Automated Linux Secure Host Baseline for Real-Time Applications Requiring SCAP Compliance

Chapter · September 2020

DOI: 10.1007/978-3-030-58703-1_10

4 authors, including:

Zack Kirkendoll
University of Tulsa
7 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Nathan Hutchins
University of Tulsa
8 PUBLICATIONS **64** CITATIONS
SEE PROFILE

Loyd Reed Hook
University of Tulsa
38 PUBLICATIONS **214** CITATIONS
SEE PROFILE

Automated Linux Secure Host Baseline for Real-time Applications Requiring SCAP Compliance

Zack Kirkendoll[1], Matthew Lueck[1], Nathan Hutchins[1], Loyd Hook[1]

[1] University of Tulsa, Tulsa OK 74104, USA
zack-kirkendoll@utulsa.edu

Abstract. With the emergence of Linux real-time applications and the ever-growing cyber security requirements being imposed on those systems in high security environments, a need has developed for an automated and quantitative approach for developing a solution. The development and mandate for a Department of Defense Windows Secure Host Baseline has provided a framework for how to approach an equivalent Linux Secure Host Baseline. However, a method for automating a cyber security compliant Linux distribution with real-time capability has not yet been created.

Utilizing National Institute of Standards and Technology (NIST) approved tools and security guidelines (DISA STIG), this work has produced an automated Linux distribution capable of performing real-time functions with a 96.15% security compliance performance metric. The compliance deficiencies have been addressed to a satisfactory level based on their context. The promising results show that this approach can be utilized by other organizations for Linux real-time application development and deployment with customized environments specific to their needs while maintaining a high level of security posture. The methodology proposed provides a template for designing, customizing, and maintaining the solution.

Keywords: Security, Real-time, Linux, SCAP, STIG, DISA, CentOS.

1 Introduction

Industrial, aviation, and safety-critical applications often require hard guarantees for system response times. These real-time applications require operational timing constraints between triggering events and the applications response to that event. This determinism of meeting schedule deadlines is accomplished through a real-time operating system (RTOS). The latency of a task depends on other tasks running at equal or higher priorities.

The Linux Operating System (OS) has grown into the most important operating system in the world and has begun to dominate each market it has entered. Linux has a dominant use within real-time applications such as aviation, flight simulators, medical, military, controls, audio/video, with emerging use in automotive and safety-critical products. As cybersecurity requirements continue to grow, the implementation of compliant security controls become increasingly important. The security control

distribution framework should provide an automated and flexible approach for use by organizations of all sizes and use cases.

The primary objective of this work is to provide a means to automate the build and distribution of a Linux Secure Host Baseline (SHB) for use in real-time applications with acceptable Security Content Automation Protocol (SCAP) compliance utilizing freely available tools. A driving factor for this approach is the Department of Defense (DoD) Chief Information Office (CIO) memorandum dictating all DoD components transition to the Windows 10 SHB, focus on strengthening the cyber security posture of those systems, while streamlining the IT operating environment [1].

The system focused within this work utilizes the Community ENTerprise Operating System (CentOS) Linux operating system, the Real-time Linux Kernel Extensions (PREEMPT_RT Patch), Defense Information Systems Agency (DISA) Red Hat Enterprise Linux (RHEL) Security Technical Implementation Guide (STIG), and the SCAP Compliance Checker (SCC) and OpenSCAP tools. Each of these independent approaches will be covered in the following background section.

2 Background

2.1 Security Content Automation Protocol (SCAP)

SCAP is a common method for utilizing a specific set of standards for enabling automated security vulnerability compliance evaluation for organization's systems through quantitative and qualitative metrics. Generally, SCAP is a synthesis of interoperable specifications derived from community driven ideas and participation to ensure a broad range of use cases for a common evolving security goal.

The National Institute of Standards and Technology (NIST) describes SCAP content checklists as adhering "to the SCAP specification in NIST SP 800-126 for documenting security settings in machine-readable standardized SCAP formats. SCAP content checklists can be processed by SCAP-validated products, which have been validated by an accredited independent testing laboratory as conforming to applicable SCAP specifications and requirements" [2].

At a high-level, SCAP compliance is controlled using SCAP security guides (security recommendations) together with SCAP compliance tools to audit systems automatically. Generally, this means running an automated tool to check non-compliance security settings on a given system and then remediate those non-compliance settings iteratively until the system has reached a satisfactory level of compliance.

2.2 Security Technical Implementation Guide (STIG)

A STIG (security guide) is a cybersecurity methodology for standardizing security protocols within a network or computer to enhance the security posture of the entire system through configuration settings. The primary implementation of STIGs occurs on desktop computers or servers to prevent system access from unauthorized users. STIG design considerations typically cover accessibility, networks, routers, firewalls, domains, and switches. STIGs are specifically utilized by the Department of Defense

(DoD). The chosen STIG security profile for this work is `xccdf_org.ssgproject.content_profile_stig-rhel7-disa` which applies to RHEL 7 as provided by DISA. This profile will be used to determine the system compliance.

2.3 Government and Industry Collaboration – NIST, DISA, NSA, USGCB

There exists several widely accepted standardized government level security protocols and practices associated with specific agencies. These include NIST 800-53, DISA/STIG, National Security Agency (NSA), and United States Government Configuration Baseline (USGCB) which provides a set of security measures for several operating systems. While NIST is the primary agency for determining SCAP compliance approaches, both DISA and the NSA are recommended by NIST as agency-produced checklist vendors [3]. The primary focus for this work will be on NIST and DISA collaboration.

2.4 Windows 10 Secure Host Baseline (SHB)

DISA and the NSA co-developed and maintain a Microsoft Windows 10 standard desktop framework known as the DoD Windows 10 Secure Host Baseline (SHB). The purpose of the Windows SHB provides an automated and flexible approach for assisting the DoD in deploying the latest release of Windows 10 based on recommended practices, to be consumed by organizations of all sizes ranging from the large enterprise to small deployments. The SHB primarily focuses on heightening the security baseline of the operating system through Group Policy objects, compliance checks, and configuration tools for system administrators.

The framework can be downloaded from DISA and installed using a development installation of Windows 10 with no STIG policies implemented. The framework creates a reference virtual machine and applies STIG policies, and required components. Optional components and applications are installed along with any other organizational required customizations. A reference image (ISO) is then captured from the reference machine and optionally installed to deployment media (USB hard drive or optical media).

The framework includes core components and applications along with optional applications that can be installed during the build and capture phase. The framework also includes settings for the following STIGs by default:

- Microsoft Windows 10
- Microsoft .NET Framework 4
- Internet Explorer 11
- Windows Firewall

2.5 Community ENTerprise Operating System (CentOS)

Community ENTerprise Operating System (CentOS) Linux is a community maintained, stable open source operating system built with changes from the Red Hat Enterprise Linux Source Code. While being a community project, CentOS Linux does not inherit Red Hat Enterprise Linux certifications or evaluations. However, CentOS is commonly used for Linux applications requiring real-time performance or for system servers. CentOS comparatively has a longer release and support cycle which results in a very stable system.

2.6 Real-time Linux Kernel Extensions (PREEMPT_RT Patch)

A real-time operating system is needed for running applications that have real-time requirements. Real-time applications are primarily focused on determinism and maximizing response time rather than throughput. Real-time applications have operational deadlines between an event trigger and its response to that event. Scheduling through priority schemes and ensuring higher priority tasks are run prior to lower priority tasks requires preemption. Preemption allows lower priority tasks to be interrupted in favor of higher priority tasks with the intent to resume those tasks later [4].

The PREEMPT_RT patch converts Linux into a fully preemptible kernel. This approach allows the user space to preempt kernel tasks to ensure determinism within the real-time application. Utilizing this patch with Linux provides a low cost and mature solution for a real-time operating system [4].

The official PREEMPT_RT kernel patch is maintained by kernel.org with different versions matching specific versions of the mainline kernel source. The latest versions can be downloaded from The Linux Foundation's Real-Time Linux Project page. The kernel patch can be integrated into a stock Linux distribution in multiple ways including being added to the kernel source tree and built using make, precompiled packages for RHEL for Real Time 7, and CentOS 7 repository maintained by CentOS [5].

Linux provides many features which support viewing, modifying, and tuning real-time performance of systems. One such system that is installed by default is Tuna which provides a graphical user interface for monitoring and configuring of task priorities, scheduling policies, and CPU affinities of tasks and threads in real-time to ensure real-time requirements are being met.

2.7 SCAP Compliance Tools – OpenSCAP (OSCAP), SCAP Compliance Checker (SCC)

SCAP compliance tools are validated scanners capable of authenticated vulnerability scanning available for both DoD and non-DoD use. These tools aid administrators and auditors with assessment, measurement, and enforcement of security baselines that adhere to some security standard. In this case the security standard is maintained by NIST with specific SCAP versioning certification. Generally, the usage and process of using compliance tools is an iterative process of checking the system against the security policies and remediating any discrepancies and vulnerabilities that are dis-

covered. DISA provides benchmarks for use with SCC, but not OSCAP, so SCC is used for determining compliance for this system.

OpenSCAP is a security compliance tool that checks security configuration settings and examines indicators of compromise by comparing the existing operating system settings to the chosen security policies. OpenSCAP is validated by NIST for use on RHEL, however, these certifications do not directly apply to CentOS even though CentOS is built from the Red Hat source. In this scenario, OpenSCAP is utilized for applying the RHEL 7 DISA STIG security profile to the CentOS 7 operating system during the initial setup process and remediating some of the deficiencies discovered.

The SCC is an automated compliance scanning tool that leverages the DISA STIGs for analyzing and reporting on the security posture of the scanned system [6]. The tool has no license cost to government or contractors, performs compliance scanning using SCAP content, creates an HTML report page as an output, and has graphical and command line interfaces [7].

At a high level, STIGs are just guidelines and during adjudication the examiner of the SCAP compliance results must determine whether non-remediated deficiencies are acceptable as risks for that organization using a more nuanced approach.

3 Related Work

3.1 Security Hardening Technology

The initial basis for generating hardening scripts used within the proposed framework has been expanded on from Frank Caviggia's CentOS 7 project used for a standard distribution that is not capable of real-time performance [8]. Caviggia developed these scripts while working at Red Hat and along with others in the community continues to maintain them. The primary goal of Caviggia's scripts is to configure and harden the baseline CentOS ISO using SCAP Security Guide (SSG). Red Hat maintains a RHEL based fork of Caviggia's project [9]. The security hardening is implemented using RHEL's standard kickstart technology.

3.2 Automated Hardening and Testing Linux

Henttunen presents a method to sufficiently securing a Linux operating system running on a VMWare workstation utilizing automated installation scripts and open source auditing tools [12]. The approach focused on a USGCB security standard compliant hardened environment rather than the DISA/STIG compliance approach. The VMWare environment is applicable to non-real-time and server applications typically used for cloud-based services.

4 DESIGN AND IMPLEMENTATION

The goal of the Linux SHB development is to mirror the framework and design approach achieved by the Windows SHB while providing adequate SCAP compliance through quantifiable metrics and enabling real-time capability. This approach will include the necessary real-time patch required for real-time application deployment through use of a custom system profile. The re-spin script and kickstart files were developed and tested against CentOS 7.6 1810. The approach for a re-spin is to modify and provide updates to an existing ISO image. The high-level approach described within this section can be visualized using Fig. 1.

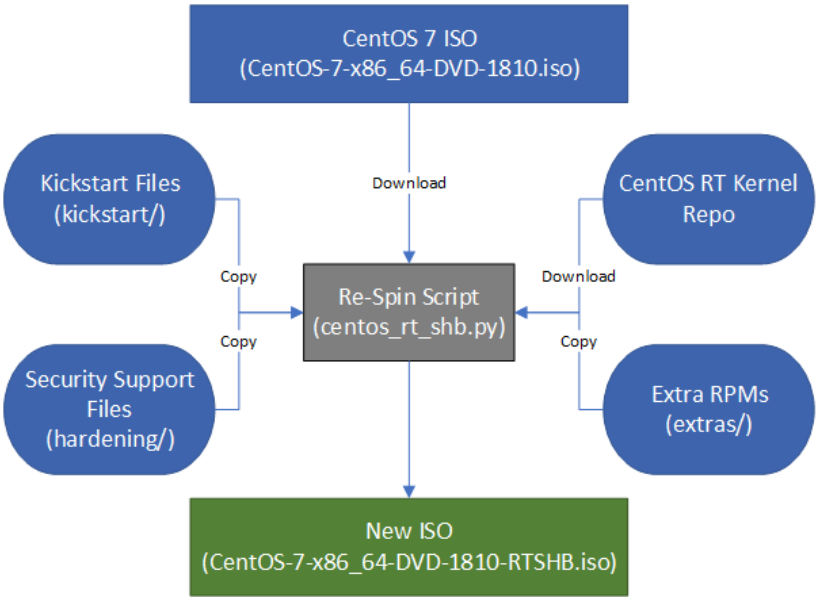

Fig. 1. CentOS ISO Re-Spin Process

4.1 Base Linux Distribution

The base operating system distribution selected is CentOS 7. This specific distribution is freely available, closely matches Red Hat Enterprise Linux releases, and has been widely used in industry. This approach will allow exact STIG configuration items to be applied and the same SCAP compliance tools to be utilized across both systems. CentOS will include the PREEMPT_RT Patch within its installation process.

4.2 Test Setup and Build Environment

The test setup utilizes a virtual machine for ease of use and reproducibility. This approach will not limit functionality and will translate directly for use on real hardware. The virtual machine software used is Oracle VirtualBox 5.2.22. The computer hard-

ware used is a Dell with Intel i7 x64-based processor, 16GB of RAM, and running Windows 10.

The CentOS re-spin script (centos_rt_shb.py) must be executed on an existing CentOS 7 installation with access to the internet. It is recommended that the build machine has at least 40GB of free hard drive space and at least 8GB of RAM. This script can be started within a virtual machine or on real hardware.

4.3 ISO Build Approach

The script is started within an existing CentOS 7 installation and will require the following packages. The script will install these packages using yum if they are not already installed.

- `yum-utils`
- `genisoimage`
- `createrepo`
- `isomd5sum`

The latest CentOS 7 ISO is downloaded directly from the CentOS website through their verified distribution directory. This provides the base Linux distribution used for generating the SHB ISO.

The real-time patch to the kernel will be integrated into the repository of the installation media. The real-time repository is cloned to the local machine and the downloaded packages are copied into the Package directory of the extracted ISO image. The script modified the ISO's package group repository data to include the RT group that was defined within the real-time repository. When the `createrepo` command is run on the new ISO package directory, the RT group is available for installation by the standard CentOS graphical installer and kickstart files.

4.4 Security Hardening Approach

The security hardening approach utilizes Red Hat's standard kickstart technology. The CentOS 7 kickstart project created by Caviggia that is community maintained is the base implementation that is expanded upon and tailored for this approach.

The top level kickstart file is called /kickstart/hardened-centos.cfg. This is executed during the CentOS installation and calls a graphical menu (menu.py) to allow the user to configure some basic attributes of the hardened system.

The following configuration option are available (see Fig. 2).

1. **Hostname**: This field allows the user to set the hostname of the installed system.
2. **System Profile**: This dropdown menu allows the user to select which profile will be installed.
3. **System Classification**: This dropdown menu allows the user to select the classification level of the system. The following options are available.
   a. Unclassified

   b. Unclassified/FOUO
   c. Confidential
   d. Secret
   e. Top Secret
   f. Top Secret/SCI
   g. Top Secret/SCI/NOFORN
4. **SCAP Security Guide Profile**: This dropdown menu allows the user to select the security profile to implement. Currently, DISA STIG is the only option.
5. **System Hardware Information**: This displays information about the hardware of the system.
6. **Available Disks**: These check boxes allow the user to select which disk(s) to install onto.
7. **Encrypt Drives with LUKS**: This checkbox allows the user to select whether LUKS encryption is used on the filesystems.
8. **Disable USB**: This checkbox allows the user to disable USB support.
9. **Lock root**: This checkbox disables the root account. A user account called admin is added with `sudo` capabilities to be used for system administration.
10. **Install Real-Time Kernel**: This checkbox installs the real-time kernel packages. This installs the RT package group. This is part of the expanded capability.
11. **Kernel in FIPS 140-2 Mode**: This checkbox enables FIPS 140-2 encryption on the system.
12. **Required/Optional LVM Partitioning Percentage**: This allows the user to define what percentage of the installation disk should be allotted to each partition. The required partitions are required to be defined per the STIG requirements. The values entered in the fields must add up to 100%.
13. **Network Configuration**: This button opens a dialog window allowing the user to configure the network interfaces of the machine.
14. **Help**: This button opens a dialog window with help information.
15. **OK**: This button is clicked once all other configuration is complete. The user will be prompted to enter a password. The password must be 15 characters long. The entered password will become the password for the root and admin accounts, the grub password, and the key to decrypt the drive is LUKS is enabled.

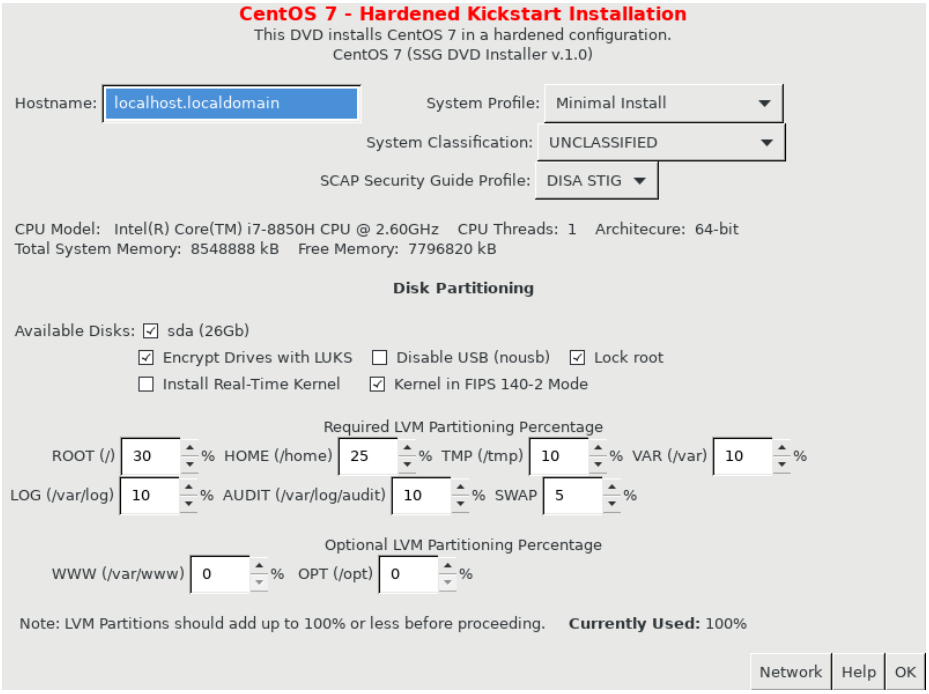

Fig. 2. CentOS Hardening Installation Menu

After the user completes the configuration using the graphical menu, the installation continues without user input.

The kickstart file utilizes the `%pre` and `%post` sections for scripting the security hardening process in its entirety. Selections from the graphical menu that were inputted by the user adds custom code to these sections as appropriate. The script will utilize the OSCAP tool provided within the OpenSCAP packages to apply the RHEL 7 DISA STIG security profile (`xccdf_org.ssgproject.content_profile_stig-rhel7-disa`) during installation.

The security profile itself satisfies approximately 80% of the STIG requirements for the system. Additional scripts and tools (including /hardening/supplemental.sh) complete the remaining remediations. SSH services are installed by default but a user must be a member of the `sshusers` group to log in remotely, which will be managed by the administrator.

4.5 Security Tools and Services

The following tools and services are included to satisfy the STIG requirements and in general to improve the security posture of the total system.

- **Smart Card Support**: Packages to support login services requiring a CAC or other smart card.
- **Auditd**: The standard Linux auditing daemon configured per STIG requirements.
- **SSSD/realmd**: System Security Service Daemon provided to support centralized authentication mechanisms (i.e. Windows Active Directory).
- **Classification Banner**: A configurable graphical classification banner for Gnome desktops on systems that implement a graphical user interface. The banner occupies a small amount of space at the top and bottom of the screen indicating the classification level of your system. This is an open source project maintained by the Red Hat community and has become common use in Industry [10].
- **USB Guard**: Software to control access to USB devices by implementing basic whitelisting and blacklisting capabilities based on device attributes [11].

4.6 System Profiles

The hardening kickstart file (/kickstart/hardened-centos.cfg) defines multiple profiles (/hardening/profiles/) that specify which packages and settings get installed. This aids in abstracting and specifying those profile configurations for ease of use, viewing, and modifying. These profiles are modifiable, new profiles can be added, and existing profiles removed easily. Except for the Real-time Host profile, all other profiles listed here are from Caviggia's original project. Those additional profiles have not been evaluated for SCAP compliance.

The following profiles are defined within the scope of this system:

- **Minimal Install**: This profile implements a minimum subset of packages during installation. Graphical support (`Xwin`) is not included.
- **Real-time Host**: This profile is designed to be used in a real-time environment as either a development or production host computer. Graphical support (`Xwin`) is included with this profile. The focus of this work and associated analysis occurs within this profile.
- **IPA Authentication Server**: This profile implements the minimal package set plus the packages needed to support IPA Authentication Server (Red Hat Identity Manager).
- **Ovirt KVM Server**: This profile implements the packages necessary to allow Ovirt Manager to be installed on the system. After installation, the script ovirt-engine-install.sh must be run to finish installing the Ovirt Manager packages.
- **Standalone KVM Server**: This profile installs the packages necessary for a standard KVM Virtualization Server.

4.7 Installation and Automation

The final ISO output is installed to the target machine using bootable media in the same form as the standard CentOS ISO distribution and works in the same manner. The ISO will fit on a standard DVD. For larger ISOs that include large amounts of

packages and data, a Blu-ray or USB bootable media may be required. When the media is booted the user is presented with the following menu selection.

Selecting Install CentOS 7 Secure Host Baseline will start the standard Linux SHB installation kickstart (/kickstart/hardened-centos.cfg). After making the selection, the user will be presented with the SHB installation menu (see Fig. 2). Once the menu selections are complete, the installation with continue automatically until the system is ready for reboot.

5 CUSTOMIZING THE DISTRIBUTION

The Linux Secure Host Baseline distribution is designed to be fully customizable and configurable to meet the organization requirements for the deployed system. The two primary methods for customizing the distribution is for changing profiles or kickstart files.

5.1 Profiles

Currently, the primary usage for the individual profiles is to ensure a specific configuration is accomplished during the installation process. The configurations can be modified through the /hardening/profile_package.cfg file and its associated profile Python configuration script (i.e. /hardening/profiles/Realtime_Host.py). The Real-time Host profile configures the disk partitions, runs the hardening and remediation scripts using OSCAP, configures the firewall, and reinstalls `Xwin` for the graphical interface. This method facilitates the final configuration for the system. New profiles can be generated that utilize different configuration settings which can be selected during the installation process to allow for variations in systems.

5.2 Custom Kickstart Files

User defined kickstart files can be added to the kickstart directory to accomplish modifications that cannot be as easily completed within the profile modifications. It is recommended that the user make a copy of and modify the existing /kickstart/hardened-centos.cfg file.

6 MAINTAINING THE DISTRIBUTION

Security compliance is a constantly evolving process that requires iterative improvements and is rarely static. There are a multitude of events that could trigger improvements or reevaluation over time within the realm of cyber security depending on the deployed environment.

6.1 Trigger Events

The primary trigger events that are specific to this automated Linux SHB approach but does not directly apply to the general state of cyber security are listed below.

- New Release of CentOS.
- New Version of the Red Hat STIG is Released.
- New Version of the SCAP Compliance Checker (SCC) Tool or OpenSCAP (OSCAP) Tool is Released.

6.2 Approach for Updates

Required updates following the trigger events will likely need to occur to centos_rt_shb.py or /kickstart/hardened-centos.cfg and its associated hardening scripts to account for changes to the underlying operating system and to implement new remediations found by the SCC. Fig. 3 shows an example process to update the distribution.

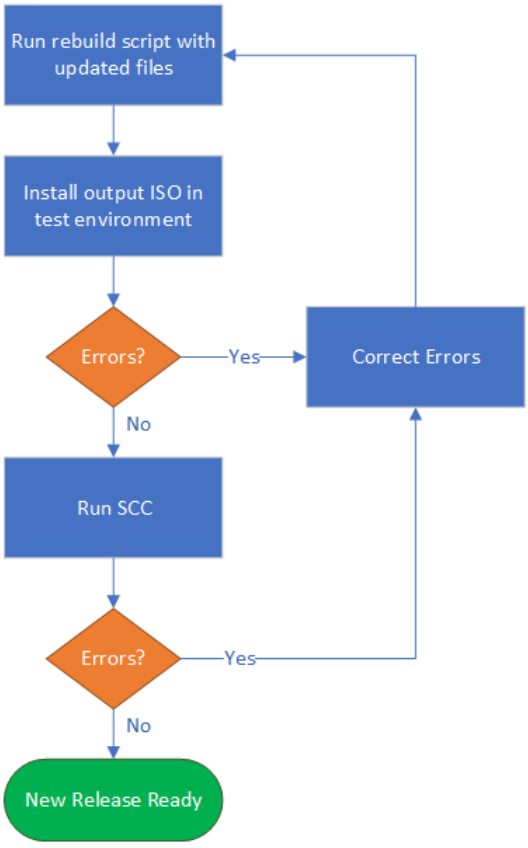

Fig. 3. ISO Update Process

7 SECURITY RESULTS ANALYSIS

The security compliance of the final system was analyzed using the SCAP Compliance Checker tool and the latest STIG SCAP content at the time that this work was performed, as shown in Fig. 4. The tool uses the SCAP content to evaluate the installed operating system and applications, and generates a report detailing the level of compliance along with deficiencies. The version of the compliance tool used was 5.2 with version 2.3 of the RHEL 7 STIG.

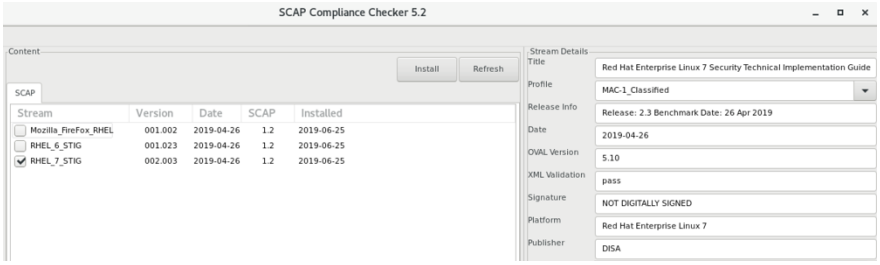

Fig. 4. SCAP Compliance Checker (SCC) Information

While the use of a security checklist can significantly improve the overall security of the system on which it has been checked against, no checklist can ensure the system is 100% secure. Given that, it is not always possible to reach 100% compliance within the chosen checklist depending on the nuances of a given system. This approach provides an approach to strive toward that 100% compliance, but not necessarily reach the 100% compliance goal.

7.1 Post Remediation Real-time Host Scan

The post remediation scan resulted in 175 out of 182 checks passed for 96.15% compliance for the Real-time Host profile as shown in Fig. 5. The qualitative guideline puts this into the green compliance status for being above 90%.

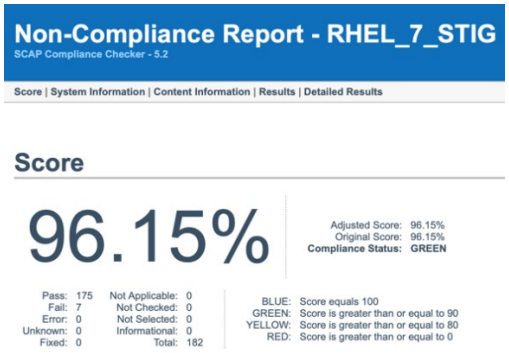

Fig. 5. SCAP Compliance Metrics

7.2 STIG Deficiencies Not Remediated

The STIG items listed in Table 1 were not remediated but their associated STIG ID and description are listed for reference.

UID 1-4 check fail due to the specific operating system used. This approach is using CentOS as opposed to RHEL. The checks are looking for packages that are only provided with REHL and cannot be remediated while using CentOS.

UID 5 fails due to X Windows display manager being installed. The profile that is being used requires X Windows to be installed for graphical user interfaces used within real-time processes. This can be remediated for profiles that do not require X Windows for normal operations.

UID 6-7 failures have unclear causes. The descriptions provided based on the STIG information should allow these checks to pass. So, more investigation would be required to remediate the failures.

Table 1. STIG Non-Compliance Deficiencies.

Unique ID (UID)	*STIG ID*	*Description*
Non-Red Hat Release		
1	SRG-OS-000480-GPOS-00227	The Red Hat Enterprise Linux operating system must be a vendor supported release. - (CCE-80349-4) – Fail
2	SRG-OS-000033-GPOS-00014	The Red Hat Enterprise Linux operating system must implement NIST FIPS-validated cryptography for the following: to provision digital signatures, to generate cryptographic hashes, and to protect data requiring data-at-rest protections in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards. - (CCE-80359-3) – Fail
3	SRG-OS-000033-GPOS-00014	The Red Hat Enterprise Linux operating system must use a FIPS 140-2 approved cryptographic algorithm for SSH communications. - (CCE-27295-5) – Fail
4	SRG-OS-000250-GPOS-00093	The Red Hat Enterprise Linux operating system must be configured so that the SSH daemon is configured to only use Message Authentication Codes (MACs) employing FIPS 140-2 approved cryptographic hash algorithms. - (CCE-27455-5) – Fail
X Windows Installed		
5	SRG-OS-000480-GPOS-00227	The Red Hat Enterprise Linux operating system must not have an X Windows display manager installed unless approved. - (CCE-27218-7) – Fail
Reason for Failure Unclear		
6	SRG-OS-000327-GPOS-00127	The Red Hat Enterprise Linux operating system

		must audit all executions of privileged functions. – Fail
7	SRG-OS-000023-GPOS-00006	The Red Hat Enterprise Linux operating system must display the Standard Mandatory DoD Notice and Consent Banner immediately prior to, or as part of, remote access logon prompts. - (CCE-27314-4) – Fail

7.3 Limitations and Proposed Next Steps

While the results of this work highlight the initial capability, the proposed solution is only a snapshot within a single instance in time. As Cyber Security is an ever-changing goal, and the software utilized is being continuously maintained and updated, there will need to be future engineering effort to facilitate improvements over time. For example, CentOS 8 will require some amount of time in deployment to ensure it has been fully vetted, as generally the newest operating systems are not used immediately in production and security environments, which will then require additional modification and testing to use this approach.

The proposed next steps for refinement include the following.

- **System Updating**: Determine the exact CentOS, RHEL STIG, and SCAP Compliance Tool updating impacts and their associated update process for affected files and scripts.
- **Patch Updating**: Determine the exact update process for incorporating the latest patches rather than generating a completely new ISO image.
- **Red Hat Enterprise Linux 7 Support**: Support for RHEL 7 should be included in the scripts. Some organizations require or will choose RHEL over CentOS, and this should only require minimal modification.
- **Deployment Readiness**: Additional testing and refinement to develop the scripts, kickstart files, and their support files should be performed to ensure the system is ready for deployment.
- **Feedback**: Iterative update processes from security administration professionals would be invaluable, as well as other use cases to generate additional profiles.
- **Additional Security Applications**: Determine the impact from including security applications that improve security posture such as a Domain Controller, Virus Scanner (i.e. McAfee), and System Logger (i.e. Kiwi Syslog).

8 CONCLUSION

The utilization of Linux is highly preferred in systems requiring real-time performance constraints. As generally real-time systems are used within industrial, aviation, and safety-critical applications, there is an inherent need or requirement to achieve a certain level of security compliance. This process can often be manual and require expertise to achieve. A goal of this work is to provide a framework to improve that current process, apply it using cost effective tools, automate the creation and distribu-

tion specific to real-time applications, and to quantitatively measure the resulting security posture of the deployed system.

The results of the SCAP Compliance Checker indicate a high level of compliance for the automated CentOS Real-time Host profile using the DISA RHEL 7 STIG. The approach presented offers an automated Linux solution to match the intent for improved security as proposed by the DoD approved Windows Secure Host Baseline while offering real-time functionality through the inclusion of the PREEMPT_RT patch.

References

1. R. O. Work, "Implementation of Microsoft Windows 10 Secure Host Baseline", official memorandum, Department of Defense, Washington, D.C., U.S.A, 2016. [Online]. Available: https://dodcio.defense.gov/Portals/0/Documents/Cyber/DSD%20Memo%20-%20Implementation%20of%20Microsoft%20Windows%2010%20Secure%20Host%20Baseline.pdf.
2. S. Quinn, M. Souppaya, M. Cook, K. Scarfone, "National Checklist Program for IT Products - Guidelines for Checklist Users and Developers", NIST Special Publication (SP) 800-70 Revision 4, 2018.
3. NIST Joint Task Force, "Security and Privacy Controls for Federal Information Systems and Organizations", NIST Special Publication 800-53 Revision 4, 2013.
4. Arch Linux, "Realtime kernel patchset". [Online]. Available: https://wiki.archlinux.org/index.php/Realtime_kernel_patchset.
5. Linux Foundation, "Real-Time Linux". [Online]. Available: https://wiki.linuxfoundation.org/realtime/start.
6. Center for Development of Security Excellence, "Getting Started with the SCAP Compliance Checker and STIG Viewer Job Aid". [Online]. Available: https://www.cdse.edu/documents/cdse/SCAP-compliance-checker-and-STIG-viewer-job-aid.pdf.
7. Space and Naval Warfare (SPAWAR) Systems Center Atlantic (SCC-LANT), "SCAP Compliance Checker User Manual for Red Hat Enterprise Linux", Version 4.0, 2015.
8. F. Caviggia, "Hardened CentOS 7 Kickstart", Jan. 2019. [Online]. Available: https://github.com/fcaviggia/hardened-centos7-kickstart.
9. Red Hat, "Hardened RHEL 7 Kickstart", Aug. 2017. [Online]. Available: https://github.com/RedHatGov/ssg-el7-kickstart.
10. F. Cavigga, "Classification Banner", Sep. 2018. [Online]. Available: https://github.com/fcaviggia/classification-banner.
11. Red Hat, "USBGuard", 2019. [Online]. Available: https://github.com/USBGuard/usbguard.
12. Henttunen, Kaisa. "Automated hardening and testing CentOS linux 7: security profiling with the USGCB baseline." (2018).