Advanced Security: macOS

Table of Contents

Advanced Security: macOS testing methodology v1.0
(12th February 2025)

1. Introduction

The SE Labs Advanced Security: macOS testing methodology is designed to assess the security capabilities of products designed to protect devices running the macOS operating system.

Examples of suitable test subjects include third-party anti-malware software running on Apple desktop and laptop systems.

This methodology is designed to test the abilities of security products to detect attacks and/ or protect against them. These attacks represent threats in the real world and operate in the test framework as if against real targets.

Where possible, practical and relevant, testers use the full attack chain, starting with the first stages of a cyber security attack and progressing as far as possible until a logical conclusion.

The test monitors the test subject's behaviour, and the subsequent report analyses the consequences, presenting clear case-by-case results and applying ratings.

This testing methodology is designed with the goal of allowing test subjects to exercise their features, to the fullest relevant extent. It uses no preconceptions about which approaches they should take.

2. Test framework

2.1 Infrastructure

This methodology supports physical or virtual test subjects.

Testing of physical devices supports:

- Microprocessor architecture: Apple silicon (ARM)
- Operating system: macOS Sonoma 14.x
- The test framework comprises physical Apple silicon-based devices.

Specific details of the equipment used, including model names, firmware versions, configuration details and any virtual hardware details are available to authorised parties engaged in the testing.

2.2 Scope

This test includes threats compatible with both x86 and ARM microprocessor architectures. These threats are a mix of commodity and targeted, APT-style attacks.

2.2.1 Overview

Commodity threats are known, publicly available files sets of files. The test may include variants of these threats that have been altered using techniques designed to evade detection.

Targeted attack testing exposes products to realistic threats that are designed on the behaviour of known attackers. The majority of the following detail relates to targeted threats. This methodology names some example attack groups. These are not necessarily appropriate for macOS testing but are used for illustration purposes.

SE Labs creates a test set by combining variations of test scenarios, as follows.

- A test scenario is based around known behaviour of an attack group (e.g. APT29).
- A test scenario contains different attack stages that represent the attack group. These could be a phishing email; use of valid credentials; theft of further credentials and so on.
- A test case is a combination of attack stages defined by its test scenario. SE Labs creates multiple variations of test cases for each test scenario. See Example test case (‘APT-example’) on page 5.
- The test set is the full range of test cases, organised into appropriate scenarios.

For example, a test set for a report might include a test scenario for APT29, which includes multiple full attack test cases, using different combinations of the appropriate attack components. It might also include a test scenario for the Turla attack group, which in turn contains a number of test cases.

SE Labs organises test scenarios into groups called Threat Series. See

2.2.3 Threat Series on page 6.

All attack stages are confirmed to work on the targets before the security products are installed.

Testing also exposes products to legitimate objects, such as applications and other files. See 2.2.4 Non-Optimal Classification and Action on page 6.

2.2.2 Test structure

A test comprises a test set of attacks that are organised into test scenarios, test cases and attack stages.

Attack stages

In this methodology, SE Labs defines each stage of a targeted attack as follows:

1. Initial Access
2. Execution
3. Action
4. Privilege Escalation
5. Post-Escalation Action

Initial Access: The process by which the attack establishes an initial connection to the target organisation.

Execution: The techniques used the create the initial payload.

Action: The successful execution of techniques that are harmful to the target (user/ network/ organisation).

Privilege Escalation: Attempts to increase access to levels higher than those available to a standard user.

Post-Escalation Action: The successful execution of techniques that are harmful to the target (user/ network/ organisation), using the elevated levels of access obtained through privilege escalation.

Attack stage techniques

Test scenarios contain attack stages in which the attacker interacts with the target. Such stages may include techniques such as:

- Content injection
- Drive-by exploits
- Exploitation of public-facing applications
- Access of external, remote systems
- Phishing emails
- Supply chain compromises
- Use of valid account credentials
- Abuse of trusted relationships
- Malicious insider activity

Test scenarios

Test scenarios represent a series of different attack stages. The cyber security industry often refers to this concept variously as the 'cyber kill chain'; 'cyber attack chain'; or simply 'attack chain'. Where possible and practical, this test methodology aims to use full attack chains.

SE Labs aims to create test scenarios that follow the general (and often quite specific) characteristics of named advanced attack groups, often referred to as Advanced Persistent Threats (APTs). For example, APT29, Scattered Spider and Turla.

Test scenarios are specified in each test plan and any subsequent reports. They are based on publicly available information. SE Labs maps key points within attack scenarios to MITRE's ATT&CK Matrix for Enterprise.

Example test case ('APT-example')

The test case below contains a series of attack stages (see Attack stages on page 4).

In combination they relate to the approach taken by the specific attack group represented in the test. In this example, the hypothetical threat group is called 'APT-example'.

Each test case is built using a series of attack stages (Initial Access, Execution etc.).

Each stage uses certain attack techniques. There may be one or more, depending on the attack group's behaviour. These techniques are labelled using a code (e.g. A drive-by compromise uses the code T1189). This code matches the reference codes provided in the MITRE ATT&CK Enterprise Matrix.

EXAMPLE TEST CASE (SCENARIO: APT-example)					
Attack Stage	**Initial Access**	**Execution**	**Action**	**Privilege Escalation**	**Post-Escalation Action**
Test case #1	T1566.003	T1059.004	T1083	T1543.001	T1040
		T1204.002	T1057		T1105
		T1078.003	T1049		T1555.001
		T1078.002	T1033		T1573.002
			T1082		T1222.002
					T1119

2.2.3 Threat Series

A test may contain multiple test scenarios (Attacker/ APT groups). For example, a product might face a range of attacks representing:

1. APT29
2. APT3
3. OilRig
4. APT33

SE Labs can test with any individual test scenario (e.g. APT29) or a combination. For the sake of efficiency, there are standardised sets of Attack Groups called the SE Labs Threat Series. The above list of test scenarios comprises Threat Series 1.

The current list is published in the Review Guides section of the SE Labs website.

2.2.4 Non-Optimal Classification and Action (NOCA)

Testing products with legitimate objects helps ensure that they are configured correctly. In this test the objects are based on work-related scenarios commonly found in enterprise, small business or consumer environments, according to the products under test.

NOCA testing is a more wide-reaching assessment than basic 'False Positive' testing, where only completely wrong classifications are important (e.g. a legitimate file is detected as a 'virus').

A NOCA result handles less definite detection terms such as 'suspicious', 'unknown' etc. and actions ranging from the definite ('block', 'allow') to the more advisory ('allow but advise manual blocking').

SE Labs may disclose legitimate objects to test participants in advanced of the testing.

The product's configuration will not change between the attack and NOCA stages of testing.

2.3 Configuration

Product configurations are made as per best practices and, ideally, as specified by the technology provider for the purposes of the test.

Configuration details are included in any subsequent report of the test's results. These may be in the form of printed output from the products; links to vendors' best practice; or exported configuration files on the SE Labs website.

3. Results

3.1 Observations

Testers see the attacks from both the attacker's and defender's perspective. They launch the attacks and attempt to take control of targets, while also monitoring the detection and protection mechanisms provided by the security product. In this way they can determine at which stages of the attack the product successfully detects and possibly mitigates the threat.

When running a protection test, the tester persists with the attack until such point as the product prevents the attack. The initial report contains detailed observations made at each discrete stage of the attack, determining how far along the attack chain it stopped the attack, if at all.

Initial data from the test includes details of how the attacks were run, including terminal input and output. The product's detection and protection logs are also included.

3.2 Analysis

Tested solutions should detect the malicious behaviours described in the test set.

There are different ways for products to present alerts. While direct references to the relevant technique codes in the MITRE ATT&CK Enterprise Matrix are valuable, they are not a requirement in this test. An accurate description in a dashboard or logs is sufficient.

The following shows an inexhaustive list that indicates results that are relevant and valuable to an analyst in the real world (valid) and results that are incorrect, misleading or missing:

Valid results:

1. “Incident involving technique T1189”
2. “Drive-by compromise”
3. “Drive-by exploit”
4. “Exploit toolkit”

Invalid results:

1. “Generic virus/ Trojan”
2. “Keylogger"*
3. “NIMDA worm”*
4. {nothing}

* If the threats are not a keylogger or the NIMDA worm, for example.

3.2.1 Threat Scoring

The following scoring scheme provides different levels of credit or penalty for varying responses to threats. In each test case the tester analyses the security product’s response to the threat through each of the five attack stages. Complete success adds four points to the product’s score.

Threat scoring is made as follows:

Detection (+1): The product detects the threat with any degree of useful information.

Blocked (+2): The product prevents the threat from carrying out any significant malicious activities.

Neutralisation (+1): The product fails to instantly block the threat but subsequently terminates running malicious processes, ‘neutralising’ the threat.

Complete Remediation (+1): After neutralising a threat, the product removes all significant traces of the attack. A Blocked threat implies Complete Remediation also.

Compromise (-5): The product fails to stop the threat.

Penalties are applied according to failures at each of the following stages, to a maximum of -5 points.

Initial Access (-0.5)
Execution: (-0.5)
Action: (-1)
Privilege Escalation: (-2)
Post-Escalation Action: (-1)

A successful remediation of the system within 60 seconds of the attack removes penalties from the Initial Access and Execution stages.

3.2.2 NOCA scoring

The following scoring scheme provides different levels of credit or penalty for varying responses to legitimate objects.

The main criteria relate to how the product classifies an object, if at all, and what recommendation or action is made. Ideally the recommendation or action would be to allow the legitimate object without causing disruption, such as requiring an administrator to take action (e.g. releasing from a quarantine system; adding to an allow-list or otherwise overriding the security system.)

Classification (Alert Type)	**Recommendation/ Action**	**Score**	**Description**
None	Allow	+10	The test subject provides low priority alerts but does not classify the legitimate object inappropriately.
Medium	Allow	+7	Following the test subject's recommendation interferes with the legitimate object, to the extent that operations are disrupted
High	Block	-10	Following the test subject's recommendation interferes with the legitimate object, to the extent that security exceptions are required to complete the test

Example threat results

The following table applies observations to

Example test case ('APT-example') on page 5.

A result highlighted in green is a completely successful detection and protection. A result highlighted in red is a completely missed detection and protection. A result with no highlighting means that the technique is out of scope for this particular test.

EXAMPLE TEST CASE (SCENARIO: APT-example)					
Attack Stage	**Initial Access**	**Execution**	**Action**	**Privilege Escalation**	**Post-Escalation Action**
Test case #1	T1566.003	T1059.004	T1083	T1543.001	T1040
		T1204.002	T1057		T1105
		T1078.003	T1049		T1555.001
		T1078.002	T1033		T1573.002
			T1082		T1222.002
					T1119

4. Change log

Date	Version	Change
12th February 2025	1.0	Document created