Third-Party Risk in Salesforce Named Credentials

By: Aaron Costello, Offensive Security Engineer at AppOmni

Introduction

The Salesforce platform provides multiple methods for secure storage of secrets at a time when the modern digital workforce needs it most. Integration with external systems is now commonplace, as businesses have become increasingly distributed across multiple cloud platforms. However, the risk of both internal and external data exposure incidents increases significantly when collaborating with larger teams. As recently as 2019, it was revealed that more than 100,000 API keys and secrets were made available to the public through Github (https://www.zdnet.com/article/over-100000-github-repos-have-leaked-api-or-cryptographic-keys/) alone.

This article provides an overview of Named Credentials (https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_callouts_named_credentials.htm), a feature introduced by Salesforce in the Spring '15 release to combat the issue of hardcoded credentials within an organization's Apex codebase. We'll also introduce a novel attack vector against Named Credentials by malicious third party extensions that AppOmni presented to Salesforce earlier this year. And we'll cover the recent additions Salesforce Engineering made to their Event Monitoring Product that allow detection of such attacks.

Named Credentials: An Overview

Prior to any technical discussion, it's important to establish a basic understanding of why this functionality was introduced to a product that already had Protected Custom Settings (https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_customsettings.htm) and Encrypted Fields.

The primary purpose of Named Credentials is to provide a managed and secure way of allowing Apex developers to authenticate to external systems, alleviating the need to embed sensitive authentication information such as tokens or credentials in their code. In some aspects, this is comparable to the functionality provided by Google's Secrets Manager, or Terraform Vault.

(https://appomni.com/wp-content/uploads/2021/09/Basic-Named-Credential-Usage.jpg)

Basic Named Credential Usage

In the above diagram, the following flow takes place in order to connect to the external system:

1. An Apex developer references a Named Credential via a callout label in their Apex code, and invokes it.
2. The corresponding endpoint and authentication information is retrieved from Named Credential storage, automatically inserted into the HTTP request, and the request is performed to the external system.
3. The external system provides a response body, which is later parsed in the Apex code for desired information.

While the above execution flow remains consistent when leveraging any Named Credential, the largest differentiator between two different pairs is their individual configurations. Two different external systems may be utilising different authentication protocols or require different certificates. It's also possible that one needs no authentication for the purpose of fetching public static resources, such as images from an S3 bucket.

But in the context of this article, the primary point of interest is the ability to define a Named Credential with an Identity Type of "Named Principal". When configured to use this specific type of identity, all Apex code referencing this Named Credential from a callout will leverage the single credential defined within the configuration. This is as opposed to utilising credentials unique to each individual end-user, which is a separate Named Credential Identity Type.

Interestingly, with respect to "Named Principal" Named Credentials, the protection of sensitive information returned from endpoints relies on the fact that Apex must be explicitly written to interact with the external system. While the authoring of Apex code directly in a Salesforce org requires the privileged "Author Apex" permission, vendor-created extensions, called "packages" in the Salesforce ecosystem, commonly contain third-party Apex. Additionally, the behavior of the Salesforce platform is to hide the source code of any Apex contained in a managed package from customers, preventing customer audits of third-party code. This behavior brings us to the next section, outlining how this capability of third-party packages can take advantage of Named Credentials within your organization.

Named Credentials & Third-Party Risks

Within the Salesforce ecosystem, third parties owning and distributing packages have full control over the Apex code that is bundled with their managed package. Since customers can't audit and review third-party Apex code, there is little additional protection that can be implemented to prevent Named Credentials declared with "Named Principal" from being accessed by any installed package.

Below is a diagram that contains one variation of such an attack:

(https://appomni.com/wp-content/uploads/2021/09/Named-Credential-Access-by-Malicious-Package.jpg)

Named Credential Access by Malicious Package

Upon installation of the managed package, the "Post-Install" script is triggered, creating an Apex Scheduled Job (1) within the instance. After a predefined period of time, this job triggers a static method within an Apex payload (2), which is also bundled into the package. The Apex payload itself can trivially enumerate existing Named Credential definitions within the subscriber org (3), and once successful, has full access to the HTTP response returned by the external system (4).

The impact of the above attack scenario is heightened by the fact that the path defined within the Named Credential can be extended, and full control over the HTTP method is in the hands of the Apex code, allowing for state-changing request via POST/PUT/PATCH/DELETE methods. As a result, the integrity of the external system is also affected, along with confidentiality. A more complex actor may attempt to fingerprint the external system within the Apex by analysing the response headers and body against known signatures of commonly used services (such as an S3 bucket). In doing so, service-specific payloads can be leveraged in an effort to move laterally to the external system, depending on the level of privilege granted by the credentials in use and endpoints exposed by the service.

As a best practice, an organization with many third-party packages should look to leverage managed protected custom settings as an alternative to Named Credentials. When contained in a managed package, protected Custom Settings and their values are invisible to Apex outside of the same containing package. If an organization wishes to continue the pattern of utilising Named Credentials, then it is advised to reconfigure the Identity Type for endpoints hosting sensitive data to “Per User”, while continuing to use “Named Principal” for retrieving non-sensitive information such as static assets.

As the leading SaaS Security Posture Management platform, AppOmni presented this and other risk scenarios around Named Credentials to the Salesforce security team. In response, Salesforce extended the Event Monitoring product to include an event type that captures usage of Named Credentials, including their use by managed packages. Customers can leverage this new log source with the Winter ’22 release update (https://help.salesforce.com/s/articleView?id=release-notes.rn_security_em_nc_event_type.htm&type=5&release=234) to audit the use of named credentials and alert on unauthorized access by third parties.

SHARE

PLATFORM (HTTPS://APPOMNI.COM/?PAGE_ID=1181)

Why AppOmni(https://appomni.com/why-appomni/)

SOLUTIONS (HTTPS://APPOMNI.COM/SOLUTIONS/)

Box(https://appomni.com/solutions/box/)

Github(https://appomni.com/solutions/github/)

Microsoft Office 365(https://appomni.com/solutions/microsoft-office-365/)

Salesforce(https://appomni.com/solutions/salesforce/)

ServiceNow(https://appomni.com/solutions/servicenow/)

Slack(https://appomni.com/solutions/slack/)

Zoom(https://appomni.com/solutions/zoom/)

RESOURCES (HTTPS://APPOMNI.COM/RESOURCES/)

AO Labs(https://appomni.com/resources/aolabs/)

Articles & White Papers(https://appomni.com/whitepapers/)

Videos & Podcasts(https://appomni.com/media/)

News(https://appomni.com/news/)

Blog(https://appomni.com/blog/)

Press Releases(https://appomni.com/press-releases/)

SaaS Terms & Conditions (https://appomni.com/software-as-a-services-terms-and-conditions/)

COMPANY (HTTPS://APPOMNI.COM/ABOUT-US/)

Leadership(https://appomni.com/leadership/)

Partners(https://appomni.com/partners/)

Investors(/about-us#investors)

Careers(/careers)

(https://www.linkedin.com/company/appomni

(https://twitter.com/AppOmniSecurity)

Privacy Policy (/privacy-policy/) | Terms of Service (/terms-of-service/)

Got any questions? I'm happy to help.

/)