THE CUSTOMER SUCCESS PLATFORM
SALES SERVICE MARKETING COMMUNITY ANALYTICS APPS

Secure Coding: Storing Secrets In Your Salesforce Instance

Kyle Tobener
Product Security Engineer
@KyleKyle

Maxwell Feldman
Product Security Engineer

dreamforce

Safe Harbor

Safe harbor statement under the Private Securities Litigation Reform Act of 1995:

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.

No Photos Required….

Slides and demos will be made available after the talk!

Primary Topic Today: Secrets

- We will be covering developer-oriented topics on secret storage for the Salesforce Platform

- Specific features to cover include:
  - Secrets in custom fields
  - Secrets in encrypted custom fields
  - Secrets in custom settings

- Useful for anyone in the following areas:
  - Salesforce Developers (primarily)
  - Salesforce Administrators
  - Prospective Partners

What is a secret?

- Simple Definition: A piece of data that requires higher than normal protection

- For Our Purposes: A secret will be a piece of data that nobody should see, like a password or encryption key

Who do we secure secrets from?

- Attackers
- Regular Users
- Partners
- **Administrators (Biggest Challenge)**

Basically everyone… Why?

- Theft of data
- Impersonation
- Privilege escalation

Secret Storage: Custom Field

Custom Field – Storage Method

1. Create an object with a custom field to store secret
2. Make object private
3. Remove CRUD/FLS from all profiles
4. Only access secret field through Apex

Custom Field – Breakdown

Pros

- Simple
- Easily updated
- CRUD is used to prevent most users from seeing the object
- FLS is used to prevent users from seeing the field

Cons

- CRUD is included in many privileged permissions
- FLS can be updated by admins, potentially exposing the secret
- Anyone who can deploy Apex code can discover the secret

Demo: Secrets in Custom Fields

Trivia!

Which permissions bypass the FLS protections safeguarding a secret stored in a custom field? Please choose from the following list:

a) Modify All Data
b) View All Data (Profile)
c) Customize Application
d) Deploy Apex
e) View All Data (Object Specific)

Trivia (answered)!

Which permissions bypass the FLS protections safeguarding a secret stored in a custom field? Please choose from the following list:

a) **Modify All Data**
b) View All Data (Profile)
c) **Customize Application**
d) **Deploy Apex**
e) View All Data (Object Specific)

Secret Storage – Encrypted Custom Field

Encrypted Custom Field – Storage Method

1. Create a new field of type “Text (Encrypted)”
2. Choose a mask type (depending on the secret type)
3. Configure the FLS of the new field such that zero profiles have read access
4. Use Apex to store and access the secret

Note: Some *may consider FLS to be optional since the contents of the field are obscured, but “View Encrypted Data” is a global permission, so any user with this permission could view any public encrypted field. Employing FLS results in the most secure iteration of this storage method.*

Encrypted Custom Field – Breakdown

Pros

- Simple
- Encryption is managed by the platform
- Field is obscured from users without FLS and CRUD being needed

Cons

- View Encrypted Fields profile permission is global, not field specific, and reveals the secret
- Anyone who can deploy Apex code can discover the secret

Demo: Secrets in Encrypted Custom Fields

Trivia!

The following list contains possible ways of viewing the contents of encrypted custom fields. Please tell us which options would show the contents in clear text (no obfuscation) and explain!

a) Stack trace viewer in the developer console
b) Debug log output from `system.debug(object.encryptedField__c);`
c) Workflow field update copying encrypted field to unencrypted field
d) Trigger field update copying encrypted field to unencrypted field
e) Webservice that returns secret as a string

Trivia (answered)!

The following list contains possible ways of viewing the contents of encrypted custom fields. Please tell us which options would show the contents in clear text (no obfuscation) and explain!

a) Stack trace viewer in the developer console
b) Debug log output from `system.debug(object.encryptedField__c);`
c) **Workflow field update copying encrypted field to unencrypted field**
d) **Trigger field update copying encrypted field to unencrypted field**
e) **Webservice that returns secret as a string**

Secret Storage – Managed Protected Custom Setting

Managed Protected Custom Settings – Storage Method

1. Create a managed package
2. Create a protected custom setting inside the package
3. Create a Visualforce page inside the package to create/update the secret
   - (transient string, should not return secret to the view state)
4. Access and use the secret inside the managed package

Custom Settings Overview

Custom settings are stripped down sObjects exposed to the application cache, enabling efficient access for developers.

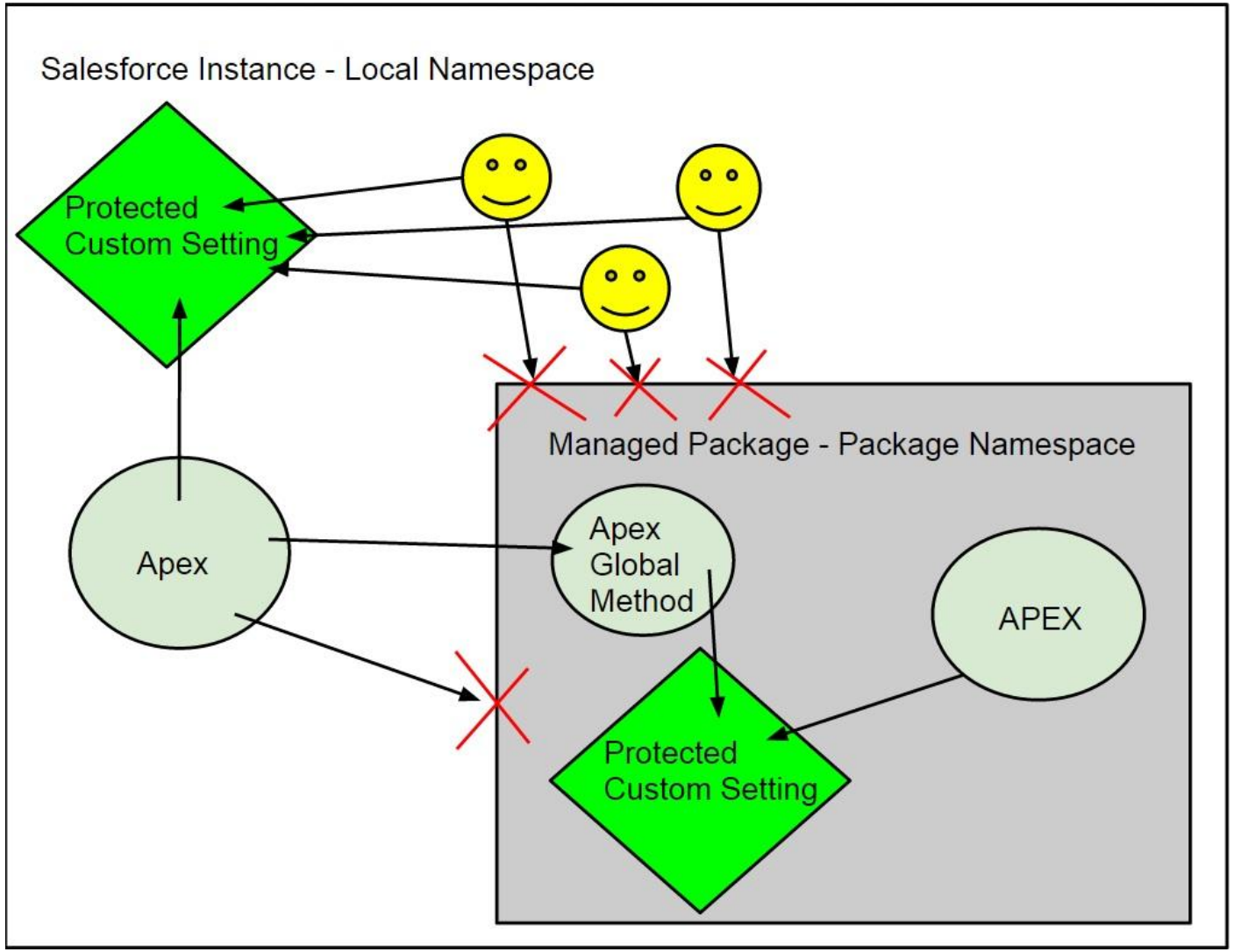
Managed Protected versus Unmanaged Protected: What is the difference?

Protected Custom Settings can only be accessed from the namespace they exist in.

- In a managed package, the namespace is that of the package
- In an unmanaged package, the namespace is the local namespace

What does this mean? Managed protected custom settings offer security benefits, while unmanaged protected custom settings are worse than regular sObjects (because they lack FLS and CRUD settings).

Custom Setting Diagram



Managed Protected Custom Setting – Breakdown

Pros

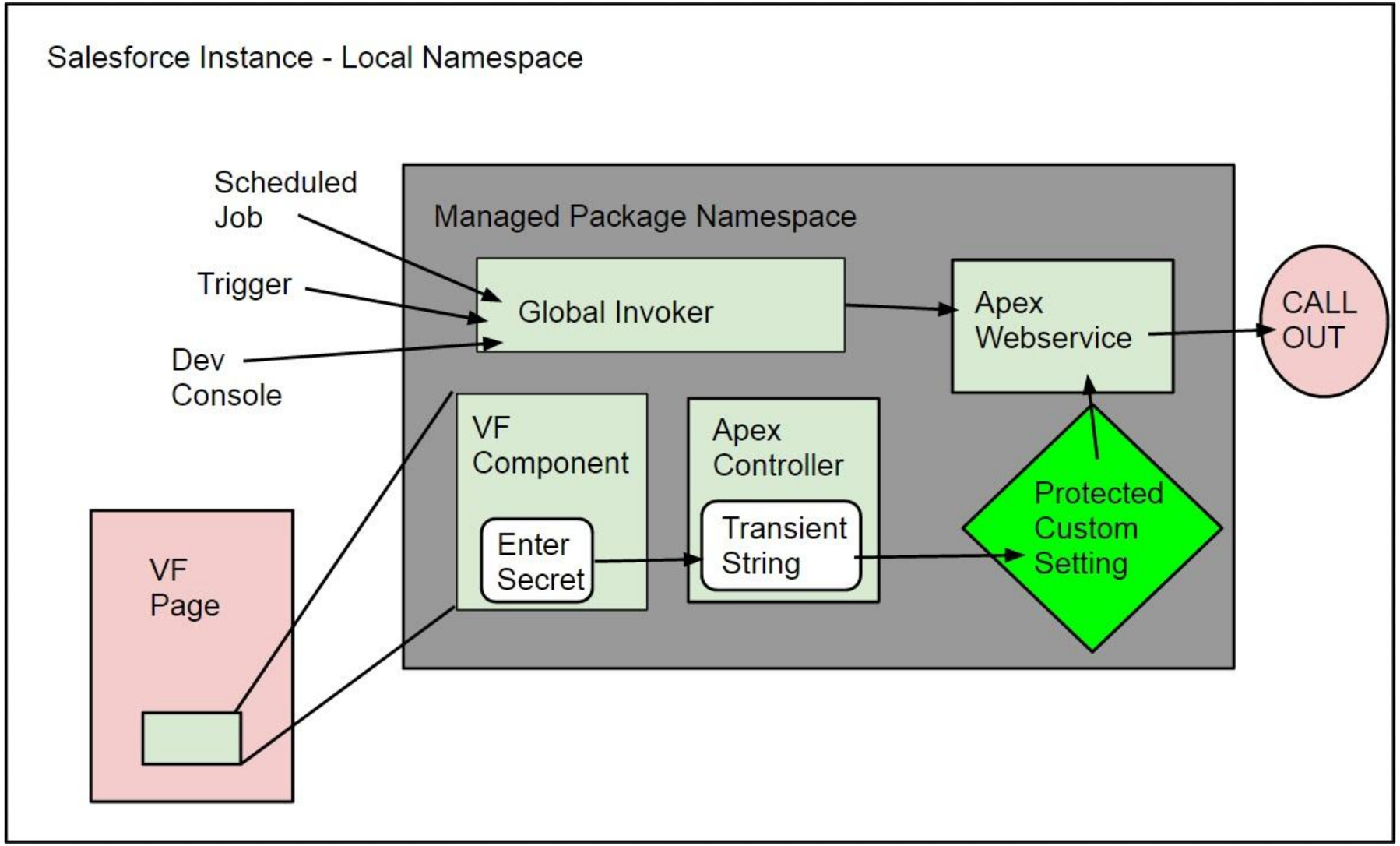
- Secret only available to Apex code within managed package namespace
- Can store encryption key to scale

Cons

- Requires a managed package!
- Methods must be well-coded to prevent secret exposure

Demo: Secrets in Custom Settings

Managed Package Architecture



Trivia!

"Can you see any problems with how the following implementation that uses a managed protected custom setting to store the password for an external callout?"

```
1   global void basicAuthCallout(string url){
2     HttpRequest req = new HttpRequest();
3     req.setEndpoint(url);
4     String pw = customSetting.getAll().values()[0];
5     String authorizationHeader = 'BASIC '
6      +EncodingUtil.base64Encode(Blob.valueOf('admin :'+pw));
7     req.setHeader('Authorization', authorizationHeader);
8     Http http = new Http();
9     HTTPResponse res = http.send(req);
10  }
```

Trivia (answered)!

Accepting a URL from outside the managed package permits leakage of the secret!

The URL should originate from within the package or be tied to the secret.

```
1   global void basicAuthCallout(){
2     HttpRequest req = new HttpRequest();
3     req.setEndpoint('https://api.somewhere.com');
4     String pw = customSetting.getAll().values()[0];
5     String authorizationHeader = 'BASIC '
6      +EncodingUtil.base64Encode(Blob.valueOf('admin :'+pw));
7     req.setHeader('Authorization', authorizationHeader);
8     Http http = new Http();
9     HTTPResponse res = http.send(req);
10  }
```

Recap

Here are the forms of secret storage that we covered:

1. Custom Field
   - Pro – Simple. FLS & CRUD prevents most user access
   - Con – Can be bypassed by users with elevated permissions (Modify All Data, Author Apex)
   - ❖ Works well with: Sensitive data with no encryption requirements
2. Encrypted Custom Field
   - Pro – More secure than basic custom fields. Prevents most user access. Supports masking options
   - Con – Can be bypassed by users with elevated permissions (Modify All Data, Author Apex)
   - ❖ Works well with: Sensitive data with masking or encryption requirements
3. Managed Protected Custom Setting (Secret Storage Best Practice)
   - Pro – Most secure option. Protects against users with elevated permissions such as Modify all Data
   - Con – Requires a managed package. Requires careful attention to code
   - ❖ Works well with: Passwords, oAuth Tokens, Encryption Keys

Additional Resources

- Secure Coding Guidelines - https://developer.salesforce.com/page/Secure_Coding_Storing_Secrets
- Intro to Managed Packages - https://developer.salesforce.com/page/An_Introduction_to_Packaging
- Salesforce StackExchange - http://salesforce.stackexchange.com/questions/tagged/security
- Developer.Salesforce.com Security Forum - https://developer.salesforce.com/forums (full link hidden)
- Security Office Hours (Partners) - http://security.force.com/security/contact/ohours
- Security Implementation Guide - https://developer.salesforce.com/./securityImplGuide/ (full link hidden)

Slides + Demo

- Get Slides Here:
  - DF Chatter Group – Link Here
  - @kylekyle Twitter – https://www.twitter.com/kylekyle

- Want to play with our demo code?
  - Dreamforce Demo Trial Signup: https://security.secure.force.com/DFtrialsignup

Secure Development Sessions

Secure Coding: Field-level Security, CRUD, and Sharing
Monday, October 13 @ 11:00 a.m. - 11:40 a.m.

Secure Coding: Storing Secrets in Your Salesforce Instance
Monday, October 13 @ 2:00 p.m. - 2:40 p.m.

Building Secure Mobile Apps
Monday, October 13 @ 5:00 p.m. - 5:40 p.m.

Protect Your Data Against Malicious Scripts
Tuesday, October 14 @ 11:00 a.m. - 11:40 a.m.

Secure Coding: External App Integration
Wednesday, October 15 @ 9:00 a.m. - 9:40 a.m.

Secure Coding: SSL, SOAP, and REST
Thursday, October 16 @ 10:30 a.m. - 11:10 a.m.

dreamforce

Announcements:

Force.com Code Scanner now supports Salesforce1 and JavaScript! Try it here: http://bit.ly/SF1Scanner

Chimera Web App Scanner alpha nominations are open. Partners apply at: http://bit.ly/SFChimera

Live security office hours are available in the Partner Zone.

salesforce

Q&A

Thank You