CIS Control 18: Penetration Testing

Test the effectiveness and resiliency of enterprise assets through identifying and exploiting weaknesses in controls (people, processes, and technology), and simulating the objectives and actions of an attacker.

Why is this CIS Control Critical?

A successful defensive posture requires a comprehensive program of effective policies and governance, strong technical defenses combined with appropriate action from people. However, it is rarely perfect. In a complex environment where technology is constantly evolving and new attacker tradecraft appears regularly; enterprises should periodically test their controls to identify gaps and assess their resiliency. This test may be from an external network, internal network, application, system, or device perspective. It may include social engineering of users or physical access control bypasses.

Often, penetration tests are performed for specific purposes, like as a "dramatic" demonstration of an attack, usually to convince decision-makers of their enterprise's weaknesses and as a means to test the correct operation of enterprise defenses ("verification") • To test that the enterprise has built the right defenses in the first place ("validation")

Independent penetration testing can provide valuable and objective insights about the existence of vulnerabilities in enterprise assets and humans and the efficacy of defenses and mitigating controls to protect against adverse impacts to the enterprise. They are part of a comprehensive, ongoing program of security management and improvement. They can also reveal process weaknesses, such as incomplete or inconsistent configuration management or end-user training.

Penetration testing differs from vulnerability testing, described in CIS Control 7. Vulnerability testing just checks for the presence of known, insecure enterprise assets and stops there. Penetration testing goes further to exploit those weaknesses to see how far an attacker could get and what business process or data might be impacted through the exploitation of that vulnerability. This is an important detail, and often penetration testing and vulnerability testing are incorrectly used interchangeably. Vulnerability testing is exclusively automated scanning with sometimes manual validation of false positives, whereas penetration testing requires more human involvement and analysis, sometimes supported through the use of custom tools or scripts. However, vulnerability testing is often a starting point for a penetration test.

Another common term is "Red Team" exercises. These are similar to penetration tests in that vulnerabilities are exploited; however, the the difference is the focus. Red Teams simulate specific attacker TTPs to evaluate how an enterprise's environment would withstand an attack from a specific adversary or category of adversaries.

18.1: Establish and Maintain a Penetration Testing Program

Establish and maintain a penetration testing program appropriate to the size, complexity, industry, and maturity of the enterprise. Penetration testing program characteristics include scope, such as network, web application, Application Programming Interface (API), hosted services, and physical premise controls; frequency; limitations, such as acceptable hours, and excluded attack types; point of contact information; remediation, such as how findings will be routed internally; and retrospective requirements.

Asset Type	Security Function	Implementation Groups
Documentation	Govern	2, 3

Dependencies

- None

Inputs

1. `GV53` Penetration Testing Program Documentation
2. Date of last update to the penetration testing program documentation

Operations

1. Determine if Input 1 `GV53` exists within the enterprise

    1. If Input 1 exists, M1 = 1
    2. If Input 1 does not exist, M1 = 0

2. Check Input 1 for completeness. At a minimum, it should include the scope of the program, frequency, point of contact information, remediation, and retrospective requirements.

    1. For each component included in the documentation, assign a value of 1. Sum the values. (M2)

3. Compare Input 2 to the current date. Capture timeframe in months (M3)

Measures

- M1 = Output of Operation 1
- M2 = Sum of components included in the documentation
- M3 = Timeframe in months since the last update to the documentation

Metrics

- If M1 is 0, this Safeguard receives a failing score. The other metrics don't apply.
- If M3 is greater than twelve months, then this Safeguard is measured at a 0 and receives a failing score. The other metrics don't apply.

Completeness

Metric	The percentage of minimum components included in the program documentation
Calculation	`M2 / 5`

18.2: Perform Periodic External Penetration Tests

Perform periodic external penetration tests based on program requirements, no less than annually. External penetration testing must include enterprise and environmental reconnaissance to detect exploitable information. Penetration testing requires specialized skills and experience and must be conducted through a qualified party. The testing may be in a clear box or an opaque box.

Asset Type	Security Function	Implementation Groups
Network	Detect	2, 3

Dependencies

- Safeguard 18.1: Establish and Maintain a Penetration Testing Program

Inputs

1. `GV54`: Most Recent External Penetration Report

Operations

1. Check Input 1 `GV54` for the date of the most recent external penetration test. Compare the date to the current date and capture the timeframe in months (M1)

Measures

- M1 = Timeframe since the last external penetration test

Metrics

- If M1 is greater than twelve months, then this Safeguard is measured at a 0 and receives a failing score. The other metrics don't apply.

18.3: Remediate Penetration Test Findings

Remediate penetration test findings based on the enterprise’s documented vulnerability remediation process. This should include determining a timeline and level of effort based on the impact and prioritization of each identified finding.

Asset Type	Security Function	Implementation Groups
Network	Protect	2, 3

Dependencies

- Safeguard 18.2: Perform Periodic External Penetration Tests

Inputs

1. `GV53`: Penetration Testing Program Documentation
2. `GV54`: Most Recent External Penetration Report
3. External penetration report prior to most recent report

Operations

1. Use the findings in Input 3 to identify and enumerate the vulnerabilities outlined (M1)

2. Use the findings in Input 2 `GV54` to identify the vulnerabilities outlined

3. Compare the output of Operation 1 and Operation 1

    1. Identify and enumerate vulnerabilities found in Input 3 that continue to be in Input 2 (M2)
    2. Identify and enumerate vulnerabilities found in Input 3 that no longer appear in Input 2 (M3)

4. Using the program documentation from Input 1 `GV53`, determine whether the output of Operation 3.2 is still within scope based on enterprise's policy

    1. Identify and enumerate vulnerabilities within scope (M4)
    2. Identify and enumerate vulnerabilities out of scope (M5)

Measures

- M1 = Count of initial vulnerabilities identified by a penetration test
- M2 = Count of successfully remediated vulnerabilities
- M3 = Count of vulnerabilities that have not been remediated
- M4 = Count of unremediated vulnerabilities still in scope
- M5 = Count of unremediated vulnerabilities out of scope

Metrics

Compliance

Metric	**The percent of successfully remediated or still within scope vulnerabilities identified in the initial penetration test findings.**
Calculation	`(M3 + M4) / M1`

18.4: Validate Security Measures

Validate security measures after each penetration test. If deemed necessary, modify rulesets and capabilities to detect the techniques used during testing.

Asset Type	Security Function	Implementation Groups
Network	Protect	3

Dependencies

- Safeguard 18.1: Establish and Maintain a Penetration Testing Program

Inputs

1. `GV53`: Penetration Testing Program Documentation
2. `GV54`: Most Recent External Penetration Report
3. `GV55`: Most Recent Internal Penetration Report

Operations

1. Check Input 1 `GV53` to determine if it includes an enterprise process for validating security measures after a penetration test

    1. If the process exists, M1 = 1
    2. If the process does not exist, M1 = 0

2. Using the findings from both Input 2 `GV54` and Input 3 `GV55`, as applicable, identify and enumerate security measures that are required modification (M2)

3. For each security measure identified in Operation 2, check if modifications have been made

    1. Identify and enumerate security measures that have been modified per the enterprise's defined process (M3)
    2. Identify and enumerate security measures not yet modified per the enterprise's defined process (M4)

Measures

- M1 = Output of Operation 1
- M2 = Count of security measures requiring modification
- M3 = Count of security measures requiring modification that are properly addressed
- M4 = Count of security measures requiring modification that are not yet addressed

Metrics

- If M1 is 0, this Safeguard receives a failing score. The other metrics don't apply.

Compliance

Metric	**The percentage of security measures requiring modification that have been properly addressed.**
Calculation	`M3 / M2`

18.5: Perform Periodic Internal Penetration Tests

Perform periodic internal penetration tests based on program requirements, no less than annually. The testing may be a clear box or an opaque box.

Asset Type	Security Function	Implementation Groups
Network	Detect	3

Dependencies

- Safeguard 18.1: Establish and Maintain a Penetration Testing Program

Inputs

1. `GV55`: Most Recent Internal Penetration Report

Operations

1. Check Input 1 `GV55` for the date of the most recent internal penetration test. Compare the date to the current date and capture the timeframe in months (M1)

Measures

- M1 = Timeframe since the last internal penetration test

Metrics

- If M1 is greater than twelve months, then this Safeguard is measured at a 0 and receives a failing score. The other metrics don't apply.