Continuous ~~Integration~~ Intrusion: Why CI tools are an attacker's best friends

Nikhil Mittal

About me

- Twitter - @nikhil_mitt
- Blog – http://labofapenetrationtester.com
- Github - https://github.com/samratashok/
- Creator of Kautilya and Nishang
- Interested in Offensive Information Security, new attack vectors and methodologies to pwn systems.
- Previous Talks/Trainings
  - Blackhat, Defcon, CanSecWest, Shakacon and more.

Shameless Self Plug :)

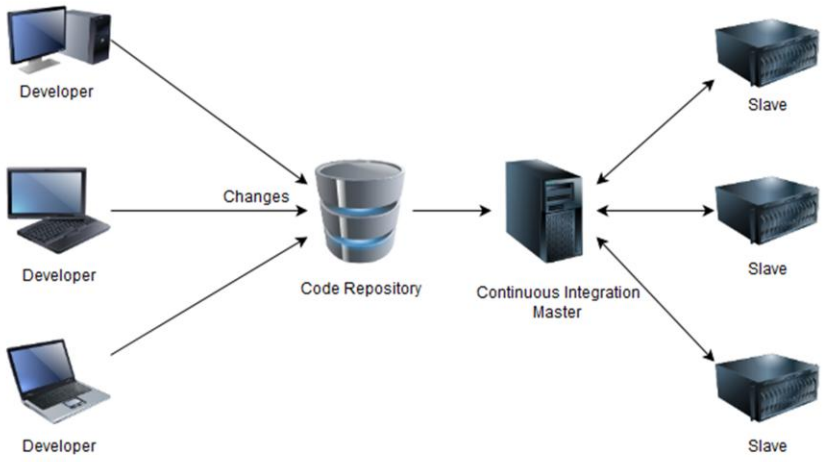
- Check out my trainings:
  - PowerShell for Penetration Testers:
    http://www.labofapenetrationtester.com/p/trainings.html
  - Basic Infrastructure Hacking with NoSoSecure at BH USA 2016:
    https://www.blackhat.com/us-16/training/basic-infrastructure-hacking.html

What are Continuous Integration (CI) Tools?

- “Continuous integration describes a set of software engineering practices that speed up the delivery of software by decreasing integration times. Software that accomplish this practice is called continuous integration software.” - Wikipedia

More about Continuous Integration:
http://www.martinfowler.com/articles/continuousIntegration.html

Basic diagram of Continuous Integration



Why CI tools are important for Hackers?

- In a typical enterprise setup, the build jobs are executed on master build server(s) and/or slave machines.
- If a hacker manages to get access to a CI tool, he effectively owns substantial part of the build process, source code (most of the times) and high privilege access to all the machines running slaves/agents.
- This access could be used for lateral movement to get access to more machines.

Abusing popular CI tools

- Jenkins (and Hudson)
- TeamCity
- Go
- CruiseControl
- We will mostly have a look at feature abuse and mis-configurations of the above tools, they may or may not be called vulnerabilities.

* Hudson was not evaluated separately. Most of the things which apply on Jenkins should apply on Hudson as well.

Jenkins

- Jenkins (http://jenkins-ci.org/) is arguably the most widely used Continuous Integration tool. It is an open source tool which supports distributed agents and build process.
- Hudson (http://hudson-ci.org/) was not tested separately since Jenkins and Hudson are very similar in code base and operations.

Jenkins

- No authentication in the default installation.
- No protection against brute force attacks.
- No password complexity/policy for user passwords.
- Runs with SYSTEM or high privilege user on Windows (never seen it running with non-admin privileges).
- Most Jenkins instance provide for user enumeration and access to build logs to everyone.

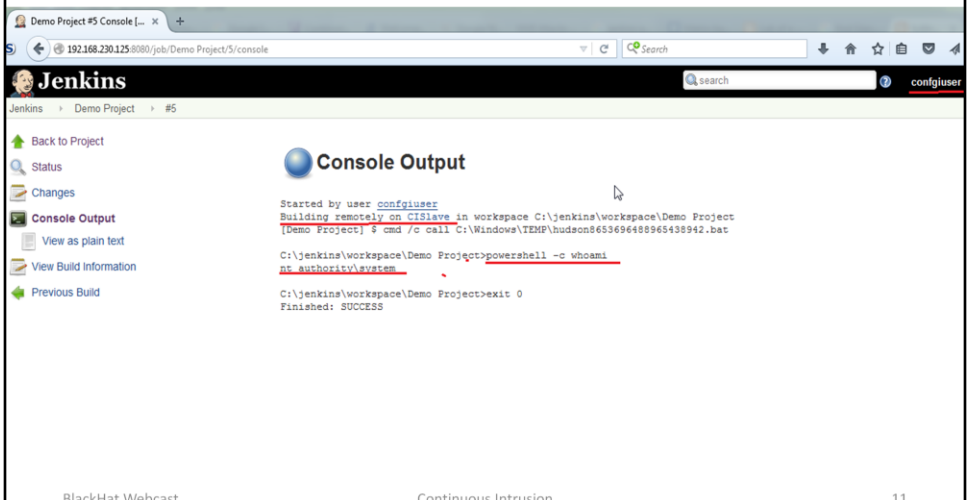
Jenkins

Abusing Build Steps

- If a user has the ability to “Add build step” (non-admin users can do that, if configured so) he can execute commands and scripts on the host Operating System.
- Due to the distributed nature of Jenkins, it is possible to execute commands and scripts on large number of slaves/agents.

http://www.labofapenetrationtester.com/2014/08/script-execution-and-privilege-esc-jenkins.html

Executing PowerShell/Windows commands using “Add Build Step”



The rights of the user to add or change build configuration are managed using Matrix based security or Project-based Matrix Authorization Strategy.
https://wiki.jenkins-ci.org/display/JENKINS/Matrix-based+security

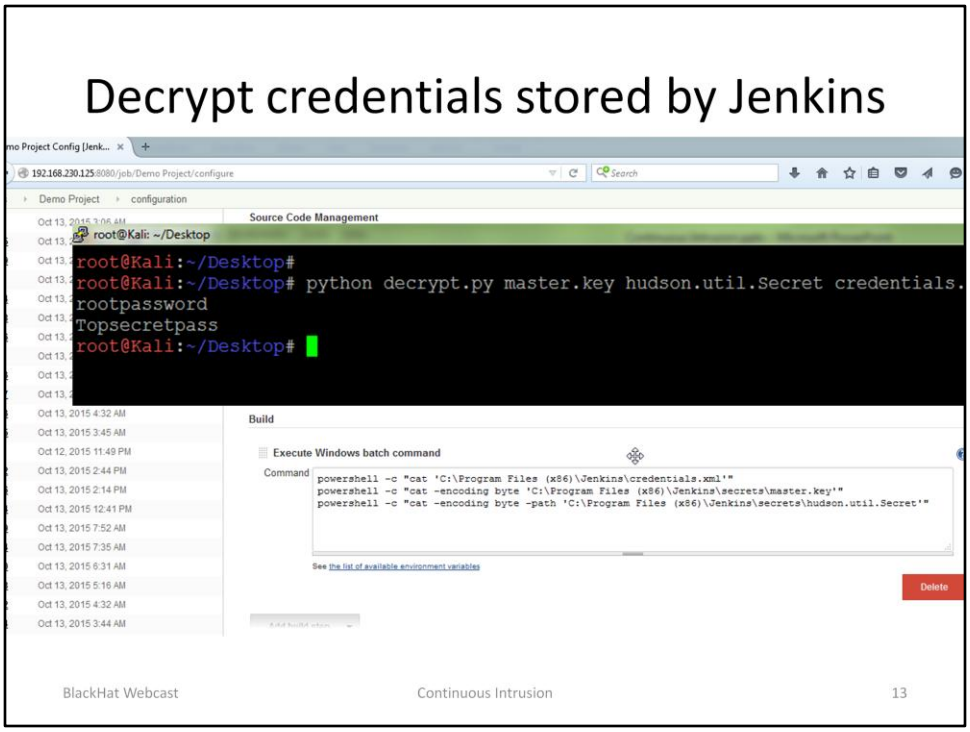
When running commands on a Windows machine we can leverage PowerShell to execute advanced scripts using this method.

Jenkins

Stored Credential retrieval in cleartext

- If a user has the ability to “Add build steps” **on master**, it is possible to retrieve all kind of credentials (passwords, SSH private keys and their passphrases etc) stored in Jenkins in clear text.
- Taken from http://thiébaud.fr/jenkins_credentials.html

Taken from http://thiébaud.fr/jenkins_credentials.html



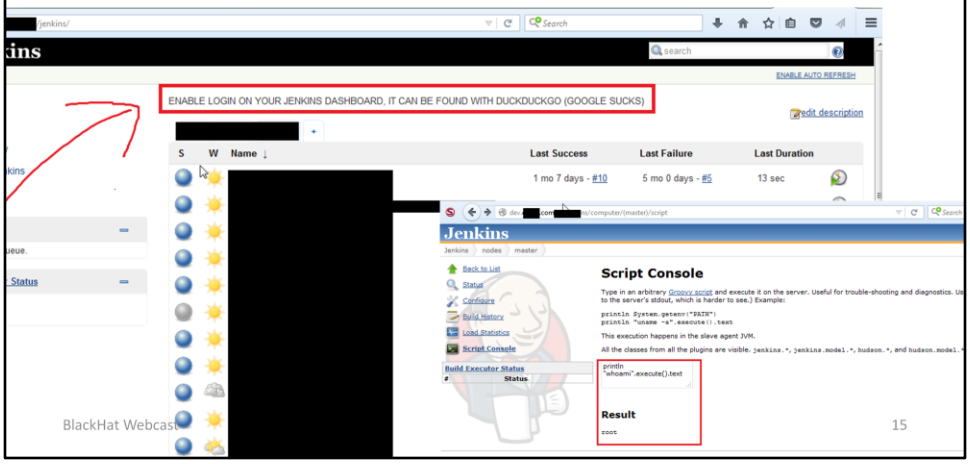
We need credentials.xml from $JENKINS_HOME and master.key and hudson.util.secret from $JENKINS_HOME/secrets/

We are reading the keys master.key and hudson.util.secret in bytes and will convert them back to file on our own machine. On a Windows machine the conversion could be done by using TextToExe.ps1 from Nishang.
https://github.com/samratashok/nishang/blob/master/Utility/TexttoExe.ps1

Jenkins

- Google Dorks: intitle:"Dashboard [Jenkins]" for Jenkins instances and intitle:"Dashboard [Jenkins]" intext:"Manage Jenkins“ for instances providing unauthenticated admin access.
- Sensitive data, database credentials, Git credentials, SSH keys and more could be accessed. Ridiculously simple to abuse.

ENABLE LOGIN ON YOUR JENKINS DASHBOARD, IT CAN BE FOUND WITH DUCKDUCKGO (GOOGLE SUCKS)




EVERYTIME YOU DON'T GIVE A LIVE DEMO
GOD KILLS A KITTEN
memegenerator.net

Demo of a Penetration Test Scenario

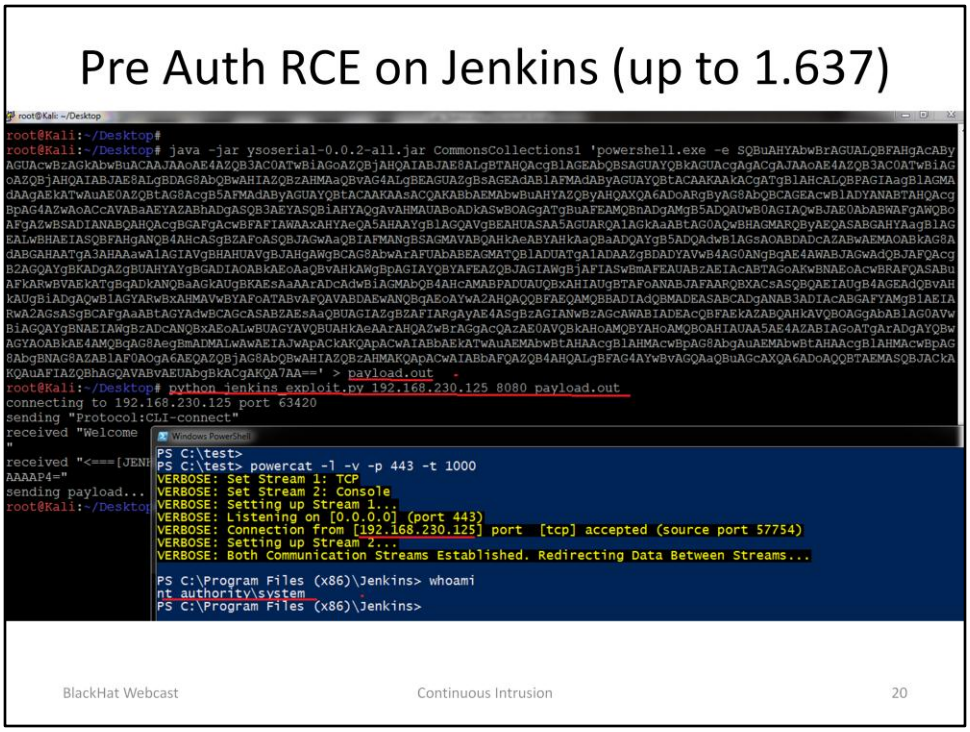
- A Jenkins (or any other CI tool) agent running on a Windows machine.
- Execute PowerShell scripts on it to enumerate available tokens.
- Reuse token of a Domain Administrator.
- Too easy? You will be surprised how often it works ;)


GENTLEMEN, WE HAVE THE DOMAN ADMIN PRIVILEGES
memegenerato

UnSerialize vulnerability in CI tools

- Unserialize vulnerability in common-collections Java library resulting in Pre-Auth Remote Code Execution (Released on 28th January 2015)
  http://www.slideshare.net/frohoff1/appseccali-2015-marshalling-pickles
- More details here:
  http://foxglovesecurity.com/2015/11/06/what-do-weblogic-websphere-jboss-jenkins-opennms-and-your-application-have-in-common-this-vulnerability/
- Jenkins and Go released fixes
  https://wiki.jenkins-ci.org/display/SECURITY/Jenkins+Security+Advisory+2015-11-11
  http://www.go.cd/2015/11/09/deserialization-vulnerability-commons-collections.html

Also see: https://github.com/foxglovesec/JavaUnserializeExploits
https://github.com/frohoff/ysoserial



I am using an encoded one line PowerShell reverse shell from Nishang as the payload in the above screenshot.
(https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcpOneLine.ps1)

Is it only Jenkins?

NO!

Missing Security Controls

- No/poor protection against brute force attacks.
  - Multiple attempts
  - No password policy
  - No captcha
  - Login over HTTP
- Insecure storage of SSH keys and other credentials stored by the tool.
- Higher privileges (SYSTEM) on Windows for master and slaves.

Mis-Configurations

- Agent/Build executor on master.
- Read permission to anonymous/guest users.
- Unencrypted communication between masters and slaves.
- Poor User practices:
  - Use of username as password. More so when the CI tool has its own users.
  - Passwords in build parameters.
- Publicly exposed CI tools.

Feature Abuse

- Command Execution on the Operating System without needing high privileges on the CI tool.

Defense (Users)

- No build agent/executor should run on the master EVER.
- Restrict privileges which allow configuration of build steps.
- Pool agents together for specific projects.
- Secure the administrators dashboard.
- Expose a CI tool to the internet only if really required.
- Do not provide read privileges to Anonymous users.
- Ask users at least not to use username as password.

Defense (CI Tools Developers)

- Enforce password policies (complexity, expiry, captcha on login failures etc.)
- A user must install agent/slave explicitly on the master. Do not include it in the install package. Show a warning if an agent/slave is installed on the master.
- Do not store credentials and keys in cleartext on master.
- Assume that your users don't know a thing about security.

What others are doing?

- "The Jenkins project has received multiple credible reports indicating that unsecured, publicly accessible instances of Jenkins are being targeted and infected with malware" https://wiki.jenkins-ci.org/display/SECURITY/Jenkins+Security+Advisory+2015-10-01
- Facebook used to run an unauthenticated Jenkins instance http://blog.dewhurstsecurity.com/2014/12/09/how-i-hacked-facebook.html
- "We got hacked. We were running a (unauthenticated) Jenkins instance in this machine." – plumber.eu https://plumbr.eu/blog/plumbr-blog/we-got-hacked

Previous Work

- Is this your pipe? Hijacking the build pipeline. (Defcon 22): https://goo.gl/9iTHmz
- https://www.pentestgeek.com/penetration-testing/hacking-jenkins-servers-with-no-password/
- http://www.labofapenetrationtester.com/2014/06/hacking-jenkins-servers.html
- http://www.labofapenetrationtester.com/2014/08/script-execution-and-privilege-esc-jenkins.html
- http://zeroknock.blogspot.in/2013/08/protect-your-software-development-web.html

Conclusion

- CI tools lack many basic security controls. Still, paying attention to them and configuring them properly will reduce the attack surface.
- Due to the distributed nature of their working, compromise of UI/dashboard even with non-admin privs may result in access to all slave machines.
- Running a poorly configured CI tool is like providing a ready-to-use botnet for anyone to exploit.

Thank you

- Questions?
- Please provide feedback.
- Follow me @nikhil_mitt
- nikhil.uitrgpv@gmail.com
- http://labofapenetrationtester.com/
- https://github.com/samratashok/ContinuousIntrusion