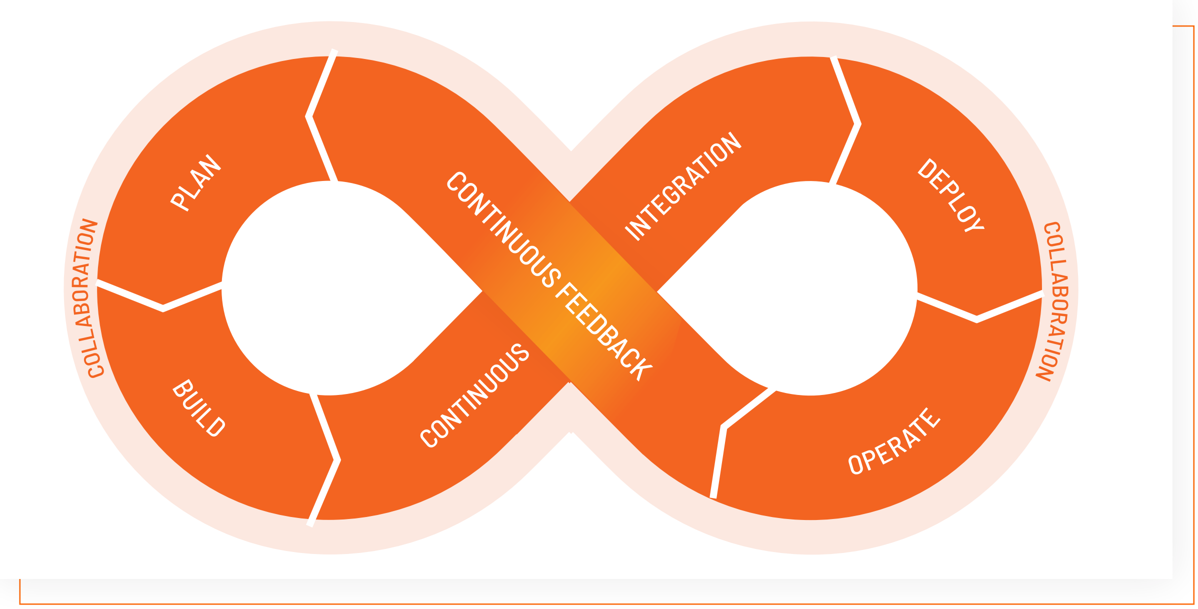
WHITEPAPER

Attivo NETWORKS®

INTEGRATING DECEPTION WITH DEVOPS

DevOpsSec or **DevSecOps** is the process of integrating security best practices as part of the development and deployment process. DevOps is a continuous cycle that uses automation throughout the phases of planning, development, and deployment.

Every organization has its DevOps methods, and this whitepaper focuses on ways an attacker can exploit a weakness in the organization's DevOps practices while examining how one can integrate deception into the DevOps cycle to provide internal security monitoring and protect critical data.



DEVOPS SERVICES AND THE ATTACK CYCLE

The DevOps services allow teams to collaborate, develop code, build, and deploy applications. The following are some of the commonly used DevOps services in an organization.

Repositories: Source code repositories (Gitlab, GitHub, Git, SVN, etc.)
CI/CD Automation Tools: Jenkins, Puppet, Chef, Ansible, etc.
Project Management Tools: JIRA, Kanban Boards
Bug/Test tracking tools: Trac, Bugzilla, JIRA, etc.
Active Directory: Kerberos authentication to DevOps services, Service accounts for automation
Production Applications: Organization apps deployed for production

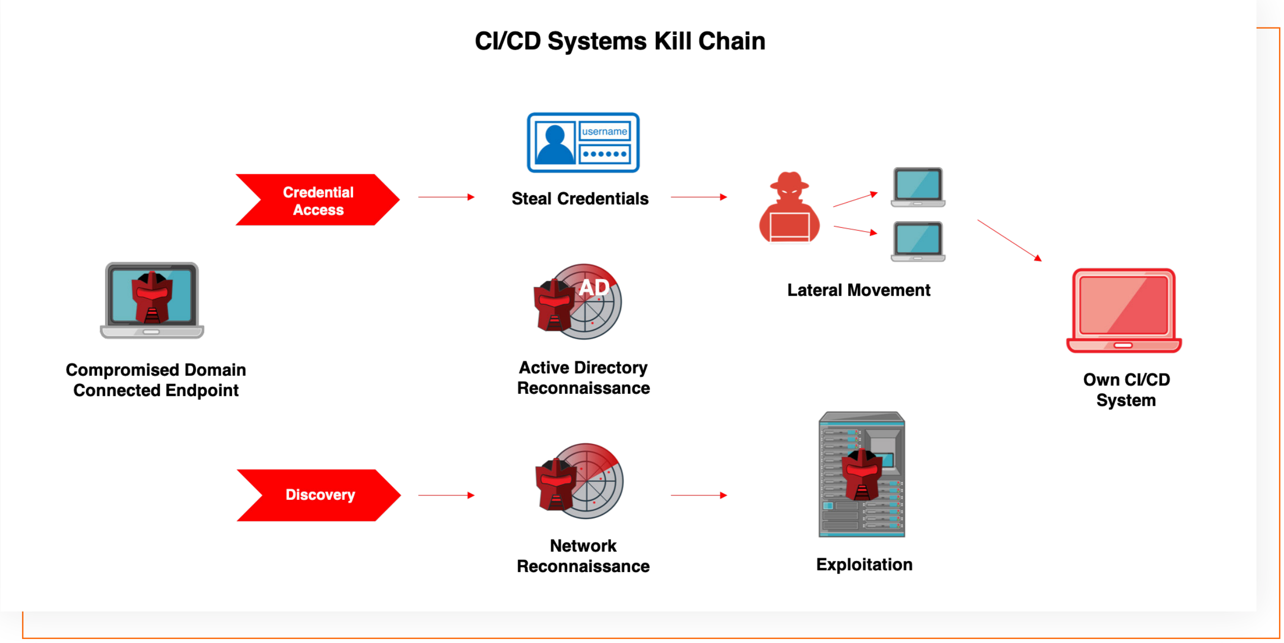
Azure DevOps is another popular tool that allows one to manage software projects, from planning and development through testing and deployment.

DevOps services are becoming an attractive target for attackers since they have access to the entire development and production infrastructure. Attackers can compromise DevOps systems by exploiting 0-day or unpatched vulnerabilities, using stolen credentials, exploiting misconfigurations, etc. Attacks against DevOps services fall under two categories – those originating from internal networks, and those that arise from production applications outside the network. Most organizations provide access to DevOps services using Active Directory user groups with varying levels of access permissions. DevOps security solutions should detect attackers targeting users with access permissions to DevOps services.

INTEGRATING DECEPTION INTO DEVOPS

Organizations can deploy deception across the DevOps cycle to get insights into attacker activity and alert on any misconfigurations. Deception complements existing DevSecOps security controls and can help with threat modeling and generation of IOCs (Indicators of Compromise) in each phase of DevOps.

Deception can protect DevOps services used internally in the network. Attackers from a compromised endpoint must perform Active Directory or Network reconnaissance to identify DevOps systems to exploit for any 0-day or known vulnerabilities. Deception can detect attackers early in the internal reconnaissance phases of the attack life cycle and misdirect them away from production assets.



Planning Phase:

The DevOps planning phase begins with requirements gathering, design planning, and tracking project planning activities. Design documents contain confidential information about the project, database schemas, and the organization's intellectual property. Attackers can steal credentials from an infected system, exploit vulnerabilities, etc. to target repositories, file shares, etc. and take these documents.

Deception Use Cases: Organizations can deploy deception during the planning phase to steer attackers towards decoys assets like the following and away from production assets.

1. decoy file servers to host decoy project documents
2. decoy code repositories (Gitlab, GitHub servers, etc.)
3. decoy web-based project management tools (JIRA, Kanban board, etc.)
4. decoy breadcrumbs on endpoints, pointing to decoy systems.
5. decoy network shares pointing to decoy file servers to protect against ransomware that targets network shares
6. decoy documents to detect intruders stealing organization intellectual property.

Build Phase:

The build phase starts along with software development, and organizations use various continuous integration (CI) and continuous delivery (CD) solutions to automate build and test cycles.

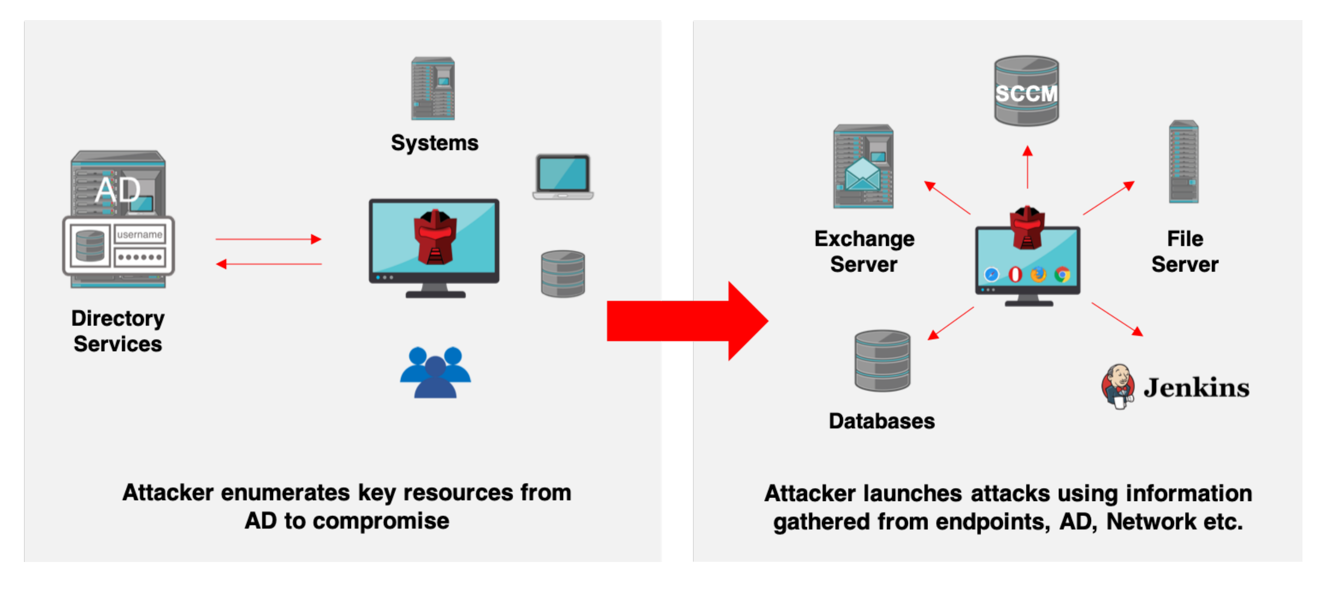
Since Jenkins is a commonly-used open-source automation server that organizations adopt for (CI/CD) process, this work uses it as a CI/CD tool for the following examples. Jenkins is one of the primary systems that attackers target to gain access to the entire CI/CD pipelines. Organizations deploy Jenkins servers within the internal network, and attackers can target them to exploit vulnerabilities, misconfigurations, or steal credentials for access.

Attackers perform network reconnaissance, Active Directory enumeration, or look for artifacts on endpoints to identify an organization's Jenkins servers.

Attackers can use the following methods to enumerate Active Directory to find Jenkins systems.

- "net group "domain computers" /domain to find all computers in the domain and find Jenkins servers, file servers, Exchange servers, etc.
- Use PowerShell queries to discover operating systems, servers, etc., from Active Directory.

  ([adsisearcher]"objectcategory=computer").findall() | ForEach {([adsi]$_.path).operatingsystem}
- Using Service Principle Scanning "setspn -Q */*.", to discover services running on servers


AD
username
Directory Services
Systems
Attacker enumerates key resources from AD to compromise
SCCM
Exchange Server
File Server
Databases
Jenkins
Attacker launches attacks using information gathered from endpoints, AD, Network etc.

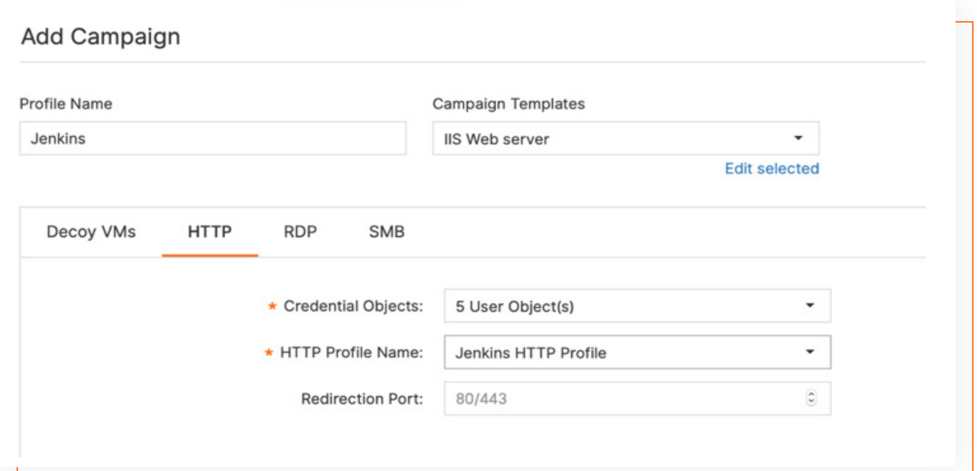
Deception Use Cases: Organizations can deploy decoy Jenkins servers during the build phase to steer attackers performing reconnaissance towards decoy Jenkins systems and away from production servers.

1. Deploy decoy Jenkins servers on the same operating system as a production system. The decoy Jenkins should be identical to the production Jenkins systems in terms of OS, Software Version, etc.

2. Deploy decoy breadcrumbs on endpoints pointing to the decoy Jenkins systems. These decoy credentials redirect attackers looking for Jenkins artifacts on endpoints to decoy systems.

3. Hide production Jenkins systems from Active Directory. When attackers query Active Directory using any of the methods mentioned above, they discover decoy systems instead of real production systems.

The Attivo Networks ThreatDefend platform supports deploying decoy Jenkins campaigns.


Add Campaign

Profile Name: Jenkins

Campaign Templates: IIS Web server

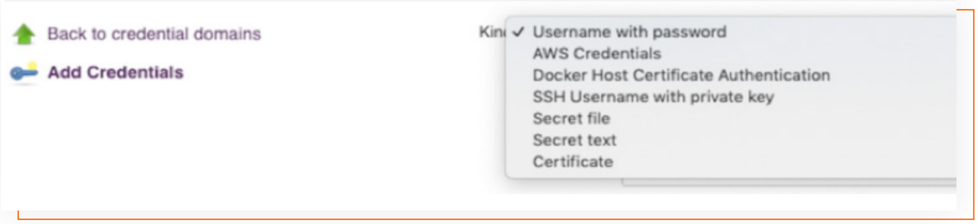
* Credential Objects: 5 User Object(s)

* HTTP Profile Name: Jenkins HTTP Profile

Redirection Port: 80/443

Deploy Phase:

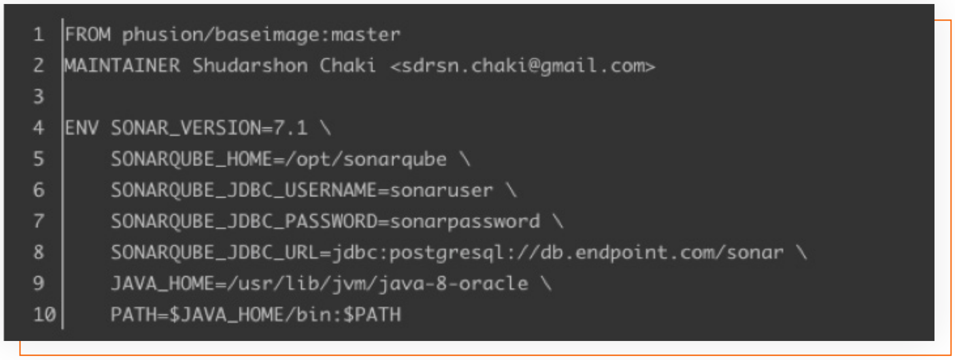
The deployment phase is one of the crucial interfaces between the Dev and Ops (Development and Operation) phases. During the deployment phase, the CI/CD system needs access to various credentials, keys, secrets, etc., to authenticate with external systems to deploy and manage applications.



- ✓ Username with password
- AWS Credentials
- Docker Host Certificate Authentication
- SSH Username with private key
- Secret file
- Secret text
- Certificate

Once attackers get access to Jenkins systems, they can steal credentials stored in Jenkins to various systems or upload malware or malicious scripts to them.

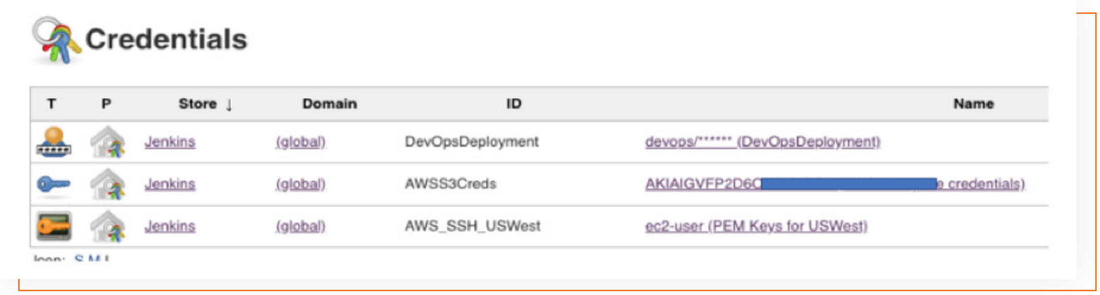
Developers also pass credentials, secrets, keys, and other access tokens in various forms to applications to gain access to backend resources like databases, file servers, etc. Once attackers land on target systems, they look for target database systems, credentials, key hashes, etc. from configuration files.



```
FROM phusion/baseimage:master
MAINTAINER Shudarshon Chaki <sdrsn.chaki@gmail.com>

ENV SONAR_VERSION=7.1 \
    SONARQUBE_HOME=/opt/sonarqube \
    SONARQUBE_JDBC_USERNAME=sonaruser \
    SONARQUBE_JDBC_PASSWORD=sonarpassword \
    SONARQUBE_JDBC_URL=jdbc:postgresql://db.endpoint.com/sonar \
    JAVA_HOME=/usr/lib/jvm/java-8-oracle \
    PATH=$JAVA_HOME/bin:$PATH
```

Deception Use Cases: Organizations can deploy decoy credentials as part of the Jenkins credential store. The credentials can point to decoy systems that it can monitor for persistence or malicious payload drops.



T	P	Store ↓	Domain	ID	Name
		Jenkins	(global)	DevOpsDeployment	devoos/****** (DevOpsDeployment)
		Jenkins	(global)	AWSS3Creds	AKIAIGVFP2D6C[redacted]e credentials)
		Jenkins	(global)	AWS_SSH_USWest	ec2-user (PEM Keys for USWest)

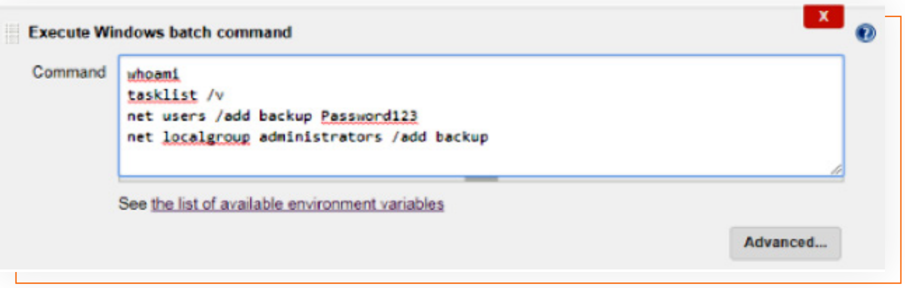
Attackers target running processes and configuration files to search for database and file server credentials. Organizations can lure attackers by passing deceptive servers, credentials, or secrets as environmental variables or other parameters in configuration files and script files that they may use for deployment.

Organizations can also embed deception in docker configuration files. Attackers who are in the network can search for parameters the DevOps team passes (server names, credentials) to docker containers. DevOps teams can plant deception in configuration files and misdirect attackers to decoy systems.

Operate Phase:

The operational phase is the last phase of the CI/CD cycle, during which organizations need to deploy production code, monitor the environment for any downtimes, security breaches, etc.

Since the CI/CD systems itself could be one of the primary targets, attackers could deploy crypto miners, ransomware, targeted malware, and other malicious applications on remote systems. Attackers can also use scripts to create accounts for persistence on target systems.


Execute Windows batch command
X
Command
See the list of available environment variables
Advanced...

```
whoami
tasklist /v
net users /add backup Password123
net localgroup administrators /add backup
```

Deception Use Cases: Organizations can use the following means to detect and prevent attacks in the operate phase:

- Create network decoys that mimic the production apps
- Deploy endpoint lures at significant endpoints to mislead an attacker towards decoys
- Selectively embed decoy documents and files at coveted locations
- Use attack surface reduction technology to monitor the credentials exposure and changes in privileges

DECEPTION CHECKLIST FOR DEVOPS SECURITY

We discussed several techniques that can help prevent and detect attacks much earlier in the attack cycle. The checklist below summarizes these for the various stages of DevOps:

- Integrate CI/CD Systems (Jenkins, etc.), code repositories systems (Gitlab, GitHub, etc.), file servers, Kanban project management servers, etc. into the organization's Active Directory.

- Prevent attackers from enumerating and finding one's production CI/CD, code repositories, file servers, Kanban, etc. from Active Directory and hiding real servers.

- Deploy decoy CI/CD, code repositories systems, file servers, Kanban project management tools as decoys.

- Redirect attackers enumerating organization Active Directory to decoy CI/CD, code repositories systems, file servers, Kanban project management, etc.

- Deploy decoy breadcrumbs on endpoints pointing to decoy CI/CD, code repositories, file servers, Kanban project management servers, etc.

- Embed deceptive credentials as part of scripts and deploy them in parallel with one's existing scripts used for continuous operations.

- Deploy decoy documents on endpoints and file servers

- Monitor activities at the cmd line or PowerShell level to detect various discovery activities

- Deploy Attack Surface Reduction technology that continuously assesses and alerts when a privileged credential gets exposed.

CONCLUSION

In addition to organizations securing and correctly configuring DevOps CI/CD systems, they should also be using detection technology that will detect its exploitation, attempts to steal credentials, and misconfigurations, which can result in attackers exposing its stored various secrets. Additionally, since attackers can modify pipelines and deploy malware or add accounts for persistence, DevOps teams should look to new security controls such as deception technology to defend Jenkins and other CI/CD solutions from compromise.

ABOUT ATTIVO NETWORKS®