSH client queries including the **ciphers**, **MACs**, and **key-exchange algorithms** supported by **my local SSH client**:

```
1 shi-on@MacBook-Pro7 ~/.ssh $ ssh -Q cipher
2 3des-cbc
3 aes128-cbc
4 aes192-cbc
5 aes256-cbc
6 rijndael-cbc@lysator.liu.se
7 aes128-ctr
8 aes192-ctr
9 aes256-ctr
10 aes128-gcm@openssh.com
11 aes256-gcm@openssh.com
12 chacha20-poly1305@openssh.com
13 shi-on@MacBook-Pro7 ~/.ssh $ ssh -Q mac
14 hmac-sha1
15 hmac-sha1-96
16 hmac-sha2-256
17 hmac-sha2-512
18 hmac-md5
19 hmac-md5-96
20 umac-64@openssh.com
21 umac-128@openssh.com
22 hmac-sha1-etm@openssh.com
23 hmac-sha1-96-etm@openssh.com
24 hmac-sha2-256-etm@openssh.com
25 hmac-sha2-512-etm@openssh.com
26 hmac-md5-etm@openssh.com
27 hmac-md5-96-etm@openssh.com
28 umac-64-etm@openssh.com
29 umac-128-etm@openssh.com
30 shi-on@MacBook-Pro7 ~/.ssh $ ssh -Q kex
31 diffie-hellman-group1-sha1
32 diffie-hellman-group14-sha1
33 diffie-hellman-group14-sha256
34 diffie-hellman-group16-sha512
35 diffie-hellman-group18-sha512
36 diffie-hellman-group-exchange-sha1
37 diffie-hellman-group-exchange-sha256
38 ecdh-sha2-nistp256
39 ecdh-sha2-nistp384
40 ecdh-sha2-nistp521
41 curve25519-sha256
42 curve25519-sha256@libssh.org
43 sntrup4591761x25519-sha512@tinyssh.org
```

4. SSH in verbose mode:

```
1 shi-on@MacBook-Pro7 ~/.ssh $ ssh -vv sa1149@agate.cs.unh.edu
2 OpenSSH_8.1p1, LibreSSL 2.7.3
3 debug1: Reading configuration data /etc/ssh/ssh_config
4 debug1: /etc/ssh/ssh_config line 47: Applying options for *
5 debug1: Connecting to agate.cs.unh.edu port 22.
6 debug1: Connection established.
7 debug1: identity file /Users/shi-on/.ssh/id_rsa type 0
```

```
8 debug1: identity file /Users/shi-on/.ssh/id_rsa-cert type -1
9 debug1: identity file /Users/shi-on/.ssh/id_dsa type -1
10 debug1: identity file /Users/shi-on/.ssh/id_dsa-cert type -1
11 debug1: identity file /Users/shi-on/.ssh/id_ecdsa type -1
12 debug1: identity file /Users/shi-on/.ssh/id_ecdsa-cert type -1
13 debug1: identity file /Users/shi-on/.ssh/id_ed25519 type -1
14 debug1: identity file /Users/shi-on/.ssh/id_ed25519-cert type -1
15 debug1: identity file /Users/shi-on/.ssh/id_xmss type -1
16 debug1: identity file /Users/shi-on/.ssh/id_xmss-cert type -1
17 debug1: Local version string SSH-2.0-OpenSSH_8.1
18 debug1: Remote protocol version 2.0, remote software version OpenSSH_8.1
19 debug1: match: OpenSSH_8.1 pat OpenSSH* compat 0x04000000
20 debug1: Authenticating to agate.cs.unh.edu:22 as 'sa1149'
21 debug1: SSH2_MSG_KEXINIT sent
22 debug1: SSH2_MSG_KEXINIT received
23 debug2: local client KEXINIT proposal
24 debug2: KEX algorithms: curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-
nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-
hellman-group14-sha256,diffie-hellman-group14-sha1,ext-info-c
25 debug2: host key algorithms: ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-
nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ecdsa-
sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519-cert-
v01@openssh.com,rsa-sha2-512-cert-v01@openssh.com,rsa-sha2-256-cert-v01@openssh.
com,ssh-rsa-cert-v01@openssh.com,ssh-ed25519,rsa-sha2-512,rsa-sha2-256,ssh-rsa
26 debug2: ciphers ctos: chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr
,aes128-gcm@openssh.com,aes256-gcm@openssh.com
27 debug2: ciphers stoc: chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr
,aes128-gcm@openssh.com,aes256-gcm@openssh.com
28 debug2: MACs ctos: umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64-
@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1
29 debug2: MACs stoc: umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64-
@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1
30 debug2: compression ctos: none,zlib@openssh.com,zlib
31 debug2: compression stoc: none,zlib@openssh.com,zlib
32 debug2: languages ctos:
33 debug2: languages stoc:
34 debug2: first_kex_follows 0
35 debug2: reserved 0
36 debug2: peer server KEXINIT proposal
37 debug2: KEX algorithms: curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-
nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-
hellman-group18-sha512,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-
-sha1
38 debug2: host key algorithms: rsa-sha2-512,rsa-sha2-256,ssh-rsa,ecdsa-sha2-nistp256,
ssh-ed25519
39 debug2: ciphers ctos: aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-
ctr,aes256-cbc,aes128-gcm@openssh.com,aes128-ctr,aes128-cbc
40 debug2: ciphers stoc: aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-
ctr,aes256-cbc,aes128-gcm@openssh.com,aes128-ctr,aes128-cbc
41 debug2: MACs ctos: hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha1,umac-128-
@openssh.com,hmac-sha2-512
42 debug2: MACs stoc: hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha1,umac-128-
@openssh.com,hmac-sha2-512
43 debug2: compression ctos: none,zlib@openssh.com
44 debug2: compression stoc: none,zlib@openssh.com
```

```
45 debug2: languages ctos:
46 debug2: languages stoc:
47 debug2: first_kex_follows 0
48 debug2: reserved 0
49 debug1: kex: algorithm: curve25519-sha256
50 debug1: kex: host key algorithm: ecdsa-sha2-nistp256
51 debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit>
      compression: none
52 debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit>
      compression: none
53 debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
54 debug1: Server host key: ecdsa-sha2-nistp256 SHA256:9/iGsnFwRo+
      Y5rFvcgAenpWV68T3LlclnhycYTxtr8
55 debug1: Host 'agate.cs.unh.edu' is known and matches the ECDSA host key.
56 debug1: Found key in /Users/shi-on/.ssh/known_hosts:28
57 debug2: set_newkeys: mode 1
58 debug1: rekey out after 134217728 blocks
59 debug1: SSH2_MSG_NEWKEYS sent
60 debug1: expecting SSH2_MSG_NEWKEYS
61 debug1: SSH2_MSG_NEWKEYS received
62 debug2: set_newkeys: mode 0
63 debug1: rekey in after 134217728 blocks
64 debug1: Will attempt key: /Users/shi-on/.ssh/id_rsa RSA SHA256:hxXIAqYgANs++5
      HrJTj9qymB5tqH2aAwQUHSc9PAGcU
65 debug1: Will attempt key: /Users/shi-on/.ssh/id_dsa
66 debug1: Will attempt key: /Users/shi-on/.ssh/id_ecdsa
67 debug1: Will attempt key: /Users/shi-on/.ssh/id_ed25519
68 debug1: Will attempt key: /Users/shi-on/.ssh/id_xmss
69 debug2: pubkey_prepare: done
70 debug1: SSH2_MSG_EXT_INFO received
71 debug1: kex_input_ext_info: server-sig-algs=<ssh-ed25519,ssh-rsa,rsa-sha2-256,rsa-
      sha2-512,ssh-dss,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521>
72 debug2: service_accept: ssh-userauth
73 debug1: SSH2_MSG_SERVICE_ACCEPT received
74 debug1: Authentications that can continue: publickey,gssapi-keyex,gssapi-with-mic,
      password
75 debug1: Next authentication method: publickey
76 debug1: Offering public key: /Users/shi-on/.ssh/id_rsa RSA SHA256:hxXIAqYgANs++5
      HrJTj9qymB5tqH2aAwQUHSc9PAGcU
77 debug2: we sent a publickey packet, wait for reply
78 debug1: Server accepts key: /Users/shi-on/.ssh/id_rsa RSA SHA256:hxXIAqYgANs++5
      HrJTj9qymB5tqH2aAwQUHSc9PAGcU
79 debug1: Authentication succeeded (publickey).
80 Authenticated to agate.cs.unh.edu ([132.177.4.36]:22).
81 debug1: channel 0: new [client-session]
82 debug2: channel 0: send open
83 debug1: Requesting no-more-sessions@openssh.com
84 debug1: Entering interactive session.
85 debug1: pledge: network
86 debug1: client_input_global_request: rtype hostkeys-00@openssh.com want_reply 0
87 debug1: Remote: /home/csg/sa1149/.ssh/authorized_keys:1: key options: agent-
      forwarding port-forwarding pty user-rc x11-forwarding
88 debug1: Remote: /home/csg/sa1149/.ssh/authorized_keys:1: key options: agent-
      forwarding port-forwarding pty user-rc x11-forwarding
89 debug2: channel_input_open_confirmation: channel 0: callback start
90 debug2: fd 5 setting TCP_NODELAY
91 debug2: client_session2_setup: id 0
92 debug2: channel 0: request pty-req confirm 1
93 debug1: Sending environment.
94 debug1: Sending env LANG = en_US.UTF-8
95 debug2: channel 0: request env confirm 0
```

```

96 debug2: channel 0: request shell confirm 1
97 debug2: channel_input_open_confirmation: channel 0: callback done
98 debug2: channel 0: open confirm rwindow 0 rmax 32768
99 debug2: channel_input_status_confirm: type 99 id 0
100 debug2: PTY allocation request accepted on channel 0
101 debug2: channel 0: rcvd adjust 2097152
102 debug2: channel_input_status_confirm: type 99 id 0
103 debug2: shell request accepted on channel 0
104 /**
105 * Please read /etc/motd.ssh for an important notice
106 * regarding ssh access to agate
107 */
108 Last login: Mon Apr  6 14:44:04 2020 from 67.189.133.88
109 /**
110 * Please read /etc/motd.ssh for an important notice
111 * regarding ssh access to agate
112 */
113 [sa1149@agate ~]$
```

I have listed the ciphers used in the exchange. As the authentication to ssh server goes in two steps we can see two set of used ciphers for each side which adds up to 4 sets. **Ciphers used from client to server (ctos) and from server to client (stoc):**

```

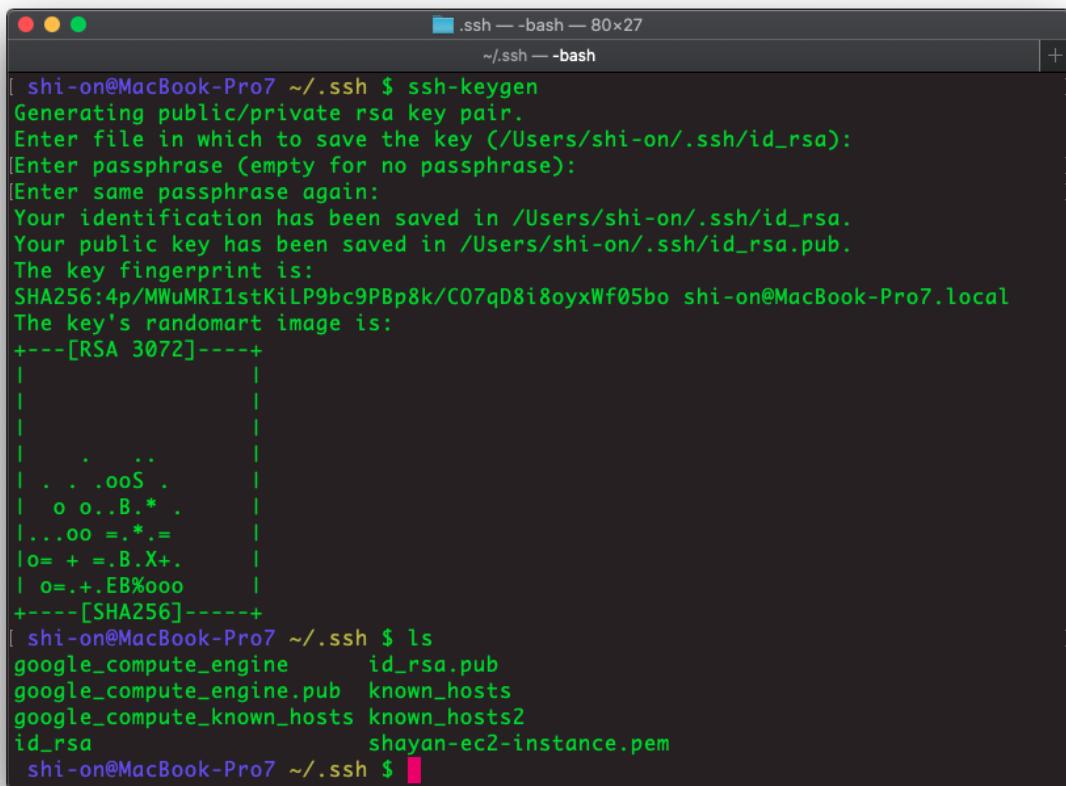
1 ciphers client->server:
2 chacha20-poly1305@openssh.com,
3 aes128-ctr,
4 aes192-ctr,
5 aes256-ctr,
6 aes128-gcm@openssh.com,
7 aes256-gcm@openssh.com
8
9 ciphers server->client:
10 chacha20-poly1305@openssh.com,
11 aes128-ctr,
12 aes192-ctr,
13 aes256-ctr,
14 aes128-gcm@openssh.com,
15 aes256-gcm@openssh.com
16
17
18
19 ciphers client->server:
20 aes256-gcm@openssh.com,
21 chacha20-poly1305@openssh.com,
22 aes256-ctr,
23 aes256-cbc,
24 aes128-gcm@openssh.com,
25 aes128-ctr,
26 aes128-cbc
27
28 ciphers server->client:
29 aes256-gcm@openssh.com,
30 chacha20-poly1305@openssh.com,
31 aes256-ctr,
32 aes256-cbc,
33 aes128-gcm@openssh.com,
34 aes128-ctr,
35 aes128-cbc
```

5. ciphers, MACs, and key-exchange algorithms supported by Agate:

```
[sa1149@agate ~]$ ssh -Q cipher
3des-cbc
aes128-cbc
aes192-cbc
aes256-cbc
rijndael-cbc@lysator.liu.se
aes128-ctr
aes192-ctr
aes256-ctr
aes128-gcm@openssh.com
aes256-gcm@openssh.com
chacha20-poly1305@openssh.com
[sa1149@agate ~]$ ssh -Q mac
hmac-sha1
hmac-sha1-96
hmac-sha2-256
hmac-sha2-512
hmac-md5
hmac-md5-96
umac-64@openssh.com
umac-128@openssh.com
hmac-sha1-etm@openssh.com
hmac-sha1-96-etm@openssh.com
hmac-sha2-256-etm@openssh.com
hmac-sha2-512-etm@openssh.com
hmac-md5-etm@openssh.com
hmac-md5-96-etm@openssh.com
umac-64-etm@openssh.com
umac-128-etm@openssh.com
[sa1149@agate ~]$ ssh -Q kex
diffie-hellman-group1-sha1
diffie-hellman-group14-sha1
diffie-hellman-group14-sha256
diffie-hellman-group16-sha512
diffie-hellman-group18-sha512
diffie-hellman-group-exchange-sha1
diffie-hellman-group-exchange-sha256
ecdh-sha2-nistp256
ecdh-sha2-nistp384
ecdh-sha2-nistp521
curve25519-sha256
curve25519-sha256@libssh.org
sntrup4591761x25519-sha512@tinyssh.org
[sa1149@agate ~]$
```

6. Generate a key pair locally and place it on Agate in order to login to Agate without password:

1. on the local machine
2. on Agate
3. login without password



The screenshot shows a terminal window titled '.ssh — bash — 80x27' with the command 'ssh-keygen' being run. The output shows the generation of an RSA key pair, including the public key (id_rsa.pub) and private key (id_rsa). The terminal also displays the SHA256 fingerprint of the key.

```
[ shi-on@MacBook-Pro7 ~/.ssh $ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/shi-on/.ssh/id_rsa):
[Enter passphrase (empty for no passphrase):
[Enter same passphrase again:
Your identification has been saved in /Users/shi-on/.ssh/id_rsa.
Your public key has been saved in /Users/shi-on/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:4p/MWuMRI1stKiLP9bc9PBp8k/C07qD8i8oyxWF05bo shi-on@MacBook-Pro7.local
The key's randomart image is:
+---[RSA 3072]---+
| |
| |
| |
| . . .
| . . .ooS .
| o o..B.* .
|...oo =.*.=
|o= + =.B.X+.
| o=+.EB%ooo
+---[SHA256]---+
[ shi-on@MacBook-Pro7 ~/.ssh $ ls
google_compute_engine      id_rsa.pub
google_compute_engine.pub   known_hosts
google_compute_known_hosts known_hosts2
id_rsa                      shayan-ec2-instance.pem
shi-on@MacBook-Pro7 ~/.ssh $ ]
```

```
[[sa1149@agate .ssh]$ ls
authorized_keys id_rsa id_rsa.pub known_hosts
[[sa1149@agate .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQGDNc4st0PuHuEDYvbJMBYZjsg6bpL17W1RZ7R11w1t3
NOVzDUpiqdtaGsB/ySxlzHnc/K5a1eiA+R4KpNovlG+j3PAj7IJ37UKs5w707D18LzwxPKsN4f+gIDW0
BVzA48t2LmZq4kb4sm2sETsU4ogaX09MQ5/Wpc5q8CbNKW0GJzT5ReYzu+FCBJUj6gwdGnV2gjsCsS5x
GCULyK/MQtKcZaxyFHY0Fle0FK/LplErFx2+j7o6PbV1g2s5+k8HTRfxjoZgj7nL40XGDLr0LwFEwVmX
cKx+ReHGRz0WMRXXM1tJ43x8Y9w6uSvIFtzZ1gqqxBfFe19lRmlcBnAGX2UzKS0Czkn8EcKtSljFYsw6
rGk3Zv29A2aMv37alCsbG09H8zBmR4/fNEsY/l8Msse38X9xnxCjFLd0Xhit5/klyCdPYPJ5p8z3e2V0
oNEEecdc5in1/Up4PRq0DPcWirfffYzZ+WyfWZqd3aLeWiZmlqwj0hDloElTJ1NWUAUA/tk= shi-on@MacBook-Pro7.local
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQG6caKvT+HcEoWEi38RdqZ702L1UcHV1/HKGNb8v
N/BmfPTXGiCrz10vwkI0oCSkrMQIIuin54tfRKZrncejXF6TLmULHRhmM79BuDmzhbeUBcFW02tMb0N
E/ZLFVG1DUYrIaZMo/j0LjdSjodSiDPjI6m+1BZkMlYccZkUmLSPkXeYta6XE2tIZhMZZeweZDjVSU2
BaoGWUbin47WRKhMj4Kwh+oKVDJ/ruMbJT1fBVpSCZ8i29k/d+3TVVWSNTe87qXMMXtrGc1Sq0PUvnr
OW89v49Hs3RP+YFEJSxnFz3J0IK6tvRqtmsjXTN7aWNhSsCyNdmuAPX512b7 sa1149@thinkstation
rmdp
[sa1149@agate .ssh]$ ]
```

```
[ shi-on@MacBook-Pro7 ~/.ssh $ ls
google_compute_engine id_rsa.pub
google_compute_engine.pub known_hosts
google_compute_known_hosts known_hosts2
id_rsa shayan-ec2-instance.pem
[ shi-on@MacBook-Pro7 ~/.ssh $ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQGDNc4st0PuHuEDYvbJMBYZjsg6bpL17W1RZ7R11w1t3
NOVzDUpiqdtaGsB/ySxlzHnc/K5a1eiA+R4KpNovlG+j3PAj7IJ37UKs5w707D18LzwxPKsN4f+gIDW0
BVzA48t2LmZq4kb4sm2sETsU4ogaX09MQ5/Wpc5q8CbNKW0GJzT5ReYzu+FCBJUj6gwdGnV2gjsCsS5x
GCULyK/MQtKcZaxyFHY0Fle0FK/LplErFx2+j7o6PbV1g2s5+k8HTRfxjoZgj7nL40XGDLr0LwFEwVmX
cKx+ReHGRz0WMRXXM1tJ43x8Y9w6uSvIFtzZ1gqqxBfFe19lRmlcBnAGX2UzKS0Czkn8EcKtSljFYsw6
rGk3Zv29A2aMv37alCsbG09H8zBmR4/fNEsY/l8Msse38X9xnxCjFLd0Xhit5/klyCdPYPJ5p8z3e2V0
oNEEecdc5in1/Up4PRq0DPcWirfffYzZ+WyfWZqd3aLeWiZmlqwj0hDloElTJ1NWUAUA/tk= shi-on@MacBook-Pro7.local
[ shi-on@MacBook-Pro7 ~/.ssh $ ssh sa1149@agate.cs.unh.edu
/**
 * Please read /etc/motd.ssh for an important notice
 * regarding ssh access to agate
 */
Last login: Mon Apr  6 16:16:37 2020 from 67.189.133.88
/**
 * Please read /etc/motd.ssh for an important notice
 * regarding ssh access to agate
 */
[[sa1149@agate ~]$ echo "Success!"
Success!
[sa1149@agate ~]$ ]
```

2 Password

1. I have used my VPS as a Linux machine which has **CentOS Linux release 7.7.1908 (Core)** installed. I created a new user 'victim' and set '123' as the password for this user:

```
(base) [root@vps505943 run]# adduser victim
(base) [root@vps505943 run]# passwd
Changing password for user root.
New password:
(base) [root@vps505943 run]# setpasswd
-bash: setpasswd: command not found
(base) [root@vps505943 run]# passwd victim
Changing password for user victim.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
(base) [root@vps505943 run]# ls
```

2. As the shadow file is visible below, I have two hashed passwords in the file for the two users available on this machine.

Explanation of the fields available in the shadow file, in order from left to right:

1. username
2. password:
 - (a) hashing algorithm id: 1 -> MD5
 - (b) salt
 - (c) hashed value
3. last change
4. min days change
5. max days change
6. prior warning days

```
(base) [root@vps505943 etc]# cat shadow
root:$1$6ZBEvvpH$57hpskujc0aqp46162IlU0:18277:0:99999:7:::
bin:*:17834:0:99999:7:::
daemon:*:17834:0:99999:7:::
adm:*:17834:0:99999:7:::
lp:*:17834:0:99999:7:::
sync:*:17834:0:99999:7:::
shutdown:*:17834:0:99999:7:::
halt:*:17834:0:99999:7:::
mail:*:17834:0:99999:7:::
operator:*:17834:0:99999:7:::
games:*:17834:0:99999:7:::
ftp:*:17834:0:99999:7:::
nobody:*:17834:0:99999:7:::
systemd-network:!!!:18231::::::
dbus:!!!:18231::::::
polkitd:!!!:18231::::::
sshd:!!!:18231::::::
postfix:!!!:18231::::::
chrony:!!!:18231::::::
centos:!!!:18268:0:99999:7:::
mysql:!!!:18268::::::
saslauth:!!!:18268::::::
dovecot:!!!:18268::::::
dovenull:!!!:18268::::::
tss:!!!:18268::::::
named:!!!:18268::::::
cwpssrv:!!!:18268::::::
cwpsvc:!!!:18268::::::
login:!!!:18268::::::
clamupdate:!!!:18268::::::
amavis:!!!:18268::::::
clamscan:!!!:18268::::::
vmail:!!!:18268::::::
vacation:!!!:18268:::::
opendkim:!!!:18268:::::
rpc:!!!:18268:0:99999:7:::
nginx:!!!:18268::::::
deluge:!!!:18268::::::
victim:$1$jbttxa0tD$TiEBhdPGR0XlcUfehp4sv0:18358:0:99999:7:::
(base) [root@vps505943 etc]#
```

3. I installed John the Ripper 1.8 by taking the following steps.

```
1#!/bin/bash
2# Centos 7 John the Ripper Installation
3yum -y install wget gpgme
4yum -y group install "Development Tools"
5cd
6wget http://www.openwall.com/john/j/john-1.8.0.tar.xz
7wget http://www.openwall.com/john/j/john-1.8.0.tar.xz.sign
8wget http://www.openwall.com/signatures/openwall-signatures.asc
9gpg --import openwall-signatures.asc
```

```
10 gpg --verify john-1.8.0.tar.xz.sign
11 tar xvJ john-1.8.0.tar.xz
12 cd john-1.8.0/src
13 make clean linux-x86-64
14 cd ../run/
15 ./john --test
16 #password dictionary download
17 wget -O - http://mirrors.kernel.org/openwall/wordlists/all.gz | gunzip -c > openwall
.dico
```

4. After the installation, I used JtR to crack victim's password. As this was a very common and unsafe password, we don't even need to use a prolific word list. As it is displayed in the following graph JtR could quickly find the password, '123':

```
(base) [root@vps505943 run]# ls
ascii.chr  john.conf  john.rec      makechr      relbench  unshadow
digits.chr  john.log   lm_ascii.chr  openwall.dico  unafs
john        john.pot   mailer       password.lst  unique
(base) [root@vps505943 run]# cp /etc/shadow .
(base) [root@vps505943 run]# ./john shadow
Loaded 2 password hashes with 2 different salts (md5crypt [MD5 32/64 X2])
Press 'q' or Ctrl-C to abort, almost any other key for status
123          (victim)
1g 0:00:00:05 41% 2/3 0.2000g/s 14413p/s 14417c/s 14417C/s cssndr..rdsx
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
(base) [root@vps505943 run]#
```