FEBRUARY 2016

virus BULLETIN

Covering the global threat landscape

VBWEB COMPARATIVE REVIEW FEBRUARY 2016

INTRODUCTION

Whether you click on a link in a spam email that leads to malware being downloaded onto your machine or browse a website whose advertisements serve an exploit kit to your vulnerable browser, there are many ways for you to get infected through HTTP.

This isn't a new phenomenon, and many security vendors offer solutions that are designed to block malicious HTTP traffic before it reaches the local network, thus removing the threat before the user is reliant on the capabilities of their installed endpoint security solutions. We've had many requests from vendors to test such solutions, and for the past few years, we have been working on developing such a test.

For a number of reasons, this has been far from trivial.

Firstly, while malicious web requests have long been a plague in absolute terms, relatively speaking, they are extremely rare. Most web requests are both legitimate and harmless, and simply making a large number of requests to random URLs is unlikely to result in a large enough number of malicious responses to carry out a test.

Moreover, even those URLs that *do* lead to malware often only serve the malware once, and often have other ways to prevent security researchers from receiving the malicious response – so there isn't really such a thing as a simple 'malicious URL' that could be used in the test.

After a lot of hard work, and with some helpful feedback from many industry members, we are pleased finally to have finished building the test.

We will, of course, continue to make tweaks and adjustments to the methodology both as the threat landscape evolves and as we get requests from vendors to do so – as with other *Virus Bulletin* tests, we run these tests for the security community and it is only by fully engaging with the community that we can do this. For now, though, we are ready to run our tests on real products and publish the results.

This is the report of the first test. Welcome to VBWeb.

THE TEST PHILOSOPHY

The philosophy behind the VBWeb tests rests on two pillars: the fact that we consider 'cases', not 'URLs', and the fact that all traffic is cached and replayed.

A *case* starts with a URL and is anything that happens as a consequence of the URL being entered into a browser: the images and stylesheet files that are loaded, the Flash files that are downloaded and played, and whatever kind of maliciousness is happening.

Of course, different URLs lead to different cases, but the same URL doesn't always result in the same case. For instance, the time the request was made, the browser that made the request, its add-ons, and the IP address from which the request was made, can all influence the response (and often do). Sometimes, even client- or server-side randomness means that under the very same circumstances, the same URL will lead to different cases.

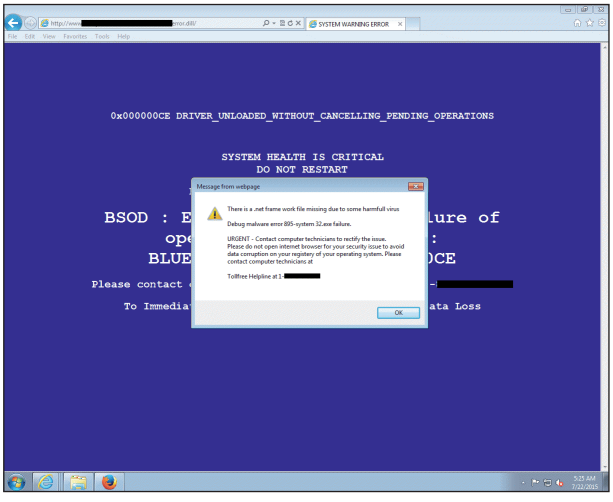
And that explains the second pillar. As with all of *Virus Bulletin*'s certification tests, VBWeb is a comparative test in which products are tested against each other. The only way to do this fairly is to make sure the very same content is served to each product.

Hence in our tests, a potentially malicious request is made without any filter in front of it. The responses and resulting

ISSN 1749-7027

subsequent requests are stored in cache. Then the same request is immediately replayed for all products, to whom responses are being served from this cache.

If a request gave a malicious response, a product is considered to have blocked the case if they didn't serve the malicious part of the response to the browser. It doesn't matter here whether the initial URL was blocked or whether only the payload was blocked.



THE TEST METHODOLOGY

During the test period, which ran from 4 to 15 December 2015, we used a number of public sources combined with our own research to open URLs that we had reason to believe could serve a malicious response in one of our test browsers, selected randomly.

When our systems deemed the response likely enough to fit one of various definitions of 'malicious', we made the same request in the same browser a number of times, each time with one of the participating products in front of it. The traffic to the filters was replayed from our cache.

Note that we did not need to know at this point whether the response was actually malicious, thus our test didn't depend on instances already known to the industry or community. During the review of the corpus days later, we analysed the responses and included cases in which the traffic was indeed malicious.

We looked at various types of malicious responses, including spam/scam sites and phishing pages, but we decided to concentrate only on direct malware downloads (where the URL resulted directly in the download of malware) and drive-by downloads (where the URL was an HTML page that forced the browser to download and/or install malware in the background).

We also sent legitimate traffic to the products, to ensure that their use wouldn't unreasonably hinder the user's browsing experience.

Prior to running the test, we decided that a product would pass the test, and would thus be awarded VBWeb certification and the VBWeb logo, if it blocked a minimum of 70% of the total corpus of malware and drive-by downloads.

It should be noted here that web filters only play a part in the prevention of malware making it to the machine. Thus, while it would be great if such products were to block 100% of malicious responses, users and system administrators are advised not to rely solely on this, and are urged to follow best practices, from making sure browsers and operating systems are as up to date as possible, to running security products on the endpoints.

In this test, we checked products against 259 URLs serving malware and 98 drive-by downloads (exploit kits).

The test focused on unencrypted HTTP traffic. It did not look at extremely targeted attacks or vulnerabilities in the products themselves.

TEST MACHINES

We used two virtual machines, selected randomly, from which to make requests. On each machine, an available browser was selected at random.

We found that, in practice, we were far more likely to be given a malicious response for the *Windows 7* machine using either version of *Internet Explorer*; hence most cases that ended up in the test, used this configuration.

Windows XP Service Pack 3 Home Edition 2002 (x86)

This machine had the following software installed:

- *ADOBE Flash Player* 12 *Active X* 12.0.0.38
- *ADOBE Flash Player* 12 plug-in 12.0.0.43
- *ADOBE Reader XI* 11.0.0.0
- *Apple Application Support* 2.0.1
- *Apple QuickTime* 7.70.80.34
- *ORACLE* Java 7 update 51 7.0.510
- *VLC* media player 2.1.3

The following browsers were installed:

- *Windows Internet Explorer* 8 (8.0.6001.18072)
- *Mozilla Firefox* 28.0

Windows 7 Service Pack 1 Ultimate 2009 (x86)

This machine had the following software installed:

- *ADOBE Flash Player* 13 *Active X* 13.0.0.182
- *ADOBE Flash Player* 13 plug-in 13.0.0.182
- *ADOBE Reader XI* 11.0.0.0
- *Apple Application Support* 2.0.1
- *Apple QuickTime* 7.70.80.34
- *Piriform CCleaner* 5.0.4
- *ORACLE Java 7* update 51 7.0.510
- *Microsoft* .NET framework 4.5.2 (4.5.51.209)
- *Microsoft Silverlight* 5.1.10411.0
- *VLC* media player 2.1.3

The following browsers were installed:

- *Windows Internet Explorer* 11 (11.0.09600.17843 update 11.0.20)
- *Windows Internet Explorer* 9 (9.0.8112.16421 update 9.0.37)
- *Mozilla Firefox* 28.0

RESULTS

As with all of *Virus Bulletin*'s certification tests, we offer the opportunity for products to be tested 'publicly' or 'privately', with developers of the products entered into the private tests receiving feedback, but their results not being made public. For obvious reasons, once a test has started, participants may not switch from 'public' to 'private' or (vice versa).[1]

Somewhat understandably, given that this is a new test, developers of only one product had enough confidence in both their product and our test to go public this time. A number of products were included to be tested privately, and it is worth pointing out that over the past few years, a number of dry runs had taken place: the tests were thus not entirely new to many of the participants.

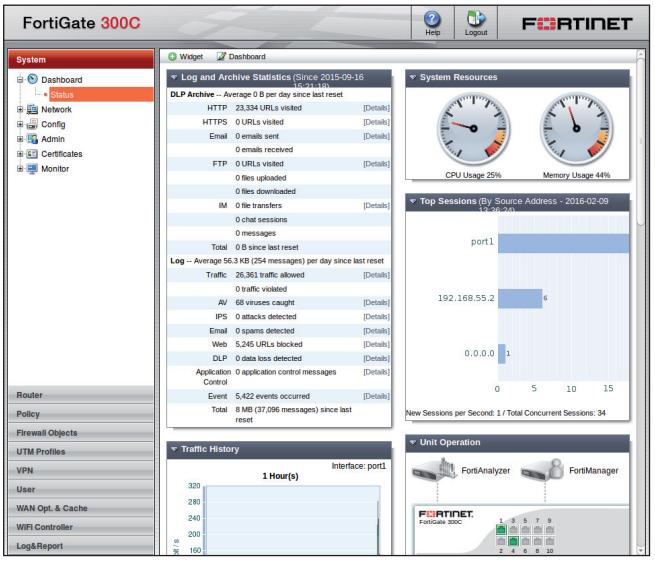
FortiGate

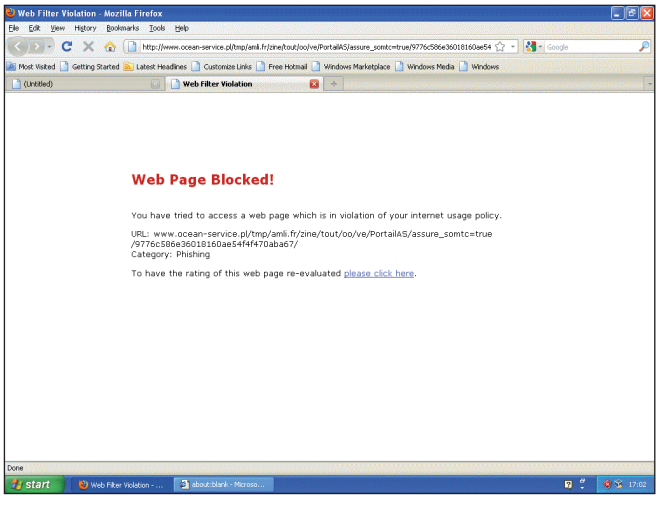
Malware block rate: 97.7%

Drive-by download block rate: 41.4%

Total block rate: 83.5%

Fortinet's developers clearly have a lot of confidence in their *FortiGate* appliance, as they were the only ones to submit their product to the first public test.





Described by the company as a 'High Performance Next-Gen Firewall', *FortiGate* does far more than just filter HTTP traffic.

The product runs as a transparent proxy to filter web traffic, which means that end-users and system administrators do not need to configure browsers and other tools that use HTTP to use the proxy. We found setting up the small, shiny white box quick and straightforward.

FortiGate can be managed through a web interface that is easy to work with, and which looks similar to that of the *FortiMail* email security appliance, which we have been testing successfully in the VBSpam tests for many years.

The company also has an excellent test record in our VB100 tests, so it didn't come as too much of a surprise to find that the *FortiGate* appliance blocked all but six

[1] The goal of our tests is to be informative and fair; if issues occur that are deemed to be beyond the product's control, we allow a switch from public to private.

malicious downloads before they could reach the targeted organization's network.

When it came to drive-by downloads, the product's performance was less impressive, though it did block a significant number of them. Most of these were stopped by blocking the 'gate' that would lead to the exploit kit, which shows that *Fortinet*'s researchers and developers are keeping up with this evolving part of the threat landscape.

In total, the product blocked over 83% of the malware in this test, making it a deserving winner of the very first VBWeb award.

The next VBWeb test will run in March 2016, with the results scheduled for publication in April. Developers interested in submitting products, or who want to know more about how Virus Bulletin can help their company measure or improve its web filter performance, should email martijn.grooten@virusbtn.com.

Editor: Martijn Grooten

Chief of Operations: John Hawes

Security Test Engineers: Scott James, Tony Oliveira, Adrian Luca, Ionuț Răileanu, Chris Stock

Sales Executive: Allison Sketchley

Editorial Assistant: Helen Martin

Developer: Lian Sebe

Consultant Technical Editor: Dr Morton Swimmer

Tel: +44 (0)1235 555139 Fax: +44 (0)1865 543153

Email: editorial@virusbtn.com Web: https://www.virusbtn.com/