Watchmaker

Applied Configuration Management

2016.06.14

Plus3 IT Systems
12030 Sunrise Valley Drive
Suite 110
Reston, VA 20191

PLUS 3
IT SYSTEMS

Executive Overview

The Situation

Creating and managing custom system images is a complex task. A secure configuration must be enforced, patches must be applied, and agents must be installed. Prior to creating the image, the source system must be specially modified or configured so that systems deployed from the image have wholly-unique identities (such as SSH fingerprints). Some software agents use mutually-authenticated certificates to encrypt communications with a central server. Pre-installation of agents frequently breaks authentication that depends on unique client certificates. Static system images tend to be of sizes that are difficult to transfer to isolated networks and to remote sites (data centers or branch offices). Many organizations employ manual processes for one or more of the image creation steps. Manual processes are inherently prone to introducing errors and inconsistencies into builds. This creates unintended differences between build-generations. These differences cause delays due to manual-remediation of deltas and the need to solved bugs caused by unanticipated deltas.

Custom images are intended to make it easier and faster to deploy systems that are identical and compliant with organizational and application requirements. And they do! At first. Unfortunately, the time and overhead of maintaining a set of images makes sustainment difficult. Frequently, it requires a staff of build-maintainers to keep the workload from becoming overwhelming. This can unsustainably increase the costs of maintaining the imaging software and processes. Adding insult to injury, it necessitates a slow release-tempo that may not keep up with an organization's or application's needs. Application owners, who are attempting to target an application build against a known baseline, often get frustrated with small build differences between one site and another. Undocumented changes between one version of an image and the next is another source of frustration. Worst is when the build process itself contains manual steps following initial provisioning, leading to further small changes from build to build, even when the underlying image is kept identical.

The Solution

Watchmaker[1] automates the process of configuring a system at provisioning time. This eliminates the need for customized images. Rather than creating custom, monolithic images over and over again, Watchmaker can be used with vanilla images straight from the vendor(s). Watchmaker is a self-contained and easily-configured Python package that uses simple configuration file formats. Watchmaker manages the installation of a configuration management solution and applies specific configuration states. Further, Watchmaker supports re-execution, and can be used to re-apply the configuration baseline to the system over time. Organizations that use this tool can greatly reduce the frustrations and shortcomings associated with maintaining a variety of custom images. This allows

[1] https://github.com/plus3it/watchmaker

organizations to greatly reduce time, costs and overhead for provisioning systems across sites while retaining the required level of business-driven customizations.

The Methodology

Watchmaker is named in homage to the intricate detail and precision that watchmakers endured during the early years of watchmaking. An analogy is drawn between consistently-made, precise, well-timed watches and consistent complex system configurations and hardenings.

Watchmaker is written with extensibility and portability as primary objectives for the package. The tool exercises the following stages when configuring a system:

- System preparation
- Security hardening
- Common configuration
- Organization configuration
- Application configuration

System Preparation (Bootstrap)

The system preparation stage uses a bootstrapping process to begin configuration of an arbitrary system image. Bootstrappers can be executed during the initial system boot or anytime after the system is running. The bootstrappers initialize the system environment to meet the dependencies required to install and run Watchmaker, including tasks such as installing internal Root CA certificates, downloading and installing Python, and installing Watchmaker itself.

Once Watchmaker is installed, the bootstrapper executes the Watchmaker application. As part of the preparation stage, Watchmaker installs and initializes a Configuration Management (CM) tool to execute the remaining stages. Watchmaker is modular and supports plugins to allow Watchmaker to work with many CM tools. Included with Watchmaker is support for Salt and a number of Salt Formulas. Salt is the CM tool; Salt Formulas are the instructions, or the set of states, that will be applied to configure the system. A user can easily extend Watchmaker to support other CM tools, such as Puppet, Chef, or Ansible.

Security Hardening

When Watchmaker executes, the main step is to apply a security-hardened baseline. The primary security baseline is the set of lockdowns prescribed by the DISA STIG[2]. Many items in the DISA STIG are

[2] http://iase.disa.mil/stigs/Pages/index.aspx

either non-technical controls or are simply guidance on how to run a secure system or environment. Watchmaker relies on SCAP content[3] - derived from the DISA STIGs - to identify guidance that can be applied through system configuration settings. The relevant lockdowns prescribed by the STIG are quite comprehensive and form a good basis to minimize the exposed attack surface of the system. Organizations or application-owners may choose, in following stages, to relax specific settings as needed.

Common Configuration

Common configuration items will be applied to the new system if no errors occurred in the previous phase of Watchmaker. "Common" is intended to mean configuration states that apply to all systems regardless of organization or application. Common configuration states that are managed at this stage include:

- Setting the hostname
- Configuring the time service
- … others as needed

Organization Configuration

Organizationally-defined configuration items will be applied after error-free completion of the common configuration step. This stage applies configuration items that are mandated by the owning organization. This phase allows an organization to ensure systems are always built in a way that will always pass the organization's requirements. Typical configuration states in this stage include:

- Joining a domain
- Installing agents, such as auditing or antivirus solutions
- Configuring update repositories or settings

Application Configuration

After organization-defined configurations have been applied, then the final phase will apply application-specific configurations. These are configurations defined by the application team that will be responsible for the operation of the system. The user of Watchmaker defines a configuration file containing specifications used to initialize a Configuration Management utility with the desired application states. Watchmaker will then apply the defined states.

[3] http://iase.disa.mil/stigs/scap/Pages/index.aspx

Conclusion

Plus3 IT saw an opportunity to solve the difficult problem of originating and maintaining customized system images. In Watchmaker, Plus3 IT's automation engineers built a flexible, layer-oriented approach to solving the problem. Rather than being reliant on a fixed starting state, Watchmaker focuses on achieving a desired end-state. This approach allows an organization's application owner to ensure compliance with organizational mandates as well as create consistent platforms on which to run applications. Watchmaker's automation also allows organizations to delegate the provisioning activity to application owners rather than relying on an intermediary team of build-specialists. This allows the organization to be flatter and to increase release tempos without sacrificing organizational standards.