



## Application Security: Techniques and Best Practices



Welcome to module 2 of *Security Best Practices in Google Cloud*—Application Security: Techniques and Best Practices.

## Module overview

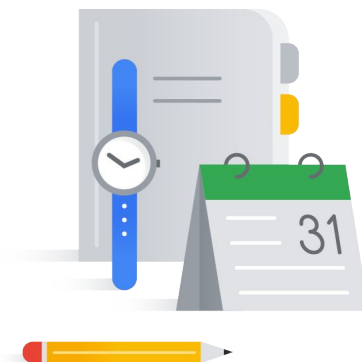
Types of application security vulnerabilities

Web Security Scanner

Threat: Identity and Oauth phishing

BeyondCorp Enterprise and Identity-Aware Proxy

Secret Manager



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 secure Compute Engine applications with IAP in a lab.

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

# Application Security

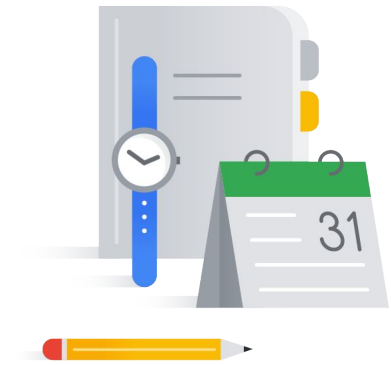
## Types of application security vulnerabilities

Web Security Scanner

Threat: Identity and Oauth phishing

BeyondCorp Enterprise and  
Identity-Aware Proxy

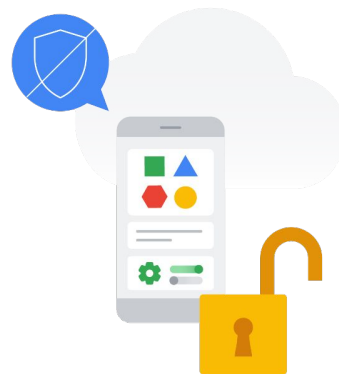
Secret Manager



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
    - Often undefined and remains untested
    - Deadline pressures promote release of vulnerable code
- Applications have become the most common target of attackers.
- Security not usually part of application design process



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.

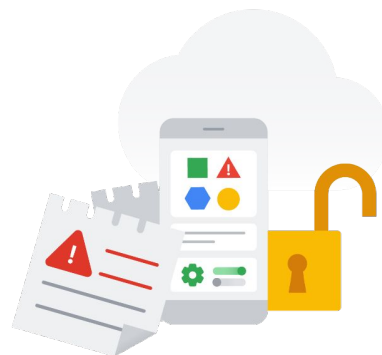
Deadline pressures also promote the release of vulnerable code.

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

Often times security is not part of the design process in application development—in many cases it is an afterthought. This means instead of being proactive about security, many times application security becomes a reactive process.

## Common application vulnerabilities

- Injection
  - SQL injection
  - LDAP injection
  - HTML injection
- Cross-site scripting (XSS)
- Weak authentication and access control
- Sensitive data exposure
- Security misconfiguration
- Using components with known vulnerabilities



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.

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.

# Application Security

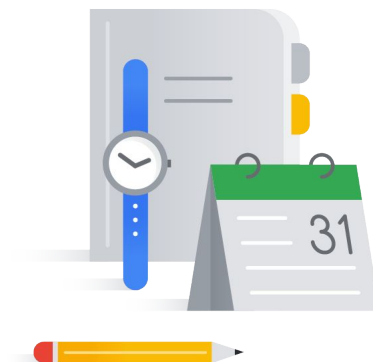
Types of application security vulnerabilities

## Web Security Scanner

Threat: Identity and OAuth phishing

BeyondCorp Enterprise and Identity-Aware Proxy

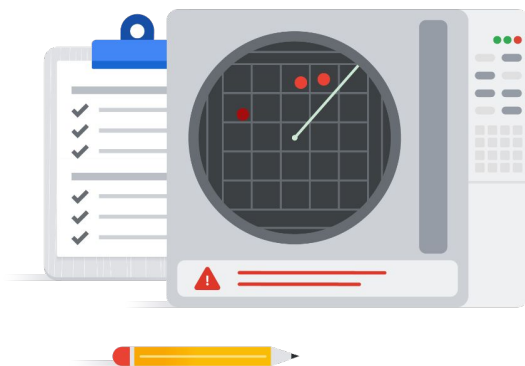
Secret Manager



Next, let's look at how Web Security Scanner can be used to identify vulnerabilities in your applications. Afterwards, you will have hands-on practice with 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
- Supports categories in OWASP Top Ten



Google Cloud

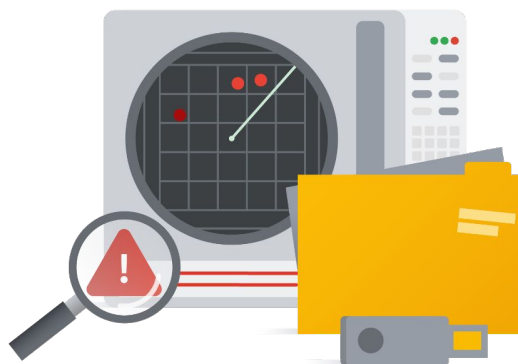
Web Security Scanner is a web security scanner which probes for common vulnerabilities in App Engine, Google Kubernetes Engine, and Compute Engine applications. It can automatically scan and detect four common vulnerabilities, including cross-site-scripting, Flash injection, mixed content (HTTP in HTTPS), 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.

Web Security Scanner supports categories in the [OWASP Top Ten](#), a document that ranks and provides remediation guidance for the top 10 most critical web application security risks, as determined by the Open Web Application Security Project (OWASP). A link to the OWASP Top Ten document is contained in this module's course resources.



## 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.
- Vulnerabilities reported in:
  - Web Security Scanner "Results" tab
  - Security Command Center
  - Cloud Logging



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.

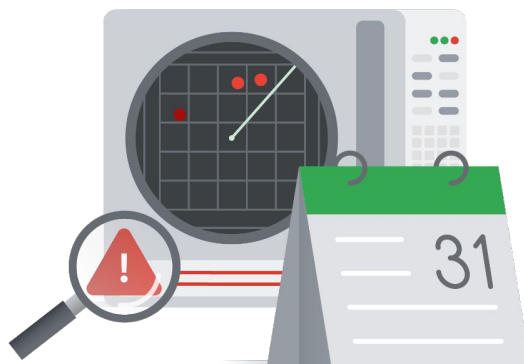
The scanner’s application logic has been optimized to help avoid false positives.

You can find the vulnerabilities reported in a few different places:

- In the Web Security Scanner "Results" tab.
- In Security Command Center.
- In Cloud Logging

## 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.

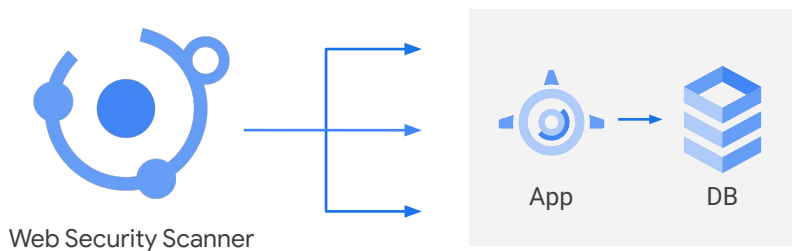


Scans can run on a schedule, or can be manually initiated at any time. This allows the scanner to both perform scans on demand or have them scheduled to run in the future.

Scheduled scans can be configured to run regularly on an automated schedule, 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.

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.

## Avoiding unwanted impact

Use one or more of the following tactics:

- ✓ Run scans in test environment
- ✓ Use test accounts
- ✓ Block specific UI elements
- ✓ Block specific URLs
- ✓ Use backup data

Google Cloud

Tactics to avoid unwanted impact may include:

**Running scans in a test environment** to ensure your production systems are not affected. Ensure that your test environment is as close as similar as possible to your production environment so you don't receive any false positives or negatives.

**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 a scan 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.

# Lab Intro

Identify Application Vulnerabilities  
with Security Command Center



In this lab you'll use Web Security Scanner, a built-in service of Security Command Center, to detect a vulnerability in a Python Flask application. The vulnerability will be fixed and then Web Security Scanner will be used to verify it has been corrected.

# Application Security

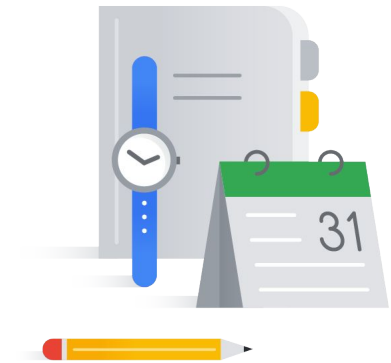
Types of application security vulnerabilities

Web Security Scanner

Threat: Identity and Oauth phishing

BeyondCorp Enterprise and Identity-Aware Proxy

Secret Manager

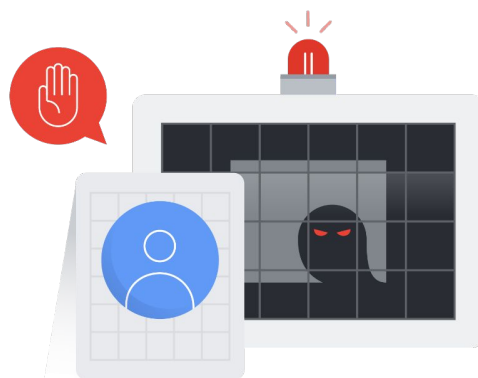


Next, we will consider identity and Oauth phishing threats.

## 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.



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.

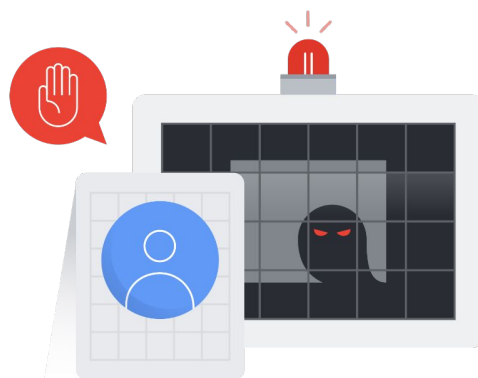
Sometimes information by itself does not appear to be sensitive, like your age or email address. But when combined with other information collected, these seemingly innocuous identity fragments could help an attacker impersonate you.



## 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.

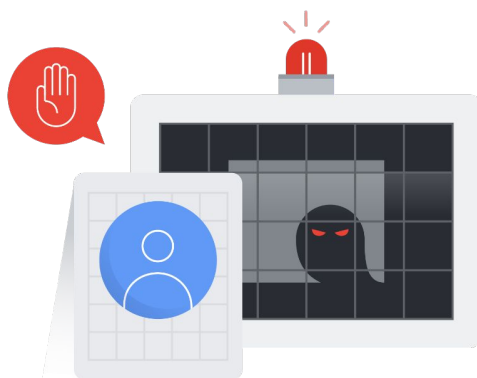
However, once the user has typed their email address and password into the site, that information is then given to the hacker, and they will have been phished.

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.



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.

# Application Security

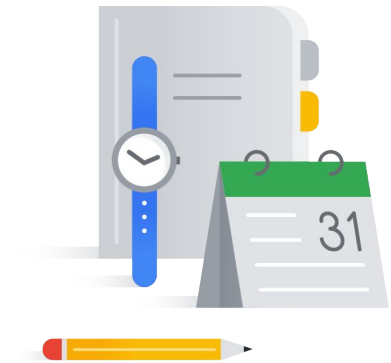
Types of application security vulnerabilities

Web Security Scanner

Threat: Identity and Oauth phishing

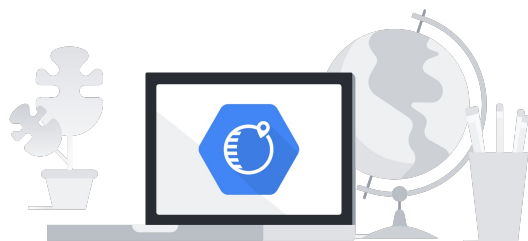
[BeyondCorp Enterprise and Identity-Aware Proxy](#)

Secret Manager



You will now see how the Identity-Aware Proxy, or IAP, can be used to control access to your cloud applications. You will also get to secure Compute Engine applications with IAP in a lab.

## BeyondCorp (zero trust) security model



- Access is granted based on a **user's identity and context** of the request.
- Shifts controls from the **network** layer to the **application** layer.
- Allows users to work from **any location & device**.

In 2011, Google came up with a new approach for access management - the BeyondCorp enterprise security model. It started as an internal Google initiative to enable every employee to work from untrusted networks without the use of a traditional VPN.

BeyondCorp shifts access controls from the network perimeter to individual users and devices, thereby allowing employees to work more securely from any location. This transforms work into a truly edgeless world.

## BeyondCorp Enterprise uses

BeyondCorp Enterprise lets you:

- Protect sensitive data
- Ensure key systems only accessible from certain devices
- Provide DLP for corporate data
- Gate access based on user location
- Protect apps in hybrid deployments



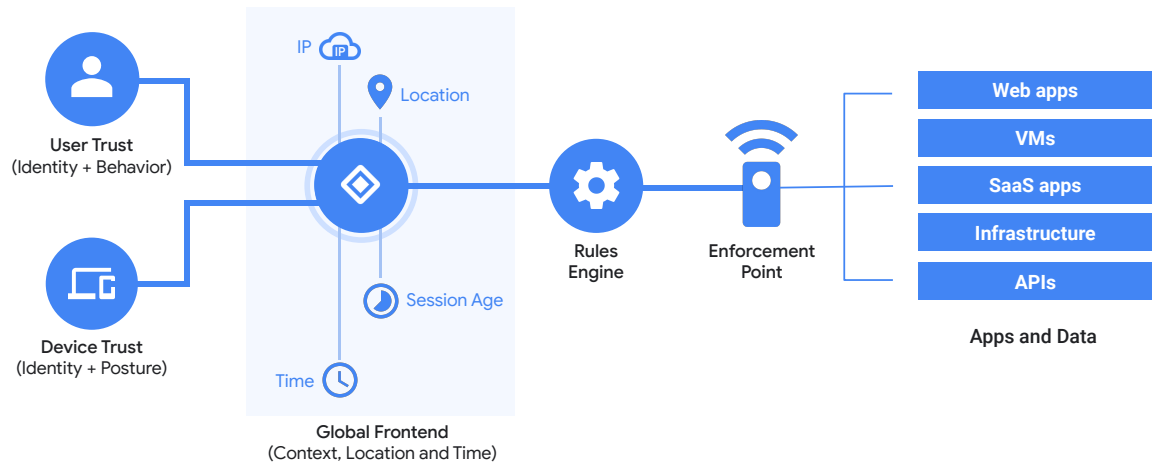
Google Cloud

As end users work outside of the office more often and from many different types of devices, enterprises have common security models they are looking to extend to all users, devices, and applications.

Here are some common use cases:

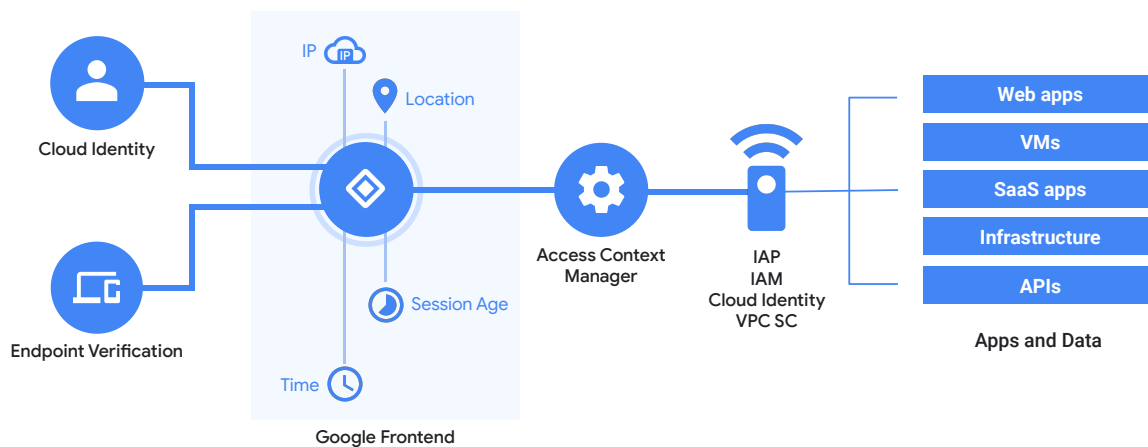
- Ensure employees are prevented from copying and pasting sensitive data into email or saving data into personal storage such as Google Drive.
- Only allow enterprise-managed devices to access certain key systems.
- Provide DLP protections for corporate data.
- Gate access based on a user's location.
- Protect applications in hybrid deployments that use a mix of Google Cloud, other cloud services platforms, or on-premises resources.

## BeyondCorp solution components



The following demonstrates the baseline solution components for BeyondCorp Enterprise.

## Enabling BeyondCorp with Enterprise Beyondcorp



As you can see, each solution component can be fulfilled by a specific Google Cloud service.

## Enforcement points

There are three key enforcement points

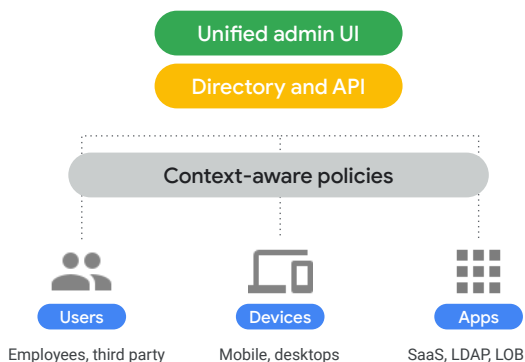
- ✓ Cloud Identity
- ✓ VPC Service Controls
- ✓ IAP

There are three key enforcement points that we will discuss now.



## Cloud Identity - for Google Workspace apps

- Extend context-aware for Google Workspace tools
- Enforce access controls for Gmail, Docs, Sheets
- 3rd party solutions via SAML



[cloud.google.com/identity/](https://cloud.google.com/identity/)

The first enforcement point we have already covered at length in the **Cloud Identity** lesson of the **Securing access to Google Cloud** module.

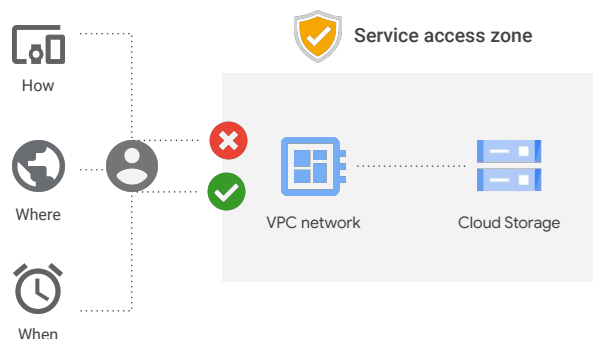
This is used to extend context-aware for Google Workspace tools, enforce access controls for Gmail, Docs, Sheets, and 3rd party solutions via SAML.

## VPC Service Controls - for Google Cloud APIs

VPC Service Controls allow users to define a security perimeter around Google Cloud resources.

With VPC Service Controls, you can:

- Enforce context-aware access.
- Mitigate data exfiltration risks.
- Extend security boundaries across cloud environments.
- Centrally manage security policies.



[cloud.google.com/vpc-service-controls](https://cloud.google.com/vpc-service-controls)

Google Cloud

We also talked about VPC Service Controls, which have a number of important security features that allow you to mitigate data exfiltration risks, configure private communication between cloud resources and hybrid VPC networks, and enforce context-aware access controls (e.g., allow access based on user location or source IP address).

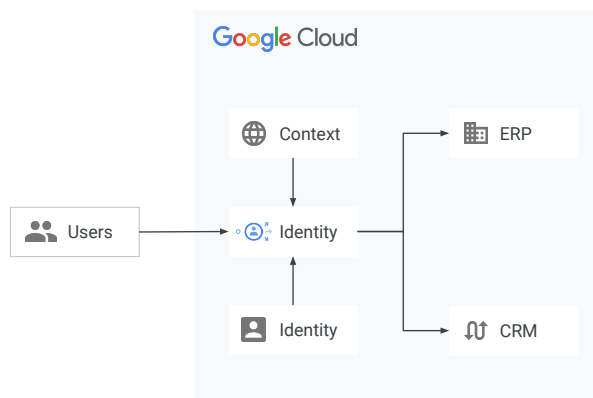
You can also manage these virtual security perimeters centrally, so it's easy to add or take away projects and services from a service access zone that you set up.

## IAP - for web apps / VMs

A reverse proxy sits in front of every data request.

- Provides a central authentication and authorization layer for applications over HTTPS.
- Replaces end-user VPN tunnels.
- Replaces authentication and authorization layers needed for web applications hosted on Google Cloud.

The proxy needs to have near zero latency to provide “invisible” security.



[cloud.google.com/iap/](https://cloud.google.com/iap/)

Google Cloud

IAP is an enterprise security model that enables every employee to work from untrusted networks without the use of a VPN.

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.

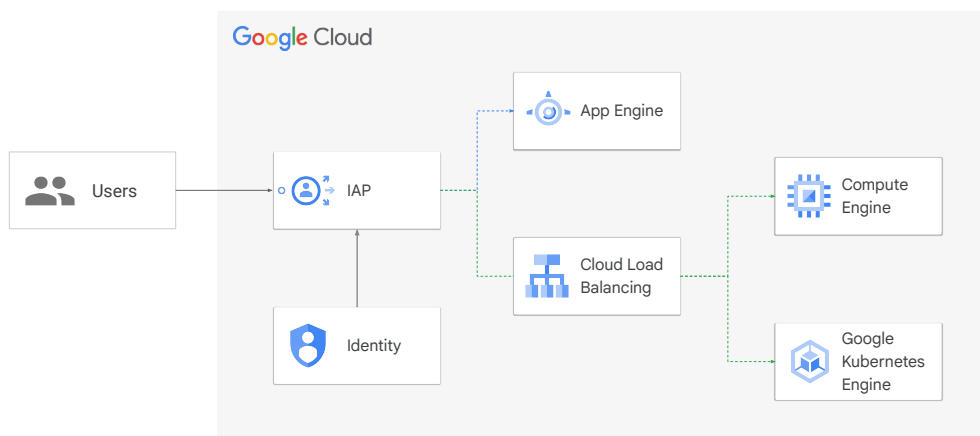
## Identity-Aware Proxy (IAP)

- 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.



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)



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 allows provides context-aware access for SSH and RDP, which allows you to control access to VMs based on a user's identity and context (e.g. device security status, location, etc).

IAP also supports web-apps that even run outside Google Cloud.

## 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.

# Lab Intro

Securing Compute Engine  
Applications with BeyondCorp  
Enterprise



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

- Configure OAuth Consent.
- Set up OAuth access credentials.
- Set up IAP access for the deployed application.
- And use IAP to restrict access to the application.

# Application Security

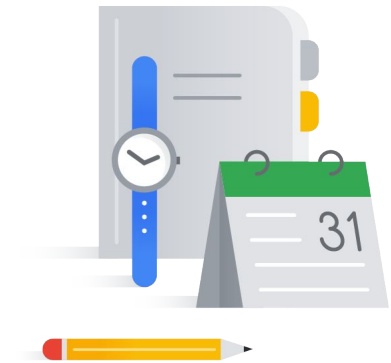
Types of application security vulnerabilities

Web Security Scanner

Threat: Identity and Oauth phishing

BeyondCorp Enterprise and Identity-Aware Proxy

[Secret Manager](#)



In this lesson, 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.



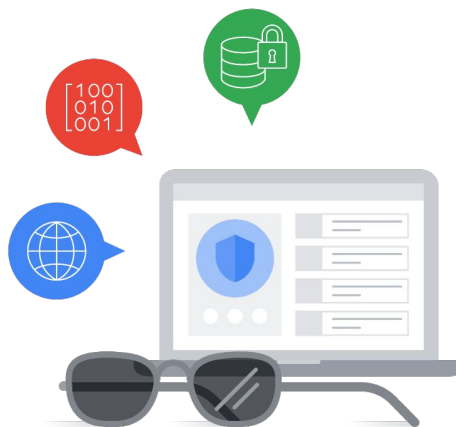
Many applications require credentials to authenticate; for example, API keys, passwords, or certificates. For applications to start or open automatically, this information must be readily available and stored somewhere in your environment.

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 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.

## 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.

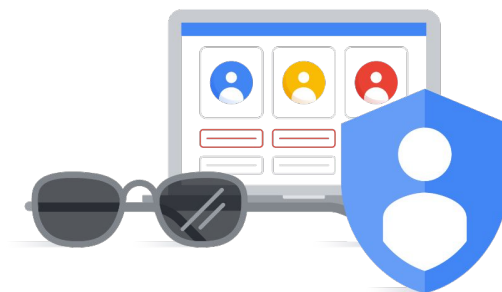


By default, only project owners can create and access secrets within their project.

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 Assessor 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.

## 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.



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.

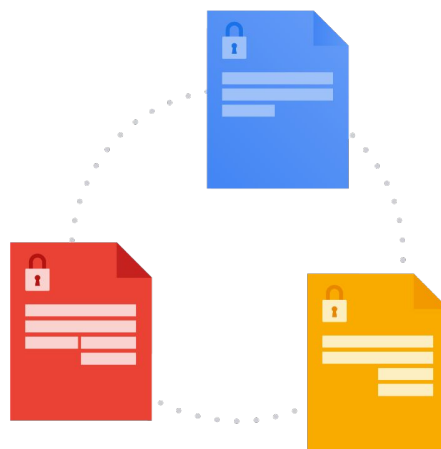
In addition to the version ID, the most recent version is automatically assigned a version label called `latest`.

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 supports rotation schedules on secrets.
  - Sends messages to Pub/Sub topics configured on the secret based on the provided rotation frequency and rotation time.



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 supports rotation schedules on secrets. Secret Manager sends messages to Pub/Sub topics configured on the secret based on the provided rotation frequency and rotation time.

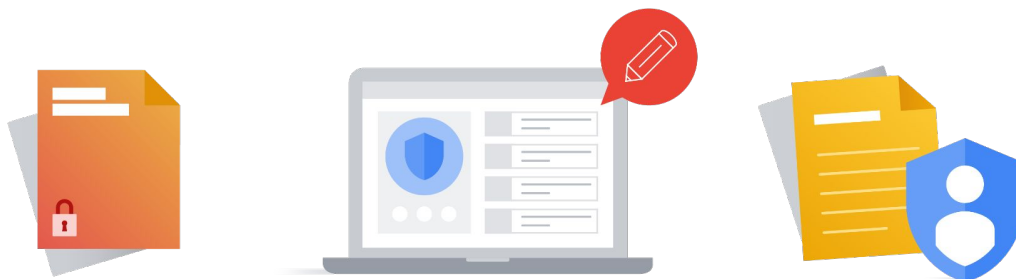
Periodically rotating secrets helps to:

- Limit the duration a leaked secret remains valid and exposes a vulnerability.
- Continuously exercise rotation flow to ensure process reliability.

For more information, see [Create rotation schedules in Secret Manager](#).

## 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.



Google Cloud

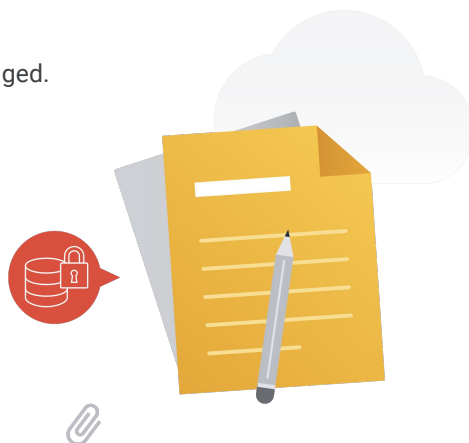
Admin Activity Access audit logs cannot be disabled. These logs do not count toward your log ingestion quota.

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.



# Lab Intro

Configuring and Using Credentials  
with Secret Manager



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

## Module review

- Application security is often ill-defined and untested in software development projects
- Focusing attention on common application vulnerabilities during development and deployment helps.
- Web Cloud Security Scanner can automatically scan and detect common application vulnerabilities.
  - It can be run on a schedule or on-demand.
  - It creates a load on your applications while running, however there are ways to reduce the impact of that load.



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

- 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.



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.