Carnegie Mellon University
Software Engineering Institute

A Practitioner’s Guide to Designing and Developing Hands-On Cybersecurity Skilling Continuation Labs

Richard Weise
Christopher Herr
Nicholas Giruzzi

December 2025

TECHNICAL REPORT
CMU/SEI-2025-TR-010
DOI: 10.1184/R1/30648815

CERT® Division

https://www.sei.cmu.edu

Design: REV-03.18.2016.0 | Template: 09.04.2025

This material is based upon work funded and supported by the Department of War under Air Force Contract Nos. FA8702-15-D-0002, and FA870225DB003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM25-1425

Table of Contents

List of Figures

List of Tables

Abstract

Federal cybersecurity professionals face a unique set of threats, risks, regulations, and requirements. The Software Engineering Institute (SEI) leveraged its experience with cybersecurity best practices, federal government guidance and recommendations, and workforce development practices to deliver engaging, specialized training for federal cybersecurity professionals. In partnership with the Cybersecurity and Infrastructure Security Agency (CISA), the Cyber Mission Readiness (CMR) directorate of the CERT Division at the SEI developed a set of Skilling Continuation Labs (SCLs) to provide novel, relevant, and unique hands-on immersive training to upskill the federal cybersecurity workforce. To help support training lab developers, this report draws on CMR team members’ expertise in developing high-fidelity cybersecurity training labs for the federal workforce and provides recommendations and guidelines for developing effective and immersive hands-on cybersecurity labs.

1 Introduction

The Software Engineering Institute has long used engaging design to develop training. Examples include interactive videos, hands-on guided lessons, cyber challenges, and team-building exercises. This training was delivered at conference workshops, capture-the-flag competitions, and teaching events. Participants commonly cite the immersive nature of this training as their favorite part and describe how the training provides tangible resumé builders. As indicated in Cyber Mission Readiness (CMR) team's fact sheet, *Approach to Skilling the Cyber Workforce*, deliberate practice and comprehension validation through immersive, hands-on content are an effective means of building critical capabilities [SEI 2024]. This rehearsal for real-world operations and problem-solving allows defenders to affirm their skill sets and strengthen their overall mission readiness.

Unlike traditional classroom and passive education models (e.g., lectures, videos), immersive hands-on options enable learners to acquire tangible skills and work in groups to solve real-world cybersecurity problems. Immersive training platforms that allow learners to practice concepts in realistic environments can bolster capacity building and upskilling, better equipping cyber defenders to join the workforce and counter the threats facing our nation.

Learners have specific training needs, whether they are cyber operators with baseline technical knowledge, experienced professionals seeking career progression, or anyone simply wanting to acquire new skill sets. Building on users' existing abilities (i.e., *upskilling*) calls for a model that captures learner interest while incorporating new information. The aim of Skilling Continuation Labs (SCLs) is just that—reinforce fundamental knowledge and skills while teaching new skills in emerging tools and techniques.

The recent SCL project with CISA's Cyber Capacity Building group prescribed a series of topic-focused immersive labs. For the existing cyber audience, CISA advertised these labs as being "created for cyber professionals with some IT/cyber experience who fall into the upper-beginner to lower-intermediate cyber skill range […] these labs will provide a continuation of skills for the cyber professional" [CISA 2025a]. Mainly focusing on enterprise IT system cybersecurity, the cybersecurity concepts addressed in the SCL could be applied to operational technology, Internet of Things (IoT), industrial control systems, and integrated systems.

This report describes the methods and best practices that the CMR team used to design and develop the SCLs so that others can emulate them when creating their own hands-on cyber training content.

2 Advancing Skills-Based Training and Assessment Methodologies

2.1 Background

Immersive training provides an engaging and relatable experience, where learners can attain new lesson concepts and practice applying new skills. Unlike traditional classroom methods and passive education models (e.g., lectures, asynchronous video), immersive scenario-based content offers an environment where learners have a familiar context where they can better acquire knowledge and skills. Morrison and Brantner's study on what learning a new job requires found that as much as 70% of adult learning comes from experience [Morrison 1992].

Further, a 2021 Idaho National Laboratory study on cybersecurity skills development asserts, "the lack of adequate training makes it difficult for industry experts to upskill or stay up to date on new advancements within the field of cybersecurity" and that hands-on experience is an underutilized tool for teaching cybersecurity students and professionals [Beason 2021]. The 2025 National Association of Colleges and Employers (NACE) Job Outlook survey supports the notion that candidates and professionals seeking new employment must acquire and improve their skills to meet job requirements [NACE 2024]. NACE found that over 70% of employers surveyed use skills-based hiring most of the time, and more than two-thirds use skills-based hiring practices as early in the process as the interview and screening stages.

A great method for achieving practical experience and experiential learning in cybersecurity topics is to complete hands-on lab exercises in a virtual environment. Learners can perform risky operations, such as penetration and exploitation testing or malware analysis, in a sandbox environment, where mistakes become learning opportunities rather than critical errors. Further, these environments provide learners with a chance to observe telemetry data, detect activities, and rehearse tasks before they deploy to work on a production network. Convenience is an added benefit of using this method, as virtual labs are typically available on demand and are highly accessible, repeatable, and easily scalable.

2.2 Driving Engagement and Consistency

Immersive content strategies continue to evolve, seeking to enhance the foundational goals of creating relatable, repeatable, and engaging experiences to foster learning. Poorly designed learning activities can lead to learner frustration and create a barrier to learning. To maximize effectiveness, the SCL project incorporated the following core tenets:

- **Engage the Learner.** Relatability is critical to learning. Real-world scenarios provide a relatable experience that allows users to apply learning objectives. For example, demonstrating a real-world exploit of a vulnerability within a familiar system or application emphasizes the impact of these vulnerabilities. The learner can easily understand the severity and consequences in a tangible context.

- **Justify the Process.** Do not leave actions unexplained. Reinforce why steps are taken within the lab. This explanation helps the learner understand why each step is important and provides a sense of relatability, increasing learner retention and interest in the material. Lab guides commonly use callouts to emphasize and refresh key facts, terms, and technologies.
- **Continually Reinforce Learning.** Assess the learner's comprehension with knowledge and grading checks throughout the training. This provides the learner with a sense of accomplishment and reassurance that they are following along correctly, which further drives their engagement.
- **Maintain Consistency.** Each lab guide follows a structured template and virtual environment to leverage a standard topology where possible. Consistency among labs ensures that the lab's style and structure is not an obstacle to learning. A consistent structure allows learners to focus on the learning objectives rather than the delivery method.
- **Develop Standalone Scoped Content:** Each tightly scoped learning asset is designed to be a standalone lesson with its own learning objectives. In this way, prerequisite training is not required, and the granular strategy makes customization and content updates easier.

These tenets apply to each phase of the lab design and development process to provide consistent, engaging, and immersive training content.

2.3 Acquire, Apply, Affirm

Taken together, these strategies illustrate a broader upskilling philosophy advocated by the SEI of "**Acquire, Apply, Affirm**," a progression designed for professionals who already possess foundational skills and desire structured opportunities to improve those skills or to learn new ones. Where the military training model has coined the phrase "crawl, walk, run," it doesn't lend itself to upskilling those who already have foundational, novice-level skills (i.e., already *walking)*. Topic-focused, hands-on, guided labs with assessments take users through the phases of acquiring, applying, and affirming the lesson's skills. An SCL includes unguided mini-challenges, where learners solve a new problem based on the skills they learned and applied during the guided walkthrough of the lab. By completing guided walkthroughs and mini-challenges, learners acquire new knowledge while applying new concepts and skills that are further complemented with layered assessments to affirm comprehension.

2.4 Standard Network Topologies

Using a standard topology increases lab development velocity while also bringing consistency to the learning experience and reducing resource constraints. Developers use only the required pieces of the topology for each lab. Network devices (e.g., routers and firewalls, network addressing schemes, sensors, standard client systems) can be easily slotted in and reused without reconfiguring them or building them from scratch.

2.5 The Skills Hub

The lessons the CMR team has learned from its years of content development leveraging virtual devices and simulations has led to advancements, efficiencies, and best practices. The CMR team designed a single authoritative lab management server for the President's Cup Cybersecurity

Competition, known as the “Challenge Server” [SEI 2025a]. This server allows developers to configure startup and grading scripts and hosts in-lab artifacts from a centralized website within the virtual environment. The CMR team used these functions to minimize the number of changes required from the standard topology, reduce storage needs, speed up development time, and increase environment reliability. Variables can also be injected during the start-up phase to dynamically alter artifacts, increasing the training’s repeatability and ensuring that no two instances are identical. The “Challenge Server” was rebranded as the “Skills Hub” for the SCL project and enhancements were made to improve its core functionality and add features to support learning.

3 Lab Development Methodology

3.1 Development Workflow

Lab development comprises the following six core components:

1. The lab topic must be selected and researched to determine its viability.
2. The developer must determine the learning objectives, outcomes, and the lab outline.
3. The developer must build the supporting virtual environment, install the necessary tools, and set up the necessary services that support the lab.
4. The developer must walk through the lab objectives in the final environment and draft the step-by-step guided walkthrough.
5. The lab must be reviewed and tested for accuracy and functionality.
6. The lab can be provided as an open source lab or published.

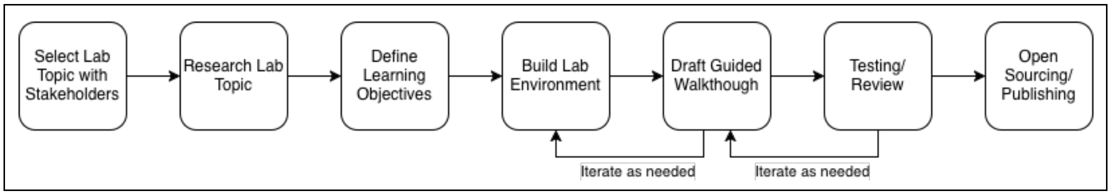


Figure 1: Lab Development Workflow

The two biggest time sinks in the lab development process are building the environment and authoring the guided walkthrough. A 2021 study on how long it takes to create training found that the average time required to create a roughly 20-minute full engagement (i.e., game, scenario, or simulation) training item was 155 hours [Defelice 2021]. Each SCL requires 30 to 90 minutes of learner engagement time to complete, depending on the density of the material, as some labs are simpler than others. Therefore, anyone might expect that a fully functioning SCL would take a developer weeks, if not months, to design and build.

However, the CMR team saved time and costs during development by implementing efficiencies, such as using standard lab environment topologies, document templates, assessment procedures, and the authoritative Skills Hub to host and drive lab artifacts. These efficiencies ensure that developers are not “reinventing the wheel” for each lab. Combined with their wealth of experience creating hands-on immersive training scenarios, individual CMR developers were able to complete each SCL in roughly four weeks once a topic was chosen. Of course, how long it took to develop each SCL depended on the complexity of the material, including quality assurance, testing, and review.

3.2 Standard Virtual Network Topology

Consistency is crucial when creating learning labs. One method of ensuring consistency is by using a standard topology. Maintaining the same underlying infrastructure—whether a lab is designed to be introductory, intermediate, or advanced—allows the learner to focus on the skills being taught without needing to relearn key environmental details.

Starting from a standard topology also benefits lab content developers. With operating systems pre-installed and the base networking established, developers can start lab creation immediately. This reduces development time and lessens the number of inconsistencies when multiple labs are being constructed concurrently by multiple developers. Standard topologies also encourage experimentation. With the ability to quickly create a base environment, developers can test new ideas when establishing a lab's training path. If an item does not work as expected, the amount of development time lost is minimized.

Developers can customize their labs by adding to or removing from the standard topology. They also determine whether learners can access a virtual machine console. In some cases, virtual machines are hidden from the learner to simulate adversaries. In the case of the Skills Hub, the system is intentionally restricted to ensure the integrity of assessments. Generally, limiting the number of available consoles is beneficial because it reduces confusion about which "console" learners should be connected to at a given time.

The SCLs were built using TopoMojo, an open source topology creation tool that is an integrated part of the SEI's Crucible framework [SEI 2025b, 2025c]. TopoMojo allows developers to create virtual machine templates that they can quickly and easily add to a Workspace to create a stock topology [SEI 2025d]. When a lab is created and launched from the standard topology, a temporary "scratch" disk is created from the base template disk. Creating temporary disks in this way provides storage savings, since only a single instance of the base disk is required to support concurrent launches of the same virtual machine.

Using base disks helps to simplify ongoing maintenance. When a virtual machine requires an update (e.g., adding a new tool, adding and applying a certificate), the changes need to be committed only to the base template disk. Since the lab virtual machines use the base disk when they are deployed, every lab that uses that template is automatically updated to point to the most recent version. This is significantly less labor-intensive than performing updates manually across multiple lab environments.

The SCL environments leverage open source operating systems to make them as broadly accessible as possible. These environments replicate real-world topologies and include firewalls, routers, and security appliances in addition to user workstations. Network segmentation is used to separate systems, and multiple environments can be contained within a single Workspace. The topology a developer creates depends on the needs of the learners and the requirements of the lab being built.

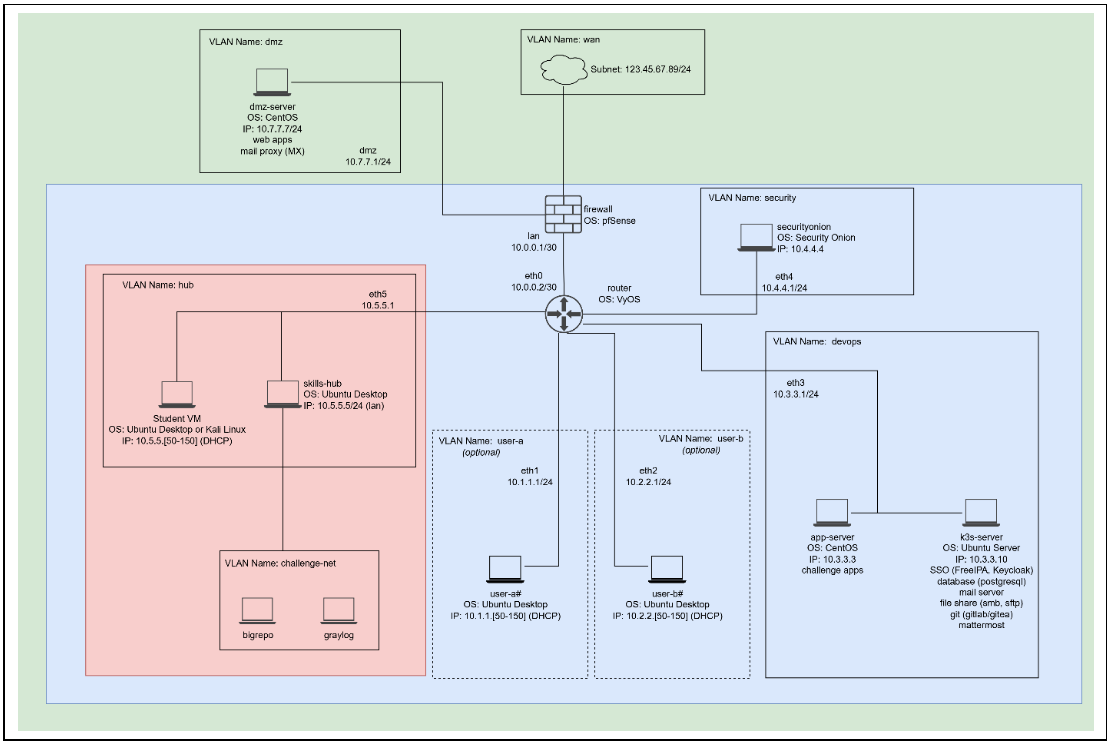
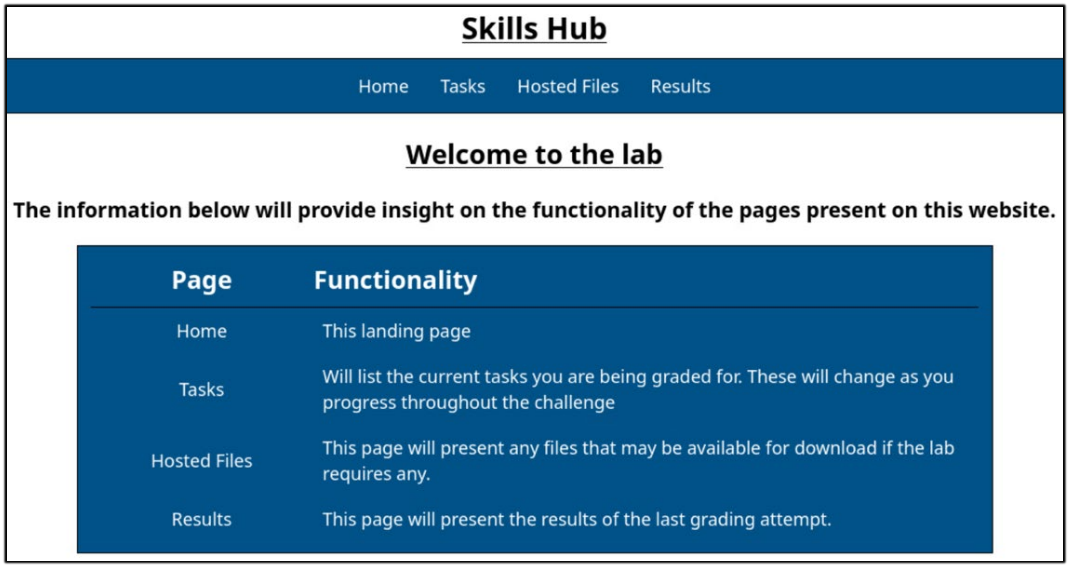


Figure 2: Standard Lab Topology Diagram

3.2.1 Skills Hub

Every lab contains an authoritative Skills Hub server. Learners do not have console access to the server; instead, they access it using a website within the lab environment. There, they can download required files, access in-lab bookmarks, and trigger checks to track their progress through the lab. When a check fails, output from the Skills Hub server provides feedback to help the learner identify and correct mistakes.



Skills Hub

Home Tasks Hosted Files Results

Welcome to the lab

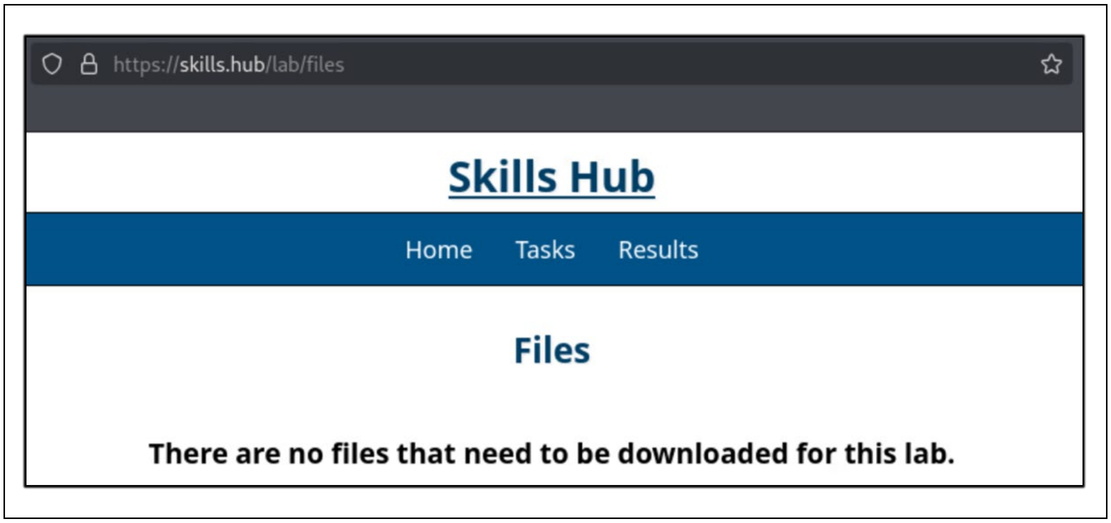
The information below will provide insight on the functionality of the pages present on this website.

Page	Functionality
Home	This landing page
Tasks	Will list the current tasks you are being graded for. These will change as you progress throughout the challenge
Hosted Files	This page will present any files that may be available for download if the lab requires any.
Results	This page will present the results of the last grading attempt.

Figure 3: Skills Hub Main Page

For developers, the Skills Hub acts as a central authoritative server and repository. All lab health and grading scripts are triggered by and run from the Skills Hub server. These scripts are configured to log assessment results and errors to the Skills Hub so that a record of the learners' progress is maintained. The Skills Hub can also act as an aggregator to collect logs from the virtual machines within the lab environment. Having these logs in a central location greatly speeds up troubleshooting during lab development and production.

Not every lab requires all the features available in the Skills Hub. For example, a lab may not have any files to download. Each developer can customize the Skills Hub by updating a YAML file to disable the features that are not being used to avoid confusion by the learner.



https://skills.hub/lab/files

Skills Hub

Home Tasks Results

Files

There are no files that need to be downloaded for this lab.

Figure 4: Skills Hub – Removing Unnecessary Services

3.3 Determining Training Objectives

Training content developers have more expertise than the SCLs' intended learners; thus, the developer's assumptions and generalizations can result in content that is either too advanced or too simplistic. One way to humanize the audience (i.e., the learner) is to create a persona, grounding the lab design process in real needs and expectations. A persona is not a single individual; rather, it is a composite representation of the type of learner most likely to engage with the material. Reflecting on who the learners are, what knowledge they bring, their skill level, and what they hope to achieve ensures that labs are tailored to the learner's actual starting point rather than developer assumptions. This alignment increases both the lab's engagement and effectiveness.

During SCL development, the project team employed the backwards course design framework[1] and used SMART criteria[2] to assess learning objectives. This means that developers identified and articulated the lab's expected results and used them as the target during the build process. There is no shortage of topics within cybersecurity. Working collaboratively with mission partners and stakeholders and equipped with a persona, developers can make choices about learning priorities. Developers use collaborative discussions to jointly (with mission partners and stakeholders) consider the needs and gaps that the training content, lab, or other content aims to address.

Cybersecurity labs can be grouped into two categories:

- **Training cybersecurity labs** are structured to cultivate a learner's skills through practice and progression. These labs are designed to teach or enhance specific skills that learners already have.
- **Teaching cybersecurity labs** introduce a learner to a subject or topic they have little to no previous experience with. For example, some SCLs show the anatomy of specific cyber attacks in a step-by-step process, allowing learners to see firsthand how they are carried out.

The differences between these two categories lie mostly in their approach to conveying information and the scope of their content. The design process is roughly identical for each. These lab categories and their minor differences are defined in more detail in Section 3.5.

Documenting tasks and objectives on a storyboard helps to visualize how the learner will progress through the lab. By laying out the sequence of events, developers can ensure each stage of the lab aligns with the learner's needs and the final learning objective. This approach also helps developers anticipate moments where the learner might struggle or where additional details might be needed. The storyboard provides the developer with an opportunity to adjust pacing and content. It is easier to adapt, revise, and refine a storyboard than it is to rebuild an entire lab. Storyboards are typically written to show the high-level tasks a learner will complete during each phase of the lab.

[1] A backward course design framework starts with the desired end state and works backwards to derive learning objectives based on the tasks necessary to reach those goals.

[2] See Section 3.6.1 to learn more about SMART criteria.

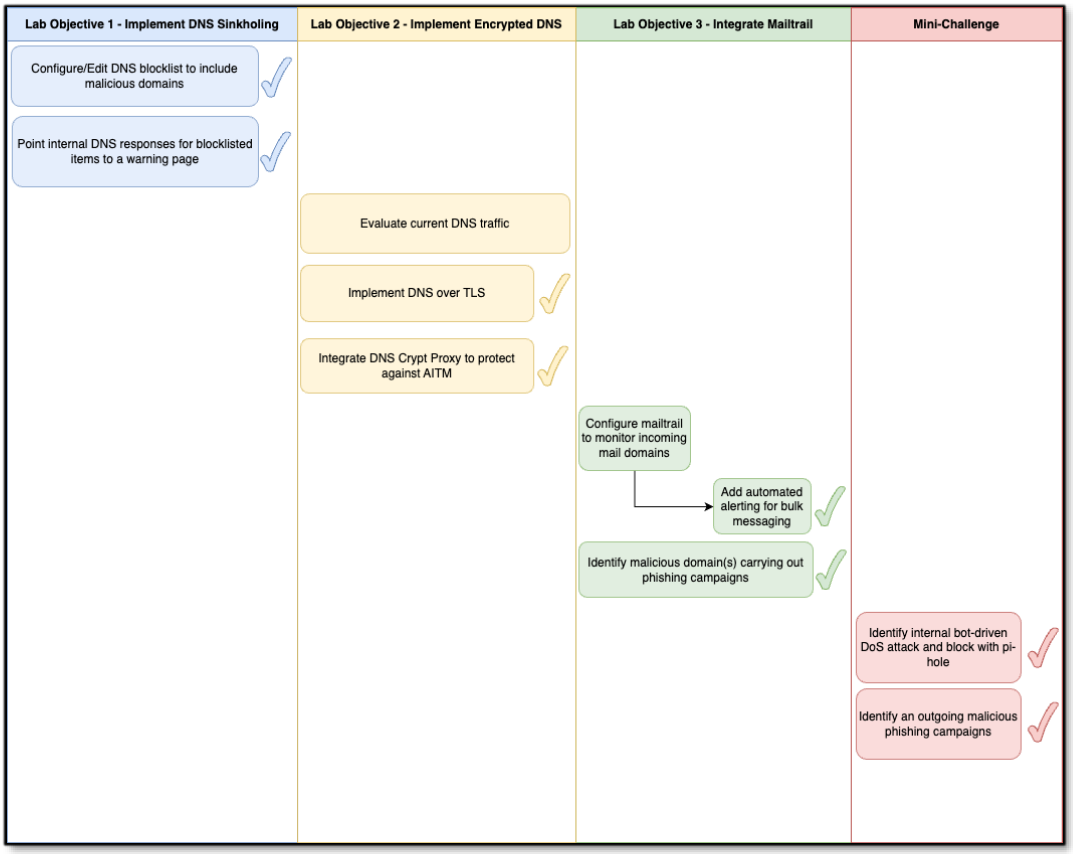

Lab Objective 1 - Implement DNS Sinkholing
Lab Objective 2 - Implement Encrypted DNS
Lab Objective 3 - Integrate Mailtrail
Mini-Challenge
Configure/Edit DNS blocklist to include malicious domains
Point internal DNS responses for blocklisted items to a warning page
Evaluate current DNS traffic
Implement DNS over TLS
Integrate DNS Crypt Proxy to protect against AITM
Configure mailtrail to monitor incoming mail domains
Add automated alerting for bulk messaging
Identify malicious domain(s) carrying out phishing campaigns
Identify internal bot-driven DoS attack and block with pi-hole
Identify an outgoing malicious phishing campaigns

Figure 5: Lab Roadmap

In the example storyboard outlined in Figure 5, the lab was initially designed with three main objectives and a mini-challenge. In the final version of the lab, some objectives were reordered to achieve a better flow, and some tools were swapped out as part of the iterative development process, but the core objectives of the lab remained the same. The objectives covered in the lab prepared the learner for the mini-challenge at the end, where they apply what they learned to solve a related problem.

3.4 Targeting Applicable Framework Alignments and Skills

Each SCL was mapped to the *Federal Civilian Executive Branch (FCEB) Operational Cybersecurity Alignment (FOCAL) Plan* strategic document and the *NICE Workforce Framework for Cybersecurity (NICE Framework)* [CISA 2024, NICCS 2025]. Determining which FOCAL areas and NICE work roles were applicable to each SCL was typically a backwards-oriented process.

Alignment with the FOCAL plan primarily targeted the following areas: vulnerability management; defensible architecture, including zero trust principles, and incident detection and response. Developers also identified the NICE Framework tasks and skills that most closely aligned with the actions and objectives completed within a lab. Then, the appropriate work roles were assigned based on the applicable tasks and skills and the developer's judgement.

Each lab clearly states the applicable FOCAL plan areas and NICE work roles, skills, and tasks. With these alignments, learners can easily focus on labs that address the skills and tasks they wish to target. Leveraging existing frameworks helps to identify supporting tasks that relate to the lab's learning objectives, and any framework can be applied to immersive capacity-building content.

3.5 Lab Design

The structure of a lab directly impacts the learner's ability to understand and successfully complete it. Once a lab's objectives are determined, the next step is to deconstruct the content and determine how best to present it throughout the guided walkthrough. As mentioned before, there are two main categories of labs: training labs and teaching labs. Depending on its purpose, a lab's content and structure may differ slightly.

3.5.1 Teaching Labs vs. Training Labs

Teaching labs assume that the learner has little or no knowledge of the subject matter. Introducing learners to these new concepts and skills is the primary goal. These labs tend to be tightly focused on a single topic area, and they use knowledge checks and simple completion checks to assess the learner's understanding of the content rather than their ability to apply new skills to a problem. For this reason, most teaching labs do not include mini-challenges.

Training labs are meant to enhance the learners' existing skills. The learner already has a baseline understanding of the concepts, techniques, and tools being used. The lab focuses on teaching new concepts and skills that build on that foundational knowledge, while also assessing the learner's understanding and progress along the way. These labs are designed to be progressive (i.e., each phase builds on the previous one).

In addition to the guided walkthrough, training labs like the SCLs typically conclude with a mini-challenge. A mini-challenge is a culminating assessment, where the learner completes a set of objectives with minimal guidance. The goal of a mini-challenge is to assess a learner's ability to apply what they have learned to a new problem. Mini-challenges require learners to synthesize a solution that is drawn from their previous work in the training lab.

3.5.2 Phases and Scoring

Each lab is split into multiple phases. This modular approach allows labs to be "chunked" into more digestible bites that address primary objectives and smaller supporting tasks in each phase. Each phase consists of multiple microlearning pieces, and each lab consists of one or more phases. These smaller, easier-to-process microlearning pieces make it easier for learners to retain information, reduce the required cognitive load, and alleviate learner fatigue throughout the lab. Lab phases are either sequential—working from start to finish in chunks toward a singular overarching goal—or they highlight individual, correlated, standalone topics tied to a larger topic area.

All SCLs used a standard total point value of 1,000. For training labs, the mini-challenge was assigned a static 300-point value (i.e., 30% of the total points). Remaining points were divided among the different phases. The 70/30 breakdown allowed for granular reporting on how many learners were completing only the guided walkthrough portion of the lab and how many attempted and passed the mini-challenge. The total and percentage of points per phase is not as important as making sure that each lab has a comparative total. This comparative total allows content developers to gauge the learner's completion and progression equally among all labs by reviewing the scores of multiple learners across multiple lab sessions.

Each lab phase includes some combination of knowledge checks or grading checks. Weighted point values are assigned to each question, and these values are totaled to calculate the point value the learner achieved in each phase. By requiring prerequisite scores to advance to the next phase of the lab, learners are gated to ensure they have the necessary understanding and skill to advance. These gates also guarantee that the state of the environment is as expected (i.e., correct) before the learner progresses to the next portion of the guided walkthrough.

Using scoring allows content developers, training team personnel, and learners to gauge the immersive lab training's effectiveness, participation rate, and success. While learners aren't taking the lab primarily for scores, these scores allow them to assess themselves on a known scale and quantifiable measure of improvement.

3.6 Incorporated Assessments

3.6.1 Assessment Design

While traditionally used for setting goals or objectives, assessments use SMART criteria in every lab. The following list describes the components of SMART criteria:

- **Specific.** Assessments are tied to learning objectives or key points that are critical for learners to understand to successfully complete the lab.
- **Measurable.** Assessments measure how well the learner can identify key information or complete key tasks related to the lab.
- **Achievable/Attainable.** All tasks and assessed items in the lab are addressed within the lab material (i.e., the learner can complete the lab with only the information provided within that lab). Solution guides are provided for the mini-challenges to assist learners.
- **Relevant.** Assessments align with the learning objectives of the lab. At a more granular level, assessments of key objectives occur immediately following that step in the guided walkthrough or mini-challenge.
- **Time-Bound.** Assessments must be completed during a specific phase. Only after correctly answering the assessment questions can the learner progress forward in the lab. Additionally, the lab defines the time limit or expected duration of the lab. Labs should not be taken by learners indefinitely.

3.6.2 Knowledge Checks

Knowledge check questions are used to assess and reinforce learner understanding of the lab material. However, not all questions are created equal. The following are examples of bad, good, and better questions and why they are judged that way:

A **bad question:** “What tool/command did you just run?”

> This question does not assess a learner’s understanding of the material, only the learner’s recall of the step they just took in the lab. It also does not require the learner to interact with the lab to correctly answer the question. This is a question asked for the sake of asking a question.

A **good question**: “What tool can be used to analyze network traffic at the packet level?”

> This question assesses the learner’s recall of the learning objective material and requires them to understand the tools available to them in the lab. It requires more thought than the bad question, but it has the same deficiency. The learner can answer the question without interacting with the lab.

A **better question**: “What filter option can be used in Wireshark to view only the packets that correspond to web traffic coming from 10.10.10.100?”

> This question requires the learner to understand how Wireshark works and determine the correct filter for the scenario provided. The learner can use trial and error within the lab environment or consult reference material to find and verify a solution. Having this question precede another question about the content of the filtered packet helps ensure the learner has the required knowledge to progress.

3.6.3 Grading Checks

Grading checks ensure that learners have completed the assigned tasks successfully and that the lab is in the correct state to advance to the objective. When a grading script is created, it validates that the objective was achieved versus simply checking that specific actions were taken. For example, if tasked with correctly configuring and running a service, the grading check should validate that the service is accessible, not simply that it is running as a process or that a port is open. More details on grading checks are provided in the following sections.

3.6.4 Flexibility and Integrity Enforcement

Assessments are as permissive as possible without jeopardizing the integrity of the validation. Knowledge checks should ignore the learner’s capitalization, white space, and special characters that may be a result of a typographical error if the remaining string is accurate. In instances where submission strings must have a specific format, such as a file path or IP address, the lab instructions must provide guidance on the expected format.

Assessment developers also need to be aware of the ways a learner can circumvent the grading script. For example, if the learner is expected to write a firewall rule to block a specific traffic type from one host to another, they must not be able to pass the check by simply disabling a network interface instead. Obviously, this is not the desired outcome for the lab, and it would leave the lab in an incorrect state. For this reason, additional checks are conducted to ensure the environment is still operating as expected, in addition to performing the desired check.

It can be difficult to address cheating, shortcutting, or brute force methods every time they occur. In general, if brute forcing a solution would take as much effort and demonstrate the same level of skill as performing the intended solution, you can usually ignore the unintended workaround. However, an important goal in lab design is to discourage cheating and shortcutting. One way to meet that goal is by providing all the necessary tools and guidance within the lab to support learner success.

3.6.5 Assessment and Grading Check Feedback

Proper feedback is as important as the checks themselves. When the learner triggers a grading check, they believe they have completed the task correctly. If the grading check determines the task has not been completed correctly, it is essential that the learner receives specific feedback on why the check failed. This information reduces the likelihood that they will get overly discouraged and provides a path to resolution. The larger and more complex the task is, the more critical this feedback becomes in helping the learner identify and correct the error.

Concrete examples of grading check scripts and feedback are provided in Appendix A.

3.7 Crafting Effective Guided Walkthroughs

Consistent and proper formatting and styling is an essential part of instructional design. Step-by-step guides, such as lab guides or manuals, need to be precise and easy to follow. The CMR team members used previous templates for lab guides as a basis for creating SCL lab guides and incorporated improvements based on feedback and best practices they learned from their past work in this space.[3] Example guides are available on CISA's prescup-challenges GitHub repository [CISA 2025b].

The CMR team authored mini-challenge solution guides as well as guided walkthroughs, but they included only the steps needed to solve the mini-challenge objectives. Appendix B provides a list of SCL lab guide components and the best practices used to develop each one.

3.8 Quality Assurance and Testing

An important final step in creating training content is testing and validation. Each SCL went through several layers of quality assurance and testing. Initial testing was performed by experienced non-technical staff members who represented true novice learners. The CMR team believed that if these non-technical staff members could easily follow along, digest the material, and complete the labs as written, then the target audience of upper beginner to lower intermediate learners should do just fine.

These initial reviews identified deficiencies that ranged from simple typographical errors in the written guide to confusing lab processes or issues with virtual machines or applications. Once developers corrected these deficiencies, the same reviewers would validate the updates and retest the lab from scratch.

[3] *The CMR team chose Markdown as the language for the SCLs' web-based training platforms. Using Markdown makes it easier to publish documentation on open source platforms such as GitHub. The same formatting decisions could likewise be applied to other word processing tools and applications.*

The labs were required to pass multiple linting checks. Documentation and technical writing experts conducted multiple layers of review to ensure that each guide was appropriately concise and had a consistent tense, voice, and tone.

Finally, before the labs were published, a technical lead reviewed each lab step-by-step to ensure it had a consistent format, styling, and structure. This review also included testing the final version of the lab itself by progressing through learner steps. The technical lead provided the viewpoint of a technical expert who might think of things the developer missed and that a non-technical reviewer might not realize is important. Examples of items this reviewer might identify include methods for circumventing a grading check, an alternate solution to a mini-challenge that was not accounted for in the grading script, and unintentional actions that could lead to issues later in the lab.

Each lab underwent multiple rounds of testing and validation before it was sent to mission partners and stakeholders for review.

4 Future Training Improvements

The CMR team continually strives to improve its immersive training development and delivery processes. After every project, developers discuss what improvements should be made to support future projects. Following the SCL project, the CMR team recommended the following improvements:

- **Skills Pathways.** The CMR team is currently in the process of adding skills pathways to the Crucible platform to support roadmaps of content and increase a learner's skill level in specific areas of cybersecurity or in certain cybersecurity roles. Adding skills pathways would walk learners through a progression of items of increasing difficulty, allowing them to advance towards skills mastery.
- **Bookmarking Session Status.** Currently, a learner must complete a lab in a single session. While using lab phases allows learners to tackle tasks in segments, learners must restart the lab and repeat work they had previously done if their work did not complete the entire session. With additional development time, labs that require sequential phases could be scripted to stage the environment to the final state of the highest phase completed by the learner. Labs with sequential or isolated phases would store a learner's progress so they would not need to repeat previously completed assessment items.
- **Support Chat Agent.** Tools for on-demand learner-initiated support and guidance could help learners when they are stuck. Leveraging logs and large language models (LLMs), the CRM directorate could create a chat agent capable of answering user questions and providing guidance and hints.
- **Self-Configured Environment.** An environment that learners can configure for performing cyber skills training would allow the learner to select from a list of skills to practice, while the environment that supports those skills is deployed automatically. Skills-based injects and activities could be applied to the environment, and supporting scenario documentation could be generated with LLM assistance.

5 Where to Find More

The SCLS are published in CISA's prescup-challenges/skilling-continuation-labs GitHub repository [CISA 2025c].

At the time of writing this report, the repository consisted of the following 19 labs, but this list will grow as more work is completed. (A * denotes an SCL with mini-challenges.)

- Automated Defenses*
- Detection Rules in Logging Made Easy*
- DNS and Name-based Security Solutions*
- Fast Flux Attacks and Defensive Measures
- Hardening Kubernetes: Pod Security*
- Incident Response with Velociraptor*
- Introduction to SCADA Using Modbus and BACnet
- Living Off the Land Attacks*
- Log Managements with Logging Made Easy*
- Network Segmentation with ICS or HMI*
- Phishing Mitigation with MFA
- Secure Programming*
- Secure Programming with Rust*
- Secure Programming: Dependency Supply Chain Attacks
- Weaponized Archive Files: Extract at Your Own Risk
- Weaponized .svg Files: The Hidden Threat in Vector Graphics
- Web Application Penetration Testing with Brute Force Attacks
- Web Application Penetration Testing with Cross Site Scripting*
- XZ Utils: A Case Study of Supply Chain Trust

Federal civilians and Department of War (DoW) personnel can play through the SCLs in the CISA Practice Area on its President's Cup Cybersecurity Competition Website [CISA 2025d].

Appendix A: Grading Checks

Grading Checks confirm that a learner has successfully completed a portion of the lab and that the lab is in a correct state. While there are common elements present in most grading scripts, each is unique and dependent on what is being checked. The walkthrough in this appendix demonstrates Grading Check 1 from Phase 1 of the Network Segmentation with ICS/HMI lab, which is available at CISA's prescup-challenges/skilling-continuation-labs GitHub repository [CISA 2025c]. This walkthrough will not provide the exact code used; instead, it will go through the approach taken when constructing this grading check.

Starting from the desired end state and working backwards is often the best way to construct a grading script. In Figure 6, Grading Check 1 verifies that the local password policy was updated correctly, as this is the first step in properly securing the system from unauthorized access and ensuring password policy guidelines and best practices are followed.

From the lab guide, learners are instructed to update a line of configuration code in the `/etc/pam.d/`common-password file to include `password requisite pam_pwquality.so retry=3 maxrepeat=3 minlen=8`. This code enforces a limit of three failed logon attempts before a timeout is implemented and requires that all passwords have a minimum length of eight characters. The grading check needs to verify that those configurations are present in that configuration file.

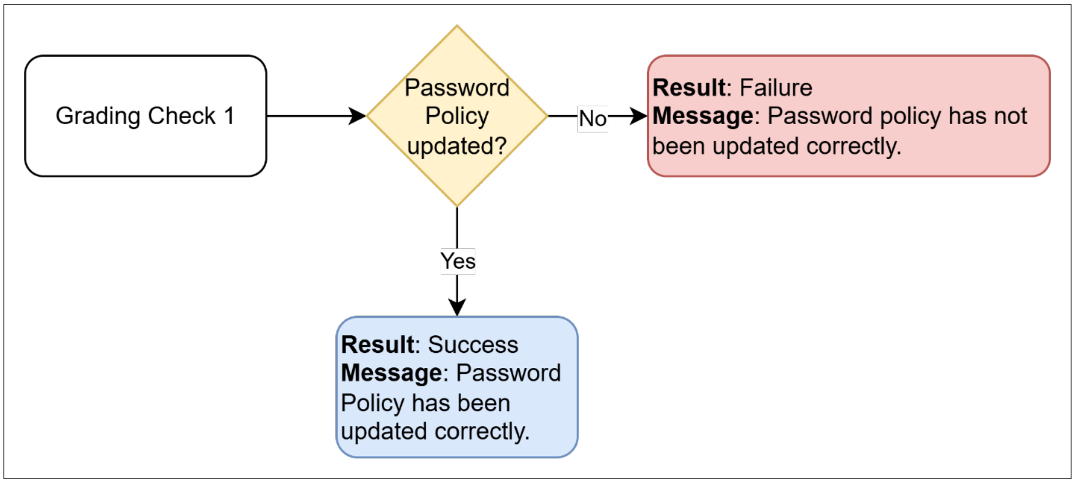

Grading Check 1
Password Policy updated?
No
Result: Failure
Message: Password policy has not been updated correctly.
Yes
Result: Success
Message: Password Policy has been updated correctly.

Figure 6: Grading Check Workflow Part 1

Instead of parsing the entire configuration file, the script focuses on what changed. The sed tool captures the specific line of interest by using the command `sed -n "25p" /etc/pam.d/common-password`. This command will print only line 25 of the configuration file. Comparing this output with the expected output, the script can verify that the configuration was updated correctly.

This command allows the lab to check the configuration locally, but it must run the check from the authoritative Skills Hub server so that all grading attempts are properly logged. Connecting via Secure Shell (SSH) allows the server to run the command remotely to capture the configuration.

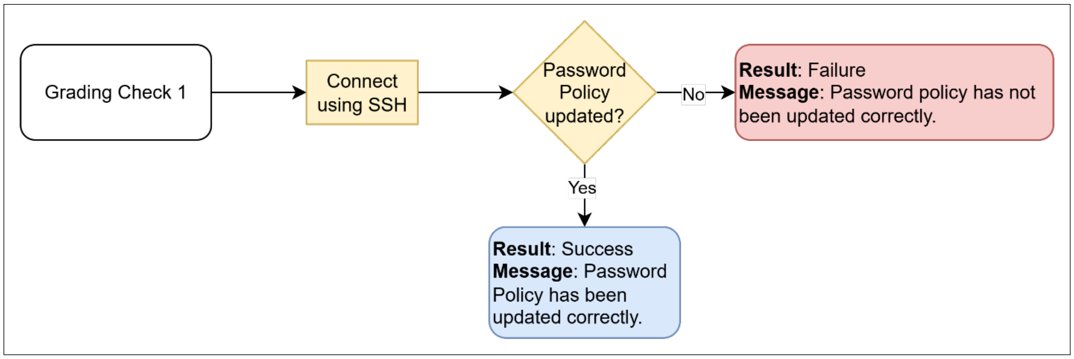

Grading Check 1
Connect using SSH
Password Policy updated?
No
Result: Failure
Message: Password policy has not been updated correctly.
Yes
Result: Success
Message: Password Policy has been updated correctly.

Figure 7: Grading Check Workflow Part 2

The grading script now connects to the server via SSH and runs a command to capture line 25 of the `/etc/pam.d/common-password` configuration file. If the configuration matches what is expected, the learner receives a success message, and the grading check advances to the next stage. If not, they receive a failure message.

By adding the SSH connection to the process, the check introduces a new element to account for in the grading script. If the SSH connection fails, the learner will not get any feedback, even if they implemented the configuration correctly. The script must account for this failure so that the learner receives feedback to let them know what needs to be corrected to advance the lab.

A further modification to the grading workflow adds a condition where a failure message is provided if the SSH connection fails. Another added condition reports a failure message if an exception is raised, providing more robust output to the learner.

The updated grading workflow now looks like this:

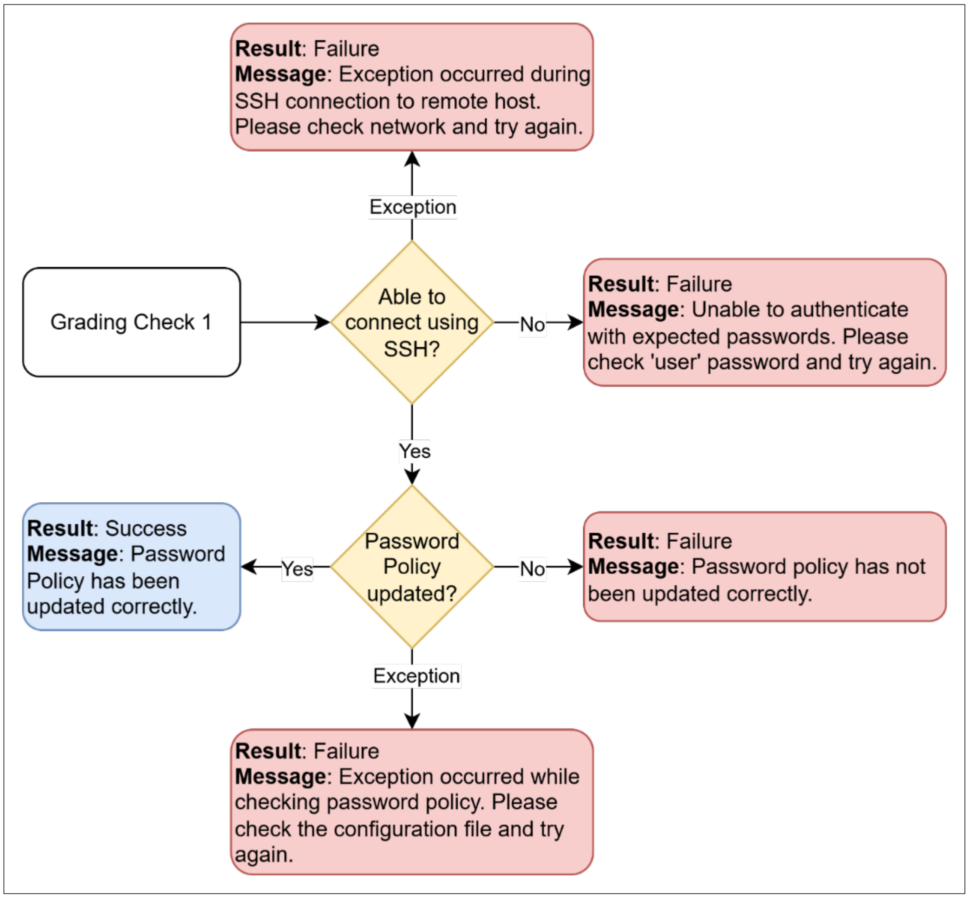

Result: Failure
Message: Exception occurred during SSH connection to remote host. Please check network and try again.
Exception
Grading Check 1
Able to connect using SSH?
No
Result: Failure
Message: Unable to authenticate with expected passwords. Please check 'user' password and try again.
Yes
Result: Success
Message: Password Policy has been updated correctly.
Yes
Password Policy updated?
No
Result: Failure
Message: Password policy has not been updated correctly.
Exception
Result: Failure
Message: Exception occurred while checking password policy. Please check the configuration file and try again.

Figure 8: Grading Check Workflow Part 3

Now that all possible outcomes are accounted for, this portion of the grading check can be considered complete. In this lab, the later stages of the overall grading check verify that the password was updated to match the new policy, that the Human Machine Interface (HMI) server was accessible over SSH, and that local firewall rules were implemented so that only the expected host is able to make this connection.

For more details, complete grading scripts for SCLs are available in the CISA SCL GitHub repository [CISA 2025c]. The specific grading script from this example is provided on the CISA GitHub repository [CISA 2025e].

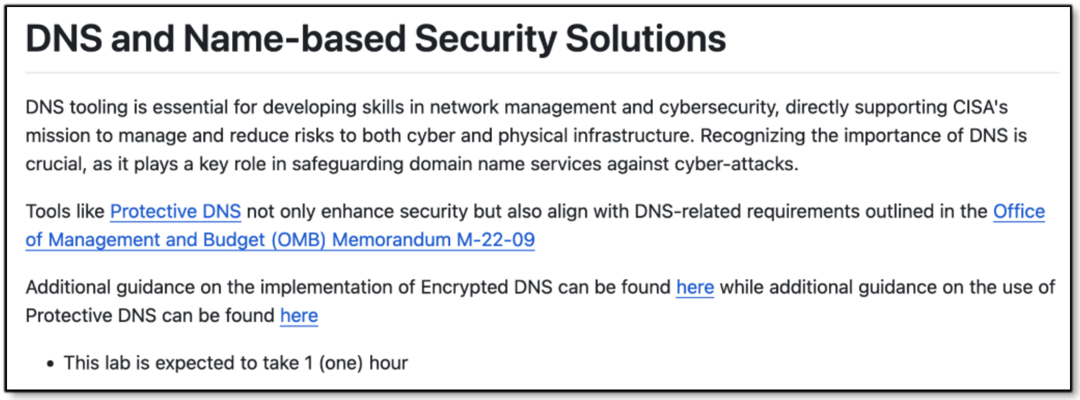
Appendix B: Lab Guide Components

This appendix lists the components of each lab guide and some best practices for developing each one.

Lab Guide Components

Introduction

The first component, the introduction, that appears in each SCL lab (as shown in Figure 9) includes a statement that underpins the lab topic by emphasizing its relevance and necessity. These impact statements include supporting information drawn from reputable sources to strengthen the need for understanding the skills included in the lab. These statements further engage the learner prior to beginning the lab, while the linked references allow the learner to gather more context about the topic. Learners can then decide whether the lab aligns with their training goals and understand what the expected outcomes will be. The introduction also includes the estimated time to complete the lab, so that learners can allot the necessary time before beginning.



DNS and Name-based Security Solutions

DNS tooling is essential for developing skills in network management and cybersecurity, directly supporting CISA's mission to manage and reduce risks to both cyber and physical infrastructure. Recognizing the importance of DNS is crucial, as it plays a key role in safeguarding domain name services against cyber-attacks.

Tools like Protective DNS not only enhance security but also align with DNS-related requirements outlined in the Office of Management and Budget (OMB) Memorandum M-22-09

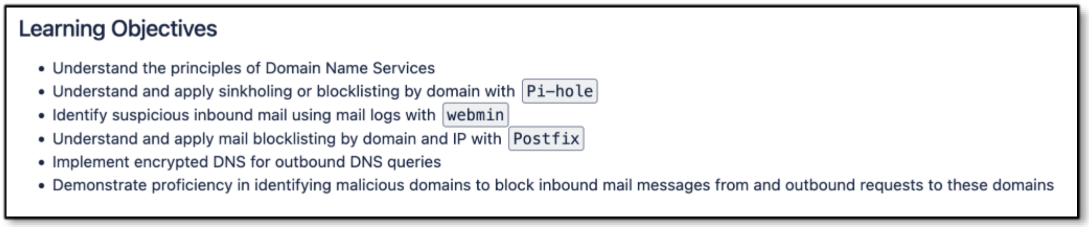
Additional guidance on the implementation of Encrypted DNS can be found here while additional guidance on the use of Protective DNS can be found here

- This lab is expected to take 1 (one) hour

Figure 9: Lab Guide Introduction

Learning Objectives

This component (as shown in Figure 10) lists the learning objectives to provide an outline of the major and minor tasks the learner will perform to complete the lab. These objectives build off the impact statement in the lab's introduction. Learning objective statements are restated in the guided walkthrough documents using Bloom's Taxonomy.



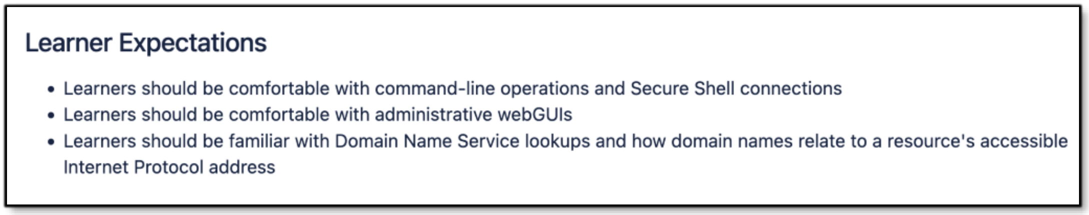
Learning Objectives

- Understand the principles of Domain Name Services
- Understand and apply sinkholing or blocklisting by domain with `Pi-hole`
- Identify suspicious inbound mail using mail logs with `webmin`
- Understand and apply mail blocklisting by domain and IP with `Postfix`
- Implement encrypted DNS for outbound DNS queries
- Demonstrate proficiency in identifying malicious domains to block inbound mail messages from and outbound requests to these domains

Figure 10: Lab Guide Learning Objectives

Learner Expectations

This component (as shown in Figure 11) outlines the prerequisite knowledge or experiences the learner is expected to have before beginning the lab. For example, one such expectation is that "Learners should be comfortable with command-line operations." As the intended audience for these SCLs are professionals in the upper beginner to lower intermediate range, lab guides assume that the learner knows the basics of using a command prompt or terminal. These guides can be tailored to each individual lab depending on the tasks and objectives to be performed.



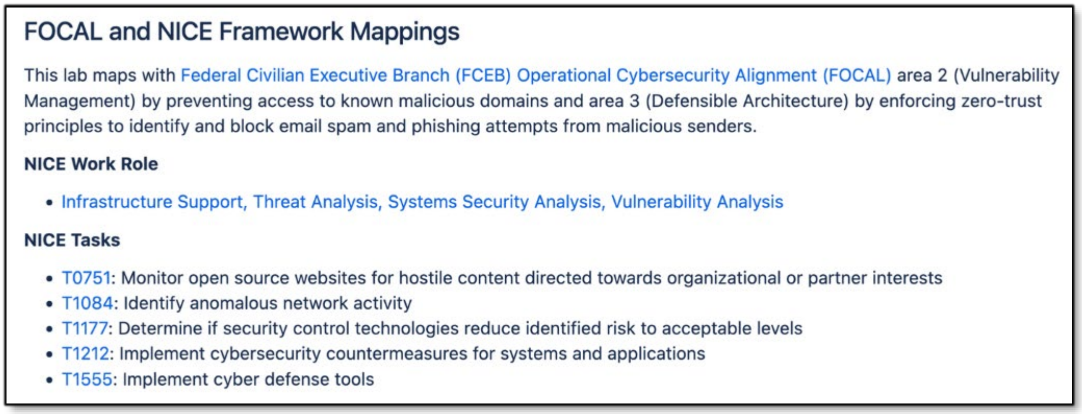
Learner Expectations

- Learners should be comfortable with command-line operations and Secure Shell connections
- Learners should be comfortable with administrative webGUIs
- Learners should be familiar with Domain Name Service lookups and how domain names relate to a resource's accessible Internet Protocol address

Figure 11: Lab Guide Learner Expectations

Framework Mappings

This component of the lab guide (as shown in Figure 12) lists the FOCAL Plan areas and NICE Framework work roles and tasks that align with the learning objectives and training tasks performed in the lab [CISA 2024, NICCS 2025]. Skills are mapped at the end of the lab guide as part of the conclusion.



FOCAL and NICE Framework Mappings

This lab maps with Federal Civilian Executive Branch (FCEB) Operational Cybersecurity Alignment (FOCAL) area 2 (Vulnerability Management) by preventing access to known malicious domains and area 3 (Defensible Architecture) by enforcing zero-trust principles to identify and block email spam and phishing attempts from malicious senders.

NICE Work Role

- Infrastructure Support, Threat Analysis, Systems Security Analysis, Vulnerability Analysis

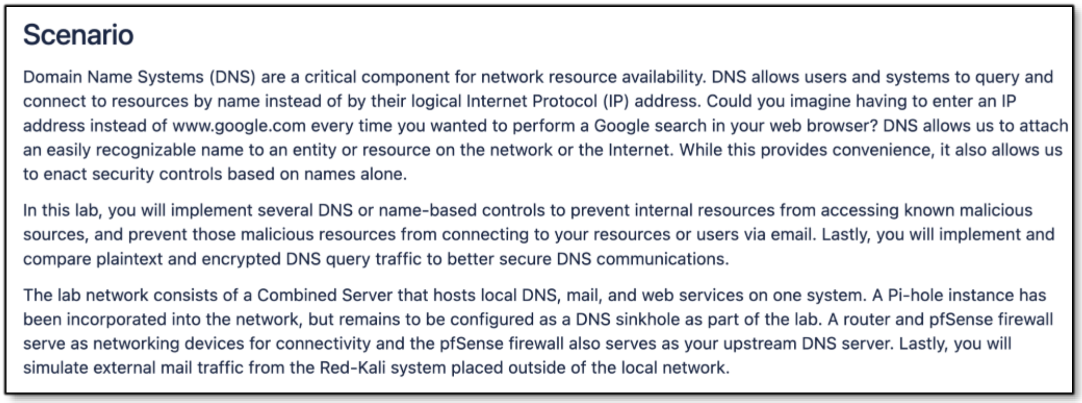
NICE Tasks

- T0751: Monitor open source websites for hostile content directed towards organizational or partner interests
- T1084: Identify anomalous network activity
- T1177: Determine if security control technologies reduce identified risk to acceptable levels
- T1212: Implement cybersecurity countermeasures for systems and applications
- T1555: Implement cyber defense tools

Figure 12: Lab Guide Framework Alignments

Scenario

This component (as shown in Figure 13) outlines a scenario (if provided), introduces the learner to the lab environment and describes the various parts of that environment and the purpose of each.



Scenario

Domain Name Systems (DNS) are a critical component for network resource availability. DNS allows users and systems to query and connect to resources by name instead of by their logical Internet Protocol (IP) address. Could you imagine having to enter an IP address instead of www.google.com every time you wanted to perform a Google search in your web browser? DNS allows us to attach an easily recognizable name to an entity or resource on the network or the Internet. While this provides convenience, it also allows us to enact security controls based on names alone.

In this lab, you will implement several DNS or name-based controls to prevent internal resources from accessing known malicious sources, and prevent those malicious resources from connecting to your resources or users via email. Lastly, you will implement and compare plaintext and encrypted DNS query traffic to better secure DNS communications.

The lab network consists of a Combined Server that hosts local DNS, mail, and web services on one system. A Pi-hole instance has been incorporated into the network, but remains to be configured as a DNS sinkhole as part of the lab. A router and pfSense firewall serve as networking devices for connectivity and the pfSense firewall also serves as your upstream DNS server. Lastly, you will simulate external mail traffic from the Red-Kali system placed outside of the local network.

Figure 13: Lab Guide Scenario

Network diagrams (as shown in Figure 14) are included as a visual representation of the lab environment, including network addressing schemes, host names, and operating systems.

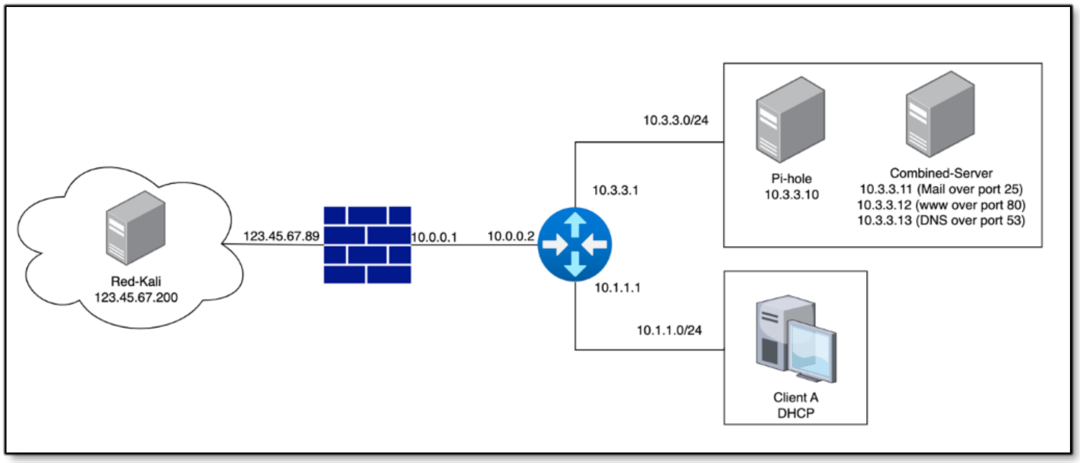
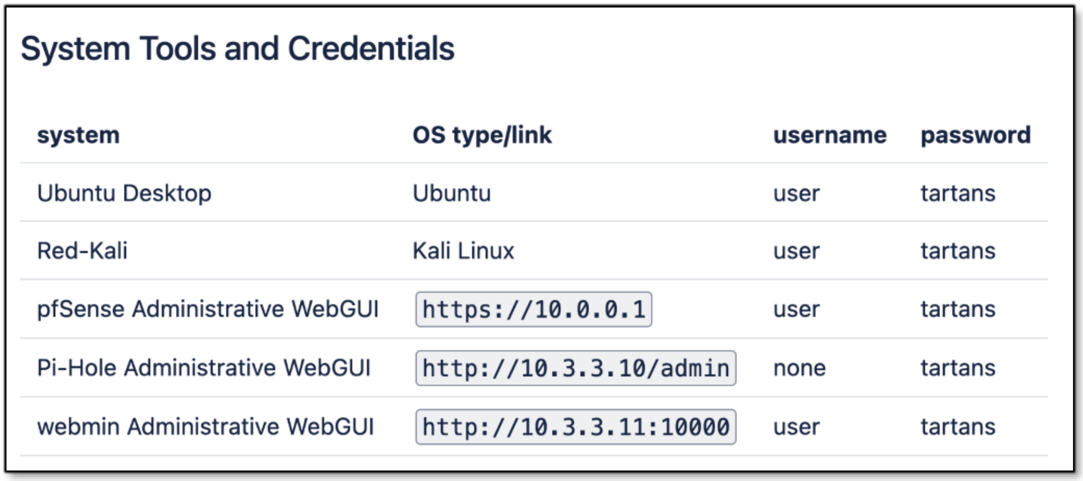


Figure 14: Lab Guide Network Diagram

System Tools and Credentials

A table provides a list of systems and applications (as shown in Figure 15) that the learner will interact with throughout the lab. Information such as system name, operating system or application link, usernames, and passwords are a minimum requirement for this list. This list should only include the items that users must directly access, not every credential used within the environment.



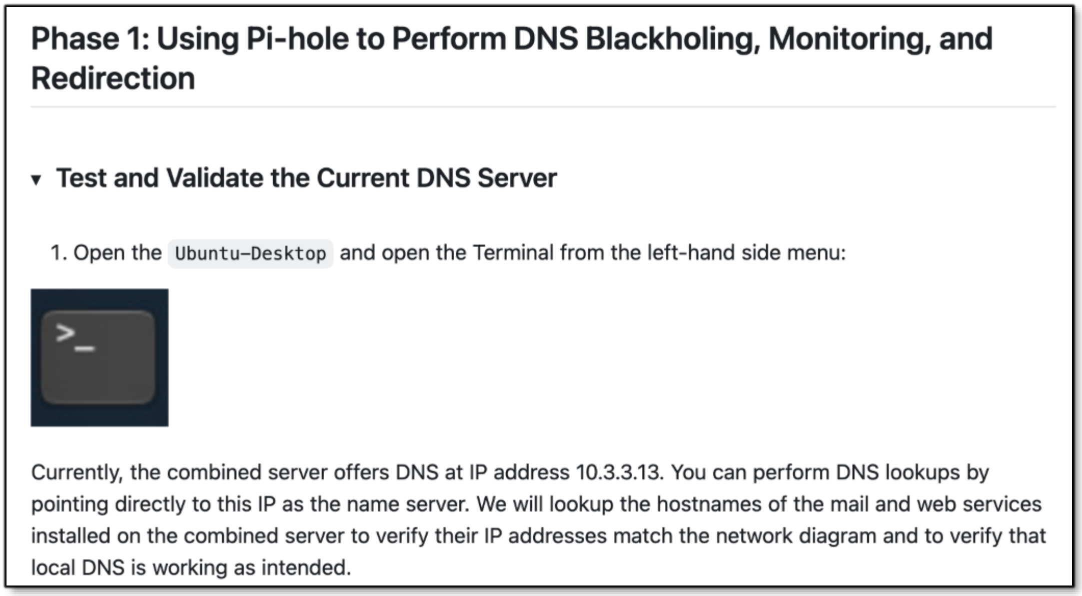
System Tools and Credentials

system	OS type/link	username	password
Ubuntu Desktop	Ubuntu	user	tartans
Red-Kali	Kali Linux	user	tartans
pfSense Administrative WebGUI	`https://10.0.0.1`	user	tartans
Pi-Hole Administrative WebGUI	`http://10.3.3.10/admin`	none	tartans
webmin Administrative WebGUI	`http://10.3.3.11:10000`	user	tartans

Figure 15: Lab Guide Credentials

Step-by-Step Instructions

Lab tasks and instructions (as shown in Figure 16) comprise the principal component of the lab guide. These tasks and instructions are broken up into phases or sections by major task, and subsections may represent smaller subtasks that support the major tasks. The methodology for what constitutes a phase, major task, minor task, etc. are outlined in Section 3.5.2. The step-by-step instructions make up the bulk of the lab guide.



Phase 1: Using Pi-hole to Perform DNS Blackholing, Monitoring, and Redirection

▾ **Test and Validate the Current DNS Server**

1. Open the `Ubuntu-Desktop` and open the Terminal from the left-hand side menu:

Currently, the combined server offers DNS at IP address 10.3.3.13. You can perform DNS lookups by pointing directly to this IP as the name server. We will lookup the hostnames of the mail and web services installed on the combined server to verify their IP addresses match the network diagram and to verify that local DNS is working as intended.

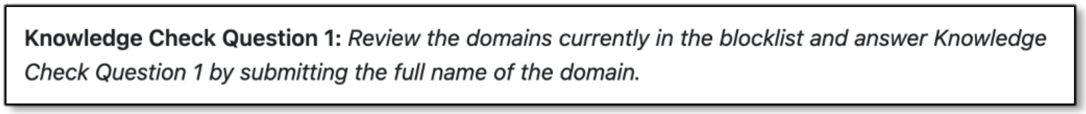
Figure 16: Lab Guide Instructions

Screenshots

The step-by-step portion of the lab guide includes screenshots to help the learner affirm they are looking in the correct place or have achieved the intended results. Visual support greatly enhances the effectiveness of learning and helps break up long “walls” of text where it is easy for them to lose their place.

Knowledge Checks

Knowledge checks are components (as shown in Figure 17) that are included as part of the step-by-step process of completing the lab. Knowledge checks are written in the guide at the point where the learner can answer the associated question. These well-positioned knowledge checks ensure that the learner assesses their comprehension and validates their progress as they go. Answers, once submitted, can then be supplied to the appropriate endpoint, whether an SEI Gameboard [SEI 2025e], external Learning Management System (LMS) or Learning Record Store (LRS) content item, or via document submissions.

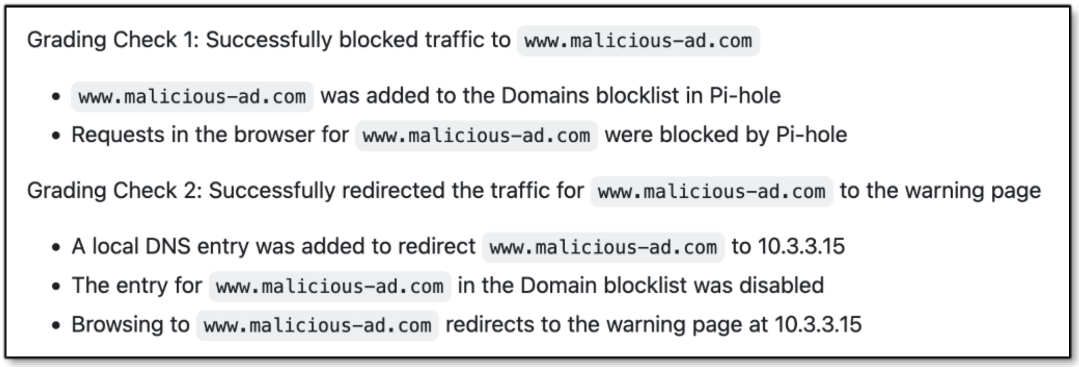


Knowledge Check Question 1: *Review the domains currently in the blocklist and answer Knowledge Check Question 1 by submitting the full name of the domain.*

Figure 17: Lab Guide Knowledge Check

Grading Checks

Grading checks are components (as shown in Figure 18) that are included as part of the step-by-step process of completing the lab. Grading checks include information about how to run the check, what the check is asking for, and the conditions for passing the check. Grading checks appear before the learner progresses to the next task or as a culminating assessment gate at the end of a major task. Flags received by the grading check can then be supplied to the appropriate endpoint, whether SEI Gameboard, external LMS/LRS content item, or via document submissions.



Grading Check 1: Successfully blocked traffic to `www.malicious-ad.com`

- `www.malicious-ad.com` was added to the Domains blocklist in Pi-hole
- Requests in the browser for `www.malicious-ad.com` were blocked by Pi-hole

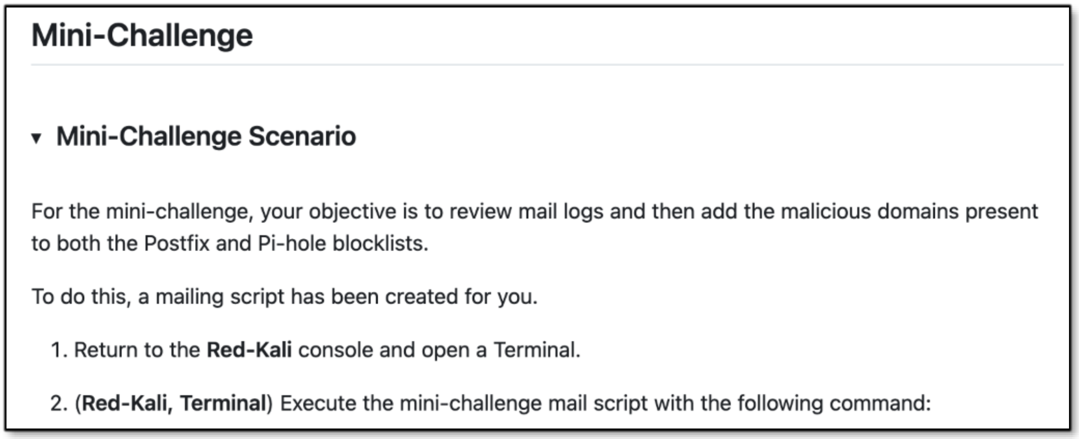
Grading Check 2: Successfully redirected the traffic for `www.malicious-ad.com` to the warning page

- A local DNS entry was added to redirect `www.malicious-ad.com` to 10.3.3.15
- The entry for `www.malicious-ad.com` in the Domain blocklist was disabled
- Browsing to `www.malicious-ad.com` redirects to the warning page at 10.3.3.15

Figure 18: Lab Guide Grading Checks

Mini-Challenge

For SCLs that include a mini-challenge (as shown in Figure 19 and Figure 20), this component outlines the tasks, conditions, and standards for solving the mini-challenge. Any instructions to initiate the mini-challenge are included in the lab guide.



Mini-Challenge

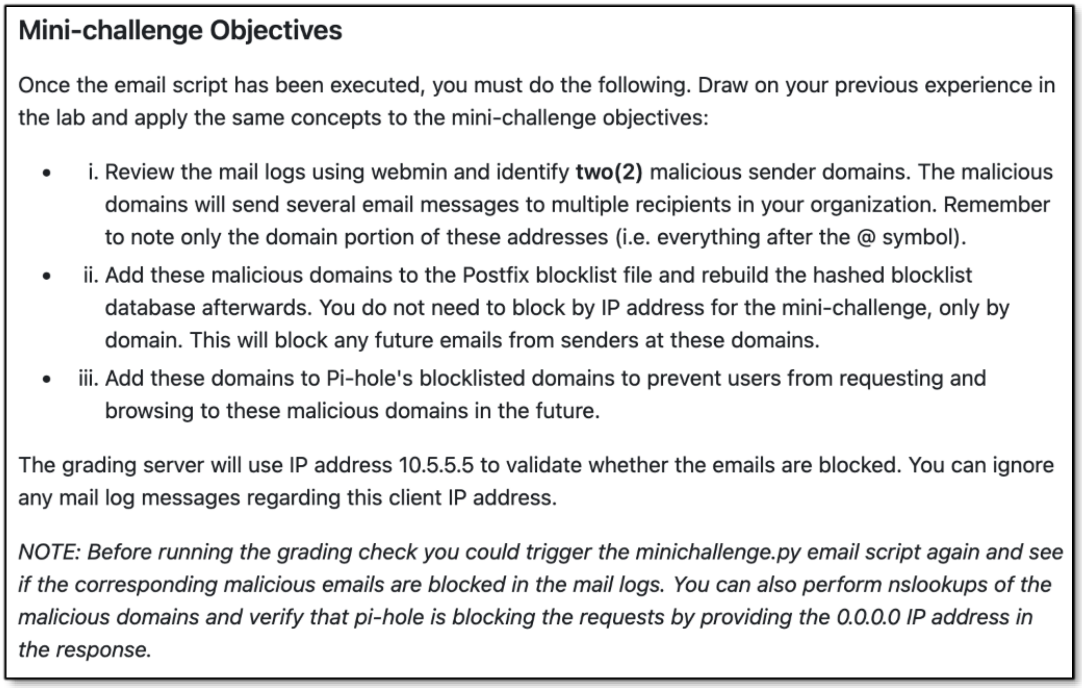
▾ Mini-Challenge Scenario

For the mini-challenge, your objective is to review mail logs and then add the malicious domains present to both the Postfix and Pi-hole blocklists.

To do this, a mailing script has been created for you.

1. Return to the **Red-Kali** console and open a Terminal.
2. (**Red-Kali, Terminal**) Execute the mini-challenge mail script with the following command:

Figure 19: Lab Guide Mini-Challenge



Mini-challenge Objectives

Once the email script has been executed, you must do the following. Draw on your previous experience in the lab and apply the same concepts to the mini-challenge objectives:

- i. Review the mail logs using webmin and identify **two(2)** malicious sender domains. The malicious domains will send several email messages to multiple recipients in your organization. Remember to note only the domain portion of these addresses (i.e. everything after the @ symbol).
- ii. Add these malicious domains to the Postfix blocklist file and rebuild the hashed blocklist database afterwards. You do not need to block by IP address for the mini-challenge, only by domain. This will block any future emails from senders at these domains.
- iii. Add these domains to Pi-hole's blocklisted domains to prevent users from requesting and browsing to these malicious domains in the future.

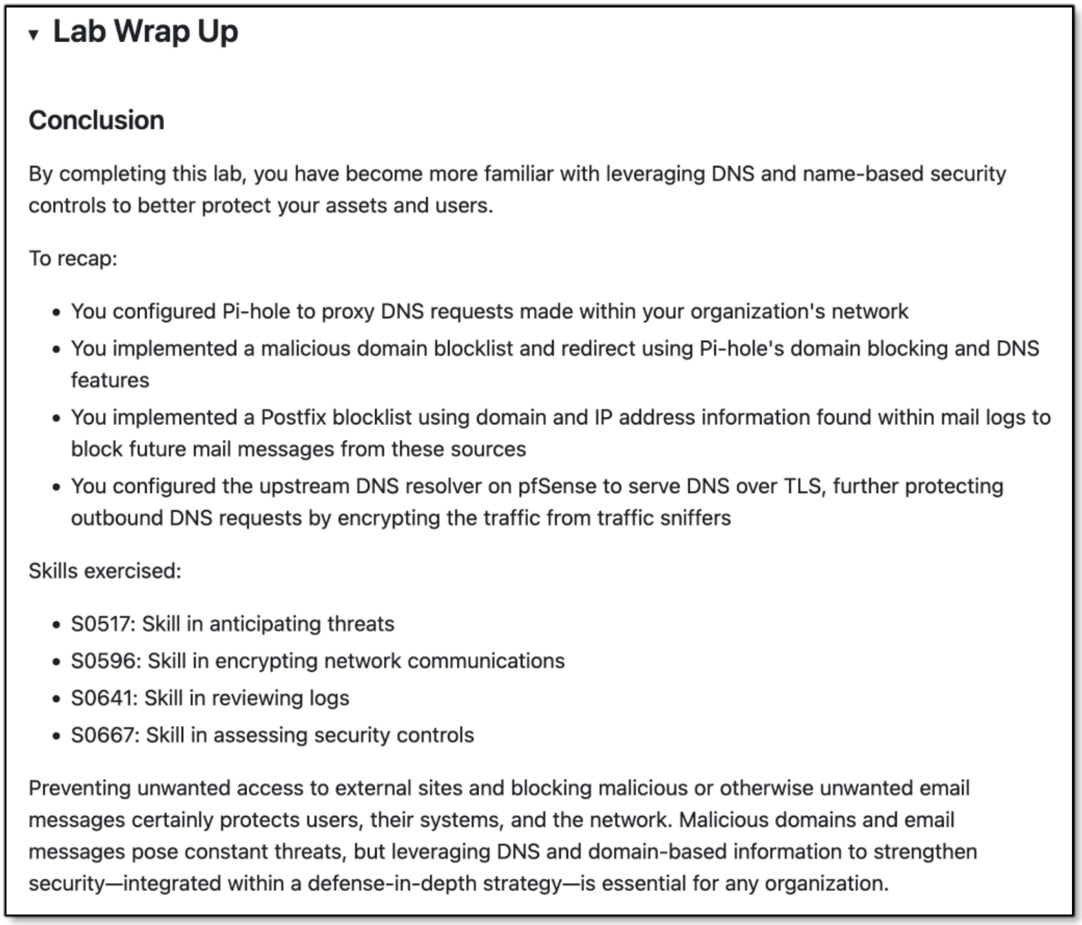
The grading server will use IP address 10.5.5.5 to validate whether the emails are blocked. You can ignore any mail log messages regarding this client IP address.

NOTE: Before running the grading check you could trigger the minichallenge.py email script again and see if the corresponding malicious emails are blocked in the mail logs. You can also perform nslookups of the malicious domains and verify that pi-hole is blocking the requests by providing the 0.0.0.0 IP address in the response.

Figure 20: Lab Guide Mini-Challenge Objectives

Lab Wrap Up

This component includes the lab conclusion (as shown in Figure 21), the list of skills exercised (from the NICE Framework), a list of references, and an answer key (where applicable). The lab conclusion reinforces the learning objectives by recapping the activities completed within the lab and restating why these skills are important.



▾ Lab Wrap Up

Conclusion

By completing this lab, you have become more familiar with leveraging DNS and name-based security controls to better protect your assets and users.

To recap:

- You configured Pi-hole to proxy DNS requests made within your organization's network
- You implemented a malicious domain blocklist and redirect using Pi-hole's domain blocking and DNS features
- You implemented a Postfix blocklist using domain and IP address information found within mail logs to block future mail messages from these sources
- You configured the upstream DNS resolver on pfSense to serve DNS over TLS, further protecting outbound DNS requests by encrypting the traffic from traffic sniffers

Skills exercised:

- S0517: Skill in anticipating threats
- S0596: Skill in encrypting network communications
- S0641: Skill in reviewing logs
- S0667: Skill in assessing security controls

Preventing unwanted access to external sites and blocking malicious or otherwise unwanted email messages certainly protects users, their systems, and the network. Malicious domains and email messages pose constant threats, but leveraging DNS and domain-based information to strengthen security—integrated within a defense-in-depth strategy—is essential for any organization.

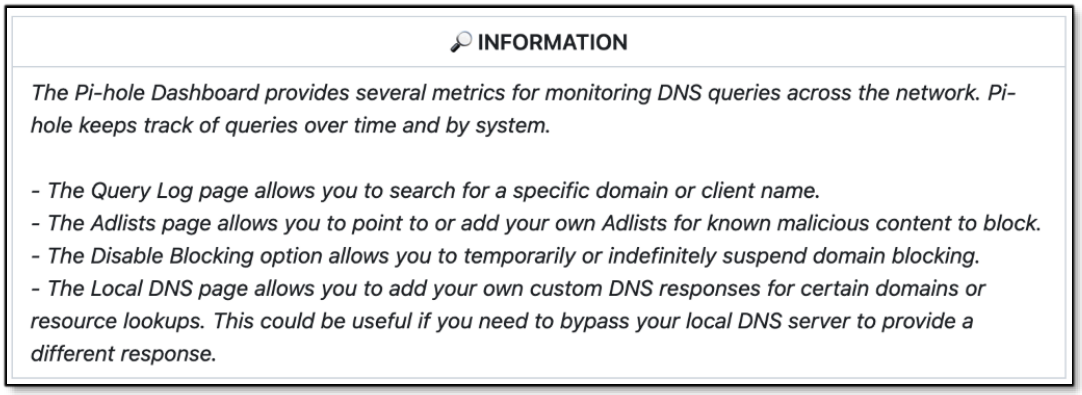
Figure 21: Lab Guide Wrap Up

Key Lab Guide Enhancements

The following enhancements were made to the SCL lab guides.

Callouts

Callouts (as shown in Figure 22) serve to emphasize key pieces of information. Examples include providing more detail on a specific tool or technology, reminding the learner of a core concept, or special notes that the learner should pay attention to. These callouts are placed in uniquely formatted blocks that help visually distinguish important information from the general step-by-step lab instructions.



🔎 **INFORMATION**

The Pi-hole Dashboard provides several metrics for monitoring DNS queries across the network. Pi-hole keeps track of queries over time and by system.

- The Query Log page allows you to search for a specific domain or client name.
- The Adlists page allows you to point to or add your own Adlists for known malicious content to block.
- The Disable Blocking option allows you to temporarily or indefinitely suspend domain blocking.
- The Local DNS page allows you to add your own custom DNS responses for certain domains or resource lookups. This could be useful if you need to bypass your local DNS server to provide a different response.

Figure 22: Lab Guide Callouts

Unicode

Unicode characters and other symbols (as shown in Table 1) can be used to highlight icons or specific actions within the step-by-step instructions. These are handy when the learner is looking for a button with the same icon within the application on the screen.

Table 1: Unicode Examples

Symbol	HTML Entity	Name
☰	&#9776;	"Hamburger Menu"
⋮	&#8942;	Vertical Ellipsis
✅	&#9989;	Green Checkmark
❌	&#10060;	Red X
🚫	&#128683;	Not Permitted

Formatting

Visual cues (as shown in Table 2) help the learner distinguish key text from the surrounding text. Examples include credentials, system or application names, filenames, file paths, labels, and important terms. Code and code blocks are also helpful for presenting commands and other computer text, such as scripts and configuration file contents. A style guide should be used to maintain consistent formatting and to prevent the overuse of these cues.

Table 2: Formatting Examples

Style	When to Use	Example
bold	Useful for directing the user's attention; perfect for emphasizing system names, application credentials, and buttons where the user should click	"Click the **File** menu and then **Open....**"

Style	When to Use	Example
italics	Can be used for more subtle callouts, like tabs in an interface, terms that are being defined for the first time, and other emphasis that doesn't need bold	"A *pod* is the smallest schedulable unit in Kubernetes."
`code`	Used for terminal commands, filenames, and paths, and source code using three backticks (```)	"Run `ifconfig` to display the current network information."

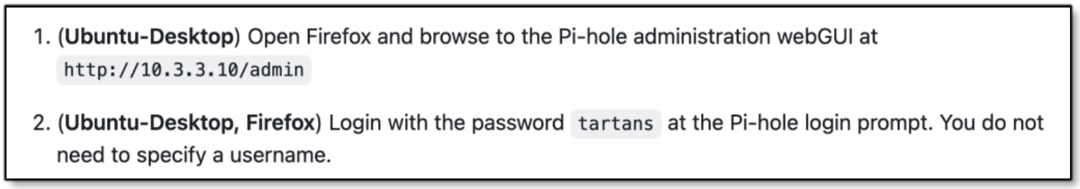
Steps vs. Statements

All steps are numbered items in the lab guide. Numbered steps allow the learner to easily identify what they must do and serve as a reference marker for their process (e.g., "Repeat Steps 2-5 for each user in the list").

Additional text that provides preparatory information or describes the outcome of a step is not numbered, as these are statements and not actions the learner must perform. These statements reinforce and help explain the previous step or introduce the upcoming step.

Context Clues

Each numbered step in an SCL includes a preface (as shown in bold in Figure 23) that lists the system and application being used. This information allows the learner to affirm they have the appropriate context for the step (e.g., using the Terminal on Kali or using Firefox on Ubuntu). These context clues are vital in labs where learners navigate among multiple systems and/or applications.



1. **(Ubuntu-Desktop)** Open Firefox and browse to the Pi-hole administration webGUI at `http://10.3.3.10/admin`
2. **(Ubuntu-Desktop, Firefox)** Login with the password `tartans` at the Pi-hole login prompt. You do not need to specify a username.

Figure 23: Lab Guide Context Clues

References

URLs are valid as of the publication date of this report.

[Beason 2021]
Beason, R. et al. *Evaluation of Hands-On Cybersecurity Skills Development*. INL/EXT-21-64359. Idaho National Laboratory. 2021. https://www.osti.gov/biblio/1825671

[CISA 2024]
Cybersecurity and Infrastructure Security Agency (CISA). Federal Civilian Executive Branch (FCEB) Operational Cybersecurity Alignment (FOCAL) Plan. *CISA Website*. September 16, 2024. https://www.cisa.gov/resources-tools/resources/federal-civilian-executive-branch-fceb-operational-cybersecurity-alignment-focal-plan

[CISA 2025a]
Cybersecurity and Infrastructure Security Agency (CISA). CISA Cyber Training Bulletin: September – October 2025. 2025. *CISA Website*. September 8, 2025. https://content.govdelivery.com/accounts/USDHSCISA/bulletins/3efc59d

[CISA 2025b]
Cybersecurity and Infrastructure Security Agency (CISA). President’s Cup Challenges. *GitHub*. August 2025. https://github.com/cisagov/prescup-challenges

[CISA 2025c]
Cybersecurity and Infrastructure Security Agency (CISA). Skilling Continuation Labs. *GitHub*. August 2025. https://github.com/cisagov/prescup-challenges/tree/main/skilling-continuation-labs

[CISA 2025d]
Cybersecurity and Infrastructure Security Agency (CISA). President's Cup Cybersecurity Competition Practice Area. *President’s Cup Cybersecurity Competition Website*. November 12, 2025 [accessed]. https://pccc.cisa.gov/gb/practice

[CISA 2025e]
Cybersecurity and Infrastructure Security Agency (CISA). Network Segmentation with ICS or HMI Lab Python Grading Script. *GitHub*. August 2025. https://github.com/cisagov/prescup-challenges/blob/main/skilling-continuation-labs/network-segmentation-with-ics-or-hmi/lab/skillshub/grading.py

[Defelice 2021]
Defelice, R. A. How Long Does It Take to Develop Training? New Question, New Answers [blog post]. *ATD Blog*. January 13, 2021. https://www.td.org/content/atd-blog/how-long-does-it-take-to-develop-training-new-question-new-answers

[Morrison 1992]

Morrison, R. F. & Brantner, T. M. What Enhances or Inhibits Learning a New Job? A Basic Career Issue. *Journal of Applied Psychology*. Volume 77. Issue 6. December 1992. Pages 926–940. https://doi.org/10.1037/0021-9010.77.6.926

[NACE 2024]

National Association of Colleges and Employers (NACE). Job Outlook 2025. *NACE Website*. 2024. https://www.naceweb.org/docs/default-source/default-document-library/2024/publication/free-report/2025-nace-job-outlook.pdf?Status=Master&sfvrsn=4aa12a02_3

[NICCS 2025]

National Initiative for Cybersecurity Careers and Studies (NICCS). NICE Framework for Cybersecurity (NICE Framework). NICCS Website. August 28, 2025. https://niccs.cisa.gov/tools/nice-framework

[SEI 2025a]

Software Engineering Institute (SEI). Challenge Server. *GitHub*. May 2025. https://github.com/cmu-sei/Challenge-Server

[SEI 2025b]

Software Engineering Institute (SEI). TopoMojo. *GitHub*. Nov 2025. https://github.com/cmu-sei/TopoMojo/

[SEI 2025c]

Software Engineering Institute (SEI). Crucible. *GitHub*. Nov 2025. https://github.com/cmu-sei/crucible

[SEI 2025d]

Software Engineering Institute (SEI). Workspace and Gamespace Documentation. *Software Engineering Institute GitHub*. Nov 2025. https://cmu-sei.github.io/crucible/topomojo/#workspace-and-gamespace

[SEI 2025e]

Software Engineering Institute (SEI). Gameboard. *GitHub*. Oct 2025. https://github.com/cmu-sei/Gameboard

[SEI 2024]

Software Engineering Institute (SEI), Carnegie Mellon University (CMU). *Approach to Skilling the Cyber Workforce*. SEI. 2024. https://www.sei.cmu.edu/library/approach-to-skilling-the-cyber-workforce/

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. **AGENCY USE ONLY**
(Leave Blank)

2. **REPORT DATE**
November 2025

3. **REPORT TYPE AND DATES COVERED**
Final

4. **TITLE AND SUBTITLE**
A Practitioner's Guide to Designing and Developing Hands-On Cybersecurity Skilling Continuation Labs

5. **FUNDING NUMBERS**
FA870225DB003

6. **AUTHOR(S)**
Richard Weise, Christopher Herr, and Nicholas Giruzzi

7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. **PERFORMING ORGANIZATION REPORT NUMBER**
CMU/SEI-2025-TR-010

9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
SEI Administrative Agent
AFLCMC/AZS
5 Eglin Street
Hanscom AFB, MA 01731-2100

10. **SPONSORING/MONITORING AGENCY REPORT NUMBER**
n/a

11. **SUPPLEMENTARY NOTES**

12A **DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified/Unlimited, DTIC, NTIS

12B **DISTRIBUTION CODE**

13. **ABSTRACT (MAXIMUM 200 WORDS)**

Federal cybersecurity professionals face a unique set of threats, risks, regulations, and requirements. The Software Engineering Institute (SEI) leveraged its experience with cybersecurity best practices, federal government guidance and recommendations, and workforce development practices to deliver engaging, specialized training for federal cybersecurity professionals. In partnership with the Cybersecurity and Infrastructure Security Agency (CISA), the Cyber Mission Readiness (CMR) directorate of the CERT Division at the SEI developed a set of Skilling Continuation Labs (SCLs) to provide novel, relevant, and unique hands-on immersive training to upskill the federal cybersecurity workforce. To help support training lab developers, this report draws on CMR team members' expertise in developing high-fidelity cybersecurity training labs for the federal workforce and provides recommendations and guidelines for developing effective and immersive hands-on cybersecurity labs.

14. **SUBJECT TERMS**
capacity building, workforce development, cybersecurity, Skilling Continuation Labs, SCL

15. **NUMBER OF PAGES**
38

16. **PRICE CODE**

17. **SECURITY CLASSIFICATION OF REPORT**
Unclassified

18. **SECURITY CLASSIFICATION OF THIS PAGE**
Unclassified

19. **SECURITY CLASSIFICATION OF ABSTRACT**
Unclassified

20. **LIMITATION OF ABSTRACT**
UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102