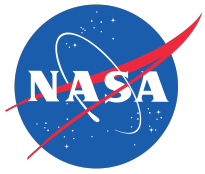

NASA TECHNICAL SPECIFICATION

National Aeronautics and Space Administration

NASA-SPEC2601APP.OpenSSH v1.4
Approved: 2021-07-30

Superseding: NASA-SPEC2601APP.OpenSSH 1.3

OPENSSH
SECURITY CONFIGURATION SPECIFICATION

TABLE OF CONTENTS

DOCUMENT HISTORY LOG

Document Version	Approval Date	Description
1.4	2021-07-30	Baseline
1.3	2021-03-31	Baseline
1.2	2020-08-25	Changes from AWARE
1.1	2020-02-14	Baseline, FIPS 140-2 checks added
1.0	2019-10-09	Baseline
0.9	2019-08-16	Initial Draft

FOREWORD

This NASA Technical Specification is published by the National Aeronautics and Space Administration (NASA) to describe technical requirements for purchased or in-house items, services, functions, or processes for NASA programs and projects.

This NASA Technical Specification is approved for use by NASA Headquarters and NASA Centers and Facilities, and applicable technical requirements may be cited in contract, program, and other Agency documents. It may also apply to the Jet Propulsion Laboratory (a Federally Funded Research and Development Center (FFRDC)), other contractors, recipients of grants and cooperative agreements, and parties to other agreements only to the extent specified or referenced in applicable contracts, grants, or agreements.

Adherence to this NASA Technical Specification ensures compliance with NASA-STD-2601, *Minimum Cybersecurity Requirements for Computing Systems*, which defines operating system and application security requirements that must be implemented on NASA information systems. This Specification provides the enforceable, measurable details of NASA-STD-2601.

Requests for information, corrections, or additions to this Specification can be made via the "Contact ASCS" form, found here: https://cset.nasa.gov/ascs/requests/.

Refer to NASA-SPEC-2600 - *Enumeration of ASCS Cybersecurity Requirements* (Signed - 2021-07-30)

Michael Witt

Senior Agency Information Security Officer

1. SCOPE

1.1. Purpose

The purpose of this NASA Technical Specification is to provide mandatory version and configuration guidance for OpenSSH deployment and operation.

1.2. Applicability

This NASA Technical Specification is approved for use by NASA Headquarters and NASA Centers and Facilities, and applicable technical requirements may be cited in contract, program, and other Agency documents. It may also apply to the Jet Propulsion Laboratory (a Federally Funded Research and Development Center (FFRDC)), other contractors, recipients of grants and cooperative agreements, and parties to other agreements only to the extent specified or referenced in applicable contracts, grants, or agreements.

1.3. Tailoring

In accordance with NASA-STD-2601, *Minimum Cybersecurity Requirements for Computing Systems*, any and all risk-based decisions (RBDs) to tailor this NASA Technical Specification in order to meet the needs of a specific program, project, or system *SHALL* be approved by the responsible Information System Owner (ISO) and Authorizing Official (AO) and formally documented by the ISO or system administrators in the System Security Plan (SSP) under program or project requirements.

NASA-STD-2601 mandates that the AO *SHALL* ensure that only systems posing an acceptable level of risk to Agency assets, data, and personnel are approved for production operation and that the ISO *SHALL* ensure all necessary documentation is produced and maintained.

Note that some NASA Technical Specifications include configuration settings that are classified as "critical". Settings that are classified as "critical" *SHALL NOT* be eligible for any modifications through the application of an RBD.

1.4. Authority

The Agency Chief Information Officer (CIO) and Senior Agency Information Security Officer (SAISO) have authorized the Cybersecurity Standards and Engineering Team (CSET) via the Agency Security Configuration Standards (ASCS) initiative to create binding Technical Standards related to Agency cybersecurity topics.

The NASA Technical Standards Program (NTSP), sponsored by the Office of the NASA Chief Engineer, recognizes CSET as a standards-developing organization within the Agency. NTSP provides access to all technical standards at: https://standards.nasa.gov/.

2. APPLICABLE DOCUMENTS

2.1. General

The documents listed in this section contain provisions that constitute requirements of this NASA Technical Specification. These documents can serve as additional support in meeting the requirements defined in this Specification.

2.2. Government Documents

Document Number or Descriptor	Document Title
FIPS 140-2	*Security Requirements for Cryptographic Modules*
FIPS 186-4	*Digital Signature Standard*
NASA-STD-2601	*Minimum Cybersecurity Requirements for Computing Systems*
NIST SP 800-130	*A Framework for Designing Cryptographic Key Management Systems*
NPR 2810.1	*Security of Information Technology*

2.3. Non-Government Documents

Document Number or Descriptor	Document Title
OpenSSH Manual	Official OpenSSH Documentation

3. ACRONYMS, ABBREVIATIONS, AND DEFINITIONS

Table 1. Acronyms and Abbreviations

Term	Expanded
NASA	National Aeronautics and Space Administration
ACL (Procurement)	Assessed and Cleared List
ACL (Security)	Access Control List
AO	Authorizing Official
ASCS	Agency Security Configuration Standards
ASRL	Address Space Layout Randomization
CDM	Continuous Diagnostics and Mitigation
CIO	Chief Information Officer
CIS	Center for Internet Security
CMVP	Cryptographic Module Validation Program
CSET	Cybersecurity Standards and Engineering Team
DAC	Directory Access Controls
DAR	Data At Rest
DCCP	Datagram Congestion Control Protocol
DHCP	Dynamic Host Configuration Protocol
DISA	Defence Information System Agency
ESD	Enterprise Service Desk
FFRDC	Federally Funded Research and Development Center
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Modernization Act
FPKI	Federal Public Key Infrastructure
GSSAPI	Generic Security Service Application Program Interface
ICAM	Identity, Credential, and Access Management
ICMP	Internet Control Message Protocol
IDI	ICAM Device Integration
ISO	Information System Owner
JPL	Jet Propulsion Laboratory
LVM	Logical Volume Manager
MAC (Network)	Media Access Control
MAC (Security)	Mandatory Access Control

Term	Expanded
NCTR	NASA Client Trust Reference
NFS	Network File System
NIST	National Institute of Standards and Technology
NIS	Network Information Service
NPR	NASA Procedural Requirement
NSA	National Security Agency
NTAM	NASA Trust Anchor Management
NTSP	NASA Technical Standards Panel
OCIO	Office of the Chief Information Officer
PKI	Public Key Infrastructure
RBD	Risk Based Decision
RISCS	Risk Information and Security Compliance System
ROP	Return Oriented Programming
SAISO	Senior Agency Information Security Officer
SCAP	Security Content Automation Protocol
SCTP	Stream Control Transmission Protocol
SGID	Set Group ID
SOC	Security Operations Center
SSH	Secure Shell
SSP	System Security Plan
STIG	Security Technical Implementation Guide
SUID	Set User ID
TFTP	Trivial File Transfer Protocol
XCCDF	eXtensible Configuration Checklist Description Format

4. VERSION AND LIFECYCLE

This Specification applies to the Application defined in the chart below, as well as to any more recent version(s) of this Application.

Function	Secure Shell (SSH) application
Application	OpenSSH
Minimum Version	Vendor supplied latest version built to meet FIPS 140-2 CMVP validation

5. SECURITY CONFIGURATIONS

5.1. Required Settings

This section contains all settings **required** for Agency compliance, as well as a detailed description for each setting. **Required** controls are monitored, scored, and reported by the Agency.

Settings that are classified as **required** *MAY*, in limited and justified instances, be amended through the proper application of an RBD. To pursue an RBD, see section 1.3, Tailoring, of this Specification.

5.2. Critical Severity Settings

Security configuration settings that are classified as **critical** are reported to the Department of Homeland Security (DHS). All settings found in this section *SHALL NOT* be eligible for any modifications through the application of a risk based decision (RBD).

NASA-ASCS-40042: Disable Host-Based Authentication

SSH's cryptographic host-based authentication is more secure than `.rhosts` authentication. However, it is not recommended that hosts unilaterally trust one another, even within an organization.

To disable host-based authentication, add or correct the following line in `/etc/ssh/sshd_config` :

```
HostbasedAuthentication no
```

Rationale

SSH trust relationships mean a compromise on one host can allow an attacker to move to other trusted hosts without needing authentication. Additionally, host-based authentication does not permit configuring command restrictions or limits on what can be done on the destination server when accessed. Because of this, it is not recommended for automated access. It is not recommended for interactive users either, because it does not present an interactive login. This would not be considered a good practice, especially for accounts with elevated privileges.

NASA ASCS ID	NASA-ASCS-40042
Severity	**Critical**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST.IR.7966, NIST SP 800-53A CM-6.1 (iv)
Control	HostbasedAuthentication
Control Setting	no

Table 2. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221697
Red Hat Enterprise Linux 7	V-71959

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^HostbasedAuthentication"
SED_INLINE="s/^HostbasedAuthentication.*$/HostbasedAuthentication no/"
ECHO_APPEND="HostbasedAuthentication no"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="HostbasedAuthentication"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40061: Disable SSH Access via Empty Passwords

To explicitly disallow SSH login from accounts with empty passwords, add or correct the following line in `/etc/ssh/sshd_config` :

```
PermitEmptyPasswords no
```

Any accounts with empty passwords *SHALL* be disabled immediately, and the PAM configuration prevents users

from being able to assign themselves empty passwords.

Rationale

Configuring this setting for the SSH daemon provides additional assurance that remote login via SSH will require a password, even in the event of misconfiguration elsewhere.

NASA ASCS ID	NASA-ASCS-40061
Severity	**Critical**
Group	ssh/services
Other References	NIST SP 800-53 Revision 4 IA-2 (2), NIST SP 800-53 IA-2 (2), NIST SP 800-53A IA-2 (2).1
Control	PermitEmptyPasswords
Control Setting	no

Table 3. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221688
Red Hat Enterprise Linux 7	V-71939
SuSE Linux Enterprise 12	V-217268
SuSE Linux Enterprise 15	V-235032
Ubuntu 18.04 LTS	V-219314
Ubuntu 20.04 LTS	V-238218

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^PermitEmptyPasswords"
SED_INLINE="s/^PermitEmptyPasswords.*$/PermitEmptyPasswords no/"
ECHO_APPEND="PermitEmptyPasswords no"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="PermitEmptyPasswords"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40063: Disable Override of User Environment Options

To ensure users are not able to override environment options to the SSH daemon, add or correct the following line in `/etc/ssh/sshd_config` :

```
PermitUserEnvironment no
```

Rationale

Some configurations of SSH environment options can potentially allow users to bypass access restriction.

NASA ASCS ID	NASA-ASCS-40063
Severity	**Critical**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53A CM-6.1 (iv)
Control	PermitUserEnvironment
Control Setting	no

Table 4. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221696
Red Hat Enterprise Linux 7	V-71957
Red Hat Enterprise Linux 8	V-230330
SuSE Linux Enterprise 12	V-217269
Ubuntu 18.04 LTS	V-219314
Ubuntu 20.04 LTS	V-238218

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^PermitUserEnvironment"
SED_INLINE="s/^PermitUserEnvironment.*$/PermitUserEnvironment no/"
ECHO_APPEND="PermitUserEnvironment no"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="PermitUserEnvironment"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40068: Allow Only SSH Protocol 2

Only SSH protocol version 2 connections shall be permitted. The default setting in `/etc/ssh/sshd_config` is correct and can be verified by ensuring that the following line appears:

`Protocol` has been deprecated since version 7.4p1, adding the setting will cause `sshd` to generate unwanted log messages.

Setting in OpenSSH Version prior to 7.4p1

```
Protocol 2
```

Rationale

SSH protocol version 1 is an insecure implementation of the SSH protocol and has many well-known vulnerability exploits. Exploits of the SSH daemon could provide immediate root access to the system.

NASA ASCS ID	NASA-ASCS-40068
Severity	**Critical**
Group	ssh/services
Other References	NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53 CM-6 b, NIST SP 800-53A IA-5 (1).1 (v), NIST SP 800-53 Revision 4 IA-5 (1) (c), NIST SP 800-53A CM-6.1 (iv), NIST SP 800-53 IA-5 (1) (c)
Control	Protocol
Control Setting	2

Table 5. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221856
Red Hat Enterprise Linux 7	V-72251
Red Hat Enterprise Linux 8	V-230501
Ubuntu 18.04 LTS	V-219308

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^Protocol"
SED_INLINE="s/^Protocol.*$/Protocol 2/"
ECHO_APPEND="Protocol 2"

if [[ "$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep protocol )" != "" ]]
then
  if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
  then
    sed -i "$SED_INLINE" $CONFIG_FILE
  else
    echo $ECHO_APPEND >> $CONFIG_FILE
  fi
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="Protocol"
EXPECTED_VALUE="2"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40082: Enable Use of Privilege Separation

When enabled, SSH will create an unprivileged child process that has the privilege of the authenticated user. To enable privilege separation in SSH, add or correct the following line in the `/etc/ssh/sshd_config` file:

`UsePrivilegeSeparation` has been deprecated since version 7.5, adding the setting will cause `sshd` to generate unwanted log messages.

Setting in OpenSSH Version prior to 7.5

```
UsePrivilegeSeparation sandbox
```

Rationale

SSH daemon privilege separation causes the SSH process to drop root privileges when they are not needed, which decreases the impact of software vulnerabilities in the unprivileged section.

NASA ASCS ID	NASA-ASCS-40082
Severity	**Critical**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53A CM-6.1 (iv)
Control	UsePrivilegeSeparation
Control Setting	sandbox

Table 6. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221863
Red Hat Enterprise Linux 7	V-72265
SuSE Linux Enterprise 12	V-217278
SuSE Linux Enterprise 15	V-235011

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^UsePrivilegeSeparation"
SED_INLINE="s/^UsePrivilegeSeparation.*$/UsePrivilegeSeparation sandbox/"
ECHO_APPEND="UsePrivilegeSeparation sandbox"

if [[ "$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep useprivilegeseparation )"
!= "" ]]
then
  if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
  then
    sed -i "$SED_INLINE" $CONFIG_FILE
  else
    echo $ECHO_APPEND >> $CONFIG_FILE
  fi
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="UsePrivilegeSeparation"
EXPECTED_VALUE="sandbox"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  # provision for RHEL6
  if grep "Red Hat Enterprise Linux.*6\..*" /etc/redhat-release
  then
    if [[ "$ACTUAL_VALUE" != "yes" ]]
    then
      RESULT=FAIL
      REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
    fi
  else
    RESULT=FAIL
    REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
  fi
fi

end_fixlet
```

NASA-ASCS-40085: Enable Encrypted X11 Forwarding

By default, remote X11 connections are not encrypted when initiated by users. SSH has the capability to encrypt remote X11 connections when SSH's `X11Forwarding` option is enabled.

To enable X11 Forwarding, add or correct the following line in `/etc/ssh/sshd_config` :

```
X11Forwarding yes
```

Rationale

Open X displays allow an attacker to capture keystrokes and to execute commands remotely. Additionally, disabling X11 forwarding does not prevent users from forwarding X11 traffic, as users can always install their own forwarders.

NASA ASCS ID	NASA-ASCS-40085
Severity	**Critical**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53A CM-6.1 (iv)
Control	X11Forwarding
Control Setting	yes

Table 7. STIG Cross-References

Distribution	STIG Reference
Red Hat Enterprise Linux 7	V-72303
SuSE Linux Enterprise 12	V-217280
SuSE Linux Enterprise 15	V-235013

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^X11Forwarding"
SED_INLINE="s/^X11Forwarding.*$/X11Forwarding yes/"
ECHO_APPEND="X11Forwarding yes"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="X11Forwarding"
EXPECTED_VALUE="yes"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40087: Verify Permissions on SSH Server Private Key Files

To properly set the permissions of `/etc/ssh/*_key` , run the command:

```
chmod 0600 /etc/ssh/*_key
```

Rationale

If an unauthorized user obtains the private SSH host key file, the host could be impersonated.

NASA ASCS ID	NASA-ASCS-40087
Severity	**Critical**
Group	permissions/system

Table 8. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221859
Red Hat Enterprise Linux 7	V-72257
Red Hat Enterprise Linux 8	V-230287

Distribution	STIG Reference
SuSE Linux Enterprise 12	V-217276
SuSE Linux Enterprise 15	V-235009

bash fix

```
for KEYFILE in $(ls /etc/ssh/*_key)
do
  chmod 600 $KEYFILE
done
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet
RESULT=PASS
REASON=""

for KEYFILE in $(ls /etc/ssh/*_key)
do
  if [[ "$( stat -c "%A" $KEYFILE | grep -vE -- "-rw-------" )" ]]
  then
    RESULT=FAIL
    REASON="$REASON$KEYFILE does not have at most 600 permission. "
  fi
done

end_fixlet
```

NASA-ASCS-40088: Enable the OpenSSH Service

The SSH server service, sshd, is commonly needed. The `sshd` service can be enabled with the following command:

```
systemctl enable sshd.service
```

Rationale

Without protection of the transmitted information, confidentiality and integrity may be compromised because unprotected communications can be intercepted and either read or altered.

NASA ASCS ID	NASA-ASCS-40088
Severity	**Critical**
Group	ssh/services

Table 9. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221848
Red Hat Enterprise Linux 8	V-230526
SuSE Linux Enterprise 12	V-217264
SuSE Linux Enterprise 15	V-234860
Ubuntu 18.04 LTS	V-219313
Ubuntu 20.04 LTS	V-238215

bash fix

```
SERVICE="sshd"
if which systemctl &> /dev/null
then
  systemctl enable $SERVICE &> /dev/null
  systemctl start $SERVICE &> /dev/null
else
  service $SERVICE start &> /dev/null
  chkconfig $SERVICE on &> /dev/null
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet
RESULT=PASS
REASON=""

SERVICES=(sshd)
for SERVICE in ${SERVICES[*]}
do
  if which systemctl &> /dev/null
  then
    if [ "$( systemctl is-enabled $SERVICE )" != "enabled" ]
    then
      RESULT=FAIL
      REASON="$REASON$SERVICE are not enabled. "
    fi
    if [ "$( systemctl is-active $SERVICE )" != "active" ]
    then
      RESULT=FAIL
      REASON="$REASON$SERVICE are not active. "
    fi
  else
    CHKCONF_REGEX="0:off\s+1:off\s+2:on\s+3:on\s+4:on\s+5:on\s+6:off"

    if ! chkconfig --list $SERVICE | grep -qE "$CHKCONF_REGEX"
    then
      RESULT=FAIL
      REASON="$SERVICE is not enabled. "
    fi
    if ! service $SERVICE status &> /dev/null
    then
      RESULT=FAIL
      REASON="$REASON$SERVICE is not active. "
    fi
  fi
done

end_fixlet
```

NASA-ASCS-40089: Verify Permissions on SSH Server Public Key Files

To properly set the permissions of `/etc/ssh/*.pub` , run the command:

```
chmod 0644 /etc/ssh/*.pub
```

Rationale

If a public host key file is modified by an unauthorized user, the SSH service may be compromised.

NASA ASCS ID	NASA-ASCS-40089
Severity	**Critical**
Group	permissions/system

Table 10. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221858
Red Hat Enterprise Linux 7	V-72255
Red Hat Enterprise Linux 8	V-230286
SuSE Linux Enterprise 12	V-217275
SuSE Linux Enterprise 15	V-235008

bash fix

```
for KEYFILE in $(ls /etc/ssh/*.pub)
do
  chmod 644 $KEYFILE
done
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet
RESULT=PASS
REASON=""

for KEYFILE in $(ls /etc/ssh/*.pub)
do
  if [[ "$( stat -c "%A" $KEYFILE | grep -vE -- "-r[w\-]-[r\-]--[r\-]--" )" ]]
  then
    RESULT=FAIL
    REASON="$REASON$KEYFILE does not have at most 644 permission. "
  fi
done

end_fixlet
```

5.3. High Severity Settings

Settings that are classified as **high** severity include Category 1 settings, as defined by the Defense Information Systems Agency (DISA) as well as settings that ASCS and other ASCS-approved sources have deemed to be classified as **high** severity. These settings target any vulnerability which, if exploited, would directly and immediately result in the loss of confidentiality, integrity or availability (FIPS Publication 199, Standards for Security Categorization of Federal Information and Information Systems).

NASA-ASCS-40035: Enable SSH Warning Banner

To enable the warning banner and ensure it is consistent across the system, add or correct the following line in `/etc/ssh/sshd_config` :

```
Banner /etc/issue
```

Agency Banner Information

Rationale

The warning message reinforces policy awareness during the login process and facilitates possible legal action against attackers. Alternatively, systems whose ownership should not be obvious should ensure usage of a banner that does not provide easy attribution.

NASA ASCS ID	NASA-ASCS-40035
Severity	**High**
Group	ssh/services
Other References	NIST SP 800-53 Revision 4 AC-8 c 3, NIST SP 800-53 AC-8 b, NIST SP 800-53A AC-8.2 (ii), NIST SP 800-53 AC-8 a, NIST SP 800-53 Revision 4 AC-8 a, NIST SP 800-53A AC-8.1 (iii), NIST SP 800-53 AC-8 c, NASA-ASCS-20036, NIST SP 800-53A AC-8.2 (i), NIST SP 800-53A AC-8.1 (ii), NIST SP 800-53 Revision 4 AC-8 c 2, NIST SP 800-53A AC-8.2 (iii), NIST SP 800-53 Revision 4 AC-8 b, NIST SP 800-53 Revision 4 AC-8 c 1
Control	Banner
Control Setting	/etc/issue

Table 11. STIG Cross-References

Distribution	STIG Reference
Red Hat Enterprise Linux 7	V-72225
Red Hat Enterprise Linux 8	V-230225
SuSE Linux Enterprise 12	V-217263
SuSE Linux Enterprise 15	V-234805

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^Banner"
SED_INLINE="s/^Banner.*$/Banner \/etc\/issue/"
ECHO_APPEND="Banner /etc/issue"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="Banner"
EXPECTED_VALUE="/etc/issue"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40036: Disable Challenge Response Authentication

The challenge-response authentication method is a single factor method that is most simply a password (a server provides the challenge "Password: " for the client to return the password), but can also be used for more sophisticated schemes (CHAP, CRAM-MD5, and SCRAM).

Single factor authentication is considered weak. By default configuration, the `sshd` service will allow PAM password authentication through the challenge-response mechanism. The `ChallengeResponseAuthentication` control *SHALL* be set to `no`:

```
ChallengeResponseAuthentication  no
```

Rationale

Use strong authentication methods.

NASA ASCS ID	NASA-ASCS-40036
Severity	**High**
Group	ssh/services
Control	ChallengeResponseAuthentication
Control Setting	no

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^ChallengeResponseAuthentication"
SED_INLINE="s/^ChallengeResponseAuthentication.*$/ChallengeResponseAuthentication no/"
ECHO_APPEND="ChallengeResponseAuthentication no"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="ChallengeResponseAuthentication"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40038: Use Only Approved Ciphers

Enforce the use of strong ciphers. Counter (CTR) mode is also preferred over cipher-block chaining (CBC) mode. The following line in `/etc/ssh/sshd_config` demonstrates a limited set of ciphers:

```
Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com
```

NIST provides a list of ciphers that are acceptable for use.

On RHEL 8 the system level crypto policy needs to be disabled in order to enforce the sshd_config entry. The line `CRYPTO_POLICY=` should be set to empty in the `/etc/sysconfig/sshd` file.

Rationale

Weak ciphers have been to shown to have potential and documented exploits that can lead to system compromise or communication interception.

NASA ASCS ID	NASA-ASCS-40038
Severity	**High**
Group	ssh/services

Other References	NIST SP 800-53 AC-17 (2), NIST SP 800-53A IA-7.1, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 AC-17 (2), NIST SP 800-53 Revision 4 IA-7, NIST SP 800-53A CM-6.1 (iv), NIST SP 800-53A AC-17 (2).1, NIST SP 800-53 IA-7
Control	Ciphers
Control Setting	aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com

Table 12. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221840
Red Hat Enterprise Linux 7	V-72221
Red Hat Enterprise Linux 8	V-230252
SuSE Linux Enterprise 12	V-217270
SuSE Linux Enterprise 15	V-234816
Ubuntu 18.04 LTS	V-219312
Ubuntu 20.04 LTS	V-238217

bash fix

```
if grep -E "^(\s+)?Ciphers" /etc/ssh/sshd_config &> /dev/null
then
  sed -i "s/^\(\s\+\)\?Ciphers\s.*$/Ciphers\ aes128-ctr,aes192-ctr,aes256-ctr,aes128-
gcm@openssh.com,aes256-gcm@openssh.com/" /etc/ssh/sshd_config
else
  echo "Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com" >>
/etc/ssh/sshd_config
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

APPROVED_CIPHERS=( aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com )
CIPHERS=( $( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "ciphers" | sed "s/,/\
/g" | sed -e  "s/\s+/\ /g" | cut -d" " -f2- ) )

REASON=""
RESULT=PASS

if [ ${#CIPHERS[@]} -gt 0 ]
then
  for c in ${CIPHERS[*]}
  do
    MATCH=/bin/false
    for a in ${APPROVED_CIPHERS[*]}
    do
      if [[ "$c" == "$a" ]]
      then
        MATCH=/bin/true
      fi
    done
    if ! $MATCH
    then
      RESULT=FAIL
      REASON="${REASON}$c not in approved Ciphers, "
  fi
  done
fi

end_fixlet
```

NASA-ASCS-40039: Set SSH Client Alive Count

To ensure the SSH idle timeout occurs precisely when the `ClientAliveInterval` is set, edit `/etc/ssh/sshd_config` as follows:

```
ClientAliveCountMax 0
```

Rationale

This ensures a user login will be terminated as soon as the `ClientAliveInterval` is reached.

NASA ASCS ID	NASA-ASCS-40039
Severity	**High**

Group	ssh/services
Other References	NIST SP 800-53A SC-10.1 (ii), NIST SP 800-53 Revision 4 SC-10, NIST SP 800-53 Revision 4 AC-12, NIST SP 800-53 SC-10
Control	ClientAliveCountMax
Control Setting	0

Table 13. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221851
Red Hat Enterprise Linux 7	V-72241
Red Hat Enterprise Linux 8	V-230244
SuSE Linux Enterprise 12	V-217273
SuSE Linux Enterprise 15	V-234830
Ubuntu 18.04 LTS	V-219310
Ubuntu 20.04 LTS	V-238212

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^ClientAliveCountMax"
SED_INLINE="s/^ClientAliveCountMax.*$/ClientAliveCountMax 0/"
ECHO_APPEND="ClientAliveCountMax 0"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="ClientAliveCountMax"
EXPECTED_VALUE="0"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40040: Set SSH Idle Timeout Interval

SSH allows administrators to set an idle timeout interval. After this interval has passed, the idle user will be automatically logged out.

To set an idle timeout interval, edit the following line in `/etc/ssh/sshd_config` as follows:

```
ClientAliveInterval  900
```

The timeout interval is given in seconds. To have a timeout of 15 minutes, set interval to 900.

If a shorter timeout has already been set for the login shell, that value will preempt any SSH setting made here. Keep in mind that some processes may stop SSH from correctly detecting that the user is idle.

Rationale

Terminating an idle ssh session within a short time period reduces the window of opportunity for unauthorized personnel to take control of a management session enabled on the console or console port that has been left unattended.

NASA ASCS ID	NASA-ASCS-40040
Severity	**High**

Group	ssh/services
Other References	NIST SP 800-53A SC-10.1 (ii), NIST SP 800-53 Revision 4 SC-10, NIST SP 800-53 Revision 4 AC-12, NIST SP 800-53 SC-10
Control	ClientAliveInterval
Control Setting	900

Table 14. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221849
Red Hat Enterprise Linux 7	V-72237
SuSE Linux Enterprise 12	V-217272
SuSE Linux Enterprise 15	V-234827
Ubuntu 18.04 LTS	V-219311
Ubuntu 20.04 LTS	V-238213

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^ClientAliveInterval"
SED_INLINE="s/^ClientAliveInterval.*$/ClientAliveInterval 900/"
ECHO_APPEND="ClientAliveInterval 900"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="ClientAliveInterval"
EXPECTED_VALUE="900"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" -gt "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING $ACTUAL_VALUE exceeded $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40041: Disable GSSAPI Authentication

SSH should not permit extraneous or unnecessary authentication mechanisms like GSSAPI. To disable GSSAPI authentication, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
GSSAPIAuthentication no
```

Rationale

The GSSAPI mechanism only uses SHA-1 hashing for signatures and is susceptible to collision based attacks. Additionally, GSSAPI authentication is used to provide additional authentication mechanisms to applications. Allowing GSSAPI authentication through SSH exposes the system's GSSAPI to remote hosts, increasing the attack surface of the system.

NASA ASCS ID	NASA-ASCS-40041
Severity	**High**
Group	ssh/services
Other References	NIST SP 800-53A CM-6.1 (v), NIST SP 800-53A CM-3.1 (v), NIST SP 800-53 Revision 4 CM-3 f, NIST SP 800-53 CM-6 c, NIST SP 800-53 CM-3 e, NIST SP 800-53 Revision 4 CM-11 (2), NIST SP 800-53 Revision 4 CM-5 (1), NIST SP 800-53 Revision 4 CM-6 c

Control	GSSAPIAuthentication
Control Setting	no

Table 15. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221860
Red Hat Enterprise Linux 7	V-72259
Red Hat Enterprise Linux 8	V-230291

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^GSSAPIAuthentication"
SED_INLINE="s/^GSSAPIAuthentication.*$/GSSAPIAuthentication no/"
ECHO_APPEND="GSSAPIAuthentication no"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="GSSAPIAuthentication"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40045: Use Only Strong Host Key Algorithms

Use only strong host key algorithms. The `/etc/ssh/sshd_config HostKeyAlgorithms` control restricts host keys allowed to be used on the system.

```
HostKeyAlgorithms ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-
v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-
nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256
```

NIST maintains a list of acceptable algorithms.

Rationale

Weak host key algorithms have the potential or have been shown to be susceptible to counterfeit or collision attacks that could lead to system compromise.

NASA ASCS ID	NASA-ASCS-40045
Severity	**High**
Group	ssh/services
Control	HostKeyAlgorithms
Control Setting	ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256

bash fix

```
if grep -E "^(\s+)?HostKeyAlgorithms\s" /etc/ssh/sshd_config &> /dev/null
then
  sed -i "s/^\(\s\+\)\?HostKeyAlgorithms\s.*$/HostKeyAlgorithms\ ecdsa-sha2-nistp256-cert-
v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-
v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-
nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256/" /etc/ssh/sshd_config
else
  echo "HostKeyAlgorithms ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-
v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-
nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256" >>
/etc/ssh/sshd_config
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

APPROVED_KEYA=( ecdsa-sha2-nistp256-cert-v01@openssh.com ecdsa-sha2-nistp384-cert-v01@openssh.com
ecdsa-sha2-nistp521-cert-v01@openssh.com ssh-rsa-cert-v01@openssh.com ecdsa-sha2-nistp256 ecdsa-
sha2-nistp384 ecdsa-sha2-nistp521 ssh-rsa rsa-sha2-512 rsa-sha2-256 )
KEYA=( $( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "hostkeyalgorithms" | sed
"s/,/\ /g" | sed -e  "s/\s+/\ /g" | cut -d" " -f2- ) )

REASON=""
RESULT=PASS

if [ ${#KEYA[@]} -gt 0 ]
then
  for k in ${KEYA[*]}
  do
    MATCH=/bin/false
    for a in ${APPROVED_KEYA[*]}
    do
      if [[ "$k" == "$a" ]]
      then
        MATCH=/bin/true
      fi
    done
    if ! $MATCH
    then
      RESULT=FAIL
      REASON="${REASON}$k not in approved HostKeyAlgorithms, "
    fi
  done
fi

end_fixlet
```

NASA-ASCS-40048: Disable SSH Support for .rhosts Files

SSH can emulate the behavior of the obsolete rsh command in allowing users to enable insecure access to their accounts via `.rhosts` files.

To ensure this behavior is disabled, add or correct the following line in `/etc/ssh/sshd_config` :

```
IgnoreRhosts yes
```

Rationale

SSH trust relationships mean a compromise on one host can allow an attacker to move to other trusted hosts

without needing authentication.

NASA ASCS ID	NASA-ASCS-40048
Severity	**High**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53A CM-6.1 (iv)
Control	IgnoreRhosts
Control Setting	yes

Table 16. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221852
Red Hat Enterprise Linux 7	V-72243

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^IgnoreRhosts"
SED_INLINE="s/^IgnoreRhosts.*$/IgnoreRhosts yes/"
ECHO_APPEND="IgnoreRhosts yes"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="IgnoreRhosts"
EXPECTED_VALUE="yes"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40051: Disable Kerberos Authentication

Unless needed, SSH should not permit extraneous or unnecessary authentication mechanisms, such as Kerberos. To disable Kerberos authentication, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
KerberosAuthentication no
```

Rationale

Kerberos authentication for SSH is often implemented using GSSAPI. If Kerberos is enabled through SSH, the SSH daemon provides a means of access to the system's Kerberos implementation. Vulnerabilities in the system's Kerberos implementations may be subject to exploitation. Kerberos is rarely used for automated processes and is not recommended due to the risks of implicit access and the lack of command restrictions.

NASA ASCS ID	NASA-ASCS-40051
Severity	**High**
Group	ssh/services
Other References	NIST SP 800-53A CM-6.1 (v), NIST SP 800-53A CM-3.1 (v), NIST SP 800-53 Revision 4 CM-3 f, NIST SP 800-53 CM-6 c, NIST SP 800-53 CM-3 e, NIST SP 800-53 Revision 4 CM-11 (2), NIST SP 800-53 Revision 4 CM-5 (1), NIST SP 800-53 Revision 4 CM-6 c

Control	KerberosAuthentication
Control Setting	no

Table 17. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221861
Red Hat Enterprise Linux 7	V-72261
Red Hat Enterprise Linux 8	V-230291

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^KerberosAuthentication"
SED_INLINE="s/^KerberosAuthentication.*$/KerberosAuthentication no/"
ECHO_APPEND="KerberosAuthentication no"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="KerberosAuthentication"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40052: Use Only Strong Key Exchange Algorithms

Enforce the use of strong key exchange algorithms. The `/etc/ssh/sshd_config KexAlgorithms` control restricts use of key exchange algorithms:

```
KexAlgorithms ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512
```

Systems that are using legacy versions of OpenSSH may not have access to any of the Key Exchange Algorithms in the acceptable list, use of diffie-hellman-group-exchange-sha256 is provided by OpenSSH as a bridge to upgrade of software. A Risk Based Decision (RBD) would need to be made to allow use of this algorithm for use of older OpenSSH instances. Upgrade to modern OpenSSH is more appropriate then maintaining version of software that has known security concerns.

Rationale

Use of weak key exchange algorithms has the potential or has been shown to allow attacks on the initial connection of SSH. Compromised key exchange would allow for a man in the middle attack.

NASA ASCS ID	NASA-ASCS-40052
Severity	**High**
Group	ssh/services
Control	KexAlgorithms
Control Setting	ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512

bash fix

```
if grep -E "^(\s+)?KexAlgorithms\s" /etc/ssh/sshd_config &> /dev/null
then
  sed -i "s/^\(\s\+\)\?KexAlgorithms\s.*$/KexAlgorithms\ ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512/" /etc/ssh/sshd_config
else
  echo "KexAlgorithms ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512" >> /etc/ssh/sshd_config
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

APPROVED_KEXA=( ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group14-
sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 )
KEYA=( $( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "kexalgoritihms" | sed
"s/,/\ /g" | sed -e  "s/\s+/\ /g" | cut -d" " -f2- ) )

REASON=""
RESULT=PASS

if [ ${#MACS[@]} -gt 0 ]
then
  for k in ${KEXA[*]}
  do
    MATCH=/bin/false
    for a in ${APPROVED_KEXA[*]}
    do
      if [[ "$k" == "$a" ]]
      then
        MATCH=/bin/true
      fi
    done
    if ! $MATCH
    then
      RESULT=FAIL
      REASON="${REASON}$k not in approved KexAlgorithms, "
    fi
  done
fi

end_fixlet
```

NASA-ASCS-40054: Set Time to Disconnect During Login

`LoginGraceTime` is the time set until the server disconnects if the user has not successfully logged in. Set the value to `120` (2 minutes).

```
LoginGraceTime     120
```

Rationale

Unbound login sessions can be a vector of attack (DoS) and shall be limited. As such, user interactive login methods need to have a bounded time set before disconnecting.

NASA ASCS ID	NASA-ASCS-40054

Severity	**High**
Group	ssh/services
Control	LoginGraceTime
Control Setting	120

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^LoginGraceTime"
SED_INLINE="s/^LoginGraceTime.*$/LoginGraceTime 120/"
ECHO_APPEND="LoginGraceTime 120"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="LoginGraceTime"
EXPECTED_VALUE="120"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" -gt "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING $ACTUAL_VALUE exceeds $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40055: Set LogLevel

The OpenSSH server logs to the AUTH facility of syslog, at the INFO level by default. In order to record more information, such as failed authentication attempts, increase the logging level to `VERBOSE`.

Add or correct the following line in the `/etc/ssh/sshd_config` file:

```
LogLevel VERBOSE
```

Rationale

SSH provides several logging levels with varying amounts of verbosity. `DEBUG` is specifically not recommended other than strictly for debugging SSH communications since it provides so much data that it is difficult to identify important security information. `INFO` level is the basic level that only records login activity of SSH users. In many situations, such as Incident Response, it is important to determine when a particular user was active on a system. The logout record can eliminate those users who disconnected, which helps narrow the field. `VERBOSE` will display failed attempts to authenticate and can be used in conjunction with firewall and application rules to defend against brute force attacks.

NASA ASCS ID	NASA-ASCS-40055
Severity	**High**
Group	ssh/services
Control	LogLevel
Control Setting	VERBOSE

Table 18. STIG Cross-References

Distribution	STIG Reference
SuSE Linux Enterprise 12	V-217265
SuSE Linux Enterprise 15	V-234815

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^LogLevel"
SED_INLINE="s/^LogLevel.*$/LogLevel VERBOSE/"
ECHO_APPEND="LogLevel VERBOSE"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="LogLevel"
EXPECTED_VALUE="VERBOSE"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40056: Use Only Strong Hashing Algorithms for MACs

Ensure use of appropriate hashing algorithms are being used for MACs. The following line in `/etc/ssh/sshd_config` demonstrates allowable MAC hashing algorithms:

```
MACs hmac-sha2-256,hmac-sha2-512,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com
```

On RHEL 8 the system level crypto policy needs to be disabled in order to enforce the sshd_config entry. The line `CRYPTO_POLICY=` should be set to empty in the `/etc/sysconfig/sshd` file.

Rationale

The Message Authentication Code (MAC) is used to ensure packet integrity, use of weak hashing algorithms can compromise messages using collisions techniques, preventing the integrity of a packet from being ensured.

NASA ASCS ID	NASA-ASCS-40056
Severity	**High**
Group	ssh/services
Other References	NIST SP 800-53A AC-17 (2).1, NIST SP 800-53 AC-17 (2), NIST SP 800-53 Revision 4 AC-17 (2)

Control	MACs
Control Setting	hmac-sha2-256,hmac-sha2-512,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com

Table 19. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221857
Red Hat Enterprise Linux 7	V-72253
Red Hat Enterprise Linux 8	V-230251
SuSE Linux Enterprise 12	V-217271
SuSE Linux Enterprise 15	V-234826
Ubuntu 20.04 LTS	V-238216

bash fix

```
if grep -E "^(\s+)?MACs\s" /etc/ssh/sshd_config &> /dev/null
then
  sed -i "s/^\(\s\+\)\?MACs\s.*$/MACs\ hmac-sha2-256,hmac-sha2-512,hmac-sha2-256-
etm@openssh.com,hmac-sha2-512-etm@openssh.com/" /etc/ssh/sshd_config
else
  echo "MACs hmac-sha2-256,hmac-sha2-512,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com" >> /etc/ssh/sshd_config
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

APPROVED_MACS=( hmac-sha2-256 hmac-sha2-512 hmac-sha2-256-etm@openssh.com hmac-sha2-512-
etm@openssh.com )
MACS=( $( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "macs" | sed "s/,/\ /g" |
sed -e  "s/\s+/\ /g" | cut -d" " -f2- ) )

REASON=""
RESULT=PASS

if [ ${#MACS[@]} -gt 0 ]
then
  for m in ${MACS[*]}
  do
    MATCH=/bin/false
    for a in ${APPROVED_MACS[*]}
    do
      if [[ "$m" == "$a" ]]
      then
        MATCH=/bin/true
      fi
    done
    if ! $MATCH
    then
      RESULT=FAIL
      REASON="${REASON}$m not in approved MACs, "
    fi
  done
fi

end_fixlet
```

NASA-ASCS-40059: Set Max Unauthenticated Concurrent Sessions

Specifies the maximum number of concurrent, unauthenticated connections to the SSH daemon. Additional connections will be dropped until authentication succeeds or the `LoginGraceTime` expires for a connection.

```
MaxStartups     10:30:100
```

Rationale

Restrict concurrent, unauthenticated connections.

NASA ASCS ID	NASA-ASCS-40059

Severity	**High**
Group	ssh/services
Control	MaxStartups
Control Setting	10:30:100

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^MaxStartups"
SED_INLINE="s/^MaxStartups.*$/MaxStartups 10:30:100/"
ECHO_APPEND="MaxStartups 10:30:100"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="MaxStartups"
EXPECTED_VALUE="10:30:100"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40060: Disable Password Authentication

Password authentication uses a single authentication factor; as such, it is considered a weak authentication method. The Agency requires use of strong authentication methods. The `PasswordAuthentication` control specifies whether password authentication is allowed.

```
PasswordAuthentication     no
```

Rationale

Use of smartcard authentication is mandatory at the Agency.

NASA ASCS ID	NASA-ASCS-40060
Severity	**High**
Group	ssh/services
Control	PasswordAuthentication
Control Setting	no

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^PasswordAuthentication"
SED_INLINE="s/^PasswordAuthentication.*$/PasswordAuthentication no/"
ECHO_APPEND="PasswordAuthentication no"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="PasswordAuthentication"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40062: Disable SSH Root Login

The root user *SHALL NOT* be allowed direct SSH access to a system over a network. To disable root access via SSH, add or correct the following line in `/etc/ssh/sshd_config` :

```
PermitRootLogin no
```

Rationale

Even though the communications channel may be encrypted, an additional layer of security is gained by extending the policy to prevent access directly as root. In addition, accessing the system with a user-specific account provides individual accountability for actions performed and helps to minimize direct attack attempts on root's password.

NASA ASCS ID	NASA-ASCS-40062
Severity	**High**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53A CM-6.1 (iv)
Control	PermitRootLogin
Control Setting	no

Table 20. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221854
Red Hat Enterprise Linux 7	V-72247
Red Hat Enterprise Linux 8	V-230296
SuSE Linux Enterprise 12	V-217267
SuSE Linux Enterprise 15	V-234870

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^PermitRootLogin"
SED_INLINE="s/^PermitRootLogin.*$/PermitRootLogin no/"
ECHO_APPEND="PermitRootLogin no"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="PermitRootLogin"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40065: Set the Standard Network Port

TCP Port 22 is the IANA standard port for SSH. The `Port` setting is used to specify the TCP port that sshd listens on and *SHALL* be set to `22` or unset.

```
Port    22
```

Rationale

By requiring port `22` use for SSH, the Agency can better monitor legitimate traffic and identify potential malicious behavior on the network.

NASA ASCS ID	NASA-ASCS-40065
Severity	**High**
Group	ssh/services
Other References	NIST IR 7966, NASA SOC-SAR-2019-0014
Control	Port
Control Setting	22

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^Port"
SED_INLINE="s/^Port.*$/Port 22/"
ECHO_APPEND="Port 22"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="Port"
EXPECTED_VALUE="22"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "^$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40069: Use Only Strong Public Key Types

Ensure strong public key types are enforced. The `/etc/ssh/sshd_config PubkeyAcceptedTypes` control restricts use of key types:

```
PubkeyAcceptedKeyTypes ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-
v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-
nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256
```

Rationale

The public key is used after the Key Exchange to authenticate the connecting party, with use of weak public keys there is a potential to be able to determine the private key.

NASA ASCS ID	NASA-ASCS-40069
Severity	**High**
Group	ssh/services
Control	PubkeyAcceptedKeyTypes

Control Setting	ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256

bash fix

```
if grep -E "^(\s+)?PubkeyAcceptedKeyTypes\s" /etc/ssh/sshd_config &> /dev/null
then
  sed -i "s/^\(\s\+\)\?PubkeyAcceptedKeyTypes\s.*$/PubkeyAcceptedKeyTypes\ ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256/" /etc/ssh/sshd_config
else
  echo "PubkeyAcceptedKeyTypes ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256" >> /etc/ssh/sshd_config
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

SETTING=""
APPROVED_PKEYS=( ecdsa-sha2-nistp256-cert-v01@openssh.com ecdsa-sha2-nistp384-cert-v01@openssh.com
ecdsa-sha2-nistp521-cert-v01@openssh.com ssh-rsa-cert-v01@openssh.com ecdsa-sha2-nistp256 ecdsa-
sha2-nistp384 ecdsa-sha2-nistp521 ssh-rsa rsa-sha2-512 rsa-sha2-256 )
PKEYS=( $( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "pubkeyacceptedkeytypes"
| sed "s/,/\ /g" | sed -e  "s/\s+/\ /g" | cut -d" " -f2- ) )

RESULT=PASS
REASON=""

if [ ${#PKEYS[@]} -gt 0 ]
then
  for k in ${PKEYS[*]}
  do
    MATCH=/bin/false
    for a in ${APPROVED_PKEYS[*]}
    do
      if [[ "$k" == "$a" ]]
      then
        MATCH=/bin/true
      fi
    done
    if ! $MATCH
    then
      RESULT=FAIL
      REASON="${REASON}$k not in approved PubkeyAcceptedKeyTypes, "
    fi
  done
fi

end_fixlet
```

NASA-ASCS-40070: Enabled Public Key Authentication

Public key authentication is based on the use of public key cryptography. The `PubkeyAuthentication` control specifies whether public key authentication is allowed.

```
PubkeyAuthentication   yes
```

Rationale

Public key authentication is a strong authentication method.

NASA ASCS ID	NASA-ASCS-40070
Severity	**High**
Group	ssh/services
Control	PubkeyAuthentication
Control Setting	yes

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^PubkeyAuthentication"
SED_INLINE="s/^PubkeyAuthentication.*$/PubkeyAuthentication yes/"
ECHO_APPEND="PubkeyAuthentication yes"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="PubkeyAuthentication"
EXPECTED_VALUE="yes"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40075: Enable Use of Strict Modes Checking

SSHs StrictModes option checks file and ownership permissions in the user's home directory `.ssh` folder before accepting login. If world-writable permissions are found, login is rejected. To enable StrictModes in SSH, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
StrictModes yes
```

Rationale

If other users have access to modify user-specific SSH configuration files, they may be able to log into the system as another user.

NASA ASCS ID	NASA-ASCS-40075
Severity	**High**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53A CM-6.1 (iv)
Control	StrictModes
Control Setting	yes

Table 21. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221862
Red Hat Enterprise Linux 7	V-72263
Red Hat Enterprise Linux 8	V-230288
SuSE Linux Enterprise 12	V-217277
SuSE Linux Enterprise 15	V-235010

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^StrictModes"
SED_INLINE="s/^StrictModes.*$/StrictModes yes/"
ECHO_APPEND="StrictModes yes"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="StrictModes"
EXPECTED_VALUE="yes"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40076: Set Logging on Subsystem for Secure File Transfer

The sshd provides the `Subsystem` declaration to provide additional extension to the SSH mechanism. In order to provide secure file transfer, sshd utilizes the sftp application. Configure the subsystem for file transfer with appropriate flags for logging.

The location of the sftp-server may differ based on the system used. The sftp-server can be found in multiple ways. The package management system or `mlocate` tool can be used. See Fix Text for an example.

Red Hat Enterprise Linux Example:

```
Subsystem    sftp /usr/libexec/openssh/sftp-server -f AUTHPRIV -l INFO
```

This control is dependent on the rsyslog facility, the use of the `AUTH` or `AUTHPRIV` corresponds with the rsyslog configuration to identify the resultant log file inclusion.

It is the responsibility of the System Owner to ensure that the proper entries into the logs are able to be monitored. For ensuring that the log entries go to the /var/log/secure location, make sure that the sshd_config SyslogFacility entry for `AUTH` or `AUTHPRIV` corresponds with the rsyslog.conf entry `auth. or authpriv.`.

As an example, the default RHEL7 implementation of rsyslog will have `*.info` and `authpriv.none` for /var/log/messages stated in the /etc/rsyslog.conf configuration. This ensures that no entry for `AUTHPRIV` will be in the /var/log/messages file, but any `INFO`-related entry will be. As with the sshd_config `SyslogFacility`, if the value is set to `AUTH`, then this will result in any auth.info messages ending up in /var/log/messages.

Rationale

Logging to `AUTHPRIV` facility makes it easier for administrators to see relevant errors and successes. Use of the `AUTHPRIV` facility will also assist log aggregation tools.

NASA ASCS ID	NASA-ASCS-40076
Severity	**High**
Group	ssh/services
Control	Subsystem
Control Setting	sftp /usr/libexec/openssh/sftp-server -f AUTHPRIV -l INFO

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^Subsystem"

if which yum &> /dev/null
then
  SFTP_LIBS=$(rpm -q $(rpm -q --whatprovides $(which sshd)) --dump | grep sftp-server | grep -Ev
"(man|doc|\.build-id)" | awk '{print $1}' | sed "s/\//\\\\\//g" )
elif which apt &> /dev/null
then
  SFTP_LIBS=$( dpkg-query -S sftp-server | grep -Ev "(man|doc|\.build-id)" | grep sftp-server | awk
'{print $2}' | sed "s/\//\\\\\//g" )
elif which zypper &> /dev/null
then
  SFTP_LIBS=$(rpm -q $(rpm -q --whatprovides $(which sshd)) --dump | grep sftp-server | grep -Ev
"(man|doc|\.build-id)" | awk '{print $1}' | sed "s/\//\\\\\//g" )
fi

for SFTP_LIB in ${SFTP_LIBS[*]}
do
  if [ ! -h $SFTP_LIB ]
  then
    SED_INLINE="s/^Subsystem.*$/Subsystem sftp $SFTP_LIB -f AUTH -l INFO/"
    ECHO_APPEND="Subsystem sftp $SFTP_LIB -f AUTH -l INFO"
    if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
    then
      sed -i "$SED_INLINE" $CONFIG_FILE
    else
      echo $ECHO_APPEND >> $CONFIG_FILE
    fi
  fi
done
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=FAIL
REASON="sftp library server not configured"

if which yum &> /dev/null
then
  SFTP_LIBS=$(rpm -q $(rpm -q --whatprovides $(which sshd)) --dump | grep sftp-server | grep -Ev
"(man|doc|\.build-id)" | awk '{print $1}')
elif which apt &> /dev/null
then
  SFTP_LIBS=$( dpkg-query -S sftp-server | grep -Ev "(man|doc|\.build-id)" | grep sftp-server | awk
'{print $2}' )
elif which zypper &> /dev/null
```

```
then
  SFTP_LIBS=$(rpm -q $(rpm -q --whatprovides $(which sshd)) --dump | grep sftp-server | grep -Ev
"(man|doc|\.build-id)" | awk '{print $1}')
else
  RESULT=FAIL
  REASON="Could not determine package manager"
fi

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "subsystem" | cut
-d" " -f2- )"

SFTP_LIB=""
for SFTP_VAL in ${SFTP_LIBS[*]}
do
  if [[ "$(echo $ACTUAL_VALUE | grep $SFTP_VAL )" ]]
  then
    RESULT=PASS
    REASON=""
    SFTP_LIB=$SFTP_VAL
  fi
done

if [[ "$SFTP_LIB" ]]
then
  ELEMENTS=( "sftp" $SFTP_LIB "-f (AUTH|AUTHPRIV)" "-l (INFO|VERBOSE)" )

  MATCH=/bin/false
  for e in ${ELEMENTS[*]}
  do
    if $( echo -- $ACTUAL_VALUE | grep -Eq -- "$e"  )
    then
      MATCH=/bin/true
    else
      RESULT=FAIL
      REASON="${REASON}Subsystem element $e is missing, "
    fi
  done
fi

end_fixlet
```

NASA-ASCS-40078: Set Syslog Facility

The sshd service uses the host logging facility, the `SyslogFacility` dictates the destination logging to use. The `AUTH` or `AUTHPRIV` facility will result in logging to the host authentication logs (i.e. `/var/log/secure`).

```
SyslogFacility    AUTHPRIV
```

This control is dependent on the rsyslog facility. The use of the `AUTH` or `AUTHPRIV` corresponds with the rsyslog configuration to identify the resultant log file inclusion.

Ensuring that the proper entries into the logs are able to be monitored is the responsibility of the System Owner. For ensuring that the log entries go to the /var/log/secure location, make sure that the sshd_config SyslogFacility entry for `AUTH` or `AUTHPRIV` corresponds with the rsyslog.conf entry `auth. or authpriv.`.

As an example, the default RHEL7 implementation of rsyslog will have `*.info` and `authpriv.none` for /var/log/messages stated in the /etc/rsyslog.conf configuration. This ensures that no entry for `AUTHPRIV` will be in the /var/log/messages file, but any `INFO`-related entry will be. If the sshd_config SyslogFacility value is set to `AUTH`, then this will result in any auth.info messages ending up in /var/log/messages.

Rationale

Logging to `AUTHPRIV` facility makes it easier for administrators to see relevant errors and successes. Use of the `AUTHPRIV` facility will also assist log aggregation tools.

NASA ASCS ID	NASA-ASCS-40078
Severity	**High**
Group	ssh/services
Control	SyslogFacility
Control Setting	AUTH or AUTHPRIV

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^SyslogFacility"
SED_INLINE="s/^SyslogFacility.*$/SyslogFacility AUTHPRIV/"
ECHO_APPEND="SyslogFacility AUTHPRIV"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="SyslogFacility"
EXPECTED_REGEX="^(AUTH|AUTHPRIV)$"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "^$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif ! [[ $ACTUAL_VALUE =~ $EXPECTED_REGEX ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not AUTH or AUTHPRIV"
fi

end_fixlet
```

NASA-ASCS-40079: Enable TCP Keep Alive

Specifies whether the system should send TCP keepalive messages to the other side. If they are sent, death of the connection or crash of one of the machines will be properly noticed.

```
TCPKeepAlive     yes
```

Rationale

Maintaining keepalive messages provides the server with assurance of persistent connection to endpoint. The server will notice if the network goes down or the client host crashes. This avoids infinitely hanging sessions.

NASA ASCS ID	NASA-ASCS-40079
Severity	**High**
Group	ssh/services
Control	TCPKeepAlive
Control Setting	yes

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^TCPKeepAlive"
SED_INLINE="s/^TCPKeepAlive.*$/TCPKeepAlive yes/"
ECHO_APPEND="TCPKeepAlive yes"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="TCPKeepAlive"
EXPECTED_VALUE="yes"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40081: Disable Login Usage

The sshd service can pass the post login environment control to the host system. The host system utilized different controls for local connected users then sshd does. The sshd service is more commonly better equipped to handle the specialized remote connection.

`UseLogin` has been deprecated since version 7.4p1, adding the setting will cause `sshd` to generate unwanted log messages.

Setting in OpenSSH Version prior to 7.4p1

```
UseLogin     no
```

Rationale

Post login features are best handled by sshd and not handed over to other operating system services.

NASA ASCS ID	NASA-ASCS-40081
Severity	**High**
Group	ssh/services
Control	UseLogin
Control Setting	no

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^UseLogin"
SED_INLINE="s/^UseLogin.*$/UseLogin no/"
ECHO_APPEND="UseLogin no"

if [[ "$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep uselogin )" != "" ]]
then
  if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
  then
    sed -i "$SED_INLINE" $CONFIG_FILE
  else
    echo $ECHO_APPEND >> $CONFIG_FILE
  fi
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="UseLogin"
EXPECTED_VALUE="no"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40086: Install the OpenSSH Server Package

The `openssh-server` package should be installed. The `openssh-server` package can be installed with the following command:

```
yum install openssh-server
```

or

```
apt-get install openssh-server
```

or

```
zypper install openssh
```

Rationale

Without protection of the transmitted information, confidentiality and integrity may be compromised because unprotected communications can be intercepted and either read or altered.

NASA ASCS ID	NASA-ASCS-40086
Severity	**High**
Group	ssh/services

bash fix

```
if which yum &> /dev/null
then
  if ! rpm -q openssh-server &> /dev/null
  then
    yum -y install openssh-server
  fi
elif which apt &> /dev/null
then
  if [[ "$( dpkg-query -W -f '${db:Status-Status}' openssh-server )" != "installed" ]]
  then
    apt-get -y install openssh-server
  fi
elif which zypper &> /dev/null
then
  if ! rpm -q openssh &> /dev/null
  then
    zypper --non-interactive install openssh
  fi
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet
RESULT=PASS
REASON=""

if which yum &> /dev/null
then
  if ! rpm -q openssh-server &> /dev/null
  then
    RESULT=FAIL
    REASON="openssh-server not found on system"
  fi
elif which apt &> /dev/null
then
  if [[ "$( dpkg-query -W -f '${db:Status-Status}' openssh-server )" != "installed" ]]
  then
    RESULT=FAIL
    REASON="openssh-server not found on system"
  fi
elif which zypper &> /dev/null
then
  if ! rpm -q openssh &> /dev/null
  then
    RESULT=FAIL
    REASON="openssh not found on system"
  fi
else
  RESULT=FAIL
  REASON="no package manager to query"
fi

end_fixlet
```

NASA-ASCS-40093: Configure SSH to use System Crypto Policy

Crypto policies provide centralized control over crypto algorithms used by many packages. SSH is supported by crypto policy, but the SSH configuration may be set up to ignore it. To check that crypto policies settings are configured correctly, ensure that the `CRYPTO_POLICY` variable is either commented or not set at all in the `/etc/sysconfig/sshd`.

Rationale

Overriding the system crypto policy makes the behavior of the SSH service violate expectations, and makes system configuration more fragmented.

NASA ASCS ID	NASA-ASCS-40093
Severity	**High**

Group	software/system

bash fix

```
if [ -e /etc/sysconfig ]
then
  if grep -E "^(\s+)?CRYPTO_POLICY" /etc/sysconfig/sshd 2> /dev/null
  then
    sed -iE "s/^(\s+)?CRYPTO_POLICY/#CRYPTO_POLICY/" /etc/sysconfig/sshd
  fi
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
if [ -f /etc/sysconfig/sshd ]
then
  CHECK=$(grep -i "CRYPTO_POLICY" "/etc/sysconfig/sshd" 2>/dev/null | grep -v "^#" | awk '{print
$2}')
  if [ "$CHECK" ]
  then
    RESULT=FAIL
    REASON="CRYPTO_POLICY value set"
  fi
fi

end_fixlet
```

NASA-ASCS-40096: Ensure OpenSSH is built with the FIPS Object Model

Federal Information Processing Standard (FIPS) 140-2 requires use of a CMVP validated cryptographic module to be available on a system. The OpenSSH project is built using OpenSSL, which provides a validated Object Model that can be built into the distribution’s cryptographic software. This Object Model is identified by a label provided by the binary of either "fips" or "FIPS". In many cases, a vendor will provide a "private label" validation as well.

To check if a system has access to the FIPS Object Model provided by the OpenSSH project:

```
ssh -V

# On RHEL8.1, it should return something like: OpenSSH_8.0p1, OpenSSL 1.1.1c FIPS  28 May 2019
```

If the FIPS Object Model has been provided by the distribution vendor, the "FIPS" or "fips" designation will be returned along with the version of the software.

Rationale

Federal Information Processing Standard (FIPS) requirements on all government agencies specifies the need for use of the Cryptographic Module Validation Program (CMVP) validated cryptographic module.

NASA ASCS ID	NASA-ASCS-40096
Severity	**High**
Group	ssh/services

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet
REASON=""
RESULT=PASS

if ! [ "$( ssh -V 2>&1 | grep -Ei "fips" )" ]
then
  RESULT=FAIL
  REASON="OpenSSH is not built with FIPS 140-2 Object Model for OpenSSL"
fi

end_fixlet
```

5.4. Low Severity Settings

Settings that are classified as **low** severity are deemed highly beneficial by ASCS. These settings are expected to be implemented, unless there is a justifiable cause not to. These settings target any vulnerability which degrades measures to protect against the loss of confidentiality, integrity or availability (FIPS Publication 199, Standards for Security Categorization of Federal Information and Information Systems).

Electing not to implement these settings is not scored as deviation from compliance.

NASA-ASCS-40031: Set Address Family for IPv4 and IPv6

Specifies which address family should be used by sshd. Valid arguments are "any", "inet" (use IPv4 only), or "inet6" (use IPv6 only).

```
AddressFamily any
```

Rationale

Mandatory dual-stack IPv4 and IPv6.

NASA ASCS ID	NASA-ASCS-40031
Severity	**Low**

Group	ssh/services
Control	AddressFamily
Control Setting	any

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^AddressFamily"
SED_INLINE="s/^AddressFamily.*$/AddressFamily any/"
ECHO_APPEND="AddressFamily any"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="AddressFamily"
EXPECTED_VALUE="any"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40034: Set Authentication Methods to publickey

Specifies the authentication methods that must be completed in order to grant access to a user. Enforcing `publickey` will result in requiring the use of publickey as the only method of authentication allowed. This will prevent the ability

to use passwords or other methods of authentication.

```
AuthenticationMethods    publickey
```

Rationale

Additional enforcement of using stronger methods for authentication.

NASA ASCS ID	NASA-ASCS-40034
Severity	**Low**
Group	ssh/services
Control	AuthenticationMethods
Control Setting	publickey

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^AuthenticationMethods"
SED_INLINE="s/^AuthenticationMethods.*$/AuthenticationMethods publickey/"
ECHO_APPEND="AuthenticationMethods publickey"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="AuthenticationMethods"
EXPECTED_VALUE="publickey"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40050: Set Type of Service and DSCP on IP Header

Specifies the IPv4 type-of-service or DSCP class for the connection.

```
IPQoS        lowdelay throughput
```

Rationale

Recommended for systems that have non-standard operating system defaults. The sshd default is acceptable.

NASA ASCS ID	NASA-ASCS-40050
Severity	**Low**
Group	ssh/services
Control	IPQoS
Control Setting	lowdelay throughput

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^IPQoS"
SED_INLINE="s/^IPQoS.*$/IPQoS lowdelay throughput/"
ECHO_APPEND="IPQoS lowdelay throughput"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="IPQoS"
EXPECTED_VALUE="lowdelay throughput"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40066: Enable Printing of Last Log

When enabled, SSH will display the date and time of the last successful account logon. To enable LastLog in SSH, add or correct the following line in the `/etc/ssh/sshd_config` file:

```
PrintLastLog yes
```

Rationale

Providing users feedback as to when account accesses last occurred facilitates user recognition and reporting of unauthorized account use.

NASA ASCS ID	NASA-ASCS-40066
Severity	**Low**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53A CM-6.1 (iv)
Control	PrintLastLog
Control Setting	yes

Table 22. STIG Cross-References

Distribution	STIG Reference
Oracle Linux 7	V-221853
Red Hat Enterprise Linux 7	V-72245
Red Hat Enterprise Linux 8	V-230382
SuSE Linux Enterprise 12	V-217266
SuSE Linux Enterprise 15	V-234881

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^PrintLastLog"
SED_INLINE="s/^PrintLastLog.*$/PrintLastLog yes/"
ECHO_APPEND="PrintLastLog yes"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="PrintLastLog"
EXPECTED_VALUE="yes"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40080: Enable DNS Lookup for Client Connections

Specifies whether sshd should look up the remote host name and check that the resolved host name for the remote IP address maps back to the very same IP address.

```
UseDNS    yes
```

Rationale

Clients attempting to connect should always have a DNS entry.

NASA ASCS ID	NASA-ASCS-40080
Severity	**Low**
Group	ssh/services
Control	UseDNS
Control Setting	yes

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^UseDNS"
SED_INLINE="s/^UseDNS.*$/UseDNS yes/"
ECHO_APPEND="UseDNS yes"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="UseDNS"
EXPECTED_VALUE="yes"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40090: Enable SSH Server firewalld Firewall exception

By default, inbound connections to SSH's port are allowed. If the SSH server is being used but denied by the firewall, this exception should be added to the firewall configuration.

To configure `firewalld` to allow access, run the following command:

```
firewall-cmd --permanent --add-service=ssh
```

Rationale

If inbound SSH connections are expected, adding a firewall rule exception will allow remote access through the SSH port.

NASA ASCS ID	NASA-ASCS-40090
Severity	**Low**
Group	ssh/services
Control	sshd_listening_port
Control Setting	22

bash fix

```
if which firewall-cmd &> /dev/null
 then
   firewall-cmd --permanent --add-service=ssh
   firewall-cmd --reload
 else
   iptables -A INPUT -p tcp --dport 22 -j ACCEPT
   service iptables save
 fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet
RESULT=PASS
REASON=""

if which firewall-cmd &> /dev/null
then
  if ! firewall-cmd --list-services | grep -E "\bssh\b" &> /dev/null
  then
    RESULT=FAIL
    REASON="port 22 is not open in firewall-cmd"
  fi
else
  if ! iptables -L INPUT | grep -E "^ACCEPT.*tcp.*ssh" &> /dev/null
  then
    RESULT=FAIL
    REASON="port 22 is not open in iptables"
  fi
fi

end_fixlet
```

NASA-ASCS-40091: Disable SSH Support for User Known Hosts

Specifies whether sshd should ignore the user's ~/.ssh/known_hosts during `HostbasedAuthentication`.

This should be disabled.

To ensure this behavior is disabled, add or correct the following line in `/etc/ssh/sshd_config` :

```
IgnoreUserKnownHosts yes
```

Rationale

SSH can allow a user authentication by using a cached remote system's public keys. Configuring this setting for the SSH daemon provides additional assurance that SSH will require a user-supplied authentication. This control is recommended, along with the required control to disable `HostbasedAuthentication`.

NASA ASCS ID	NASA-ASCS-40091
Severity	**Low**
Group	ssh/services
Other References	NIST SP 800-53 CM-6 b, NIST SP 800-53 Revision 4 CM-6 b, NIST SP 800-53A CM-6.1 (iv)
Control	IgnoreUserKnownHosts
Control Setting	yes

Table 23. STIG Cross-References

Distribution	STIG Reference
Red Hat Enterprise Linux 7	V-72249

bash fix

```
CONFIG_FILE=/etc/ssh/sshd_config
EXISTS_IF_REGEX="^IgnoreUserKnownHosts"
SED_INLINE="s/^IgnoreUserKnownHosts.*$/IgnoreUserKnownHosts yes/"
ECHO_APPEND="IgnoreUserKnownHosts yes"

if [[ $( grep -E $EXISTS_IF_REGEX $CONFIG_FILE ) ]]
then
  sed -i "$SED_INLINE" $CONFIG_FILE
else
  echo $ECHO_APPEND >> $CONFIG_FILE
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

RESULT=PASS
REASON=""
SETTING="IgnoreUserKnownHosts"
EXPECTED_VALUE="yes"

ACTUAL_VALUE="$( sshd -T -C user=root -C host=localhost -C addr=localhost | grep "$( echo $SETTING
| tr '[:upper:]' '[:lower:]' )" | awk '{print $2}')"

if ! [[ "$ACTUAL_VALUE" ]]
then
  RESULT=PASS
  REASON="current version of sshd does not have this setting"
elif [[ "$ACTUAL_VALUE" != "$EXPECTED_VALUE" ]]
then
  RESULT=FAIL
  REASON="$SETTING set to $ACTUAL_VALUE not $EXPECTED_VALUE"
fi

end_fixlet
```

NASA-ASCS-40095: Use Only Strong Hostbased Accepted Key Types

Specifies the host key types accepted by the server for hostbased authentication.

```
HostbasedAcceptedKeyTypes ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-
v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-
nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256
```

Rationale

HostbasedAuthentication should be disabled. However, in cases where HostbasedAuthentication is being used, it is important to ensure that the correct Host Key Types are used for each server. Proper Host Key Types provide trust for known host keys in order to prevent man-in-the-middle attacks.

NASA ASCS ID	NASA-ASCS-40095
Severity	**Low**
Group	ssh/services
Control	HostbasedAcceptedKeyTypes

Control Setting	ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256

bash fix

```
if grep -E "^(\s+)?\s" /etc/ssh/sshd_config
then
  sed -i "s/^\(\s\+\)\?HostbasedAcceptedKeyTypes\s.*$/HostbasedAcceptedKeyTypes\ ecdsa-sha2-
nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-
v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-
nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256/" /etc/ssh/sshd_config
else
  echo "HostbasedAcceptedKeyTypes ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-
cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ecdsa-
sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,rsa-sha2-512,rsa-sha2-256" >>
/etc/ssh/sshd_config
fi
```

CDM check

```
#!/bin/bash
. ./util/xccdflib
init_fixlet

APPROVED_KEYA=( ecdsa-sha2-nistp256-cert-v01@openssh.com ecdsa-sha2-nistp384-cert-v01@openssh.com
ecdsa-sha2-nistp521-cert-v01@openssh.com ssh-rsa-cert-v01@openssh.com ecdsa-sha2-nistp256 ecdsa-
sha2-nistp384 ecdsa-sha2-nistp521 ssh-rsa rsa-sha2-512 rsa-sha2-256 )
KEYA=( $( sshd -T -C user=root -C host=localhost -C addr=localhost | grep
"hostbasedacceptedkeytypes" | sed "s/,/\ /g" | sed -e  "s/\s+/\ /g" | cut -d" " -f2- ) )

REASON=""
RESULT=PASS

if [ ${#KEYA[@]} -gt 0 ]
then
  for k in ${KEYA[*]}
  do
    MATCH=/bin/false
    for a in ${APPROVED_KEYA[*]}
    do
      if [[ "$k" == "$a" ]]
      then
        MATCH=/bin/true
      fi
    done
    if ! $MATCH
    then
      RESULT=FAIL
      REASON="${REASON}$k not in approved HostbasedAcceptedKeyTypes, "
    fi
  done
fi

end_fixlet
```

APPENDIX A: SPECIFIC OPERATION GUIDANCE

setroubleshoot

Some systems owners have found that performance issues have been found when the CDM checks have been run on a system. This performance hit can be troublesome. The CDM events in question are cause by the BigFix client being confined properly in its context, but calling upon other utilities, such as `auditctl` and `sshd`, to determine system settings. A feature in BigFix captures STDOUT and STDERR into a file denoted <ASCS ID>.detect.log, which is a context of `var_t` while the context of the calling resource is not. SELinux properly prevents the action from writing into the file, however due to SELinux auditing a record of the event, even without data, is recorded.

We have found that in the case that a system is running the `setroubleshoot` utility, that additional log entries from the audit log are posted into the system logs. This behavior of `setroubleshoot` can cause additional processing issues on some systems. `setroubleshoot` is loaded on systems that are running a graphical desktop and the AVC utility. The chain of events from the BigFix client being monitored by SELinux, an audit record being recorded, `setroubleshoot` analyzing the records via `audispd`, posting an alert to AVC, AVC posting an alert to the desktop environment, can cause a noticeable toll on resources.

Mitigation of this resource consumption currently is to either place a SELinux `donotaudit` entry into the system to quite the messages being generated, or to remove `setroubleshoot` from the system.

To quiet the auditing of SELinux on the specific BigFix controls

```
cat > bigfix_quiet.te << END_OF_FILE
module bigfix_quiet 1.0;

require {
        type iptables_t;
        type sshd_t;
        type var_t;
        type initrc_t;
        type auditctl_t;
        class file write;
}

#============= auditctl_t ==============

#!!!! WARNING: 'var_t' is a base type.
dontaudit auditctl_t var_t:file write;

#============= iptables_t ==============

#!!!! WARNING: 'var_t' is a base type.
dontaudit iptables_t var_t:file write;

#============= sshd_t ==============

#!!!! WARNING: 'var_t' is a base type.
dontaudit sshd_t var_t:file write;
END_OF_FILE

checkmodule -M -m -o bigfix_quiet.mod bigfix_quiet.te
semodule_package -o bigfix_quiet.pp -m bigfix_quiet.mod
semodule -i bigfix_quiet.pp
```

APPENDIX B: ADDITIONAL GUIDANCE

Secondary Authentication

In some systems there is an increase in authentication beyond what standard systems require. In these cases the use of a secondary authentication can provide an additional layer of protection from credentials that may have been compromised.

Using the `AuthenticationMethods` establishes the required methods for access to the system. A value of `publickey,publickey` would require two specific public key values, such as one on the PIV card and a secondary public key from the local machine.

Availability Considerations

On systems that require high availability for connection, and/or have a high number of users for the resources available to the system, measures should be considered for limiting the users capacity to overuse connections or resources.

The `MaxSessions` value set to `1` to `4` can provide protection for users consuming more then 1 to 4 concurrent sessions on a single connection.