Google Cloud

Google Infrastructure Security Design Overview

Google Cloud Whitepaper

Google Cloud

Table of Contents

Introduction 2

Secure Low Level Infrastructure 3

Security of Physical Premises
Hardware Design and Provenance
Secure Boot Stack and Machine Identity

Secure Service Deployment 4

Service Identity, Integrity, and Isolation
Inter-Service Access Management
Encryption of Inter-Service Communication
Access Management of End User Data

Secure Data Storage 7

Encryption at Rest
Deletion of Data

Secure Internet Communication 8

Google Front End Service
Denial of Service (DoS) Protection
User Authentication

Operational Security 9

Safe Software Development
Keeping Employee Devices and Credentials Safe
Reducing Insider Risk
Intrusion Detection

Securing the Google Cloud Platform (GCP) 11

Conclusion 13

Additional Reading 13

The content contained herein is correct as of January 2017, and represents the status quo as of the time it was written. Google's security policies and systems may change going forward, as we continually improve protection for our customers.

CIO-level summary

- Google has a global scale technical infrastructure designed to provide security through the entire information processing lifecycle at Google. This infrastructure provides secure deployment of services, secure storage of data with end user privacy safeguards, secure communications between services, secure and private communication with customers over the internet, and safe operation by administrators.

- Google uses this infrastructure to build its internet services, including both consumer services such as Search, Gmail, and Photos, and enterprise services such as G Suite and Google Cloud Platform.

- The security of the infrastructure is designed in progressive layers starting from the physical security of data centers, continuing on to the security of the hardware and software that underlie the infrastructure, and finally, the technical constraints and processes in place to support operational security.

- Google invests heavily in securing its infrastructure with many hundreds of engineers dedicated to security and privacy distributed across all of Google, including many who are recognized industry authorities.

Introduction

This document gives an overview of how security is designed into Google's technical infrastructure. This global scale infrastructure is designed to provide security through the entire information processing lifecycle at Google. This infrastructure provides secure deployment of services, secure storage of data with end user privacy safeguards, secure communications between services, secure and private communication with customers over the internet, and safe operation by administrators.

Google uses this infrastructure to build its internet services, including both consumer services such as Search, Gmail, and Photos, and enterprise services such as G Suite and Google Cloud Platform.

We will describe the security of this infrastructure in progressive layers starting from the physical security of our data centers, continuing on to how the hardware and software that underlie the infrastructure are secured, and finally, describing the technical constraints and processes in place to support operational security.

Google Infrastructure Security Layers

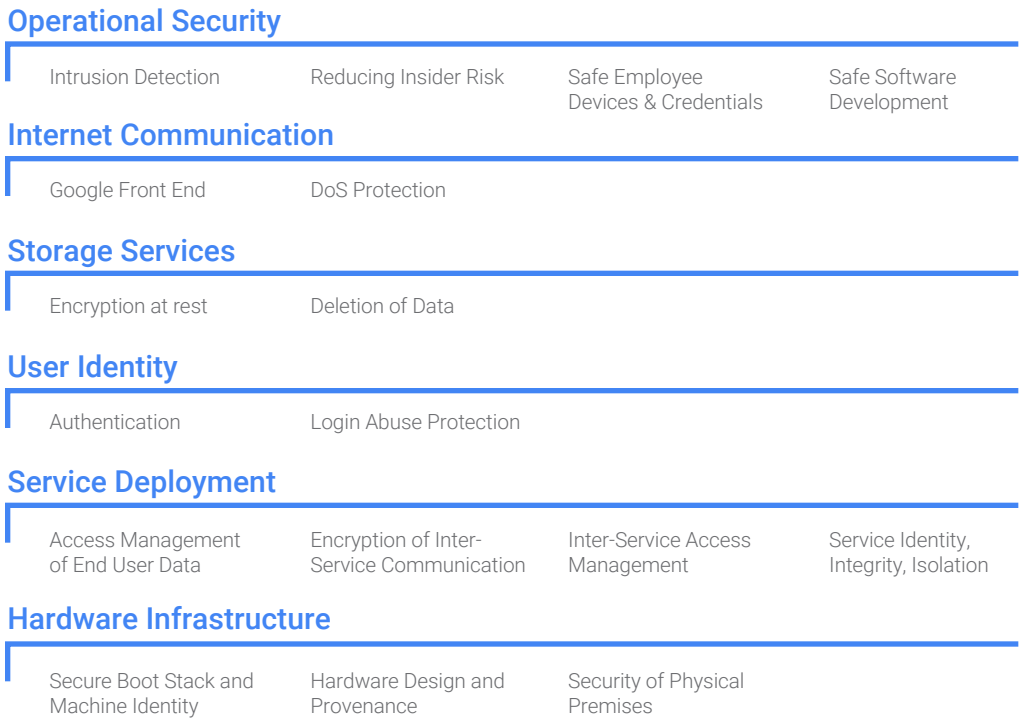

Figure 1.
Google Infrastructure Security Layers:
The various layers of security starting from hardware infrastructure at the bottom layer up to operational security at the top layer. The contents of each layer are described in detail in the paper.

Secure Low Level Infrastructure

In this section we describe how we secure the lowest layers of our infrastructure, ranging from the physical premises to the purpose-built hardware in our data centers to the low-level software stack running on every machine.

Security of Physical Premises

Google designs and builds its own data centers, which incorporate multiple layers of physical security protections. Access to these data centers is limited to only a very small fraction of Google employees. We use multiple physical security layers to protect our data center floors and use technologies like biometric identification, metal detection, cameras, vehicle barriers, and laser-based intrusion detection systems. Google additionally hosts some servers in third-party data centers, where we ensure that there are Google-controlled physical security measures on top of the security layers provided by the data center operator. For example, in such sites we may operate independent biometric identification systems, cameras, and metal detectors.

Hardware Design and Provenance

A Google data center consists of thousands of server machines connected to a local network. Both the server boards and the networking equipment are custom-designed by Google. We vet component vendors we work with and choose components with care, while working with vendors to audit and validate the security properties provided by the components. We also design custom chips, including a hardware security chip that is currently being deployed on both servers and peripherals. These chips allow us to securely identify and authenticate legitimate Google devices at the hardware level.

Secure Boot Stack and Machine Identity

Google server machines use a variety of technologies to ensure that they are booting the correct software stack. We use cryptographic signatures over low-level components like the BIOS, bootloader, kernel, and base operating system image. These signatures can be validated during each boot or update. The components are all Google-controlled, built, and hardened. With each new generation of hardware we strive to continually improve security: for example, depending on the generation of server design, we root the trust of the boot chain in either a lockable firmware chip, a microcontroller running Google-written security code, or the above mentioned Google-designed security chip.

Each server machine in the data center has its own specific identity that can be tied to the hardware root of trust and the software with which the machine booted. This identity is used to authenticate API calls to and from low-level management services on the machine.

Google has authored automated systems to ensure servers run up-to-date versions of their software stacks (including security patches), to detect and

A Google data center consists of thousands of server machines connected to a local network. Both the server boards and the networking equipment are custom designed by Google.

We use cryptographic authentication and authorization at the application layer for inter-service communication. This provides strong access control at an abstraction level and granularity that administrators and services can naturally understand.

diagnose hardware and software problems, and to remove machines from service if necessary.

Secure Service Deployment

We will now go on to describe how we go from the base hardware and software to ensuring that a service is deployed securely on our infrastructure. By 'service' we mean an application binary that a developer wrote and wants to run on our infrastructure, for example, a Gmail SMTP server, a BigTable storage server, a YouTube video transcoder, or an App Engine sandbox running a customer application. There may be thousands of machines running copies of the same service to handle the required scale of the workload. Services running on the infrastructure are controlled by a cluster orchestration service called Borg.

As we will see in this section, the infrastructure does not assume any trust between services running on the infrastructure. In other words, the infrastructure is fundamentally designed to be multi-tenant.

Service Identity, Integrity, and Isolation

We use cryptographic authentication and authorization at the application layer for inter-service communication. This provides strong access control at an abstraction level and granularity that administrators and services can naturally understand.

We do not rely on internal network segmentation or firewalling as our primary security mechanisms, though we do use ingress and egress filtering at various points in our network to prevent IP spoofing as a further security layer. This approach also helps us to maximize our network's performance and availability.

Each service that runs on the infrastructure has an associated service account identity. A service is provided cryptographic credentials that it can use to prove its identity when making or receiving remote procedure calls (RPCs) to other services. These identities are used by clients to ensure that they are talking to the correct intended server, and by servers to limit access to methods and data to particular clients.

Google's source code is stored in a central repository where both current and past versions of the service are auditable. The infrastructure can additionally be configured to require that a service's binaries be built from specific reviewed, checked in, and tested source code. Such code reviews require inspection and approval from at least one engineer other than the author, and the system enforces that code modifications to any system must be approved by the owners of that system. These requirements limit the ability of an insider or adversary to make malicious modifications to source code and also provide a forensic trail from a service back to its source.

We have a variety of isolation and sandboxing techniques for protecting a service from other services running on the same machine. These techniques include normal Linux user separation, language and kernel-based sandboxes, and hardware virtualization. In general, we use more layers of isolation for riskier workloads; for example, when running complex file format converters on user-supplied data or when running user supplied code for products like Google App Engine or Google Compute Engine. As an extra security boundary, we enable very sensitive services, such as the cluster orchestration service and some key management services, to run exclusively on dedicated machines.

Inter-Service Access Management

The owner of a service can use access management features provided by the infrastructure to specify exactly which other services can communicate with it. For example, a service may want to offer some APIs solely to a specific whitelist of other services. That service can be configured with the whitelist of the allowed service account identities and this access restriction is then automatically enforced by the infrastructure.

The owner of a service can use access management features provided by the infrastructure to specify exactly which other services can communicate with it.

Google engineers accessing services are also issued individual identities, so services can be similarly configured to allow or deny their accesses. All of these types of identities (machine, service, and employee) are in a global name space that the infrastructure maintains. As will be explained later in this document, end user identities are handled separately.

The infrastructure provides a rich identity management workflow system for these internal identities including approval chains, logging, and notification. For example, these identities can be assigned to access control groups via a system that allows two party-control where one engineer can propose a change to a group that another engineer (who is also an administrator of the group) must approve. This system allows secure access management processes to scale to the thousands of services running on the infrastructure.

In addition to the automatic API-level access control mechanism, the infrastructure also provides services the ability to read from central ACL and group databases so that they can implement their own custom, fine-grained access control where necessary.

Encryption of Inter-Service Communication

Beyond the RPC authentication and authorization capabilities discussed in the previous sections, the infrastructure also provides cryptographic privacy and integrity for RPC data on the network. To provide these security benefits to other application layer protocols such as HTTP, we encapsulate them inside our infrastructure RPC mechanisms. In essence, this gives application layer isolation and removes any dependency on the security of the network path. Encrypted inter-service communication can remain secure even if the network is tapped or a network device is compromised.

Services can configure the level of cryptographic protection they want for each infrastructure RPC (e.g. only configure integrity-level protection for low value data inside data centers). To protect against sophisticated adversaries who may be trying to tap our private WAN links, the infrastructure automatically encrypts all infrastructure RPC traffic which goes over the WAN between data centers, without requiring any explicit configuration from the service. We have started to deploy hardware cryptographic accelerators that will allow us to extend this default encryption to all infrastructure RPC traffic inside our data centers.

Access Management of End User Data

A typical Google service is written to do something for an end user. For example, an end user may store their email on Gmail. The end user's interaction with an application like Gmail spans other services within the infrastructure. So for example, the Gmail service may call an API provided by the Contacts service to access the end user's address book.

We have seen in the preceding section that the Contacts service can be configured such that the only RPC requests that are allowed are from the Gmail service (or from any other particular services that the Contacts service wants to allow).

This, however, is still a very broad set of permissions. Within the scope of this permission the Gmail service would be able to request the contacts of any user at any time.

Since the Gmail service makes an RPC request to the Contacts service on behalf of a particular end user, the infrastructure provides a capability for the Gmail service to present an "end user permission ticket" as part of the RPC. This ticket proves that the Gmail service is currently servicing a request on behalf of that particular end user. This enables the Contacts service to implement a safeguard where it only returns data for the end user named in the ticket.

The infrastructure provides a central user identity service which issues these "end user permission tickets". An end user login is verified by the central identity service which then issues a user credential, such as a cookie or OAuth token, to the user's client device. Every subsequent request from the client device into Google needs to present that user credential.

When a service receives an end user credential, it passes the credential to the central identity service for verification. If the end user credential verifies correctly, the central identity service returns a short-lived "end user permission ticket" that can be used for RPCs related to the request. In our example, that service which gets the "end user permission ticket" would be the Gmail service, which would pass it to the Contacts service. From that point on, for any cascading calls, the "end user permission ticket" can be handed down by the calling service to the callee as a part of the RPC call.

To protect against sophisticated adversaries who may be trying to tap our private WAN links, the infrastructure automatically encrypts all infrastructure RPC traffic which goes over the WAN between data centers.

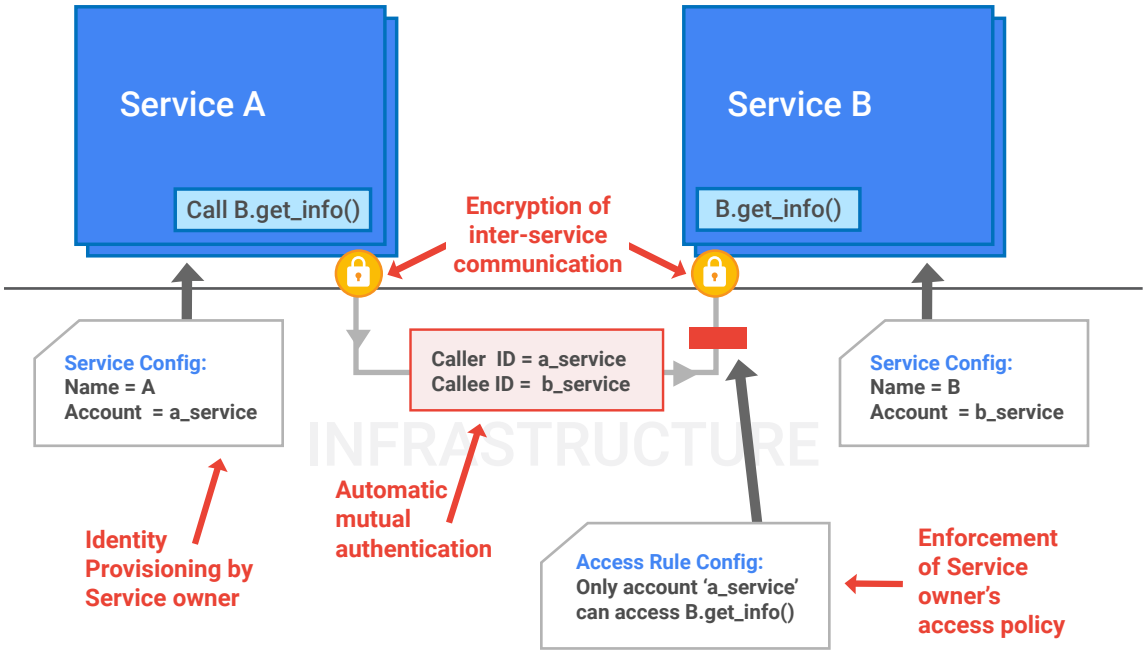

Figure 2.
Service Identity and Access Management: The infrastructure provides service identity, automatic mutual authentication, encrypted inter-service communication and enforcement of access policies defined by the service owner.

Secure Data Storage

Up to this point in the discussion, we have described how we deploy services securely. We now turn to discussing how we implement secure data storage on the infrastructure.

Encryption at Rest

Google's infrastructure provides a variety of storage services, such as BigTable and Spanner, and a central key management service. Most applications at Google access physical storage indirectly via these storage services. The storage services can be configured to use keys from the central key management service to encrypt data before it is written to physical storage. This key management service supports automatic key rotation, provides extensive audit logs, and integrates with the previously mentioned end user permission tickets to link keys to particular end users.

Performing encryption at the application layer allows the infrastructure to isolate itself from potential threats at the lower levels of storage such as malicious disk firmware. That said, the infrastructure also implements additional layers of protection. We enable hardware encryption support in our hard drives and SSDs and meticulously track each drive through its lifecycle. Before a decommissioned encrypted storage device can physically leave our custody, it is cleaned using a multi-step process that includes two independent verifications. Devices that do not pass this wiping procedure are physically destroyed (e.g. shredded) on-premise.

Deletion of Data

Deletion of data at Google most often starts with marking specific data as "scheduled for deletion" rather than actually removing the data entirely. This allows us to recover from unintentional deletions, whether customer-initiated or due to a bug or process error internally. After having been marked as "scheduled for deletion," the data is deleted in accordance with service-specific policies.

When an end user deletes their entire account, the infrastructure notifies services handling end user data that the account has been deleted. The services can then schedule data associated with the deleted end user account for deletion.

The Google Front End ensures that all TLS connections are terminated using correct certificates and following best practices such as supporting perfect forward secrecy.

Secure Internet Communication

Until this point in this document, we have described how we secure services on our infrastructure. In this section we turn to describing how we secure communication between the internet and these services.

As discussed earlier, the infrastructure consists of a large set of physical machines which are interconnected over the LAN and WAN and the security of inter-service communication is not dependent on the security of the network. However, we do isolate our infrastructure from the internet into a private IP space so that we can more easily implement additional protections such as defenses against denial of service (DoS) attacks by only exposing a subset of the machines directly to external internet traffic.

Google Front End Service

When a service wants to make itself available on the Internet, it can register itself with an infrastructure service called the Google Front End (GFE). The GFE ensures that all TLS connections are terminated using correct certificates and following best practices such as supporting perfect forward secrecy. The GFE additionally applies protections against Denial of Service attacks (which we will discuss in more detail later). The GFE then forwards requests for the service using the RPC security protocol discussed previously.

In effect, any internal service which chooses to publish itself externally uses the GFE as a smart reverse-proxy front end. This front end provides public IP hosting of its public DNS name, Denial of Service (DoS) protection, and TLS termination. Note that GFEs run on the infrastructure like any other service and thus have the ability to scale to match incoming request volumes.

Denial of Service (DoS) Protection

The sheer scale of our infrastructure enables Google to simply absorb many DoS attacks. That said, we have multi-tier, multi-layer DoS protections that further reduce the risk of any DoS impact on a service running behind a GFE.

After our backbone delivers an external connection to one of our data centers, it passes through several layers of hardware and software load-balancing. These load balancers report information about incoming traffic to a central DoS service running on the infrastructure. When the central DoS service detects that a DoS attack is taking place, it can configure the load balancers to drop or throttle traffic associated with the attack.

At the next layer, the GFE instances also report information about requests that they are receiving to the central DoS service, including application layer information that the load balancers don't have. The central DoS service can then also configure the GFE instances to drop or throttle attack traffic.

User Authentication

After DoS protection, the next layer of defense comes from our central identity service. This service usually manifests to end users as the Google login page. Beyond asking for a simple username and password, the service also intelligently challenges users for additional information based on risk factors such as whether they have logged in from the same device or a similar location in the past. After authenticating the user, the identity service issues credentials such as cookies and OAuth tokens that can be used for subsequent calls.

Users also have the option of employing second factors such as OTPs or phishing-resistant Security Keys when signing in. To ensure that the benefits go beyond Google, we have worked in the FIDO Alliance with multiple device vendors to develop the Universal 2nd Factor (U2F) open standard. These devices are now available in the market and other major web services also have followed us in implementing U2F support.

The sheer scale of our infrastructure enables Google to simply absorb many DoS attacks. That said, we have multi-tier, multi-layer DoS protections that further reduce the risk of any DoS impact on a service running behind a GFE.

Operational Security

Up to this point we have described how security is designed into our infrastructure and have also described some of the mechanisms for secure operation such as access controls on RPCs.

We now turn to describing how we actually operate the infrastructure securely: We create infrastructure software securely, we protect our employees' machines and credentials, and we defend against threats to the infrastructure from both insiders and external actors.

Safe Software Development

Beyond the central source control and two-party review features described earlier, we also provide libraries that prevent developers from introducing certain classes of security bugs. For example, we have libraries and frameworks that eliminate XSS vulnerabilities in web apps. We also have automated tools for automatically detecting security bugs including fuzzers, static analysis tools, and web security scanners.

As a final check, we use manual security reviews that range from quick triages for less risky features to in-depth design and implementation reviews for the most risky features. These reviews are conducted by a team that includes experts across web security, cryptography, and operating system security. The reviews can also result in new security library features and new fuzzers that can then be applied to other future products.

In addition, we run a Vulnerability Rewards Program where we pay anyone who is able to discover and inform us of bugs in our infrastructure or applications. We have paid several million dollars in rewards in this program.

Google also invests a large amount of effort in finding 0-day exploits and other security issues in all the open source software we use and upstreaming these issues. For example, the OpenSSL Heartbleed bug was found at Google and we are the largest submitter of CVEs and security bug fixes for the Linux KVM hypervisor.

We run a Vulnerability Rewards Program where we pay anyone who is able to discover and inform us of bugs in our infrastructure or applications.

Keeping Employee Devices and Credentials Safe

We make a heavy investment in protecting our employees' devices and credentials from compromise and also in monitoring activity to discover potential compromises or illicit insider activity. This is a critical part of our investment in ensuring that our infrastructure is operated safely.

Sophisticated phishing has been a persistent way to target our employees. To guard against this threat we have replaced phishable OTP second factors with mandatory use of U2F-compatible Security Keys for our employee accounts.

We make a large investment in monitoring the client devices that our employees use to operate our infrastructure. We ensure that the operating system images for these client devices are up-to-date with security patches and we control the applications that can be installed. We additionally have systems for scanning user-installed apps, downloads, browser extensions, and content browsed from the web for suitability on corp clients.

Being on the corporate LAN is not our primary mechanism for granting access privileges. We instead use application-level access management controls which allow us to expose internal applications to only specific users when they are coming from a correctly managed device and from expected networks and geographic locations. (For more detail see our additional reading about 'BeyondCorp'.)

Reducing Insider Risk

We aggressively limit and actively monitor the activities of employees who have been granted administrative access to the infrastructure and continually work to eliminate the need for privileged access for particular tasks by providing automation that can accomplish the same tasks in a safe and controlled way.

This includes requiring two-party approvals for some actions and introducing limited APIs that allow debugging without exposing sensitive information.

Google employee access to end user information can be logged through low-level infrastructure hooks. Google's security team actively monitors access patterns and investigates unusual events.

Intrusion Detection

Google has sophisticated data processing pipelines which integrate host-based signals on individual devices, network-based signals from various monitoring points in the infrastructure, and signals from infrastructure services. Rules and machine intelligence built on top of these pipelines give operational security engineers warnings of possible incidents. Our investigation and incident response teams triage, investigate, and respond to these potential incidents 24 hours a day, 365 days a year. We conduct Red Team exercises to measure and improve the effectiveness of our detection and response mechanisms.

Rules and machine intelligence built on top of signal monitoring pipelines give operational Security Engineers warnings of possible incidents.

Securing the Google Cloud Platform (GCP)

In this section, we highlight how our public cloud infrastructure, GCP, benefits from the security of the underlying infrastructure. In this section, we will take Google Compute Engine (GCE) as an example service and describe in detail the service-specific security improvements that we build on top of the infrastructure.

GCE enables customers to run their own virtual machines on Google's infrastructure. The GCE implementation consists of several logical components, most notably the management control plane and the virtual machines themselves.

The management control plane exposes the external API surface and orchestrates tasks like virtual machine creation and migration. It runs as a variety of services on the infrastructure, thus it automatically gets foundational integrity features such as a secure boot chain. The individual services run under distinct internal service accounts so that every service can be granted only the permissions it requires when making remote procedure calls (RPCs) to the rest of the control plane. As discussed earlier, the code for all of these services is stored in the central Google source code repository, and there is an audit trail between this code and the binaries that are eventually deployed.

The GCE control plane exposes its API via the GFE, and so it takes advantage of infrastructure security features like Denial of Service (DoS) protection and centrally managed SSL/TLS support. Customers can get similar protections for applications running on their GCE VMs by choosing to use the optional Google

Cloud Load Balancer service which is built on top of the GFE and can mitigate many types of DoS attacks.

End user authentication to the GCE control plane API is done via Google's centralized identity service which provides security features such as hijacking detection. Authorization is done using the central Cloud IAM service.

The network traffic for the control plane, both from the GFEs to the first service behind it and between other control plane services is automatically authenticated by the infrastructure and encrypted whenever it travels from one data center to another. Additionally, the infrastructure has been configured to encrypt some of the control plane traffic within the data center as well.

Each virtual machine (VM) runs with an associated virtual machine manager (VMM) service instance. The infrastructure provides these services with two identities. One identity is used by the VMM service instance for its own calls and one identity is used for calls that the VMM makes on behalf of the customer's VM. This allows us to further segment the trust placed in calls coming from the VMM.

GCE persistent disks are encrypted at-rest using keys protected by the central infrastructure key management system. This allows for automated rotation and central auditing of access to these keys.

The Google Compute Engine (GCE) control plane exposes its API via the Google Front-end (GFE), and so it takes advantage of infrastructure security features like Denial of Service (DoS) protection and centrally managed SSL/TLS support.

Customers today have the choice of whether to send traffic from their VMs to other VMs or the internet in the clear, or to implement any encryption they choose for this traffic. We have started rolling out automatic encryption for the WAN traversal hop of customer VM to VM traffic. As described earlier, all control plane WAN traffic within the infrastructure is already encrypted. In the future we plan to take advantage of the hardware-accelerated network encryption discussed earlier to also encrypt inter-VM LAN traffic within the data center.

The isolation provided to the VMs is based on hardware virtualization using the open source KVM stack. We have further hardened our particular implementation of KVM by moving some of the control and hardware emulation stack into an unprivileged process outside the kernel. We have also extensively tested the core of KVM using techniques like fuzzing, static analysis, and manual code review. As mentioned earlier, the majority of the recently publicly disclosed vulnerabilities which have been upstreamed into KVM came from Google.

Finally, our operational security controls are a key part of making sure that accesses to data follow our policies. As part of the Google Cloud Platform, GCE's use of customer data follows the GCP use of customer data policy, namely that Google will not access or use customer data, except as necessary to provide services to customers.

Conclusion

We have described how the Google infrastructure is designed to build, deploy and operate services securely at internet scale. This includes both consumer services such as Gmail and our enterprise services. In addition, our Google Cloud offerings are built on top of this same infrastructure.

We invest heavily in securing our infrastructure. We have many hundreds of engineers dedicated to security and privacy distributed across all of Google, including many who are recognized industry authorities.

We invest heavily in securing our infrastructure. We have many hundreds of engineers dedicated to security & privacy distributed across all of Google, including many who are recognized industry authorities.

As we have seen, the security in the infrastructure is designed in layers starting from the physical components and data center, to hardware provenance, and then on to secure boot, secure inter-service communication, secured data at rest, protected access to services from the internet and finally, the technologies and people processes we deploy for operational security.

Additional Reading

Please see the following papers for more detail on specific areas:

1. Physical security of our data centers
   https://goo.gl/WYlKGG
2. Design of our cluster management and orchestration
   http://research.google.com/pubs/pub43438.html
3. Storage encryption and our customer facing GCP encryption features
   https://cloud.google.com/security/encryption-at-rest/
4. BigTable storage service
   http://research.google.com/archive/bigtable.html
5. Spanner storage service
   http://research.google.com/archive/spanner.html
6. Architecture of our network load balancing
   http://research.google.com/pubs/pub44824.html
7. BeyondCorp approach to enterprise security
   http://research.google.com/pubs/pub43231.html

8. Combating phishing with Security Key & the Universal 2nd Factor (U2F) standard
   http://research.google.com/pubs/pub45409.html

9. More about the Google Vulnerability Rewards Program
   https://bughunter.withgoogle.com/

10. More about HTTPs and other load balancing offerings on GCP
    https://cloud.google.com/compute/docs/load-balancing/

11. More about DoS protection best practices on GCP
    https://cloud.google.com/files/GCPDDoSprotection-04122016.pdf

12. Google Cloud Platform use of customer data policy
    https://cloud.google.com/terms/

13. More about application security & compliance in G Suite (Gmail, Drive, etc)
    https://goo.gl/3J20R2