The Perils of SaaS Collaboration
Where are the data and who has access?

Max Feldman
5/27/21
Agenda

- Intro
- Background
- Collaboration Issues
- Tabletop Exercise
- Risk Management
- Incident Management
- Q&A
Max Feldman
Security Engineering Manager
Background
The evolution of collaboration
Pre-Pre-Cloud

- In-person
- Letters
- Telegram
- Telephone
Pre-Cloud

Mainframe

Local Network/On-Prem

Version Control

USB

Email

Internet
Cloud

Internet Grows

SaaS Begins

Docs/Code/Data via Browser
Cloud

- Salesforce
- GSuite
- M365
- Slack
- Zoom
- Github
- Workday
- ServiceNow
- ...

...
Early on, SaaS applications were accessed primarily through an enterprise network.
SaaS apps grew into complex platforms with an unlimited number of access points.
Impact of COVID and WFH on SaaS Adoption

- **10.4%** No Increase in SaaS Adoption
- **21.4%** Already heavy SaaS Adopter
- **68.2%** Increased SaaS Adoption

90% Enterprises are heavy Saas adopters or have increased SaaS adoption
Threat Models
The evolution of threats to data
## Evolution of threats: Pre-Pre-Cloud

<table>
<thead>
<tr>
<th>Pre-Pre-Cloud</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-person</td>
<td>Theft</td>
</tr>
<tr>
<td>Letters</td>
<td>Forgery</td>
</tr>
<tr>
<td>Telegram</td>
<td>Pirates</td>
</tr>
<tr>
<td>Telephone</td>
<td>Phone lines</td>
</tr>
<tr>
<td></td>
<td>Eavesdropping</td>
</tr>
<tr>
<td></td>
<td>Wrong number</td>
</tr>
<tr>
<td></td>
<td>Wrong address</td>
</tr>
</tbody>
</table>
# Evolution of threats: Pre-Cloud

<table>
<thead>
<tr>
<th>Pre-Cloud</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mainframes</td>
<td>Hacks</td>
</tr>
<tr>
<td>Local networks/on-premise</td>
<td>Viruses</td>
</tr>
<tr>
<td>Version control</td>
<td>Modern computer security threats</td>
</tr>
<tr>
<td>USB Stick</td>
<td>Organized crime</td>
</tr>
<tr>
<td>Email</td>
<td>APTs</td>
</tr>
<tr>
<td>Internet</td>
<td></td>
</tr>
</tbody>
</table>
# Evolution of threats: Cloud

<table>
<thead>
<tr>
<th>Cloud</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salesforce</td>
<td>The regular threats of doing anything with computers, plus:</td>
</tr>
<tr>
<td>GSuite</td>
<td>Shadow IT</td>
</tr>
<tr>
<td>M365</td>
<td>Misconfiguration</td>
</tr>
<tr>
<td>Slack</td>
<td>Misunderstanding of the platforms</td>
</tr>
<tr>
<td>Zoom</td>
<td>Third-party apps connected to the SaaS apps</td>
</tr>
<tr>
<td>Github</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
The complex nature of the SaaS platforms introduced new security risks.

- 3rd party apps connected to the average SaaS environment. 42
- 3rd Party Apps
- Are no longer used, but still have access to sensitive data. 22
- Contractors, MSPs
- External Users
- IoT
- Updates
The complex nature of the SaaS platforms introduced new security risks.

- 3rd party apps connected to the average SaaS environment: 42
- 3rd Party Apps: 22
- Contractors, MSPs
- External Users
- IoT
- Updates

+95% Companies with external users that have been over-provisioned.
The complex nature of the SaaS platforms introduced new security risks.

- **+95%** Companies with external users that have been over-provisioned.
- **+55%** Companies with sensitive data publicly exposed to the anonymous internet.

- **42** 3rd party apps connected to the average SaaS environment.
- **22** 3rd party apps are no longer used, but still have access to sensitive data.

Contractors, MSPs

External Users

IoT

Updates

SaaS Environment
The complex nature of the SaaS platforms introduced new security risks.

- **+55%**
  Companies with sensitive data publicly exposed to the anonymous internet.

- **+95%**
  Companies with external users that have been over-provisioned.

- **42**
  3rd party apps connected to the average SaaS environment.

- **22**
  3rd Party Apps
  Are no longer used, but still have access to sensitive data.

- **>200**
  Updates across all SaaS apps.
This is how companies are now being attacked.

Jan 21, 2021
Hacker leaks data of millions of Teespring users

What happened:
Teespring had previously evaluated a 3rd party service called Waydev which required access to some of its data. This access was implemented via OAuth.

Waydev retained the OAuth token for Teespring. A third party later accessed the token to gain access to the Teespring infrastructure.

Dec 26, 2020
Rakuten exposes 1.48 million sets of data to access from outside

What happened:
Rakuten failed to notice a change in Salesforce security settings that occurred as a part of an update in January 2016.

The update left data accessible to the public internet for nearly 5 years until it was discovered in November 2020.

Jan 19, 2021
SolarWinds attack opened up 4 separate paths to a Microsoft 365 Cloud Breach

What happened:
The perpetrators behind the SolarWinds supply chain attack were observed leveraging four separate techniques to bypass identity and access management protections and laterally move from victims’ on-premises networks to their cloud-based Microsoft 365 accounts.
Real-world Scenarios
Incidents involving or resulting from SaaS
Public Google Sheet
An easy mistake
Public Google Sheet

- Company hired third party to build websites for them
- Third party put Drupal, Akamai, New Relic creds in Google Sheet
- Google Sheet was public
- Finding those would have allowed RCE and access to everything for all those domains
What went wrong

- Putting keys in a spreadsheet
- Making that spreadsheet public
Customer Support Portal

Over-exposed data
How Can We Help?

Request Something

Knowledge Base

Ask the Community

CURRENT STATUS

No current system issues

POPULAR QUESTIONS

Why does my screen freeze when creating a new user?

When will the new update be available?

MY OPEN INCIDENTS

Unable to access group file folder.

Having trouble launching new features.

Having trouble launching new features.

More
### Case Table Permissions:
- **Agents** can Read, Write, Create, and Delete
- **Support Portal Users** can Read, Write, and Create

<table>
<thead>
<tr>
<th>Id</th>
<th>CreatedDate</th>
<th>AssignedTo</th>
<th>PortalUserId</th>
<th>Title</th>
<th>Description</th>
<th>Priority</th>
<th>Status</th>
<th>Comments</th>
<th>EscalationRequired</th>
</tr>
</thead>
<tbody>
<tr>
<td>100678</td>
<td>04/01/2021</td>
<td>mberry</td>
<td>502345</td>
<td>Frozen Screen!</td>
<td>My screen completely froze. I was</td>
<td>3</td>
<td>Closed</td>
<td>Try following this Knowledgebase No</td>
<td>No</td>
</tr>
<tr>
<td>100679</td>
<td>04/01/2021</td>
<td>tcolicchio</td>
<td>502949</td>
<td>My phone won't power on</td>
<td>I just bought this phone and it won’t power on.</td>
<td>2</td>
<td>In Progress</td>
<td>I’m sorry your new phone isn’t working No</td>
<td>No</td>
</tr>
<tr>
<td>100680</td>
<td>04/01/2021</td>
<td>tdesjardina</td>
<td>501183</td>
<td>Battery won’t hold a charge I’ve noticed that my battery seems to die faster than usual.</td>
<td>2</td>
<td>Closed</td>
<td>I’ve scheduled a service appointment No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>100681</td>
<td>04/01/2021</td>
<td>mberry</td>
<td>500059</td>
<td>Weird errors when I browse I am getting error messages that say</td>
<td>1</td>
<td>In Progress</td>
<td>I’ve escalated your case to engin Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>100662</td>
<td>04/01/2021</td>
<td>tcolicchio</td>
<td>501994</td>
<td>Having trouble launching</td>
<td>Hi. I’m not sure if I’m doing something wrong</td>
<td>2</td>
<td>Open</td>
<td></td>
<td>No</td>
</tr>
</tbody>
</table>
Javascript is querying the API to fetch case details.

We can replay a valid API query, but tamper with the id parameter.
There is something else interesting about the response. We can try the same technique, but query for Users instead of Cases.
What the outside world *SHOULD* see
What the outside world *DID* see

<table>
<thead>
<tr>
<th>TARGET OBJECT</th>
<th># OF ACCESSIBLE RECORDS</th>
<th>CREATE</th>
<th>READ</th>
<th>EDIT</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachment</td>
<td>&gt;100k</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>email_message</td>
<td>&gt;100k</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>user</td>
<td>425,513</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>—</td>
</tr>
<tr>
<td>emailtemplate</td>
<td>24,257</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>faq</td>
<td>1,079</td>
<td>—</td>
<td>✔️</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>patch_note</td>
<td>731</td>
<td>—</td>
<td>✔️</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>news_feed</td>
<td>658</td>
<td>—</td>
<td>✔️</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>contentdoc</td>
<td>317</td>
<td>—</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>contentdistro</td>
<td>311</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Any support portal user could see any data belonging to any other support portal user.
What went wrong

- Portal Administrators confused UI visibility with access control
- No security change management process for SaaS
- Security team lacked domain expertise in the access control and data sharing models of the SaaS application
Open S3 Buckets
Analogous to SaaS misconfiguration
Open S3 Buckets

- Very common
- A powerful tool, but can be misconfigured
- IaaS, but countless parallels between IaaS and SaaS security concerns
  - SaaS trailing a few years but catching up
  - Biggest parallel: Shared Responsibility Model
Open S3 Buckets

- Data exposed unintentionally, no hacking required
- During the transition to cloud, it can be easy to forget SRM
  - Happens with IaaS, happens with SaaS
- [https://github.com/nagwww/s3-leaks](https://github.com/nagwww/s3-leaks) - some examples
Open S3 Buckets

- “Someone broke into our unsecured AWS S3 silo, added 'non-malicious' code to our JavaScript SDK”
- "Exposed AWS buckets again implicated in multiple data leaks"
  - Passport scans, tax documents, background checks, job applications, expense claims, contracts, emails and salary details relating to thousands of consultants working in the UK were exposed.
- "7.2 million records were exposed, but not from the ... app"
- “Misconfigured database breaches thousands of ... patient files names, email and postal addresses, phone numbers, dates of birth and Social Security numbers...”
Open S3 Buckets

- “How a Vendor for Half the Fortune 100 Exposed a Terabyte of Backups”
- “Medical Records and Patient-Doctor Recordings Were Exposed information for employees of 181 business locations, as well as personally identifiable information (PII) for nearly 3,000 individuals was publicly exposed in an unsecured”
- “Jewelry site accidentally leaks personal details (and plaintext passwords!) of 1.3M users addresses, zip-codes, e-mail addresses, and IP addresses. He also claims the database contained plaintext passwords”
- This just takes us to 2018
Open S3 Buckets

- Lots of automation/tools exist around this now
  - Easy for IaaS-specific tools, *hard* for people
- Now: more common knowledge, expected
  - But they still happen
What went wrong

- Forgetting shared responsibility model
- Not including these concerns into design, deployment
- Security team doesn’t always have domain expertise
- Not making use of automation
  - This was harder in the beginning
Tabletop exercise

Talk through a hypothetical breach
Third-party App Breach

- **Scenario:**
  - You’re the sole security person at your organization
  - “Schedule.ly”* is a small, but popular productivity app
  - It has integrations with all the cool SaaS applications (Salesforce, etc.)
  - You find out from a news article that it’s been breached

- **What do you do?**

* not a real app
Third-party App Breach

- Does anyone in your organization actually use Schedule.ly?
  - How do you know/how can you be sure?
  - Do you have inventory of every third-party app?
    - Even if it’s a third-party app connected to a third-party app?

- You have a vendor review process for all purchases and software installs, but this app is free and able to install with just a click
  - Are your employees able to install apps into Salesforce? Slack? O365? GSuite?
Third-party App Breach

- You’re agile and anyone can install anything anywhere
- How do you find where it might be installed?
  - Do your SaaS apps have pages listing installed apps?
  - Do you need admin access to view?
  - Who are your admins?
Third-party App Breach

- Have your admins investigate:
  - It's installed in O365, Salesforce, Slack, and GSuite

- OK. Now how do you disconnect it from all of them to contain the breach?
  - Uninstall flow for each SaaS app is fairly straightforward
  - Access revoked
Third-party App Breach

- It’s no longer installed in O365, Salesforce, Slack, and GSuite
- OK. What data could it access in each?
Third-party App Breach

- What data could it access?
  - What permissions is it granted in each app?
  - What do those permissions mean?
- Hundreds of scopes and permissions (per platform, even)
  - Some grant specific access, some broad
  - Is the app installed at the system level or per-user?
OAuth Scopes to API methods

Learn even more detail about these OAuth scopes [here](#).

<table>
<thead>
<tr>
<th>OAuth Scope</th>
<th>Associated Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin.analytics:read</td>
<td>admin.analytics.getFile</td>
</tr>
<tr>
<td>admin.applications:read</td>
<td>admin.applications.approved.list</td>
</tr>
<tr>
<td>admin.applications:read</td>
<td>admin.applications.requests.list</td>
</tr>
<tr>
<td>admin.applications:write</td>
<td>admin.applications.approve</td>
</tr>
<tr>
<td>admin.applications:write</td>
<td>admin.applications.clearResolution</td>
</tr>
<tr>
<td>admin.applications:write</td>
<td>admin.applications.restrict</td>
</tr>
<tr>
<td>admin.barriers:read</td>
<td>admin.barriers.list</td>
</tr>
<tr>
<td>admin.barriers:write</td>
<td>admin.barriers.create</td>
</tr>
<tr>
<td>admin.barriers:write</td>
<td>admin.barriers.delete</td>
</tr>
<tr>
<td>admin.barriers:write</td>
<td>admin.barriers.update</td>
</tr>
<tr>
<td>admin.invites:read</td>
<td>admin.inviteRequests.approved.list</td>
</tr>
<tr>
<td>admin.invites:read</td>
<td>admin.inviteRequests.denied.list</td>
</tr>
<tr>
<td>admin.invites:read</td>
<td>admin.inviteRequests.list</td>
</tr>
</tbody>
</table>
## Books API, v1

### Scopes

<table>
<thead>
<tr>
<th>Scope</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://www.googleapis.com/auth/books">https://www.googleapis.com/auth/books</a></td>
<td>Manage your books</td>
</tr>
</tbody>
</table>

## Calendar API, v3

### Scopes

<table>
<thead>
<tr>
<th>Scope</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://www.googleapis.com/auth/calendar">https://www.googleapis.com/auth/calendar</a></td>
<td>See, edit, share, and permanently delete all the calendars you can access using Google Calendar</td>
</tr>
<tr>
<td><a href="https://www.googleapis.com/auth/calendar.events">https://www.googleapis.com/auth/calendar.events</a></td>
<td>View and edit events on all your calendars</td>
</tr>
<tr>
<td><a href="https://www.googleapis.com/auth/calendar.events.readonly">https://www.googleapis.com/auth/calendar.events.readonly</a></td>
<td>View events on all your calendars</td>
</tr>
<tr>
<td><a href="https://www.googleapis.com/auth/calendar.readonly">https://www.googleapis.com/auth/calendar.readonly</a></td>
<td>See and download any calendar you can access using your Google Calendar</td>
</tr>
<tr>
<td><a href="https://www.googleapis.com/auth/calendar.settings.readonly">https://www.googleapis.com/auth/calendar.settings.readonly</a></td>
<td>View your Calendar settings</td>
</tr>
</tbody>
</table>
# OAuth Scopes

The following OAuth scopes can be assigned to the connected app to define the type of protected resources that the client can access.

<table>
<thead>
<tr>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access Pardot services</strong> <em>(pardot_api)</em></td>
<td>Allows access to Pardot API services on behalf of the user. Manage the full extent of accessible services in Pardot.</td>
</tr>
<tr>
<td><strong>Access and manage your Chatter data</strong> <em>(chatter_api)</em></td>
<td>Allows access to Connect REST API resources on behalf of the user.</td>
</tr>
<tr>
<td><strong>Access and manage your Eclair data</strong> <em>(eclair_api)</em></td>
<td>Allows access to the Analytics REST API Charts Geodata resource.</td>
</tr>
<tr>
<td><strong>Access and manage your Wave data</strong> <em>(wave_api)</em></td>
<td>Allows access to the Analytics REST API resources.</td>
</tr>
<tr>
<td><strong>Access and manage your data</strong> <em>(api)</em></td>
<td>Allows access to the current, logged-in user's account using APIs, such as REST API and Bulk API. This scope also includes <em>chatter_api</em>, which allows access to Connect REST API resources.</td>
</tr>
<tr>
<td><strong>Access custom permissions</strong> <em>(custom_permissions)</em></td>
<td>Allows access to the custom permissions in an org associated with the connected app. This scope also shows whether the current user has each permission enabled.</td>
</tr>
<tr>
<td><strong>Access your basic information</strong> <em>(id, profile, email, address, phone)</em></td>
<td>Allows access to the identity URL service. You can request <em>profile</em>, <em>email</em>, <em>address</em>, or <em>phone</em> individually to get the same result as using <em>id</em> because they're synonymous.</td>
</tr>
</tbody>
</table>
## Calendars permissions

### Delegated permissions

<table>
<thead>
<tr>
<th>Permission</th>
<th>Display String</th>
<th>Description</th>
<th>Admin Consent Required</th>
<th>Microsoft Account supported</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Calendars.Read</strong></td>
<td>Read user calendars</td>
<td>Allows the app to read events in user calendars.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Calendars.Read.Shared</strong></td>
<td>Read user and shared calendars</td>
<td>Allows the app to read events in all calendars that the user can access, including delegate and shared calendars.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Calendars.ReadWrite</strong></td>
<td>Have full access to user calendars</td>
<td>Allows the app to create, read, update, and delete events in user calendars.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Calendars.ReadWrite.Shared</strong></td>
<td>Read and write user and shared calendars</td>
<td>Allows the app to create, read, update and delete events in all calendars the user has permissions to access. This includes delegate and shared calendars.</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Third-party App Breach

- You know where it was installed, what scopes it has in each SaaS app
  - i.e. what could have happened
- Now - what *did* happen?
- Schedule.ly could access calendar information, various Salesforce objects, various Slack channels
  - But did the attackers do that?
Third-party App Breach

- Reach out to vendors to ask for help
  - They have security teams, but this could take a while
- Look at logs from each SaaS application
  - How do you view them?
  - What is in them?
Third-party App Breach

- Assume we’ve been able to find out what data was accessed and where
- Now - does that matter?
  - Was it customer data?
  - Company financial data?
  - Employee data?
- What legal or compliance obligations are there?
Third-party App Breach

- How do we prevent this from happening again?
  - Or mitigate damage?
- How do we make the next incident easier?
  - No app is perfectly secure, breaches happen
Supplemental Data
Common Risk Assessment Findings

- **35%+**
  - Security Posture
  - Best Practices
  - Misconfigured

- **30%+**
  - Compliance Violations
  - (NIST CSF)

- **99%+**
  - Data Access Misconfigurations
  - Identified

- **55%+**
  - Public Data Records
  - Exposed in 55%+ RA

- **95%+**
  - Over-permissive Integration Accounts
  - in 95% of RA

- **20+**
  - Avg # of Insecure Secrets (Credentials)
Does this mean we shouldn’t use SaaS?

- No
Questions to start the discussion

How can you evaluate your posture?
Inventory

- What systems exist?
- What data do they hold?
- Who is in charge of them?
- Who can access them?
App Inventory

- What third-party apps exist?
- What are the settings for each SaaS application for installing third-party apps?
  - Should they be more restrictive?
  - Should apps be subject to review?
System Configuration

- What are all the configurable settings?
- What are the best practices?
A Security Double Standard

<table>
<thead>
<tr>
<th>Web Application Security</th>
<th>SaaS Application Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code analysis tools</td>
<td>✓</td>
</tr>
<tr>
<td>Vulnerability scanners</td>
<td>✓</td>
</tr>
<tr>
<td>Security guidelines &amp; best practices</td>
<td>✓</td>
</tr>
<tr>
<td>Web application firewalls</td>
<td>✓</td>
</tr>
<tr>
<td>Runtime protection</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Hope for the best</td>
</tr>
</tbody>
</table>
Recommendations
Steps you can take to collaborate securely
What should security teams be doing to better secure their SaaS?

Prepare your organization for success

1. Embrace the SaaS shared responsibility model
2. Clearly define ownership for SaaS Security
3. Recognize the limited scope of commonly used tools like CASB and pentests
4. Take a proactive approach - don’t wait for a breach to take action
What should security teams be doing to better secure their SaaS?

Prepare your organization for success
1. Embrace the SaaS shared responsibility model
2. Clearly define ownership for SaaS Security
3. Recognize the limited scope of commonly used tools like CASB and pentests
4. Take a proactive approach - don’t wait for a breach to take action

Protect your SaaS data
1. Secure your data, not just the perimeter
2. Gain visibility to who has access to your data
3. Implement guardrails for your users
4. Continuously monitor your SaaS environment
Inventory

- Take inventory ahead of time, and take it regularly
- Stay up-to-date on where data resides and who has access
- Evaluate if users are over-permissioned
- Automate this
Monitoring and Detection

- Understand the logging of your own tools but also SaaS applications
- Monitor these logs as you would others
- Treat SaaS applications with the same intentionality as any other application
Vendor Review

- Have a vendor risk assessment process in place
- Subject everything to it - large SaaS apps, small third-party apps installed into those, etc.
SDL

- Security Development Lifecycle
  - Applies to software development, but can generalize to SaaS deployment
- Powerful SaaS applications can run code or have custom apps or integrations
- Misconfiguration is common - but following a process for secure development and deployment can help
Thank You

To learn more about AppOmni visit www.AppOmni.com.