ATTACKIQ

Compliance Optimization Blueprint

Implementation Guide

January 2021

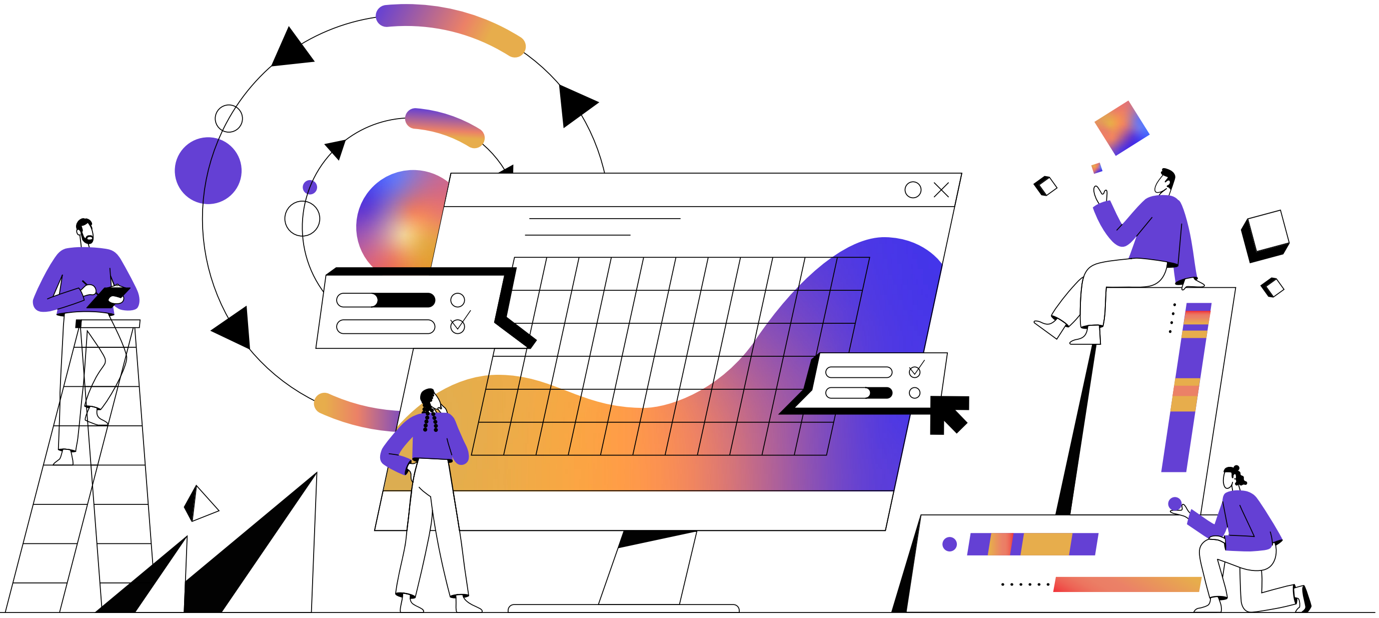


Table of Contents

1. Foreword: Validating Compliance Readiness Through Automation 3

2. Overview 4

2.1 Executive Summary 4

3. Methodology 5

3.1 Orientation 5

3.1.1 Framework Selection 5

3.1.2 Preparation & Staging 6

3.1.3 Data Collection 6

3.2 Planning 7

3.2.1 Set the Goal 7

3.2.2 Define the Process 9

3.2.2.1 Workflow 9

3.2.2.3 Reporting Chains and Procedures 10

3.2.2.4 Products 10

3.2.3 Delineate Roles and Responsibilities 11

3.2.4 Plan Asset Deployment 13

3.2.5 Analyze & Map Controls 13

3.2.5.1 Control Analysis 13

3.2.5.2 Test Case Mapping 15

3.2.6 Justify Mappings 15

3.2.7 Build and Configure Assessments 17

3.3 Execution 17

3.4 Analysis 18

3.5 Remediation 19

3.6 Validation 19

3.7 Reflection and Automation 19

4. Conclusion 20

1. Foreword: Validating Compliance Readiness Through Automation

The foundation of a successful security capability is, first and foremost, in the competence of the practitioners who create it. Our intent when creating these blueprints was to combine our industry-leading, automated security optimization platform with the essential elements of methodology and practice enablement to deliver our customers a complete and ready-to-operationalize set of solutions. Having implemented the Automated Testing Blueprint, your practice can build automation into any number of security testing and measurement use cases.

As a means of either meeting regulatory/contractual obligations, making measurable security improvements, improving the maturity of security operations, demonstrating security readiness to business leadership, or some combination thereof, adopting a cybersecurity framework is understood to be the best place to baseline and mature your capability. These frameworks offer a comprehensive set of guidelines for planning, building, operating, and maturing a security program and are, as a matter of course, kept current with the pace of increasing technological complexity.

Security frameworks are, however, complicated in and of themselves. A Center for Internet Security survey of 300 security decision makers across 17 industry verticals indicated that five key issues stood in the way of adopting a security framework across their enterprises:

1. Lack of trained staff
2. Lack of necessary tools to automate controls
3. Lack of budget
4. Lack of appropriate tools to audit continuous effectiveness of controls
5. Lack of integration among tools

Why these issues? Because security frameworks prescribe controls that are themselves non-trivial systems formed of people, processes, and technology. Be they procedural or technical, the only way to know if you've adequately implemented the controls is to actively test them. Manual testing is a manpower investment equivalent to simply implementing the controls, itself a significant effort. Deriving immediately actionable and communicable information from the results is a similarly difficult exercise without the context provided by sound analysis of how the controls interact with organizational risk.

Testing alone does not prove compliance. It takes continuous effort and expert analysis on the part of experienced professionals to ensure that regulatory requirements are met in full. That's the reason we compiled this specific blueprint: to provide a flexible means for customers to assess the efficacy and prove the implementation of controls and address some of the issues mentioned above.

AttackIQ works directly with customers to make threat-informed defense a reality with AttackIQ blueprints — like the one that you're about to read. These AttackIQ blueprints are step-by-step guides to align people, process, and technology to deliver optimization in 26 distinct solutions across the security organization. These solutions anchor to existing security pillar functions, but bring threat-informed testing automation to improve them. The risk and compliance teams can begin the journey of automating their risk models with continuous control efficacy data, as well as their regulatory remit with automation derived dashboards.

- **Ben Opel**, *Senior Director, Customer Success*
- **Brandt Mackey**, *VP, Customer Success*
- **Stephan Chenette**, *Founder and CTO*

2. Overview

2.1 Executive Summary

AttackIQ blueprints allow organizations to mature their overall security posture and maximize the value of the AttackIQ security optimization platform. The objective of this blueprint is to operationalize the pervasive and persistent use case of compliance validation.

By implementing the instructions in this blueprint, users will be able to:

- Analyze the validation requirements of a compliance framework
- Generate operational plans for continuous compliance validation
- Provide decision makers with actionable compliance metrics
- Implement a process for baselining an environment's compliance status
- Understand compliance controls as the defensive best practices that they are

This document covers a specific implementation of the security optimization lifecycle, which enables continuous validation of compliance framework controls. It defines a methodology whereby security, governance/risk/compliance (GRC), and infrastructure organizations can decompose framework control statements and guidance into actionable, operational implementations of the AttackIQ Security Optimization Platform.

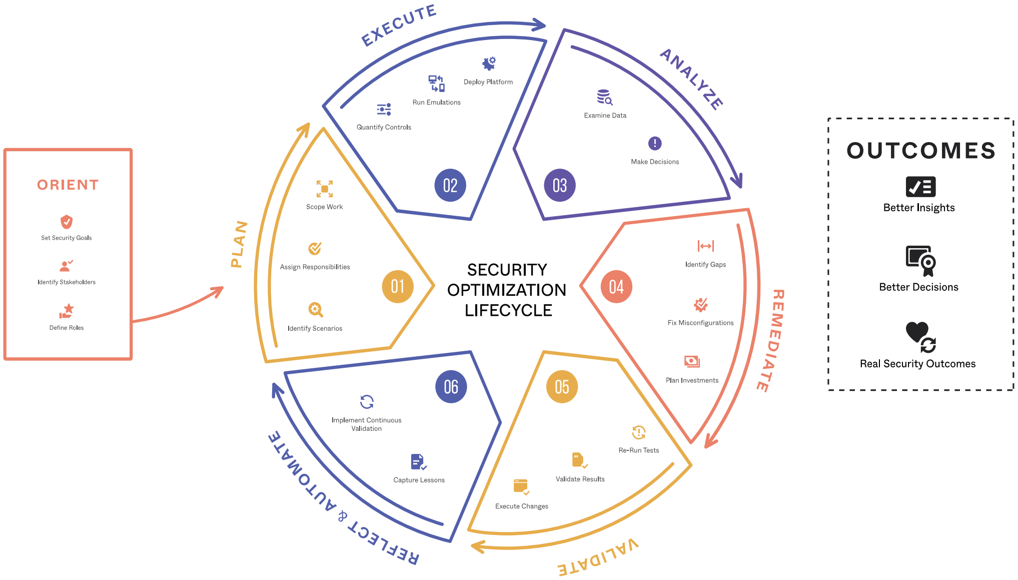


Figure 1: The Security Optimization Lifecycle

This document is written and intended for all stakeholders involved in the compliance process: from GRC leaders to EDR technicians and other personnel who operate or interact with the AttackIQ platform and the insights it produces. It is strongly recommended that all stakeholders read this document to understand the steps required to fulfill the ultimate goal of demonstrating effective validation of the technical aspects in adopted compliance frameworks.

As a resulting business outcome of implementing this blueprint, your organization will be able to demonstrate measurable proof of its compliance program's efficacy in the face of both audit and real-world attack.

3. Methodology

3.1 Orientation

3.1.1 Framework Selection

Compliance framework selection is generally done for you — if your vertical or mission requirements are within a set of regulated, subject to official oversight, or otherwise governed categories, one more framework is is prescribed as a matter of law.
If not, adopting a compliance or policy framework is still a best practice for establishing and understanding your baseline security posture and is a handy set of guidelines for any security program.

Your specific requirements for framework adoption will vary depending on a combination of factors including but not limited to: countries/states/provinces in which you operate, transmit, or store data; specific industry vertical(s) you occupy; and the size and scope of your organization. A general alignment and description of common compliance frameworks is seen in the below figure:

	Applicability	Adoption Difficulty
NIST CSF	General	Moderate
NIST SP 800-53* **	U.S. Federal	Higher
ISO/IEC 27001	General	Higher
CIS Guidelines* **	General	Lower
GDPR	European Union	Higher
PCI DSS	Credit Card Processors*	Varies as scale changes
SOX	U.S. Public Companies**	Higher
FFIEC CAT	U.S. Financial	Moderate

*Data retention, security, and privacy; **Publicly Traded; ***Controls Mapped to MITRE ATT&CK Techniques

If your organization has no regulatory or legal requirement to adopt a framework, consider first implementing the CIS guidelines. The combination of relatively low adoption difficulty and general applicability to any modern IT architecture/program make it an ideal place to start. This isn't to say that the CIS guidelines are purely entry-level, as they prescribe three maturity levels for every control that, if implemented thoughtfully and thoroughly, can underpin a highly successful and resilient enterprise security capability.

3.1.2 Preparation & Staging

To prepare for your team for compliance optimization, ensure that the following statements are true:

- Management in the IT, security, and governance/risk/compliance organizations are onboard respective to the potential benefits and risks

- Supporting budget and resource allocations are available or positioned

- Project plan skeletons and meetings times are staged to support an inter-department planning effort

- Stakeholders are identified and notified — see appendix A for an example RACI chart supporting a compliance dashboarding project

NOTE: When identifying stakeholders to engage with this process, ensure members of the GRC (governance, risk, and compliance) organization are accounted for as critical SME support in addition to business unit representation as appropriate/available.

> Concurrence from your GRC organization is a critical step; these stakeholders are the most important to engage with early and often because any rationalization of how a control is or is not validated is dependent on the auditor's opinion. That opinion is the stock-and-trade of your GRC organization.

3.1.3 Data Collection

Planning delays due to missing information are among the most common reasons for otherwise promising projects to be abandoned. To avoid this, ensure you have the following on hand before you bring stakeholders to the table:

- Any compliance framework tracking tools in use
- Documentation relevant to your selected frameworks
- All extant documentation on known weaknesses in security or "accepted risks"
- Documentation on compensating controls relating to the above
- Detailed results of the latest audit (as applicable)
- An inventory of security/risk capabilities, processes, policies, and documentation
- Asset inventories and network diagrams to define the technical landscape
- Assumed control implementations and asset mappings

3.2 Planning

"No plan survives first contact with the ~~enemy~~ auditor." – Anonymous

Planning is the most important and complex part of this blueprint. Planning consists of:

1. **Setting the Goal:** Determining the goal of the compliance dashboarding project, e.g., "Provide a persistent, objective, and easily communicable means of [control framework] control validation and compliance readiness measurement."

2. **Delineating Roles and Responsibilities:** Specifying which parts of the process are under the purview of which stakeholder(s).

3. **Defining the Process:** Defining the execution workflow, schedules, reporting chains, and products.

4. **Analyzing and Mapping Controls:** Analyzing your selected framework, parsing it into those controls that are subject to technical/procedural/policy validation, and constructing a mapping of control-to-scenario by which you can provide a justifiable validation of each control in your selected framework.

5. **Planning Asset Deployment:** Deciding, based on framework guidelines and architectural factors, precisely where and in what quantity to deploy agents.

6. **Justifying Mappings:** Describing to auditors and outside parties exactly how your test plan validates the specified controls.

7. **Configuring:** Preparing mapped scenarios for execution in your environment and in support of your control validation goal.

3.2.1 Set the Goal

This is your opportunity to define exactly what it is you mean to accomplish. In this stage, you'll determine the requirements for your compliance validation and dashboarding capability as products of an overarching mission statement. The mission of any project or capability has two parts:

Task	Purpose
What it's supposed to do	**Why it's supposed to do it**
"Provide a persistent, objective, and easily communicable means of [control framework] control validation and compliance readiness measurement."	"In order to automate the control audit process to improve periodicity, technical quality, and cost of demonstrating compliance."

3.2.1 Set the Goal (cont.)

This is an example of a mission, but it's been written to elucidate the three key elements of compliance optimization:

- **The solution must be persistent.** It should provide you with an updated status on-demand and enable differential comparisons of before-and-after updates or changes.

- **The solution must be objective.** Maturity, and in some cases basic competence, requires a third party's input. You need to demonstrate control validation beyond a statement or demonstration of the presence of a tool/ capability.

- **The solution must communicate easily.** None of this work does you any good beyond your own peace of mind as a defender or compliance professional if it can't present digestible results to every level of management and audit.

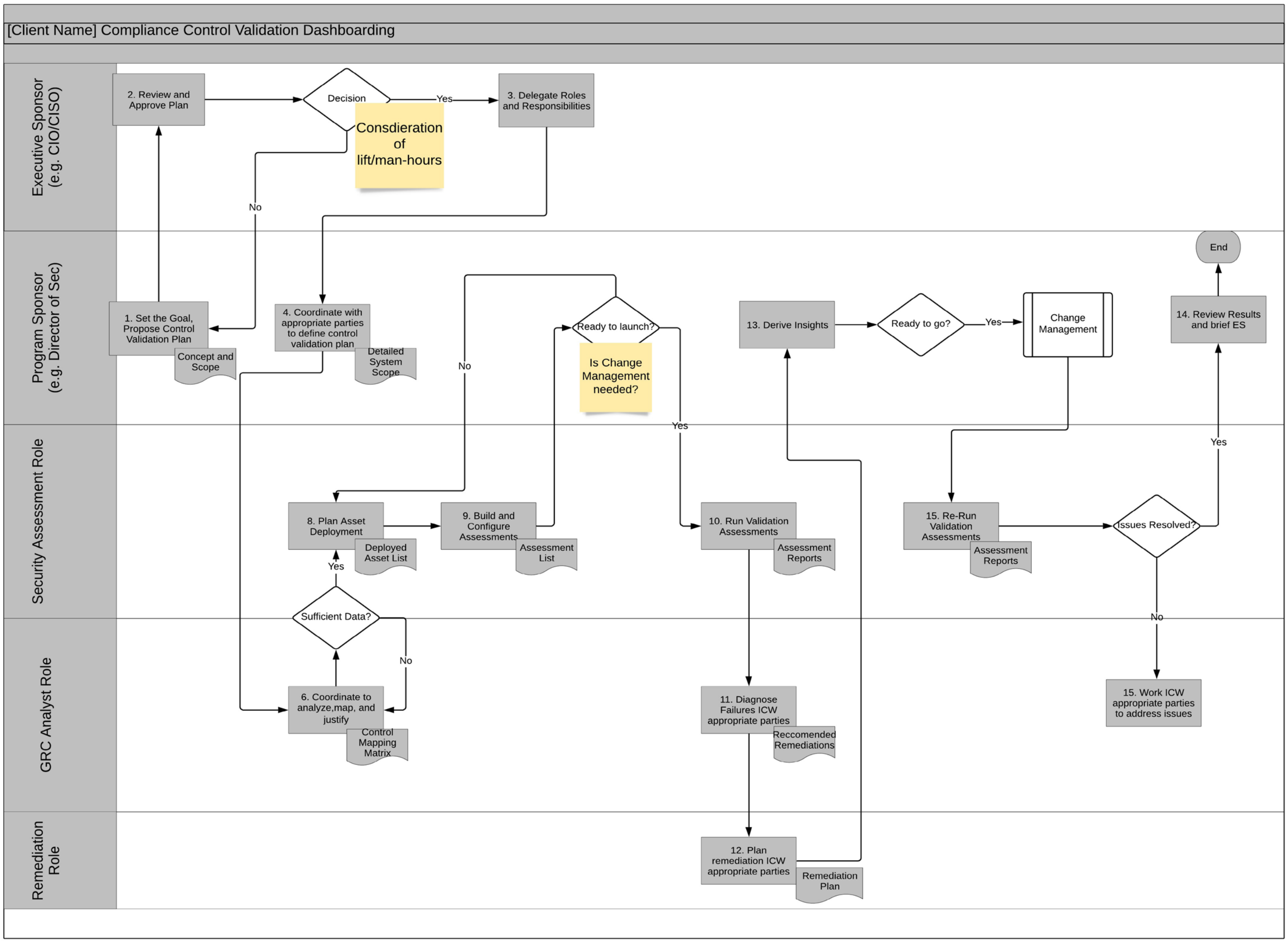
These are the **design principles** on which your specific solution should be based. Each dashboard, report, and workflow should implement these principles in some way as to ensure what you produce and implement is immediately effective as a means of both bringing early- and late-stage visibility to your compliance status. Once you have a mission statement that encompasses your intent for the project, you can begin describing a process that will achieve it.

3.2.2 Define the Process

This is when you'll define how you'll execute and utilize the solution. Key elements are:

3.2.2.1 Workflow

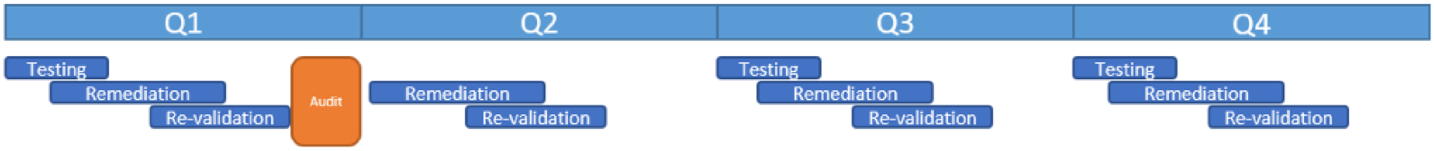
Define the process for the entire exercise; an example is provided below. Note that notifications to appropriate parties in Security, GRC, Infrastructure, Change Management, and Management are included in order to prevent confusion or undue alarm — this is a best practice for all security testing:


[Client Name] Compliance Control Validation Dashboarding
Executive Sponsor (e.g. CIO/CISO)
Program Sponsor (e.g. Director of Sec)
Security Assessment Role
GRC Analyst Role
Remediation Role
2. Review and Approve Plan
Decision
Yes
Consdieration of lift/man-hours
3. Delegate Roles and Responsibilities
No
End
1. Set the Goal, Propose Control Validation Plan
Concept and Scope
4. Coordinate with appropriate parties to define control validation plan
Detailed System Scope
No
Ready to launch?
Is Change Management needed?
13. Derive Insights
Ready to go?
Yes
Change Management
14. Review Results and brief ES
Yes
Yes
8. Plan Asset Deployment
Deployed Asset List
9. Build and Configure Assessments
Assessment List
10. Run Validation Assessments
Assessment Reports
15. Re-Run Validation Assessments
Assessment Reports
Issues Resolved?
Yes
Sufficient Data?
No
No
6. Coordinate to analyze,map, and justify
Control Mapping Matrix
11. Diagnose Failures ICW appropriate parties
Reccomended Remediations
15. Work ICW appropriate parties to address issues
12. Plan remediation ICW appropriate parties
Remediation Plan

AttackIQ Confidential and Proprietary

3.2.2.2 Schedule

Your validation and reporting schedule should pivot around audit timelines (as applicable), but should also frame out a series of tests and evaluations of your compliance status in advance of and following those events. It's important to test early and often, beginning with the results of your latest audit as a baseline immediately after the resulting remediation efforts and using the next audit as your "no later than" time to complete all tests.



3.2.2.3 Reporting Chains and Procedures

Information flow is paramount, and ensuring the correct results and requirements reach the right hands will be the key to gaining any benefit from the project.

3.2.2.4 Products

The reports and products of compliance optimization should be tuned to reflect the varying levels of technical expertise, scope of action, and decision authority resident in your stakeholders. This generally breaks down into five columns:

Regulator	GRC	Management	Security	Infrastructure
CARES	**CARES**	**CARES**	**CARES**	**CARES**
Framework Adoption	Regulatory Risk	Business Risk	Control Performance	Availability & QOS
NEEDS	**NEEDS**	**NEEDS**	**NEEDS**	**NEEDS**
Proof Of Control Implementation	Regulatory Control Validation	Decision Points & Metrics	Detection & Prevention Statistics	Resilience & Change Forecasts

3.2.3 Delineate Roles and Responsibilities

The roles and responsibilities involved in this project largely mirror those described in the foundational automated testing blueprint but with the following additions:

Role / Function	Responsibility
GRC Sponsor	The GRC sponsor sits in a co-equal role to the executive sponsor (and may be the same person). This will often be a delegate of the CISO or other senior leader whose oversight ensures the project contributes meaningfully to the organization's compliance goals.
GRC Analyst	A GRC analyst serves as the functional contributor to the project's actual implementation of compliance control validation and dashboarding. This role owns the justification of scenario-based test cases in alignment to compliance control requirements.
Business Unit Representatives	Business unit representatives participate in planning and execution to ensure the unique compliance requirements and implementation considerations of their respective units are represented and considered.
Security Assessment Role	The security assessment role owns the AttackIQ assessments and the associated scenarios. They own the execution of scheduling or running assessments on-demand or via the API and are typically responsible for accessing and sharing results from the assessments. In the context of compliance validation, they are responsible for selecting scenarios and building tests that prove technical validation of compliance controls.

3.2.3 Delineate Roles and Responsibilities (cont.)

A sample RACI chart including these roles is provided below:

- **Accountable** (A),
- **Responsible** (R),
- **Consulted** (C), and
- **Informed** (I).

Key:

- **ES** = Executive Sponsor/GRC Sponsor
- **PS** = Program Sponsor
- **SME** = Subject Matter Expert
- **SAR** = Security Assessment Role
- **GRC** = GRC Analyst Role
- **RR** = Remediation Role
- **IR** = Implementation Role
- **OR** = Other Stakeholders

Step	Task	ES	PS	SME	SAR	GRC	RR	OS
Planning								
1	Set The Goal	A/R	C	C	C	C	C	C
2	Assign Roles and Responsibilities	C	A/R	C/I	C/I	C/I	C/I	I
3	Define the Process	C/I	A/R	C	C	C	C	I
4	Analyze and Map Controls	I	C	C	C	A/R	C	I
5	Plan Asset Deployment	C/I	C	C	A/R	C	C	I
6	Justify Mappings	I	A	C	C	R	C	I
7	Build and Configure Assessments		I	C	A/R	C	C	I
Execution								
11	Run Validation Assessments			C	A/R	C	C	
Analysis								
12	Diagnose Failures		I	C	C	A/R	C	
Remediate								
13	Prioritize & Remediate Failures	I	I	I	C	I	A/R	
Validation								
14	Re-test		I	I	A/R	C	I	
Reflection and Automation								
15	Derive Insights and Begin CCV	C	A/R	C	C	C	I	I

3.2.4 Plan Asset Deployment

Exactly where, how, and in what quantities you should deploy AttackIQ assets is determined by your compliance framework guidelines, IT architecture, and the specific degree of confidence you're required to provide for each validation. Refer to the automated testing blueprint for more detail on general asset deployment strategy; this blueprint provides amplifying guidance specifically for compliance optimization.

- Consider **security zones and specifically identified technologies** and/or data as identified in your compliance framework when planning asset deployment. Unless prohibited by operational considerations, all systems hosting, processing, or transmitting such data or existing in such zones must have at least one test point present to provide a representative sample to the auditor.

- Consider variance in **configurations and technologies** between sites, enclaves, and business units in the same light as above.

- Ensure you can provide an **estimation of confidence in the applicability of your results** to the broader enterprise if you are unable to test a significant enough number of assets. This may be prescribed by your compliance framework, auditor, or management — refer to the agent deployment strategy guide for more information. (If you are a customer, you have access to that guide and all of our blueprints. If you would like to review that guide or our other blueprints, please contact our team at info@attackiq.com.)

3.2.5 Analyze & Map Controls

3.2.5.1 Control Analysis

Each of your compliance controls will have a specific means of validation:

- **Technical:** The control is validated by proof of specific tools or infrastructure functioning in a specific way.
- **Procedural:** The control is validated by demonstration of a staff process in response to a notional or actual event, alert, or other stimulus.
- **Policy:** The control is validated by proof of existing policy to support it.

A control validation may be a product of any one or combination of these means. Below is an example of a purely technical validation of NIST AC-6:

NIST SP 800-53/AC-6 Least Privilege		
Description	**Type**	**Test Case**
(1) Authorize Access to Security Functions	Technical	1: Attempt account creation as user
		2: Attempt log modification as user

Below is an example of a multipart validation of PCI DSS SAQ-C question 2.1(b). The technical validation is an attempted brute force of given systems using credential lists specific to those systems' known default credentials; the procedural validation is a demonstration of organizational change/configuration management practices relevant to the control specification; and the the policy validation is a review of relevant organizational policy concerning system installation, initial configuration, and testing. As highlighted below, it can be helpful to identify the ATT&CK technique associated with your proposed test case as a means of picking out procedures or scenarios for implementing the case.

3.2.5.1 Control Analysis (cont.)

PCI DSS/SAQ-C 2.1		
Description	**Validation Type**	**Test Case**
(b) Are unnecessary default accounts removed or disabled before installing a system on the network?	Technical	1: Brute-force selected hosts/apps with relevant default credential lists T1110.003 T1078.001
	Procedural	
	Policy	

Below is an example of a purely non-technical validation of PCI DSS SAQ-C question 2.2(b):

PCI DSS/SAQ-C 2.2		
Description	**Validation Type**	**Test Case**
(b) Are system configuration standards updated as new vulnerability issues are identified, as defined in Requirement 6.1?	Procedural	Review gold disk issue and implementation timestamps in relation to Vuln/Risk Management working group papers.
	Policy	Verify existing policy enforces No-Less-Than timelines between configuration decisions and whole-of-fleet implementation.

This alignment of control-validation-type test case is a simple and effective means of organizing your analysis and the analysis process serves the additional purpose of deepening your understanding of the control's purpose in relation to how you've implemented it.

For organizations tied to NIST SP 800-53, something of a shortcut exists. MITRE Engenuity's Center for Threat-Informed Defense has recently released a complete mapping of NIST controls to the ATT&CK framework techniques that they mitigate: https://github.com/center-for-threat-informed-defense/attack-control-framework-mappings.

Article:
CTID Project Review: Compliance Control-to-ATT&CK Mappings

Training:
ATT&CK-to-Compliance Framework Mappings: NIST SP 800-53

3.2.5.2 Test Case Mapping

Having determined the techniques/procedures best suited to test and validate your compliance, next is the process of mapping the technical control validations to the AttackIQ scenario library to provide an automated implementation of each such validation. As you consider each test case, ask the following questions:

1. Does this need to execute with specific credentials other than SYSTEM/root? Remember that the AttackIQ test point executes all scenarios with these credentials by default.
2. Based on the most probable adversary targets as well as specific compliance requirements as they apply to asset types, does this scenario need to run on and/or target a specific asset within your environment? How does the actual kill chain implemented reflect reality?
3. Bearing in mind that the AttackIQ platform gives results based on prevention/detection, how will you interpret the output as it relates to the validation of the control?

Group those test cases requiring the same credentials; each group will become a separate assessment in 3.2.6.

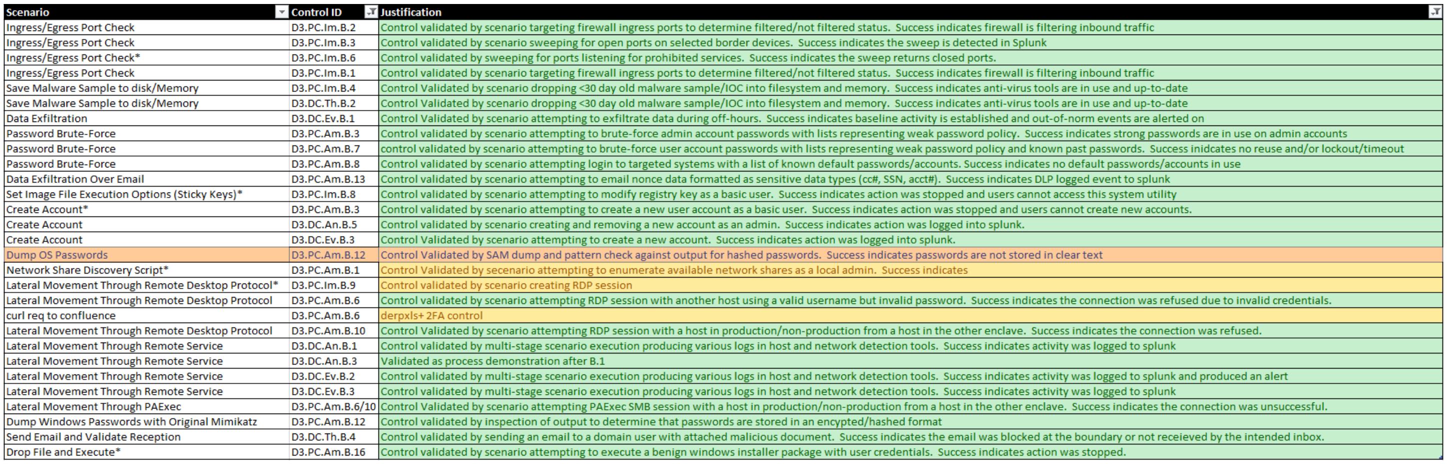
3.2.6 Justify Mappings

At this point, you know why and how the tests you've designed will validate compliance controls. Having documented your work, it is a matter of translating your internal knowledge of how it all comes together into something consumable by auditors and outside parties.

Doing so means referencing the specific language of the control and its applicability to your environment in conjunction with a mature understanding of adversary tradecraft (procedural and technical). Each justification should be a clear and concise description of how the scenario(s) selected specifically demonstrates your implementation of the control.

3.2.6 Justify Mappings (cont.)

The below figure is drawn from an analysis of the U.S. Federal Financial Institutions Examination Council (FFIEC) Cybersecurity Assessment Tool (CAT):



Scenario	Control ID	Justification
Ingress/Egress Port Check	D3.PC.Im.B.2	Control validated by scenario targeting firewall ingress ports to determine filtered/not filtered status. Success indicates firewall is filtering inbound traffic
Ingress/Egress Port Check	D3.PC.Im.B.3	Control validated by scenario sweeping for open ports on selected border devices. Success indicates the sweep is detected in Splunk
Ingress/Egress Port Check*	D3.PC.Im.B.6	Control validated by sweeping for ports listening for prohibited services. Success indicates the sweep returns closed ports.
Ingress/Egress Port Check	D3.PC.Im.B.1	Control validated by scenario targeting firewall ingress ports to determine filtered/not filtered status. Success indicates firewall is filtering inbound traffic
Save Malware Sample to disk/Memory	D3.PC.Im.B.4	Control Validated by scenario dropping <30 day old malware sample/IOC into filesystem and memory. Success indicates anti-virus tools are in use and up-to-date
Save Malware Sample to disk/Memory	D3.DC.Th.B.2	Control Validated by scenario dropping <30 day old malware sample/IOC into filesystem and memory. Success indicates anti-virus tools are in use and up-to-date
Data Exfiltration	D3.DC.Ev.B.1	Control Validated by scenario attempting to exfiltrate data during off-hours. Success indicates baseline activity is established and out-of-norm events are alerted on
Password Brute-Force	D3.PC.Am.B.3	Control validated by scenario attempting to brute-force admin account passwords with lists representing weak password policy. Success indicates strong passwords are in use on admin accounts
Password Brute-Force	D3.PC.Am.B.7	control validated by scenario attempting to brute-force user account passwords with lists representing weak password policy and known past passwords. Success inidcates no reuse and/or lockout/timeout
Password Brute-Force	D3.PC.Am.B.8	Control validated by scenario attempting login to targeted systems with a list of known default passwords/accounts. Success indicates no default passwords/accounts in use
Data Exfiltration Over Email	D3.PC.Am.B.13	Control validated by scenario attempting to email nonce data formatted as sensitive data types (cc#, SSN, acct#). Success indicates DLP logged event to splunk
Set Image File Execution Options (Sticky Keys)*	D3.PC.Im.B.8	Control validated by scenario attempting to modify registry key as a basic user. Success indicates action was stopped and users cannot access this system utility
Create Account*	D3.PC.Am.B.3	Control validated by scenario attempting to create a new user account as a basic user. Success indicates action was stopped and users cannot create new accounts.
Create Account	D3.DC.An.B.5	Control validated by scenario creating and removing a new account as an admin. Success indicates action was logged into splunk.
Create Account	D3.DC.Ev.B.3	Control Validated by scenario attempting to create a new account. Success indicates action was logged into splunk.
Dump OS Passwords	D3.PC.Am.B.12	Control Validated by SAM dump and pattern check against output for hashed passwords. Success indicates passwords are not stored in clear text
Network Share Discovery Script*	D3.PC.Am.B.1	Control Validated by secenario attempting to enumerate available network shares as a local admin. Success indicates
Lateral Movement Through Remote Desktop Protocol*	D3.PC.Im.B.9	Control validated by scenario creating RDP session
Lateral Movement Through Remote Desktop Protocol	D3.PC.Am.B.6	Control validated by scenario attempting RDP session with another host using a valid username but invalid password. Success indicates the connection was refused due to invalid credentials.
curl req to confluence	D3.PC.Am.B.6	derpxls+ 2FA control
Lateral Movement Through Remote Desktop Protocol	D3.PC.Am.B.10	Control Validated by scenario attempting RDP session with a host in production/non-production from a host in the other enclave. Success indicates the connection was refused.
Lateral Movement Through Remote Service	D3.DC.An.B.1	Control validated by multi-stage scenario execution producing various logs in host and network detection tools. Success indicates activity was logged to splunk
Lateral Movement Through Remote Service	D3.DC.An.B.3	Validated as process demonstration after B.1
Lateral Movement Through Remote Service	D3.DC.Ev.B.2	Control validated by multi-stage scenario execution producing various logs in host and network detection tools. Success indicates activity was logged to splunk and produced an alert
Lateral Movement Through Remote Service	D3.DC.Ev.B.3	Control validated by multi-stage scenario execution producing various logs in host and network detection tools. Success indicates activity was logged to splunk
Lateral Movement Through PAExec	D3.PC.Am.B.6/10	Control Validated by scenario attempting PAExec SMB session with a host in production/non-production from a host in the other enclave. Success indicates the connection was unsuccessful.
Dump Windows Passwords with Original Mimikatz	D3.PC.Am.B.12	Control validated by inspection of output to determine that passwords are stored in an encypted/hashed format
Send Email and Validate Reception	D3.DC.Th.B.4	Control validated by sending an email to a domain user with attached malicious document. Success indicates the email was blocked at the boundary or not receieved by the intended inbox.
Drop File and Execute*	D3.PC.Am.B.16	Control validated by scenario attempting to execute a benign windows installer package with user credentials. Success indicates action was stopped.

Note that the justification details the execution of two scenarios and their configurations and utilizes the language of "validated/not validated" to express the information you need for this use case rather than the AttackIQ platform's usual language regarding prevention and detection.

A common pitfall when mapping technical validation techniques to compliance controls is the failure to interpret the intent of the control. Control statements are generally explicit in nature, and when they aren't, there's generally supporting documentation to support analysis somewhere. Remember to brainstorm validation justification with each phrase of the control language explicitly repeated somewhere.

3.2.7 Build and Configure Assessments

In the last step before implementation, you'll combine your scenario mapping with the answers to the questions in 3.2.5.2.

For those test cases that can execute as SYSTEM/root, start by creating a custom assessment and create a test for each control subject to validation, populating the tests with the scenarios you mapped earlier.

For the test case groups requiring execution with other credentials, create assessments from the Managed Privileges Assessment Template, set the execution credentials as appropriate in Assessment Setup, and continue on as above.

As always, some scenarios will execute as needed "out-of-the-box," and the AttackIQ platform offers you the ability to modify and configure others to suit the specific needs of your use case.

Ensure that you follow organizational change management processes and check your implementation plan for any possible need for Change Management review before executing.

3.3 Execution

With all actions coordinated and socialized per both this document and the automated testing blueprint, this phase boils down to the coordinated and deliberate execution of your validation assessments against your selected assets.

As with all security tests, ensure full transparency with and appropriate notification of all stakeholders before, during, and immediately after the assessments complete.

Refer to the automated testing blueprint for more information on general execution considerations.

3.4 Analysis

When analyzing the results of your validation assessments, consider:

- If a compliance control failed validation, do you have a compensating control in place elsewhere in your architecture, security stack, or policy base to cover it? Can you test and prove that control?
- Which mapped scenarios went undetected/unprevented to cause the failure and how?
- Did any scenarios error out and result in an indeterminate outcome for a control? Why?
- Of those test cases you defined as requiring both technical and policy/procedure validation, have you validated the latter?
- Are there any systemic or architectural issues identified by any trend of validation failures, e.g., missed, detection-based validations in privileged access management and anomaly detection due to inadequate or ineffective system/security log auditing?

Below is an example of reporting output that would enable analysis, reporting, and remediation planning by GRC and security staff:

Control ID#	Outcome	Scenario Results	Justification	Compensating Control
ID# 654321 (test 1)	Validated	[Scenario Name] Passed	Control validated by scenario targeting firewall ingress/egress ports to determine filtered/not filtered status, Success indicates firewall is filtering inbound traffic	Conpenseting Control: Access Segmenetion in place
		[Scenario Name] Passed		
ID# 123456 (test 2)	Not Validated	[Scenario Name] Failed	Not Validated; control failed to detect >30 day old matwate sample 10Bd to memory	
		[Scenario Name] Passed	Validated by scenario failing on-disk load of malware sample	None
ID# 741369 (test 3)	Not Validated	[Scenario Name] Failed	Not Validated; control failed to detect or prevent Lateral movement attemps	None
ID# 963147 (test 4)	Validated [Compenseted]	[Scenario Name] Failed	Control Failed validation; Compensated	Compenating Control: Physical Access Policy and BIOS enforcement in Plece
ID# 753159 (test 5)	Validated [Compenseted]	[Scenario Name] Failed	Control Failed validation; Compensated	Compensating Controll Admin account auditing in place
		[Scenario Name] Passed	Control validated by scenario prevented from creating new account as Admin	None

3.5 Remediation

Remediation of findings in a compliance validation project should be a joint endeavor of the security, infrastructure, and GRC teams. Prioritization falls to the GRC team, having the best practical understanding of the organization's appetite for compliance-based risk. That being said, compliance controls run the gamut from highly-involved implementations requiring complex configuration management and policy orchestration to simple settings relating to password complexity; management should maintain oversight of remediation efforts to ensure resources are aligned in accordance with the organization's goals and risk management strategy.

3.6 Validation

Compliance validation as described in this blueprint can be considered a reframing of continuous security validation (CSV) with a focus on risk management. Just as in CSV, when a compliance control validation fails and its root cause is traced, the AttackIQ Security Optimization Platform provides a simple means of re-testing it on-demand as many times and as often as required. The difference, however, is clear: compliance adoption is measured not by operational success or risk reduction, but by an objective measurement and observation by professional auditors.

Also as with CSV, validation and revalidation of compliance readiness should be continuous, with failed validations re-tested until they pass and passed validations re-tested until they fail. In the context of enterprise governance, risk, and compliance, this process serves the dual purpose of enforcing timely, periodic reviews of all policies which are closely tied to technical implementation. This keeps the organization's documentation as current as its technical controls. **Such a process must be integrated with organizational change management policies, processes, and personnel.**

3.7 Reflection and Automation

Having completed an initial test of implemented controls, your consideration of how the testing program can evolve should incorporate answers to these questions:

- How to address apparent or suggested systemic issues in framework adoption and enforcement.
- Which tools/capabilities are functioning in accordance with requirements, and which are not? Where should investments be made, increased, or reduced?
- What is the best tempo/schedule for testing efficacy of compliance controls?
- What is the best means for further validation of control implementation, e.g., third parties?

Most of these will vary based on your organization and its specific requirements, but the testing schedule is best started with one per quarter at the very least. Based on your requirements and audit schedule, you should ensure that time is given to all involved parties to address discovered issues, plan fixes, and implement them before either the next test or the audit.

Consider how to use this data before budgeting for future enhancement; this blueprint process can inform how you invest against improving compliance scores to meet OKRs or other metrics. As an opportunity to model the problem and investment plan, this process enables you to define and plan for discrete goals.

4. Conclusion

This blueprint implementation guide aims to provide guidance for operationalizing the AttackIQ platform in support of your compliance adoption and continuous validation use case.

If you've completed this blueprint, we highly recommend bringing additional use cases, solutions, and blueprints up in your next discussion with the AttackIQ customer success team to gain further insights, value, and business outcomes.

For more information about additional use cases, solutions, and blueprints, please visit the Solutions page on the AttackIQ website.

ATTACKIQ

U.S. Headquarters
2901 Tasman Dr. Suite 112
Santa Clara, CA 95054
+1 (888) 588-9116
info@attackiq.com