Keywords: anti-malware; scam; phishing, testing process; guidelines

AMTSO
2026-04-08

Version 1.0

AMTSO Scam and Phishing Testing Guidelines

Abstract

These Guidelines provide technical definitions and details required for effective testing of scam and phishing attack vendors in Test Subject Vendor products.

Notice and Disclaimer of Liability Concerning the Use of AMTSO Documents

This document is published with the understanding that AMTSO members are supplying this information for general educational purposes only. No professional engineering or any other professional services or advice is being offered hereby. Therefore, you must use your own skill and judgment when reviewing this document and not solely rely on the information provided herein.

AMTSO believes that the information in this document is accurate as of the date of publication although it has not verified its accuracy or determined if there are any errors. Further, such information is subject to change without notice and AMTSO is under no obligation to provide any updates or corrections.

You understand and agree that this document is provided to you exclusively on an as-is basis without any representations or warranties of any kind whether express, implied, or statutory. Without limiting the foregoing, AMTSO expressly disclaims all warranties of merchantability, non-infringement, continuous operation, completeness, quality, accuracy, and fitness for a particular purpose.

In no event shall AMTSO be liable for any damages or losses of any kind (including, without limitation, any lost profits, lost data, or business interruption) arising directly or indirectly out of any use of this document including, without limitation, any direct, indirect, special, incidental, consequential, exemplary, and punitive damages regardless of whether any person or entity was advised of the possibility of such damages.

This document is protected by AMTSO's intellectual property rights and may be additionally protected by the intellectual property rights of others.

Contents

AMTSO Scam and Phishing Testing Guidelines

1. Introduction

In today's digital landscape, the internet has become the primary battleground for cyberattacks. Cybercriminals employ various tactics to deceive and exploit unsuspecting users. The increasing frequency and sophistication of cyber-attacks have led to a surge in demand for web protection solutions. Numerous cybersecurity companies offer products and services that claim to safeguard users from online threats. However, the effectiveness of these solutions varies significantly.

To ensure that users are adequately protected, it is essential to rigorously test web protection solutions. This testing process should evaluate the ability of the solution to detect and block various types of cyber-attacks, including those mentioned above. By identifying the strengths and weaknesses of different solutions, users can make informed decisions about which products to trust.

Independent testing organizations play a crucial role in evaluating the effectiveness of web protection solutions. These organizations conduct unbiased tests and publish their findings, allowing users to compare the performance of different products. This document aims to provide guidance on conducting unbiased tests and ensuring impartial test results.

1.1. Contributing Members

AMTSO acknowledges the contributions made to the construction of these Guidelines by the following individuals and companies along with the general membership of the AMTSO Scam and Phishing Working Group.

Dennis Batchelder (AppEsteem)
Stefan Dumitrascu (Artifact Security)
Erica Marotta (Artifact Security)
Marcel Wabersky (AV-TEST)
Erik Heyland (AV-TEST)
Megan Squire (F-Secure)
Jan Sirmer (Gen Digital)
Paul Walker (McAfee)
Zenonas Funka (Nord Security)
Artur Lagutin (Nord Security)
Dainius Razinskas (Nord Security)
Domininkas Verbickas (Nord Security)
Andrey Voitenko (VMRay)

2. Core Definitions

Spam refers to an unsolicited message designed to promote or distribute digital content to a mass audience without individual targeting. Spam is often the source of indirect harm including wasted resources (time, digital storage, system processing), reduced productivity, and might provide a gateway for additional harm. Examples include unsolicited e-mails, advertisements, SMS messages, and robocalls.

Scam refers to an unsolicited message that leverages manipulative tactics designed to deceive the recipient and induce them to take an action that benefits the sender at the expense of the recipient. Just like spam, a scam message is one the recipient did not request or initiate and is typically unexpected, but has purpose for the sender. Manipulative tactics attempt to bypass rational decision-making through false urgency, emotion, impersonation, language, or requests for non-public information. The inference of deceptive intent from the consistent use of manipulate tactics and false premises distinguishes scam from spam, where the manipulative tactics are to convince the user to, for example, exchange money for a product or service, whereas for a scam it may be for nothing or significantly less than expected. Scams inherently aim to cause harm to the recipient which can take various forms, including financial loss, data or identity theft, unsolicited software installation, and/or emotional distress.

Manipulative tactics are designed to create a false impression by exploiting a human trait. These tactics exploit trust (impersonate someone you trust), diligence (impersonate an ordinary administrative task), ambition (exclusive deals or opportunities), self-esteem (developing relationships with strangers), uncertainty (engender a sense of danger or risk), and/or curiosity (leverage secrecy, intrigue, or news). Such tactics are not intended to provide testing criteria which should be reported but can provide clarity regarding the human attack vectors underpinning a scam.

Phishing is a form of social engineering and a scam where the attacker attempts to deceive the recipient into revealing sensitive information or installing malware such as viruses, worms, adware, or ransomware. Phishing methodologies can revolve around impersonation or techniques that makes the attacker appear legitimate.

Phishing attacks include spoofing telephone calls pretending to be a trusted caller, using email to bait victims to install malware or malicious software, sending mass amounts of spam to trick individuals into entering a fake website, or inserting a fake URL in their messages or within website content. Such attacks target information that can be used in further scams or immediately exchanged for financial gain. Such targets include passwords, login credentials, credit card numbers, social security numbers, and personal banking details.

3. General Test Design Considerations

When designing a test, whether it be an evaluation of a single solution's capabilities or a comparative of multiple solutions, close attention should be paid to ensuring the test is fair, balanced, and unbiased. The aim of such testing is to provide useful information on the quality of the solution under test and the level of protection it provides.

3.1. Product Selection

Solutions should be selected for testing based on claims made regarding their ability to protect against the types of threats being considered for evaluation. Where a solution provider does not make specific claims in the area being analyzed, but where a general user might expect protections to be included, a tester may choose to include the solution in the test, but should make clear that no specific claims of efficacy have been made.

When preparing solutions for testing, testers should aim to activate and exercise all components of a solution. Where a test focuses on specific sub-components of a broader protective suite and some other features are disabled or not installed, the test methodology and report should highlight this fact and provide justification for the approach taken.

3.2. Test Environment

Testers may choose to limit their test to a specific platform or attack vector, and in such cases should select products which support the chosen environment and threat vectors. In many cases, both threats and protections may extend across multiple platforms and vectors, such as messaging received on mobile devices directing target victims to take actions on computers (and vice versa), and a more comprehensive test should include the full environment impacted by the threat scenarios and covered by the solutions being analyzed. This may include multiple devices and services, including cloud-based provision accessed from different devices.

In general, devices used for testing should be representative of those in general use, with latest versions of operating systems in place and latest updates applied. Where a tester chooses to apply a different selection of versions and/or updates, this should be made clear in the test methodology and report, along with justification for the choices made.

Test systems should be reverted to a clean state prior to installation of solution components and execution of each test case, to ensure no contamination of results from previous testing. Connectivity to resources required by protective solutions and by test cases should be verified prior to testing.

3.3. Test Case Selection

Test cases for scam and phishing testing fall into two broad categories: “real-world” examples, harvested or captured from the internet, and “artificial” test cases created or modified by the tester to suit their testing requirements. Examples of real-world test cases might include sets of URLs known to be hosting phishing pages, or more sophisticated data sets combining both the online resources and penetration vectors used by the original attacks. Artificial test cases might include “simulated” phishing sites emulating techniques used by real-world attackers, or more advanced scenarios covering the full attack chain of a scam or phishing attack, where interactions between the “attacker” and the “victim” can be carried out manually by the tester, or automated.

> *Informational: Types of Scam*
>
> *Scams take many different forms, targeting different sorts of people in varied situations with multiple end-goals. The most common end-goal is of course financial, whether it be persuading*

someone to remit payment for fake or non-existent goods, tricking someone into granting access to banking and other money-moving systems, or deceiving someone into revealing login details for another type of system or service which the scammer can then monetize. The "fake shop" scam type is widely deployed, often luring in users with misleading spam advertising and offering either openly admitted counterfeit goods (which may or may not ultimately be provided to the purchaser) or fully spoofing legitimate traders, complete with branding etc, which can be highly convincing. High-end and luxury goods are a particular target of this type of scam. Tests measuring detection of fake shops have already been designed and run [see note 6], using databases of known real-world examples as a test bed. A more extreme niche of this marketplace is the "fake pharma" scam, with significant infrastructure built up including payment systems. Target markets often include regions where specific medications are unavailable, and conditions regarding which the sufferer may wish to retain a greater degree of privacy or anonymity – this can be actively harmful beyond mere loss of money, as in some examples counterfeit, contaminated, or entirely spurious medications have been shipped to scam victims [see note 5].

When using corpuses of existing attacks, testers may choose to replicate each attack as accurately as possible, or may introduce the threats in a simplified or modified manner. Where test cases are introduced to the test system in a different way than they arrive in real-world cases, this should be made clear as the change of vector may impact the response of protection solutions and render results less accurate than using more representative vectors.

Informational: Penetration Vectors

Scam operators make contact with their victims via a wide range of routes, including but not limited to email (including both untargeted spam and more personalized spearphishing), messaging services, social media, dating sites, job boards, search optimization or advertising, voice calls and SMS, and even "snail mail". During the course of a complex scam the communication vector may change, such as switching from social media chat to one-to-one messaging. To properly replicate the full "attack chain" of a scam, testers may need to manage multiple comms vectors, potentially on different devices, and should ensure that protective solutions have access to monitor any vector covered by their protective features.

Testers should be aware that online resources used in scams often have a very short time-to-live, as they may be disabled or taken down by law enforcement, hosting providers, or their own operators. Testers should confirm that any online resources in the test case attack chain are active and operational at the time of testing; failure to alert about or block access to a non-functioning resource should not be considered a failure of protection.

Informational: Test Case Collection Approaches

Testers can obtain their test cases through a variety of routes, including in-house harvesting and collection of URLs of known phishing messages and sites, or accessing collections aggregated by a third party. When using third-party sources testers should take into account the likelihood that security providers will also have access to such sources. It is also necessary to verify test cases

gathered in this way, as inclusion in a public or private listing of known-bad sites does not guarantee that the site will be active, or serving the same content, at the time of testing.

A more advanced approach to test case collection could include creation of a number of in-house "identities", with a realistic footprint in a range of areas including social media, messaging and email. These identities can be maintained over periods of time and used as "honeypots" to attract real-world scammer attacks, from which the attack chains can be harvested in full or in part to use in testing.

To avoid issues with test cases ceasing to operate, testers may choose to create or simulate their test cases. Care should be taken to ensure such simulations accurate reflect real-world conditions; as the reputation of URLs and other resources is a key tool in many protection systems, testers should take into account the fact that the reputation (or lack thereof) of their simulation could have a serious impact on the ability of protective solutions to properly identify and protect against them. Testers may choose to artificially apply a suspicious or known-bad reputation to their simulated test cases, for example by reporting to reputation-monitoring systems; again, testers should be aware that this may not accurately reflect real-world scenarios and should take this into account in their evaluation processes. "Seeding" an artificial test case in intelligence-sharing systems can support some forms of time-to-detect testing, but care should be taken to avoid polluting community sharing systems with unnecessary and irrelevant data.

3.4. Confirmation and Verification

As with all testing, testers should confirm and verify their results prior to publication. Testers may opt to include the test subject developers in this process, providing pre-publication access to test data and allowing vendors to challenge unexpected or questionable outcomes. Where a test case is judged to be unsuitable after a dispute from a participant, it is important that the outcomes of that test case are removed from the results of all test subjects to ensure balance. The AMTSO Testing Protocol Standard provides a framework and further guidance to facilitate communications between testers and test subjects.

Also common to most tests, an element of false-positive testing is also required. Testers should include in their test methodology an approach to checking for false alarms, covering the same vectors used elsewhere in the test. This could include browsing to legitimate shopping and banking sites, and other commonly-used web resources, as well as sending and receiving legitimate emails and messages, and monitoring for any warnings, alerts or advice provided by the security solution.

3.5. Scoring and Rating

Measuring the effectiveness of scam & phishing protections can be more complex than in many other types of cybersecurity evaluations, as the outputs and alerts are often granular with different degrees of warning provided to users, and in some cases warnings may be ambiguous or provide only general guidance to the user, rather than a definitive decision on whether something can be trusted.

Testers should develop a rigorous scoring process taking such ambiguities into account, document

their approach in their test methodology and reports, and apply it equally across all tested solutions. Simpler scoring methods can be applied to some types of test, such as time-to-detect or time-to-alert measurements. False-positive measurements should also be included in scoring calculations, and performance in different areas of the test should be weighted appropriately according to their significance.

4. Social Engineering and AI-driven User Guidance

Scams and phishing generally involve a significant element of social engineering – manipulation of human behaviors to achieve the scammer's desired outcome. This may be as simple as an email, message or website posing as a legitimate provider of goods or services, or a much more complicated chain of interactions whereby the scammer builds trust and even "friendship" over a period of time, and potentially using multiple contact vectors, before coming to the point of extorting money or information. While technological measures can mitigate against some forms of deception, such as checking that the source and other identifying elements match the purported company or person, in many cases there is limited opportunity for direct technical intervention.

To help deal with more "human" cases, many security providers have developed guidance tools, often AI-driven, to help users identify potentially deceptive material. In the main, these involve uploading text or images for analysis, returning with a report estimating the likelihood of the situation being analyzed leading to a scam. These responses may have higher or lower levels of certainty, and may provide different degrees of user guidance. Such tools rely on the user having sufficient initial suspicion to submit the case for review.

Testing such measures can be complex and time-consuming. To achieve a fully realistic test, the tester may choose to set up a set of fake "identities" (as described above), harvest all communications received by these identities, and submit them for review by the AI tools. Further steps might even include interacting with anyone who contacts the fake identity to move further along the potential scam attack chain, and submitting each step of communications. The tester can then rate the accuracy, certainty level, and user guidance provided – to achieve this fairly and evenly, a detailed rating process should be developed, clearly explained in the test methodology, and applied equally across all solutions tested. For balance, tests should also include legitimate communications, either from the fake identity or from real identities, to measure the level of false alarms and inaccurate guidance. False positive tests should be included at an appropriate level relative to the quantity of actual scams being evaluated.

5. Appendix A: Example Testing Scenarios

The following are examples of testing scenarios which testers may choose to implement as part of their evaluations, detailing the steps to create and run the test. Testers are invited to contribute their own scenarios to AMTSO to include in this community resource.

5.1. Detection Time to Block a New Phishing Site

This test is designed to measure how quickly cybersecurity solutions can identify and block a newly created phishing site that mimics real-world techniques. Specifically, this test involves the deployment of a controlled phishing website and the metrics collected to analyze performance.

The test has four discrete steps.

1. Phishing Site Creation
    - Set up a domain and host a phishing page (e.g., fake login form resembling a known service like Gmail, Facebook, etc.).
    - Ensure realistic design, SSL certificate, and evasion techniques (e.g., obfuscation, delay scripts).
2. Time Zero
    - Record the exact time when the phishing page is made publicly accessible (with global DNS propagation).
3. Monitoring
    - Test multiple security solutions (browsers, endpoint security, DNS filtering tools, etc.) by attempting to access the phishing page at regular (hourly) intervals.
    - Logging requirements include time until first detection/block, detection method (URL block, page scan, content inspection), type of warning shown (hard block, soft warning, ignored), and false positives (if any)
4. End Condition
    - Test ends when all solutions block the site or after a maximum test duration (e.g., 72 hours).

Among the metrics collected are the following.

- Time-to-detect per solution (The time it takes for a solution to block the phishing site.)
- Blocking method (The method used by the solution to block the site (URL block, page scan, content inspection)).
- User warning clarity
- Detection consistency across systems/OS/browsers

Compliance and Ethics concerns include the following.

- Make sure the site is isolated, hosted in a controlled environment, and uses non-functional credential submission (e.g., forms that do not store or transmit data).
- Mark it in robots.txt and metadata to discourage indexing.
- Inform vendors in advance if you are testing under a cooperative model (or not, if blind-testing).

5.2. Embedded Phishing Page Detection

This test is designed to verify whether security solutions can detect and block phishing content embedded within a legitimate-looking parent site using <iframe> or <object> tags. Phishing actors often embed malicious login forms or fake websites into otherwise harmless pages using HTML embedding techniques. This test assesses whether security products can inspect embedded content and not just the outer page.

This test has established preconditions that include a controlled environment with multiple cybersecurity products (browsers, endpoint software, DNS filters), a legitimate-appearing parent site is hosted under test administrator control, and a separate phishing-like page (e.g., fake login page) hosted elsewhere under test administrator control.

The test has four discrete steps.

1. Create Embedded Phishing Page Setup
    - ○ Host a parent page at parent.example.com that appears benign.
    - ○ Embed a phishing-like page from malicious.example.com using:
      <iframe src="https://malicious.example.com/login.html"
      width="600" height="400"></iframe>or
      <object data="https://malicious.example.com/login.html"
      width="600" height="400"></object>
2. Time Zero (Activation)
    - ○ Make the embedded page publicly accessible and note the time.
3. Testing Phase
    - ○ Open the parent page in various environments and note whether:
        - ■ The parent page loads fully.
        - ■ The embedded content is displayed, blocked, or warned about.
        - ■ The solution blocks the embedded domain (not just the outer shell).
    - ○ Repeat this at regular intervals (e.g., every 30 minutes) to monitor delayed detection.
4. Edge Variants (Optional)
    - ○ Add JavaScript-based dynamic loading of the iFrame.
    - ○ Use content obfuscation in the phishing page to bypass static detection

Possible expected results and outcomes include the following.

Case: Parent page loads, embedded content is phishing
Expected Behavior: Security software should block the embedded content regardless of parent page's legitimacy.
Case: Embedded phishing page is served via HTTPS
Expected Behavior: Detection should not be bypassed due to SSL.
Case: User interaction triggers form submission
Expected Behavior: Software should block submission or warn.
Case: Delayed load (e.g., JS injection)
Expected Behavior: Detection should occur even if phishing content loads after a delay.

Among the metrics collected are the following.

- Block status of embedded content
- Block status of parent page (if triggered)
- Time-to-detect for embedded phishing
- Method of detection (URL-based, content-based, behavioral)
- False positives on benign embedded content (if you test against control cases)

Compliance and Ethics concerns include the following.

- All phishing pages used should be non-functional, contain dummy content, and not store any user input.
- Hosted in an isolated test environment, preferably on private infrastructure.
- Should not use branding or trademarks unless permission is granted or working under a security testing exception framework

5.3. Exploiting Deprecated Java Script Functions

This test is designed to test a web protection systems' ability to detect and mitigate malicious behavior implemented through deprecated or legacy JavaScript functions. These functions can be exploited for obfuscation, delayed redirection, or stealth form injection in phishing attacks. To assess whether security products can detect and mitigate phishing or malicious behavior implemented through deprecated or legacy JavaScript function.

Modern phishing toolkits sometimes use legacy JavaScript functions that are still supported by browsers but are less scrutinized by modern security engines. Functions like eval(), unescape(), document.write(), and setTimeout("code") can be used to obfuscate code or delay malicious behavior, making detection harder.

This test has established preconditions that include a controlled test environment with modern browsers such as Chrome, Firefox, or Edge, and a Java Script implementation. Additionally, a test domain capable of hosting and serving scripted phishing pages and a security software or solution to do the detection are required.

The test has four discrete steps.

1. Prepare a Malicious JavaScript Payload
   Example: var code =
   "%64%6f%63%75%6d%65%6e%74%2e%6c%6f%63%61%74%69%6f%6e%20%3d%20%27%68%74%74%70%3a%2f%2f%70%68%69%73%68%20%73%69%74%65%2e%63%6f%6d%27";

   eval(unescape(code));

2. Host the Payload - Serve the JavaScript from a test domain

3. Launch the Test - Access the malicious script in browsers and monitored environments

4. Variants (Optional) - Use document.write(), setTimeout(), etc.

Possible expected results and outcomes include the following.

- Obfuscated redirect with eval(unescape()): should be detected and blocked.
- Script injects login form via document.write(): should be blocked.
- setTimeout("code", 2000): delayed execution should be caught.
- Script executes with no alerts: failure to detect.

Among the metrics collected are the following.

- Time-to-detect Time taken to detect obfuscated or delayed scripts
- Block method Pre-execution, runtime, or no block
- Detection source Static scan, content inspection, heuristic
- User warning clarity of warnings presented to the user
- Consistency across environments
- Number of false positive detections

Compliance and Ethics concerns include the following.

- JavaScript must be non-destructiveUse safe redirection (e.g., to example.com)
- Scripts should not collect or transmit data
- Test domains should be isolated and not indexed
- Do not mimic real services unless ethically authorize

5.4. Identity Testing

Attackers leverage identity to gain access to a target organization via multiple methods. Examples of this are but not limited to: Token stealing (ATT&CK T1528), Kerberos Ticket stealing or forging (ATT&CK T1558), Multi-Factor Authentication Request Generation (ATT&CK T1621). These scenarios assess the security solution ability to assess typical user behaviour, login method patterns, and location/resource misuse.

Preconditions:

- Target network of human & non-human identities (workstation, servers, applications etc.)
- Legitimate traffic/behaviour ongoing before and during the testing phase

For detecting techniques related to anomalous login it is recommended to have a *"learning period/baseline behaviour"* where the security solution is informed of the usual behaviour of the identities under test. Legitimate staff-like operations should occur during this period and during the testing phase.

- For example: Communication patterns between internal identities via email/IM applications, application use of a typical target organisation should be reflected.

The test scenario has the following discrete steps for evaluation of token stealing.

Initial Access – via phishing page or infostealers targeting typical token storage locations.

Token Extraction – via MiTM attacks, memory access, session cookie stealing
Token Reuse – Authenticate as the victim or Lateral Movement

The test scenario has the following discrete steps for evaluation of MFA request generation.

Initial Access – via phishing or valid credentials
MFA trigger – initial push notification sent
MFA flood – high volume of notifications sent to targeted users. Depending on the provider the rate at which MFA thresholds change the test case should adapt as such.

Identity based attacks are very common ways to gain initial foothold and/or move laterally across an organisation. As such a clear scope of the attack should be defined for what metrics should be considered.

For example, replicating a "malware-free" attack can be done only via Identity based techniques with the sole goal to achieve persistence in cloud footprint of an organisation. While it's not deploying a file-based malware, maintaining access to the target organisation with the purpose to intercept communication patterns between the target and its external suppliers is a long process that can last weeks or months until the attacker deploys something more disruptive or using BEC methods to redirect funds for their own gain.

6. Appendix B: Reference Materials

[1] "What is phishing, and how does it work?", NordVPN, https://nordvpn.com/blog/what-is-phishing/

[2] "Scams: A definitive guide", NordVPN, https://nordvpn.com/cybersecurity/glossary/scam/

[3] "Sham Definitions", AppEsteem, https://customer.appesteem.com/home/shamdefinitions

[4] "Scam and Phishing Evaluation", Artifact Security, https://www.artifactsecurity.co.uk/scam-phishing

[5] "PharmaFraud: how illegal online pharmacies endanger your health and your wallet", Avast Security, https://blog.avast.com/pharmafraud-fake-pharmacies

[6] "Fake-Shops Detection Test", AV-Comparatives, https://www.av-comparatives.org/fake-shops-detection-test-2025/