NIST Special Publication 800
NIST SP 800-219r2 ipd

Automated Secure Configuration Guidance From the macOS Security Compliance Project (mSCP)

Initial Public Draft

Eric Trapnell
Bob Gendler
Dan Brodjieski
Allen Golbig
Blair Heiserman
Mark Trapnell
Murugiah Souppaya
Karen Kent

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-219r2.ipd

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication 800
NIST SP 800-219r2 ipd

Automated Secure Configuration Guidance From the macOS Security Compliance Project (mSCP)

Initial Public Draft

Eric Trapnell
Mark Trapnell
Computer Security Division
Information Technology Laboratory

Murugiah Souppaya
**Former NIST employee; all work for this publication was done while at NIST.*

Bob Gendler
Customer Access and Support Division
Office of Information Systems Management

Dan Brodjieski

Allen Golbig
Jamf

Blair Heiserman
Information Technology Security and Networking Division
Office of Information Systems Management

Karen Kent
Trusted Cyber Annex

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-219r2.ipd

June 2026

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Arvind Raman, NIST Director and Under Secretary of Commerce for Standards and Technology

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Authority
This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3551 et seq., Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

NIST Technical Series Policies
Copyright, Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History
Approved by the NIST Editorial Review Board on YYYY-MM-DD [Will be added to final publication.]
Supersedes NIST Series XXX (Month Year) DOI [Will be added to final publication, if applicable.]

How to Cite this NIST Technical Series Publication
Trapnell E, Gendler B, Brodjieski D, Golbig A, Heiserman B, Trapnell M, Souppaya M, Kent K (2026) Automated Secure Configuration Guidance From the macOS Security Compliance Project (mSCP). (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-219r2 ipd. https://doi.org/10.6028/NIST.SP.800-219r2.ipd

Author ORCID iDs
Eric Trapnell: 0000-0001-9315-3732
Bob Gendler: 0000-0002-8928-6492
Blair Heiserman: 0009-0003-8779-6231
Mark Trapnell: 0000-0002-5266-3610

Murugiah Souppaya: 0000-0002-8055-8527
Karen Kent: 0000-0001-6334-9486

Public Comment Period
June 22, 2026 – August 14, 2026

Submit Comments
applesec@nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

The macOS Security Compliance Project (mSCP) provides resources that enable system administrators, security professionals, security policy authors, information security officers, and auditors to secure and assess macOS desktop and laptop, iOS, and visionOS system security in an automated way. This publication gives an overview of the practical, actionable recommendations (i.e., secure baselines and associated rules) that are available on the project's GitHub site, which is continuously curated and updated to support each new release of macOS, iOS, and visionOS. This publication also describes use cases for leveraging the mSCP content. Updates from the previous version of this publication highlight the mSCP's next generation of improvements, including simpler OS version and rule management as well as an expanded audience.

Keywords

Apple; baseline; configuration management; endpoint device security; iOS; macOS; macOS Security Compliance Project (mSCP); operating system security; security compliance; visionOS.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Supplemental Content

The mSCP's GitHub site is at https://github.com/usnistgov/macos_security, and the project documentation is at https://pages.nist.gov/macos_security/.

Trademark Information

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

   i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or

   ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: applesec@nist.gov

Table of Contents

Table of Figures

Acknowledgments

The mSCP is a multi-organizational, multi-industry, international effort that includes contributions from many sources. The authors wish to thank the many individuals who have helped develop the mSCP, including but not limited to:

NIST
Jason Blake
Stephanie Roberts

NASA
Gary Gapinski
Elyse Anderson
Joshua Glemza

Declarative IT GmbH
Henry Stamerjohann

Apple
Jamie Richardson
Chris Stone

Root 3
Jordy Witteman

Emory University School of Medicine
Elias G Kikano

Coursera
Cody Keats

Falconwood, Inc
Stephen Beale

AGE Solutions
Aaron Kegerreis

Center for Internet Security
Edward Byrd

Leidos
John Mahlman

Recognition is also extended to Isabel Van Wyk from NIST for editing the document. Finally, portions of this document are based on content from the mSCP NIST Pages, so the work of all project contributors is appreciated.

1. Introduction

NIST established the open-source macOS Security Compliance Project (mSCP) to provide organizations with security configuration guidance for macOS, iOS, and visionOS in a machine-readable and executable format. The mSCP is continuously updated, and the latest mSCP security checklist content is maintained on the GitHub page [1].

Security checklists are groups of technical settings used to configure a system to meet an organization's risk posture and requirements or to verify that a system complies with those requirements. The mSCP seeks to simplify the macOS, iOS, and visionOS security development cycle by reducing the amount of effort required to implement security checklists. This collaboration between federal agencies minimizes duplicative effort that would otherwise be needed for these agencies to create, maintain, and administer individual security checklists. Additionally, the secure checklist content is easily extensible by other parties to implement their own security requirements or frameworks.

Organizations that use mSCP content — particularly security checklist examples — should take a risk-based approach to select appropriate settings and values that match their organizational risk tolerance.

1.1. Purpose and Scope

This document provides a high-level overview of the mSCP, its components, and some common use cases, including online project documentation for in-depth technical information and instructions. This document is intended to be independent of mSCP covered content version releases; updates will be released if there are substantial changes to the mSCP. Section 1.6 provides an overview of new features.

The information in this document regarding the details of the mSCP GitHub site is accurate as of the time of publication. For the latest detailed information on mSCP content or the content itself, readers should visit the mSCP GitHub page and mSCP NIST Pages.

Organizations that need to reference a NIST Special Publication to demonstrate compliance with United States Government mandates for adopting secure configurations for their macOS, iOS, and visionOS devices may reference this document.

1.2. Audience

This document and the mSCP GitHub site are intended for system administrators, security professionals, policy authors, privacy officers, and auditors who have responsibilities involving macOS, iOS, and visionOS security. Vendors of device management, security, configuration assessment, and compliance tools that support mSCP content may find this document and the GitHub site to be helpful.

1.3. Relevance to NIST SP 800-70 and the National Checklist Program

The security checklists from the mSCP GitHub page are included in the National Checklist Program. NIST Special Publication (SP) 800-70r5 (Revision 5) [2] explains that federal agencies are required to use appropriate security configuration checklists from the National Checklist Program when available. Part 39 of the Federal Acquisition Regulations, Section 39.101 paragraph (c) states,

> In acquiring information technology, agencies shall include the appropriate information technology security policies and requirements, including use of common security configurations available from the National Institute of Standards and Technology's website at https://checklists.nist.gov. Agency contracting officers should consult with the requiring official to ensure the appropriate standards are incorporated.

1.4. Baselines, Checklists, and Benchmarks

The mSCP includes checklists and benchmarks, which are collectively referred to as *baselines* in this document. A baseline is synonymous with the SP 800-70 definition of checklists but is built from a catalog of recommended configuration settings based on available technical controls, not from a checklist or benchmark. The catalog of controls should be customized to build a specific checklist or baseline based on the organization's risk profile. Implementing every item is not likely to be possible or sensible in many operational scenarios.

Baselines can be used to assist in the creation of security benchmarks. A *benchmark* differs from a baseline in that it defines values in addition to a set of controls. Benchmarks are published by organizations that have made risk-based decisions, such as NIST, DISA, NSA, CISA, or vendor produced checklists. Organizations should also define their own benchmarks. These values are called organization-defined values (ODVs), and they exist throughout the baselines and can be set during customization. Organizations can also deviate from benchmarks and document justifications for their specific operational scenario.

1.5. Document Structure

The remaining sections and appendices of this document are as follows:

- Section 2 provides an overview of the project, including its goals and how its content can be used.
- Section 3 explains the major components of the mSCP and provides pointers to additional information on component usage.
- The References section lists the works cited throughout this document.
- Appendix A briefly discusses how mSCP can help meet the needs of people in several roles.

- Appendix B provides examples of how a security professional might use mSCP content.
- Appendix C provides an example of how to create a benchmark using ODVs.
- Appendix D lists selected acronyms and abbreviations used in this document.

1.6. New Features

With its June 2026 release, the mSCP seeks to increase its usability, expand its audience, improve consistency, reduce costs, and make it easier to follow updates. The enhancements include:

- **Branch unification.** Previously, different OS releases had separate branches. Now, users and developers can benefit from lower maintenance requirements with a single branch.
- **YAML rule file unification.** Previously, the operation of YAML rule files was system-specific. In the next generation of the mSCP, the rule files now declare the applicable operating system (OS), which results in a greatly reduced number of rule files when managing multiple operating system versions. Section 3.2 provides an in-depth explanation of YAML rule files.
- **Containerized mSCP.** The mSCP is now capable of running in a containerized environment. This allows for quicker deployment due to simplified installation requirements. See the mSCP NIST Pages for more details.
- **Improved Python support.** The mSCP now contains Python classes and libraries that allow third-party developers to interface with the project and invoke mSCP functionality directly from their own tools.
- **Granular `mobileconfig` file creation.** One `mobileconfig` file can now contain a single setting. Exceptions and exemptions are now easier to manage because settings can be controlled at a finer level.
- **Rule exclusion.** When tailoring baselines, configuration rules can now be excluded. This exclusion is properly reflected in the documentation to improve traceability. For an example of rule exclusion, see Appendix C.3.
- **Vendor manifest files.** Newly supported JSON-formatted outputs of an organization's tailored security benchmarks allow third-party tools to more easily import mSCP data.
- **Guidance documentation translation.** The mSCP now supports guidance documentation generation in languages other than English, and additional language support is planned for the future. See the mSCP NIST Pages and Appendix B.3 for more information.
- **Simplified interface.** A new mSCP command line interface (CLI) now combines the functionality of several scripts to improve ease of use.
- **Markdown support.** Markdown is now a supported documentation format.

2. Project Description

The mSCP is an open-source project that provides a programmatic approach to generating and using macOS, iOS, and visionOS security configuration checklists. The project's content can be used to create customized security checklists of technical security controls by leveraging a library of rules, each of which is mapped to requirements in one or more existing security standards, regulations, or frameworks. This approach provides stability in versioning and consistency of the content. Unifying and standardizing mSCP content checklist efforts via the mSCP means that updating security guidance is simplified and radically accelerated, even as new versions are introduced annually.

The mSCP started in August 2019 as a collaboration among operational information technology (IT) security staff from NIST, the National Aeronautics and Space Administration (NASA), the Defense Information Systems Agency (DISA), and the Department of Energy's (DOE) Los Alamos National Laboratory (LANL).[1] mSCP maps macOS, iOS, and visionOS settings to SP 800-53 with an extensible, modern approach to security guidance that could be used by any organization (e.g., government, enterprise, education) that needs to adhere to security compliance frameworks and policies.

As of this writing, the configuration settings represent guidance and best practices from SP 800-53r5 [4], SP 800-171r3 [5], the macOS DISA Security Technical Implementation Guide (STIG) [6], the Committee on National Security Systems (CNSS) Instruction (CNSSI) Number 1253 [7], the Center for Internet Security (CIS) Critical Security Controls Version 8 [8], and internal organizational security guidance from NIST, NASA, and LANL.

2.1. Project Goals

Apple releases new macOS, iOS, and visionOS versions every year, and agencies and organizations must typically wait for guidance or accept risks before deploying the new version. Many agencies and organizations created their own internal security configuration, which delayed the deployment of the new version as well as new hardware that only supported the new version. The mSCP helps organizations upgrade sooner by routinely releasing settings near the OS release. Generally, the technical security settings in mSCP-supported platforms do not drastically change between releases, with only a handful of new settings being introduced. By pursuing a rules-based approach, mSCP rules that remain applicable can be reused and incorporated into guidance for the latest version of supported context. This enables quicker adoption of new security features that are not offered in prior versions.

The goals of the mSCP are to:

- Develop recommended security baselines using a risk-based approach
- Normalize and accelerate annual adoption of the new operating system and hardware by providing guidance to meet the security needs of new operating systems as early as possible

[1] See https://github.com/usnistgov/macos_security#authors for a current list of project contributors.

- Reduce worldwide efforts in creating annual guidance by unifying and consolidating compliance efforts into a single project
- Develop a methodology to foster collaboration between baseline authors to reduce overhead and redundancy
- Establish a unified approach for the configuration and assessment of controls across multiple sources and tools
- Enable the customization of existing content and the creation of new content, including custom baselines that meet organization-specific security requirements
- Provide device management and security tool vendors, auditors, and Apple with insight into customer security configuration needs

2.2. mSCP Content Use

Any organization can use mSCP content to set and assess the security configuration of mSCP covered content. Security baselines can map existing guidance or controls (e.g., SP 800-53r5 [4]) or be customized to meet an organization's specific needs.[2] In mSCP terminology, a security baseline is represented as a *baseline file* that designates the rules for meeting a specific set of requirements. The mSCP provides a library of *rules* that are settings for baseline-included content. Each rule is mapped to a requirement within a security standard or framework. Baseline files and rules comprise much of the mSCP's content.

The mSCP offers the following example baselines with descriptions adapted from Federal Information Processing Standards Publication (FIPS) 199 [9]:

- The ***SP 800-53r5 low baseline*** maps the minimum controls required to secure a system that is defined as a low-impact information system. The loss of confidentiality, integrity, or availability could be expected to have a **limited** adverse effect on organizational operations, organizational assets, or individuals.
- The ***SP 800-53r5 moderate baseline*** maps the minimum controls required to secure a system that is defined as a moderate-impact information system. The loss of confidentiality, integrity, or availability could be expected to have a **serious** adverse effect on organizational operations, organizational assets, or individuals.
- The ***SP 800-53r5 high baseline*** maps the minimum controls required to secure a system that is defined as a high-impact information system. The loss of confidentiality, integrity, or availability could be expected to have a **severe or catastrophic** adverse effect on organizational operations, organizational assets, or individuals.

Organizations that use any baseline example should take a risk-based approach to selecting the appropriate settings and ODVs, depending on the context under which the baseline will be applied. Organizations should tailor any of the baselines to include controls that are specific to

[2] Per customization and deviation guidance in SP 800-70r5, "Organizations should carefully evaluate the checklist settings and then make any changes necessary to adapt the settings to the organization's environment, requirements, policies, risk tolerance, and security objectives. This is particularly true for checklists that are intended for an environment with significantly different security needs."

their needs and to produce evidence of control enforcement. Additional information on baseline customization can be found in SP 800-70 [2], which discusses the importance of customizing and testing baselines before applying them to a production system.

The mSCP provides a command-line interface (CLI) that can be used with baselines for several purposes, including:

- Scripts and profiles for configuring mSCP content
- Generating a mapping between security standards, regulations, and frameworks
- Producing human-readable documentation in a variety of formats
- Customizing existing baselines

mSCP content can also be used to generate Security Content Automation Protocol (SCAP) content for automated security compliance scans. The generated SCAP follows the SCAP 1.4 specification [10]. The generation of SCAP content uses an Extensible Stylesheet Language Transformations (XSLT) file to create an Extensible Configuration Checklist Description Format (XCCDF) document with an accompanying Open Vulnerability and Assessment Language (OVAL) document. The XCCDF and OVAL documents are bundled into an SCAP data stream collection document with accompanying files that include Common Platform Enumeration (CPE) dictionary [11] information and an Open Checklist Interactive Language (OCIL) document. This creates an SCAP 1.4 document that validates using the NIST SCAP Content Validation Tool[3] and can be used by SCAP tools on macOS. More information on SCAP content generation is available at the project page.

[3] See https://csrc.nist.gov/Projects/security-content-automation-protocol/scap-releases/scap-1-4.

3. mSCP Components

This section provides an overview of several components of the mSCP: security baseline files, configuration profiles and utilities, content generation, customization capabilities, and directories. More information about all of these is available at the mSCP NIST Pages.

3.1. Catalog of Controls

The mSCP provides a library of technical security settings called *rule files* that map to various compliance requirements. Individual rules can demonstrate compliance with multiple security compliance frameworks. For example, applying and checking against a rule that is tagged with multiple controls (e.g., SP 800-53r5's CM-7 and CM-7(1) [4] and SP 800-171r3's 03.04.06 [5]) shows applicability to multiple SP 800-53 baselines (i.e., low, moderate, and high) as well as the Cybersecurity Maturity Model Certification's (CMMC) Level 1 and Level 2 [12] for that rule. A "test once, comply with many" approach reduces the burden of securing and demonstrating compliance. These technical security settings in the mSCP do not drastically change between OS releases. Typically, only a handful of new settings are introduced. By using a rules-based approach, applicable mSCP content can be reused and incorporated into guidance for the latest OS version.

An organization's defined benchmark can be assembled from all available rule files or just a subset. By using the mSCP's tailoring functionality, organizations can choose both a specific technical security posture and an overall compliance posture. If compliance with multiple frameworks is required, a single custom benchmark can be created that combines the rules needed for each framework. Different ODVs for a rule may change the compliance posture. For example, a password length of 12 may only meet SP 800-53 low impact requirements, but a password length of 15 may meet SP 800-53 low, moderate, and high requirements as well as CMMC Level 1 and 2.

3.2. Security Baseline Files

In the mSCP, a security baseline is defined in a Yet Another Markup Language (YAML) file. A YAML file is a human-readable file format commonly used by configuration files in which data are stored and/or transmitted. A baseline YAML file consists of the following required fields:

- `title` — A human-readable name for the baseline
- `description` — A short description of the baseline, including its use case and target OS version
- `authors` — Maintainers of the baseline
    - `name` — Author's name
    - `organization` — Author's organization
- `parent_values` — Source of values for ODVs
- `platform` — Platform for the security baseline or benchmark

- **os** — The operating system for the baseline file (i.e., macOS, iOS, visionOS)
- **version** — Version of the operating system the baseline file supports

- **profile** — The security content portion of the baseline
  - **section** — A keyword for organizing settings
  - **rules** — The names of the rule files that are a part of this baseline

The code shown in Fig. 1 provides a partial example of a YAML file that illustrates the use of these fields (with field names bolded).

```yaml
title: "Apple macOS 26 (Tahoe) Test Baseline"
description: |
  This guide describes the prudent actions to take when securing a macOS 26
system against the Test Baseline.
authors:
- name: Joe Doe
  organization: National Institute of Standards and Technology
parent_values: recommended
platform:
  os: macOS
  version: 26.0
profile:
- section: Authentication
    rules:
      - auth_pam_login_smartcard_enforce
      - auth_pam_su_smartcard_enforce
      - auth_pam_sudo_smartcard_enforce
      - auth_smartcard_allow
- section: "Auditing"
    rules:
      - audit_acls_files_configure
      - audit_acls_files_mode_configure
      - audit_acls_folder_wheel_configure
```

Fig. 1. Security baseline YAML file

3.2.1. Rule File Composition

A YAML rule file requires the following fields, which are from the Rules section of the mSCP NIST Pages:

- `id` — The id should match the file name without the YAML file extension.
- `title` — The title is a human-readable title of the rule.
- `discussion` — The discussion should provide a concise description of the intended use of the rule.
- `references` — The references include a mapping of the security frameworks, guidance, and individual controls that have been mapped to the rule. The official repository of

NIST CCEs [13] provides more information. Potential references include material from the following:

- nist — Includes a Common Configuration Enumeration (CCE) identifier (required) and references to SP 800-53r5 and SP 800-171r3. A NIST reference is required for all rules.
- disa — Includes references for security frameworks provided by the Defense Information Systems Agency.
- cis — Includes references from the Center of Internet Security.
- bsi — Includes references from the Federal Office for Information Security for Germany.

- `platforms` — Applicable platforms for the rule. Options include:
  - macOS — The macOS platform should be added for rules that are intended for macOS. Key properties include:
    - enforcement_info — Information on how to perform a check and, where applicable, a fix for the rule.
    - result — The result of the check for the rule.
    - benchmarks — The benchmarks that require the rule.
    - introduced — The OS version that introduced the setting.[4]
  - iOS — The iOS platform should be added for rules that are intended for iOS/iPadOS. Key properties include:
    - benchmarks — The benchmarks that require this rule.
    - introduced — The OS version that introduced the setting.[4]
  - visionOS — The visionOS platform should be added for rules that are intended for visionOS. Key properties include:
    - benchmarks — The benchmarks that require this rule.
    - introduced — The OS version that introduced the setting.[4]

Additionally, optional fields may be specified:

- odv — If a rule supports the ODV functionality, then the odv section should be present. At a minimum, this field should contain a `hint` (i.e., provides a description when tailoring a baseline) and a default value that replaces the $ODV variable. The following properties are required when specifying an odv:
  - `hint` — Information about the ODV, including a datatype (required), description (required), and validation criteria.

[4] If a setting was never officially documented by Apple, this value is "-1."

    - recommended — Suggested ODVs for specific benchmarks that do not prescribe values.

- mobileconfig_info — The information needed for creating the mobileconfig configuration profile. PayloadType and PayloadContent are required properties.
- ddm_info — The information required for creating the declarative device management (DDM) configuration is required in the ddm_info area. Required properties depend on the declaration_type.
- tags — Tags are keywords used to categorize and identify related rules, and they can be added to or modified as needed. Tags can also be used to make index-based searching of the rules faster and easier.

The code in Fig. 2 provides a notional example of a YAML rule file (with field names bolded).

```
id: os_genmoji_disable
title: Disable Genmoji AI Creation
discussion: |
  Apple Intelligence features such as Genmoji that use off device AI _MUST_
be disabled.
references:
  nist:
    cce:
      macos_26:
        - CCE-95196-2
      macos_15:
        - CCE-94196-3
      ios_26:
        - CCE-95480-0
      ios_18:
        - CCE-94535-2
      visionos_26:
        - CCE-95577-3
    800-53r5:
      - CM-7
      - CM-7(1)
    800-171r3:
      - 03.04.06
  disa:
    cci:
      - CCI-000381
      - CCI-001774
    srg:
      - SRG-OS-000095-GPOS-00049
    disa_stig:
      macos_26:
        - APPL-26-005140
      macos_15:
        - APPL-15-005140
      ios_26:
        - AIOS-26-017400
      ios_18:
        - AIOS-18-017400
      visionos_26:
        - AVOS-02-017400
    cmmc:
      - CM.L2-3.4.6
      - CM.L2-3.4.7
platforms:
  macOS:
```

```
    '26.0':
      benchmarks:
        - name: disa_stig
    '15.0':
      benchmarks:
        - name: disa_stig
    enforcement_info:
      check:
        shell: |-
          /usr/bin/osascript -l JavaScript << EOS

$.NSUserDefaults.alloc.initWithSuiteName('com.apple.applicationaccess')\
          .objectForKey('allowGenmoji').js
          EOS
        result:
          string: 'false'
    introduced: '15.0'
  iOS:
    '26.0':
      supervised: false
      benchmarks:
        - name: ios_stig
    '18.0':
      supervised: false
      benchmarks:
        - name: ios_stig
    introduced: '18.0'
  visionOS:
    '26.0':
      supervised: true
    introduced: '2.4'
tags:
  - 800-53r5_low
  - 800-53r5_moderate
  - 800-53r5_high
  - cnssi-1253_low
  - cnssi-1253_high
  - 800-171
  - cmmc_lvl2
  - cmmc_lvl1
  - cnssi-1253_moderate
mobileconfig_info:
  - PayloadType: com.apple.applicationaccess
    PayloadContent:
      - allowGenmoji: false
```

Fig. 2. YAML rule file

3.2.2. Rule File Categories

The mSCP organizes YAML files in the `rules` directory into the following subdirectories, each of which corresponds to a category of settings:

- **`audit`** — OpenBSM[5]
- **`auth`** — Smartcard authentication
- **`icloud`** — Apple's iCloud/Apple Account service

[5] See OpenBSM at https://github.com/openbsm/openbsm.

- **os** — Rules to configure the OS that do not fit into the other categories
- **pwpolicy** — Password policy
- **system_settings** — Settings controlled within the System Settings application on macOS and the Settings application on iOS and visionOS

The rules directory also includes a **supplemental** subdirectory, which contains additional information that supports the guidance provided by the baselines. Supplemental content contains information for rules that are not part of an existing baseline but could be beneficial for certain use cases. Supplemental content may not have mappings and may not contain the YAML rule file check and fix sections mentioned in Sec. 3.2.1. Supplemental content can be added to enhance baselines if organizational requirements are different than the system baseline requirements.

3.3. Configuration Profiles and Scripts

When an mSCP YAML file is processed, it yields a compliance script, configuration profile (mobileconfig file), and Declarative Device Management (DDM) content as outputs. All are used to apply configuration settings to a system.

A *configuration profile* is an Extensible Markup Language (XML) formatted file with a mobileconfig extension that contains a configuration payload. Apple's operating systems can automatically configure themselves based on a mobileconfig file's contents upon installation. Configuration profiles offer a convenient, Apple-supported mechanism for applying security settings to an Apple environment. Additionally, they can be cryptographically signed to ensure integrity and authenticity. These factors make configuration profiles the preferred vehicle for configuration delivery. However, mobileconfig files cannot modify all Apple device settings, so a configuration script or DDM configuration is needed for settings that are not supported by mobileconfig files. The developer documentation page provides an example configuration profile and brief descriptions of its properties.

Declarative device management (DDM) is a newer configuration mechanism that is also supported by the mSCP as Apple continues to develop it. Using JavaScript Object Notation (JSON), DDM seeks to simplify the device management process and reduce the enforcement effort required of both administrators and servers by shifting more configuration responsibility to the client systems. Since the feature set is currently evolving, some configuration items are not yet available in DDM, whereas some newer settings are DDM only. The Apple DDM developer documentation page and mSCP NIST Pages offer for more in-depth explanations.

A *compliance script* is a shell script containing configuration and remediation code that can directly manipulate operating system files. The script content is derived from all YAML rule files that have a mobileconfig value of false and belong to the specified baseline. The YAML rule file must contain the fix section in order to generate its corresponding configuration and remediation entry.

3.4. Content Generation

The mSCP provides a single CLI to generate baseline files, compliance scripts, and other types of content. The content types available for generation are described in the following subsections.

3.4.1. Generate Baseline

The `mscp.py baseline` subcommand compiles a list of security rules into a single baseline YAML file. This subcommand can be used to modify an existing security benchmark or create a new one. The baseline creation will default to the most current mSCP-supported version if the `--os_version` and `--os_name` arguments are not supplied. The mSCP NIST Pages provides additional information.

3.4.2. Generate Guidance

The `mscp.py guidance` subcommand can produce human-readable documentation in several formats, a Z shell script to evaluate compliance, and configuration profiles for mobile device management (MDM) deployment.

The `guidance` subcommand takes a baseline file and produces a human-readable guide with information from the YAML rules files. The script can create documentation in several formats but always generates an AsciiDoc file. AsciiDoc (`.adoc`) is a plain text format that uses markup conventions for traditional document formatting and organization. AsciiDoc files are easily transformable into many other formats via the `generate_guidance.py` script, including MarkDown, HTML, PDF, and Excel. The Excel format is particularly useful for quickly viewing all of the rules of a baseline, and it contains all of the data in the YAML rules files.

The `guidance` subcommand can also create configuration profiles (`mobileconfig` files), DDM artifacts, and a shell script used to evaluate and remediate compliance. Using the `-s` argument, the `guidance subcommand` will generate the shell script and an `org.{baseline}.audit.plist`. This `audit.plist` file can be used to set an exemption to organizational rules for approved users so that compliance checks can succeed without findings. To create an exemption for a rule, the `exempt` field should be set to `true`, and an `exempt_reason` should be added.

The mSCP NIST Pages provide more information on the `guidance subcommand`.

3.4.3. SCAP Generation

The SCAP generation `subcommand`, `scap`, can generate an SCAP 1.4 document, XCCDF document, or OVAL file. This builds content from available tags within the YAML files and does not need to be pointed to a baseline file.

For more information, see the mSCP NIST Pages.

3.4.4. Generate Mapping

The `mapping` subcommand allows for the quick creation of custom rules and baselines for a compliance framework not published by the mSCP. This subcommand requires a user-created comma-separated values (CSV) file with control identifiers that map to a new framework (CSV column 1) from another already that is defined by the project (CSV column 2). By default, the subcommand is designed to map a framework to the SP 800-53r5 [4] set of controls. Adding the `-f` option allows for mapping to another supported framework. The mSCP NIST Pages provide more information on the `mapping` subcommand.

3.4.5. Generate JSON Manifest

The generate guidance's `--manifest` subcommand allows for the creation of a JSON manifest file that can be used by other tools to process the rules in an organization-created benchmark. The JSON file contains all of the information that may be needed for importing into a tool.

3.5. macOS Security Compliance Script

The `mscp.py guidance` subcommand has an option to produce a compliance script written for a Z shell (see Sec. 3.4.2).

The `{baseline}_compliance.sh` script runs interactively by default. It can evaluate a system's conformance to a baseline or remediate any incorrectly configured settings. Alternatively, the script can autonomously assess a system with the `-check` argument or automatically remediate baseline settings with `-fix`.

Figure 3 shows an example of the results of running the script.

```
[INFO] [2026-03-19T14:26:52Z] system_settings_time_server_enforce failed
(Result: , Expected: “true”)
[INFO] [2026-03-19T14:26:52Z] system_settings_token_removal_enforce failed
(Result: , Expected: “1”)
[INFO] [2026-03-19T14:26:52Z] system_settings_touch_id_settings_disable
failed (Result: 0, Expected: “1”)
[INFO] [2026-03-19T14:26:52Z] system_settings_touch_id_unlock_disable failed
(Result: , Expected: “false”)
[INFO] [2026-03-19T14:26:53Z] system_settings_usb_restricted_mode passed
(Result: true, Expected: “true”)
[INFO] [2026-03-19T14:26:53Z]
system_settings_wallet_applepay_settings_disable failed (Result: 0, Expected:
”1”)
```

Fig. 3. Compliance script sample output

For more information on the mSCP compliance script, see the mSCP NIST Pages.

3.6. Customization

Organizations should make risk-based decisions on what controls and rules to use and how to apply them, as stated in SP 800-70r5 and SP 800-53r5 controls PL-10 and PL-11. Organizations can generate their own customized baselines and content outside of that provided by the

project. Additionally, organizations can add content for internal-only controls that are not suitable for inclusion in a global baseline. Customization primarily takes place within the custom folder. Examples of customizations that are supported by mSCP include:

- **Baselines:** A `baseline` folder can be included within the `custom` folder to create customized baselines that fit an organization's needs. These baseline files may include rule, section, and template customization (discussed below). An existing baseline can be configured to create a custom benchmark. Customizing an included benchmark may affect its compliance with the original requirements of that benchmark.
- **Rules:** Existing rules can have their setting values overridden via the `custom` folder instead of modifying the mSCP-supplied rule file. New rules can be created and added to existing baselines or to user-defined baselines. Organizations can create their own discussions, checks, results, fixes, and mappings of rules to security frameworks that are not included in the project. In order to override an existing rule, the custom rule file name must match an existing rule so that the mSCP CLI's generate guidance functionality will pick up the new values. New rules that *are not included in the mSCP must be listed in the baseline YAML file* specified when running `mscp.py guidance`.
- **Sections:** Custom sections can be used to organize existing or custom YAML rule files. Sections defined in the `custom` folder must be included in a baseline YAML file in order to be used by the `mscp.py guidance` subcommand.
- **Templates:** Custom templates can be used to define new template structures for the project and affect the organization and appearance of generated documentation. The template files must match the name of an existing template and will override that template when running the `mscp.py guidance` subcommand.
- **Logos:** An organization can include a custom logo when invoking the mSCP CLI's guidance functionality and using the `-l` argument to point to an image file.
- **Tailoring:** The mSCP CLI allows for a baseline to be tailored. During the tailoring process, there will be a prompt for each control to be included in the baseline. If a rule is to be excluded, there is a prompt to provide justification. This justification will be included in the documentation. Any rule containing an ODV will also prompt to have its values customized. If a value is not supplied to a control with an ODV, it will use the default value in the rule file. Appendix C provides an example of tailoring with ODVs.

3.7. Directories

mSCP source-code releases that are available on the mSCP GitHub include the following directories:

- **baselines** — A symlink that points to src/mscp/data/baselines
- **`build`** — Placeholder for all generated scripts, documents, and configuration profiles
- **`custom`** — Used to create customized baselines, rules, sections, or templates to meet an organization's requirements

- **rules** — A symlink that points to `src/mscp/data/rules`
- **schema** — YAML schema definitions used for validation
- **src/mscp** — Python source code and project resources
    - **data:**
        - **baselines** — The set of all supported baselines included in mSCP
        - **images** — Image resources used for guidance generation
        - **includes** — Contains the YAML-based libraries required for guidance generation
        - **locales** — Localization files for language and translation support
        - **rules** — The library of all mSCP rule files
        - **sections** — YAML content that describes the various sections used in the guidance
        - **templates** — Template files used for guidance generation
        - **themes** — Formatting and style definitions used for guidance generation

References

[1] macOS Security Compliance Project (2026) *macOS Security Compliance Project*. Available at https://github.com/usnistgov/macos_security

[2] Quinn SD, Heiserman B (2026) National Checklist Program for IT Products: Guidelines for Checklist Users and Developers. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-70r5. https://doi.org/10.6028/NIST.SP.800-70r5

[3] Joint Task Force Transformation Initiative (2013) Security and Privacy Controls for Federal Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-53r4, Includes updates as of January 22, 2015. https://doi.org/10.6028/NIST.SP.800-53r4

[4] Joint Task Force (2020) Security and Privacy Controls for Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-53r5. Includes updates as of December 10, 2020. https://doi.org/10.6028/NIST.SP.800-53r5

[5] Ross R, Pillitteri V (2024) Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-171r3. https://doi.org/10.6028/NIST.SP.800-171r3

[6] Department of Defense (2026) *DISA STIG for macOS*. Available at https://www.cyber.mil/stigs/downloads

[7] Committee on National Security Systems (2022) Security Categorization and Control Selection for National Security Systems. (National Security Agency, Ft. Meade, MD), Committee on National Security Systems Instruction (CNSSI) No. 1253. Available at https://www.cnss.gov/CNSS/issuances/Instructions.cfm

[8] Center for Internet Security (2026) *CIS Critical Security Controls Version 8*. Available at https://www.cisecurity.org/controls/v8/

[9] National Institute of Standards and Technology (2004) Standards for Security Categorization of Federal Information and Information Systems. (U.S. Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 199. https://doi.org/10.6028/NIST.FIPS.199

[10] Prisaca D, Quinn SD, Vander Pol J, Harris D (2025) The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.4. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-126r4 ipd. https://doi.org/10.6028/NIST.SP.800-126r4.ipd

[11] National Institute of Standards and Technology (2025). *Official Common Platform Enumeration (CPE) dictionary*. Available at https://nvd.nist.gov/products/cpe

[12] Department of Defense (2024). *Cybersecurity Maturity Model Certification (CMMC) Model Overview*. Available at https://dodcio.defense.gov/Portals/0/Documents/CMMC/ModelOverviewv2.pdf

[13] National Institute of Standards and Technology (2025) *CCE Platform Listing*. Available at https://ncp.nist.gov/cce

Appendix A. mSCP User Roles

The mSCP was designed to meet the needs of different security roles. These perspectives are briefly examined below.

Security policy authors define the policies for their organizations. The customization and ease of extensibility offered by the mSCP facilitate new content creation. Policy authors will need to familiarize themselves with the YAML rule file format (Sec. 3.2.1), particularly the ability to map rules directly to references. Additionally, the generate mapping functionality (Sec. 3.4.4) enhances portability between compliance frameworks.

System administrators and security professionals are responsible for configuring the systems under their purview. They implement the guidance issued by security policy authors. Security professionals may wish to generate baselines (Sec. 3.4.1), guidance (Sec. 3.4.2), and configuration using the macOS Security compliance script (Sec. 3.5).

Auditors approach mSCP covered content security compliance from a validator perspective, seeking proof that a system is configured in the required way. They are more interested in system setting documentation and compliance evidence than technical tools, such as configuration scripts. Both needs can be met by the mSCP. The generate guidance capability (Sec. 3.4.2) provides the necessary documentation in a variety of formats, including HTML, PDF, and Excel. The macOS Security compliance script (Sec. 3.5) assesses a system and produces a log of the results. Additionally, some auditors may be interested in examining YAML rule content directly (Sec. 3.2.1).

Information security officers have a variety of goals but are ultimately responsible for ensuring that systems are configured according to their organizational requirements. To accomplish this, they need policy documentation (Sec. 3.4.2) and the results of compliance scans (Sec. 3.5). Information security officers may also be responsible for reviewing the security rules proposed by the policy authors. If this is the case, they may be interested in YAML rule file components (Sec. 3.2.1).

Vendors of device management, security, configuration assessment, and compliance tools can produce a series of audit files based on mSCP content to support different macOS, iOS, and visionOS versions and associated security baselines. These audit files can be maintained, tested, published, and supported by the tool vendors. Tool customers can download and import the content into the tool to assess the state of their system against a particular baseline in an automated way.

Appendix B. Example of mSCP Usage by a Security Professional

This appendix provides examples of how a security professional might use mSCP content, though people in other roles might perform some of the same actions. mSCP utilities are not meant to replace enterprise-class configuration and management tools. Configurations should be tested on development systems before being deployed on end users' systems. The examples illustrated below were accurate at the time of publication. See the mSCP NIST Pages for up-to-date usage guidance.

B.1. List the Available Baselines

mSCP components rely on the setup detailed on the Getting Started Page. After completing the process for the desired operating environment, view the available baseline options using the `mscp.py` CLI, as shown in Fig. 4.

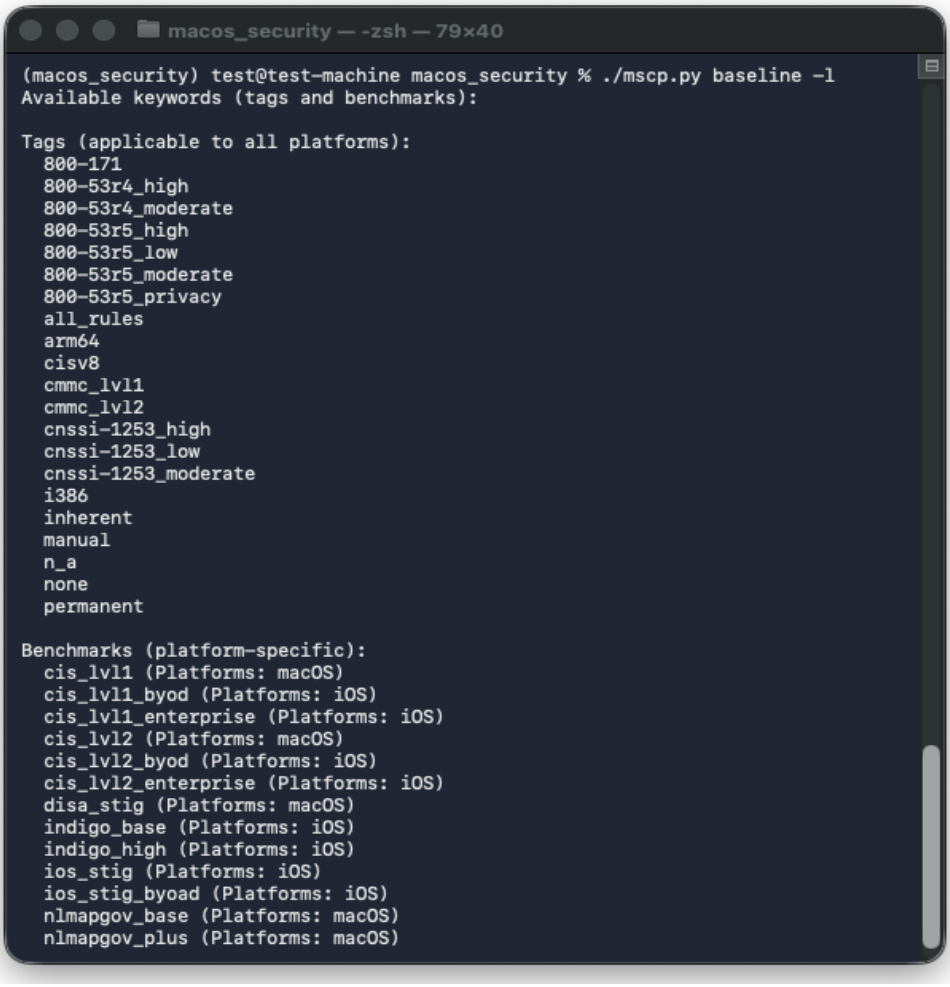

Fig. 4. Viewing the available baselines

Next, choose a baseline, and use the `mscp.py` CLI with the `baseline` subcommand to create a baseline YAML file. Figure 5 illustrates this step for the SP 800-53r5 moderate baseline for macOS 26.

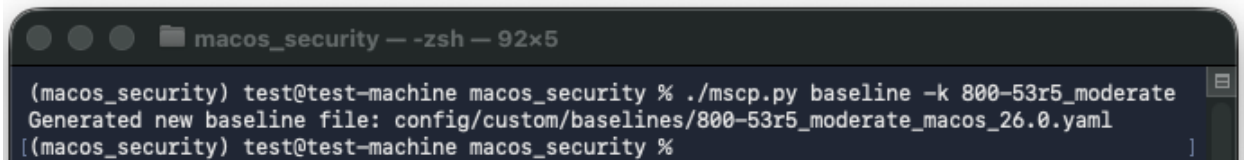

Fig. 5. Generating a baseline

B.2. Localization (Optional)

Once an organization's benchmark has been generated, the mscp.py utility can be used to generate documents in supported languages. The mSCP NIST Pages section describes how to use the Generate Guidance function to generate documents in supported languages.

B.3. Creating the mSCP Compliance Script and Configuration Profiles

Using the `mscp.py` CLI with the `guidance` subcommand, create the mSCP compliance script and configuration profiles, as shown in Fig. 6.

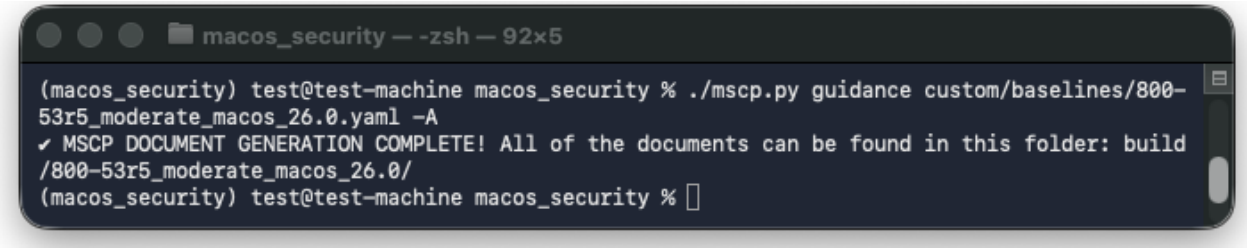

Fig. 6. Generating the compliance script and configuration profiles

B.4. Running a Compliance Scan

The mSCP compliance script must be executed with administrator privileges so that it can access all of the settings, as shown in Fig. 7.

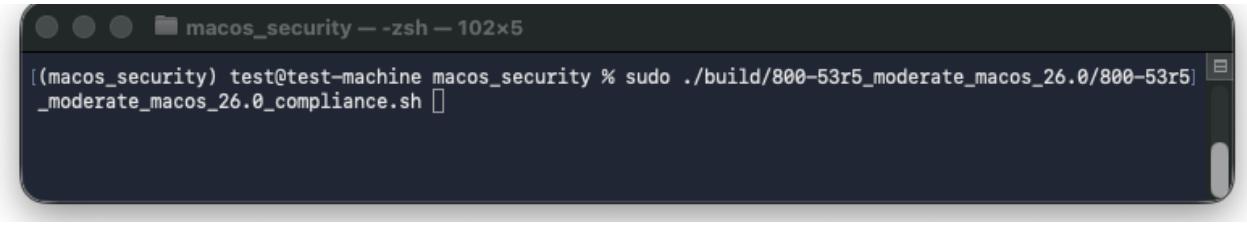

Fig. 7. Running the mSCP compliance script

Figure 8 shows the main menu presented by the mSCP compliance script.

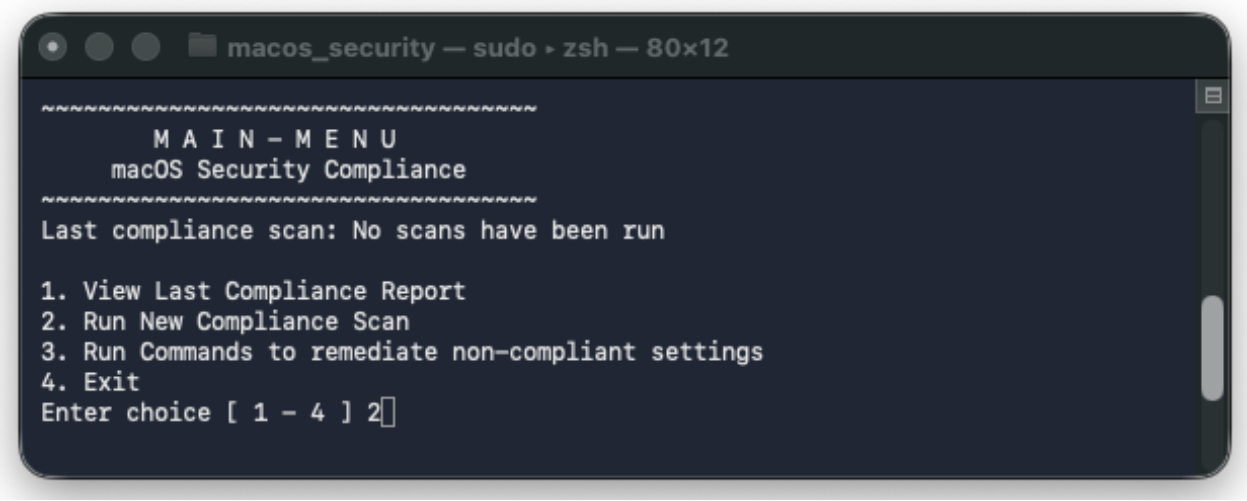

Fig. 8. Selecting "Run New Compliance Scan" from the main menu

Selecting option 2, "Run New Compliance Scan," from the main menu launches the scan.

Figure 9 shows output from the scan, which in this case reflects numerous rule failures, each indicating a deviation from the expected configuration.

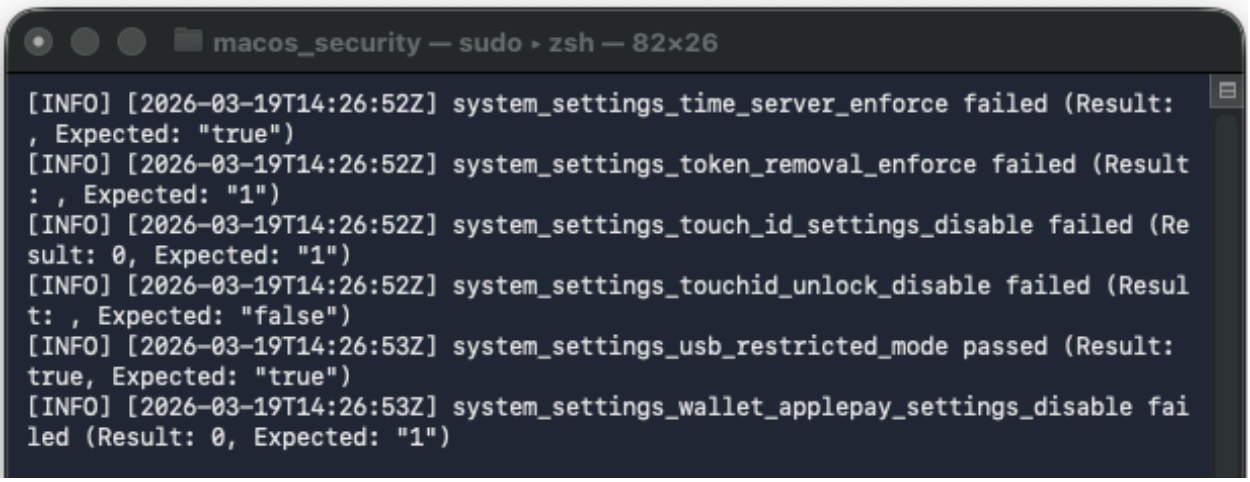

Fig. 9. Compliance scan output

Selecting option 1, "View Last Compliance Report," from the main menu displays a summary of the compliance report results. Figure 10 shows results indicating that 35 tests passed and 140 tests failed for an overall score of 20.00 % compliant.

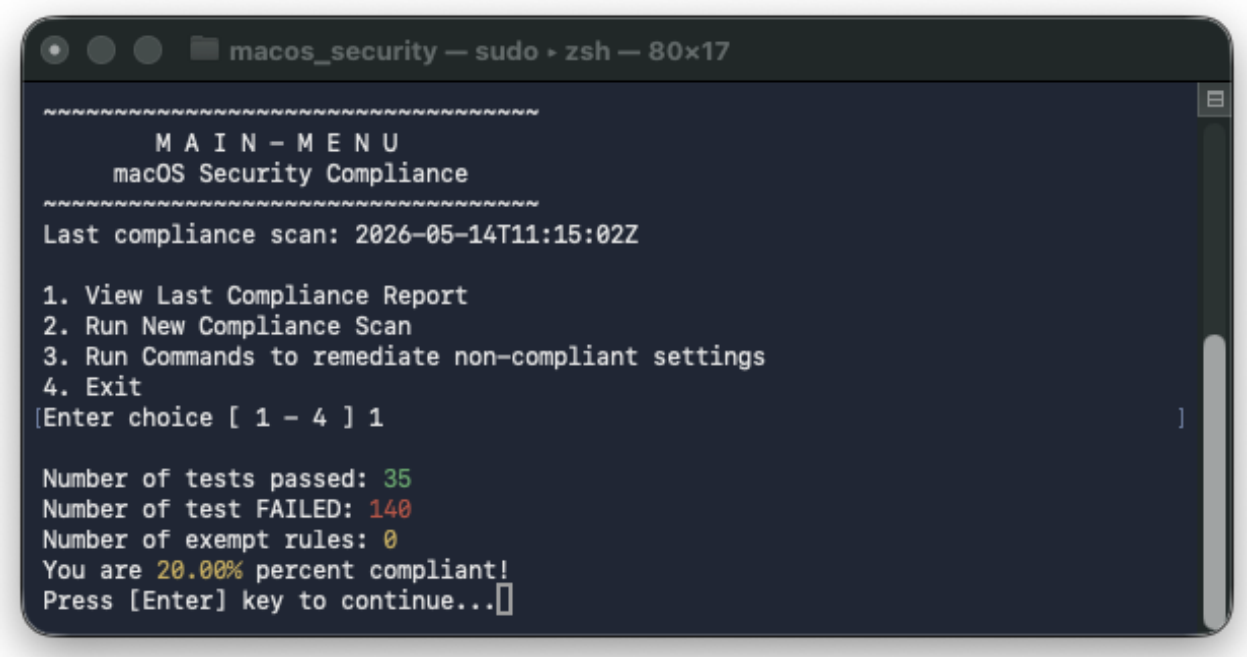

Fig. 10. Viewing a compliance report

B.5. Fixing Non-Compliant Settings

Selecting option 3, “Run Commands to remediate non-compliant settings,” begins the process of fixing non-compliant settings that were discovered during a previous compliance scan. Figure 11 illustrates the disclaimer to be reviewed and accepted before fixes are initiated. This disclaimer indicates the potential risk in applying fixes.

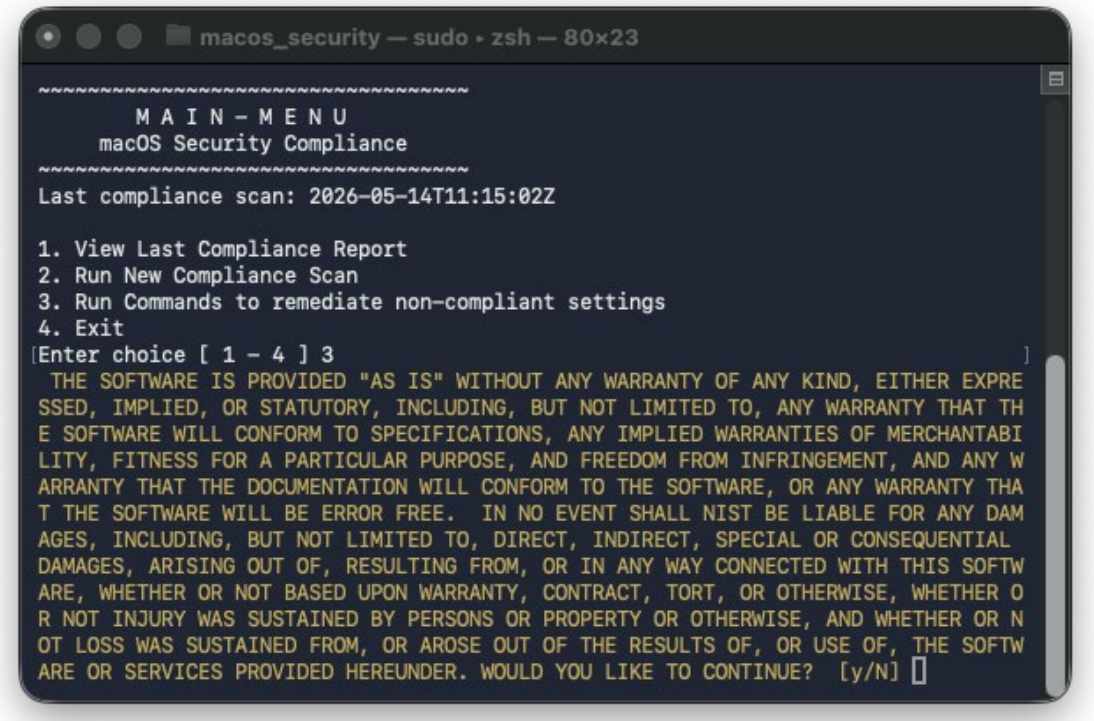

Fig. 11. Disclaimer for non-compliant settings remediation

After the disclaimer statement is accepted, the fixes are applied to the system, as shown in Fig. 12.

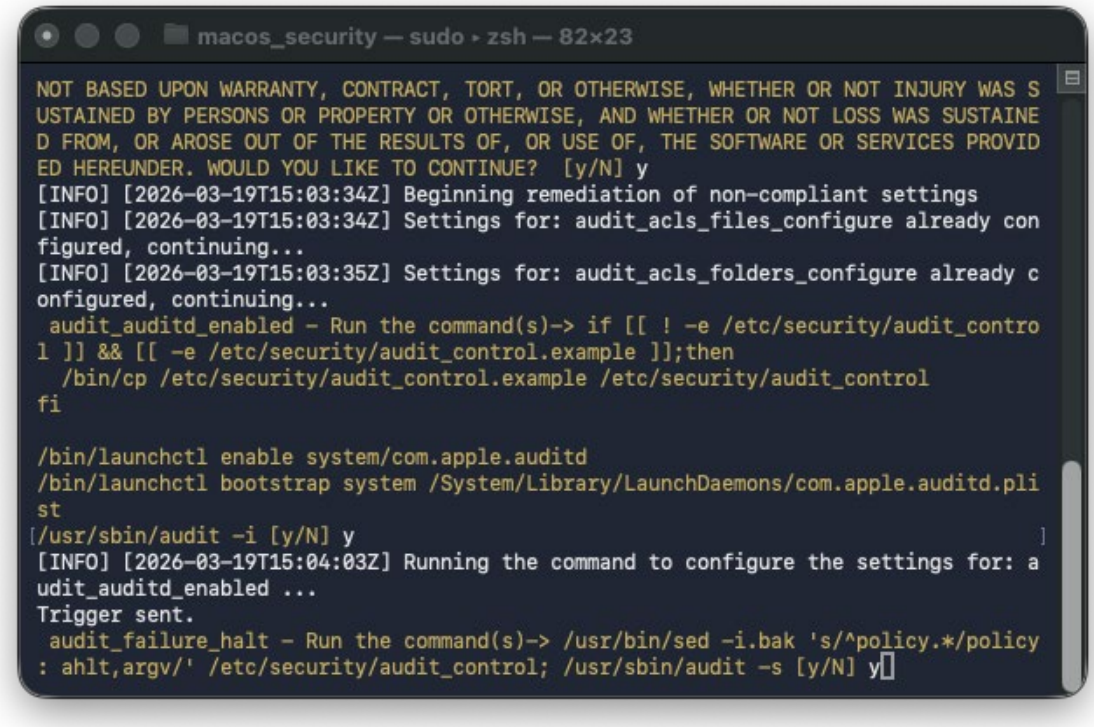

Fig. 12. Interactively configuring settings

Appendix C. Example of Creating a Benchmark Using ODVs

This appendix provides an example of tailoring a baseline to create a custom benchmark using the `mscp.py` CLI.

C.1. Initiating the Tailoring Process

The `mscp.py`'s `-t` option for the `baseline` argument is used to customize and tailor the specified baseline. The utility prompts for a benchmark name, author name, and organization for the benchmark being created, as shown in Fig. 13.

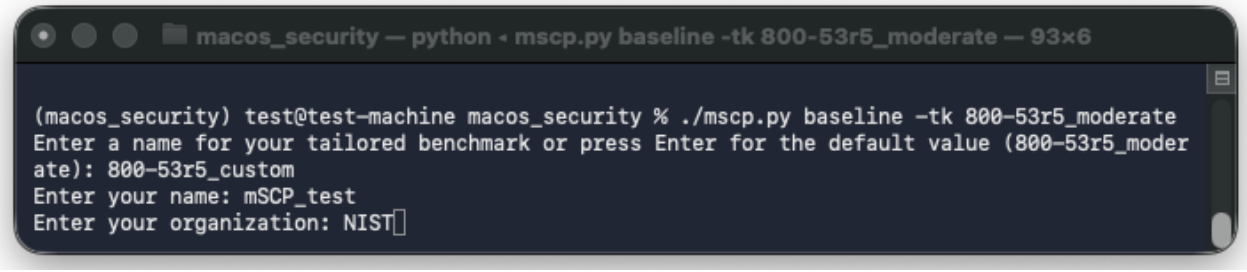

Fig. 13. Prompt for benchmark name

C.2. Including Rules and Specifying ODVs

For each rule that exists in the specified starting baseline, a prompt asks whether the rule should be included in the custom benchmark. Entering a "?" in response to a rule being included will display a description of that rule, as shown in Fig. 14.

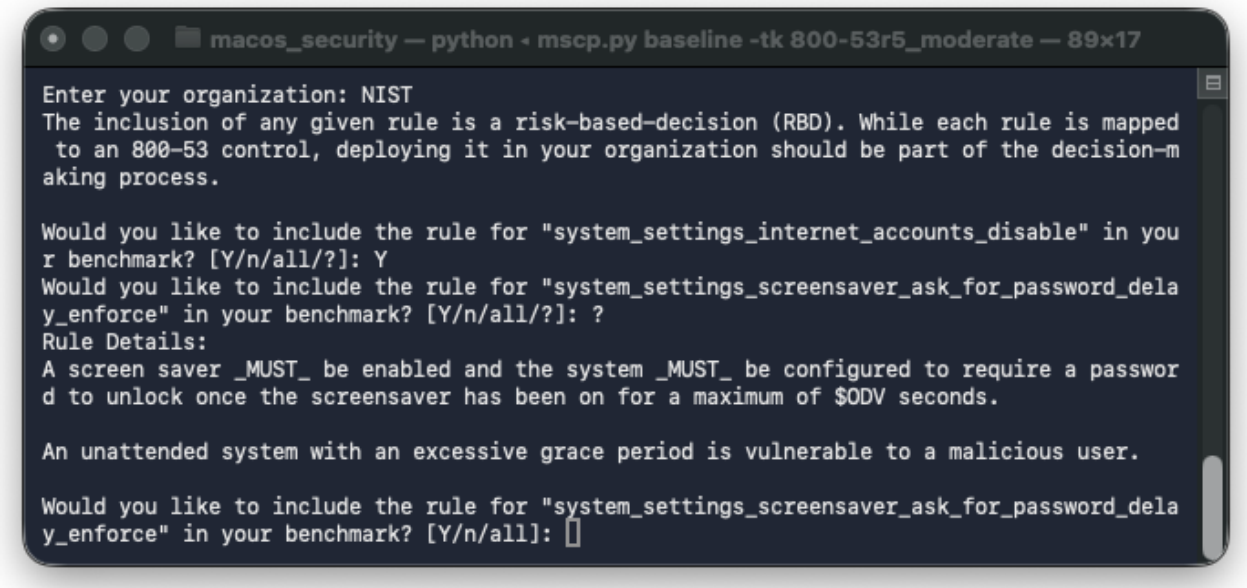

Fig. 14. Prompt for rule file inclusion

After electing to include a rule that accepts an ODV, a prompt asks the user to enter their own value or use the displayed default, as shown in Fig. 15.

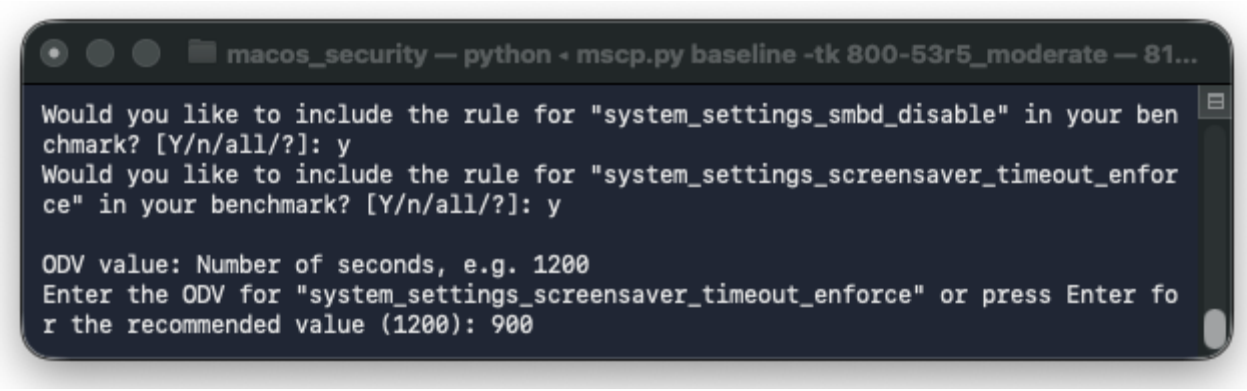

Fig. 15. Rule prompting for an ODV

C.3. Excluding Rules

Rules can be excluded by entering “n” at the prompt. If a rule is selected for exclusion, the user will be asked to provide a reason, which will be included in the generated documentation. Figure 16 demonstrates this process.

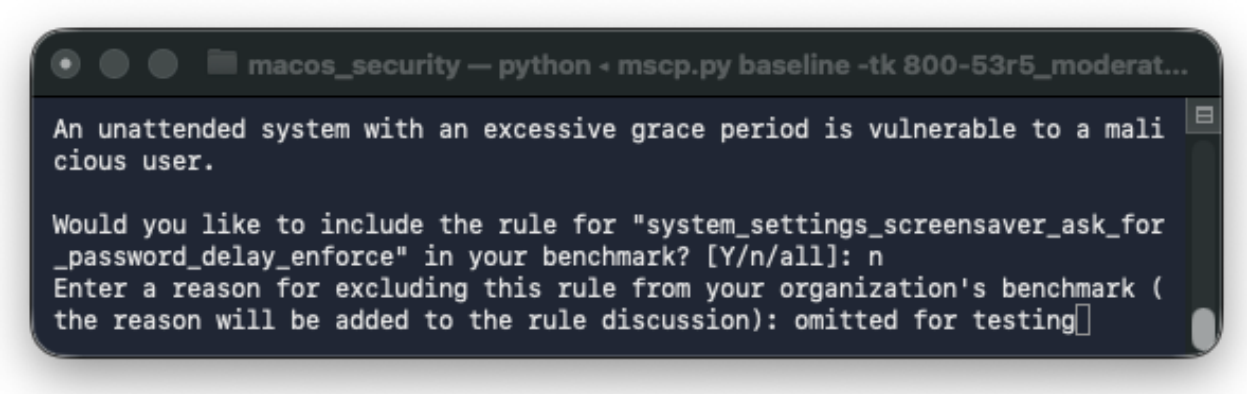

Fig. 16. Excluding a rule

Appendix D. List of Symbols, Abbreviations, and Acronyms

Selected acronyms and abbreviations used in this paper are defined below.

BSI
Bundesamt für Sicherheit in der Informationstechnik (German Federal Office for Information Security)

CCE
Common Configuration Enumeration

CIS
Center for Internet Security

CMMC
Cybersecurity Maturity Model Certification

CLI
Command Line Interface

CNSS
Committee on National Security Systems

CNSSI
Committee on National Security Systems Instruction

DDM
Declarative Device Management

DISA
Defense Information Systems Agency

FIPS
Federal Information Processing Standards

JSON
JavaScript Object Notation

LANL
Los Alamos National Laboratory

mSCP
macOS Security Compliance Project

NASA
National Aeronautics and Space Administration

NIST
National Institute of Standards and Technology

ODV
Organization-Defined Value

OVAL
Open Vulnerability and Assessment Language

916 **SCAP**
917 Security Content Automation Protocol

918 **SP**
919 Special Publication

920 **STIG**
921 Security Technical Implementation Guide

922 **XCCDF**
923 Extensible Configuration Checklist Description Format

924 **YAML**
925 Yet Another Markup Language

Appendix E. Change Log

In June 2026, the following changes were made to this report:

- Moved the discussion on Baselines, Checklists, and Benchmarks to Sec. 1.4
- Added Sec. 1.6 overviewing new mSCP features
- Added Sec. 3.1 explaining the relationship between mSCP rule files and various compliance frameworks
- Updated Sec. 3.2 and subsections with the new rule file structure and usage
- Updated Sec. 3.3 to include a brief description of DDM
- Added Sec. 3.4.5 on the generate JSON manifest capability
- Updated Sec. 3.7 with the current project directories
- Updated Appendix B to use the most recent mSCP syntax
- Added Appendix C.3 explaining rule exclusion
- Removed an appendix detailing an example of mSCP usage by an assessment tool vendor
- Language adjustments throughout to promote clarity
- Added mentions of the other operating systems covered by the mSCP
- Updated screenshots and example listings
- Updated references
- Updated mSCP-related terminology throughout to match current usage
- Reformatted all content to match the latest NIST SP template