Welcome to the Application Security: Techniques and Best Practices module.

Agenda

- Types of Application Security Vulnerabilities
- Web Security Scanner
- Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application
- Threat: Identity and Oauth Phishing
- Identity-Aware Proxy
- Lab: Configuring Identity Aware Proxy to Protect a Project
- Secret Manager
- Lab: Configuring and Using Credentials with Secret Manager
- Quiz and Module Review

Google Cloud

In this module, we will discuss application security techniques and best practices

We will start with a discussion of a few common types of application security vulnerabilities.

And together we will see how Web Security Scanner can be used to identify vulnerabilities in your applications. You will get to perform a lab that uses Web Security Scanner to detect a vulnerability in an App Engine application.

Then the threat of Identity and Oauth phishing will be reviewed.

After that, you will see how the Identity-Aware Proxy or IAP can be used to control access to your cloud applications. You will also get to configure IAP in a lab.

Finally, we will discuss application security techniques and best practices.

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

Quiz and Module Review

Google Cloud

Let's begin with some types of application security vulnerabilities.

Applications are common targets

- Developer's main objective is often features and functionality:
  - Security is often neglected.
- Applications have become the most common target of attackers.

Google Cloud

Developers are frequently given a requirements document that defines the desired features and functional requirements for the application.

The code is written and tested many times against these features … but, the security of the application is often undefined and remains untested.

Attackers also know this and will attack applications more often than any other target - more often than networks, more often than infrastructure.

Common application vulnerabilities (1/2)

- Injection
  - SQL injection
  - LDAP injection
  - HTML injection
- Cross-site scripting (XSS)
- Weak authentication and access control

Google Cloud

To help you better secure *your* applications, let's discuss a few common application vulnerabilities. Please note that this is not meant to be a comprehensive list of all vulnerabilities - that would be impossible to create. That is because application vulnerabilities are quite variable, and are highly dependent upon how an application is constructed, and what resources they use. Therefore, in this module, we will mainly be discussing some of the more commonly found vulnerabilities.

Some of the most common application vulnerabilities are categorized as "Injection flaws." Injection flaws occur when some form of malicious content can be "injected" into an application from an attacker and the application will then accept and interpret that content. Injection flaws come in many flavors, including SQL injection, LDAP injection, and HTML injection.

Another common vulnerability is cross-site scripting, or XSS as it is also known. Cross-site scripting is actually a form of injection where the attacker is able to inject javascript into an application, and the code injection originates from a different site. Cross-site scripting allows attackers to execute scripts in the victim's browser which can then hijack user sessions, deface web sites, or redirect the user to other malicious sites.

Authentication, access control, and session management are "application logic" functions which are often implemented insecurely. When insufficient control over authentication and access is exercised, this vulnerability allows attackers to compromise passwords, keys, or session tokens, or to exploit other implementation

flaws to assume other users' identities.

Common application vulnerabilities (2/2)

- Sensitive data exposure
- Security misconfiguration
- Using components with known vulnerabilities

Google Cloud

Applications in general, including many web applications, also do not properly protect sensitive user data. Attackers may steal or modify weakly-protected data to facilitate credit card fraud, identity theft, or other data and identity crimes. Sensitive data risks being compromised any time it is transferred without extra protection. Secure data transfer requires encryption at rest or in transit, and special precautions when exchanged with the browser.

Security misconfiguration is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must also be patched/upgraded in a timely fashion in order to keep them secure.

Applications using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts. Components, such as libraries, frameworks, and other software modules generally run with the same privileges as the main application itself. As a result, when a vulnerable component is exploited, an attack may facilitate serious data loss or even a server takeover.

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

Quiz and Module Review

Google Cloud

Next, let's look at how Web Security Scanner can be used to identify vulnerabilities in your applications. After that, in a lab, you will use Web Security Scanner to detect a vulnerability in an App Engine application.

Web Security Scanner checks your applications for common vulnerabilities

- XSS
- Flash injection
- Mixed content
- Clear text passwords
- Use of insecure JavaScript libraries

Google Cloud

Web Security Scanner is a web security scanner which probes for common vulnerabilities in App Engine, Google Kubernetes Engine, and Compute Engine applications. It can automatically scan and detect four common vulnerabilities, including cross-site-scripting, Flash injection, mixed content (HTTP in HTTPS), and outdated/insecure libraries. You can easily set up, run, schedule, and manage your security scans using the Scanner and this is free for Google Cloud users.

How the scanner works

- Navigates every link it finds (except those excluded).
- Activates every control and input.
- Logs in with specified credentials.
- User-agent and maximum QPS can be configured.
- Scanner is optimized to avoid false positives.

Google Cloud

After you set up a scan, Web Security Scanner automatically “crawls” through your application, following all links within the scope of your starting URLs. Certain links can also be excluded if needed.

Once the scanner has been launched. It will attempt to exercise or activate as many user inputs, controls, and event handlers as possible.

If you need to test areas of an application that are only accessible after authentication, the scanner can be provided with application credentials that will allow it to authenticate to the system. Once authenticated, the scanner can then scan resources which are only available to authenticated users.

Additionally certain properties can be configured in the scanner, including the user agent type, and a maximum request rate (queries per second) to be used when performing a scan.

The scanner’s application logic has been optimized to help avoid false positives.

Scan scheduling

- Scans can be scheduled or manually initiated.
- Scans can be configured to run on a preset schedule.
- Scan duration scales with size and complexity of application; large apps can take hours to complete.

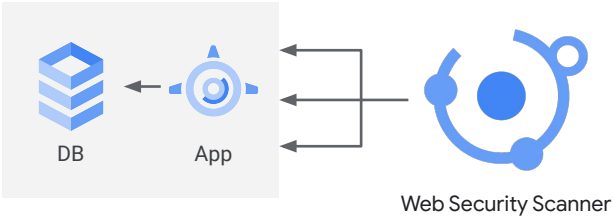
Google Cloud

Scans can run on a schedule, or can be manually initiated at any time. This allows the scanner to both perform scans on demand or have them scheduled to run in the future.

Scheduled scans can be configured to run regularly on an automated scheduled, for example, once per day, or once per month.

The duration of the scan can depend on the size and complexity of your application. Large, complex applications can take hours to be completely scanned, and as a result, such scans may need to be run less frequently.

Security scanner considerations



- The scanner generates real load against your application.
- The scanner can change state data in your application.

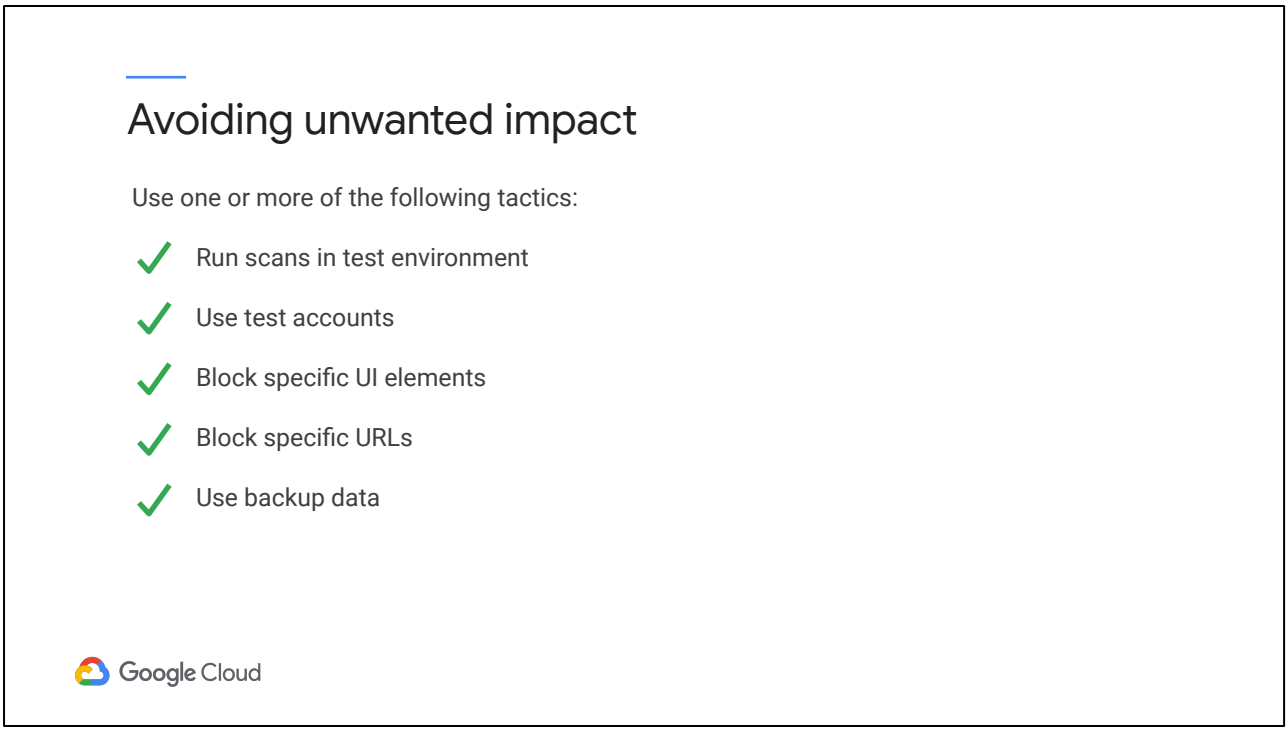
Google Cloud

When using Web Security Scanner, there are a few things to consider.

When performing scans, the scanner generates a real measurable load against your application as it populates fields, pushes buttons, clicks links, and so on.

The scanner can, and almost certainly will, change state data in your application, so it should always be used with caution. For example, when scanning a blog application that allows public comments, Web Security Scanner may post test strings as comments on all your blog articles.

In an email sign-up page, Web Security Scanner may generate large numbers of test emails.



Tactics to avoid unwanted impact may include:

Running scans in a test environment to ensure your production systems are not affected.

Using test accounts when providing the scanner with authentication information. Many applications present a special workflow during a user's first-time login, such as accepting terms, creating a profile, and so on. If your application has a different flow for first-time users, keep this in mind when choosing a user account for your scan. Because of the different workflow, a new user account completing the first time flow can yield different scan results than an established user account. It's best to scan with an account that is in the normal user state, after the first-time flow is complete.

Blocking specific UI elements that you do not want to be activated by applying the CSS class “inq-no-click”. Event handlers attached to this element are not activated during crawling and testing, regardless of whether they are inline JavaScript, or attached using addEventListener, or attached by setting the appropriate event handler property.

Blocking specific URLs in the scanner by specifying URL patterns that will not be crawled or tested.

And, finally, by making a backup of all data before scanning so the data can be restored to its original state prior to the scan execution if needed. Depending upon the

type and size of your application, there may be other steps you can take to minimize disruption and impact on your services when scanning.

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

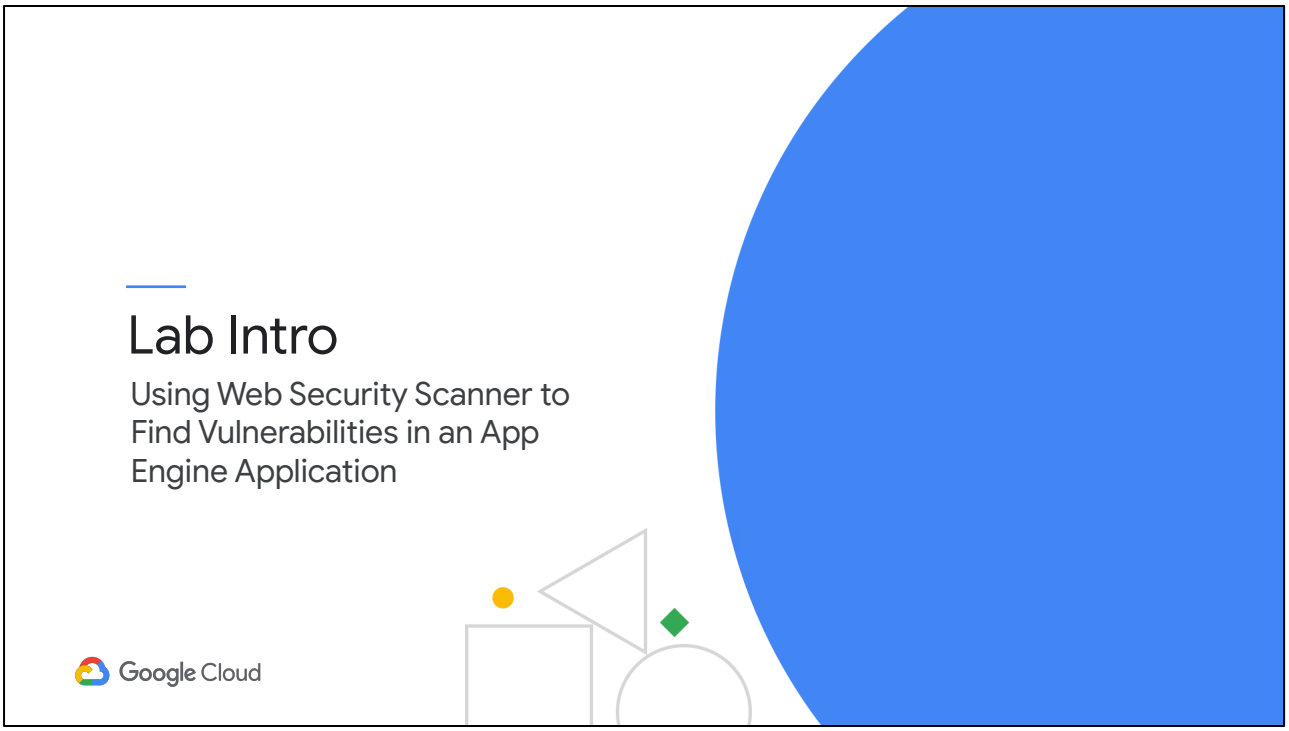
Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

Quiz and Module Review



In this lab, you will use Web Security Scanner to detect a vulnerability in an App Engine Application. The vulnerability will be fixed and then Web Security Scanner will be used to verify it has been corrected.

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

Quiz and Module Review

Google Cloud

Next, we will consider identity and Oauth phishing threats.

Phishing attacks

Phishing is the fraudulent attempt to obtain sensitive information, such as usernames, passwords, and credit card details.

Phishing attacks pose a constant threat to businesses.

Google Cloud

Phishing attacks, an attempt to obtain sensitive information from a user or site, pose a constant and significant threat to business. The information targeted can be anything of potential value to the hacker, especially usernames, passwords, and credit card numbers.

Sometimes information by itself does not appear to be sensitive, like your age or email address. But when combined with other information collected, these seemingly innocuous identity fragments could help an attacker impersonate you.

Identity phishing

Stealing someone's identity or credentials.

Phishing attacks pose a constant threat to businesses.

Google Cloud

Identity - especially online - is a complex set of fragments that can be gathered and combined to compromise a system or a user. The key to reducing risk of impersonation is to hide or secure as many of the pieces of the “identity puzzle” as possible. One way a phisher may attempt to gain identity information is by creating web sites that look almost exactly like trusted web sites, and then creating a scenario where a user is strongly enticed to click into that site and give up personal information. For example, a hacker may create a duplicate Facebook website on their own servers, designed to lure in the unwary, and then send an email or popup notice to a visitor that convinces them that they must log into “Facebook” for some reason. However, once the user has typed their email address and password into the site, that information is then given to the hacker, and they will have been phished.

The hacker could then login to their real Facebook account and take all the info available there. Depending on how you have your Facebook account set up, which games you may have played and which services you may have used, they may be able to gain enough additional information to compromise your identity on other sites. It can be a long process to steal enough identity fragments to compromise someone’s identity, but it can be worth it to the hacker.

OAuth phishing

Phishing technique that takes advantage of the Open Authentication (OAuth) standard to gain backend access to user accounts.

Google Cloud

OAuth phishing is a type of credential phishing that takes advantage of the Open Authentication (OAuth) standard to gain access to user accounts without needing to know the user's password. OAuth phishing exploits the trust relationship users have with well-known online service providers, as well as the trust relationship those providers have with their own third-party applications. It is accomplished by tricking users into granting persistent access to their accounts.

OAuth phishing attacks do not use the typical approach used in email phishing (spoofed URL link, sign-in request, or attached file). That makes this type of phishing more difficult for a less experienced user to detect and may produce a higher rate of success for the hacker. In some cases, the deception is very well designed and may even affect more experienced and competent users.

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

Quiz and Module Review

Google Cloud

Lastly you will see how the Identity-Aware Proxy, or IAP, can be used to control access to your cloud applications. You will also get to configure IAP in a lab.

Identity-Aware Proxy (IAP)

Google Cloud

Identity-Aware Proxy (IAP) provides a central authentication and authorization layer for your applications over HTTPS.

IAP replaces end-user VPN tunnels or the need to apply an authentication and authorization layer in front of a web application hosted on Google Cloud.

At Google, we use IAP to control access to internal applications without the need to use end-user VPNs. A Google employee can simply access needed applications from anywhere.

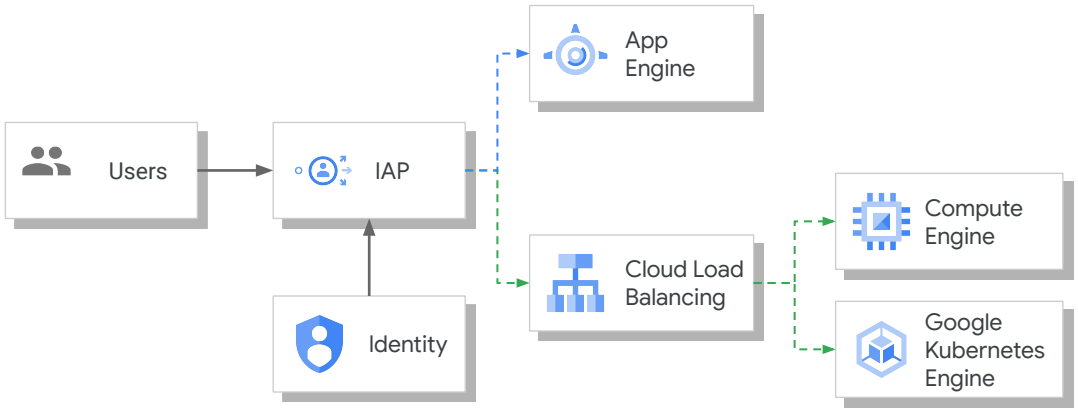
Identity-Aware Proxy (IAP)

- Controls access to your cloud applications running on Google Cloud.
- Verifies a user's identity.
- Determines whether that user should be allowed to access the application.

Google Cloud

Whenever you attempt to log into an application, your request is forwarded to IAP, which requires the user to log in. Once logged in, the proxy will determine if the user is allowed to access that application. If authorization is obtained for the user, they are then forwarded to the requested application page.

Identity-Aware Proxy (IAP)



Google Cloud

IAP lets you manage access to App Engine instances, Compute Engine instances, and Google Kubernetes Engine clusters with a central authentication and authorization layer for your applications over HTTPS.

IAP provides a simpler administration process

- Deploys in minutes
  - No VPN to implement/maintain
  - No VPN clients to install
- Saves end user time
- Faster to sign in to than a VPN

Google Cloud

IAP provides a much simpler administration process and with reduced operational overhead than more traditional VPN solutions. There is no VPN to implement or VPN clients to install and maintain. It also makes the end user experience more streamlined, as the user no longer must launch the VPN client and sign into the VPN.

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

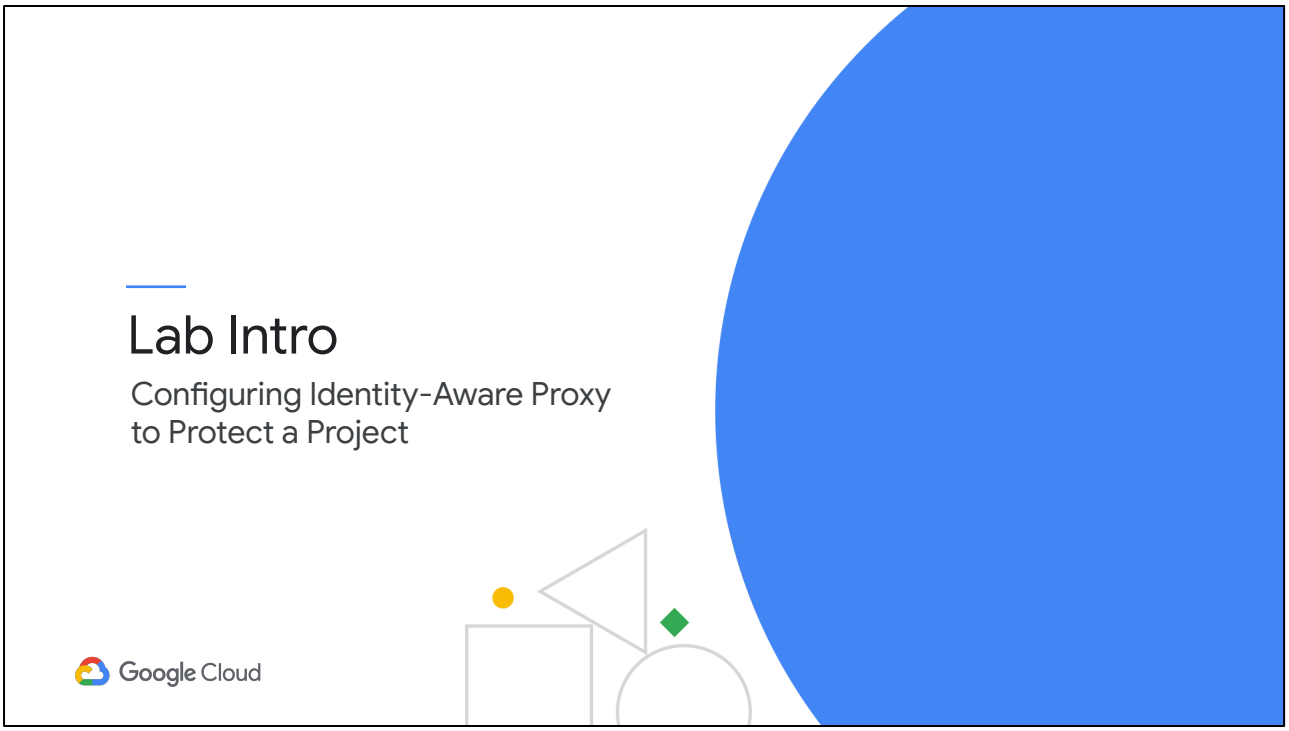
Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

Quiz and Module Review



In this lab, you learn how to perform the following tasks:

- Configure the OAuth consent screen
- Setup IAP access, and
- Enable IAP

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

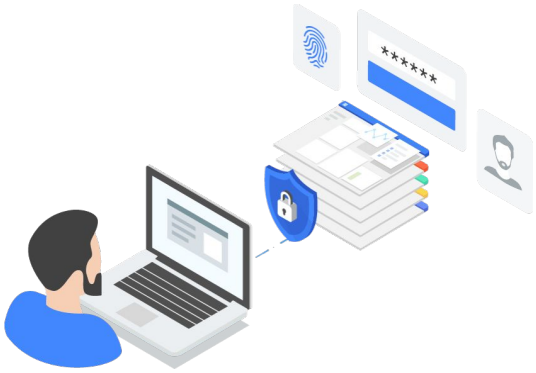
Quiz and Module Review

Google Cloud

In this module, we will discuss Secret Manager and how it is used to store and access sensitive data.

Storing credentials securely

- Many applications require credentials to authenticate; for example, API keys, passwords, or certificates.
- Storing this information in a flat text file makes access easy but requires file protection.
- Secret Manager provides a secure, convenient way to store sensitive information.



Google Cloud

Many applications require credentials to authenticate; for example, API keys, passwords, or certificates. For applications to start or open automatically, this information must be readily available and stored somewhere in your environment.

Storing this information in a flat text file makes the information easy to access, but it requires file protection. You do not want everyone to see this information. Of course, one way to protect files is to use the file operating system to apply permissions to restrict access as needed. This can lead to “secret sprawl” - with secrets scattered across the an organization’s cloud and on-premises infrastructure.

Secret Manager provides a secure, convenient way to store sensitive information. You can store, manage, and access secrets as binary blobs or text strings. Only users with the appropriate permissions can view the contents of the secret. Secret Manager reduces secret sprawl within a Google Cloud deployment. When secrets are secured in a single place, secret sprawl is eliminated. You can use IAM to determine who has access to the secrets, and also the kind of access they have.

Secret Manager features

- Global names and replication
- Versioning
- Follows principles of least privilege
- Audit logging
- Strong encryption

Google Cloud

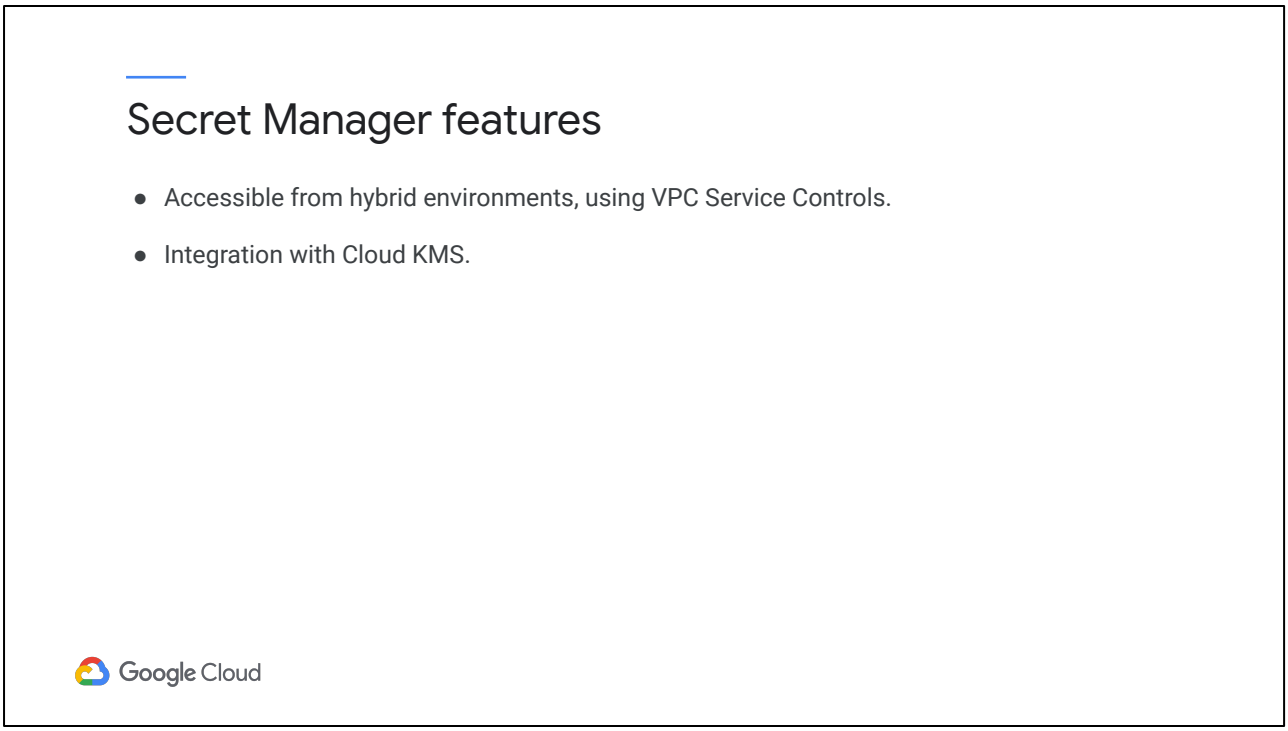
Global names and replication. The secret name is global but, optionally, its data can be stored regionally. You will learn more about this later in the module.

Versioning. Secrets can be versioned. Each version can have different secret data to protect. There is no limit on the number of versions that can be stored.

Principles of least privilege: Secrets are created at the project level. Only project owners have permissions to create and access secrets within their project. Other roles must explicitly be granted permissions through IAM.

Audit logging: With Cloud Audit Logging enabled, every interaction with Secret Manager generates an audit entry. You can ingest these logs into anomaly detection systems to spot abnormal access patterns and alert on possible security breaches.

Strong encryption: Secret Manager manages server-side encryption keys on your behalf using the same hardened key management systems used for Google's own encrypted data, including strict key access controls and auditing. Secret Manager encrypts user data at rest using AES-256. There is no setup or configuration required, no need to modify the way you access the service, and no visible performance impact. Your secret data is automatically and transparently decrypted when accessed by an authorized user.The Secret Manager API always communicates over a secure HTTP(s) connection.



Accessible from hybrid environments, using VPC Service Controls. You can configure VPC Service Controls to allow only specified hosts to have access to the Secret Manager APIs. Other hosts will not be able to access your secrets. This applies to hosts both inside and outside of Google Cloud.

Integration with Cloud KMS: Secret Manager integrates with Cloud KMS. If desired, Cloud KMS can encrypt the version for a secret before storing it. After retrieving the version, Cloud KMS would also then be needed to decrypt it. Remember that the version holds the sensitive value that you are trying to protect, such as the password.

Access control using IAM

- By default, only project owners can create and access secrets within their project.
- Use IAM to grant roles and permissions at the level of the Google Cloud organization, folder, project, or secret.

Google Cloud

By default, only project owners can create and access secrets within their project.

Use IAM to grant roles and permissions at the level of the Google Cloud organization, folder, project, or secret. On the next slide, we will look at roles that control access to secrets.

Working with IAM roles

- IAM roles:
  - secretmanager.admin: Can view, edit, and access a secret.
  - secretmanager.secretAccessor: Can only access secret data.
  - secretmanager.viewer: Can view a secret's metadata and its versions, but can't edit or access secret data.
- Always apply permissions at the lowest level in the resource hierarchy.

Google Cloud

The Secret Manager Admin role can view, edit, and access a secret. Effectively, this role provides full access to administer Secret Manager resources.

The Secret Manager Secret Accessor role can only access secret data. This is useful for service accounts or for situations when all that is needed is the ability to access the sensitive data held inside the secret.

The Secret Manager Viewer role can view the secret's metadata and its versions but cannot edit or access secret data.

As always, apply permissions at the lowest level of the resource hierarchy to obtain the result you desire. For example, if you want a user to have access to five secrets within the project, grant that user permission only to those five secrets. For information about specific permissions within each role, refer to the Secret Manager documentation or use the Google Cloud Console.

Using the Cloud Console

- Secret Manager can be configured and used in the Cloud Console.
- Before you can use Secret Manager, you must first enable the Secret Manager API.
- Only users with the appropriate role/permissions can use Secret Manager.

Google Cloud

Secret Manager can be configured and used in the Cloud Console. The Security menu includes a Secret Manager option, where you can create, edit, and view secrets.

Note that before you can use Secret Manager, you must first enable the Secret Manager API. That is because the Cloud Console accesses Secret Manager features through the API. Failure to do this results in an error when you try to access the Secret Manager page.

Only users with the appropriate role/permissions can use Secret Manager. You learned about these roles in the previous slides.

Replicating secret data

- Secret names are global.
- Secret data is regional.
- If you have no preference where secret data is stored, use automatic replication.
- To pick the regions in which the secret data is stored, use manual replication.

Google Cloud

A secret is a project-level resource, and its name is global. Users with sufficient IAM permissions can access the secret's data and/or metadata. You can choose between *automatic* and user-managed replication policies, so you control where your secret data is stored.

Although secret names are global, secret data is regional. In other words, the data for a secret can be stored in one or more regions. Some enterprises want full control over the regions in which their secrets are stored, while others do not have a preference. The default behavior is no regional preference.

If you have no preference regarding the region where secret data is stored, use automatic replication. This option is selected by default.

To pick the regions in which the secret is stored, at secret creation time, select manual replication, and then select the regions that you want the data to be stored in. For a list of regions where secret data can be stored, see the Secret Manager documentation.

Secret Manager API

- The Secret Manager API enables access to Secret Manager features within a program.
- It is useful for automating access to secrets.
- Before Secret Manager features can be used, the Secret Manager API must first be enabled.

Google Cloud

The Secret Manager API enables access to Secret Manager features within a program. Thus, any language that supports sending HTTP requests can access the API and work with secrets and their data.

The Secret Manager API is useful for automating access to secrets. For example, suppose when you reboot a Compute Engine instance, a startup script attempts to access a database. You would like to automatically log in to the database. Logging in requires a username and a password to authenticate. If the username and password are stored as secrets, you can add logic to the startup script that uses the Secret Manager API to obtain the version data for the username and password secrets.

Before Secret Manager features can be used, the Secret Manager API must first be enabled. This is true even if you are accessing these features from within the Google Cloud Console and never plan to call the Service Manager API from programs or scripts. That is because on the backend, Google Cloud Console uses the API.

Versioning

- Secrets can be versioned.
- Each version is assigned a version ID.
- The most recent version is automatically assigned the label latest.
- Secrets cannot be modified, but they can be disabled or deleted.
- Individual versions can be selected and disabled or destroyed.

Google Cloud

Secrets can be versioned. There is no limit to the number of versions that can be created for a given secret.

Each version is automatically assigned an ordinal version ID: 1, 2, 3, 4, etc. You can refer to individual versions of a secret to view the secret value or the secret's metadata.

In addition to the version ID, the most recent version is automatically assigned a version label called latest.

You cannot modify a version, but you can disable it or delete it. Instead of modifying a new version, add a version with the correct data. There is no limit on the number of versions that can be kept.

Individual versions can be selected and then disabled or destroyed. This prevents users from accessing data for a secret that is incorrect. It is possible to disable all the versions of a secret or to delete the entire secret.

Secret rotation

- To rotate a secret, add a new version to it.
- Any enabled version can be accessed.
- Secret Manager does not have an automatic rotation feature.
  - Use Cloud Functions and the Secret Manager API to perform automatic rotation.

Google Cloud

To rotate a secret, add a new version to it. All versions are numbered, starting from 1. Users can refer to the latest version using the label “latest.

Any version can be accessed, as long as it is enabled. To prevent a version from being used, disable that version.

Secret Manager does not have an automatic rotation feature. However, you can use the Secret Manager API with Cloud Functions to perform automatic rotation. For more information, see the Secret Manager API and the Cloud Functions documentation.

Admin Activity Access audit logs

- Admin Activity Access audit logs cannot be disabled.
- All activity that creates or changes secret data is logged here.
- Setting or getting IAM policy information about secrets is logged.

Admin Activity Access audit logs cannot be disabled. These logs do not count toward your log ingestion quota.

All activity that creates or changes secret data is logged here.

Setting or getting IAM policy information about secrets is logged. Only the version access is not logged here. You will learn more about that on the next slide. Note that Secret Manager doesn't write System Event audit logs.

Data Access audit logs

- Data Access audit logs are disabled by default.
- If logs are enabled, all access to the secret data is logged.

Google Cloud

Data Access audit logs are disabled by default. Data Access logs do count toward your log ingestion quota, and, therefore, there is a cost involved.

When enabled, Data Access audit logs contain API calls that read the configuration or metadata of resources, as well as user-driven API calls that create, modify, or read user-provided resource data. Data Access audit logs do not record the data-access operations on resources that are publicly shared. In other words, nothing is recorded for anything that is available to All Users or All Authenticated Users or that can be accessed without logging into Google Cloud.

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

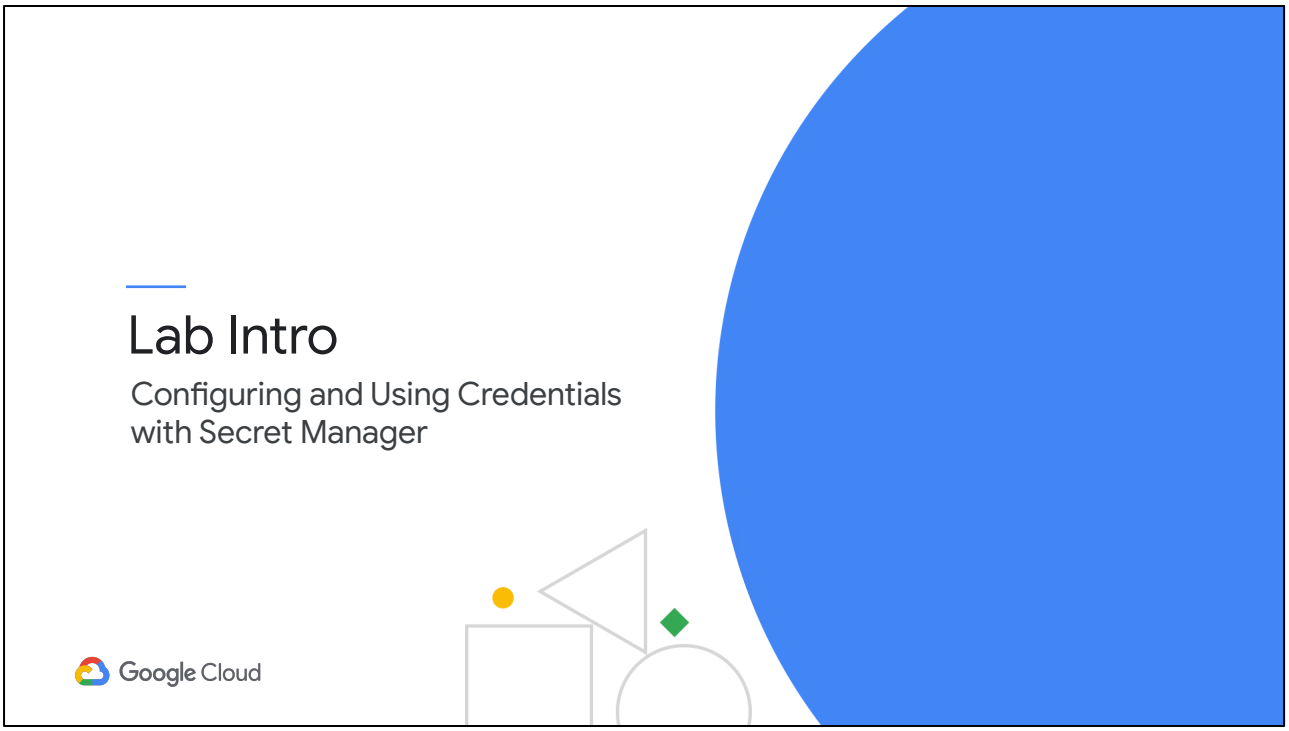
Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

Quiz and Module Review



In this lab, you will learn how to add application credentials to Secret Manager and then access those credentials using the Secret Manager API.

Agenda

Types of Application Security Vulnerabilities

Web Security Scanner

Lab: Using Web Security Scanner to Find Vulnerabilities in an App Engine Application

Threat: Identity and Oauth Phishing

Identity-Aware Proxy

Lab: Configuring Identity Aware Proxy to Protect a Project

Secret Manager

Lab: Configuring and Using Credentials with Secret Manager

Quiz and Module Review

Quiz #1

Question

Which TWO of the following statements about Application Security are TRUE?

A. Developers are commonly given a requirements document that clearly defines security requirements for the application.

B. Applications in general, including many web applications, do not properly protect sensitive user data.

C. "Injection Flaws" are the least frequently found application security issue.

D. Applications are the most common target of cyber attack.

Quiz #1

Answer

Which TWO of the following statements about Application Security are TRUE?

A. Developers are commonly given a requirements document that clearly defines security requirements for the application.

B. Applications in general, including many web applications, do not properly protect sensitive user data.

C. "Injection Flaws" are the least frequently found application security issue.

D. Applications are the most common target of cyber attack.

Google Cloud

B. Attackers will target web applications in order to steal or modify weakly-protected data to facilitate credit card fraud, identity theft, or other data and identity crimes.
D. Attackers know that the security of an application is often undefined and remains untested.

Quiz #2

Question

Which TWO of the following are common application security vulnerabilities?

A. Insecure logins.
B. Mixed content.
C. Outdated or insecure libraries.
D. Personalized data in object names.
E. User data in images.

Quiz #2

Answer

Which TWO of the following are common application security vulnerabilities?

A. Insecure logins.
B. Mixed content.
C. Outdated or insecure libraries.
D. Personalized data in object names.
E. User data in images.

Google Cloud

B. Web Security Scanner will find instances in your applications where HTTP and HTTPS calls are mixed.
C. Web Security Scanner will check your code libraries to make sure they are current and secure.

Quiz #3

Question

Which TWO of the following statements are TRUE when discussing the threat of OAuth and Identity Phishing?

A. Being "hacked" on a social site can lead to being "hacked" on more critical websites, depending on your social site's account settings.

B. Credit card data is the only information that is useful to cyber hackers.

C. Even small, unimportant pieces of personal data need to be secured from phishing attacks.

D. Look-alike phishing sites are generally pretty easy to spot.

Quiz #3

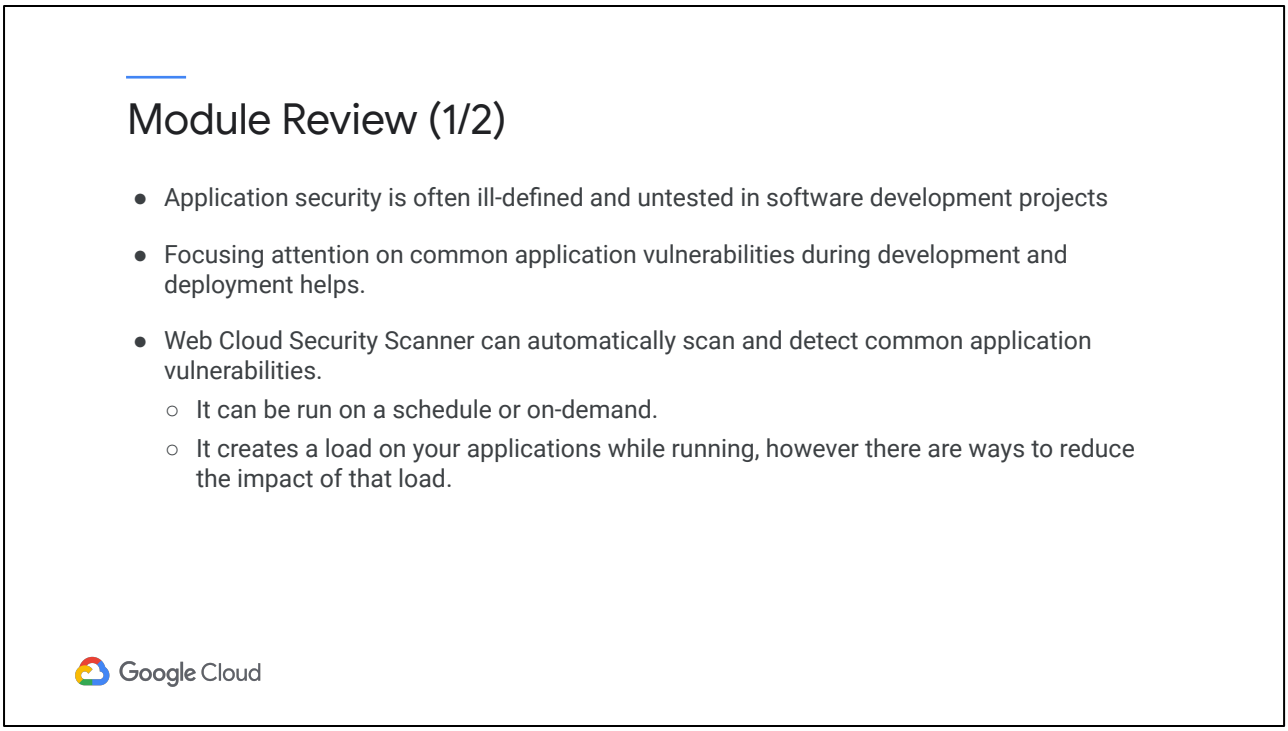
Answer

Which TWO of the following statements are TRUE when discussing the threat of OAuth and Identity Phishing?

A.	Being "hacked" on a social site can lead to being "hacked" on more critical websites, depending on your social site's account settings.
B.	Credit card data is the only information that is useful to cyber hackers.
C.	Even small, unimportant pieces of personal data need to be secured from phishing attacks.
D.	Look-alike phishing sites are generally pretty easy to spot.

Google Cloud

A. Some social sites allow users to "link" their accounts on other sites - for example, gaming sites - and that link can often be exploited.
C. While a small piece of personal data may not be enough on its own to allow a phisher to obtain access to a user's more sensitive data, over time these small bits can often be combined to allow user accounts to be compromised.



In this module, we discussed application security, and how to mitigate common application vulnerabilities. Let's do a brief review.

Application security is often ill-defined and untested in software development projects. Hackers know this, so they target applications more often than they target infrastructure or networks!

A complete list is impossible to create, but there are known common application vulnerabilities that can be mitigated. Common application vulnerabilities include injection vulnerabilities, where malicious code is "injected" into the system and the data underlying it. These generally take the form of SQL injection, LDAP injection, and HTML injection. Another common vulnerability is cross-site scripting, or XSS, which is actually a form of injection where the attacker is able to inject javascript from a different site into an application, allowing attackers to execute malicious scripts in the victim's browser.

Web Security Scanner is a web security scanner which probes for common vulnerabilities in Google App Engine and compute engine applications. It can automatically scan and detect common vulnerabilities, including cross-site-scripting, Flash injection, mixed content (HTTP in HTTPS), and outdated/insecure libraries. Web Security Scanner can be run manually, or on a schedule. Web Security Scanner

can impact the performance of your applications while it is scanning, but there are steps you can take to minimize this impact.

Module Review (2/2)

- Identity and OAuth phishing is another common application vulnerability.
- Identity-Aware Proxy replaces your much less secure VPN tunnels.
  - This is the method Google uses to secure its applications.
  - It works with Compute Engine and Google Kubernetes Engine instances.
- Secret Manager makes sensitive data needed by your applications secure and easy to access.

Google Cloud

Another common threat to application security is identity and OAuth “phishing.” Phishing attacks are an attempt to obtain sensitive information from a user or site, especially usernames, passwords, and credit card numbers. OAuth phishing is a type of credential phishing that takes advantage of the Open Authentication (OAuth) standard to gain access to user accounts without needing to know the user’s password by exploiting the trust relationship users have with well-known online service providers and third-party applications.

IAP replaces end-user VPN tunnels or the need to apply an authentication and authorization layer in front of a web application hosted on Google Cloud. Your request is forwarded to IAP, which requires the user to log in, and the proxy will determine if the user is allowed to access that application. IAP works with Compute Engine instances and Google Kubernetes Engine clusters with a central authentication and authorization layer over HTTPS.

Secret Manager makes sensitive data needed by your applications secure and easy to access. It also simplifies the management of sensitive data by putting it in a single place, controlled by IAM.

I hope this module has given you some helpful information and tools you can use to secure your applications.


Google Cloud