MRG Effitas Online Banking / Browser Security Certification Project – Q3 2014

Contents

Introduction

MRG Effitas has published an Online Banking Browser Security report every year for the last four years. Since 2013, that single report has been replaced by quarterly assessments. This report is the assessment for Q2 2014.

Whilst this report sits in much the same space as our previous reports, it employs a range of much more sophisticated assessments that result in an extremely accurate level of efficacy assessments, so much so that we now award quarterly certifications to products that meet specific assessment criteria.

MRG Effitas provides two levels of testing: Level 1, where we simply test a vendor's product and provide a report for that quarter's assessment, and Level 2 (which includes Level 1), where we liaise with the vendors during testing and alert them to any issues found with their technology and provide all engineering and technical support required for them to counter these issues. The purpose of Level 2 participation is that it serves as an external QA service for vendors, helping them improve the efficacy of their product. Level I and II reports are published separately.

This is a Level 1 report.

Executive summary

This Certification Programme can also serve as educational material for average users as it raises awareness about financial malware and all the dangers that face users when they do online banking, using online payment services such as PayPal or just using any other form of online shopping.

It should be noted that financial malware earned its name because, in most cases, it attempts to grab the user name and password from places which are used for online transactions. Another thing financial malware can do is steal login credentials from popular social Networking websites such as Facebook, Twitter, LinkedIn etc.

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 financial malware works, how it 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.

We tested a group of internet security suits and anti-financial fraud applications. With internet security suits 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 popup alerts or questions.

Out of all the products we tested, only four managed to pass all three stages of the test. Out of those four applications, two are dedicated anti-financial fraud products, these being **Quarri POQ** and **Wontok SafeCentral**, the other two were internet security suites. These Internet Security Suits, **Kaspersky Internet Security** and **Webroot SecureAnywhere** actually harden the existing users browser with their *SafeMoney (Kaspersky) and IdentityShield (Webroot)* technologies.

It is our belief that users of internet security suites would find a product that hardened their existing browser would be less intrusive on their daily activities than one that employed a separate proprietary browser that may appear alien to their usual experience and limit them from conducting other work whilst said browser was active.

Certification

In order to attain MRG Online Banking / Browser Security Certification, a product must pass every test during the quarter. Applications that meet this specification will be given certification for that quarter.

MRG Effitas Online Banking Browser Security Certification for Q3 2014 is awarded to the following products:

- **Kaspersky Internet Security**
- **Quarri POQ**
- **Webroot SecureAnywhere**
- **Wontok SafeCentral**

The purpose of this report

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 about measuring product ability to detect malware. In the threat landscape of today and that of the foreseeable future, detection, although important, should not be the primary metric.

Malware, for some time now, has been engineered for one primary purpose: to generate revenue for cyber criminals. Global cybercrime is set to generate higher revenues than almost any other crime by the end of 2014 and is regarded by some national governments as a bigger threat than nuclear war.[i]

In continuing to generate revenues, cybercriminals have two primary objectives:

1. To ensure the crimeware they use is exceptionally stealthy, so as to evade detection of security products and thereby enable it to reside on the victim's device, performing its function for as long as possible;

2. To capture as much confidential and valuable information on the user and the enterprise as possible. Commonly, the most valuable data harvested from users will be logon credentials or passwords entered into browsers during online banking sessions or other online ecommerce activities, and the most valuable data harvested from companies is intellectual property, business plans and customer information.

It is well evidenced that crimeware is particularly difficult to detect. Once it has infected a system, it is unlikely to be detected for some time, possibly days or weeks in some cases. In such instances, the victim system is exposed to the threat of data exfiltration by the crimeware. Remediation or detection after exfiltration has occurred commonly has low value as the victim's private banking credentials are likely to have been stolen, although sometimes it can prevent further damage.

In 2010 MRG Effitas began reverse engineering financial Malware to create simulators that employ the same "Man in the Browser" attacks as the in the wild code, and so were for the first time able to determine whether secure browsers were capable of preventing data exfiltration. This was so revolutionary that in 2012 the BBC based a TV programme on our work – BBC Click, "The Man in the Browser" - http://www.youtube.com/watch?v=DUnZMwXCkyw

Why do we use simulators? We have been asked this question countless times in the past and we always answer such questions with the following:

Simulators are used in every industry and sector, including aerospace, automotive, law enforcement, the military and finance. Nobody questions the validity of using simulators in these sectors as it is a well-known fact that simulators improve performance.

There are two major types of simulators, one that is used to teach students (e.g. pilots) and the other to simulate various types of attacks (e.g. military). This is exactly why MRG Effitas decided to start creating simulators. By developing test tools we try to simulate attacks that may not be as prevalent at present but may become more so in the future (which can be just around the corner). Simulators can point out potential weaknesses in products and even use new types of attacks that can be useful for developers as they can learn about these from a Testing Lab rather than from their users when an attack of this type occurs in the wild.

All the attack methods implemented by our simulators are valid and could be used or are being used by certain types of less prevalent malware. It should be noted that high prevalence results if a known type of malware is used in large scale attacks. However, as highlighted before, some malware attacks cannot be used in large scale attacks but the outcome can be even more lucrative than with the highly prevalent ones.

Although employing these reverse engineering techniques to create simulators was revolutionary, MRG Effitas never stands still and always continues to innovate. As of Q2 2014, the Online Banking / Browser Security Certification Programme includes the use of real, fully operational botnets such as ZeuS, Citadel, SpyEye etc., as part of efficacy assessment.

For the certification programme, MRG Effitas has chosen IBM as its technology partner. The use of IBM's unique SoftLayer cloud computing technology has enabled MRG Effitas to create complete botnets which exactly model those in the wild, whilst ensuring they are contained in secure lab conditions and pose no threat to the public.

See https://www.youtube.com/watch?v=n_sd-RKnkrQ&feature=youtu.be

Cat and Mouse Game: It is no secret that when it comes to financial malware, vendors have a lot of work on their hands. Bad guys use various techniques to evade detection and even build special modules in "Builders" to disable certain applications. Luckily, so far developers have been able to respond to these "enhancements" swiftly.

Nowadays, most of the financial malware is based on the leaked source code of the most notorious Banking Trojan ZeuS. Another source code that is publicly available is the Carberp code. It should be noted that Carberp is an even more advanced and more sophisticated piece of financial malware than ZeuS.

While these are the best known and most prevalent pieces of financial malware, we notice a rise in new and more sophisticated financial malware that may not be targeting users globally, but is more regional and created to target specific groups of users, organizations or banks.

For all that we mention in this report, it is imperative for us to spread awareness about these threats among all user levels. Browser security software is becoming more necessary than ever before and this is not limited to Windows users only, given that financial malware is cross-platform and attacks mobile phone users too. It is very important to mention this because more and more advertisements encourage users to use their mobile phones for payments and even use phones as credit cards. We strongly believe that this should not be done without the awareness about possible dangers in conducting such transactions.

Products that pass all tests during a quarter will receive the MRG Effitas certification for secure online banking.

In providing these quarterly certifications, the MRG Effitas Online Banking / Browser Security 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 current financial malware attacks.

Tests employed

In this assessment (Q3 2014) we ran the following tests:

In the Wild Real Financial Malware Test

In total, 406 live ITW samples were used. The tests were performed using financial malware only, including, *inter alia*, the following: ZeuS (ZeuS P2P, Ice IX, KINS, Power ZeuS, Ramnit, Licat (Murofet)), Citadel, SpyEye and Zberp, Dyre.

Botnet Test

MRG Effitas is proud to present the world's first real, public botnet test. In this test, we acquired leaked builders from real financial malware (ZeuS, Citadel, SpyEye etc), created the droppers and configured the C&C servers in the safe SoftLayer environment. Because this test uses real financial malware, where data exfiltration can be tested as it happens in the wild, the test efficiently maps the real-world threats users face today. These builders and droppers are available to everyone for free, thus the threats provide an entry level for criminals and are common threats in the wild.

API Hooking Simulator Test

Financial malware developers always find new ways to bypass current protection technologies. However, the traditional way is to do so via the API hooking technique. This technique is a two-step process. First, the malware injects itself into the browser process, then hooks (redirects) the API calls, where the password can be found in a buffer passed to the function as a parameter. In this test, we used 4 different methods for the process injection phase. After a successful attack, the attacker can either extract passwords, session cookies, credit card/CVV numbers from the web sessions, or inject html forms into the web sessions (e.g. credit card number and CVC/CVV code), because the SSL encryption takes place after the API calls. The purpose of testing with simulators is that the simulator is unknown to the security solution and thus it won't detect the simulator using traditional AV

methods, which are known to be bypassed easily. This test measures the protection capabilities against zero day threats.

Security Applications Tested

- Avast Internet Security - 2014.9.0.2018
- AVG Internet Security - 2014.0.4714
- Avira Internet Security - 14.0.4.672
- BitDefender Internet Security - 7.55517
- ESET Smart Security - 7.0.317.4
- F-Secure Internet Security - 2.0.6 303
- G Data Internet Security - 25.0.1.2
- Kaspersky Internet Security with Safe Money - 15.0.0.463 (a)
- McAfee Internet Security - 16.8.708
- Microsoft Security Essentials - 4.5.216.0
- Norton Internet Security - 20.5.0.28
- Quarri POQ - 4.2.0.2517
- Sophos Endpoint Security and Control - 10.3
- ThreatTrack Vipre Internet Security - 7.0.6.2
- Trend Micro Titanium Internet Security - 7.0.12.55
- Webroot SecureAnywhere - 8.0.4.84
- Wontok SafeCentral - 3.1.21.3897

Samples used in the In-The-Wild real financial malware test

Sample selection is of fundamental importance to this and all similar tests. In the case of the Online Banking / Browser Security Certification – In the Wild Test, all samples used are "live" and "in the wild", by which we mean they are residing at the URLs selected or created by the cybercriminals and they are not from a time lagged ITW list. As these are live ITW samples, they represent current zero day threats that can present an issue with sample verification. There is no effective and reliable way to verify samples before testing that does not introduce possible artificial sample submission or delay, so all verification is conducted after testing. Tests performed using samples that are later proven to be invalid are excluded from the results. The type of samples used is decided by MRG Effitas on the basis of a mixture of criteria, cantering about key relevancies:

1. Prevalence – they are widespread and so represent the most common threats.
2. Growth – they may be few now, but our research shows they are rapidly expanding.
3. Innovation – they employ innovative techniques to counter security measures.
4. It is malware having financial motives, by either stealing login credentials, initiating transactions, or doing web injects.

In total, 406 live ITW samples were used. The tests were conducted using financial malware only, including, *inter alia*, the following:

- **SpyEye**
- **Citadel**
- **Zberp**
- **Dyre**

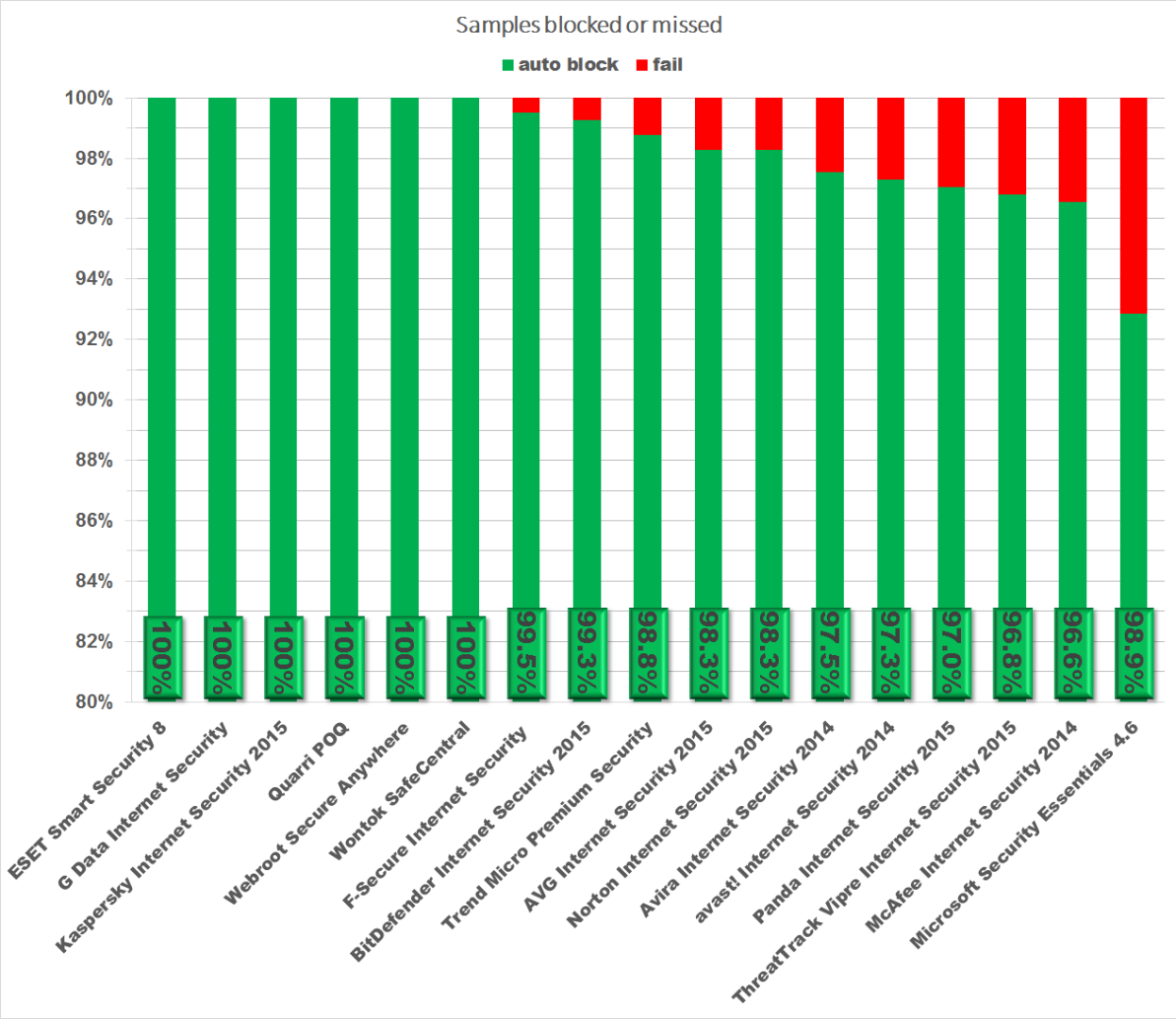
- **ZeuS** clones like
  - ZeuS P2P
  - Ice IX
  - KINS
  - Power ZeuS
  - Ramnit
  - Licat (Murofet)

Test Results

The tables below show the results of testing in the Online Banking / Browser Security Certification Programme.

Q3 2014 In the Wild real financial malware test results

The table below shows the results of testing using In-The-Wild real financial malware.



Q3 2014 Botnet tests results

The table below shows the results of testing using real financial malware.

Product	Zeus	Citadel	SpyEye
avast! Internet Security 2014 with Safe browser only	✓	✓	✓
avast! Internet Security 2014 without Safe browser	✓	✓	✓
AVG Internet Security 2015	✓	✓	✗
Avira Internet Security 2014	✓	✓	✗
BitDefender Internet Security 2015 with Safepay only	✓	✓	✓
BitDefender Internet Security 2015 without Safepay	✓	✓	✗
ESET Smart Security 8	✓	✓	✓
F-Secure Internet Security with banking protection only	✓	✗	✗
F-Secure Internet Security without banking protection	✓	✓	✗
G Data Internet Security	✓	✓	✓
Kaspersky Internet Security 2015 with safe browser only	✓	✓	✓
Kaspersky Internet Security 2015 without safe browser	✓	✓	✓
McAfee Internet Security 2014	✓	✓	✗
Microsoft Security Essentials 4.6	✓	✓	✓
Norton Security 2015	✓	✓	✗
Panda Internet Security 2015	✓	✓	✗
Quarri POQ	✓	✓	✓
ThreatTrack Vipre Internet Security 2015	✓	✓	✗
Trend Micro Premium Security	✓	✓	✓
Webroot Secure Anywhere	✓	✓	✓
Wontok SafeCentral	✓	✓	✓

✓	The application prevented the malware from capturing login data within the same session.
✗	The application failed to prevent the malware from capturing login data within the same session.

During the tests we witnessed many problems with endpoint protection systems. Following is a non-exhaustive list of problems:

- Inconsistent behaviour/block: Some vendors failed to protect the user in the first test, but protected the user after the first test. During the first test, the protected browser usually crashed and was restarted automatically. If the user was protected 4 times from 5 attempts, we marked these as transient failures and the products were marked as having passed.
- Missing alert: Some vendors detected the threat during the security product installation, but failed to warn the user about the detected and removed threat. However, the detailed AV log revealed the threat detection and removal.
- Missing log and alert: Some vendors detected the threat during the security product installation, but failed to warn the user about the detected and removed threat, and even the detailed AV log was empty.
- Some vendors would have failed the test without the mandatory restart in the test methodology. These vendors had not suggested or enforced any restart after product installation or threat removal.
- Some vendors detected the threat and removed the malware from the file system, but the threat was not removed from the memory. After threat removal, the security product did not suggest any restart to the user. This was marked as a fail, as users tend to use the OS without restarting for weeks.
- Some safe browsers are using browser types that are not targeted by financial malware. As a result, even if the malware was running in the background and without any active protection, these browsers passed the test.
- A vendor detected the Citadel malware as SpyEye.
- A vendor detected all three malware samples, and gave the option to block the threat. Still, it did not prevent the malware from stealing login credentials.

Q3 2014 API hooking simulator test results

The table below shows the results of testing using the API hooking malware simulator.

Product	Api hooking 1	Api hooking 2	Api hooking 3	Api hooking 4
avast! Internet Security 2014 with Safe browser only	✓	✓	✓	✓
AVG Internet Security 2015	✗	✗	✗	✗
Avira Internet Security 2014	✗	✗	✗	✗
BitDefender Internet Security 2015 with Safepay only	✓	✓	✓	✓
ESET Smart Security 8	✗	✗	✗	✗
F-Secure Internet Security with banking protection only	✗	✗	✗	✗
G Data Internet Security	!	!	!	!
Kaspersky Internet Security 2015 with safe browser only	✓	✓	✓	✓
McAfee Internet Security 2014	✗	✗	✗	✗
Microsoft Security Essentials 4.6	✗	✗	✗	✗
Norton Internet Security 2015	✗	✗	✗	✗
Panda Internet Security 2015	✗	✗	✗	✗
Quarri POQ	✓	✓	✓	✓
ThreatTrack Vipre Internet Security 2015	✗	✗	✗	✗
Trend Micro Premium Security	✗	✓	✗	✗
Webroot Secure Anywhere	✓	✓	✓	✓
Wontok SafeCentral	✓	✓	✓	✓

✓	The application blocked the simulator
!	The application alerted but was unable to block the simulator
✗	The application failed to block the simulator

Detailed Description of the Tests

In The Wild real financial malware test

For detailed description of the In the Wild financial malware test, please read the methodology.

Botnet test

Builders and webserver components of the financial malware ZeuS, Citadel and SpyEye etc have been leaked in previous years. We used these leaked builders to build our in-house C&C malware network. The C&C servers are operated at the cloud provider SoftLayer in a safe environment, thus the whole infrastructure is as close to real financial malware as possible, simulating attackers either buying resources at cloud providers or hacking legitimate websites and placing the C&C server there.

By operating the C&C server in our environment, we could determine with 100% certainty whether data exfiltration had really occurred or not. The builders and droppers were not modified/obfuscated/encrypted in any way other than by default in the builder.

API hooking simulator test

The API hooking technique is a two-step process. First, the malware injects itself into the browser process, then hooks (redirects) the API calls which can contain the passwords as function parameters. The four different methods we use for process injection are: Context Switching, Internal Debugger, CreateRemoteThread, CreateRemoteThread Extended. We hooked the HttpSendRequestA/W API calls to capture the login credentials.

The CreateRemoteThread and CreateRemoteThread Extended methods use standard Windows functions to allocate memory in the target process, maps a DLL to the remote process, and finally the (malicious) functions in the DLL are executed because DLL_PROCESS_ATTACH is triggered.

The Context Switch method uses standard Windows functions to allocate memory in the target process, find a running remote thread to hijack in the target process. It saves the current EIP and sets it to the address of the LoadLibrary function, writes the function and parameters (injected DLL name) in the remote process, the hijacked thread executes the LoadLibrary call, and finally the (malicious) functions in the DLL are executed because DLL_PROCESS_ATTACH is triggered.

After a successful attack, the attacker can either extract passwords, session cookies, credit card/CVV numbers from the web sessions, or inject html forms into the web sessions (e.g. credit card number and CVC/CVV code), because the SSL encryption takes place after the API calls.

Appendix 1

Methodology Used in the Q2 2014 Online Banking Certification - In the Wild Test

1. Windows 7 Ultimate Service Pack 1 64 bit operating system is installed on a virtual machine, all updates are applied and third party applications installed and updated according to our "Average Endpoint Specification".
2. An image of the operating system is created.
3. A clone of the imaged systems is made for each of the security applications to be used in the test.
4. An individual security application is installed using default settings on each of the systems created in 5 and then, where applicable, it is updated and shut down. If the installer has the option to participate in cloud protection, or PUA protection, all of these are enabled.
5. Testing is conducted by:
    a. Downloading the sample using Internet Explorer to the desktop, closing Internet Explorer, conducting a context menu scan or, where unavailable, a system scan, and then executing the sample.
6. A test is deemed to have been passed based on the following criteria:
    a. The security application blocks the URL where the sample is located, thus preventing its download.
    b. The security application detects the sample whilst it is being downloaded to the desktop.
    c. The security application detects the sample during the context or system scan.
    d. The security application detects the sample when it is executed according to the following criteria:
        i. It identifies the sample as being malicious and either automatically blocks it or pauses its execution, advises the user not to execute it and awaits user input.
7. A test is deemed to have been failed based on the following criterion:
    a. The security application fails to detect the sample under conditions 6a, 6b, 6c or 6d.
8. Testing is conducted with all systems having internet access.
9. Each individual test for each security application is performed from a unique IP address. All security applications are fully-functional unregistered versions or versions registered anonymously, with no connection to MRG Effitas.

Methodology Used in the Q3 2014 Online Banking Certification – Real Botnet Test

1. Windows 7 Ultimate Service Pack 1 64 bit operating system is installed on a virtual machine, all updates are applied and third party applications installed and updated according to our "Average Endpoint Specification".
2. An image of the operating system is created.
3. Real ZeuS, Citadel and SpyEye ect droppers are installed onto clean systems without protection, thus simulating a pre-infected state.
4. A clone of the imaged systems 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 5 and then, where applicable, it is updated and shut down. If the installer has the option to participate in cloud protection, or PUA protection, all of these are enabled.
6. A clone of the system as it is at the end of 5 is created, and the system is started.
7. Each real financial malware test is conducted by:
    a. Starting a new instance of Internet Explorer (or the Safe Browser) and navigating to https://www.paypal.com/en/cgi-bin/webscr?cmd=_login-submit. 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 will be tested.

   b. Text is entered into the Account login page of https://www.paypal.com/en/cgi-bin/webscr?cmd=_login-submit using the keyboard, or using a virtual keyboard if the application under test provides such functionality, and then the "log in" button is pressed.

8. A test is deemed to have been passed (marked as a green checkbox) based on the following criteria:
   a. The security application detects the real financial malware when the security application is installed, and a mandatory scan is made.
   b. The security application detects the real financial malware when it is executed according to the following criteria:
      i. 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.
      ii. It identifies the real financial malware as suspicious or unknown and gives the option to run in a sandbox or safe restricted mode, and, when run in this mode, it meets the criterion c or d below.
   c. The security application prevents the real financial malware from capturing and sending the logon data to the MRG results page, whilst giving no alerts or giving informational alerts only.
   d. 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, when they are executed or installed on that system.
9. A test is deemed to have been failed (marked as a red cross) based on the following criteria:
   a. The security application fails to detect the real financial malware after restart and then:
      i. The security application fails to prevent the real financial malware from capturing and sending the logon data to the MRG results page location (malware C&C server), and gives no alert or provides informational alerts only.
      ii. 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, when they are executed or installed on that system.
   b. The security application identifies the malware as real financial malware or unknown malware? and gives the option to run in a sandbox or safe restricted mode, and, when run in this mode, it:
      i. Fails to prevent the real financial malware from capturing and sending the logon data to the MRG results page or local store, and gives no alert or provides informational alerts only.
      ii. Displays warnings and user action input requests that are indistinguishable in meaning from those displayed in response to legitimate applications, when they are executed or installed on that system.
10. Testing is conducted with all systems having internet access.
11. Each individual test for each security application is conducted from a unique IP address.
12. All security applications are fully-functional unregistered versions or versions registered anonymously, with no connection to MRG Effitas.

Because we did not use 0-day malware in this test, but 1-2 years old or even older malware versions, when a security application provided both traditional AV engines and safe browser solutions, the security application was tested in two modes. In the first mode, all protections were turned on and the safe browser was used. In the second mode, the AV engine was turned off by either putting the malware on the exclusion list or disabling the AV engine itself. The safe browser was still used for this test. Thus, the second test simulates a malware which

bypasses traditional AV engines, and only the safe browser is there as a last line of defence to protect the banking session.

Methodology Used in the Q3 2014 Online Banking Certification – API Hooking Simulator Test

1. Service Pack 1 64 bit operating system is installed on a virtual machine, all updates are applied and third Windows 7 Ultimate party applications installed and updated according to our "Average Endpoint Specification".
2. An image of the operating system is created.
3. A clone of the imaged systems is made for each of the security applications to be 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, it is updated. If restart is recommended by the application (visible to the user), the system is restarted. If the installer has the option to participate in cloud protection, or PUA protection, all of these are enabled.
5. A clone of the system as it is at the end of 4 is created, and the system is started.
6. The API hooking simulator is started onto the clean systems with protection installed.
7. Each API hooking simulator test is conducted by:
    a. Starting a new instance of Internet Explorer (or the safe browser) and navigating to https://www.paypal.com/en/cgi-bin/webscr?cmd=_login-submit. 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 going to be tested.
    b. Trying to inject the simulator into the browser process.
    c. Text is entered into the Account login page of https://www.paypal.com/en/cgi-bin/webscr?cmd=_login-submit using the keyboard, or using a virtual keyboard if the application under test provides such functionality, and then the "log in" button is pressed.
8. A test is deemed to have been passed (marked as a green checkbox) based on the following criteria:
    a. The security application detects the malware simulator when it is executed according to the following criteria:
        i. It 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.
        ii. It identifies the simulator as suspicious or unknown and gives the option to run in a sandbox or safe restricted mode, and, when run in this mode, it meets the criterion c below.
    b. The security application prevents the simulator from injecting itself into the browser process.
    c. The security application does not allow the hooking/redirection of the API calls, or even with successful hooking, the password cannot be captured from the browser.
9. A test is deemed to have been failed (marked as a yellow warning) based on the following criteria:
    a. The security application fails to detect the simulator and then:
        i. The security application fails to prevent the simulator the simulator from injecting itself into the browser process, and gives no alert or provides informational alerts only.
        ii. The security application allows the hooking/redirection of the API calls, and the password can be captured from the browser.
    b. The security application identifies the simulator as malware or unknown and gives the option to run in a sandbox or safe restricted mode, and, when run in this mode, it:
        i. Fails to prevent the simulator the simulator from injecting itself into the browser process, and gives no alert or provides informational alerts only.

ii. The security application allows the hooking/redirection of the API calls, and the password can be captured from the browser.

10. Testing is conducted with all systems having internet access.
11. Each individual test for each security application is conducted from a unique IP address.
12. All security applications are fully-functional unregistered versions or versions registered anonymously, with no connection to MRG Effitas.

[i] http://www.theinquirer.net/inquirer/news/2285740/cyber-crime-is-a-bigger-threat-than-nuclear-war-uk-government-warns