How Halo Works: a Technical Summary

dbice
posted this on Jan 27, 15:42

How Halo Works

A Technical Summary

Contents

What is Halo?

Halo Architecture

Halo Security Modules

Halo Platform Services

Integrating Halo into Server Orchestration

Halo Accounts, Users, and Roles

Learn More About Halo

What is Halo?

CloudPassage® Halo® is a software-defined security (SDSec) platform that is purpose-built to automate and orchestrate security and compliance across any mix of cloud, virtualized, or bare-metal infrastructure. It is a software-as-a-service (SaaS) offering that relieves you of the burden of designing, purchasing, implementing, testing, and upgrading your own hardware and software security solutions.

Halo is a highly automated and it integrates easily with your existing infrastructure. It usually can be deployed in under an hour.

In action, Halo continually monitors the state of all your server hosts, conducting regular scans of your entire infrastructure to detect any changes, misconfigurations, or unauthorized logins that might indicate attack or compromise. It alerts you to the existence of any vulnerabilities that attackers might exploit. Halo also applies strong, customizable firewall protection and multi-factor authentication to all of your hosts to prevent unauthorized access.

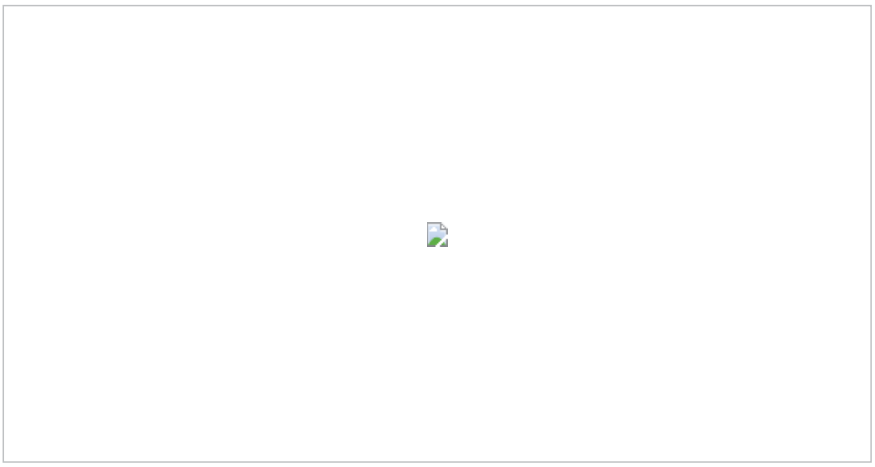
Halo gives you unprecedented visibility and control all across your infrastructure, and it supports both auditing and rapid response with its extensive logging, alerting, and reporting capabilities. Further, it is simple to use the Halo REST API to automate many of its features, and to integrate Halo with sophisticated log-analysis and SIEM solutions to provide even more in-depth analysis when you need it.

Halo Architecture

Halo is built with a patented distributed architecture that provides for maximum security power with minimum performance burden on your organization's resources. Halo's extreme scalability and dynamic coverage allow it to keep up with rapid changes in the most elastic of cloud implementations.

Halo cloud components

In use, the components of Halo are distributed across your organization's clouds and the Halo cloud, as shown below.



- The **Halo agent** is a lightweight and secure software component that takes up only a few MB of memory and runs as a service on each server instance, no matter where the instance resides. The agent automatically collects factual data about the state of the server and communicates it to the Halo analytics engine every minute for processing. As directed by policies created in the Halo Portal, the Halo agent can examine specific configuration settings or make changes such as updating firewall policies or user accounts.

  Because each server has its own agent, scaling up Halo protection when scaling up a server installation comes at essentially no performance cost to the customer. Each new server gets its own Halo agent, whose only task is to monitor that one server.

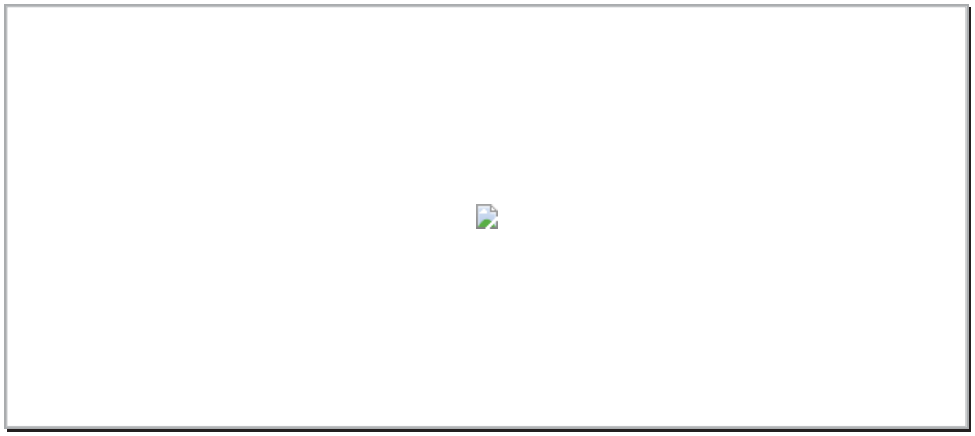
  Full-functionality and audit operational modes. The Halo agent normally has the ability to make changes to its host machine, as when updating host firewalls or managing user accounts. However, the agent can also run in audit mode, in situations where its ability to modify its host is not needed or desired.

- The **Halo security analytics engine** is a powerful, multiple-host elastic compute grid that is run and maintained by CloudPassage. The analytics engine performs sophisticated analytics that evaluate data collected by the Halo agents. The Halo analytics engine does the "heavy lifting" on behalf of the agents, conserving the servers' resources so they can continue to run applications with negligible impact from the agent.

  The engine is dynamically scalable and can manage hundreds of thousands of agents. As you increase the number of servers in your deployment, the Halo analytics engine bears the increased processing load and scales dynamically to meet the increasing demand. You do not need to do anything other than deploy the agents onto your additional servers.

  The engine is extremely hardened and highly secure. The only access to it is an HTTPS interface that requires authentication to initiate any operation. Furthermore, the analytics engine never initiates contact with any Halo agents; to protect customer security, the engine only responds to connections initiated by the agents.

- The **Halo portal** is the convenient "single pane of glass" used to manage all Halo product capabilities, create policies, set up alerting, view reports, manage users, and perform other tasks. Halo automatically monitors all servers and reports any security violations to the portal in near-real time, where you can view and respond to them.



The Halo Portal is accessible with recent versions of popular browsers and requires no additional client installation.

- The **Halo REST API** gives you an alternative to the Halo portal for managing Halo operations. Your developers can use it to automate Halo capabilities in new or existing management tools.

Halo scope

Halo can protect your infrastructure regardless of how the servers are hosted:

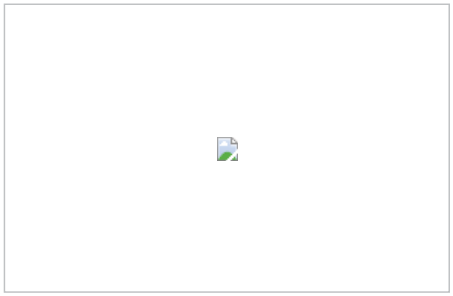
- In a *public cloud*. Your servers may be instantiated with any IaaS cloud provider, such as Amazon web services, Rackspace cloud, Microsoft Azure, Google Compute Engine, IBM SoftLayer, and others.
- In a *private cloud*. You can have you own private cloud service, implemented in your datacenter with open-source software such as OpenStack or cloudstack, or as a single-tenant service from a cloud provider.
- In a virtual or "bare-metal" *datacenter*. You can have either virtual servers, implemented with software such as VMware or Windows Server Hyper-V, or physical servers installed on hardware server machines.
- In a *multi-cloud* environment. You may use more than one public cloud provider, for example having servers in both AWS and Rackspace.
- As a *hybrid cloud*. For example, you may use a virtual or physical datacenter normally, but in times of high demand automatically spill over into a public cloud to handle the temporary need for additional servers.

In all of the above situations, Halo can automatically and securely provide the needed protection to every server. Note that Halo supports agents connecting through proxies and NAT devices. If your installation has a perimeter firewall, you can adjust that firewall to allow communication between the Halo analytics engine and your Halo agents.

Why are server groups important?

The concept of server groups is fundamental to Halo. Halo uses group-based policy management, meaning that an individual security policy is designed to apply to a group of cloned servers used for the same function.. There is no need to create an individual policy for each server. By applying policies in this manner, you can efficiently scale your protection to fleets of thousands of servers. And in the dynamic environment of the cloud, Halo instantly applies the proper policies to all new servers..

For example, your company may maintain multiple public-facing Web servers. Each serves up different content, but all run on the same operating system, run the same software and are exposed to the same level of risk from the Internet. Halo lets you leverage their commonality and manage them as a group—dramatically simplifying your administration effort, saving you time, and improving consistency across your cloud.



To come up with the best set of groups for your organization, examine all of the servers you currently use, and categorize them in terms of platform, applications, and purpose, while trying to end up with the smallest possible number of groups. The basic idea is that all the servers in a group usually need to be very similar (same O.S. and version, same applications, same firewall needs, same local user accounts), to allow a single set of policies to cover them all. For more detailed guidelines, see Organize Your Servers into Groups in the *Halo Operations Guide*.

Halo security policies: clone or create your own

Many of Halo's powerful capabilities are policy-driven, which makes them highly customizable. Halo comes with an extensive set of default policy templates that you can clone and use immediately, without alteration. However, with a small amount of extra effort, you should be able to add customizations to make your policies fit your specific environment even better. You can even create policies from scratch for maximum customization, by using the policy-creation pages in the Halo portal or by constructing them in the JSON format used by the Halo REST API.

The following Halo security modules include one or more default policy templates for your use:

- Configuration Security Monitoring (20 Linux policies, 7 Windows policies)
- File Integrity Monitoring (6 Linux policies, 9 Windows policies)
- Log-Based Intrusion Detection (2 Linux policies, 1 Windows policy)

CloudPassage believes that "defense in depth" is the best posture to adopt when applying security controls to your networked resources. That is why Halo includes such a wide offering of security modules and is continually adding new ones. You can use individual modules for targeted protection, or you can use several or all of them together, to achieve overlapping, "wrap-around" protection that surpasses the capabilities of "single point" solutions.

Where appropriate, Halo's modules are policy-driven, meaning that you can customize their functionality to better fit your own environment. And because policy application and management are group-based, you never need to design or apply protection to individual servers; you always work at the server-group level, which can greatly multiply your efficiency.

Each of the following sections explains an individual Halo security module.

Software Vulnerability Assessment

The exploitation of software vulnerabilities is a leading means of attack against networked servers, whether in or out of the cloud. For both compliance and general security reasons, organizations with networked software must ensure that all system and application components are protected from exploits that use known vulnerabilities. Patching those vulnerabilities can help you to avoid malicious exploits, remote buffer overflow attacks, denial-of-service attacks, and other security compromises.

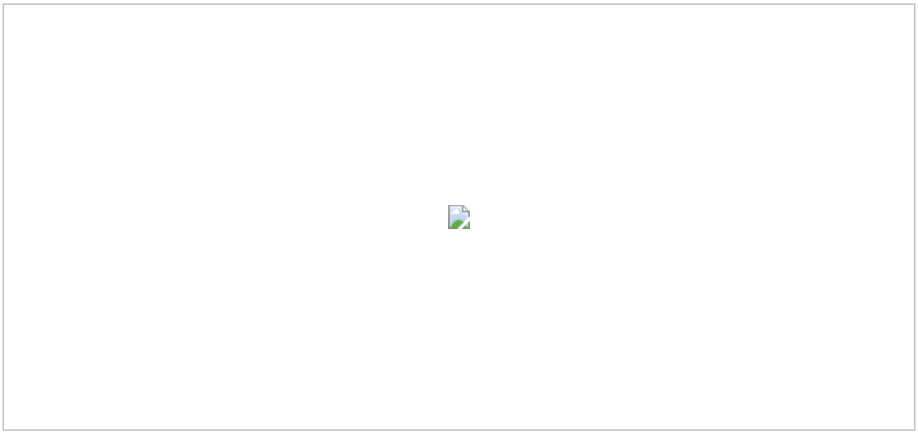
Defending against exploits. Defense against these attacks usually means installing the latest vendor-supplied security patches and upgrades. Scoring and ranking vulnerabilities for the risks they pose to your business is important to prioritizing your remediation efforts. Constant monitoring is also required, to ensure that new threats and

vulnerabilities are identified and addressed, and that software changes and upgrades are examined for vulnerabilities in a timely fashion.

Halo software vulnerability assessment is an important security module for both managing your organizations exposure to risk from compromise and for meeting compliance requirements. It regularly scans all of your protected servers to detect known vulnerable packages.

Note: Software Vulnerability Assessment is not policy-based and it is independent of server groups, because it does not need to be customized for specific environments. Automatic vulnerability scans apply to all of your servers that have installed, active Halo agents.

Vulnerability scans identify vulnerable software packages in your servers by comparing the versions of your servers' software packages (operating system, drivers, daemons, and applications) against the National Institute of Standards and Technology (NIST) database of Common Vulnerabilities and Exposures (CVE), in conjunction with other information. Each reported vulnerability has a score (assigned by NIST) according to the Common Vulnerability Scoring System (CVSS) and compared against a threshold value that you can set in the Halo Portal. Software packages with scores above your specified threshold are considered critical vulnerabilities and are flagged as such in reports.



In use, software vulnerability assessment functions this way:

1. The Halo analytics engine initiates a scan of each Halo-protected server, which causes each Halo agent to return a complete inventory of all installed software packagers to the analytics engine. (This inventory is valuable in itself, apart from security purposes.)
2. The analytics engine compares each package's name and version to the vulnerability data and records any packages that are known to be vulnerable.
3. To improve accuracy, all reported-vulnerable packages are further filtered by applying other third-party feeds and additional proprietary information generated by CloudPassage researchers.
4. The analytics engine sends the filtered list of vulnerable packages back to the portal, which displays them as scored security events.
5. A Halo user or security administrator remediates the vulnerabilities or schedules them for remediation through the organization's patch management program.

Remediating vulnerabilities. To remediate detected vulnerabilities, you can take any of these steps:

- Apply the latest patches that address the reported vulnerabilities.
- Remove unnecessary packages that contain vulnerabilities.
- Create exceptions for vulnerabilities that you will address in the near future or that do not pose an actual threat of exploit to your server installation.

Your goal in performing regular ongoing vulnerability scans should be to lower the number of reported vulnerabilities — especially critical vulnerabilities—to stay confident that your servers are well protected against software exploits.

For more details, see *Assessing Software Vulnerabilities with CloudPassage Halo*.

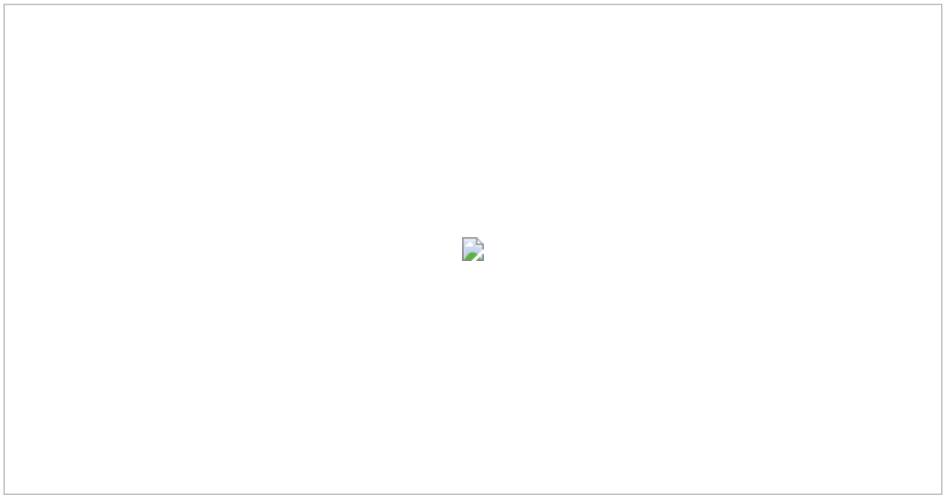
Configuration Security Monitoring

One of the most important steps you can take toward securing your cloud servers is to ensure that their operating systems and applications are properly hardened against attack. Maintaining attack-resistant software configurations

makes it much more difficult for intruders to gain a foothold on your systems.

The Configuration Security Monitoring module allows you to monitor the details of your configuration settings, system files, running processes, ownership, and permissions to ensure that there are no unauthorized or insecure values, files, processes, and so on that could compromise server security.

Halo regularly scans each server and applies a set of policy rules that specify what the secure configuration for that server should be. You can assign one of the Halo-provided configuration policies to each server group, or you can customize or build one from scratch to better fit it to each group's server configurations.



After a configuration scan completes, you can examine the results on a server's Scan Results page in the Halo portal.

You can also view the results as events on the Security Events History page in the portal. If you have set up alerting, you can view notifications in your email inbox alerting you to critical configuration-security issues that were detected.

Use that information to either (1) remediate detected issues by restoring the proper configuration settings to the affected servers, or (2) immediately notify your security team or incident response team, if an actual security breach is suspected.

For more information on setting up and using Configuration Security Monitoring, see *Monitoring Server Configuration Security with CloudPassage Halo*.

Note: If you need to customize or create your own configuration policy, you'll see that configuration policies are made of rules, and rules are made of configuration *checks*. Halo provides templates for creating over 40 different kinds of configuration checks. You can learn the details of how all configuration checks function by consulting the appendix Configuration Policy Rule Checks, in *Monitoring Server Configuration Security with CloudPassage Halo*.

Log-Based Intrusion Detection

The Halo Log-based Intrusion Detection system is a security module that monitors a server's log files for events that indicate compromise or misuse, alerting security personnel when such events are encountered.

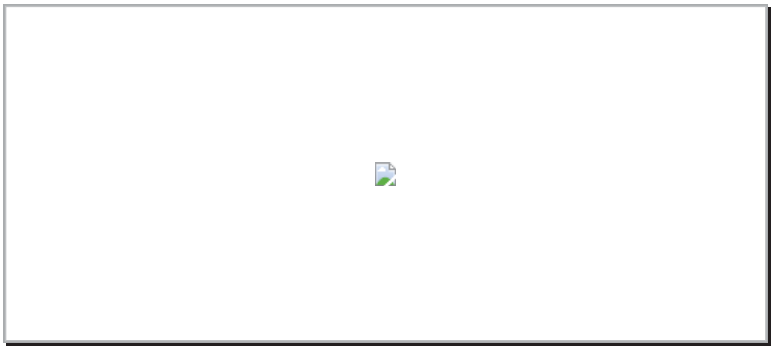
This module allows you to detect selected important events that may be recorded in any number of system or application log files on any of your servers. If you also enable Halo alerting, you can receive near-real-time alerts when the highest-priority events are logged.

The module uses Halo's scanning capabilities and policy architecture to detect and report on the most recent events of interest in near-real time after they are written to any of the log files you have specified. It allows you you to continually monitor the security of all of your server systems and applications, and be certain that you will be notified whenever

specific events of critical importance occur anywhere in your infrastructure.

In use, Halo Log-based Intrusion Detection continually scans all policy-specified log files, looking for recently logged suspicious events (also policy-specified). You can use built-in Halo policies or you can customize or create your own. Each policy consists of event-detection rules that can recognize specific event messages or IDs in specific log files:

All detected occurrences of those events are saved as Halo events, so that you can search for and view them in the Halo portal.



Events that are flagged to generate alerts will appear in your email inbox, so that you can act on them potentially much sooner.

To perform deeper analysis on any of these events, such as correlating them with other events across your installations, you may wish to integrate these Halo events into whatever log-management and analysis or SIEM solutions your organization uses.

For more information on Halo Log-Based Intrusion Detection, see *Using Log-Based Intrusion Detection with CloudPassage Halo*.

File integrity Monitoring

File Integrity Monitoring is a Halo security module that protects the integrity of your servers' system and application software. It regularly monitors them for unauthorized or malicious changes to important binaries and configuration files. Implementing file integrity monitoring helps you to detect potential intrusion and tampering, and it also helps you to comply with standards and mandates such as PCI DSS and SOX.

Baselines

Halo accomplishes file integrity monitoring by first saving a baseline record of the "clean" state of your server systems. It then periodically re-scans each server instance and compares the results to the appropriate baseline. Any differences detected are logged and reported to the appropriate administrators.

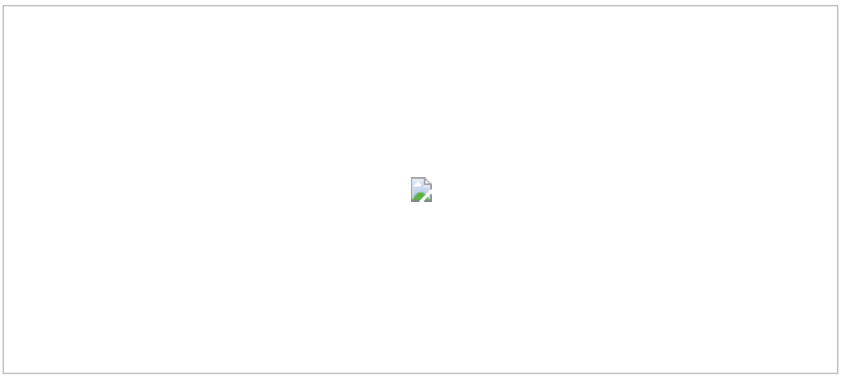
The elements that make up a baseline include (1) cryptographic checksums (signatures) and standard metadata (ownership and permissions) for all files being monitored, and (2) standard metadata for files without content, such as directories and symlinks.

If later scans reveal that a file's checksum or metadata has changed, a security event is generated. An administrator can inspect the metadata or the file itself on the server involved to understand the nature of the change and, if warranted, escalate the issue to an incident-response team.

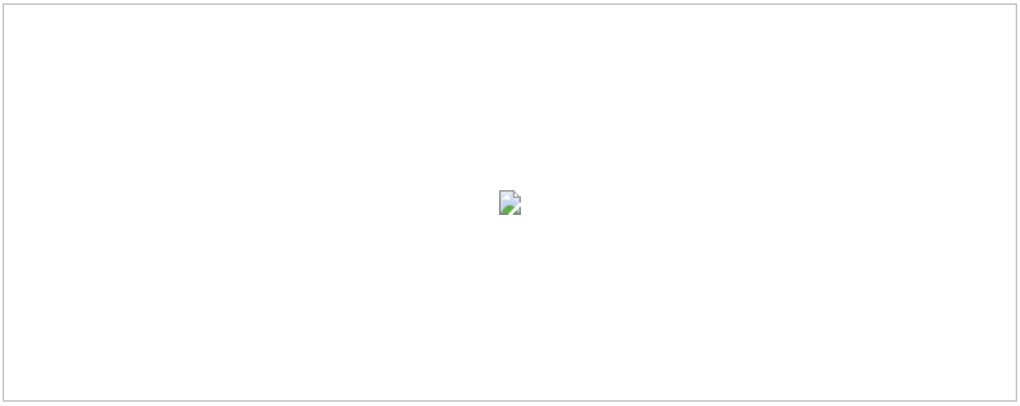
Scanning

You set up and run File Integrity Monitoring like this:

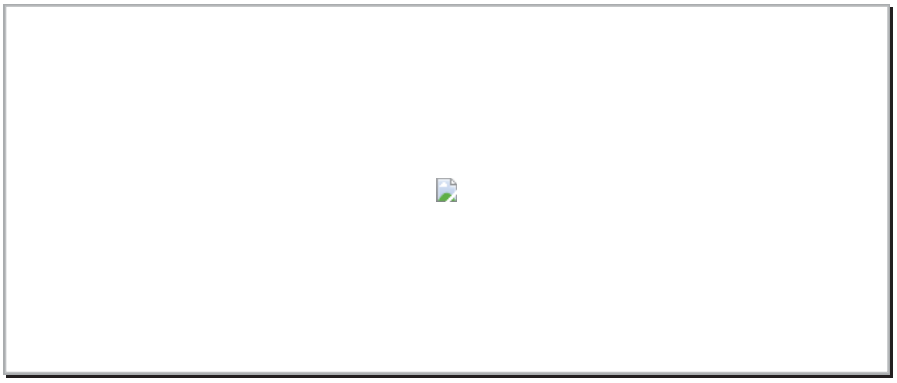
- Assign a file integrity policy to each server group you want scanned. The policy rules specify the files, directories, and software devices to be monitored.
- Assign a baseline to each server group by running a baseline scan on a secure, canonical, and clean configuration of that group's servers. For example, it might be the golden master template from which all of the group's servers are instantiated.



- Halo automatically runs regular monitoring scans on the group's servers. The Halo analytics engine compares the scan results with the baseline and reports any detected differences to the Halo portal. Modifications, deletions, or additions of files or directories, and any changes to the metadata, are detected.



After a file integrity scan completes, you can examine the results on a server's Scan Results page in the Halo portal.



If you have set up alerting, you can view notifications in your email inbox alerting you to any critical file integrity issues that were detected. You can also view the results as events on the Security Events History page in the portal.

Based on the scan results, either (1) modify your policy or re-run a baseline scan, for detected changes that were intentional and appropriate, or (2) immediately notify your security team or incident response team, if an actual security breach is suspected.

For more information on creating file integrity policies and using File Integrity Monitoring, see *Monitoring Server File Integrity with CloudPassage Halo*.

Workload Firewall Management

CloudPassage Halo automatically deploys, updates, and monitors host-based Windows or Linux firewalls for your cloud servers. Host-based firewalls can provide better protection for your cloud servers than traditional perimeter firewalls (which are for the most part lacking in public cloud environments), because they can be tailored to the exact purpose of

each type of server that you use. With Halo, you can design policies to facilitate inter-communication among the different categories of servers in your cloud, while simultaneously preventing malicious agents from gaining access.

Halo host firewalls also deploy themselves automatically and elastically, as your cloud-server population dynamically grows and shrinks. No servers are left uncovered and vulnerable to attack.

Halo firewall policies are also intelligent; they allow you to specify more than just IP addresses and ranges when defining the allowable sources or destinations of connections. For example:

- Because cloud providers typically assign arbitrary IP addresses to individual servers in the cloud, firewall implementation can involve tedious tracking of lists of server addresses. But with Halo these servers are in named server groups, so you can define high-level firewall policy rules using those group names as connection sources or destinations. Halo then uses those rules to create individual host-based firewall rules, taking care of tracking the IP addresses for you.
- To support GhostPorts multi-factor network authentication, Halo allows you to create firewall policy rules that specify usernames as sources of inbound connections. When such a user authenticates, Halo temporarily updates the appropriate firewall rule, using that user's IP address as the connection source and allowing access.

In use, Halo firewalls function like this:

- For each server group, you can use the Halo portal to create a firewall policy that is appropriate for protecting the servers in that group. The policy is a list of connection rules, controlling what specific kinds of inbound and outbound connections are permitted (or prohibited).

For an example set of firewall policies for a distributed application, see Example Firewalls: Multi-Server Web Application.

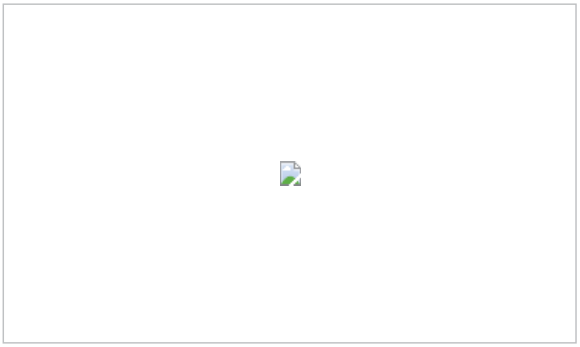
After that, Halo takes over—it installs individual Windows or iptables firewalls based on your policy on all of the servers in the policy's server group. Furthermore, Halo automatically updates all servers with any updates or changes you later make to that policy, or any changes to any of the servers' IP addresses. Halo also automatically deploys new firewalls to any servers that are added to the group in the future, such as through cloning or re-activation.

You can track the state of all your server firewalls at once from the Halo Portal. Update your firewall policies at anytime, and Halo will deploy the updates. Set up an alert so that you are notified if any firewall is tampered with.

For more information on Halo firewalls and how to create firewall policies, see *Managing Workload Firewalls with CloudPassage Halo*.

Multi-Factor Network Authentication (GhostPorts)

Halo Multi-Factor Network Authentication using GhostPorts is the most secure way to control access to services on your servers. It helps to eliminate the worry of attackers continually scanning your servers for open ports and attempting brute-force logins to those services. When Multi-Factor Network Authentication is enabled, the protected ports will be invisible to attackers when they scan your network. This makes it much harder for attackers to find a way in, because they can't even see your open ports.

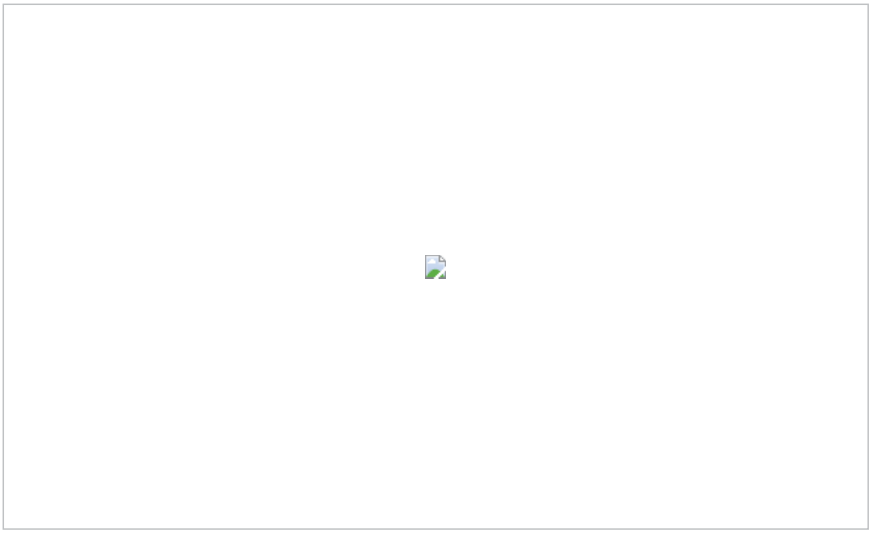


When an administrator authenticates to GhostPorts to gain secure access to a server, the administrative ports to the server are open only for a limited amount of time, and only from the administrator's current IP address. Potentially malicious users attempting to access the server at the same time are denied.

Multi-Factor Network Authentication requires both Halo login credentials and a second factor, involving a one-time password either transmitted by SMS text message or generated by a hardware device.

- For transmission by SMS, CloudPassage generates a one-time password and sends it to the GhostPorts user's mobile phone in a text message.
- For a hardware device, CloudPassage supports the YubiKey® from Yubico. A YubiKey is a one-time-password generator packaged as a USB input device. YubiKey values are unique across all of Halo, so each YubiKey can be assigned to only one GhostPorts user at a time.

Once a Halo Site Administrator has enabled GhostPorts for a user, authentication is a simple process. The user logs into the Halo Portal and authenticates to GhostPorts on the Open GhostPorts page, using either YubiKey or SMS code. In response, the Halo analytics engine temporarily opens the required ports on the required servers for access from that user's machine. The user then connects to the server outside of Halo, for example through SSH or RDP.



To enable this targeted access, you set up firewall policies that include rules for GhostPorts users. The rules determine the specific services and ports to be opened for each GhostPorts user's access.

Each time the user authenticates to GhostPorts, Halo communicates the user's source IP address to the firewalls of the servers in the target server group. The GhostPorts user now has access for a specific amount of time and only from that specific IP address. Once that time expires, Halo closes the open ports and further access is denied.

If you are a Halo site administrator, you follow these general steps to configure Multi-Factor Network Authentication for a Halo user:

1. Once your Halo firewalls are in place, set up GhostPorts users in the Halo portal. Create a new user or edit an existing user, then give the user GhostPorts access.
2. Either provide each user with a hardware token or have the user register their mobile phone number with Halo on their next login.
3. Create firewall rule(s) to enable secure access through GhostPorts.

See *Using Multifactor Network Authentication with CloudPassage Halo* for details.

Server Account Management

The Server Account Management security module allows you to monitor and audit remote access to your servers by all of the servers' local user accounts. Halo scans your servers at a frequency that you specify, gathering account information and login history for all servers, then displaying it in a centralized location where you can review and act on it.

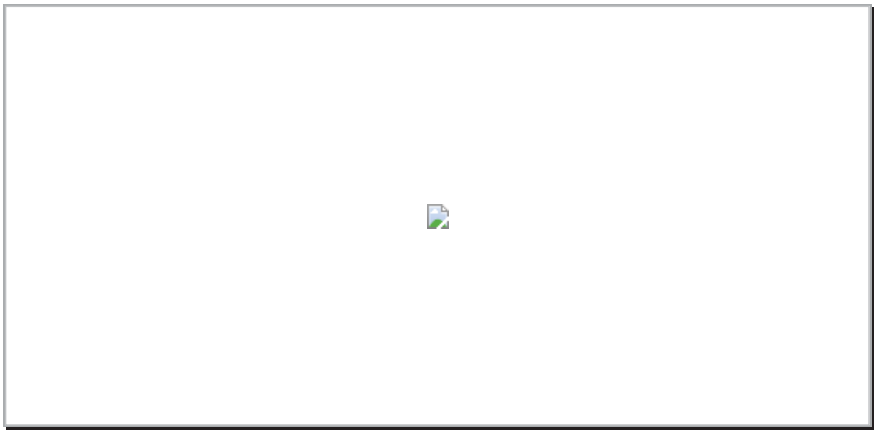
The module also provides basic account-management capabilities, allowing you to create, edit, or deactivate server accounts.

In an elastic cloud environment in which you may have hundreds of servers that come and go dynamically, using Halo for these purposes can save you time and also help to ensure complete coverage of your server installation.

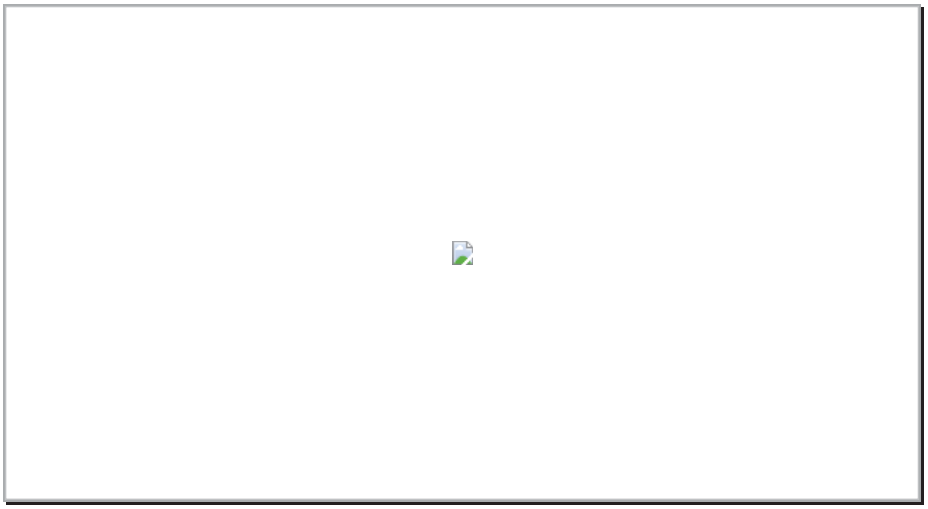
Note: Server Account Management is not policy-based and it is independent of server groups. Automatic server-access scans apply to all of your servers that have installed, active Halo agents.

Auditing Server Access

Halo regularly performs server access scans and presents detailed results in the Halo portal.



The results identify all local accounts on each server, noting each account's access privileges and recent login activity. From this information you can easily identify unexpected accounts, inappropriate privileges, and suspicious login activity on any of your servers.



Administering Server Accounts

Halo also lets you control server accounts centrally from the Portal, so that you can remediate account issues uncovered by server-access scanning. For example, you can edit an account to change its permissions, you can deactivate an account that should not exist on a particular server, and you can even create a new server account.

The Halo Portal provides you with the convenience of manipulating all server accounts through one user interface. And for even greater convenience, you can use the Halo API to script automated actions across multiple accounts at once.

See *Managing Server Accounts with CloudPassage Halo* for details.

Halo Platform Services

To help its large number of security modules to perform their tasks, Halo also includes a number of cross-module platform services that perform support functions for the modules and for Halo users. The services that have the most direct impact on Halo users are described here.

Halo Portal

All Halo users automatically have access to the Halo portal. The Portal is a web application that is hosted in the Halo cloud and can be accessed from anywhere using only a standard web browser.

The portal gives Halo users an instant view of the current security posture of their organization's server hosts—across all of the organization's cloud and non-cloud environments—from a single central location. The portal's browsing, searching, and reporting capabilities allow the user to drill into the details of any security issue to research it and, if

necessary, remediate it or report it.

You use the portal to organize and manage servers, to define and apply security policies, and to automatically or manually run security scans. Through these scans, Halo regularly monitors all servers and displays any security violations in the portal in near-real time, where you can view and respond to them. You can also use the portal to view historical scan data, and to manage user accounts and account settings across all your servers.

Halo users also access the portal to respond to security alerts and to manage their own Halo accounts. Depending on the user type (see Halo Accounts, Users, and Roles), a Halo user may also be able to use the Portal to install Halo agents, add and manage new Halo users, and perform a variety of other administrative task, including creating API keys and setting up automatic scanning schedules.

The Halo portal is highly flexible in terms of its login requirements. If you are a site administrator, you can configure it for either password authentication or multi-factor authentication, and you can customize its password requirements. You can add IP address restrictions or browser authorization requirements for login. You can also integrate portal login with your organization's single sign-on solution.

For more information, see Using the Halo Portal in the *Halo Operations Guide*.

Halo Logging and Alerting

Logging and alerting is a built-in service that captures event information generated by all of the Halo security modules, by Halo user actions, and by Halo-protected server actions. Halo stores these events centrally, and reports on them in a variety of forms, including summaries and details displayed in the Halo portal, email alerts sent to administrator inboxes, and event data exported to third-party analytical tools.

- "Logging" refers to the recording of events—such as policy violations detected during scans, or user actions such as logins—that Halo has been configured to record. Not all policy violations or user actions are necessarily logged as events.
- "Alerting" refers to the subset of events for which Halo has been configured to generate email notifications and send them to appropriate personnel. Typically only the most critical events are set up to generate alerts.

Controlling what is logged

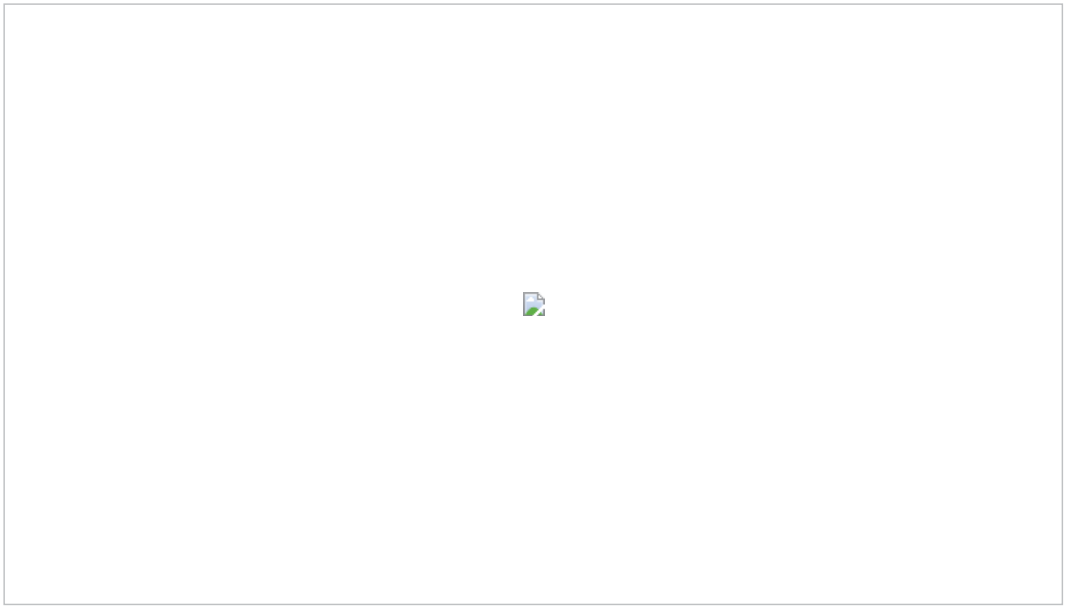
Logging and alerting is always "on" and available to all Halo users, but you decide what and how much you want logged, and who should be notified. For example:

- Most of the policy-based modules allow you to separately turn logging or alerting on or off for each rule in a policy, to flag the more serious ones as Critical, and to generate email alerts for the most serious of them.

- You can implement a *special events policy* to control the logging and alerting of server-related events across your infrastructure.
- You can control which routine *audit events* (logins, policy assignments, password changes, and so on) should be logged or alerted on.
- You can create *alert profiles*, which control who should receive email alerts for various events.

Viewing events

You can view events and alerts in several ways. You can view alerts in your email inbox, you can view event summaries on the Halo portal dashboard page, you can view event lists and event details on the portal's Server Security Events page and Server Scan Details page, and you can search for and view events on the Security Events History page.

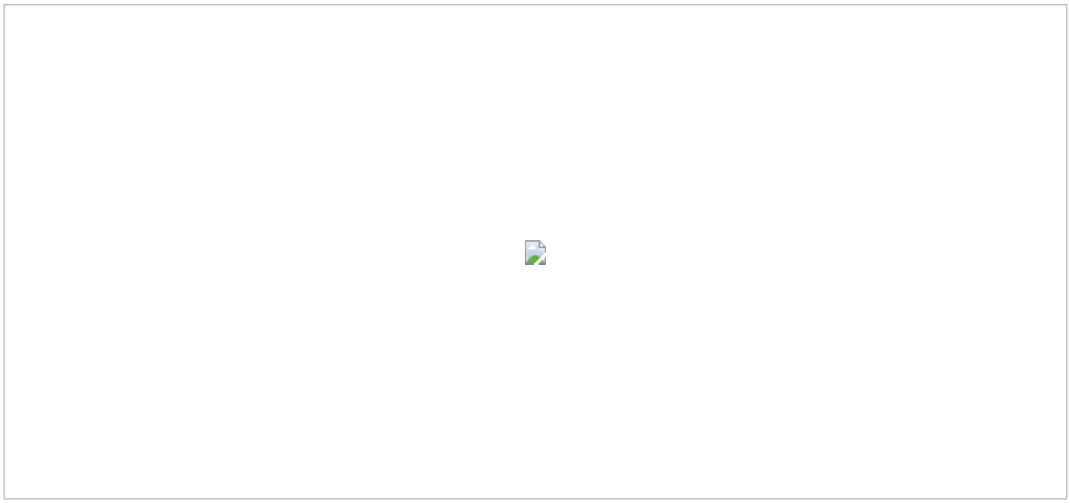


For more information on managing events and alerts, see PART 2: Halo Issues and Events in the *Halo Operations Guide*.

Integrating Events with Analytical Tools

The Halo API (see Halo REST API) includes an "Events" endpoint that clients can query to obtain complete or filtered information on all Halo security events (for example, detected server configuration errors or file-tampering indicators) within a range of date-times that you specify. CloudPassage has used this capability to create an integration tool that feeds event data to a variety of third-party tools for analysis.

CloudPassage has made the Halo Event Connector available to customers. The tool provides direct integration with Splunk Enterprise and SumoLogic, and integration through syslog to ArcSight and other tools. For more information, see https://github.com/cloudpassage/halo-event-connector-python.

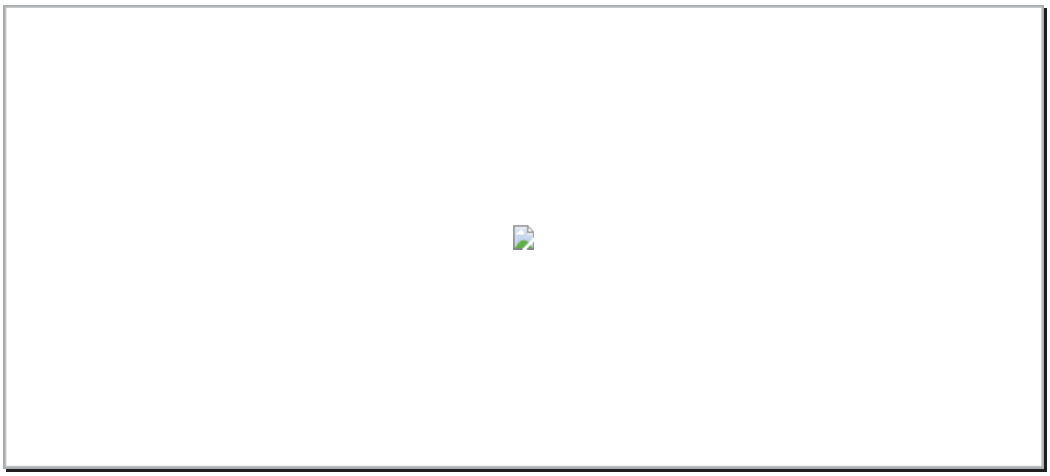


Halo Reporting

The reporting service of the Halo portal allows you to perform detailed parametric searches of the Halo database to locate items or sets of items that you may wish to act on. The current release of the reporting service focuses on searching for individual servers or collections of servers that match any of a large number of criteria.

On the Reports page in the Halo portal, you use the search-criteria selectors to set up and run a simple or complex search query. You can search for servers using any combination of over 20 criteria. The results are displayed as a

table, and you can save them in PDF or CSV format.



Examples of searches you might conduct:

- List all servers that are no longer sending heartbeats to the analytics engine.
- List all servers whose Halo agent needs to be upgraded.
- List all agents that may be compromised.
- List all servers on which a package containing a specific common vulnerability and exposure (CVE) is present.
- List all Windows Servers that have been patched to comply with a specific Windows knowledge base article.

Any of the above searches could be further filtered to restrict the scope of the results—such as to a specific O.S. version number, specified set of server groups, specific Halo agent state, and so on.

Note: All Halo reporting functionality is also available through the Halo REST API, in methods of the "Servers" and "Saved Searches" API endpoints.

For more details on the Halo reporting service, see Using Halo Reports in the *Halo Operations Guide*.

Halo REST API

The Halo application programming interface (API) is a representational state transfer (REST) interface that offers a secure, authenticated way for programs to directly access Halo functionality. Your client software can automatically perform many of the same functions that Halo portal users perform manually, such as creating and managing policies, creating or deleting server groups, and running scans.

About the API

The API accepts and returns stored Halo resources that you access through URL paths. To make an API call, your application submits an HTTP request, then parses the response. The request and response data are both in Javascript Object Notation (JSON) format.

All access to the Halo API requires authentication. First, your application or script client must authenticate with Halo to request an access token for the session. Your client then submits that token with every API call that it makes.

The API allows you to automate many aspects of Halo functionality, by manipulating the following resources through API calls:

Users *[Halo users]*	Server Groups	Servers	Server Accounts *[local users]*
Server Commands	Server Processes	Scan Scans	Scan History
Configuration Policies	File Integrity Policies	File Integrity Baselines	CVE Exceptions
Firewall Policies	Firewall Rules	Firewall Interfaces	Firewall Services
Firewall Zones	Log-based IDS policies	Special Events Policies	Events
Alert Profiles	Saved Searches	System Announcements	

The Halo API is fully documented in the *Halo REST API Developer Guide*. Related blog posts are also available at the Cloud Security Blog.

Using the API

CloudPassage customers and employees have used the API to construct server-security management tools and to integrate Halo with other systems. The Halo Toolbox is a set of GitHub repositories where CloudPassage customers and employees can share and compare code that automates tasks by calling the Halo API. The Toolbox facilitates collaboration and sharing of code.

As a simple example of the kinds of automation and integration solutions that have been developed, the following code calls the API's "List servers" method (documented as `GET https://api.cloudpassage.com/v1/servers` in the *API Developer Guide*) to retrieve information for all active Halo-protected servers, and then prints out a list of the servers' names before closing the connection.

A Python script making the call and acting on the results might look like this:

```
 tokenheader = {"Authorization": 'Bearer ' + key}
connection.request("GET", "/v1/servers", '', tokenheader)
response = connection.getresponse()
jsondata = response.read().decode()
data = json.loads(jsondata)
# iterate through json result and print out hostnames
servers = data['servers']
for server in servers:
        print server['hostname']

connection.close()
```

...or a Ruby script might look like this:

```
 result = RestClient.get "https://#{host}/v1/servers", {
        'Authorization' => "Bearer #{token}"
}

data = JSON result.body
servers = data['servers']
servers.each do |server|
        puts server['connecting_ip_address'] + " " + server['hostname']

end
```

Another widely used application of the Halo API is to export events from the Halo logging and alerting module for integration with log-analysis or SIEM applications, as mentioned in the previous section (Halo Logging and Alerting).

To examine the complete source code of these and other Halo API examples in the Toolbox, go to https://github.com/cloudpassage/api_examples.

Integrating Halo into Server Orchestration

Halo's capabilities as a highly automated cloud security platform are further augmented by its ability to integrate with orchestration tools and best practices for cloud installations.

Deploy Agents in Bulk With Automation Tools and Scripts

You can integrate Halo with cloud management and IT automation tools—such as RightScale, Puppet Labs Puppet, and OpsCode Chef—to transparently embed Halo security into an automated server provisioning process. For example:

- Puppet is a well-known tool that you can use to provision Halo across multiple servers. CloudPassage has made an example Puppet module available to customers. It uses a standalone Puppet deployment in which both the master and agent are running on the same server. You may wish to use our Puppet example as a starting point for developing your own automated Halo provisioning method. See this Blog post for more details.
- CloudPassage has also prepared a pair of Chef cookbooks—one for Linux servers and one for Windows servers—containing recipes for installing Halo Demons on your servers. You would add the appropriate cookbook to your Chef run list, and then execute the run list on a set of servers. See this Blog post for more details.
- CloudPassage has also created integration scripts with RightScale to automate the installation of Halo agents in a RightScale-managed environment. The RightScript works with any type of RightScale account, and can be run as

either a boot script or an operational script. See this forum post on the CloudPassage Support site for more details.

Install configured Agents on Your Gold Master Servers

If you use "gold master" images of your servers as templates from which to create cloud instances, upgrading and patching can become highly automated. To modify the software of a group of servers, you explicitly change just the gold master, update policies and baselines to match it, and then reinstantiate all servers from the updated template.

For even more automation, you may want to install Halo agents on the gold masters. Then, when you create server instances, each will already have an installed agent.

We recommend that you start the Halo agent service after installing, by leaving the **Start CloudPassage Halo Agent Now** checkbox selected. Doing that will ensure that any cloud instances created from the gold master will have unique Halo IDs and will receive all updated Halo policies.

Automatically add server tags

You can automate the assignment of newly instantiated servers to the right server groups by using Halo server tags. First, identify each server group with a unique alphanumeric string. Then modify your orchestration scripts to add the proper tag parameter to the startup command for each newly instantiated Halo agent.

Automatically add server labels

You can optionally define more user-friendly, explanatory labels for your Halo-protected servers, as alternatives to the host names and fully qualified domain names automatically assigned. The label will be displayed everywhere in the Halo portal UI, in place of the hostname or FQDN.

You don't specify a server label from within the Halo portal or through the Halo API; instead, you add a parameter to the Halo agent's startup command. To automate the assignment of labels, modify your orchestration scripts to include the desired label in the server-startup instructions for each newly instantiated server.

Halo Accounts, Users, and Roles 

Customers interact with Halo in a variety of ways, depending on the types of their Halo accounts, the types of Halo users they are, and the overall roles they play in their organizations. The following lists summarize the dependencies.

Types of Halo accounts:

- **Customer account**. This is the kind of account that is created when you register with CloudPassage. The majority of CloudPassage Halo accounts are customer accounts.
- **Master account**. A master account is a Halo account whose users have visibility into and control over certain other Halo accounts (its *subaccounts*). Typically, a large organization using Halo will register one master account, then each business unit, division, or subsidiary that uses Halo will register a customer account and then tie it to the master account—which makes it a subaccount of the master.
- **Subaccount**. A subaccount is a customer account that is associated with a master account. It has exactly the same capabilities and privileges as other customer accounts, except that it is tied to a master account—which means that the master account user can monitor activity on the subaccount.

If you are interested in setting up a master account, contact your CloudPassage representative or Customer Support.

Types of Halo users:

- **Standard user**. A standard user is a Halo user that has been invited by a Halo site administrator to join Halo as a standard user on the site administrator's customer account.

  Standard Halo users have access to the Halo portal and can create policies, view scan results, and use the Halo REST API. Standard users are not able to create Halo users or perform any other site administration tasks, including configuring Halo modules or generating API keys.

- **Site administrator**. A site administrator is a Halo user that has either (1)registered a customer account with

CloudPassage, or (2) been invited by a Halo site administrator to join Halo as a site administrator on that account.

Halo site administrators have the capabilities and privileges of standard users, plus the ability to install Halo agents and perform all configuration tasks available on the portal's Site Administration page, including the ability to create and manage Halo users and the ability to generate API keys.

Typical roles of Halo users:

- **System administrator / operations analyst**. A person in this role typically installs (or automates the installation of) Halo agents on servers. May also be involved with the design and implementation of Halo server groups. May be responsible for quarantining compromised server hosts.
- **Security ops analyst / security analyst**. A person in this role typically monitors a set of server groups for signs of compromise. May also be involved in designing and applying Halo security policies, as well as generating audit reports for compliance purposes and participating in forensic analyses.
- **Server administrator**. A person in this role needs access to one or more servers for administrative or other purposes. May or may not have access to the Halo portal, other than for using GhostPorts multi-factor network authentication to open the server ports. A Halo site administrator sets up this kind of access for every server admin.
- **Security architect**. A person in this role typically designs the organization's overall security program and specifies how Halo will contribute to it. May choose which Halo modules to implement, may select, design, or approve the policies for each module.
- **Emergency responder / forensic analyst**. A person in this role is aware of the latest threats and incidents, and is responsible for ensuring that a potentially compromised host is quarantined, for conducting or coordinating investigations into the breach, for recovering from it, and for serving as a point contact for either receiving or disseminating important information regarding it. May use Halo to inspect pertinent events or to export events for further analysis.

Learn More About Halo

CloudPassage provides a wide range of learning tools to help you to better understand and exploit the power of Halo. All of the Halo product documentation is available online:

- For step-by-step instructions on getting started using Halo, see the Halo QuickStart.
- For detailed instructions on all aspects of using Halo, see the *Halo Operations Guide* and other manuals in the Halo documentation forums on the CloudPassage Support site, at https://support.cloudpassage.com.
- For video introductions to specific Halo features, and for FAQs, Tips and Tricks, see the Halo Knowledge Base, also on the CloudPassage Support site.
- For technical Blog posts regarding Halo, see the CloudPassage Cloud Security Blog.
- For complete documentation of the Halo REST API, see the *Halo REST API Developer Guide*. For descriptions and code examples of automation techniques that you can apply to Halo, see the Halo Toolbox on GitHub.