Step 1: Deployment Umgebung (Simon)

Bitte Training Hetzner Umgebung nutzen
Mit Testnode puppet agent run zum Verifizieren des Setups.

Step 2: Einleitung (Martin)

Warum Secrets? Was sind Secrets? Wie schützt man Secrets?

1. Secrets im GIT Repository
2. Secrets in einem Report

Unsere Umgebung:
Workshop Deployment unter Verwendung von Bolt, Terraform und Puppet
Wir arbeiten hier mit Puppet Enterprise, alles, was wir zeigen geht auch unter Puppet Open Source

Step 3: Einführung Secrets in GIT (Martin)

Warum überhaupt Secrets in GIT?

Systeme benötigen Config Elemente (DB passwörter, ssl keys und certs, …)

2 Möglichkeiten:
- Daten verschlüssen und ablegen in GIT
- Daten an andere Stelle ablegen

Lösung 1:
- Hiera EYAML
- Hiera EYAML-GPG

Unterschiede:
EYAML hat EIN public private PKCS7 Key Paar. Public key zum Verschlüsseln, private Key zum Entschlüsseln (für alle).
EYAML-GPG: GPG Key Chain mit mehreren Keys (pro User)

Lösung 2:
- Hashicorp Vault

Hashicorp Vault ist ein webbasierter Passwort Manager mit REST API

Verfügbar ab Puppet 5.x

Ausgelagert, nicht in Hiera auf Dateisystem-Ebene (GIT).

Step 4: Einführung Secrets im Report (Martin)

Puppet liefert am Ende einen Report mit den Änderungen. Dort stehen auch gerne file diffs. Jeder, der Reports einsehen kann, kann alten und neuen Secret Wert auslesen.

Lösung: Sensitive Data Type (seit Puppet 6.x - funktional)

Bei einem Sensitive Data nimmt Puppet die Plaintext Werte aus dem Report heraus und ersetzt diese durch den String [VALUE REDACTED]

Fragen?

Step 5: DEMO EYAML (Simon)

EYAML Code Demo:
- Puppet Server: eyaml encrypt zeigen CLI
- Testnode YAML Daten
- Puppet Klasse für EYAML
- Hiera.yaml
    - lookup_key

    - options: pkcs7*key
- common.yaml
- puppet lookup command
- Testnode puppet agent -t –tags pkcs_sieben
    - 3 Ergebnisse

Fragen?

Step 6: DEMO Sensitive Data Type (Martin)

- Bestandteil Puppet DSL
- Daten Typ z.B. für Klassen Parameter
- pkcs_sieben lookup code zeigen
- Hiera liefert normalerweise String - lookup options mit convert_to nutzen

Fragen ?

Step 7: DEMO EYAML GPG (Simon)

Eine Erweiterung für EYAML. Muss als Ruby GEM nachinstalliert werden.

1. Hiera internal lookup: puppetserver gem install hiera-eyaml-gpg ruby_gpg
2. Puppet Lookup Command: /opt/puppetlabs/puppet/bin/gem install hiera-eyaml–gpg ruby_gpg

GPG Key müssen erzeugt werden. Siehe Blog Artikel
GPG Pub Key muss zum Schlüsselring hinzugefügt werden. (gpg –import <file>)

Hiera.yaml:

- lookup_key: eyaml_lookup_key
- options: gpg_gnupghome: Pfad und Datei zum Kering

Achtung: GPG Keychain darf nicht mit Passphrase geschätzt sein!!!!

Zeigen anlegen eines neuen Secrets

- Testnode puppet agent -t –tags gpg

Fragen ?

Step 8: DEMO Vault (Martin/Tim)

Ruby GEMS Installation

1. Hiera automatic data binding: puppetserver gem install vault debouncer
2. Puppet CLI lookup command puppet gem install vault debouncer

Vault erzeugt bei der Installation einen Token.
Mit diesem Root Token kann man am Vault alles machen!!!

Der Token muss auf dem Puppet Server in einer Datei hinterlegt werden::
echo “IGNORE-VAULT” > /etc/vault_token

Hinweis: IGNORE-VAULT sorgt dafür, dass Vault gar nicht benutzt wird.

Hiera.yaml Datei

- lookup_key: hiera_vault
- options:
    - address: https://xxxx:yyyy
    - token: /etc/vault_token
    - default_field: value
    - mounts:
        - puppet:
            - facts.networking.hostname

Achtung: Wenn man ausschliesslich Vault nutzt, bekommt der Vault Server alle Hiera lookup anfragen!!!
Möglichkeit in der Hiera.yaml: options: confine_to: -regexp>
Bei Mounts wird ein Verzeichnis angegeben. Die Hierarchien liegen dann in dem Verzeichnis.

Vault Zugriff
Vault Inhalt (Webinterface) zeigen

- Testnode puppet agent -t –tags vault

Fragen?

Ausblick: (Alle)

Bolt (Open Source) ist ein CLI Tool. Hier müssen Daten Plainttext übergeben werden.
Bolt Integration in PE: Die Web UI erkennt, dass es um Sensitive Daten geht und stellt das Formular Eingabefeld als Passwort (****) dar.

Hinweis: GIt Repo kommt noch - URL kommt in das Meetup Event als Kommentar.

History Puppet Server

```
  56  eyaml encrypt -n gpg --gpg-always-trust -s "A secret string to encrypt"
--gpg-recipients-file /tmp/control-repo/data/gpg_recipients
  57  cd /tmp/control-repo/
```

```
  58  cat hiera.yaml
  59  cat /etc/vault_token
  60  puppet agent -t --noop
  61  clear
  62  which eyaml
  63  eyaml
  64  eyaml encrypt
  65  eyaml encrypt --help
  66  eyaml encrypt -s 'S3cr3ts!'
  67  ls
  68  ls data/
  69  eyaml encrypt -s 'S3cr3ts!' --pkcs7-public-key data/pkcs7_keys/public_key.pkcs7.pem
  70  cat data/common.yaml
  71  puppet lookup pkcs_sieben_lookup_example
  72  eyaml
  73  eyaml decrypt --help
  74  eyaml decrypt --pkcs7-private-key=data/pkcs7_keys/private_key.pkcs7.pem --help
  75  eyaml decrypt --pkcs7-private-key=data/pkcs7_keys/private_key.pkcs7.pem -s
'ENC[PKCS7,MIICiQYJKoZIhvcNAQcDoIICejCCAnYCAQAxggIhMIICHQIBADAFMAACAQE
wDQYJKoZIhvcNAQEBBQAEggIAMeBpgF3f/i3tYaNIJ/8lAd6rwcbLeaWNEEqQ4+iK3qsRd+S
1BzJ0C4LMopoU6jUVWyE7XxYnQZkDLOoXvR3uu+u2PEGtRBmiq/noCULdWuxJu1woMiA
OeVmCyLxFZxyxV8eDyB4fL+pC30zYkPG73JK8DdouJJNBkYDUdydjpapZrcEZ5T9NB1AG1
pyePO0HxocnsvUOsvgSY15K8nxz1XPIMD4z+AGAjH3/ykgXUwozxjz35D0Fy69UfhP3RMUl
V44UR88tsRB9bIHxJEQImfIlZ+NTUVozBAwAf89ixpbVh9+IBk7zIC0ghduSqI3Y+f4QvbAq9F
Ker0yczluLP7sKtQgFE1+zO9Bg/4Xzox3+rX2KPNy+SVczprXd6lr9E0C5elA/QJ7O35mDrff2k
V4Okx9UyW0vzPFBfEerT7kVjF0cvDlFocQ2HSCfkDFAqHs6gNN5i4wkXR0sM15NU6Q3zWs
/Xjf2szx+PoSfHVVgw3yZVXo1IS1+OkuLjtItleEdMoUsmULERg7qvPkaeCCiFjE4XAy9nxC//N
Xb0GJKrN/a8oMtca7nzpjoogFpQY5+DOg2ChkLTGOn5cnyDwj/b9YiDwUFyTfhcMAAsapR7
G8kSGk6EOnC0qq2om2CN3Jp8gPYowfKWtFZB50RcNrkVLnwSazNL1qiLyMrHl4wTAYJKo
ZIhvcNAQcBMB0GCWCGSAFlAwQBKgQQceEb4vYFm1S8XHSwEh27HoAgZ0E2o31QtZL
cGgzbxeQB3ewu4cfTn9QWXiMGVLJRx9o=]'
  76  eyaml decrypt --pkcs7-public-key=data/pkcs7_keys/public_key.pkcs7.pem
--pkcs7-private-key=data/pkcs7_keys/private_key.pkcs7.pem -s
'ENC[PKCS7,MIICiQYJKoZIhvcNAQcDoIICejCCAnYCAQAxggIhMIICHQIBADAFMAACAQE
wDQYJKoZIhvcNAQEBBQAEggIAMeBpgF3f/i3tYaNIJ/8lAd6rwcbLeaWNEEqQ4+iK3qsRd+S
1BzJ0C4LMopoU6jUVWyE7XxYnQZkDLOoXvR3uu+u2PEGtRBmiq/noCULdWuxJu1woMiA
OeVmCyLxFZxyxV8eDyB4fL+pC30zYkPG73JK8DdouJJNBkYDUdydjpapZrcEZ5T9NB1AG1
pyePO0HxocnsvUOsvgSY15K8nxz1XPIMD4z+AGAjH3/ykgXUwozxjz35D0Fy69UfhP3RMUl
V44UR88tsRB9bIHxJEQImfIlZ+NTUVozBAwAf89ixpbVh9+IBk7zIC0ghduSqI3Y+f4QvbAq9F
Ker0yczluLP7sKtQgFE1+zO9Bg/4Xzox3+rX2KPNy+SVczprXd6lr9E0C5elA/QJ7O35mDrff2k
V4Okx9UyW0vzPFBfEerT7kVjF0cvDlFocQ2HSCfkDFAqHs6gNN5i4wkXR0sM15NU6Q3zWs
/Xjf2szx+PoSfHVVgw3yZVXo1IS1+OkuLjtItleEdMoUsmULERg7qvPkaeCCiFjE4XAy9nxC//N
Xb0GJKrN/a8oMtca7nzpjoogFpQY5+DOg2ChkLTGOn5cnyDwj/b9YiDwUFyTfhcMAAsapR7
G8kSGk6EOnC0qq2om2CN3Jp8gPYowfKWtFZB50RcNrkVLnwSazNL1qiLyMrHl4wTAYJKo
ZIhvcNAQcBMB0GCWCGSAFlAwQBKgQQceEb4vYFm1S8XHSwEh27HoAgZ0E2o31QtZL
cGgzbxeQB3ewu4cfTn9QWXiMGVLJRx9o=]'
  77  puppet lookup pkcs_sieben_lookup_example
  78  cat hiera.yaml
```

```
  79  nl -ba hiera.yaml
  80  ls -la /etc/puppetlabs/puppet/keys/
  81  cat data/common.yaml
  82  puppet lookup pkcs_sieben_lookup_example
  83  nl -ba site/profile/manifests/pkcs_sieben/lookup.pp
  84  cat data/nodes/testnode.yaml
  85  nl -ba site/profile/manifests/pkcs_sieben/lookup.pp
  86  nl -ba data/common.yaml
  87  vi data/common.yaml
  88  git status
  89  git add data/
  90  git commit -m 'deactivate senstive convert to'
  91  puppet code deploy production -w
  92  git log -p 1
  93  git log -p -1
  94  puppet code deploy production -w
  95  git log -1
  96  git revert c982e415b4ef4dc09ea5d4c4df190614f7f15467
  97  vi site/profile/manifests/pkcs_sieben/lookup.pp
  98  nl -ba site/profile/manifests/pkcs_sieben/lookup.pp
  99  vi data/common.yaml
 100  vi data/nodes/testnode.yaml
 101  git add data/nodes/testnode.yaml
 102  git commit -m 'sensitive als plain text in hiera'
 103  puppet code deploy production -w
 104  nl -ba site/profile/manifests/pkcs_sieben/lookup.pp
 105  nl -ba hiera.yaml
 106  ls -la /opt/puppetlabs/server/data/puppetserver/
 107  ls -la /opt/puppetlabs/server/data/puppetserver/.gnupg/
 108  nl -ba hiera.yaml
 109  nl -ba data/common.yaml
 110  puppet lookup gpg_lookup_example
 111  #eyaml encrypt -n gpg --gpg-always-trust -s "A secret string to encrypt"
--gpg-recipients-file /tmp/control-repo/data/gpg_recipients
 112  cat /tmp/control-repo/data/gpg_recipients
 113  gpg -K
 114  gpg -k
 115  ls data/
 116  ls data/gpg_pubkeys/
 117  ls data/gpg_workshop/
 118  eyaml encrypt -n gpg --gpg-always-trust -s "A secret string to encrypt"
--gpg-recipients-file /tmp/control-repo/data/gpg_recipients
 119  nl -ba data/nodes/testnode.yaml
 120  nl ba site/profile/manifests/gpg/lookup.pp
 121  puppet lookup profile::gpg::lookup::parameter_lookup --node testnode.domain.tld
 122  git log -p
 123  puppet lookup --help
```

```
  124  puppet lookup profile::gpg::lookup::parameter_lookup --node
testnode.meetup.private.betadots.training
  125  nl ba site/profile/manifests/gpg/lookup.pp
  126  nl -ba hiera.yaml
  127  cat /etc/vault_token
  128  clear
  129  nl -ba hiera.yaml
  130  cat site/profile/manifests/gpg/lookup.pp
  131  cat site/profile/manifests/vault/lookup.pp
  132  nl -ba hiera.yaml
  133  nl -ba site/profile/manifests/vault/lookup.pp
  134  vi /etc/vault_token
  135  cp /etc/vault_token /etc/vault_token.orig
  136  vi /etc/vault_token
  137  yum install bolt
  138  bolt plan show
```