independent security evaluators

HARDENING GUIDE

Google Cloud Platform

Independent Security Evaluators | www.securityevaluators.com

December 21, 2017

INTRODUCTION

This document is meant for those who may be deploying a new system or infrastructure or migrating an existing instance to the Google Cloud Platform (GCP). While the GCP provides a variety of benefits, including ready-to-configure security controls and features, it is the responsibility of the deploying party to configure GCP properly to ensure that security goals are achieved. This guide is to be used by deployment administrators to understand the limitations of GCP. This guide also spells out security hardening recommendations for GCP deployments.

This guide is current as of December 2017. Changes to the GCP after this date may invalidate certain recommendations or introduce new concerns. Further, it is the responsibility of the users to understand their deployments and the security risks involved; in fact, Google takes no responsibility for security flaws on the end user's system.

DOMAINS OF CONCERN

When deploying a system in GCP, there are three principal domains of concern – for each of which the responsibility for security falls on a different party – and of which this document addresses only the second domain.

The *first domain* is the Google backend. Google provides an infrastructure to support individual cloud environments. This backbone provides the platform on which customer deployments exist. It supports the virtual machines, storage, computational power, redundancy and scalability aspects, and much more. Google provides for the security of this domain and all of its attack surfaces, and provides documentation online for how this is handled: https://cloud.google.com/security/

The *second domain* is the GCP configuration. These are the interfaces, features, and security options that may be chosen by those managing a GCP deployment. In many cases, GCP provides the direct support for security controls, however, it is the responsibility of the deploying party to use them correctly. For example, firewalls, databases, authentication mechanisms, VPNs, initialization scripts, and virtual machine templates are made available to administrators to better secure their deployment. This guide focuses on this security domain and its attack surfaces.

The *third domain* is the virtual operating system and running application software itself. These are the virtual systems that are deployed in the GCP by the customer. These virtual systems are not managed by Google, and are entirely the responsibility of the deploying party. Because the circumstances and types of deployments in GCP can range so dramatically, it is only possible to review the security of those deployments on a case-by-case. Any deployment that handles high values or sensitive data should undergo a third-party security evaluation.

ABOUT ISE

OUR MISSION

ISE is an independent security firm in Baltimore, Maryland dedicated to aggressive defense strategies through advanced science. Our elite team of analysts and developers use scientific approaches to improve our clients' overall security posture, protect digital assets, harden existing technologies, secure infrastructures, and work with development teams to ensure product security before deployment.

Our adversary-centric perspective allows us to understand not only the actions of threat actors, but to understand their mindset as well. Although our relationship with these nefarious elements is distant, it is also intimate: we interact with them regularly and bridge that distance by a passionate desire to understand them. Adopting their perspective in our own assessments allows us to identify future attack vectors, and prepare for them. The attacks are inevitable, but can be defeated with the right methodologies, the right defenses, and the right perspective.

RELATIONSHIP WITH GOOGLE

Google and one of Google's customers jointly engaged ISE to complete a white-box, design and implementation level assessment of the Cloud Platform. In the assessment we aimed to discover any instances of missing or broken security controls, security oversights, violations of best practices, or areas in which it is likely that Google's customers may accidentally misconfigure the Platform, increasing its exposure to security threats. The assessment was performed from a general perspective; we did not restrict the scope to any specific content workflow or software deployed within the Platform, nor have we considered the impact of any security concerns on a specific workflow or piece of software. The assessment concluded in August 2015. Re-assessments concluded in May 2016, October 2016, February 2017, September 2017, and December 2017.

ISE is an independent party to Google, and does not have a stake in the outcome of our assessments. ISE does not endorse Google products, or any other product, and is motivated only to improve the security of all products we are engaged to assess.

PRODUCTS AND SERVICES

This guide focuses on the following core GCP products and services. There are additional products and services available beyond those covered by this hardening guide.

DEVELOPERS CONSOLE

The Developers Console provides an administrator with the ability to manage the Cloud Platform and other Google services from a point-and-click web interface as an alternative to the gcloud command line utility, application programming interfaces, or client libraries.

APPLICATION PROGRAMMING INTERFACES (APIS) AND CLIENTS

Google provides RESTful APIs, which allow administrators to manage GCP services in an automated and scriptable fashion. In addition to coding against the core Google APIs themselves, an administrator can access them indirectly using the gcloud and gsutil command-line utilities, or from Node.js, Python, or Go using Google-provided gcloud libraries. More general API libraries allow programmers to access a subset of Cloud Platform APIs using other languages, such as Java and the .NET Framework.

COMPUTE ENGINE

The core component of GCP is the Compute Engine. The Compute Engine provides facilities for running virtual machines (VMs), as well as supporting infrastructure, such as disk images, snapshotting, zoning, and automatic migration. Other features, such as Click to Deploy and the Container Engine, allow users to more easily and consistently deploy VM instances using templates and other automatic deployment features.

APP ENGINE

The App Engine allows developers to upload application code directly into the Cloud Platform, and have it execute transparently, with the deployment, configuration, and management of VMs handled transparently and automatically behind the scenes.

NETWORKING

Networking is the second most important feature in the GCP behind the Compute Engine itself. In addition to traditional network services and security features—such as routing, firewalling, and DNS—the Platform's networking capabilities include TCP-layer and HTTP/HTTPS-layer load balancers and a site-to-site IPsec VPN.

STORAGE

The ability to provide high-capacity, high-throughput, high-availability, and low-cost storage is a main feature of any Cloud Platform, and one of the most compelling motivators driving traditional software developers to shift to a cloud-based infrastructure. Google provides Bigtable and Datastore NoSQL databases as part of the Platform, plus a variant of the traditional MySQL database that is Google managed. The Cloud Storage component provides an object store more similar to a traditional file system, including fine-grained access control lists for each object.

BIG DATA

Google places its Big Data services within the Cloud Platform console. These include the BigQuery high-speed query storage system, Dataflow batch processor, and Pub/Sub asynchronous messaging platform.

CONCERNS AND RECOMMENDATIONS

The following table summarizes the recommendations for deployments using the GCP. A detailed description of each concern and recommendation follows after.

Do not use personal Google accounts for administration; use Google Apps for Work accounts.
Take care to delete/not store OAuth session information.
Change all passwords after Click-to-Deploy.
Restrict access to Metadata Server.
Exercise diligence when using remote startup scripts.
Use only randomly generated keys from IPsec VPN; not human generated or weak keys.
Configure IPsec VPN to only use recommended security protocols.
Configure firewall by remove default permissive rules, and adding restrictive rules.
Restrict permissive SSH access to systems for configuration using whitelists or available IPsec VPN.
Manually close any web-based SSH sessions when logging out of the web console.
Perform security audit and security hardening on VM images before use; do not use Google-provided images as-is.
Configure SQL databases to allow only SSL connections.
Protect Cloud Storage asset URLs from leakage or exposure.
Be sure to manually log out of the Google Cloud Platform console when finished using it.
Use an up-to-date and modern browser to access the web console.
Review capabilities and security limitations/assumptions when deploying GPU-accelerated network rendering software and deploy an appropriate architecture after reviewing the content in the section, “GPU Accelerated Cloud Rendering.”

CONCERN: USE OF PERSONAL GOOGLE ACCOUNTS FOR ADMINISTRATION

Any Google account may be used to set up and administer Cloud Compute. If an employee uses a personal (@gmail.com) account to set up Google Cloud infrastructure, then the employer may have difficulty gaining control of the account if the employee later resigns or is terminated. Further, employers are unable to maintain security policies on personal accounts or otherwise audit the accounts' security.

We recommend using a Google Apps for Work account for administering Google Cloud deployments. This account should be secured using the advanced features provided by Google Apps for Work, and managed by the enterprise owning the deployment, and not individuals.

CONCERN: GCLOUD UTILITY CACHES SESSION INFORMATION

Users authenticate to GCP using a Google account email address and password. To prevent a user from needing to retype credentials upon every invocation of the gcloud command line utility, the utility persistently caches the user's OAuth session information (excluding passwords) on the local system. This presents a security threat should the local system used to log in become compromised, even at a later date. Storing the OAuth access_token, client_secret, and other session information on the file system means that if an adversary gains access to the ~/.config/gcloud/credentials file, or a copy of it, the attacker may then use the session information to gain unauthorized access to the Platform user's account.

We recommend deleting credentials after they are used. The unencrypted session information can be removed by deleting the ~/.config/gcloud/credentials file after using the utility. This ensures that credentials do not remain in persistent storage, potentially available to attackers.

We recommend using a RAM disk for the ~/.config/gcloud directory. This method ensures that gcloud credentials are never written to disk. To do so, create a RAM disk and then create a symbolic link from the ~/.config/gcloud/credentials file to a file located on the RAM disk. This can be done using the Linux tmpfs file system, or using analogous techniques on Windows or Mac OS X. This ensures that the credentials are stored only in volatile memory and lost upon unmounting the RAM disk or rebooting the machine.

CONCERN: CLICK TO DEPLOY FEATURE STORES DEFAULT PASSWORDS

Click to Deploy functionality in GCP allows users to quickly and easily deploy new VMs running a desired service using predefined templates. For ease of use, many of these templates automatically assign default passwords to the services running in the VM and allow the administrator to retrieve these passwords at any time. For example, after deploying a new LAMP (Linux, Apache, MySQL, PHP) instance, the Developers Console allows the user to obtain the MySQL administrator password for the instance.

If after deploying the instance the administrator does not immediately take steps to change the MySQL administrator password, then the password will be stored within the Platform indefinitely, and the database could be compromised in the event that an unauthorized user gains access to the Developers Console and displays the password.

We recommend changing all passwords immediately after deployment. Users of the Click to Deploy feature should ensure that they change all default passwords in the new instance immediately after it is deployed to ensure that current passwords are different from any that may have been stored on disk.

CONCERNS AND RECOMMENDATIONS

CONCERN: METADATA SERVER AUTHENTICATION

Compute Engine VM instances make use of an external metadata server to store important instance-specific information, such as authorized SSH keys, network configuration, and most importantly, OAuth credentials for the machine's session account. This avoids the need for administrators to embed this information within VM images, thus streamlining the VM deployment process. The metadata server is accessible over HTTP from within the VM using the IP address 169.254.169.254.

The metadata server requires that the HTTP header `Metadata-Flavor: Google` be present in all requests as an authentication mechanism. In the event that any third party code that an administrator deploys on a Compute Engine VM is vulnerable to server-side request forgery, it may be possible for an attacker to relay unauthorized requests to the metadata server containing the Metadata-Flavor header through the vulnerable application, circumventing the security it is intended to provide.

We recommend using a host firewall to restrict access to the metadata server. Administrators can use the iptables command on a Linux machine (or the corresponding equivalent on other operating systems) to restrict outgoing connections to the metadata server, e.g., to restrict access to the root account only:

```
        iptables -A OUTPUT -d 169.254.169.254 -m owner \! --owner-uid 0 -j DROP
```

Before performing this step, administrators should audit all software running as root for server-side request forgery and verify that no low-privileged applications require legitimate access to the API.

CONCERN: INTEGRITY OF REMOTE STARTUP SCRIPTS

Using the metadata API, an administrator may pass a custom startup script to a VM, and the VM will execute the script each time it boots. The startup script can be supplied in a number of ways: inline on the metadata server, on the machine's file system, on Google Storage, or using a generic HTTP or HTTPS URL.

The VM does not perform a cryptographic integrity check on the startup script before executing it with superuser privileges. Because of this, if such a script could be replaced or modified, the modification would go undetected and could result in a malicious party gaining superuser-level control over the affected VM. When the script is supplied from a trusted source, such as the file system or metadata server, this is not an issue. When the script is obtained from an external source within a different trust domain (such as Google Storage) or an unaffiliated HTTP or HTTPS server, this situation may pose a threat.

An attacker could exploit this issue by directly attacking the external source of the script. However, in the case of HTTP, neither the data channel nor the script itself are authenticated or encrypted, and traditional network-based attacks suffice.

We recommend exercising diligence when using remote startup scripts. Administrators who wish to use the startup script feature should be careful to avoid specifying a remote URL. When a remote URL must be used, never set the script URL to point to a server controlled by an outside entity, and never use an HTTP script URL.

CONCERNS AND RECOMMENDATIONS

CONCERN: IPSEC VPN SUPPORTS ONLY PRESHARED KEY AUTHENTICATION

Typical enterprise IPsec servers and clients support three types of authentication: preshared keys (analogous to passwords), digital signatures, and X.509 certificates. The Google Cloud VPN supports only preshared keys. These keys, particularly if they are human generated, are susceptible to the same types of attacks used to brute-force guess passwords.

We recommend the use of randomly generated keys <u>only</u>. Until Google implements support for signatures or certificates, users of the Google Cloud VPN should never make use of human-generated passwords as a preshared key. Instead, use a cryptographically secure random number generator to generate a series of random bytes with sufficient entropy (i.e., 128 bits or more).

CONCERN: IPSEC VPN SUPPORTS DEPRECATED CRYPTOGRAPHY

In contrast to SSL/TLS, configuring the set of permitted cipher suites at each end of an IPsec connection can be a manual and time-consuming process. The following concerns affect the configuration of the Cloud IPsec VPN:

- *IKEv1 protocol supported*. The Cloud IPsec VPN, for compatibility, supports both IKE version 1 (introduced in 1998) and IKE version 2 (introduced in 2005). One of the goals of IKEv2 was to improve security over IKEv1, including cryptographic weaknesses.[1]

- *HMAC-MD5 supported (IKEv2).* The Cloud IPsec VPN allows HMAC-MD5 to be used for integrity checking. HMAC-MD5 is deprecated due to weaknesses in the underlying MD5 algorithm.[2]

In addition, the Cloud IPsec VPN is affected by the following limitations, regardless of any security hardening steps that an administrator performs:

- *No HMAC algorithms stronger than SHA-1 supported in ESP phase (IKEv1 and IKEv2).* The Cloud IPsec VPN allows HMAC-SHA1 to be used for integrity checking during the ESP phase, but does not support HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, or any of the SHA-3 hash functions. Although HMAC-SHA1 is still considered secure, best practices dictate that SHA-256 or a stronger algorithm be used due to weaknesses in the underlying SHA-1 algorithm.

- *AES-GCM cipher suites do not support 192-bit or 256-bit keys (IKEv2).* While the user of authenticated encryption by the IPsec VPN improves performance and provides an alternative to HMAC-SHA1 for authentication, the VPN does not support AES-192-GCM or AES-256-GCM. As a result, administrators choosing to use GCM for integrity checking are limited to 128-bit AES keys.

We recommend configuring the IPsec VPN to <u>not use</u> deprecated protocols and modes of operations. Administrators should configure their IPsec clients in accordance with their security policies. For example, we recommend using the IKE version 2 protocol, 128-bit AES in GCM mode for encryption and integrity checking, SHA2-512 as a pseudorandom

[1] RFC 4306 Appendix A, https://tools.ietf.org/html/rfc4306#page-96
[2] http://tools.ietf.org/html/rfc6151

function, and the largest Diffie-Hellman group supported by the client for key agreement. Administrators choosing to deviate from this recommendation due to compatibility issues or a desire to use 192-bit or 256-bit AES should do so only after ensuring that their configurations comply with their organizations' security policies.

We caution users that some clients are configured with cipher suites, listed in order starting with the most preferred, so it is important to list the stronger algorithms and larger key sizes first.

CONCERN: DEFAULT FIREWALL ALLOWS ALL INTER-VM COMMUNICATIONS

While the default firewall in place when setting up GCP heavily restricts incoming traffic from the Internet, it is more permissive with regard to internal traffic—essentially allowing open communication between all VM instances within the same zone. It is important that customers realize the need to harden the default firewall before allowing an environment to shift into production.

We recommend configuring the firewall to restrict inter-VM communications. Administrators should remove the default firewall rule, allowing inter-VM communication, and replace it with a deny-all policy with specific IP/protocol/port-base exceptions.

CONCERN: DEFAULT FIREWALL ALLOWS INCOMING INTERNET SSH TRAFFIC

The default firewall policy allows incoming SSH traffic to any newly created VMs. While this is legitimate in order to allow users to access and administer their machines, it increases the out-of-the-box attack surface of Google Cloud VMs. In particular, the OpenSSH server software has been affected by a number of security vulnerabilities over its existence, some have allowed unauthenticated superuser-level access. In the event that a similar vulnerability appears in the future, Cloud Platform customers employing the default firewall could be exposed.

We recommend restricting SSH access to a whitelist only. Administrators should ensure that they remove the default firewall rule that allows all Internet communication to TCP port 22. Administrators should then replace the default firewall rule with a rule permitting access only from a whitelisted source IP range.

We recommend restricting SSH access to connections established over the provided IPsec VPN. As an alternative to whitelisting, administrators should use the IPsec VPN for administering the VMs, allowing Internet-facing SSH to be disabled entirely. The appropriate rules to restrict all access to SSH from the WAN should be applied.

CONCERN: SSH CONNECTIONS TO VM INSTANCES ARE NOT TERMINATED ON LOGOUT

Web-based SSH sessions open in a separate browser window when the SSH option is selected. This browser based SSH session is not terminated when a user logs out of the web console. Inactivity timeouts are implemented on SSH connections, however, the timeout interval is at least 30 minutes. This provides an adversary within an additional attack vector to compromise user assets and accounts.

We recommend educating privileged users of this concern and instructing them to close their SSH session by closing the newly created browser window it was created in when they are finished using the web console.

CONCERN: PRE-BUILT OS IMAGES ARE SUPPLIED IN AN UNHARDENED STATE

Within the Google Cloud Platform there are a number of pre-built operating system images available to users for rapid deployment. The images provided include a range of Linux distributions (including CentOS and Ubuntu) as well as Windows 2008 R2 and Windows 2012 R2. In the current state, the OS images are supplied with default configurations. Some default configurations of OS images leave VM instances vulnerable to vulnerabilities which could be exploited by publically known, readily available proof of concepts and exploits. For example, the images may have their SSL/TLS configurations set up for maximum compatibility, rather than security.

We recommend hardening all supplied operating system images by current best practices. Hardening steps should include: closing unused ports, uninstalling unnecessary applications, updating software, configuring web servers using current best practices, and setting firewall rules.

CONCERN: DEFAULT SQL DATABASE CONFIGURATIONS ALLOW UNSECURED COMMUNICATIONS

The default configurations of both first and second generation SQL databases allow users to connect and administer them over unsecured communication channels. An adversary with a privileged position on the same network as the client could sniff network traffic, which may contain user credentials, using a packet sniffing application such as Wireshark.

We recommend restricting connections to databases via SSL by default. Google should continue to provide an option for users to communicate of unsecure channel due to user preference.

CONCERN: ASSET AUTHENTICATION INFORMATION PASSED IN URL

The Google Cloud Platform provides users with the ability to store assets in buckets. Buckets and files that reside in buckets have fine grained access controls for users that belong to a project. These permissions include reader, writer, and owner access levels.

Cloud Storage security controls operate by restricting who can obtain a download URL based on the configured access controls. After performing the authentication check, the server provides the user with a pre-signed URL containing authentication data needed to access the asset.

Once this URL is obtained, authentication is not required to access the content; the download URL could be (deliberately or inadvertently) forwarded, or it could be compromised by an attacker who gains access to the affected user’s browser history, browser cache, or bookmarks

We recommend users are educated and instructed to safeguard asset URLs while viewing locally and/or sharing URLs by utilizing a secure transmission channel.

CONCERN: LACK OF CUSTOM/TIMELY SESSION INACTIVITY TERMINATION

User sessions are not terminated after a 48-hour period of inactivity. Failing to provide an inactivity timeout of sufficiently short duration leaves Google user's sessions susceptible to unauthorized access in conjunction with unrelated client-side attacks.

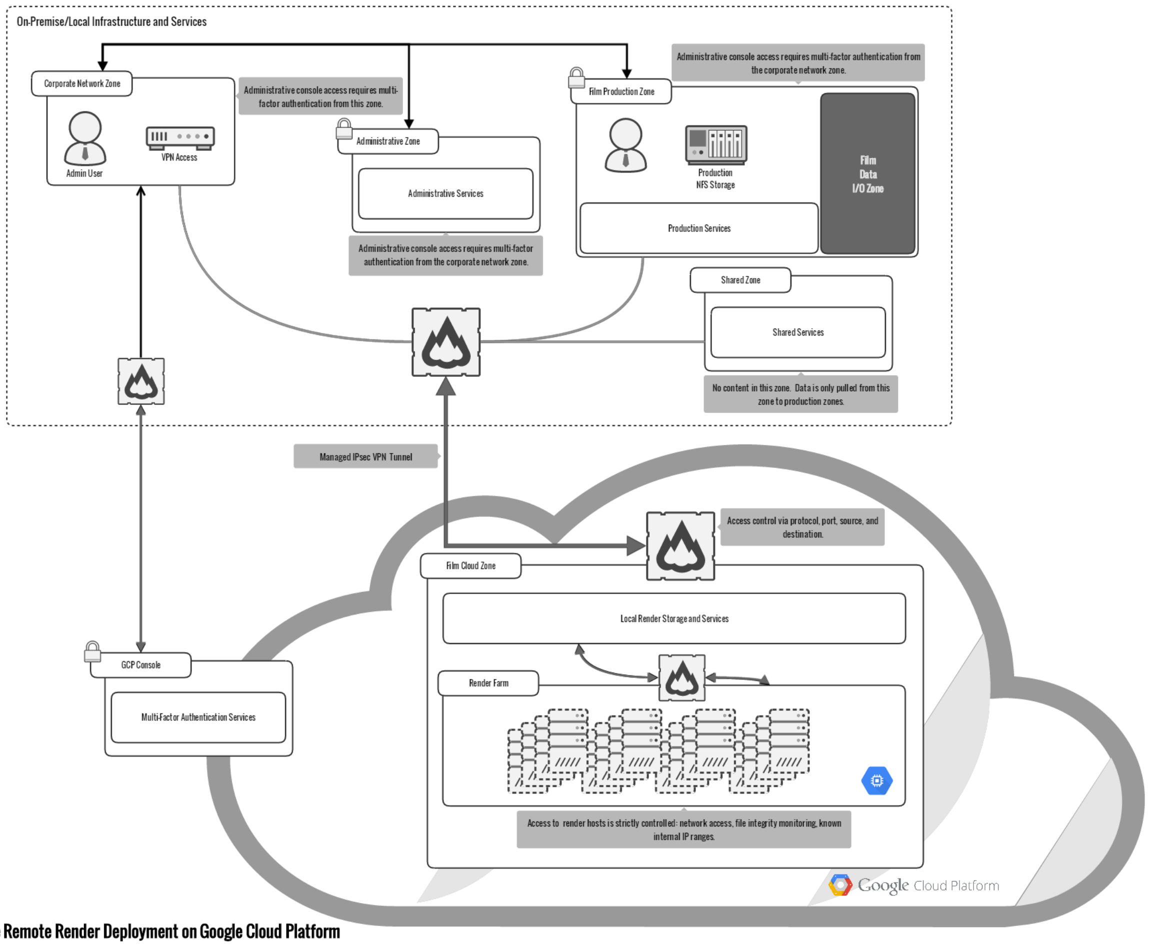
We recommend users should immediately terminate all Google Cloud Platform sessions when access to the platform is no longer needed.

CONCERN: NO FEATURE TO DISABLE CONCURRENT ACCESS FROM MULTIPLE LOCATIONS

A user has the ability to be logged into the platform concurrently from multiple computer systems, and there does not appear to be a feature allowing users or administrators to automatically log out all other sessions upon successful authentication. Highly security-conscious enterprises often desire such a feature to reduce the possibility that a user may inadvertently leave an active session unattended in a non-secure location (e.g., conference room computer) and login again elsewhere. The Cloud Platform does not offer such an automatic feature.

We recommend users should immediately terminate all Google Cloud Platform sessions when access to the platform is no longer needed.

GCP EXAMPLE SECURE DEPLOYMENT GRAPHIC


On-Premise/Local Infrastructure and Services
Corporate Network Zone
Admin User
VPN Access
Administrative console access requires multi-factor authentication from this zone.
Administrative Zone
Administrative Services
Administrative console access requires multi-factor authentication from the corporate network zone.
Film Production Zone
Administrative console access requires multi-factor authentication from the corporate network zone.
Production NFS Storage
Film Data I/O Zone
Production Services
Shared Zone
Shared Services
No content in this zone. Data is only pulled from this zone to production zones.
Managed IPsec VPN Tunnel
Access control via protocol, port, source, and destination.
Film Cloud Zone
Local Render Storage and Services
Render Farm
Access to render hosts is strictly controlled: network access, file integrity monitoring, known internal IP ranges.
GCP Console
Multi-Factor Authentication Services
Google Cloud Platform

Secure Remote Render Deployment on Google Cloud Platform
December 18, 2015

CONCERNS AND RECOMMENDATIONS

GPU Accelerated Cloud Rendering

Part of the value proposition of Google Cloud's "GPUs on Compute Instances" feature is to allows customers access to GPU-accelerated compute power without a substantial investment in on-premises GPU hardware. Among the computations that may benefit is rendering of three-dimensional graphics. Rendering is a broad field across the media and entertainment industry encompassing modeling products such as Blender, NUKE, and Autodesk Maya and 3ds Max; and rendering engines such as V-Ray RT, Octane, Redshift, and Blender. Broadly, GPU-accelerated rendering may be deployed in an interactive or batch mode of operation. Cloud rendering nodes may render locally-available modeling data stored in the cloud (e.g., in batch mode, or on behalf of interactive modeling software running in the cloud), or alternatively, they may receive network commands from remote interactive rendering software running outside of the cloud environment. Our concerns and recommendations aim to address both scenarios.

Network speed and business requirements mean that the deployments we have sketched here may not be appropriate or applicable for a particular need, but the security recommendations apply in any case.

Scenario 1: Total Cloud Rendering with OctaneRender v3

OctaneRender is capable of operating as a rendering engine with support for 3dsMax, Maya, Rhino, Modo, Nuke, and Inventor. OctaneRender may execute on a single machine provisioned with a GPU or using an arbitrarily large cluster of machines using the network rendering feature.

OctaneRender network rendering is designed to operate "over multiple computers connected through a fast local area network," see https://docs.otoy.com/manuals/products/standalone/v3/network-rendering/overview/. As a possible deployment, we envisioned running modeling software on a single machine within Google Cloud and accessed remotely using Windows Remote Desktop or similar remote access software, connected to an OctaneRender master and a set of rendering slaves located in the same Google Cloud environment as the interactive machine. Content enters and leaves the cloud environment only over transient file transfer connections; it remains in the environment for processing and rendering from the interactive computer system. Figure 1 depicts a sketch of a possible deployment. Security-relevant challenges pertaining to OctaneRender involve the licensing model: as of version 3, OctaneRender software connects to Otoy's Internet-facing licensing server to acquire and validate licenses (see https://docs.otoy.com/manuals/products/sketchup/v3/installation/activation/). For this reason, OctaneRender instances must have, at a minimum, whitelisted Internet access to the Otoy servers.

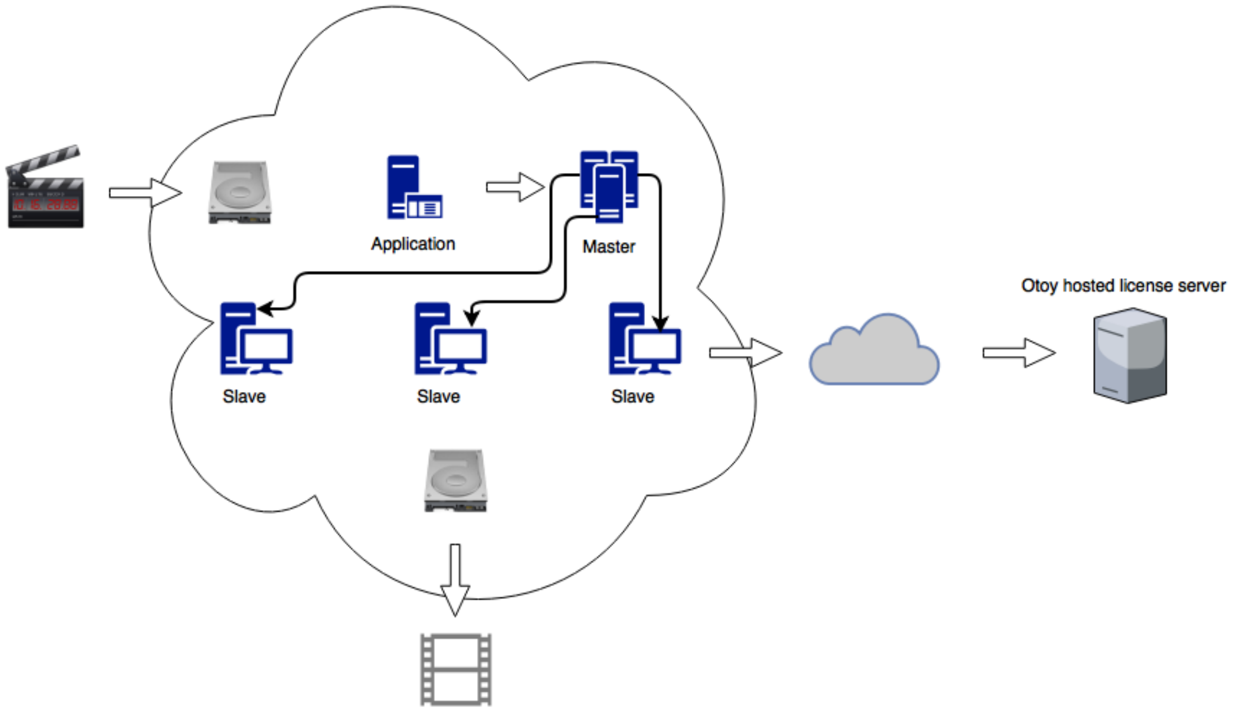


Figure 1. Sketch of total cloud rendering deployment.

RECOMMENDATIONS

We recommend considering the following when deploying cloud rendering as a "total cloud" deployment (raw modeling data, rendering software, and output all contained within the cloud).

- Ensure that the means of loading modeling input data, and retrieving outputs from the cloud are secure.
    - Use SFTP, SCP, FTP over TLS, or a commercial product specializing in providing secure file transfers in an Internet-facing environment, such as Signiant Managers+Agents or Aspera Enterprise Server.
    - Harden the protocol or product as appropriate for an environment hosting pre-release media content.
        - Securely distribute SSH host key fingerprints, and ensure that users verify them when needed.
        - Apply normal SSH hardening practices for SFTP/SCP-only users, such as disabling shell access and port forwarding.
        - Use SSH asymmetric key authentication in place of password authentication when practical.
        - For TLS-based protocols (such as FTP over TLS) ensure that the client software correctly and effectively verifies the server's hostname and certificate.

        - For commercial file transfer products, harden the configuration as recommended by the vendor.
    - Restrict connections to file transfer software by source IP ranges, if possible. Use the Google Cloud Networking "Firewall rules" option to whitelist specific source and destination IP addresses and destination ports, and name the firewall rules appropriately so that it is easy to identify their role in the rendering environment.
    - If the deployment *must* employ a less secure or robust file transfer protocol for compatibility reasons (e.g., unencrypted FTP, NFS, or SMB), then use Google Cloud VPN to encapsulate this traffic as a defense-in-depth mechanism, both to protect the traffic from eavesdropping as it travels over the Internet, and avoid exposing the server software to incoming connections from the Internet.
        - Review the recommendations provided in this hardening guide for configuring Google Cloud VPN.
- Isolate the render master and slaves from the Internet as much as possible, to reduce the environment's attack surface.
    - Do not expose administrative ports (e.g., Windows Remote Desktop, or Secure Shell) to the Internet directly. Instead, use a bastion host or Google Cloud VPN to administer the render master and slaves. This avoids the need to expose each rendering node to incoming Internet traffic, and reduces the urgency in applying patches to administrative software in event of a vulnerability disclosure.
    - Consider using Microsoft WSUS to distribute updates to Windows-based render nodes, or host a local repository mirror for Linux-based nodes. This eliminates the need to allow direct outbound Internet connections from these nodes.
    - Whitelist outbound traffic from render nodes to needed destinations only, e.g., the Otoy-hosted Internet-facing license server. Because Google Cloud (as of this writing) does not offer a means to restrict outbound traffic, use a host-based firewall such as iptables.
- Implement a central management scheme for the render nodes.
    - Consider Active Directory (Windows) or LDAP+Kerberos (Linux) to ease user provisioning and removal from a central location.
    - Consider automated deployment and configuration management such as Group Policy (Windows) or Chef, Ansible, or Puppet.

Scenario 2: Interactive Cloud Render Farm with V-Ray RT

V-Ray RT is capable of operating as a rendering engine with support for Maya, 3dsMax, Cinema 4D, Modo, Nuke, SketchUp, and Blender, or in standalone batch mode. V-Ray may execute on a single machine provisioned with a GPU, or using an arbitrarily large cluster of machines using the distributed rendering feature.

V-Ray distributed rendering consists of a distributed rendering client (which runs on the local user's machine running the modeling software) plus one or more distributed rendering servers. As a possible deployment, we envision running modeling software on-premises on a local user's machine. The local machine also runs V-Ray Render Client and has access to a set of V-Ray Render Servers running on GPU compute instances in the Google Cloud Platform. Content enters and leaves the cloud environment over V-Ray's protocol (TCP port 20206); these connections occur only over a Google Cloud VPN tunnel. There are no means to transfer and persistently store sensitive data in the cloud environment, except to the extent that V-Ray does so to support its operation. Note that rendering may require copies of certain resources such as textures to be available at each node; how these are copied and made available would be determined by the customer when designing a deployment. Figure 2 displays a sketch of a possible deployment. Security-relevant challenges pertaining to V-Ray involve the licensing model: render nodes communicate with a license server, which must be hosted on a physical machine with USB licensing dongles attached. In addition, the V-Ray client-server distributed rendering protocol is custom and does not appear to provide authentication, encryption, or other security, and should be protected using a VPN rather than being directly exposed to the Internet.

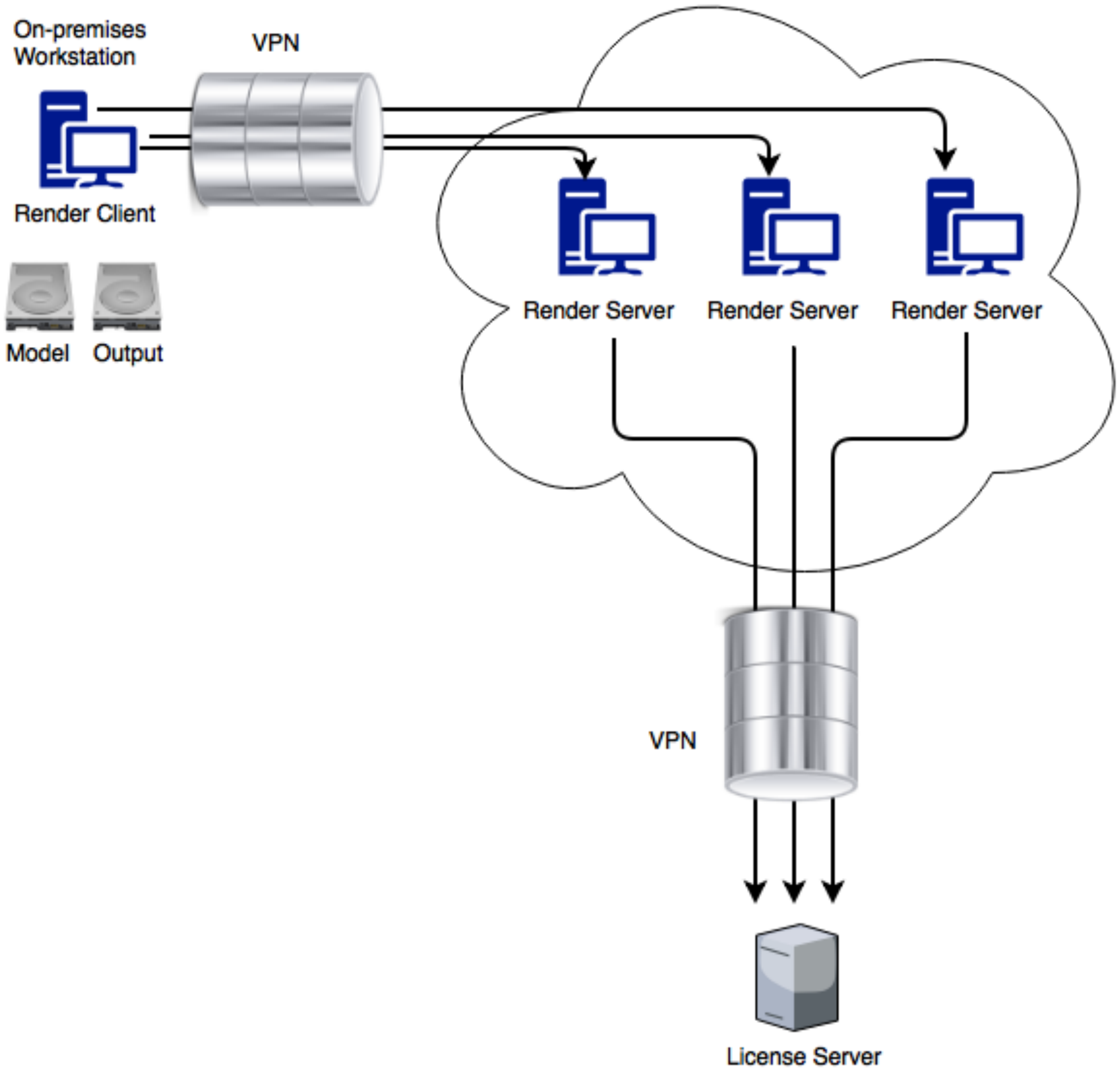


Figure 2. Sketch of interactive cloud render farm.

RECOMMENDATIONS

CONCERNS AND RECOMMENDATIONS

We recommend considering the following when deploying cloud rendering as an “interactive cloud render farm” deployment (render nodes and transient data contained within the cloud; modeling software and persistent input and output stored on-premises).

- All Internet-bound rendering and licensing communications to and from render clients/servers must occur within a VPN tunnel.
    - Based on our review of the documentation (https://docs.chaosgroup.com/display/VRAY3MAX/Set+Up+Distributed+Rendering), V-Ray distributed render communications over port 20206 are not password-protected or otherwise authenticated, and do not appear to provide encryption.
    - Based on our review of the documentation, V-Ray license communications over port 30304 are not password-protected or otherwise authenticated.
    - Both distributed rendering and license communications appear to use custom protocols. Without performing a security assessment to review the software’s input validation, sanitization, and other security controls, the software should not be considered robust enough to withstand attack from direct Internet-facing traffic.
- Network access controls for traffic traversing the VPN should be applied on a least-privilege basis.
    - Set up a dedicated on-premises subnet to host machines used by interactive users running the rendering client and modeling software (e.g., Maya or 3ds Max)
        - Each machine on this subnet requires unrestricted access to port 20206 on each render server.
        - Because there is no authentication on distributed render servers, presence on this network subnet, alone, regulates authorization to perform distributed render jobs.
        - Apply the least privilege principle to restrict unauthorized rendering; place no machines other than those performing rendering on the render client subnet.
    - Set up a dedicated on-premises subnet to host the license server and attached USB licensing dongles.
        - Each cloud render server requires unrestricted access to port 30304 on the license server.
        - On-premises render clients may require access to the license server as well.
    - Restrict traffic from the render client subnet to the cloud render servers.
        - Only traffic to port 20206 (or other necessary ports) should be permitted.
        - Use Google Cloud Network “Firewall rules” configuration and host-based firewalls on the render servers to restrict traffic.
    - Restrict traffic from the cloud render servers to the render client subnet.
        - No new connections should be allowed.
        - Use access controls on the on-premises VPN device and host-based firewalls on the render client machines to restrict traffic. This way, even a compromised Google Cloud Console account may not be used to evade access controls.
    - Restrict traffic from the cloud render servers to the license server subnet.
        - Only traffic to port 30304 (or other necessary ports) should be permitted.

        - Use access controls on the on-premises VPN device and host-based firewalls on the license server to restrict traffic. This way, even a compromised Google Cloud Console account may not be used to evade access controls.
    - Restrict traffic from the license server subnet to the cloud render servers.
        - No new connections should be allowed.
        - Use Google Cloud Network “Firewall rules” configuration and/or host-based firewalls on the render servers to restrict traffic.
- Isolate the render servers from the Internet as much as possible, to reduce the environment’s attack surface.
    - Do not expose administrative ports (e.g., Windows Remote Desktop, or Secure Shell) to the Internet directly. Instead, use an on-premises administrative subnet over the Google Cloud VPN to administer the render servers. This avoids the need to expose each rendering node to incoming Internet traffic, and reduces the urgency in applying patches to administrative software in the event of a vulnerability disclosure.
    - Consider using Microsoft WSUS to distribute updates to Windows-based render servers, or host a local repository mirror for Linux-based servers. This eliminates the need to allow direct outbound Internet connections from these servers.
    - Whitelist outbound traffic from render servers to needed destinations only, e.g., the license server. Because Google Cloud (as of this writing) does not offer a means to restrict outbound traffic, use a host-based firewall such as iptables, and incoming traffic rules on the on-premises VPN device.
- Implement a central management scheme for the render servers.
    - Consider Active Directory (Windows) or LDAP+Kerberos (Linux) to ease user provisioning and removal from a central location.
    - Consider automated deployment and configuration management such as Group Policy (Windows) or Chef, Ansible, or Puppet.

Scenario 3: Batch Mode Cloud Render Farm with Blender

Blender is a 3D creation suite that encompasses the entire 3D pipeline. Blender may be executed on a single machine provisioned with a GPU, or using an arbitrary large cluster of machines using the network rendering feature.

Blender network rendering consists of a rendering client (which runs Blender in “client mode”), a master (which runs Blender in “master mode”), and one or more rendering slaves (which runs Blender in “slave mode”) normally all housed within the same subnet. As a possible deployment, we envision running a single render client outside of Google Cloud, connected to a master node that resides within Google Cloud, via VPN, and several slave nodes—which the master node connects to. Content enters and leaves the cloud environment over Blender’s protocol; these connections only occur over a Google Cloud VPN tunnel. Models and rendered output will persistently reside on the master node until removed, which are accessible by Blender clients. Figure 3 depicts a sketch of a possible deployment.

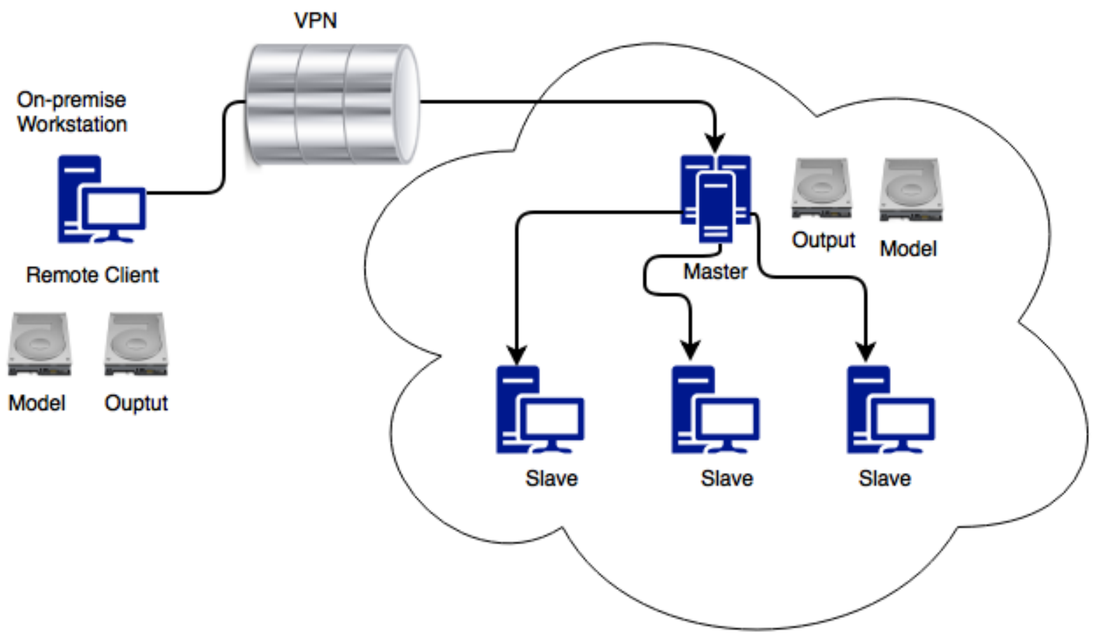


Figure 3. Sketch of batch cloud rendering farm.

RECOMMENDATIONS

We recommend the following when deploying cloud rendering as a "batch cloud rendering farm" deployment (raw modeling data, rendering software, and output contained within the cloud as well as on-premises).

- Isolate the render master and slaves from the Internet as much as possible, to reduce the environment's attack surface.
    - Do not expose administrative ports (e.g., Windows Remote Desktop, or Secure Shell) to the Internet directly. Instead, use a bastion host or Google Cloud VPN to administer the render master and slaves. This avoids the need to expose each rendering node to incoming Internet traffic, and reduces the urgency in applying patches to administrative software in the event of a vulnerability disclosure.
    - Consider using Microsoft WSUS to distribute updates to Windows-based render nodes, or host a local repository mirror for Linux-based nodes. This eliminates the need to allow direct outbound Internet connections from these nodes.
    - Whitelist outbound traffic from render slaves to needed destinations only, i.g., master node. Because Google Cloud (as of this writing) does not offer a means to restrict outbound traffic, use a host-based firewall such as iptables.
- Implement a central management scheme for the render nodes.

  - Consider Active Directory (Windows) or LDAP+Kerberos (Linux) to ease user provisioning and removal from a central location.
  - Consider automated deployment and configuration management such as Group Policy (Windows) or Chef, Ansible, or Puppet.

VPC Network Peering

Google Cloud Platform allows users to join VPC networks together across both projects and organizations. This allows for network traffic to flow freely between the peered networks in private RFC1918[3] space, so long as the peered networks do not have any overlapping IP address/CIDR ranges. The networks are "peered" through dual-association, meaning that the administrator for each network which is to be joined together must explicitly allow the association between their networks before the peering occurs. Without both associations, networks will not be joined.

Google states that this feature provides a number of benefits to users over competing technologies, such as decreased latency and cost and increased security. The media and entertainment industry could conceivably use this feature in their workflows, and as such, the limitations of this functionality are of concern. These limitations are discussed below.

CONCERN: Firewall Rules Applied to Networks Not Applied to Network Traffic Between Peered Networks

When joining networks together via the peering feature, all network traffic is allowed to freely flow between the networks by default. Network traffic both inbound and outbound to outside hosts/networks will still have the proper firewall rules applied. However, firewall rules from the networks being peered are not applied to traffic which flows between the peered networks. In some cases, this may inadvertently expose the machines on a network via an already compromised machine within this created network. It is understandable that insecure networks may need to be peered through the VPC under certain circumstances. When doing so, extra care must be taken to secure the machines within the existing network via proper host-based firewall rules which limit the network traffic allowed to and from machines based on expectations of normal network traffic.

We recommend VPC network peering only be used with trusted and properly managed networks. However, in the event that an untrusted network must be peered, proper host-based firewall rules should be configured per-host in the network which is being peered, limiting network traffic to only that which is expected under normal use.

CONCERN: Additional Networking Limitations

Network peering does not provide a set of fine-grained routing controls. Routes cannot overlap between the peered networks when a peering connection is created, though any new routes applied to a VPC network will be checked to ensure they do not overlap with those in any network that is peered. Static routes and VPNs will not be propagated to other peered networks when they are created within a single network that is part of a peering association, and network IP address/CIDR ranges cannot overlap upon the creation of a peering association. For example, VPC networks which only make use of the default subnets (e.g. auto-mode), are not able to be peered together.

Network peering is limited to no more than 25 networks being part of a peered association at any single time. This, along with other limitations noted above, make the network peering feature difficult to maintain for some use-cases within the media and entertainment industry. Therefore, network engineers will need to take special care when planning peering connections. They must ensure routes and IP address/CIDR ranges don't overlap before the peering association

[3] https://tools.ietf.org/html/rfc1918

is created. Furthermore, they must understand that many network-specific configurations are not propagated to other networks within a peering association.

We recommend VPC network peering be appropriately planned by network engineers and used only with trusted and properly managed networks. However, in the event that an untrusted network must be peered, proper host-based firewall rules should be configured per-host in the network which is being peered, limiting network traffic to only that which is expected under normal use.

Dedicated Interconnect

Google Cloud Platform gives users the option of directly connecting an on-premises network physically to Google via a colocation facility defined by Google. This service, known as directed interconnect, is meant to enable communication between an on-premises network and Google's network over private RFC1918[4] space, so long as the joined networks do not have any overlapping IP address/CIDR ranges.

CONCERN: Network Traffic Between Directed Interconnect Endpoints is Unencrypted

When joining an on premises network to Google via Directed Interconnect, all network traffic between these two networks is unencrypted. Google does not offer any in-transit encryption of network traffic as part of directed interconnect itself. As such, it is left up to the customer to implement network traffic encryption if they require it.

We recommend network engineers enable application-layer security or a VPN solution for traffic flowing between networks connected via directed interconnect to ensure such network traffic is encrypted.

[4] https://tools.ietf.org/html/rfc1918