Decommissioning a Server

Possible Jenkins Project Infrastructure Compromise

What are Elevated Privileges? Elevated privileges are **when a user is granted the ability to do more than a standard user**. A standard user is someone that has "zero administrative" privileges in any capacity.

Scenario

Last week, the infrastructure team identified the potential compromise of **a key infrastructure machine.**

This compromise could have taken advantage of what could be categorized as, an attempt to target **contributors with elevated access**.

Unfortunately, when facing the uncertainty of a *potential* compromise, the safest option is to treat it as if it were an actual incident and react accordingly.

The machine in question had access to binaries published to our primary and secondary mirrors, and to contributor account information.

Since this machine is not the source of truth for Jenkins binaries, we verified that the files distributed to Jenkins users: plugins, packages, etc, were not tampered with.

Action

We cannot, however, verify that contributor account information was not accessed or tampered with and, as a proactive measure, we are issuing a password reset for all contributor accounts.

We have also spent significant effort migrating all key services off the potentially compromised machine to (virtual) hardware so the machine can be decommissioned entirely.

Notes

A binary artifact repository stores binary artifacts along with the metadata in a defined directory structure, conceptually like a source code repository. The metadata describes the binary software artifact and includes information such as dependencies, versioning, and build promotions.

However as important as coding is, **source code is not what your customers use.**

Customers are going to run an application, which is a composite of many binaries from many source codes.

The quality of the software you deliver to your customers depends on the quality of the binaries that go into it. (This why we use SonarQube to ensure the quality of the software).

Do not forget that SonarQube does not test the source codes, but the many binaries that originate from the artifacts build by the Maven.

Solutions

What we are doing to prevent events like this in the future

As stated above, the potentially compromised machine is being removed from our infrastructure.

That helps address the immediate problem but does not put guarantees in place for the future.

To help prevent potential issues in the future we are taking the following actions:

1. **Incorporating more security policy enforcement** into our ansible driven infrastructure.
   Without a configuration management tool enforcing a given state for some legacy services, user error and manual misconfigurations can adversely affect project security.
   As of right now, all key services are managed by ansible.
2. **Balkanizing our machine and permissions model** more. The machine affected was literally the first independent (outside of Sun) piece of project

infrastructure and like many legacy systems, it grew to host a multitude of services.
We are rapidly evolving away from that model with increasing levels of user and host separation for project services.

3. In a similar vein, we have also introduced a trusted zone in our infrastructure which is not routable on the public internet, where sensitive operations, such as generating update center information, can be managed, and secured more effectively.
4. **We are performing an infrastructure permissions audit**. Some portions of our infrastructure are 6+ years old and have had contributors come and go. Any inactive users with unnecessarily elevated permissions in the project infrastructure will have those permissions revoked.

We need to communicate with the teams and the contributors connected to the server.

If it is in production, we need to create a ticket, plan, create a change request and create a decommissioning schedule, get the approval from CAB, and Plan on when to decommission the server.

1. Identify and Record

This server is going to be decommissioned.

We are going to record all necessary information and schedule the decommission.

In your timeline, be sure to account for if (and when) unexpected issues arise during the process.

2. Create a Log

Develop a comprehensive log of all actions performed during server decommissioning.

It is important that every step and compliance has been well-documented in preparation for potential audits.

Be sure to include the certificate of erasure/destruction you receive during step five or eight.

3. Locate Licenses

Locate and retain any and all software licenses for the server.

4. Terminate Contracts

With the server scheduled for decommission, any vendor maintenance for the associated hardware or software will be unnecessary.

Go ahead and schedule the cancellation of these contracts and communicate.

5. Create Backups

It is likely that there will be information within the server that needs to be retained.

Run tests to ensure that your backup process and disaster recovery are functioning.

Create a comprehensive backup and verify that all data was successfully backed up prior to decommissioning.

6. Wipe Data

If you plan to utilize data erasure software, this should be done while the server is still within its rack.

Be sure to follow all instructions from your chosen tool.

The primary benefit of data wiping is that erased data is unrecoverable and therefore more secure than alternative methods.

Additionally, wiping data is a more environmentally friendly option than physically destroying the hardware.

Data erasure also enables the resale of assets. Be certain that you are compliant with all global data and privacy regulations and protocols prior to reselling

7. Unplug

Disconnect the server from the network, then remove all subnets, access control lists (ACLs), and firewalls.

These processes can be quite complex and even lead to complications for your greater network, so be sure to work with qualified Network partners that can assist with each step.

8. Pack and Remove

It’s time to place the server on a pallet and remove it from your facility. This can be done in a few ways depending on your plan for the server. Your organization could repurpose the equipment, sell it, or dispose of it in some other fashion. Work with a trusted partner that has experience in whichever path you choose.

9. Coordinate with Other Departments

Reach out to every department within your organization that needs to be kept in the loop about the decommissioning process. For instance, accounting will need to update the books to reflect the loss of the server and to account for any software licenses.

While this server decommissioning checklist only features the high-level steps that need to be considered, this should give you a general framework for what areas will require the most time, consideration, and resources. Be sure to reference this server decommissioning checklist throughout the process.