



**MRG Effitas Online Banking Browser Security  
Assessment Project Q3 2013 - Q1 2014**

FOR MRG EFFITAS

## Contents

Introduction .....	3
The Purpose of this Report.....	3
Tests Employed.....	3
Security Applications Tested.....	4
Methodology Used in the Test.....	4
Test Results.....	8
Certification .....	10
Detailed description of the tests.....	11
Firefox test.....	11
Chrome test.....	11
Internet Explorer test I.....	12

FOR MRG EFFITAS USE ONLY

## Introduction

For four years, MRG Effitas has published Annual Online Banking Security Reports. As of 2013, we decided to publish quarterly reports; the first of these was released at the end of Q2 2013. Although testing continued, for unforeseen reasons that have been communicated to clients, we were unable to publish the subsequent quarterly reports on schedule. As a result, this report covers Q3 2013, Q4 2013 and Q1 2014.

## The Purpose of this Report

The core of our testing and ongoing research is the belief that cybercrime is the most significant threat faced by nation states and the most prevalent crime affecting corporations and individuals. This fact has recently been acknowledged by the governments of all major countries and most are now implementing strategies and policies, and allocating resources in order to counter these threats.

Over the years we have witnessed a drastic increase in the volume and diversity of malware found in the wild. MRG Effitas is currently processing over 350,000 unique malicious binaries and up to 250,000 malicious URLs every day and supplying these to our clients and other testing labs in an attempt to protect against them.

Aside from supplying zero day threats to clients and labs, our belief is that the most significant way in which we can help in the fight against cybercrime is to provide accurate and relevant assessment of security product efficacy.

MRG Effitas has been working with the IEEE, other testing labs and universities in an attempt to devise a set of testing standards that will allow accurate and relevant measurement of today's security products and also those that will be released in the next three years in the new emerging computing model.

It is vitally important that protection technologies evolve and improve – but how are we to achieve this if we are unable to accurately measure their efficacy against current and emerging threats? Product improvement cannot be achieved without the ability to measure real-world performance.

The purpose of this and our other reports is to be part of that process of measurement for the sake of improvement and efficacy assurance.

## Tests Employed

Applied metrology is complicated and imprecise science; as such, we position this and all our other work as the best assessments we can currently perform and not as an absolute or definitive determination.

In this report we ran the following tests:

In the Q1 2014 phase:

*Prevention of data exfiltration from SSL/TLS protected banking sites.*

Whilst detection is still a valuable metric, in itself it is not enough to determine real world efficacy as there will be instances where a system is compromised before a security solution is installed, or occasions where malware will bypass a preinstalled product. In these cases we need to be able to measure if active malware is able to perform data exfiltration or not.

In the Q4 2013 phase:

Attackers stealing session cookies instead of session details

Because of the heavy use of two-factor authentication for internet banking login log-in sessions, the effectiveness of password stealing has become less and less effective. Criminals have started to steal the representation of session cookies instead. After stealing a session cookie, the attackers are able to log-in

MRG Effitas has a range of simulators that employ MitB attacks, which have been used by financial malware and wider crimeware that we have reverse engineered.<sup>i</sup>

In the Q3 2013 phase:

Attackers using the victim browser to bypass protections

Because some traditional attacks like Zeus have been prevented by server side protections, e.g. analysis of the client IP, analysis of the client browser, etc., criminals are now actively exploiting new ways, which can bypass these types of protections. One new way is the “hidden browser” method, where the attackers can start a “hidden browser” on the user’s computer, and use that browser to log-in to the internet banking application and commit fraud.

Over the Q3 2013 - Q1 2014 period, we used our simulators to test the security products in the cohort against seven unique MitB attacks.

## Security Applications Tested

- AVG Internet Security 2014.0.4336
- Avira Internet Security 14.0.3.350
- BitDefender Internet Security 17.27.0.1146
- Comodo Internet Security 7.0.315459.4132
- Emsisoft Anti Malware 8.1.0.40
- Eset Smart Security 7.0.302.26
- F-Secure Internet Security 2.06 build 303
- G Data Internet Security 24.0.3.4
- Kaspersky Internet Security 14.0.0.4651
- McAfee Internet Security 12.8.944
- Microsoft Security Essentials 4.5.216.0
- Panda Internet Security 19.01.01
- Panda Safebrowser 2011
- Quarri 4.1.0.2452
- Symantec Norton Internet Security 20.4.0.40
- TrendMicro Titanium Security 7.0.1206
- Vipre Internet Security 7.0.6.2
- Webroot 8.0.4.70
- Wontok SafeCentral 3.1.21.3897
- Zemana AntiLogger 1.9.3.525

## Methodology Used in the Q1 2014 Test

1. Windows 7 Ultimate Service Pack 1 32 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”<sup>ii</sup>.
2. An image of the operating system is created.

3. The simulators 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.
7. Each Simulator test is conducted by:
  - a. Starting a new instance of Internet Explorer/Firefox/Chrome (or the Safe browser) and navigating to [https://www.paypal.com/en/cgi-bin/webscr?cmd=\\_login-submit](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, Firefox and Chrome, all three browsers are tested.
  - b. Text is entered into the Account login page of [https://www.paypal.com/en/cgi-bin/webscr?cmd=\\_login-submit](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) by the following criteria:
  - a. The security application detects the simulator when the security application is installed, and a mandatory scan is made.
  - b. The security application detects the 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 and 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 or d below.
  - c. The security application prevents the simulator from capturing and sending the logon data to the MRG results page, whilst giving no alerts or informational alerts only.
  - d. The security application intercepts the action of the simulator 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) by the following criteria:
  - a. The security application fails to detect the simulator when it is executed and then:
    - i. The security application fails to prevent the simulator from capturing and sending the logon data to the MRG results page location and gives no alert, or informational alerts only.
    - ii. The security application intercepts the action of the simulator 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 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:
    - i. Fails to prevent the simulator from capturing and sending the logon data to the MRG results page or local store and gives no alert, or 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. The test is deemed to neither pass nor fail, but warning (marked as a yellow exclamation mark) is given in the case of the following:
  - a. The browser extension in the safe browser is not working by default, but by social engineering the user can be convinced to install the malicious browser extension into the

safe browser (e.g. via social engineering), without any warnings from the safe browser. For example, this can be done by the malware intercepting HTTP traffic and redirecting the user to install the browser extension, or enable the disabled extension.

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

## Methodology Used in the Q4 2013 Test

1. Windows 7 Ultimate Service Pack 1 32 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"<sup>iii</sup>
2. An image of the operating system is created.
3. The simulators 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.
7. The cookie test is conducted by:
  - a. Starting a new instance of Internet Explorer (or the Safe browser), navigating to [https://www.paypal.com/en/cgi-bin/webscr?cmd=\\_login-submit](https://www.paypal.com/en/cgi-bin/webscr?cmd=_login-submit) , and logging in with a valid credential. Where the security application offers a secured or dedicated banking browser, this is used.
  - b. The file write activities of the browser are monitored, looking for session cookie files.
  - c. The content of the cookie files is inspected.
  - d. The content of the cookie files is copied to another browser.
  - e. In this browser, [https://www.paypal.com/hu/cgi-bin/webscr?cmd=\\_login-done](https://www.paypal.com/hu/cgi-bin/webscr?cmd=_login-done) is visited.
8. A test is deemed to have been passed (marked as a green checkbox) by the following criteria:
  - a. The security application does not write the protected session cookies to the hard disk.
  - b. The security application writes the protected session cookies to the hard disk, but in an encrypted form.
9. A test is deemed to have been failed (marked as a red cross) by the following criteria:
  - a. The security application writes the protected session cookies to the hard disk, in clear text form. The session cookie can be copied to another browser, and by navigating to [https://www.paypal.com/hu/cgi-bin/webscr?cmd=\\_login-done](https://www.paypal.com/hu/cgi-bin/webscr?cmd=_login-done), the attacker is logged into the application in the same session as the user.
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.

## Methodology Used in the Q3 2013 Test

1. Windows 7 Ultimate Service Pack 1 32 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"<sup>iv</sup>.
2. An image of the operating system is created.
3. The simulators 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.
7. The VNC Simulator test is conducted by:
  - a. Starting a new instance of Internet Explorer (or the Safe browser), navigating to [https://www.paypal.com/en/cgi-bin/webscr?cmd=\\_login-submit](https://www.paypal.com/en/cgi-bin/webscr?cmd=_login-submit) , and logging in with a valid credential. Where the security application offers a secured or dedicated banking browser, this is used.
  - b. The “hidden VNC” server is started on the client, in backconnect mode.
  - c. The connection from the victim VNC is accepted on the attacker side.
  - d. The same browser (IE or the Safe browser) is started in the “hidden VNC”.
  - e. In this hidden browser, [https://www.paypal.com/hu/cgi-bin/webscr?cmd=\\_login-done](https://www.paypal.com/hu/cgi-bin/webscr?cmd=_login-done) is visited.
8. A test is deemed to have been passed (marked as a green checkbox) by the following criteria:
  - a. The security application detects the simulator when the security application is installed, and a mandatory scan is made.
  - b. The security application detects the 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 and 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 or d below.
  - c. The security application prevents the simulator from connecting to the “attacker” VNC, whilst giving no alerts or informational alerts only.
  - d. The security application intercepts the action of the simulator 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) by the following criteria:
  - a. The security application fails to detect the simulator when it is executed and then:
    - i. The security application fails to prevent the simulator from connecting to the “attacker” VNC and gives no alert or informational alerts only.
    - ii. The security application intercepts the action of the simulator 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 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:
    - i. Fails to prevent the simulator from connecting to the “attacker” VNC and gives no alert or 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.

## Test Results

The table below shows the results of testing using the simulators employing reverse engineered MitB Attacks

### Q1 2014 results

Security application	Exact version	Firefox	Chrome	Internet Explorer BHO 1	Internet Explorer BHO 2	Safe browser	Overall
AVG Internet Security	2014.0.4336	✗	✗	✗	✗	n/a	✗
Avira Internet Security	14.0.3.350	✗	✗	✗	✗	n/a	✗
BitDefender Internet Security	17.27.0.1146	n/a	n/a	n/a	n/a	✓	✓
Comodo Internet Security	7.0.315459.4132	n/a	n/a	n/a	n/a	⚠	⚠
Emsisoft Anti Malware	8.1.0.40	✗	✗	✗	✗	n/a	✗
Eset Smart Security	7.0.302.26	✗	✗	✗	✓	n/a	✗
F-Secure Internet Security	2.06 build 303	✗	✗	✗	✗	n/a	✗
G Data Internet Security	24.0.3.4	✗	✗	✗	✓	n/a	✗
Kaspersky Internet Security	14.0.0.4651	⚠	✗	✓	✓	n/a	✗
McAfee Internet Security	12.8.944	✗	✗	✗	✗	n/a	✗
Microsoft Security Essentials	4.5.216.0	✗	✗	✗	✗	n/a	✗
Panda Internet Security	19.01.01	✗	✗	✗	✗	n/a	✗
Panda safebrowser	2011	n/a	n/a	n/a	n/a	⚠	⚠
Quarri	4.1.0.2452	n/a	n/a	✓	✓	n/a	✓
Symantec Norton Internet Security	20.4.0.40	✗	✗	✗	✓	n/a	✗
TrendMicro Titanium Security	7.0.1206	✗	✗	✗	✗	n/a	✗
Vipre Internet Security	7.0.6.2	✗	✗	✗	✓	n/a	✗
Webroot	8.0.4.70	✗	✗	✓	✓	n/a	✗
Wontok SafeCentral	3.1.21.3897	n/a	n/a	n/a	n/a	✓	✓
Zemana AntiLogger	1.9.3.525	✗	✗	✗	✓	n/a	✗

✓	The application prevented the simulator from capturing data.
✗	The application failed to prevent the simulator from capturing data.
⚠	The application failed to prevent the simulator from capturing data after the user was socially engineered to install the simulator into the browser.



## Q4 2013 results

Security application	Exact version	Cookie stealing
AVG Internet Security	2014.0.4336	✗
Avira Internet Security	14.0.3.350	✗
BitDefender Internet Security	17.27.0.1146	✗
Comodo Internet Security	7.0.315459.4132	✗
Emsisoft Anti Malware	8.1.0.40	✗
Eset Smart Security	7.0.302.26	✗
F-Secure Internet Security	2.06 build 303	✗
G Data Internet Security	24.0.3.4	✗
Kaspersky Internet Security	14.0.0.4651	✗
McAfee Internet Security	12.8.944	✗
Microsoft Security Essentials	4.5.216.0	✗
Panda Internet Security	19.01.01	✗
Panda safebrowser	2011	✓
Quarri	4.1.0.2452	✓
Symantec Norton Internet Security	20.4.0.40	✗
TrendMicro Titanium Security	7.0.1206	✗
Vipre Internet Security	7.0.6.2	✗
Webroot	8.0.4.70	✗
Wontok SafeCentral	3.1.21.3897	✓
Zemana AntiLogger	1.9.3.525	✗

✓	The application protects the session cookies.
✗	The application does not protect the session cookies.

## Q3 2013 Results

Security application	Exact version	Hidden VNC
AVG Internet Security	2014.0.4336	✓
Avira Internet Security	14.0.3.350	✓
BitDefender Internet Security	17.27.0.1146	✓
Comodo Internet Security	7.0.315459.4132	✓
Emsisoft Anti Malware	8.1.0.40	✗
Eset Smart Security	7.0.302.26	✗
F-Secure Internet Security	2.06 build 303	✗
G Data Internet Security	24.0.3.4	✓
Kaspersky Internet Security	14.0.0.4651	✓
McAfee Internet Security	12.8.944	✗
Microsoft Security Essentials	4.5.216.0	✓
Panda Internet Security	19.01.01	✗
Panda safebrowser	2011	✓
Quarri	4.1.0.2452	✓
Symantec Norton Internet Security	20.4.0.40	✗
TrendMicro Titanium Security	7.0.1206	✓
Vipre Internet Security	7.0.6.2	✓
Webroot	8.0.4.70	✓
Wontok SafeCentral	3.1.21.3897	✓
Zemana AntiLogger	1.9.3.525	✓

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

## Certification

In order to attain the 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.

The MRG Effitas Online Banking Browser Security Certification for Q1 2014 is awarded to the following products:

- BitDefender Internet Security
- Quarri Armored Browser
- Wontok SafeCentral

The MRG Effitas Online Banking Browser Security Certification for Q4 2013 is awarded to the following products:

- Panda Safebrowser
- Quarri Armored Browser
- Wontok SafeCentral

The MRG Effitas Online Banking Browser Security Certification for Q3 2013 is awarded to the following products:

- AVG Internet Security
- Avira Internet Security
- Comodo Internet Security
- G Data Internet Security
- Kaspersky Internet Security
- Microsoft Security Essentials
- Panda Safebrowser
- Quarri Armored Browser
- TrendMicro Titanium Security
- Vipre Internet Security
- Webroot SecureAnywhere
- Wontok SafeCentral
- Zemana AntiLogger

The only products that passed all certification tests conducted over the three-quarter period were:

- Quarri Armored Browser
- Wontok SafeCentral

## Detailed Description of the Tests

### Firefox test

A source code copy is downloaded from the following malicious browser extension:

<https://github.com/MRGEffitas/ZombieBrowserPack/tree/master/lite/client/firefox>

The browser extension is lightly obfuscated in order to bypass signature based detection. The browser extension is installed to the browser before any security solution is installed. The browser extension is able to steal POST data contents (which contains usernames, passwords), and send this to a server operated by MRG Effitas.

### Chrome test

A copy is downloaded from the following malicious browser extension:

<https://github.com/MRGEffitas/ZombieBrowserPack/tree/master/lite/client/chrome>

The browser extension is lightly obfuscated in order to bypass signature based detection. The browser extension is installed to the browser before any security solution is installed. The browser extension is able to steal POST data contents (which contains usernames, passwords), and send this to a server operated by MRG Effitas.

Because of the way Chrome allows the installation of browser extensions outside of the official extension store, the installation of the browser extension needs “developer mode” to be enabled in the browser. The Chrome browser warns the user about developer mode, every time the browser is started. This means that in a real-world scenario, it is less likely that users will install a malicious Chrome browser extension, than is the case for other browsers.

## Internet Explorer test 1

A proprietary browser helper add-on (BHO) has been created. The BHO is installed to the browser before any security solution is installed. The BHO is able to steal POST data contents (which contain usernames, passwords), and send this to a server operated by MRG Effitas.

## Internet Explorer test 2

A proprietary browser helper add-on (BHO) has been created, using the services of crossrider.com. The BHO is installed to the browser before any security solution is installed. The BHO is able to query the C&C server operated by MRG Effitas, and execute any JavaScript command in the context of the currently active webpage. This makes it possible to change the content of the website, web-inject attacks, cookie stealing, etc.

We developed this malicious browser extension because we saw a lot of BHO malware created by this online service. The service makes it very easy for criminals to create malicious browser extensions and BHO's.

## Hidden VNC test

The Carberp leak in June 2013 contained a patched VNC server. This VNC server creates a “hidden desktop”, which is not visible to the user. The attacker can connect to the VNC server either via backconnect or bind mode. By starting the Internet Explorer browser in the hidden desktop, the browser inherits all the session cookies from the Internet Explorer browser opened on the user's desktop. This means that if the user is logged into an internet banking application, so is the attacker.

We did not obfuscate the binaries at all, which means that basic signature detection could also stop this attack.

A similar functionality is implemented in other malware families as well, like Zeus and SpyEye.

## Session cookies stored on the disk

This test is a designed to check of the safe browser, rather than a simulator.

We logged into the [www.paypal.com](http://www.paypal.com) web application with a valid account, and checked if the session cookies are written to the hard disk in clear text. The idea behind this test is that a malicious process running on the victim's computer can monitor the session cookies written to the disk, and in case of interest, it can steal the cookies and send them to the attackers. The attackers can insert the cookie values into their browser, which results in a logged-in state, until the user logs out of the application.

Because no simulator has been used in this check, it is possible that the attack could have been prevented either by reputation based protection or by outbound firewall alerts.

---

<sup>i</sup> It is necessary to use simulators with reverse engineered MitB attacks as testing using real financial malware which relies on ITW C&C servers is unlawful under the Computer Misuse Act in the UK, as it requires that a malicious code is run on a third party computer without their knowledge or consent.

<sup>ii</sup> AES includes Internet Explorer 8.0.7601.17514, Firefox 27.0.1, and Chrome 34.0.1847.116 m, all fully updated at the time of the test.

<sup>iii</sup> AES includes Internet Explorer 8.0.7601.17514, Firefox 27.0.1, and Chrome 34.0.1847.116 m, all fully updated at the time of the test.

<sup>iv</sup> AES includes Internet Explorer 8.0.7601.17514, Firefox 27.0.1, and Chrome 34.0.1847.116 m, all fully updated at the time of the test.