SANS

CLOUD SECURITY

RESOURCES

- sans.org/cloud-security
- SANS Cloud Security
- @SANSCloudSec
- SANS Cloud Security
- Webcasts
- Blogs

SEC488: Cloud Security Essentials
License To Learn Cloud Security

SEC510: Public Cloud Security: AWS, Azure, and GCP
Multiple Clouds Require Multiple Solutions

SEC522: Defending Web Applications Security Essentials
Not a matter of "if" but "when". Be prepared for a web app attack. We'll teach you how.

SEC534: Secure DevOps: A Practical Introduction
Principles! Practices! Tools! Oh My! Start your journey on the DevSecOps road here.

SEC540: Cloud Security and DevSecOps Automation
The cloud moves fast. Automate to keep up.

SEC541: Cloud Monitoring and Threat Detection
Attackers can run, but not hide! Our radar sees all threats.

SEC557: Continuous Automation for Enterprise and Cloud Compliance
Using Cloud and DevOps Tools to Measure Security and Compliance

SEC584: Cloud Native Security: Defending Containers and Kubernetes
Deploy Scurely at the Speed of Cloud Native

SEC588: Cloud Penetration Testing
Aim your arrows to the sky and penetrate the Cloud.

FOR509: Enterprise Cloud Forensics and Incident Response
Find the Storm in the Cloud

MGT516: Managing Security Vulnerabilities: Enterprise & Cloud
Stop treating the symptoms. Cure the disease.

MGT520: Leading Cloud Security Design & Implementation
Building and leading a cloud security program

Review our Job Role Flight Plan at sans.org/cloud-security

SANS

Attack and Defend

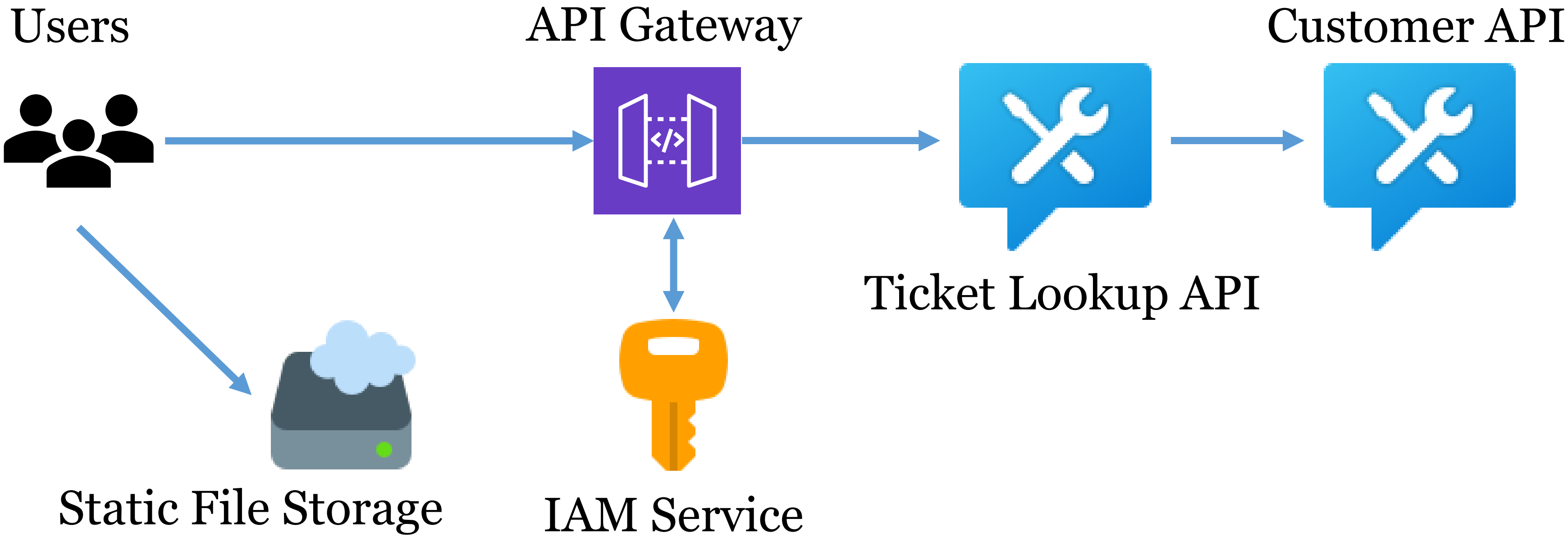
The Dangers of Modern Distributed Applications

Jason Lam and Johannes Ullrich

Agenda

- Walk through of a typical modern application
- Scenario 1 – Microservice exposure
- Scenario 2 – Magic credential
- Scenario 3 – Remote 3rd Party Content
- Lessons Learned

What the application looks like

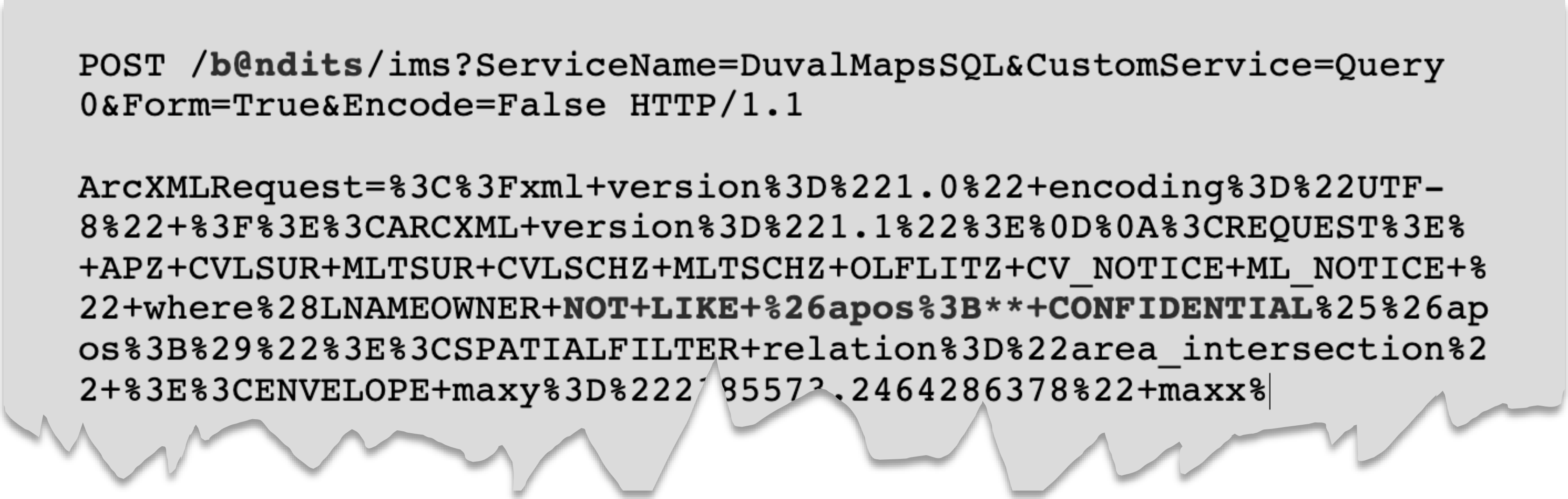


Demo

Application Walkthrough

Attacker – Microservices Exposure / Identifying Target

Browser’s Developer Tools give a lot of insight about Web applications



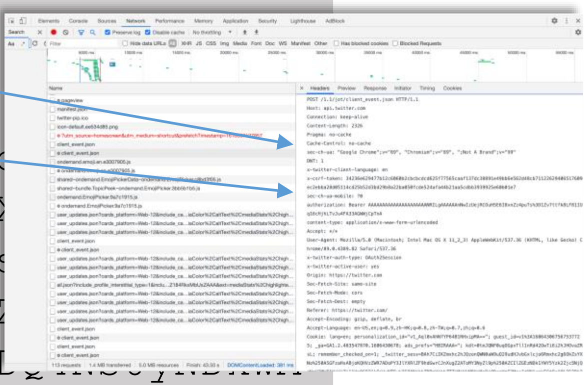
```
POST /b@ndits/ims?ServiceName=DuvalMapsSQL&CustomService=Query
0&Form=True&Encode=False HTTP/1.1

ArcXMLRequest=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22UTF-
8%22+%3F%3E%3CARCXML+version%3D%221.1%22%3E%0D%0A%3CREQUEST%3E%
+APZ+CVLSUR+MLTSUR+CVLSCHZ+MLTSCHZ+OLFLITZ+CV_NOTICE+ML_NOTICE+%
22+where%28LNAMEOWNER+NOT+LIKE+%26apos%3B**+CONFIDENTIAL%25%26ap
os%3B%29%22%3E%3CSPATIALFILTER+relation%3D%22area_intersection%2
2+%3E%3CENVELOPE+maxy%3D%222185573.246286378%22+maxx%
```

Attack and Defend: The Dangers of Modern Distributed Applications

Attacker - Identifying Target

Attempting to Connect to API with existing credentials

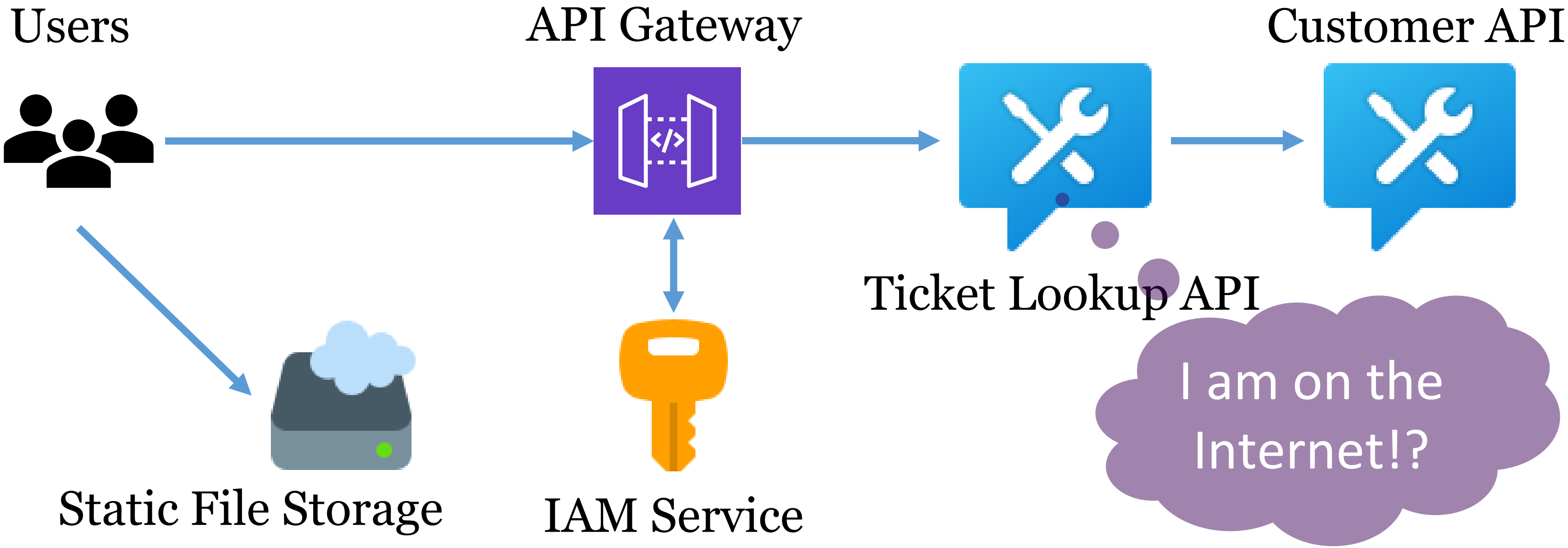
```
POST /1.1/jot/client_event.json HTTP/1.1
Host: api.example.com
….
Authorization: Bearer
eyJhbGciOiJIUzM4NCIsInR5cCI6IkpXVCJ9.eyJpc
czovL2FwaS5leG1hcGxlLmNvbSIsImF1ZCI6InVzZX
XksIHByb2RfbG9va3VwLCBzaGlwcGluZ19jb3N0LCs
zY29wZSI6InNlY3JldDpyZWFkIGJ1eS5yZWFkIHVzZ
ZXIud3JpdGUiLCJub25jZSI6IjAzOTQ4NTItMzE5ML
zU4In0.DPyHdPpU9uBb8TNGl8buF8kvz7J0ctqklSpiQM1MqG…
```



```
POST /1.1/jot/client_event.json HTTP/1.1
Host: api.twitter.com
Connection: keep-alive
Content-Length: 2326
Pragma: no-cache
Cache-Control: no-cache
sec-ch-ua: "Google Chrome";v="89", "Chromium";v="89", ";Not A Brand";v="99"
DNT: 1
x-twitter-client-language: en
x-csrf-token: [illegible]
sec-ch-ua-mobile: ?0
authorization: Bearer AAAAAAAAAAAAAAAAAAAAANRILgAAAAAA[illegible]
q16cMjXLTvJu4FA33AGWWjCpTnA
content-type: application/x-www-form-urlencoded
Accept: */*
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_3) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/89.0.4389.82 Safari/537.36
x-twitter-auth-type: OAuth2Session
x-twitter-active-user: yes
Origin: https://twitter.com
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://twitter.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-HK;q=0.8,zh-TW;q=0.7,zh;q=0.6
Cookie: [illegible]
```

Attack and Defend: The Dangers of Modern Distributed Applications

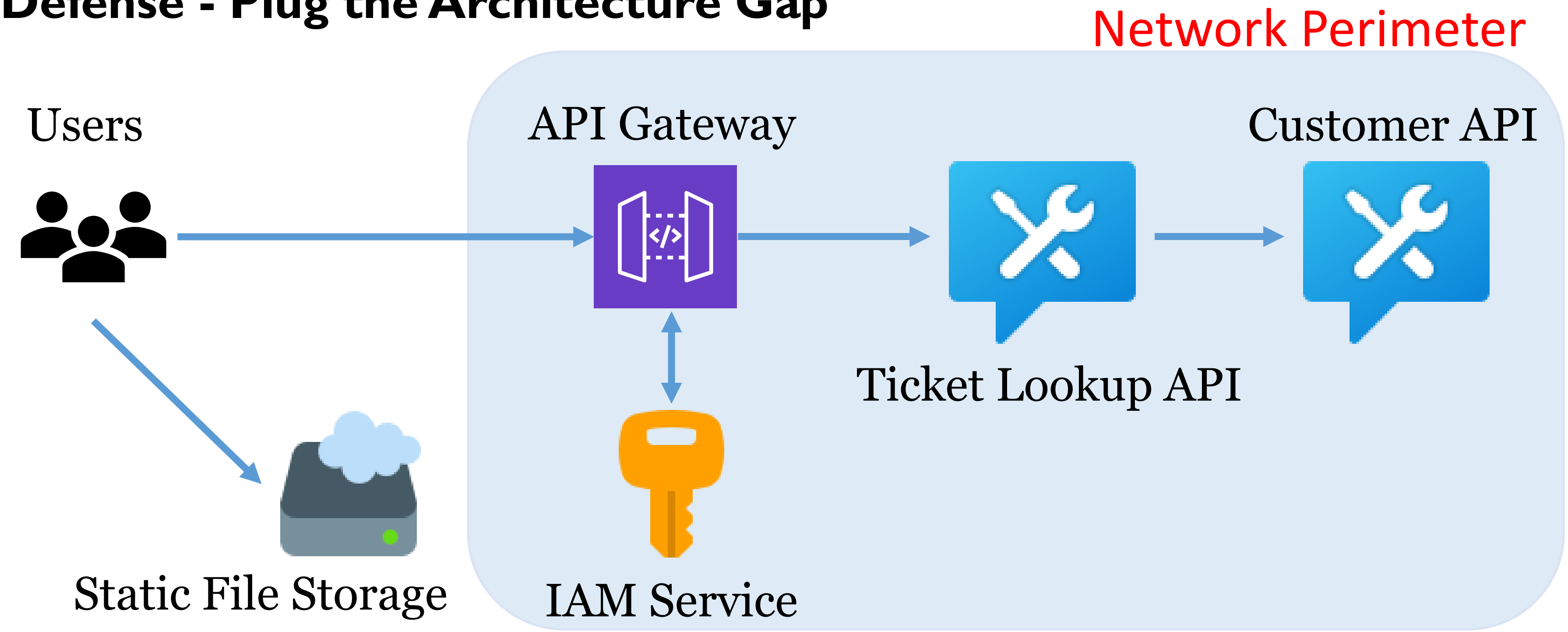
Attacker - Attacking Target



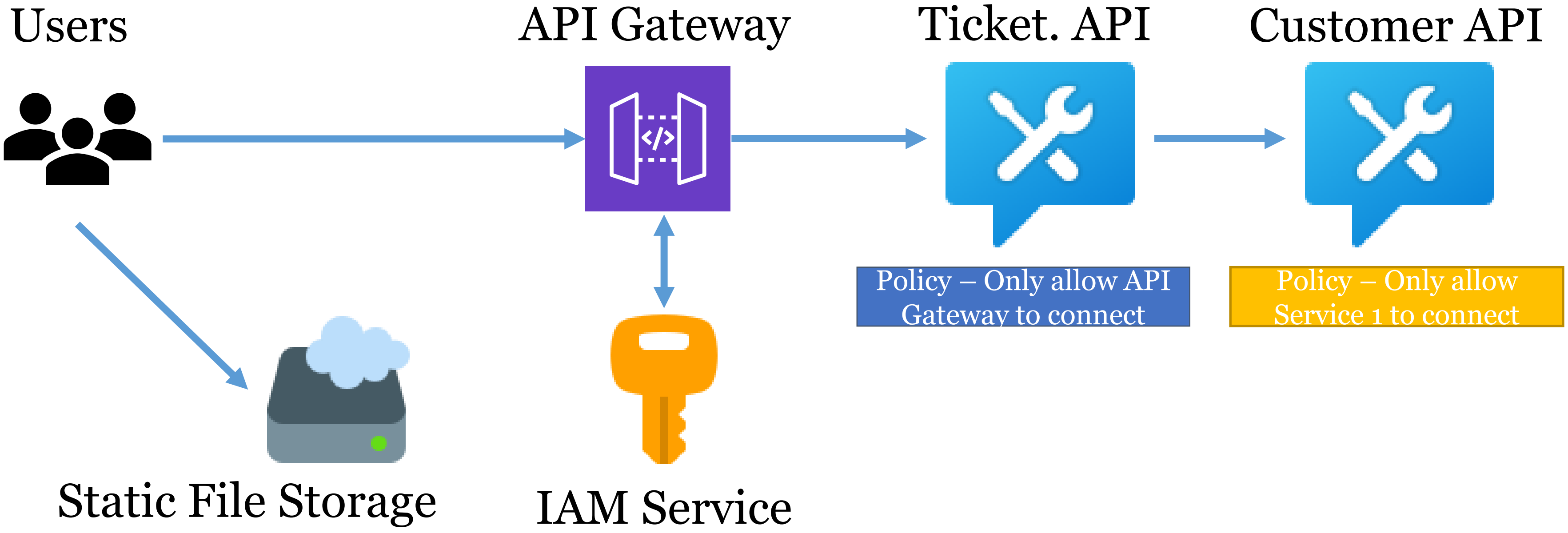
Demo

Bypassing API Gateway

Defense - Plug the Architecture Gap



Defense - Configuration Game


Users
API Gateway
Ticket. API
Customer API
Policy – Only allow API Gateway to connect
Policy – Only allow Service 1 to connect
Static File Storage
IAM Service

Defense - Configuration Game Example

```
"Statement": [
{
          "Sid": "Access-to-specific-VPC-only",
          "Principal": "*",
          "Action": "s3:*",
          "Effect": "Allow",
          "Resource": [
                    "arn:aws:s3:::my_secure_bucket",
                    "arn:aws:s3:::my_secure_bucket/*"                        ],
          "Condition": {
                    "StringNotEquals": {
                              "aws:sourceVpc": "vpc-111bbb22"
                    }
          }
}  ]
```

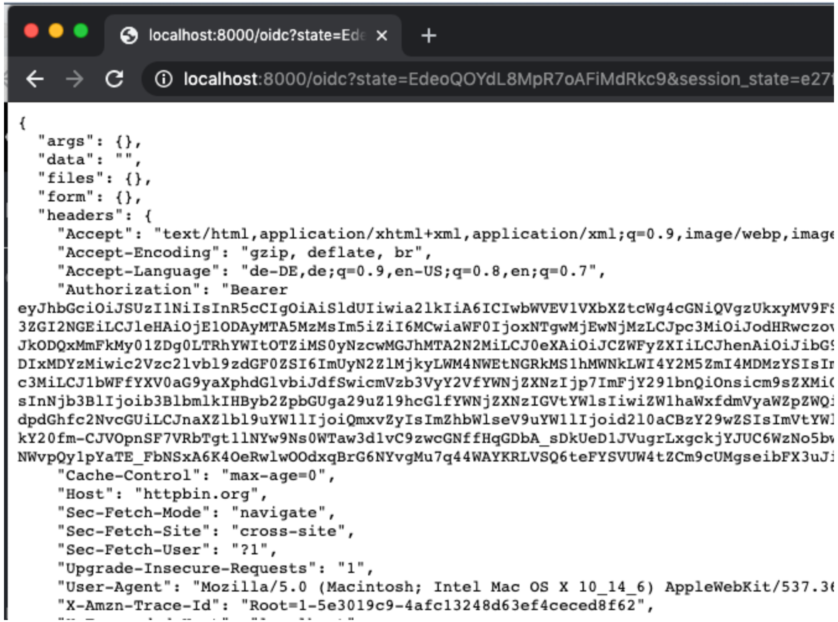
Specify the source

Attacker - What About "Bearer Tokens"

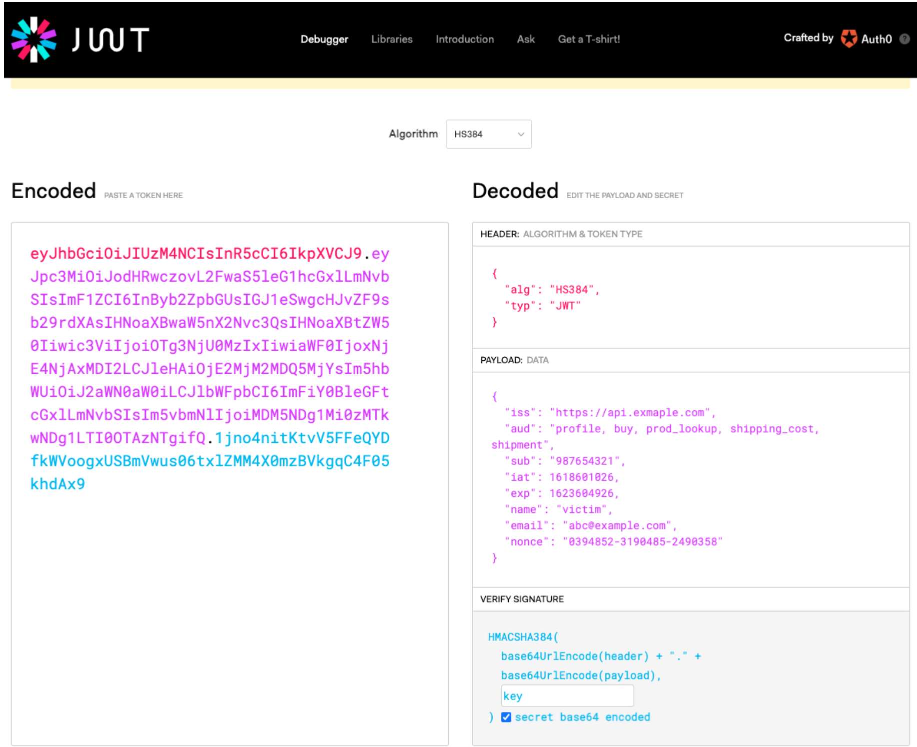
- "Magic Credentials"
- Authentication, Access Control, AND MORE
- Standard format => Easy to parse/use
- Signed token to represent "claims" and securely transmitting them between parties
- JWT is often used as a bearer token in OAuth

Attack and Defend: The Dangers of Modern Distributed Applications

Attacker – What's This JWT?


localhost:8000/oidc?state=Ed
localhost:8000/oidc?state=EdeoQOYdL8MpR7oAFiMdRkc9&session_state=e27

```
{
  "args": {},
  "data": "",
  "files": {},
  "form": {},
  "headers": {
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image
    "Accept-Encoding": "gzip, deflate, br",
    "Accept-Language": "de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7",
    "Authorization": "Bearer
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICIwbWVEV1VXbXZtcWg4cGNiQVgzUkxyMV9FS
3ZGI2NGEiLCJleHAiOjE1ODAyMTA5MzMsIm5iZiI6MCwiaWF0IjoxNTgwMjEwNjMzLCJpc3MiOiJodHRwczov
JkODQxMmFkMy01ZDg0LTRhYWItOTZiMS0yNzcwMGJhMTA2N2MiLCJ0eXAiOiJCZWFyZXIiLCJhenAiOiJibG9
DIxMDYzMiwic2Vzc2lvbl9zdGF0ZSI6ImUyN2ZlMjkyLWM4NWEtNGRkMS1hMWNkLWI4Y2M5ZmI4MDMzYSIsIn
c3MiLCJ1bWFfYXV0aG9yaXphdGlvbiJdfSwicmVzb3VyY2VfYWNjZXNzIjp7ImFjY291bnQiOnsicm9sZXMiO
sInNjb3BlIjoib3BlbmlkIHByb2ZpbGUga29uZ19hcGlfYWNjZXNzIGVtYWlsIiwiZW1haWxfdmVyaWZpZWQi
dpdGhfc2NvcGUiLCJnaXZlbl9uYW1lIjoiQmxvZyIsImZhbWlseV9uYW1lIjoid2l0aCBzY29wZSIsImVtYWl
kY20fm-CJVOpnSF7VRbTgt1lNYw9Ns0WTaw3dlvC9zwcGNffHqGDbA_sDkUeD1JVugrLxgckjYJUC6WzNo5bw
NWvpQy1pYaTE_FbNSxA6K4OeRwlwOOdxqBrG6NYvgMu7q44WAYKRLVSQ6teFYSVUW4tZCm9cUMgseibFX3uJi
    "Cache-Control": "max-age=0",
    "Host": "httpbin.org",
    "Sec-Fetch-Mode": "navigate",
    "Sec-Fetch-Site": "cross-site",
    "Sec-Fetch-User": "?1",
    "Upgrade-Insecure-Requests": "1",
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36
    "X-Amzn-Trace-Id": "Root=1-5e3019c9-4afc13248d63ef4ceced8f62",
```



Algorithm HS384

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzM4NCIsInR5cCI6IkpXVCJ9.ey
Jpc3MiOiJodHRwczovL2FwaS5leG1hcGxlLmNvb
SIsImF1ZCI6InByb2ZpbGUsIGJ1eSwgcHJvZF9s
b29rdXAsIHNoaXBwaW5nX2Nvc3QsIHNoaXBtZW5
0Iiwic3ViIjoiOTg3NjU0MzIxIiwiaWF0IjoxNj
E4NjAxMDI2LCJleHAiOjE2MjM2MDQ5MjYsIm5hb
WUiOiJ2aWN0aW0iLCJlbWFpbCI6ImFiY0BleGFt
cGxlLmNvbSIsIm5vbmNlIjoiMDM5NDg1Mi0zMTk
wNDg1LTI0OTAzNTgifQ.1jno4nitKtvV5FFeQYD
fkWVoogxUSBmVwus06txlZMM4X0mzBVkgqC4F05
khdAx9
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS384",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "iss": "https://api.exmaple.com",
  "aud": "profile, buy, prod_lookup, shipping_cost,
shipment",
  "sub": "987654321",
  "iat": 1618601026,
  "exp": 1623604926,
  "name": "victim",
  "email": "abc@example.com",
  "nonce": "0394852-3190485-2490358"
}
```

VERIFY SIGNATURE

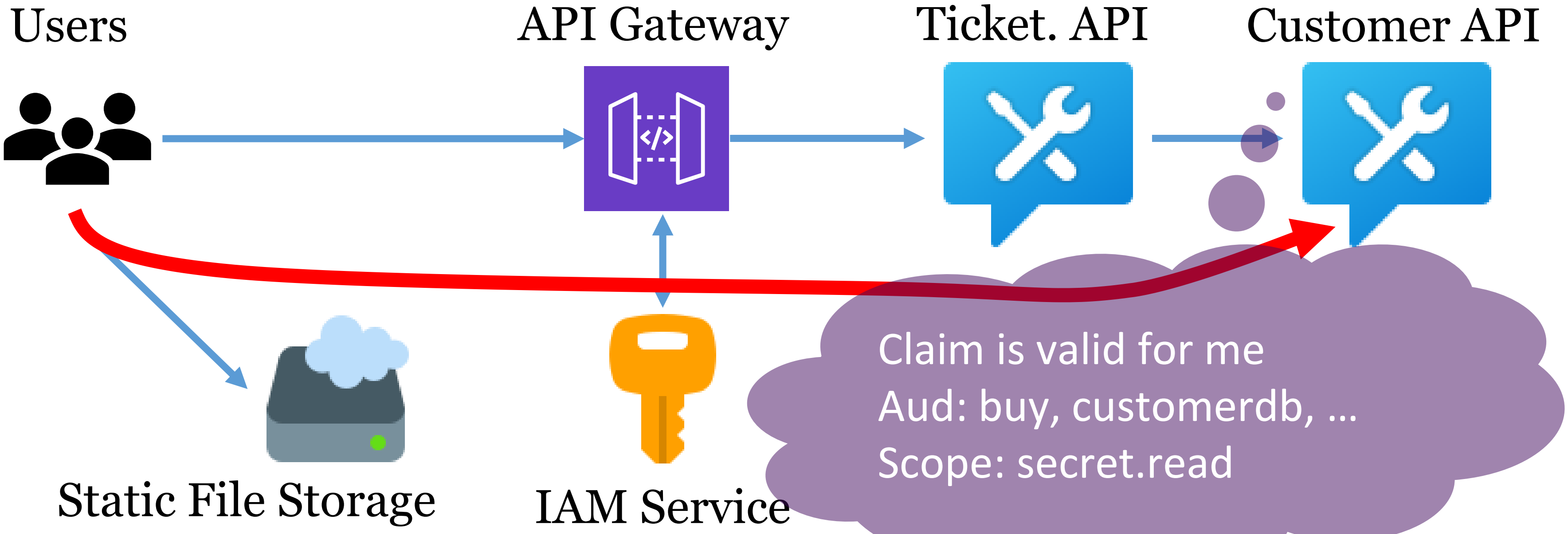
```
HMACSHA384(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  key
) ☑ secret base64 encoded
```

Attacker – Inspecting the JWT Claim

- iat and exp – the time that the token is valid
- aud – recipients this token is for
- Scope – determine the range of access granted
- There is a signature to guard against changes

iss:	54321
name:	abc
email:	abc@example.com
iat:	1618601026
exp:	1623604926
aud:	["usermgmt", "buy"…]
scope:	secret:read

Attacker – Attacking Other APIs

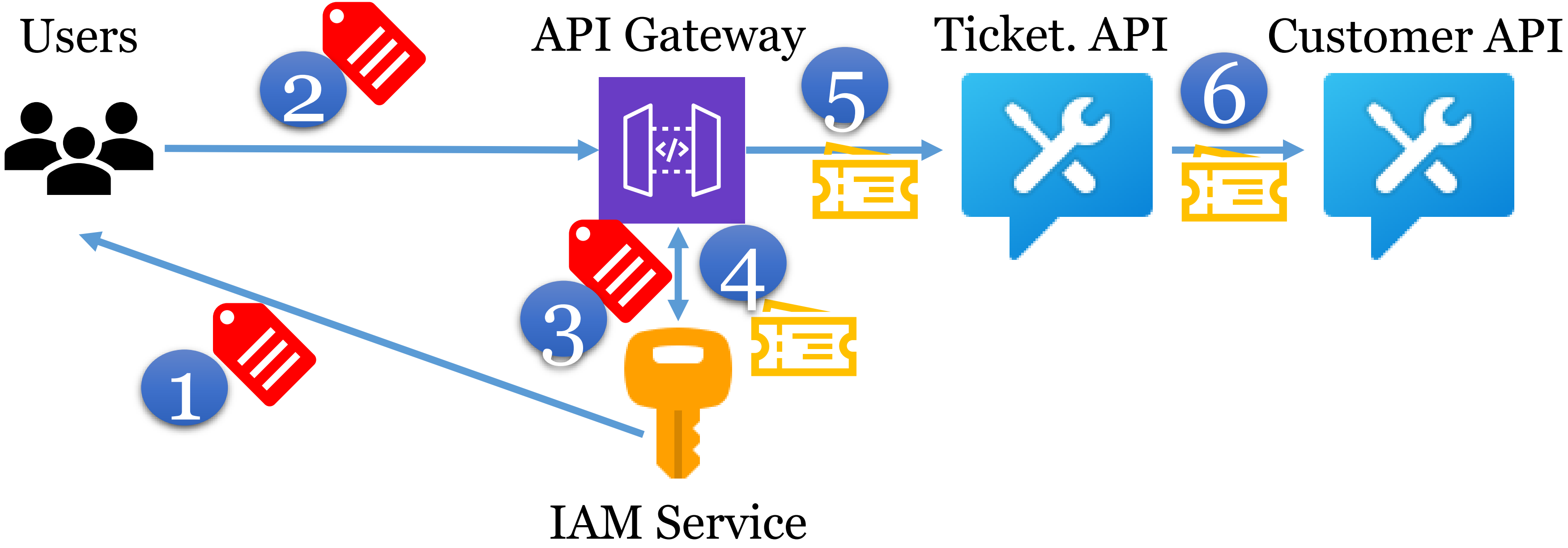


Claim is valid for me
Aud: buy, customerdb, …
Scope: secret.read

Demo

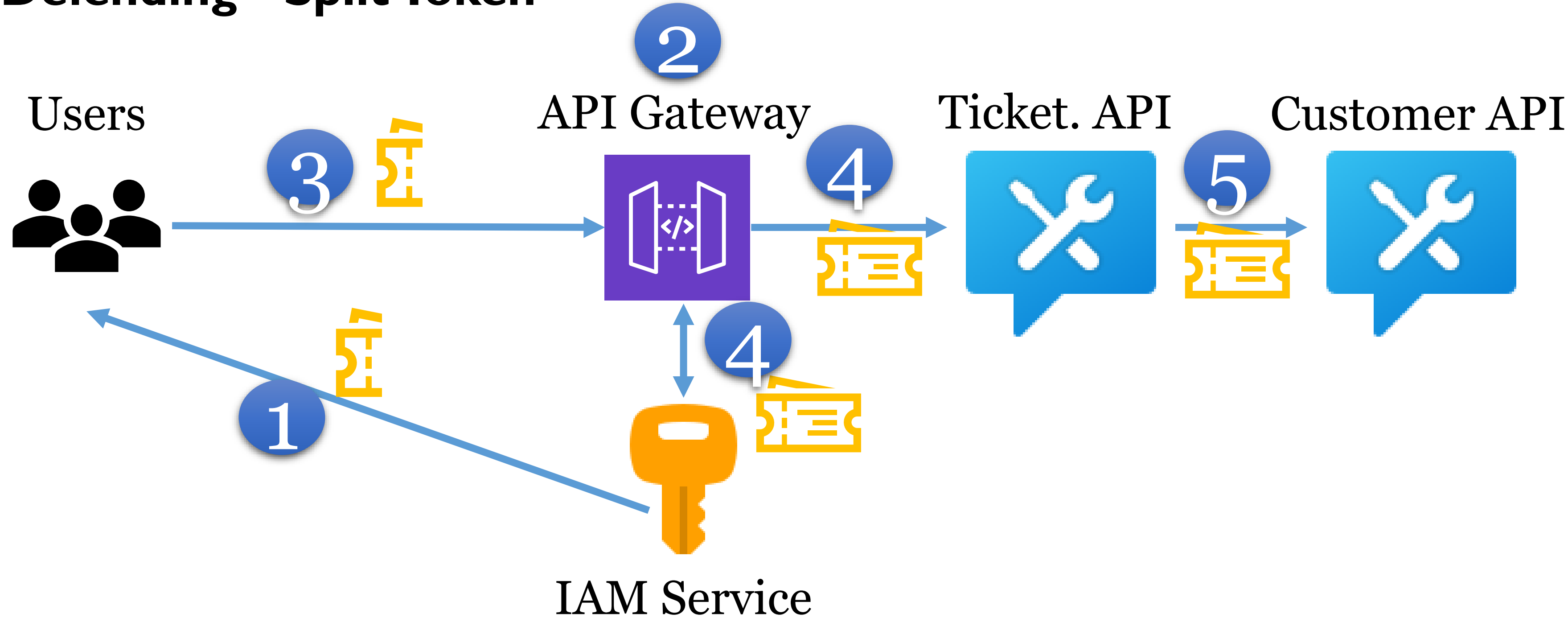
Exploring JWT Claims

Defending – Phantom Token



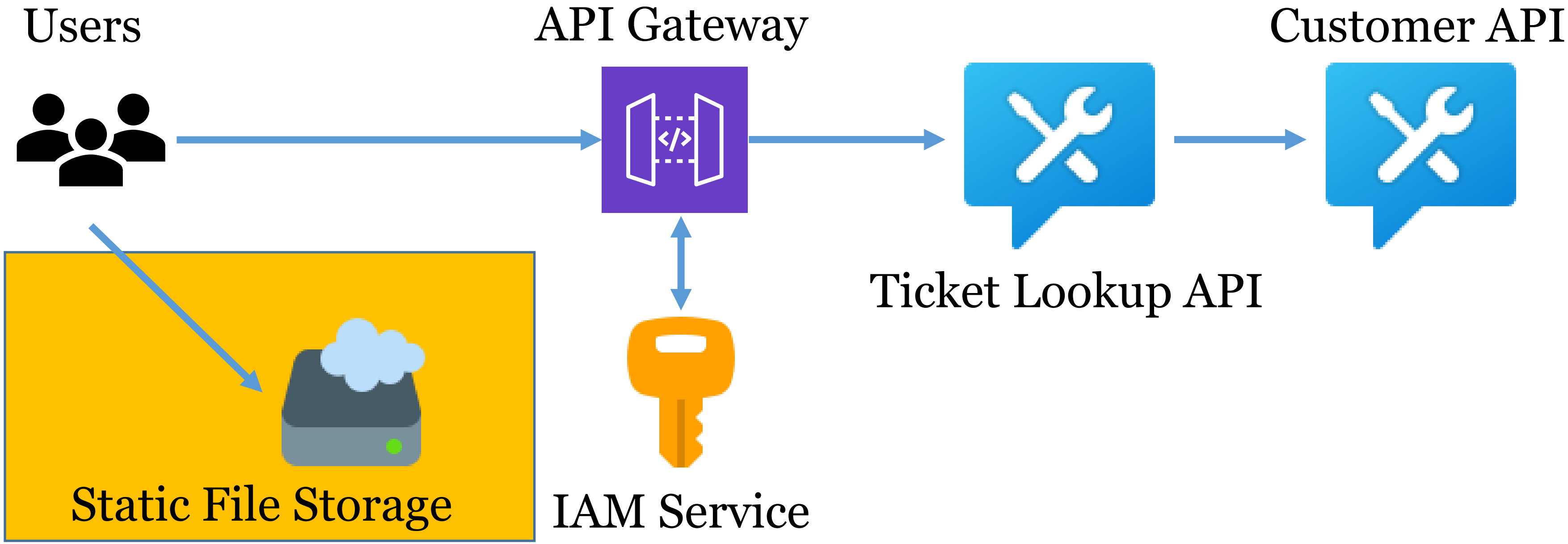
Attack and Defend: The Dangers of Modern Distributed Applications

Defending - Split Token

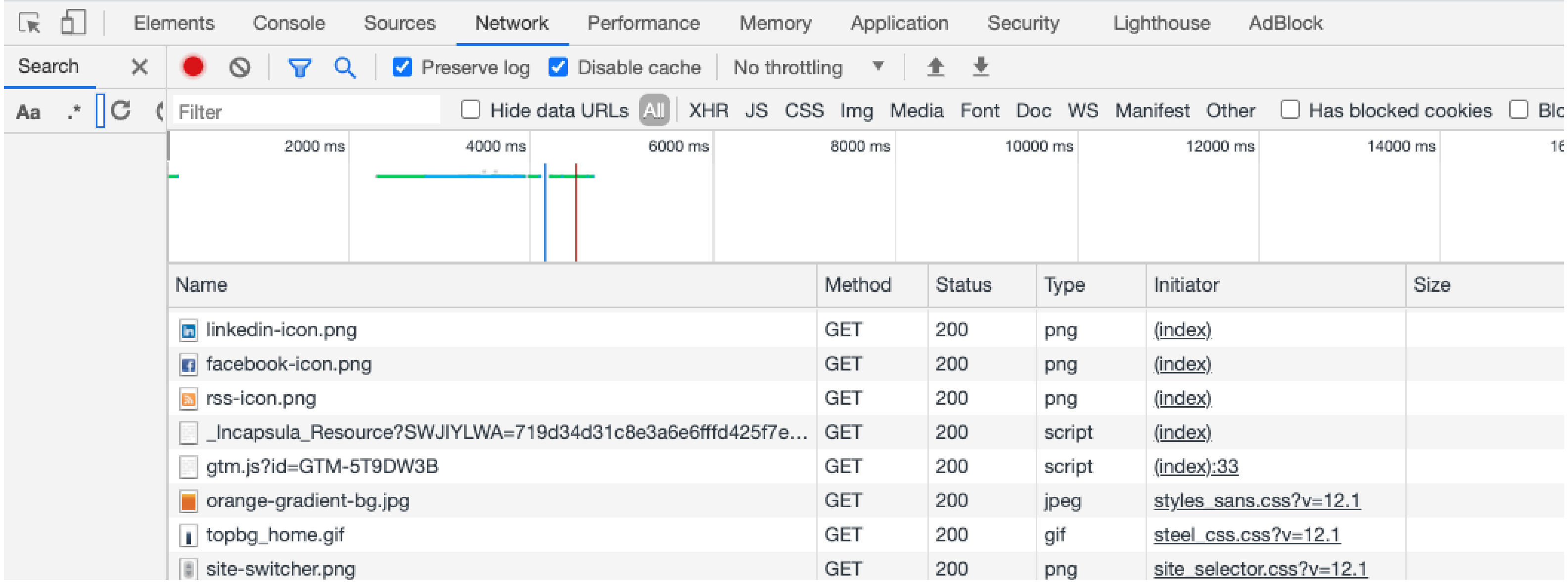


Attack and Defend: The Dangers of Modern Distributed Applications

Attacker - But what about that static content?



Attacker - Using Browser to find 3rd Party Dependencies



☑ Preserve log ☑ Disable cache No throttling

□ Hide data URLs □ Has blocked cookies □ Blo

Name	Method	Status	Type	Initiator	Size
linkedin-icon.png	GET	200	png	(index)	
facebook-icon.png	GET	200	png	(index)	
rss-icon.png	GET	200	png	(index)	
_Incapsula_Resource?SWJIYLWA=719d34d31c8e3a6e6fffd425f7e…	GET	200	script	(index)	
gtm.js?id=GTM-5T9DW3B	GET	200	script	(index):33	
orange-gradient-bg.jpg	GET	200	jpeg	styles_sans.css?v=12.1	
topbg_home.gif	GET	200	gif	steel_css.css?v=12.1	
site-switcher.png	GET	200	png	site_selector.css?v=12.1	

Demo

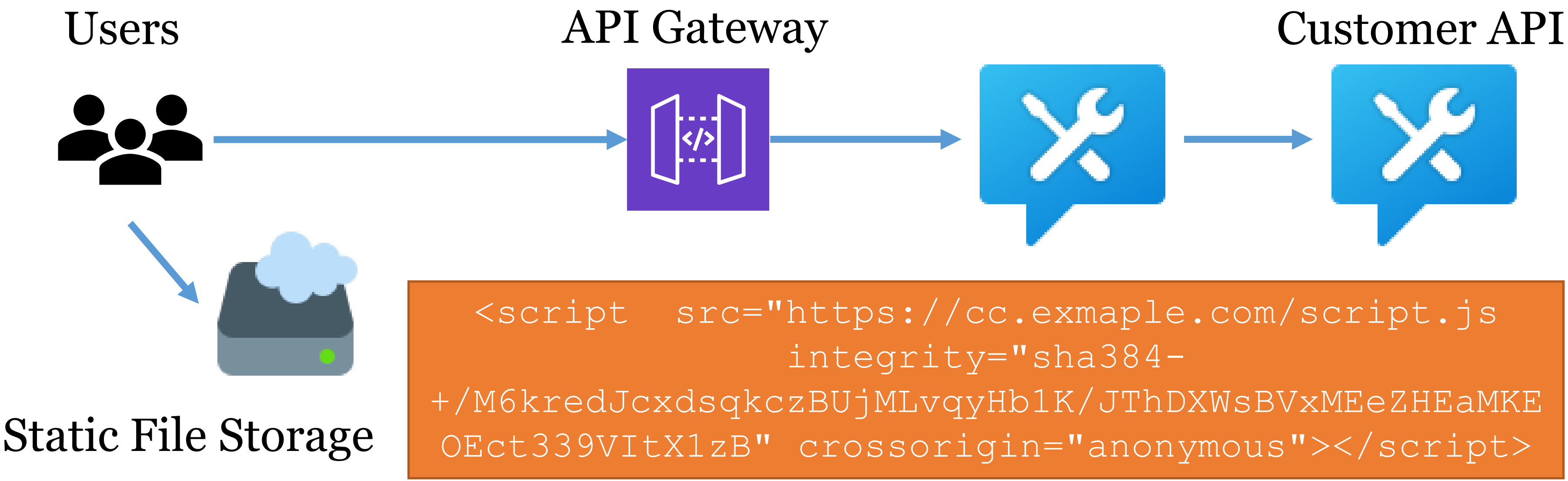
Manipulating 3rd Party Resource

Defense - Validating Remote Content

- Tricky to validate content you don't own
- Subresource Integrity (SRI)
  - In HTML - Specify an integrity check value for a remote resource
  - Browser will not load the remote content if integrity check does not match
  - Guard against unauthorized change

Attack and Defend: The Dangers of Modern Distributed Applications

Attacker - But what about that static content?



```
<script  src="https://cc.exmaple.com/script.js
integrity="sha384-
+/M6kredJcxdsqkczBUjMLvqyHb1K/JThDXWsBVxMEeZHEaMKE
OEct339VItX1zB" crossorigin="anonymous"></script>
```

Apply What You Have Learned Today

Next month:

- Review your modern application's architecture
- Review Cloud components' configuration
- Understand credential flow in applications

Next 6 months:

- Develop credential handling guidelines and reference architecture in microservice/API based applications

Attack and Defend: The Dangers of Modern Distributed Applications

Questions?

Thank You!

Slides and a Recording will be made available

Any Questions?

Jason Lam
@jasonlam_sec
jlam@sans.org

Johannes Ullrich
@johullrich
jullrich@sans.edu