Put a Lid on Those AWS S3 Buckets

SANS Cloud Security Summit

Lily Lee | Staff Security Strategist
Melisa Napoles | Solutions Engineer

May 29, 2020

splunk> turn data into doing™

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Who Are We

Melisa Napoles

Passionate about Big (and small) Data, Cloud and CyberSecurity

Lily Lee

GCIH, GMON, WiCyS Silicon Valley

Goals

- Why focus on misconfigured buckets?
- What are the anticipated and unanticipated fallouts of misconfigured buckets?
- What are the key data sources for identifying misconfigured buckets?
- How do I pivot between data sources for understanding what happened?

A Quick Recap from SANS Cloud Security Summit 2019

"Keep It Flexible–How Cloud Makes It Easier and Harder to Detect Bad Stuff"

- On-premises infrastructure mapped to corresponding AWS services
- AWS Shared Responsibility Model
- Understand what data is security-relevant; and where and how to get that data
- An in-depth look at CloudTrail to detect malicious activity
- Best practice checklist (e.g., AWS Trusted Advisor, AWS Knowledge Center)

Secure the Files in Your Amazon S3 Bucket

Common (Human) Errors

- Allowing anonymous access
- Allowing file listing
- Allowing arbitrary file upload / download
- Allowing read / writes of objects
- Allowing control of the files and objects
- Revealing ACP / ACL

Best Practices

- Restrict access to your S3 resources (IAM user permissions, bucket policies, ACLs)
- Use encryption to protect your data (at rest+in transit)
- Create data copies
- Enable versioning & S3 Object Lock
- Enable multi-factor authentication delete
- Monitor your S3 resources (S3 access logging, CloudTrail, Config)
- Use *S3 Access Points* to manage data at scale
- Use *block public access* setting
- Enable AWS Config rules (s3-bucket-public-[write|read]-prohibited)

S3 Misconfiguration Can Lead to Data Breach and Other Security Incidents

A few examples

Financial Data

Login Credentials

Payment Data

PII Data

Proprietary Data

Credit Card Skimming

Injection Attack

Config Files

Encryption Keys

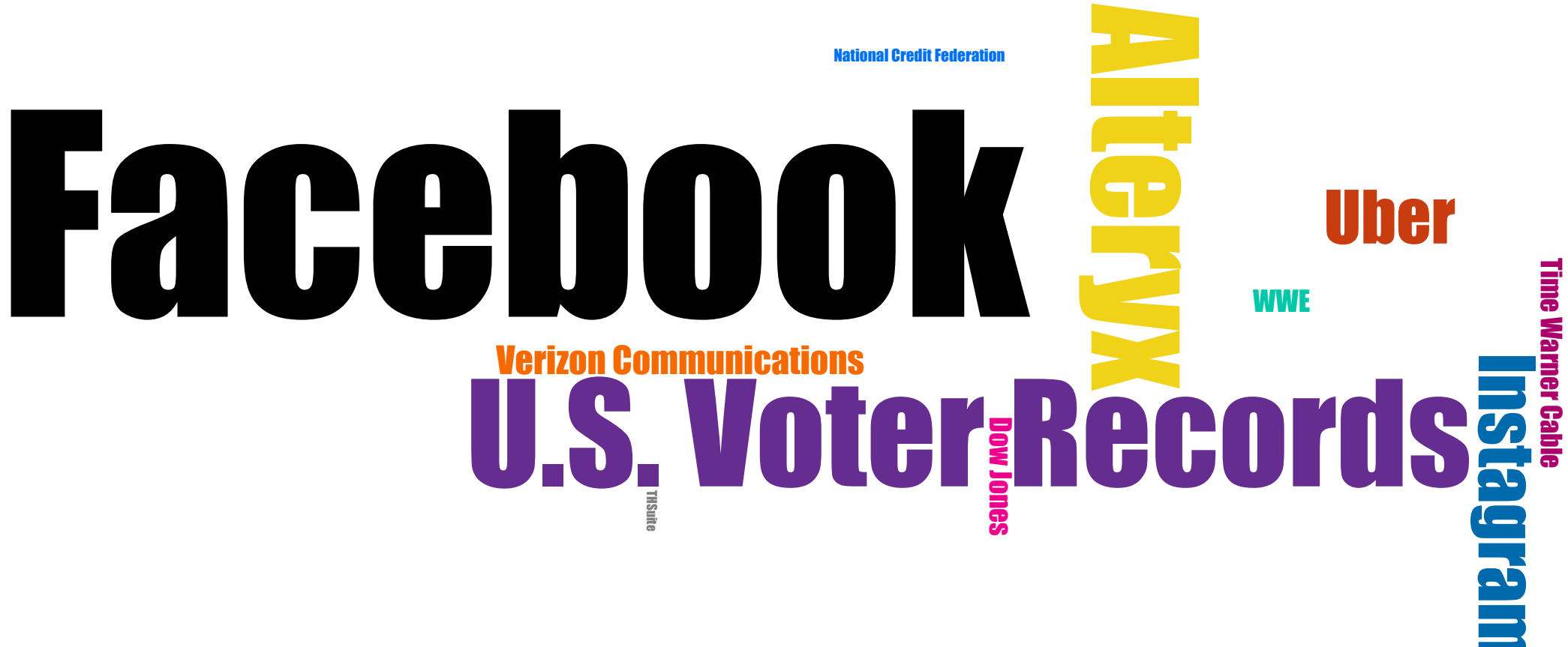
Website Content

Ransomware Attack

Denial of Wallet Attack

Inadvertent AWS S3 Data Breaches

PII exposed ranging from 30,000 to 540 million people

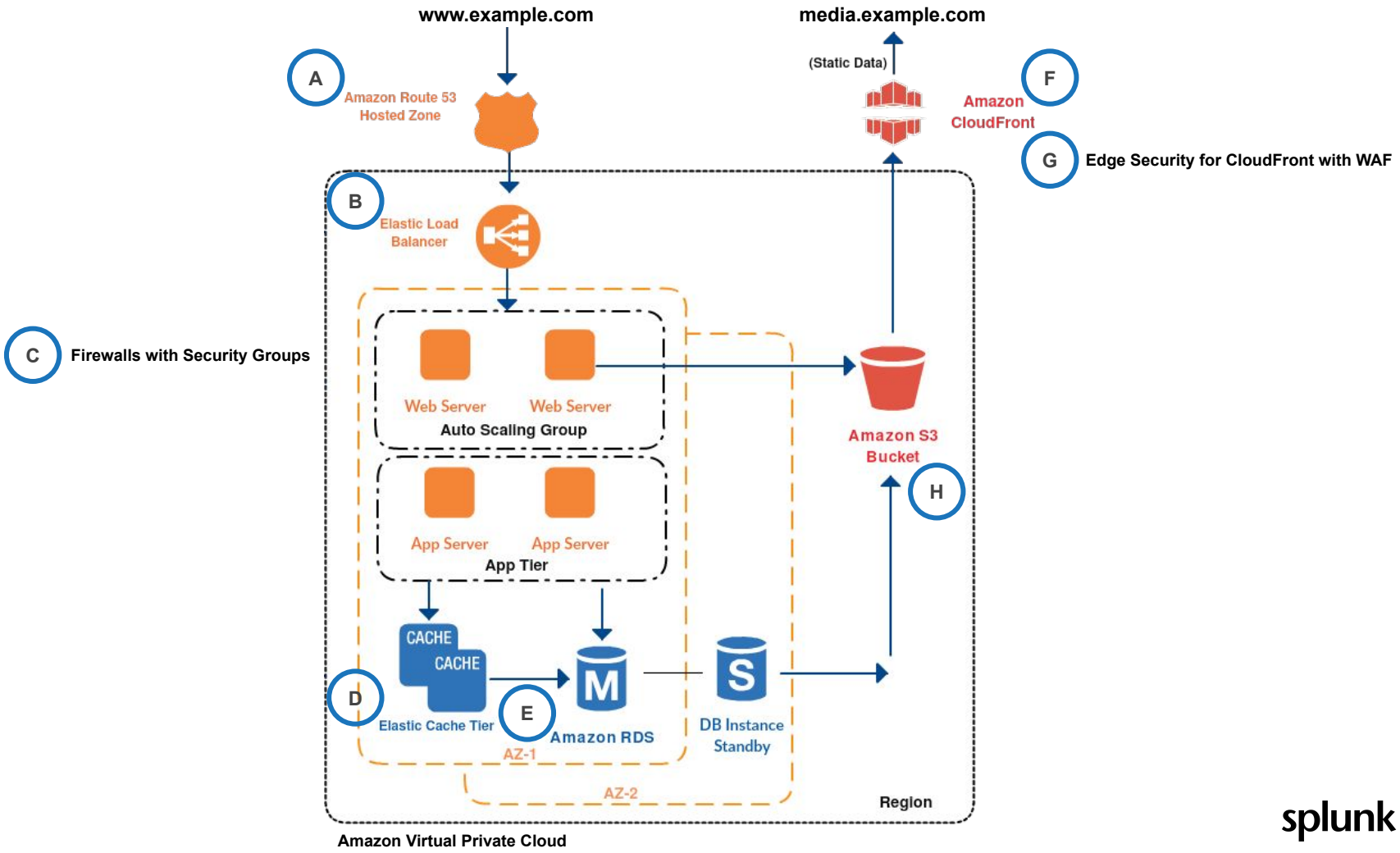


Real-World Attack Scenario

S3 Misconfiguration Leading to Cryptojacking

splunk> turn data into doing™

Example AWS Cloud Architecture for Web Application Hosting



What Is Cryptomining / Cryptojacking?

CRYPTOMINING

Using computer resources, such as CPU cycles, in exchange for money, or cryptocurrency.

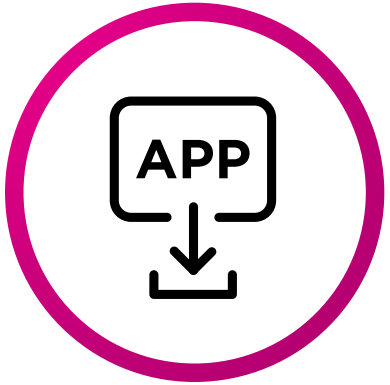
CRYPTOJACKING

The unauthorized use of someone else's device or machine to mine cryptocurrency (i.e., malicious cryptomining).

How Can Cryptojacking Occur?

Distribution mechanisms

Install cryptomining code / software



Any device that can provide CPU cycles and electricity is vulnerable

Browser-based cryptomining code

JavaScript code that executes when loaded in the web browser

Public / Guest Wi-Fi

Inject cryptomining content to HTML requests

Amazon S3

Buckets (40)

Buckets are the fundamental container in Amazon S3 for data storage. For others to access the objects in your buckets, you'll need to explicitly grant them pe

frothlywebcode | 1 match

Name	Region	Access
frothlywebcode	US East (N. Virginia) us-east-1	Public

To: allhands@froth.ly

Cc:

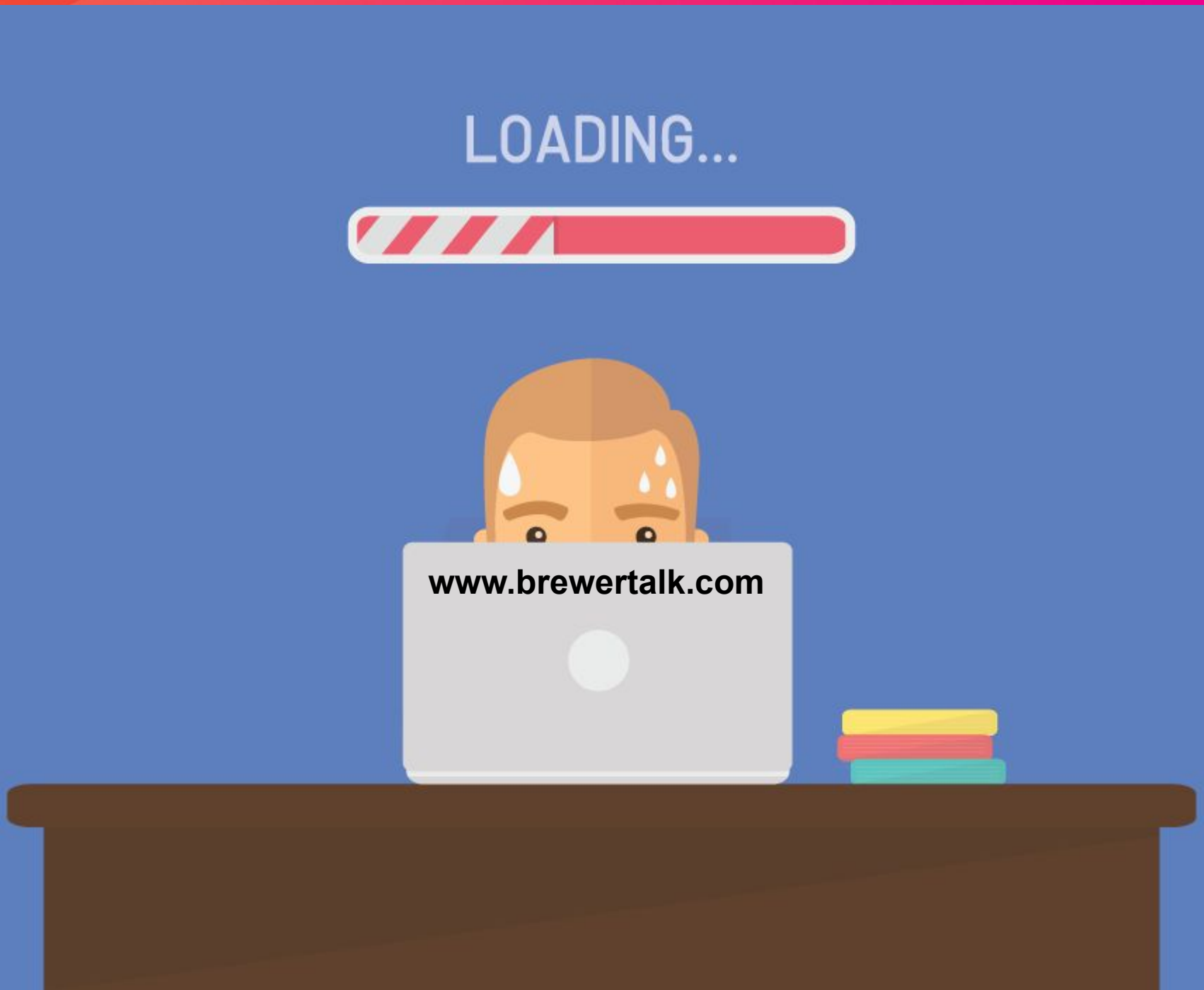
Bcc:

Subject: Improved brewertalk.com - check it out!

Hey Frothlies!

I just added some great improvements to brewertalk.com to better handle forum threads and allow for posts of multi-media kinds of files rather than just photos. And it's all running in our new swanky AWS environment ("the cloud" for those of you not sure what that is!) I think you'll be impressed enough to maybe buy me a beer! Let me know.

http://www.brewertalk.com

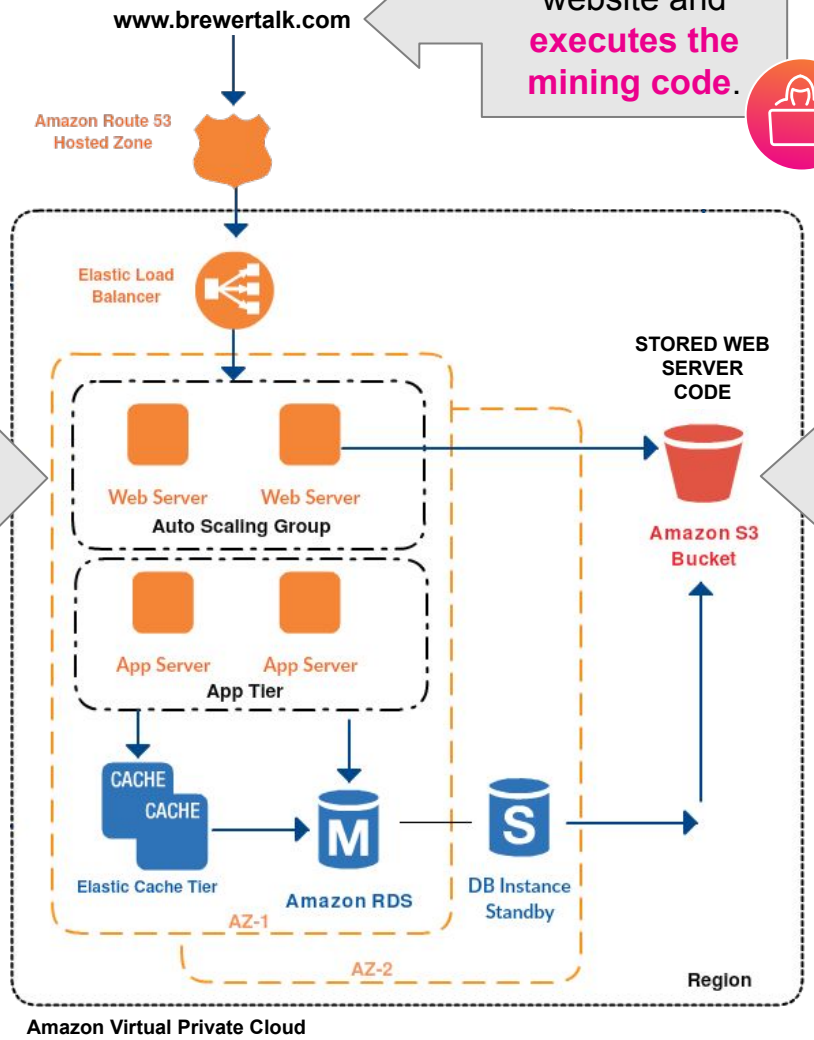

LOADING...
www.brewertalk.com

The Attack Scenario

3 User loads the website and **executes the mining code**.

2 A burst of web traffic causes the web infra to **spin up more** EC2 instances with the **new website code**.

1 Hacker **discovers** our website content in the public bucket, **downloads** it, adds the **mining code**, and **uploads a new version** of the website content back into the bucket.

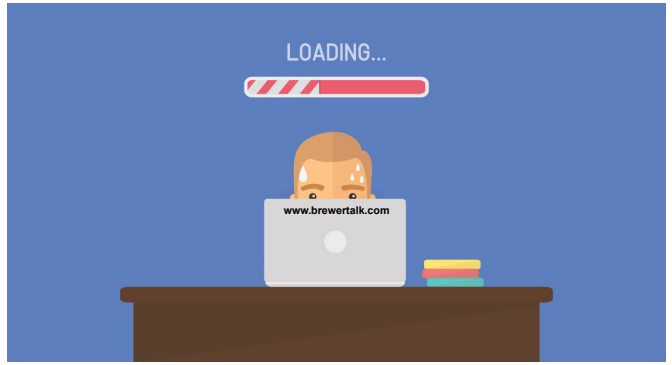


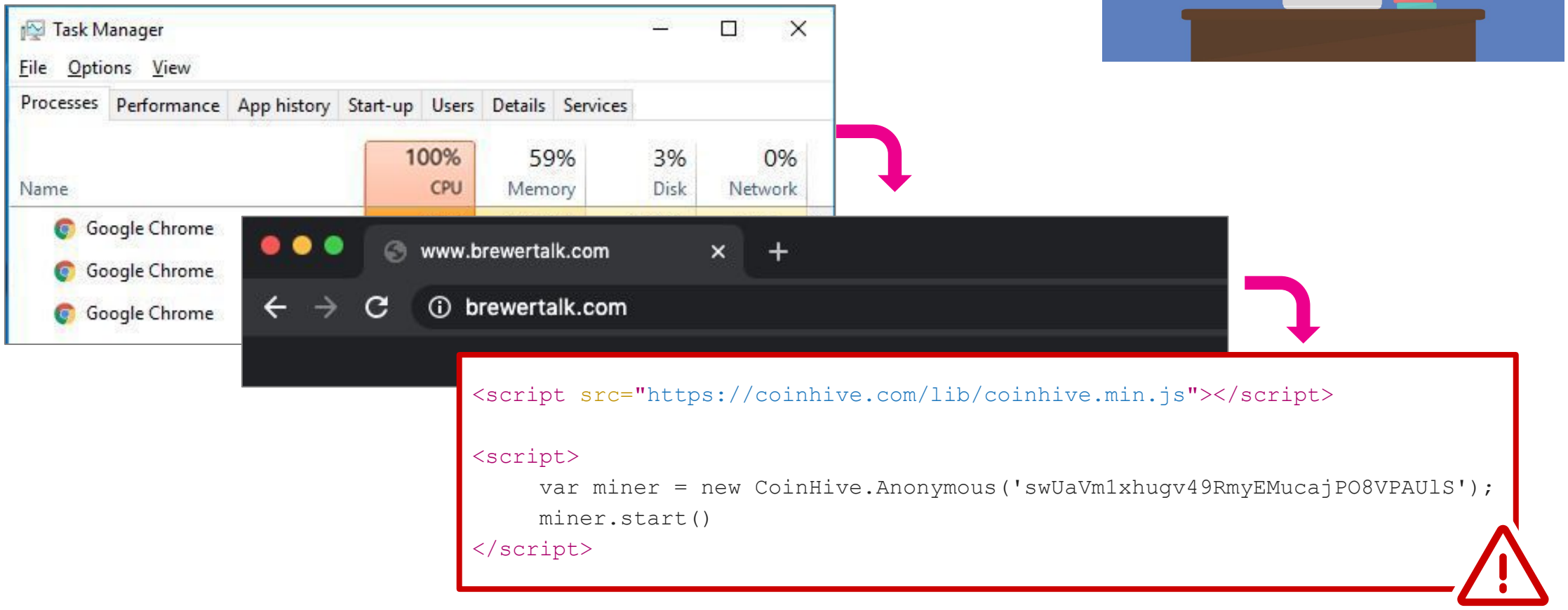
The Investigation

What Happened Post-Exploit

splunk> turn data into doing™

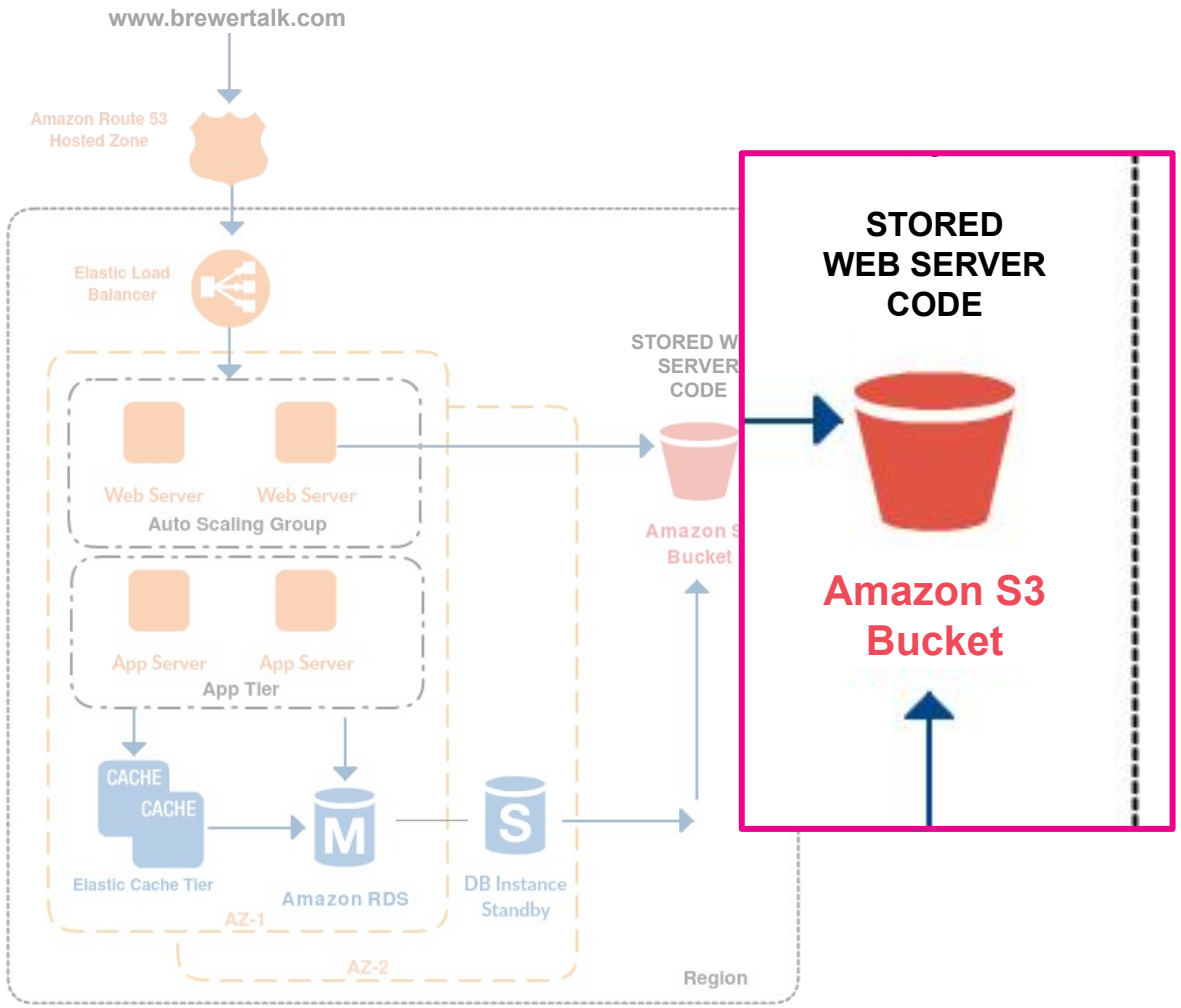
The Endpoint Investigation





Err… How did that miner code get into brewertalk.com? Those weren't *my* changes!

Reviewing AWS Logs, Part I



1

CLOUDTRAIL LOGS

Review S3 bucket permissions.

- Ensure the S3 bucket is not public.
- If exposed, confirm when and by whom.

2

S3 ACCESS LOGS

Audit S3 bucket access.

- Review all bucket activity post exposure (especially uploads and downloads).

What Does Public Read Access Look Like?

CloudTrail Logs


```
{ [-]
  awsRegion: us-west-1
  eventID: ab45689d-69cd-41e7-8705-5350402cf7ac
  eventName: PutBucketAcl
  eventSource: s3.amazonaws.com
  eventTime: 2018-08-20T13:01:46Z
  eventType: AwsApiCall
  eventVersion: 1.05
  recipientAccountId: 622676721278
  requestID: 487488D003569438
  requestParameters: { [-]
    AccessControlPolicy: { [+]
    }
    acl: [ [+]
    ]
    bucketName: frothlywebcode
  }
  responseElements: null
  sourceIPAddress: 107.77.212.175
  userAgent: signin.amazonaws.com
  userIdentity: { [-]
    accessKeyId: ASIAZB6TMXZ7OA2RDK5X
    accountId: 622676721278
    arn: arn:aws:iam::622676721278:user/bstoll
    invokedBy: signin.amazonaws.com
    principalId: AIDAJUFKXZ44LV4EN4MGK
    sessionContext: { [+]
    }
    type: IAMUser
    userName: bstoll
  }
}
```


WHERE → awsRegion

WHAT → eventName

WHEN → eventTime

WHAT → bucketName

WHO → arn


```
requestParameters: { [-]
  AccessControlPolicy: { [-]
    AccessControlList: { [-]
      Grant: [ [-]
        { [-]
          Grantee: { [-]
            URI: http://acs.amazonaws.com/groups/global/AllUsers
            xmlns:xsi: http://www.w3.org/2001/XMLSchema-instance
            xsi:type: Group
          }
          Permission: READ
        }
        { [-]
          Grantee: { [-]
            URI: http://acs.amazonaws.com/groups/global/AllUsers
            xmlns:xsi: http://www.w3.org/2001/XMLSchema-instance
            xsi:type: Group
          }
          Permission: WRITE
        }
```


HOW → URI

HOW → Permission: READ

HOW → URI

HOW → Permission: WRITE

What Happened Post Exposure?

S3 Access Logs: A Look at Download (REST.GET.OBJECT) & Upload (REST.PUT.OBJECT) Activity

Time	Source IP	Requester	Operation	Key	HTTP Status	Bytes Sent	Object Size	User Agent
2018-08-20 07:02:44	52.66.146.128	-	REST.PUT.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt	200	-	377	Boto3/1.7.62 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 Botocore/1.8.12
2018-08-20 07:03:46	35.182.246.222	-	REST.GET.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt	200	377	377	aws-cli/1.14.8 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 botocore/1.8.12
2018-08-20 07:03:46	35.182.246.222	-	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.8 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 botocore/1.8.12
2018-08-20 07:04:17	35.182.246.222	-	REST.PUT.OBJECT	frothly_html_memcached.tar.gz	200	-	3076532	Boto3/1.7.61 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 Botocore/1.8.12
2018-08-20 07:33:35	54.183.247.244	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-0cc93bade2b3cba63	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13
2018-08-20 07:59:21	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:00:23	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:04:09	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:07:04	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134
2018-08-20 08:19:19	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.PUT.OBJECT	frothly_html_memcached.tar.gz	200	-	3057116	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:23:32	54.67.37.214	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-06fea586f3d3c8ce8	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13
2018-08-20 08:25:32	52.53.233.88	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-09cbc261e84259b54	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13

Notable Activity

OPEN_BUCKET_PLEASE_FIX.txt

Time	Source IP	Requester	Operation	Key	HTTP Status	Bytes Sent	Object Size	User Agent
2018-08-20 07:02:44	52.66.146.128	REMOTE IP	UPLOAD(S) REST.PUT.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt	200	-	377	
2018-08-20 07:03:46	35.182.246.222	-	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.8 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 botocore/1.8.12
2018-08-20 07:04:17	35.182.246.222	-						Boto3/1.7.61 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 Botocore/1.8.12
2018-08-20 07:33:35	54.183.247.244	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-0cc93bade2b3cba63						aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13
2018-08-20 07:59:21	107.77.212.175	arn:aws:iam::622676721278:user/bstoll						Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:00:23	107.77.212.175	arn:aws:iam::622676721278:user/bstoll						Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:04:09	107.77.212.175	arn:aws:iam::622676721278:user/bstoll						Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:07:04	107.77.212.175	arn:aws:iam::622676721278:user/bstoll						Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134
2018-08-20 08:19:19	107.77.212.175	arn:aws:iam::622676721278:user/bstoll						Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:23:32	54.67.37.214	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-06fea586f3d3c8ce8						aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13
2018-08-20 08:25:32	52.53.233.88	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-09cbc261e84259b54						aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13

```
Hello AWS account owner,

This is a friendly notice from your
neighborhood whitehat hacker, letting
you know you have an insecurely
configured bucket in your account. This
can be used by criminals to launder
stolen information or to possibly
exfiltrate your sensitive data.

Please update your bucket permissions
to restrict access as much as possible.

Thanks,
Bucket Brigade
```

Notable Activity

frothly_html_memcached.tar.gz

Time	Source IP	Requester	Operation	Key	HTTP Status	Bytes Sent	Object Size	User Agent
2018-08-20 07:02:44	52.66.146.128	-	REST.PUT.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt	200	-	377	Boto3/1.7.62 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 Botocore/1.8.12
2018-08-20 07:03:46	35.182.246.222	-	REST.GET.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt	200	377	377	aws-cli/1.14.8 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 botocore/1.8.12
2018-08-20 07:03:46	35.182.246.222	-	REST.GET.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt	200	377	377	
2018-08-20 07:03:46	35.182.246.222	-	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	
2018-08-20 07:04:17	35.182.246.222	-	REST.PUT.OBJECT	frothly_html_memcached.tar.gz	200	-	3076532	
2018-08-20 08:00:23	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:04:09	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:07:04	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134
2018-08-20 08:19:19	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.PUT.OBJECT	frothly_html_memcached.tar.gz	200	-	3057116	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:23:32	54.67.37.214	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-06fea586f3d3c8ce8	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13
2018-08-20 08:25:32	52.53.233.88	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-09cbc261e84259b54	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13

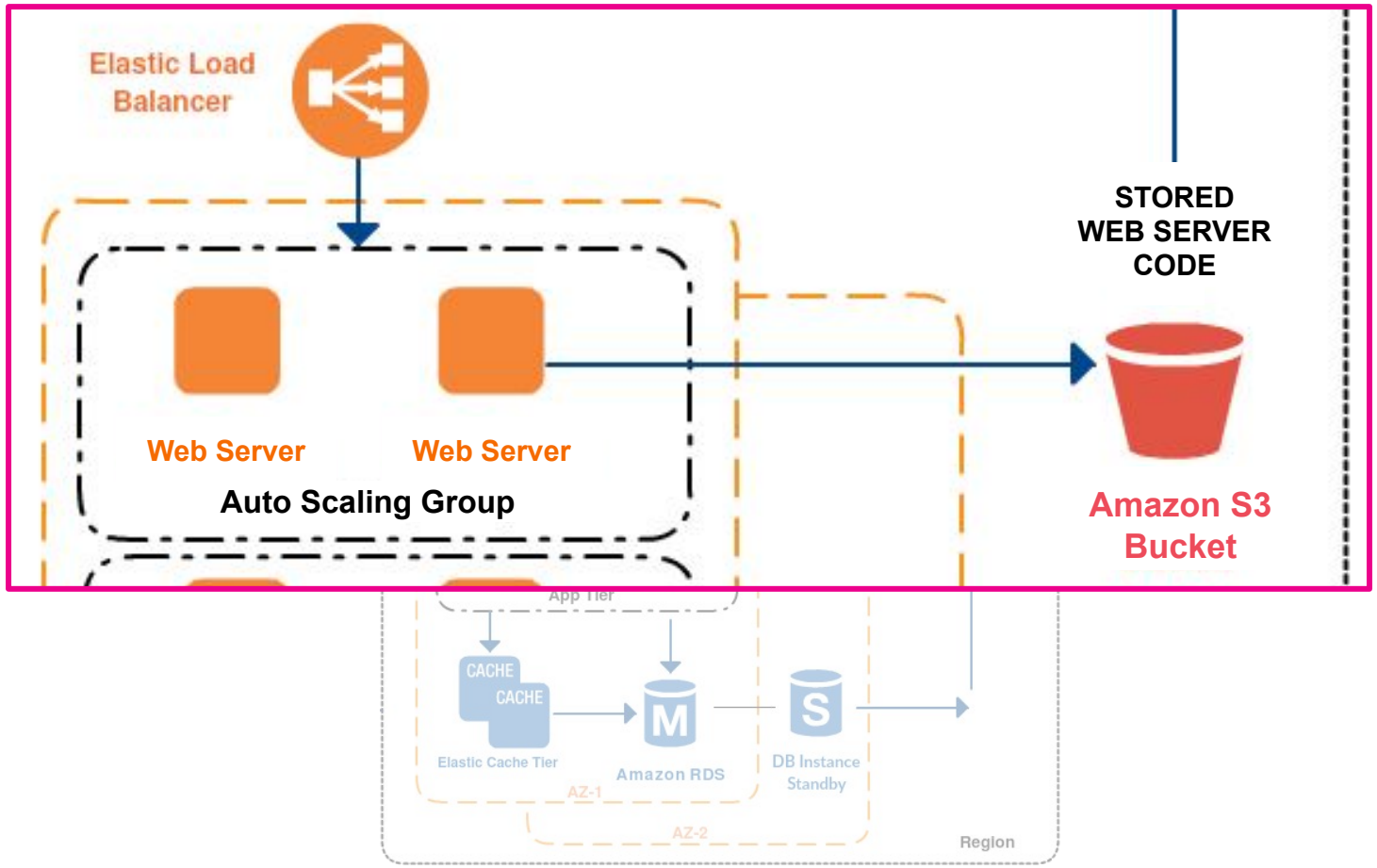
REMOTE IP

DOWNLOAD(S)

UPLOAD(S)

Another Look at the Architecture ...

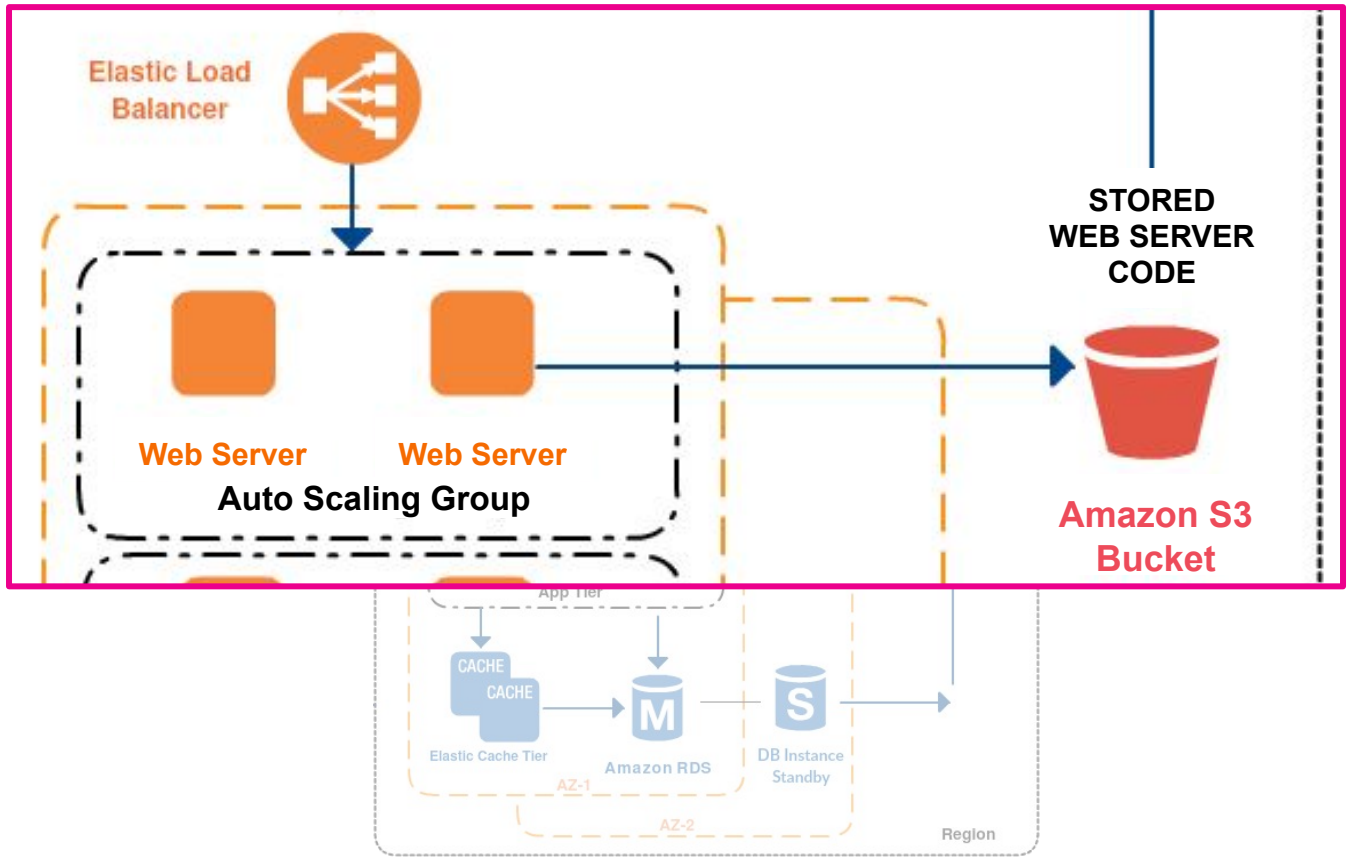
Web Server Code Stored in S3 Bucket



frothly_html_memcached.tar.gz is our web server code!

Reviewing AWS Logs, Part II

Investigate the Web Infrastructure for Suspicious Activity



3

ENDPOINT LOGS

Confirm the purpose of ‘frothly_html_memcached.tar.gz’.

4

ELB ACCESS LOGS

Review external web requests for suspicious activity.

5

CLOUDTRAIL LOGS

How did the modified code get deployed to the web servers?

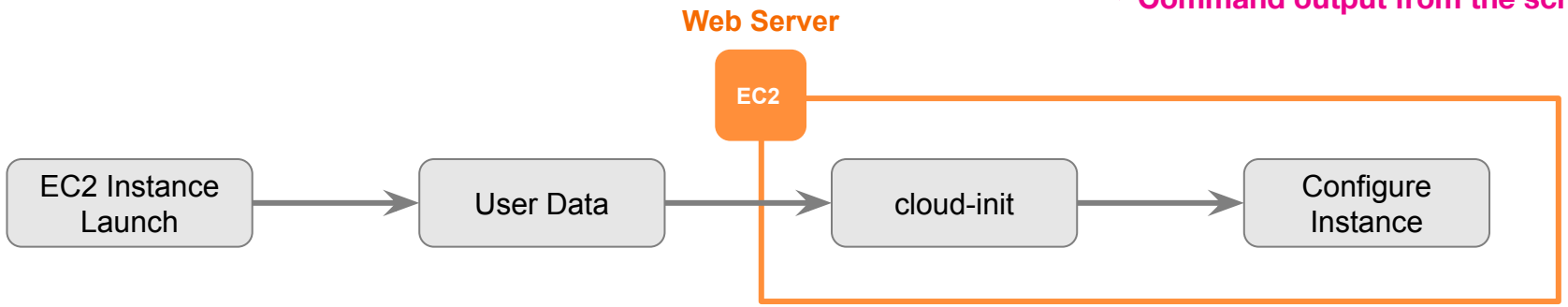
User Data and Cloud-Init

Bootstrapping an EC2 Instance

i	Time	Event
>	8/20/18 7:33:24.000 AM	Cloud-init v. 0.7.6 running 'modules:final' at Thu, 26 Jul 2018 00:45:24 +0000. Up 11.83 seconds. ... 229 lines omitted ... Completed 2.9 MiB/2.9 MiB (6.8 MiB/s) with 1 file(s) remaining download: s3://frothlywebcode/frothly_html_memcached.tar.gz to ./frothly_html_memcached.tar.gz Completed 17.1 KiB/17.1 KiB (42.1 KiB/s) with 1 file(s) remaining download: s3://frothlyweb/configs/http_conf.tar.gz to ./http_conf.tar.gz Show all 257 lines host = gacrux.i-0cc93bade2b3cba63 | source = /var/log/cloud-init-output.log | sourcetype = cloud-init-output

Command output from the script

HOW IT WORKS



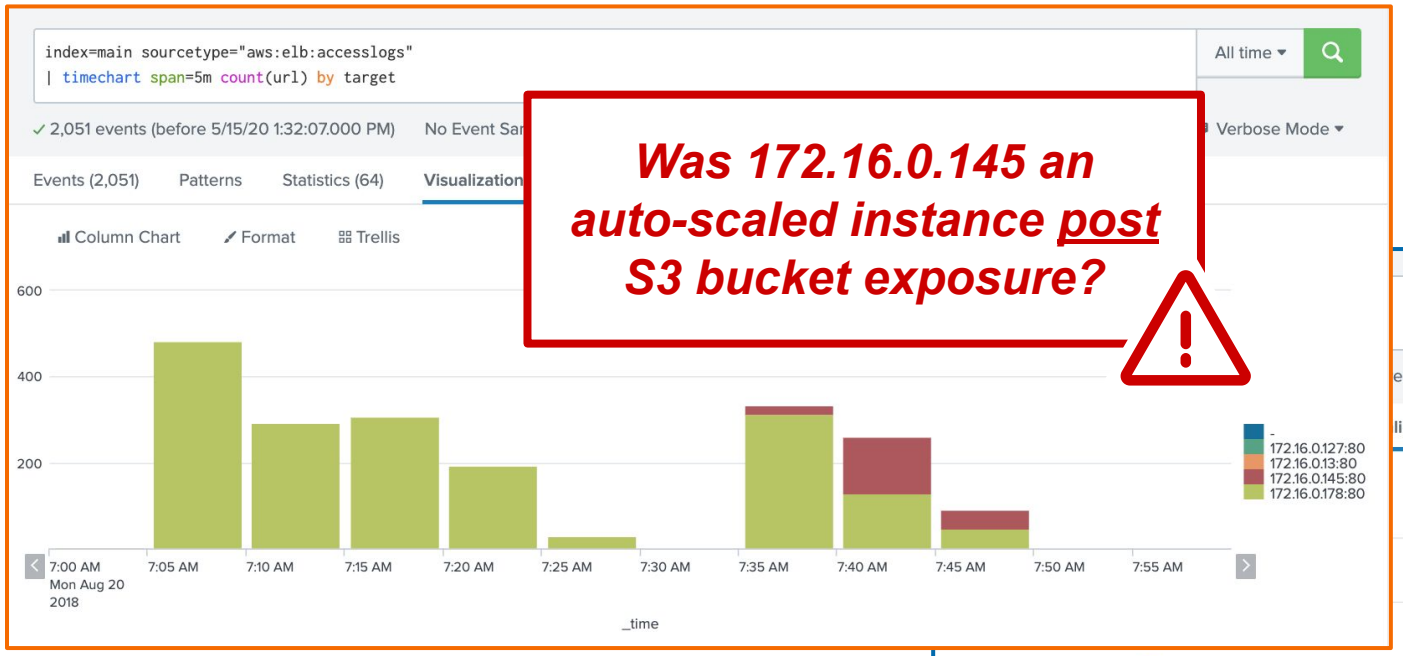
For more information: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html

Notable Web Activity–What to Look For

- Count of requests by remote IP, geolocation, user agent
- Count of requests by destination target (i.e., web tier)
- Top requests by storage object URL
- Baseline response sizes and processing times
- Analyze traffic patterns (e.g., frequency, distribution)
- Look for server errors (e.g., HTTP 503 errors)

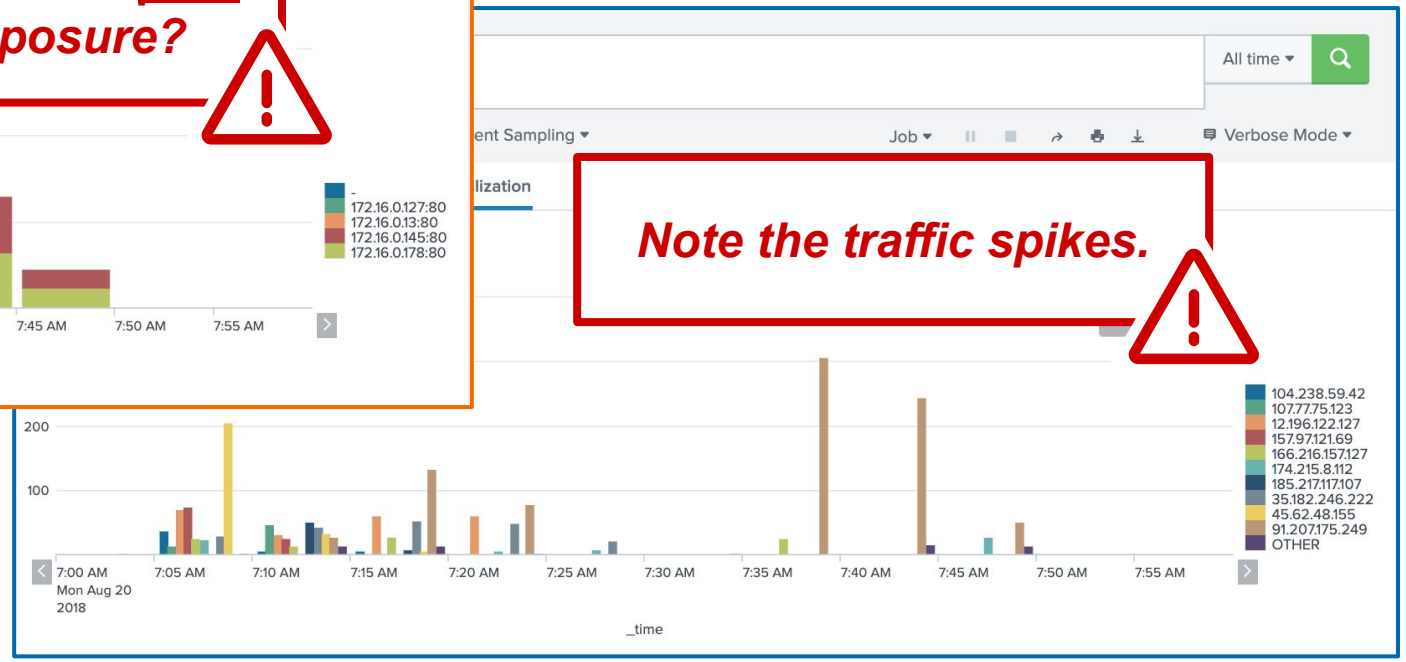
Notable Web Activity

ELB Access Logs



Was 172.16.0.145 an auto-scaled instance post S3 bucket exposure?

Count of Requests by Destination Target



Note the traffic spikes.

Count of Requests by Client IP

EC2 Auto Scaling Information

CloudTrail Logs

After the bucket was made public ...

8/20/18
7:32:59.000 AM


```
{ [-]
  awsRegion: us-west-1
  eventID: ee2f25c1-f115-47a4-
  eventName: RunInstances
  eventSource: ec2.amazonaws.c
  eventTime: 2018-08-20T13:32:59Z
  eventType: AwsApiCall
  eventVersion: 1.05
  recipientAccountId: 622676721278
  requestID: 60301c9d-04ea-42c4-b42e-ddc4c799a4df
  requestParameters: { [+]
  }
  responseElements: { [+]
  }
  sourceIPAddress: autoscaling.amazonaws.com
  userAgent: autoscaling.amazonaws.com
  userIdentity: { [+]
  }
}
```


An EC2 instance was started ...

… By EC2 Auto Scaling

The Modified Code *Was* Deployed

The associated private IP is 172.16.0.145 which is the auto-scaled instance.

Automation script downloads the tar.gz as part of initial configuration.

Object size matches modified code

Time			Operation	Key				
2018-08-20 07:02:4			REST.PUT.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt				hon/2.7.14 Linux/4.14.47-64.38
2018-08-20 07:03:4			REST.GET.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt				Python/2.7.14 Linux/4.14.47-64.
2018-08-20 07:03:46	35.182.246		REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	30		aws-cli/1.14.8 Python/2.7.14 Linux/4.14.47-64.38.amzn2. botocore/1.8.12
2018-08-20 07:33:35	54.183.247.244	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-0cc93bade2b3cba63	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	
2018-08-20 07:59:21	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:00:23	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:04:09	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:07:04	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134
2018-08-20 08:19:19	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.PUT.OBJECT	frothly_html_memcached.tar.gz	200	-	3057116	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:23:32	54.67.37.214	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-06fea586f3d3c8ce8	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13
2018-08-20 08:25:32	52.53.233.88	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-09cbc261e84259b54	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13

Recovery: Securing the S3 Bucket Permissions

Confirmed by CloudTrail Logs

TABULAR VIEW OF CLOUDTRAIL LOGS SHOWING THERE WAS PUBLIC READ / WRITE ACCESS

Time	Source IP	Region	Permission	Grantee	Bucket Name	Requester
2018-08-20 07:01:46	107.77.212.175	us-west-1	WRITE	http://acs.amazonaws.com/groups/global/AllUsers	frothlywebcode	bstoll
2018-08-20 07:01:46	107.77.212.175	us-west-1	READ	http://acs.amazonaws.com/groups/global/AllUsers	frothlywebcode	bstoll
2018-08-20 07:01:46	107.77.212.175	us-west-1	FULL_CONTROL		frothlywebcode	bstoll
2018-08-20 07:01:46	107.77.212.175	us-west-1	READ	http://acs.amazonaws.com/groups/s3/LogDelivery	frothlywebcode	bstoll
2018-08-20 07:01:46	107.77.212.175	us-west-1	READ_ACP	http://acs.amazonaws.com/groups/s3/LogDelivery	frothlywebcode	bstoll
2018-08-20 07:01:46	107.77.212.175	us-west-1	WRITE	http://acs.amazonaws.com/groups/s3/LogDelivery	frothlywebcode	bstoll
2018-08-20 07:01:46	107.77.212.175	us-west-1	FULL_CONTROL		frothlywebcode	bstoll

TABULAR VIEW OF CLOUDTRAIL LOGS SHOWING THE PUBLIC READ / WRITE ACCESS WAS REVOKED

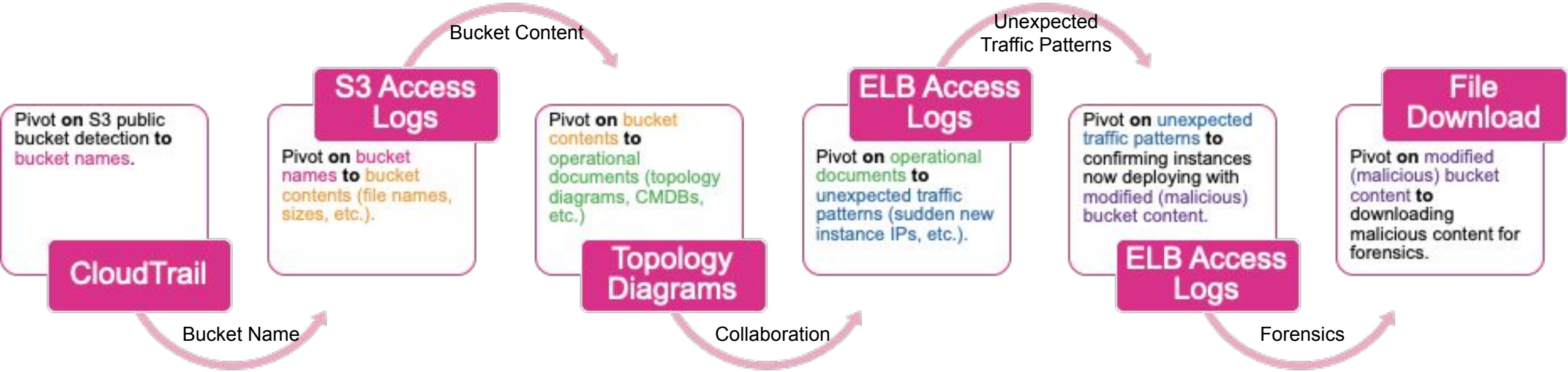
Time	Source IP	Region	Permission	Grantee	Bucket Name	Requester
2018-08-20 07:57:54	107.77.212.175	us-west-1	FULL_CONTROL		frothlywebcode	bstoll
2018-08-20 07:57:54	107.77.212.175	us-west-1	READ	http://acs.amazonaws.com/groups/s3/LogDelivery	frothlywebcode	bstoll
2018-08-20 07:57:54	107.77.212.175	us-west-1	READ_ACP	http://acs.amazonaws.com/groups/s3/LogDelivery	frothlywebcode	bstoll
2018-08-20 07:57:54	107.77.212.175	us-west-1	WRITE	http://acs.amazonaws.com/groups/s3/LogDelivery	frothlywebcode	bstoll
2018-08-20 07:57:54	107.77.212.175	us-west-1	FULL_CONTROL		frothlywebcode	bstoll

Recovery: Reverting the Website Code

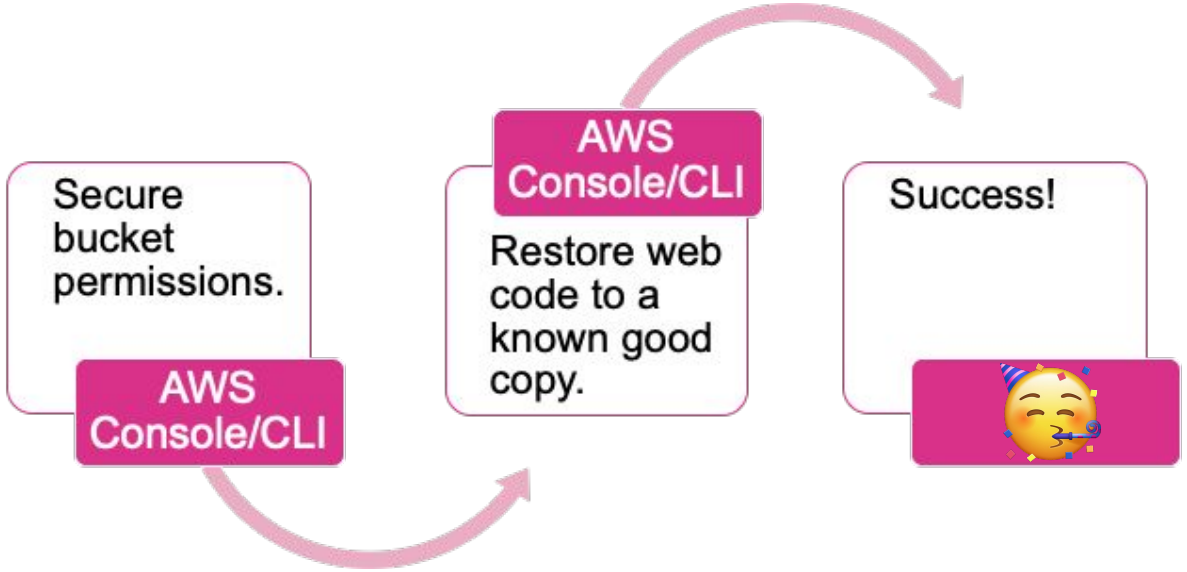
Confirmed by S3 Access Logs

Time	Source IP	Requester	Operation	Key	HTTP Status	Bytes Sent	Object Size	User Agent
2018-08-20 07:02:44	52.66.146.128	-	REST.PUT.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt	200	-	377	Boto3/1.7.62 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 Botocore/1.8.12
2018-08-20 07:03:46	35.182.246.222	-	REST.GET.OBJECT	OPEN_BUCKET_PLEASE_FIX.txt	200	377	377	aws-cli/1.14.8 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 botocore/1.8.12
2018-08-20 07:03:46	35.182.246.222	-	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.8 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 botocore/1.8.12
2018-08-20 07:04:17	35.182.246.222	-	REST.PUT.OBJECT	frothly_html_memcached.tar.gz	200	-	3076532	Boto3/1.7.61 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 Botocore/1.8.12
2018-08-20 07:33:35	54.183.247.244	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-0cc93bade2b3cba63	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13
2018-08-20 07:59:21	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3076532	3076532	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
2018-08-20 08:19:19	107.77.212.175	arn:aws:iam::622676721278:user/bstoll	REST.PUT.OBJECT	frothly_html_memcached.tar.gz	200	-	3057116	
2018-08-20 08:23:32	54.67.37.214	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-06fea586f3d3c8ce8	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	
2018-08-20 08:25:32	52.53.233.88	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-09cbc261e84259b54	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	
08:23:32		role/EC2InstanceRole/i-06fea586f3d3c8ce8						botocore/1.8.13
2018-08-20 08:25:32	52.53.233.88	arn:aws:sts::622676721278:assumed-role/EC2InstanceRole/i-09cbc261e84259b54	REST.GET.OBJECT	frothly_html_memcached.tar.gz	200	3057116	3057116	aws-cli/1.14.9 Python/2.7.14 Linux/4.14.47-56.37.amzn1.x86_64 botocore/1.8.13

Investigation Summary



Remediation Summary



Key Takeaways

- Don’t forget about core security practices when it comes to securing your data (e.g., least privilege access).
- Use a continuous monitoring and reporting solution to detect changes made by internal users AND your vendors.
- Access logging is the key to identifying misconfigured S3 buckets.
- Securing data stored in the cloud is a shared responsibility.

Thank You

splunk> turn data into doing™