Abstract

Come up with a sane process for deploying applications while achieving the separation of systems. So when one target host that is being deployed to is compromised it does not expose artifacts of other target hosts in the Jenkins ecosystem. Skip the definitions section and refer to it if you encounter an italics word that you don't understand.

Definitions

Jenkins ecosystem – The Jenkins ecosystem consists of at least 1 master node (the scheduler for scheduling jobs on build nodes) and 1 or more build nodes dedicated to building jobs. This is the whole system in which building jobs and deployment of jobs will be launched. Sometimes the Jenkins ecosystem will simply be referred to as "Jenkins". Hosts external to the Jenkins ecosystem can only communicate to the ecosystem through a single point, the master node.

Plug-in – Jenkins is a pluggable software which can be extended via plugins.

Target host – The application server where the artifact will be deployed. This hosts the public facing end user application.

Job – Lives on the Jenkins server. Is a set of steps to test, build, and/or deploy applications. It can be all in one job or split across multiple jobs. Jobs run on build nodes and the artifacts created from the job can be archived on the master node for later being retrieved by target hosts.

Artifacts – Binary blobs that are the result of a build job. In this case the java war file is an artifact.

Jenkins API – Is a generic term for the different ways the API can be accessed on the system. The Jobs have an API, the individual builds of the Jobs also have an API.

deploy_user – This user hypothetically exists only in Jenkins. It is a user which is hypothetically given read only access to jobs for accessing the API.

Authenticated zone – This is simply a sub-URL which is restricted to authorized users only on the Jenkins site. It requires user authentication to access. In this case the authenticated zone is https://<jenkins-master-node>/jenkins/jobs where "/jobs" and path children are part of the authenticated zone. User authentication is configured from Global Security [1] in Jenkins. Anonymous users lack overall read permissions.

Unauthenticated zone – Opposite of Authenticated zone. Does not require user credentials configured from Global Security. This can potentially require other forms of authentication.

Token – An arbitrary string that can be set to restrict access to the URL. Usually supplied as a GET argument e.g. "?token=mysecrettoken".

Overview of current Jenkins capabilities

The *target hosts* (Host A and Host B) are application servers. They will be where the actual application is deployed that will be used by end users. The deployment script will exist on Host A and Host B will talk to deployment jobs existing on *Jenkins* over HTTPS. The goal is to retrieve the *artifacts* (java war file) to be deployed to the application server from Jenkins over HTTPS. It will use the *Jenkins API* to determine the download location of the *artifact* to deploy and retrieve it from *Jenkins* to the target host. See Figure 1.

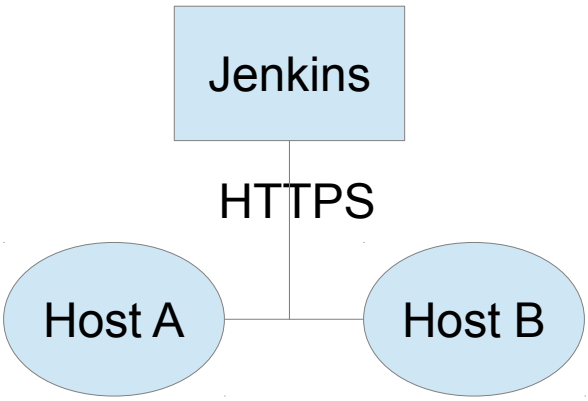

Figure 1) Communication diagram of *target hosts* talking to *Jenkins* via *API* to retrieve *artifact*.

Jenkins has an *authenticated zone* when global security is configured. The *job artifacts* and the *Jenkins API* for each job lie within this authenticated zone. In order to gain access to each deployment job the *deploy_user* must be given read access to every *job* in which it needs to interact with the *Jenkins API*. See Figure 2.

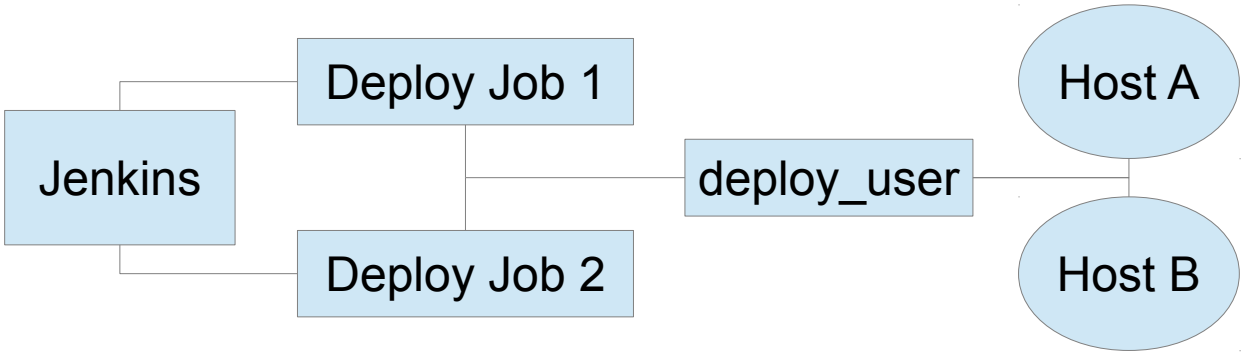

Figure 2) Diagram of showing *target hosts* communicating to *Jenkins* through a single user.

As you may infer, Figure 2 is a bad security model because if any *target host* in the figure (Host A or Host B) is compromised the deployment *artifacts* of all systems will be exposed.

Proposal for new plug-in

So far I have found a neat plug-in called the Build Token Root Plugin [2]. This creates an *unauthenticated zone* at the location "jenkins/buildByToken/" where anonymous access is allowed and builds are launched via a *token* which can be unique on a job to job basis.

I want that ability of the "Build Token Root Plugin" but for readonly API and readonly download artifacts for deployment. Ideally, the anonymous read token would be a separate setting from the build token setting in the *Job* configuration. I shall call this "anonymous read token" setting **anonymous_token**.

Access to the *unauthenticated zone* **shall be denied** if:

- No *anonymous_token* is provided.
- The *job* being called has no *anonymous_token* setting enabled and configured.
- The *anonymous_token* provided does not match the *anonymous_token* configured in the *job*.

See Figure 3 for what the new deployment model would look like.

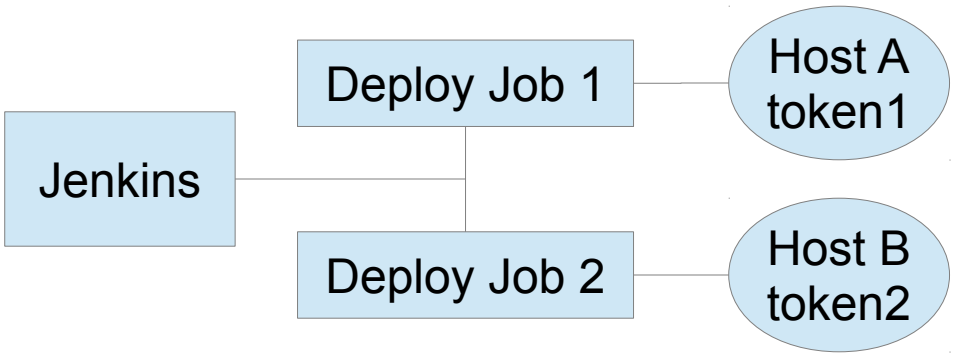

Figure 3) Each *target host* has a different *anonymous_token* configured on a job to job basis.

Since each target host is authenticating using a different *anonymous_token* if one target host is compromised then only the API and artifacts available to that specific host are able to be read. In the situation of a compromised host the attacker already has access to that information the target host is allowed to deploy (i.e. the deployed *artifact*).

This allows separation of systems in the deployment ecosystem because one compromised host does not expose the artifacts of unrelated deployments.

Summary of Plug-in specifications

Plugin will create an *unauthenticated zone* to bypass the *authenticated zones*. Access will be granted through a *token*. The *token* shall be called "*anonymous_token"* and be configurable in the job configuration page. The *anonymous_token* setting should behave exactly like the "Build token" in the *job* configuration. The *anonymous_token* will be used as authentication for accessing the *Jobs API* and *artifacts* associated with *jobs* for anonymous user deployment scripts which will live on the *target hosts*. The access given to both the *job API* and the *job artifacts* by the plug-in should be readonly and not allowed to update configurations or states in any way.

Access to the *unauthenticated zone* **shall be denied** if:

- No *anonymous_token* is provided.
- The *job* being called has no *anonymous_token* setting enabled and configured.
- The *anonymous_token* provided does not match the *anonymous_token* configured in the *job*.

Links

1. https://wiki.jenkins-ci.org/display/JENKINS/Standard+Security+Setup
2. https://wiki.jenkins-ci.org/display/JENKINS/Build+Token+Root+Plugin