Keywords: anti-malware; compliance; assessment; testing; test plan; MRG Effitas; 360 Degree Assessment

January 18, 2021
Publication Date : January 19, 2021

Version 1.1

MRG Effitas Test Plan for Q1 2021 360 Degree Assessment and Certification

Sponsored and Authored by:
MRG Effitas (Lorand Lajsz)

AMTSO Standard Compliance Statement
This Test has been designed to comply with the Anti-Malware Testing Standards Organization, Inc. ("AMTSO") Testing Protocol Standard for the Testing of Anti-Malware Solutions, Version [1.3] (the "Standard"). This Test Plan has been prepared using the AMTSO Test Plan Template and Usage Directions, Version [2.3]. Effitas Ltd. is solely responsible for the content of this Test Plan.

Table of Contents

MRG Effitas 360 Degree Assessment and Certification Test Plan

Q1 2021

1. Introduction

A first-of-its-kind test that covers all angles, our pioneering 360 Degree Protection Test targets the key threats faced by internet users. In each test case we employ the full spectrum of Early Life Malware. We use a Time-To-Detect metric to measure how long it takes each application to detect and neutralize missed threats.

MRG Effitas has a core focus on efficacy assessments in the anti-financial fraud space, but we also publish more traditional "Real World" detection tests. Our "Time to Detect Assessment Q4 2013" measured the ability of security Test Subjects to protect an endpoint from a live infection, and, in the event of a system being compromised, the time taken to detect the intrusion. The time-to-detect component relied on each security product being manually forced to conduct a scan every thirty minutes over a 24-hour period. For 2014, it was decided that a new approach was needed as the methodology applied in previous tests did not reflect how a security product would be used on an endpoint in the Real World.

In practice, many security applications will only detect an infection during a reboot/startup or if a scheduled scan has been set by default. For this assessment, time-to-detect will employ a methodology based on the endpoint being re-tested once after a 24-hour period.

This Programme is called a "360 Assessment" since it deals with the full spectrum of malware instead of just financial malware. In the 360 Assessments, trojans, backdoors, ransomware, financial malware and "other" malware are used.

2. Scope

This Certification Programme is designed to serve as a reflection of product efficacy based on what we have previously termed "metrics that matter". In many of our previous tests, particularly those that have focused on financial malware, we started with the assumption that the endpoint has already been compromised. Being one of the world's largest supplier of early-life malicious binaries and malicious URLs, and from our own simulator development, we know that all endpoints can be infected, regardless of the security solutions employed.

For us, a product's ability to block initial infection (although critical in most cases) is not the only metric that matters. One also needs to measure the time taken for the security product to detect malware on a system. When conducting these tests, we tried to simulate normal user behaviour. We are aware that a "Real World" test cannot be conducted by a team of professionals inside a lab because we understand how certain types of malware work, how malware attacks and how such attacks could be prevented. Simulating normal user behaviour means that we paid special attention to all alerts given by security applications.

A pass was given only when alerts were straightforward and clearly suggested that malicious action should be blocked. Meanwhile when a threat was detected but not blocked, this will be counted as detected. We tested a group of internet security suites and complementary security applications. With these, it is very important to note that the best choice for an average user is to keep things very simple and for the product not to present many pop-up alerts or questions.

In our next execution of this test plan, thirteen test subjects were considered. Companies expected to be represented and their target Test Subjects include the following. Specific Test Subject Vendors and Participants will be determined after the Public Test Notification has been issued.

- Avast Business Antivirus
- Avira Antivirus Pro - Business edition
- Bitdefender Gravityzone Advanced Business Security – cloud
- CrowdStrike Falcon Protect
- ESET Endpoint Security
- F-Secure Protection Service for Business
- Malwarebytes Endpoint Protection
- Microsoft Windows Defender
- Sophos Intercept X
- Symantec Endpoint Protection
- Trend Micro Worry-Free™ Services with XGEN

3. Methodology and Strategy

Since its inception in 2009, MRG Effitas has strived to differentiate itself from traditional testing houses by having its primary focus on providing "efficacy assessments" and not just performing "tests". Traditionally, testing of security software has centred on measuring a product's ability to detect malware.

Testing has evolved rapidly over the last two to three years as most labs, under the guidance of AMTSO (of which MRG Effitas is a member) strived to conduct "Real World" testing. Although there is no absolute definition of this kind of testing, loosely speaking, it involves the introduction of malware to an endpoint through a realistic vector, such as a browser or USB memory stick. Real World testing mostly involves "dynamic testing" (i.e. the malware is executed and then the ability of the security product to block the malware is measured). Several testing labs also conduct "System Rescue" tests. These assess a security product's ability to remediate a pre-infected endpoint.

Whilst both types of tests are useful and yield valid and meaningful data, MRG Effitas wanted to merge these tests and also go one step further by measuring the time security Test Subjects take to detect infections. The system was retested 24 hours after the system was first compromised, thereby giving security applications the opportunity to detect infections on restart. Measuring initial detection rates and the time taken to detect active malware is important, particularly in today's threat landscape with the mix of malware that is prevalent. As we have repeated in our previous financial malware test reports, the longer a cybercriminal can have their malware on a system, the greater the opportunity for them to be able to capture private user information including banking passwords and social media credentials, etc.

In providing these quarterly certifications, the MRG Effitas 360 Assessment & Certification Programme is the de facto standard by which security vendors, financial institutions and other corporations can attain the most rigorous and accurate determination of a product's efficacy against the full spectrum of malware that is prevalent during the period.

In-The-Wild malware test methodology

1. Windows 10 Enterprise 64-bit operating system is installed on a hardened virtual machine, all updates are applied, and third-party applications installed and updated.
2. An image of the operating system is created.
3. A clone of the imaged systems is made for each of the security applications used in the test.
4. An individual security application is installed using default settings on each of the systems created in (3) and then, where applicable, updated. If the vendor provided a non-default setting, this setting is checked whether it is realistic. If yes, the changes are documented, applied, and added to the appendix section of the report.
5. A clone of the system as at the end of (4) is created.
6. Downloading a single binary executable (or document, script, etc.) from its native URL using Chrome to the Downloads folder and then executing the binary in the clean, unprotected system. If the sample works, the sample is saved in a replay proxy to provide the same binary throughout the test.

Live URL test is conducted by the following procedure.

6.1. The sample is selected for the test and tested in the systems where a security product is installed.

6.2. The test case is retested 24 hours after the initial test if the security application failed to block the malicious binary.

Spam e-mail attachment test is conducted by the following procedure.

6.3. Microsoft Office Outlook client downloading a single email from its server to the victim system created in (4).

6.4. Opening the e-mail, saving the attachment to the Downloads folder and then executing the binary.

6.5. The test case is retested 24 hours after the initial test if the security application failed to block the malicious binary.

- **The test case is marked as "Blocked"** by either the security application blocks the URL where the malicious binary was located. Or the security application blocks the malicious binary whilst it was being downloaded to the desktop.
- **The test case is marked as "Behaviour Blocked"** if the security application blocks the malicious binary when it is executed and either automatically blocks it or postpones its execution and warns the user that the file is malicious and awaiting user input.
- **The test case is marked as "Detected"** if the security application detects the threat and sends an alert to the central console or notifies the user, but the sample is allowed to run.
- **The test case is marked as "Blocked in 24h"** if the security application fails to block or behaviour block the malicious sample but blocks it during the retest.
- **The test case is marked as "Missed"** if the security application fails to block or behaviour block the malicious sample during both tests.

7. Tests are conducted with all systems having internet access.
8. As no user-initiated scans is involved in this test, applications rely on various technologies to detect, block and remediate threats. Some of these technologies are URL blacklisting, reputation, signature, machine learning, heuristics, behaviour etc.

In-The-Wild PUA/Adware test methodology

1. Windows 10 Enterprise 64-bit operating system is installed on a hardened virtual machine, all updates are applied, and third-party applications installed and updated.
2. An image of the operating system is created.
3. A clone of the imaged systems is made for each of the security applications used in the test.
4. An individual security application is installed using default settings on each of the systems created in (3) and then, where applicable, updated. If the vendor provided a non-default setting, this setting is checked whether it is realistic. If yes, the changes are documented, applied, and added to the appendix section of the report.
5. A clone of the system as at the end of (4) is created.
6. Downloading a single binary executable (or document, script, etc.) from its native URL using Chrome to the Downloads folder and then executing the binary in the clean, unprotected system. If the sample works, the sample is saved in a replay proxy to provide the same binary throughout the test.
7. The sample is selected for the test and tested in the systems where a security product is installed.
8. The test case is retested 24 hours after the initial test if the security application failed to block the malicious binary.

- **The test case is marked as "Blocked"** by either the security application blocks the URL where the malicious binary was located. Or the security application blocks the malicious binary whilst it was being downloaded to the desktop.
- **The test case is marked as "Behaviour Blocked"** if the security application blocks the malicious binary when it is executed and either automatically blocks it or postpones its execution and warns the user that the file is malicious and awaiting user input.
- **The test case is marked as "Detected"** if the security application detects the threat and sends an alert to the central console or notifies the user, but the sample is allowed to run.
- **The test case is marked as "Blocked in 24h"** if the security application fails to block or behaviour block the malicious sample but blocks it during the retest.
- **The test case is marked as "Missed"** if the security application fails to block or behaviour block the malicious sample during both tests.

9. Tests are conducted with all systems having internet access.
10. As no user-initiated scans is involved in this test, applications rely on various technologies to detect, block and remediate threats. Some of these technologies are URL blacklisting, reputation, signature, machine learning, heuristics, behaviour etc.

False positive test methodology

1. Windows 10 Enterprise 64-bit operating system is installed on a hardened virtual machine, all updates are applied, and third-party applications installed and updated.
2. An image of the operating system is created.
3. A clone of the imaged systems is made for each of the security applications used in the test.
4. An individual security application is installed using default settings on each of the systems created in (3) and then, where applicable, updated. If the vendor provided a non-default setting, this setting is checked whether it is realistic. If yes, the changes are documented, applied, and added to the appendix section of the report.
5. A clone of the system as at the end of (4) is created.
6. Introducing the binary executables (or documents, scripts, etc.) to the clean, unprotected system via disk image or network share. If the sample works, the sample is saved to a different disk image or network share.

False Positive test is conducted by the following procedure.

6.1. Scanning the binary executables (or documents, scripts, etc.) on the disk image or on the network share.

6.2. Executing the test samples.

6.3. The sample is retested 24 hours after the initial test if the security application failed to permit the harmless file.

- **The test case is marked as "False block"** if the security application falsely identifies and blocks the binary at any stage during the test and retest.
- **The test case is marked as "Detected"** if the security application falsely identifies and the binary at any stage during the test and retest but allows it to run.
- **The test case is marked as "Allowed to run in 24h"** if the security application falsely identifies and blocks the binary at any stage during the test but allows it to run upon the retest.
- **The test case is marked as "Allowed to run"** if the security application correctly identifies the binary as harmless and allows it to run.

7. Tests are conducted with all systems having internet access.

Exploit / Fileless test methodology

1. Windows 10 Enterprise 64-bit operating system is installed on a hardened virtual machine, all updates are applied, and third-party applications installed and updated.
2. An image of the operating system is created.
3. A clone of the imaged systems is made for each of the security applications used in the test.
4. An individual security application is installed using default settings on each of the systems created in (3) and then, where applicable, updated. If the vendor provided a non-default setting, this setting is checked whether it is realistic. If yes, the changes are documented, applied, and added in the report in an appendix.

5. A clone of the system as at the end of (4) is created.

Exploit / Fileless test is conducted by the following procedure.

6. Our payloads use an exploit for the one of an installed vulnerable application. In order to simulate a realistic attack scenario, a payload is constructed to include at least one of the common CnC frameworks.
7. The opening stage of the exploit is introduced to the system and we monitor if the vulnerable application starts the initial stage payload, the exploit is being executed and if a session is established to our CnC server.
8. After navigating to the exploit site, the system is supervised if there are any new processes, loaded DLLs or CnC traffic emerge. If the exploitation is successful, the following actions are executed.
   8.1. Upload a file to the victim.
   8.2. Download a file from the victim.
   8.3. Create a process remotely.
   8.4. Read the contents of a file on the victim.
9. When user interaction is needed from the endpoint protection (e.g. site visit not recommended, etc.) the default action is chosen. When user interaction is needed from the operating system, we chose the run/allow options.
10. Throughout the test, the Process Monitor from the Sysinternals Suite and Wireshark are running (both installed to non-default directories and modified not to be detected by default anti-debugging tools).

- **The test case is marked as "Signature Block"** if the security application blocks the URL (infected URL, exploit kit URL, redirection URL, malware URL) by the URL database (local or cloud).
- **The test case is marked as "Blocked"** if the security application blocks the page containing a malicious HTML code, JavaScript (redirects, iframes, obfuscated JavaScript, etc.) or Flash files. Or if the security application blocks the downloaded payload by analysing the malware before it can be started. (reputation-based block or heuristic based block).
- **The test case is marked as "Behaviour Blocked"** if the security application blocks the downloaded payload after it has been started.
- **The test case is marked as "Detected"** if the security application detects the threat and sends an alert to the central console or notifies the user, but the attack is allowed to run.
- **The test case is marked as "Missed"** if the security application fails to detect, block or behaviour block the attack and the it can be carried out.

11. Tests are conducted with all systems having internet access.
12. As no user-initiated scans is involved in this test, applications rely on various technologies to detect, block and remediate threats. Some of these technologies are URL blacklisting, reputation, signature, machine learning, heuristics, behaviour etc.

Performance test methodology

1. Windows 10 Enterprise 64-bit operating system is installed on a physical machine, all updates are applied, and third-party applications installed and updated.
2. A backup image of the operating system is created.
3. The security application is installed, with the same configuration it is used in the other tests.
4. The following performance metrics are measured:
    - Operating system boot time
    - Size of the files installed and created by the security application. The size is measured at least one week after the installation, after virus definition updates, scans, and time passed with normal computer usage.
    - Copy time of files
    - Archive operation time
    - Opening time for (clean) files in Office applications
    - Downloading files through browser
    - Website loading time in browser. The browser should fully load a popular, complex website, from a local network URL or replay proxy to eliminate network latency.
    - AV product update time
    - System disk scan time

Every performance result is a calculated average of at least three measurements.

Real botnet test methodology

1. Windows 10 Enterprise 64-bit operating system is installed on a hardened virtual machine, all updates are applied, and third-party applications installed and updated.
2. An image of the operating system is created.
3. A Real botnet dropper is run on the clean, unprotected system, thus simulating a pre-infected state.
4. A clone of the imaged system is made for each of the security applications to be used in the test.
5. An individual security application is installed using default settings on each of the systems created in (4) and then, where applicable, updated. If the vendor provided a non-default setting, this setting is checked whether it is realistic. If yes, the changes are documented, applied, and added in the report in an appendix.
6. A clone of the system as at the end of (5) is created.

Real botnet test is conducted by the following procedure.

6.1. Starting a new instance of Firefox (or the Safe Browser) and navigating to a financial website. Where the security application offers a secured or dedicated banking browser, this is used. If the security application is designed to protect Internet Explorer, only that component is tested.

6.2. Text is entered into the Account login page of the financial website using the keyboard or using a virtual keyboard if the application under test provides such functionality, and

then the “log in” button is pressed.

- **The test case is marked as passed – a green checkmark** if the security application detects the financial malware when the security application is installed, and a mandatory scan is made. Or the security application detects the real financial malware when it is executed according to the following criteria:
  - It identifies the real financial malware as being malicious and either automatically blocks it or postpones its execution, warns the user that the file is malicious and awaits user input.
  - It identifies the real financial malware as suspicious or unknown and gives the option to run in a sandbox or safe restricted mode, which prevents the real financial malware from capturing and sending the logon data to the MRG CnC, whilst giving no alerts or giving informational alerts only. Or The security application intercepts the action of the real financial malware and displays warnings and user action input requests that are clearly different from those displayed in response to legitimate applications.
- **The test case is marked as missed – a red cross** if the security application fails to detect the real financial malware according to the following criteria:
  - The security application fails to prevent the real financial malware from capturing and sending the logon data to the MRG CnC and gives no alert or provides informational alerts only.
  - The security application intercepts the action of the real financial malware but displays warnings and user action input requests that are indistinguishable in meaning from those displayed in response to legitimate applications.
  - The security application identifies the malware and gives the option to run in a sandbox or safe restricted mode which fails to prevent the real financial malware from capturing and sending the logon data to the MRG CnC and gives no alert or provides informational alerts only.

7. Testing is conducted with all systems having internet access.

Financial malware simulator test methodology

1. Windows 10 Enterprise 64-bit operating system is installed on a hardened virtual machine, all updates are applied, and third-party applications installed and updated.
2. An image of the operating system is created.
3. A clone of the imaged systems is made for each of the security applications used in the test.
4. An individual security application is installed using default settings on each of the systems created in (3) and then, where applicable, updated. If the vendor provided a non-default setting, this setting is checked whether it is realistic. If yes, the changes are documented, applied, and added to the appendix section of the report.
5. A clone of the system as at the end of (4) is created.

Financial malware simulator test is conducted by the following procedure.

6. Where the security application offers a secured or dedicated banking browser, this is used. If the security application is designed to protect IE, only that component is tested.

   6.1. The simulator specific process is started.

   - **The test case is marked as passed – a green checkmark** if the security application identifies the simulator as being malicious and either automatically blocks it or postpones its execution, warns the user that the file is malicious and awaits user input. Or, it identifies the simulator as suspicious or unknown and gives the option to run in a sandbox or safe restricted mode which does not allow the hooking/redirection, or even with successful hooking, the personal data cannot be captured from the browser.
   - **The test case is marked as missed – a red cross** if the security application fails to identify the simulator based on the following criteria:
     - The security application allows the hooking/redirection of the event, and the personal data can be captured from the browser. Or, it fails to prevent the simulator from injecting itself into the browser process and gives no alert or provides informational alerts only.
     - The security application identifies the simulator as malware or unknown and gives the option to run in a sandbox or safe restricted mode which fails to prevent the simulator from injecting itself into the browser process and gives no alert or provides informational alerts only. Or, the security application allows the hooking/redirection of the event, and the personal data can be captured from the browser.

7. Testing is conducted with all systems having internet access.

4. Participation

AMTSO's goal with having Participants is that in exchange for cooperating (engaging with Testers and following disclosure requirements), Participants have additional rights to audit their configuration and provide commentary on Test results. There must be no additional cost to a Test Subject Vendor to be a Participant. If a Tester charges to participate in a Public Test or any related services, and a Test Subject Vendor chooses to not pay the fee, that Vendor must be able to choose to be a Participant and follow this AMTSO standard.

Opt-Out Policy: Vendors can opt out if Vendor can prove that the test system or the Product was misconfigured in a way which greatly changes the test results. E.g. important modules were unnecessarily turned off compared to default configuration, or during the test the system could never reach the cloud.

Conflict of Interest Disclosure: There is no known conflict of interest.

Funding: Funding of this project is achieved by vendors subscribing to participate in this project, this gives them more in-depth information on how their product(s) performed and, if there are any issues discovered in the product during testing, our technical team provides all the data necessary to help improve the product.

Part of the funding comes from directly licensing reports so they can be used for marketing purposes.

Vendors often silently enter testing, sometimes of their newly developed product or a product in BETA or Pre-Release phase.

Finally, part of our funding comes from third parties, they commission us to include certain product into testing, both public and private.

5. Environment

Test hardware and configuration details follow.

Virtual machine Configuration: Our Virtual Machine hardware specification calls for 4GB RAM and a dual core processor. AES includes Adobe Flash, Reader, Java, Microsoft Office 2010 or newer, Edge, Chrome & VLC Player. During installation of the security application, if an option to detect PUAs was given, it was selected.

Physical machine specification: OS: Windows 10 x64 Enterprise
CPU: Intel Core i5
Memory: 8GB
Storage: 100GB SSD
Networking: Intel Gigabit Network Connection / 802.11n Wi-Fi

Sample Relevance: In the Wild 360 / Full Spectrum Test, majority of the malicious URLs used in this test were compromised legitimate websites which served malware. We believe that such URLs pose the greatest danger to users as this is the place where they least expect to get infected. Some of the URLs pose as fake porn websites serving visitors with various types of malware. The remaining of the URLs come from our regular honeypots or, in case of ransomware and financial malware in particular, we used URLs from newly discovered distribution sites.

Geographic Limitations: There are no geographic limitations in terms of samples.

Curation Process: Voluntary Participants are given equal opportunities to participate in such Curation and feedback processes for all their respective Test Subjects.

Malware delivered by URLs used in this test can be considered as Zero Day in the true meaning of that phrase. It is our opinion that Ransomware currently poses the greatest threat to users, for this reason we choose to use more URLs serving this threat than before. Because of the wide spectrum of malware used in this project and the freshness of the samples, we used a smaller set than usual.

Applications that didn't protect the system from file encrypting ransomware cannot be certified because they could not remediate the threat as files usually cannot be decrypted. Our testing environment supports the use of VM aware malware, this is the reason why we were able to use more sophisticated threats which wouldn't run on Virtual Machines.

Distribution of Test Data: We send all failed samples to all participants, along with detailed test logs.

6. Schedule

Start Date Range: The test commencement date is January 20, 2021.

Test Duration and Calculated End Date: The test is expected to require approximately fourteen weeks (without installation of products) and is forecast to conclude on April 23, 2021.

Milestones: Delivery milestones appear in the following chart.

MRG-Effitas 360 Degree Assessment and Certification Test Project Schedule Milestones

Index	Test Activity	Start Date Range	Dependencies
1	*Test Commencement*	*January 25, 2021*	
2	*Confirm Vendor Configuration Feedback*	*January 29, 2021*	
3	*Milestone 1 – Preliminary Results*	*March 04, 2021*	*(1), (2)*
4	*Milestone 2 – Test Report First Edition – End of Testing Period*	*March 11, 2021*	*(3)*
5	*Feedback and Dispute Resolution Time – Retests as Needed*	*March 22, 2021 - April 02, 2021*	*(4)*
6	*Milestone 3 – Issue Final Report – End Date for Test*	*April 23, 2021*	*(5)*

Communications: Whenever there are significant deviations from this schedule (more than 5 days), we will notify all affected vendors within 3 business days.

Risks and Risk Management: We could not identify any risks with the test.

7. Control Procedures

Connectivity Validation: Tests are conducted with all systems having internet access. Each individual test for each security application is conducted from a unique IP address. All security applications are fully functional registered versions Each vendor can supply a utility or another in-product feature to validate proper cloud-connectivity functionality.

Logging: In the first initial email to the vendors, we will ask if they require any special logging during the test. We expect these methods and tools to be working, otherwise we cannot guarantee the result of these logs.

Updates: Each individual security application will be installed using default settings on each of the test systems defined in the Environment section of this Test Plans and then, where applicable, updated.

8. Dependencies

Participant Actions: As this test scope are consumer Test Subjects, we do not require any Vendor specific actions.

9. Scoring Process

In order to attain a quarterly MRG Effitas 360 Degree Level 1 certification award, a security application must entirely protect the system from initial infection (autoblock or behaviour protection) and a product must pass the Botnet test during the quarter.

Level 2 certification is given if the application blocks or detects any initially missed malware in at least 98% of all cases on the 24-hour retest, while the initially missed test cases are less than 10%. If a ransomware/wiper successfully runs and the files are not available anymore, Level 2 certification is not available. Applications that meet this specification will be given the level 2 certification for the quarter. PUA/adware, exploit/fileless, false positive, performance and financial malware simulator tests are not part of the level 2 certification.

In order to attain a quarterly MRG Effitas 360 Exploit Degree certification award, a security application must entirely protect the system from initial infection (autoblock or signature block or detected or behaviour protection).

In order to attain a quarterly MRG Effitas 360 Online Banking Degree certification award, a security application must entirely protect the system from initial financial malware In-the-wild infection (autoblock or behaviour protection) and a product must pass the Botnet and financial malware simulator tests during the quarter.

10. Dispute Process

After the vendors receive the test results and the logs, Vendors can dispute individual test cases if they can prove that their disagreement is justified.

11. Attestations

I understand and agree that I am submitting this Test Plan, and the following Attestations, on behalf of the entity listed below, and I represent and warrant that I have authority to bind such entity to these Attestations. All references to “I” or “me” or similar language refer to such entity. I represent and warrant that the following Attestations are true, to the best of my knowledge and belief, and each of the following commitments will be upheld to the best of my ability.

1. I will provide public notification on the AMTSO website covering my obligation for notification of a Public Test, regardless of whether a potential Participant is in actual receipt of such notification prior to the Commencement Date of a Test.
2. All products included in this Test will be analysed fairly and equally.
3. I will disclose any anticipated or known imbalance or inequity in the Test design to all Participants in the Test.
4. Although I may charge for participation in a Test, I will not charge any additional fees for a vendor to be a test subject under the Standards.
5. I will disclose any material conflicts of interest or other information that could materially impact the reliability of the Test.

6. I will disclose how the Test was funded.

I hereby affirm, to the best of my knowledge and belief that this Test Plan complies with the AMTSO Testing Standards, as of the date hereof.

Signature: /s/ Lorand Lajsz

Name: Lorand Lajsz

Test Lab: MRG Effitas

AMTSO Test ID: [AMTSO-LS1-TP035]