Global Information Assurance Certification Paper

Mitigating Browser Based Exploits through Behavior Based Defenses and Hardware Virtualization

GIAC (GCIH) Gold Certification

Author: Joseph Faust, CISSP, GCIH, GSEC, josephfaust.dc@gmail.com
Adviser: Rick Wanner

Accepted: July 23rd 2011

Abstract

As the adoption of the Internet grows worldwide, the volumes of valuable data being transmitted and stored in digital format become increasingly attractive targets. Vulnerabilities and weaknesses in web architectures and environments which are tasked with storing and preserving this data have been and continue to be exploited every day. In recent years there has been a trend to directly exploit the browser bypassing traditional perimeter security defenses to gain access to internal networks and databases, resulting in the theft of valuable corporate, non-profit and government data from closed networks. This paper studies how implementing behavior based browser defenses and application isolation through virtualization with Invincea, organizations can help bolster defenses against this type of attack and mitigate the impact of client side browser exploits. It will also examine configuration options in Invincea that can greatly help limit exposure to malware and help prevent providing the attackers a beach head into a victim's network through exploiting the browser.

Joseph Faust, josephfaust.dc@gmail.com

1. Introduction

There does not seem to be a day or week that goes by that one does not encounter a headline story about an organization being compromised and infiltrated by attackers. While in many cases, multiple vectors are used in the attack, the majority of cyber-attacks and successful targeted campaigns recently seen rely on first exploiting vulnerabilities in the client web browser to execute arbitrary code during the initial phase of the attack (*Cyveilliance, 2011).* A rash of high profile breaches suggest that typical controls to date have had limited success in preventing the exploitation of the browser (2011, Federal Computing Weekly). New approaches and ways to rethink the trust relationship between the client and the website are needed and should be explored. In the following narrative we will explore one solution that can augment an organization's security defenses through behavior based detection and hypervisor isolation, utilizing Invincea to accomplish isolation, detection and prevention of client side web threats.

2. The Browser as an Attack Vector

To understand more about client side, browser based attack and defenses, it is helpful to first analyze the reasons for the increase in popularity of targeting the browser as an attacker vector. Through this analysis one can more acutely understand the causes driving the increase in attacks against the client side browser vector.

2.1 Consistent Access Device

When looking at the growth rate of users and adoption of the Internet, one constant among adopters is the use of a web browser to access content, irrespective of platform choice or device. When finding ways to automate attacks one need not look further than the browser as a vector, as it has become the ubiquitous interface to the web.

Joseph Faust, josephfaust.dc@gmail.com

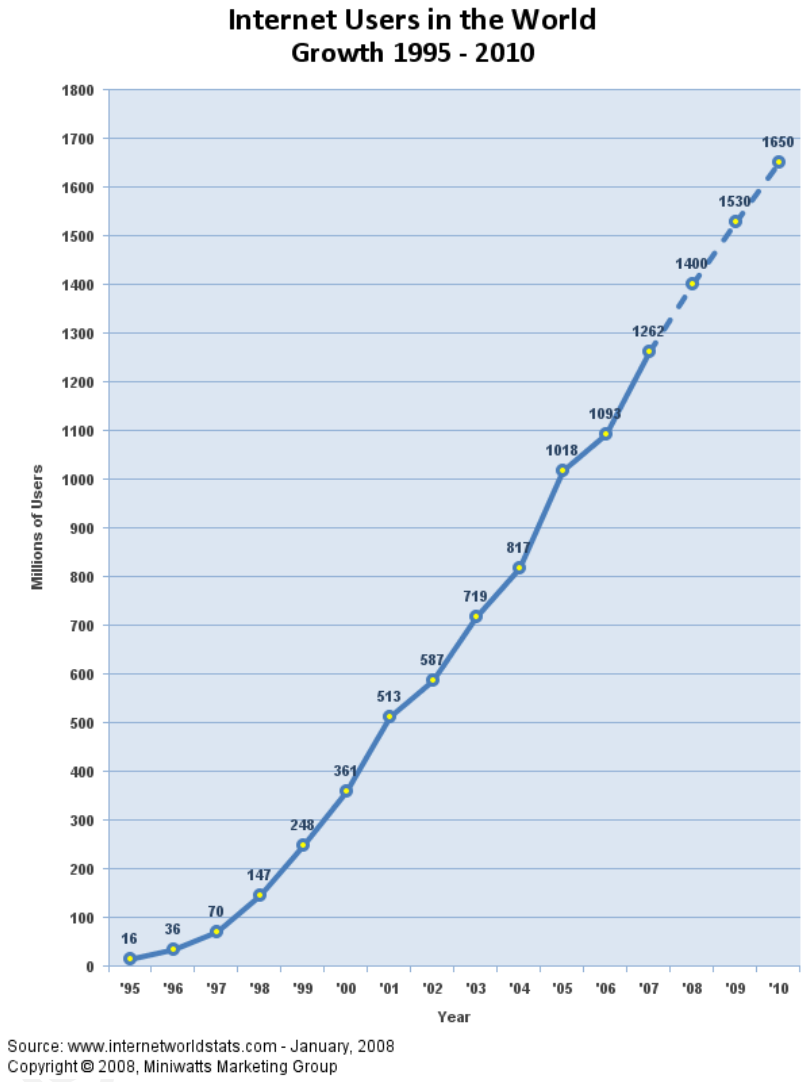

Figure 1 - Growth of the Browser Base 1995-2010 (InternetWorldStats.com, 2010)

Various browsers exist for desktop operating systems, mobile devices, televisions and smartphones and share a common implementation, making them an attractive target to exploit. Along with delivering web content and pages to interact with users, attackers have realized the exploitation potential for compromising desktop users through malware that is mass distributed from an infected website. (2011, Goodin). Economies of scale are at play and attackers will look to maximize both the field of potential victims that can be compromised and their return on the time and effort invested to create such an exploit.

Joseph Faust, josephfaust.dc@gmail.com

2. 2 High Volume of Intermediaries & Targets

Due to the efficiencies and potential cost savings for organizations to build web applications as opposed to traditional desktop based software, which can require code to be developed for each operating system environment, web software has grown exponentially. Business has realized significant benefits of moving applications off the desktop and into web applications which have helped cement the browser as a very profitable, target rich vector to attack. The advancement in the number of web applications and services moving to the cloud also has made the browser the ubiquitous interface to software today and a ripe target for exploitation. By compromising websites that businesses and government organizations use, these same sites can then serve as intermediaries for infection to anyone who simply visits these websites via a browser (Joch, 2011). The high volume of website and client devices accessing these websites through web browsers makes for an attractive target.

2. 3 Change in Motives

With the increase in important information being exchanged online and the value of such key information, the incentive to profit from stealing this critical information has increased as well. Governments and Corporations continue to rely more on this medium to conduct business and commerce. Over the years they have adopted systems which heavily rely upon the Internet and its critical infrastructure to do business. Attackers have followed the targets accordingly and have moved to operate in this space with greater sophistication. (Guerra, 2009).

2. 4 Automation of Attacking the Browser

The rise in the use and maturity of web attack toolkits, has allowed very little or no interaction to be needed on the client side for many exploits to be successful. According to the Symantec 2011 Threat Report, the proliferation of Web Attack

Toolkits drove a 93% increase in the volume of Web-based attacks in 2010 over the

volume observed in 2009 (2011, Symantec). Through the use of these toolkits and browser exploitation frameworks, attackers can automate the process of exploiting the browser with only minimal manual intervention to compromise targets (Messmer, 2011). Automation in the ability to attack and exploit the browser for monetary gain continues to drastically increase the number of successful exploitations of the client side browser and host machine (M86, 2010).

3. The Browser Then and Now

To better understand the current state of the client web browser and its insecurities and weaknesses, it is important to look back on the evolution of the browser to better understand how it has evolved to its present state today.

3.1 A Recap of Browser History

The original browser and Hypertext markup language (HTML) specification created by Time Berners-Lee, was designed for scientist and academic experts to exchange text based information across the World Wide Web (Bellis, 2010). After the first graphical browser Mosaic, was later released in the early 1990's, the business community embraced the Internet and its potential for conducting commerce.

As businesses moved to the web, they created online presences, originally serving mostly as online static brochures. With the demand to provide more capabilities and interactivity and dynamic content for end users, scripting languages were added to the client browsers, as well as plug-ins to provide additional browser capabilities. (Shinder, 2004) Such examples include flash plug-ins to support web optimized video playback, as well as other technologies allowing client side code execution through technologies such as java applets and active-x.

These advances in the browser aimed to make the user internet experience a rich, desktop like user experience. With each plug-in or extension added to the browser, the

attack surface increased, and expanded with subsequent code iterations of the client browser. These changes are reflected today in the browsers that continue to provide a desktop-like experience when interacting with purpose built websites.

In the 1990's firewalls were in the early stages of development and mass adoption across every device was not a standard. (Avolio, 2011). As firewall technology matured and adoption across the network and devices became standard, attackers continued to adapt offensive techniques and look for new attack vectors.

3.2 Fast Forward to the Present

Firewalls today are now significantly more mature than their predecessors. Although commonly probed for unpatched vulnerabilities, misconfigurations and zero day-attacks to access a target victim's machine, today's firewalls offer much improved perimeter defense compared to those available years ago. The hardening of the perimeter has had the effect in many ways, of putting more focus on attacking the browser. The browser serves as the portal to the Internet for users which crosses internal network and external network boundaries. (Goodchild, 2008)

As the firewall defenses matured, malicious code has matured in parallel with it and largely moved the attack vector to the application layer, where the browser has a communication channel to operate through the firewall (Harri, 2010).

Browsers today have grown increasingly complex over the years, with some containing over one million lines of source code (Barth, Jackson, Reis, Google Inc., 2009). This complexity has allowed for many creative exploits to be crafted and used in attacks against the browser in a variety of ways.

4. Current Defensive Controls

4.1 Limitations & Effectiveness

Protecting the browser and client machine against these types of web based malware threats has been met with limited success. A variety of traditional defenses are typically applied to protecting an organization. Unfortunately the effectiveness of these controls in defending against today's malware hasn't been enough to truly stem the flow of these threats, as evident in the large number of compromises which occur via the browser (Joch, 2011). A quick review of traditional defenses will help provide insight into the challenges traditional defenses have faced against this threat.

Working from the network defense perimeter in, the border router along with the firewall are the first several controls that can be applied against malware. The border router can be configured with an access control list to limit the volume of requests that reach the firewall, filtering out traffic that is known to be malicious based off community advisories, best practices, etc. Firewalls can also take advantage of blacklists and reputation updates, where known malicious domains are blocked based off of a community-derived and updated list. These blacklists can provide some level of defense against general malicious sites. Although the firewall as a defensive control quickly loses effectiveness in preventing malware from exploiting the browser and host machine, as today's malware has grown very dynamic. In many cases today, malware will exist for only a few hours on a given domain and then it will rapidly change to a new domain to avoid getting blocked. The malware remains ahead of the curve in detection. The firewall relies on a negative security model of looking for known bad sites to block malicious domains. This limits the defense coverage the firewall can provide against these types of threats.

Network based Intrusion Detection and Prevention Systems (IDS/ IPS) and Security Gateways also are limited in the coverage that they can provide in defending against exploits delivered through the web. These devices commonly work off packet inspection which relies on signatures of malware to be available to detect and block the

attack. Coverage in this area falls down due to the situation again where defenses are looking for a known bad. If a Zero Day, where known defensive signatures do not yet exist for the exploit code being used in an attack, then malware will not be recognized and the malicious packets will be allowed through. If a signature does exist for a piece of malware, then it will be blocked. However, advances in evading detection by packing or encrypting the malware can potentially provide a unique approach to each piece of malware, allowing it to slip past defenses. Through obfuscation, web based attacks are able to successfully avoid detection and pass through to the internal network of the target.

Traditional desktop based defenses that have been deployed and in place for years in the form of anti-virus protection have had limited effect in being able to detect and prevent many web based exploits due to the growing sophistication of the malware (Alperovitch, 2011). Once again detection can be limited due to the signature based negative security model which the Anti-Virus engines rely upon. If the malware which is attempting to exploit the machine does not match a known signature or characteristics, it bypasses the Anti-Viruses defenses. Also of concern is custom made malware that is created in targeted attacks. Malware that has been customized greatly weakens the ability to detect the malicious code. Unless the malware is found in the wild or a derivative of its' code-base is, no defensive signatures can be created. The opportunity to detect and prevent infection against the malware with a negative security model is very limited. More advanced engines have added heuristics detection abilities which have improved detection, although the scope of the processes and interactions that they monitor is quite large. With this large surface area to cover, complexity is increased and the detection engine cannot be as aggressive as it may need to be, without risking hindering the usability of the various applications by alerting with false positives. Anti-virus defenses continue to struggle with the dynamic nature / obfuscation and rapid evolution of the malware that organizations face today.

5. Challenges in Educating Common Users

5.1 Limited Options

Options exist for security professionals to safely browse the web with minimal risk of exploitation, but few practical options that do not require deeper technical knowledge have existed for the average user. The common user represents the majority of users who browse the Internet to achieve various tasks in their jobs, but are not necessarily technical users. Providing security controls and browser configuration to assist the ordinary user in making good security decisions and preventing exploits from hitting organizations has proven a difficult challenge (Gross, 2011). Security can many times be seen as an in-convenience to the tasks at hand, and when given a chance many users will simply accept and by-pass any warnings that a browser or add-on software could provide that would indicate to not visit a particular site or link.

5.2 Social Engineering the Common User

Attackers continue to use phishing heist techniques, where they attempt to gain sensitive information from unsuspecting end users by masquerading as someone else. Targeted attacks coupled with phishing tactics continue to show significant success rates. This happens despite increased user education and general awareness as the line between "authentic" versus "knockoff" is becoming harder to discern for a typical end user who comes across a malicious email attachment or fake website (Computer Associates, 2006). At the end of the day, users are concerned with getting their tasks done. No amount of training on the part of any organization will be able to provide all users with the knowledge and ability required to discern legitimate emails or web links from the bad ones, all of the time (Jakobsson & Ramzan 2008).

Joseph Faust, josephfaust.dc@gmail.com

5.3 Advice For Safe Browsing

Traditional advice to only visit reputable sites, no longer provides any significant level of protection. With the attacks that are ongoing today, little protection is afforded by visiting a familiar site versus an unfamiliar site. Many browser and client machines have been compromised by simply visiting a well-known website (Barwinski, Irvine & Levin, 2006). These sites which deliver "drive-by-download exploits" have been successfully attacked, compromised and loaded with an exploit to deliver to unsuspecting users visiting the website. (Grossman, 2008) This happens all too many times, as evident from the headlines played out in the news. Attacks against the client continue en masse, as the end user is left to make best guess security decisions on the trustworthiness of content.

6. The Present Trust Model & A New Way Forward

6.1 The Present Trust Model

We know that the browser is a weak link exposed to the Internet, but yet we continue to trust the browser. An analogy of this could be represented in looking at the medical field when attempting to contain an outbreak. A doctor who is tasked to treat patients would suit up accordingly and interact with his patients to assist them with medical treatment. When returning to an area that is free from contagious germs, the doctor would un-suit and scrub up accordingly to ensure that he would not infect the safe zone. In the present security analogy, and model that browsers operate in today, we do not have the doctor un-suit, scrub and discard their materials in the trash. Any potential germs which the doctor might have come into contact with are attempted to be detected and removed prior to the doctor returning to the safe zone, instead of throwing away the materials used while operating in the infected zone.

The implicit trust that the browser is safe and free from malware, leads to lowering defenses and ultimately compromise in many cases. The model assumes that the browser remains free from infection after navigating various sites on the Internet.

Joseph Faust, josephfaust.dc@gmail.com

Moving away from full trust of the browser sets us up for a more realistic view of the world.

6.2 Stop Trusting Your Browser

Distrust of the browser can be accomplished with full application isolation through hardware virtualization. With hardware virtualization in place the hypervisor presents a set of virtual hardware resources (CPU, network, disk, etc.) to a full OS kernel running in the virtual environment. When isolated, an application browser will run fully encapsulated, running as guest of the host machine. For the scope of this paper, we will discuss application isolation and behavioral detection with Invincea as a means to isolate the browser and relinquish the inherent trust that the browser is provided.

7. Invincea and Architecture Overview

7.1 Invincea Overview

Invincea is software that takes aim at desktop browser protection and utilizes behavior based defense to detect and analyze interactions while traversing the web. The software defense engine originates from research initially developed in an early prototype concept of a DARPA funded project that Dr. Anup Ghosh led, and which was later commercialized into formal software by Invincea (Invincea, 2011). Invincea operates in tandem with existing security protections in an organization, but operates on the key principle of distrust of the client web browser and isolating interactions with websites through a hypervisor to control exposure to malicious activity.

7.2 Invincea Architecture

Invincea' s browser protection architecture and functional duties are distributed across several pieces of software. On the desktop, standalone software is installed which enables a web browsing session that is isolated against web based threats.
A behavior analysis engine also monitors activity both from within the guest and host operating system environments to prevent the execution of malicious software.

A centralized log aggregation server, known as Invincea Threat Server is available to process information fed from each Invincea Browser client deployed across an organization's environment. This architecture is loosely coupled, allowing the client software to operate independently of the Threat Server with full protection capabilities, in the event the Threat Server is not available.

7.3 The Client Software

On each user's desktop that is in need of protection, the Invincea browser and monitoring agent libraries are installed. The browser is completely isolated in a guest operating system through hardware virtualization and it has a footprint for Memory, Disk, and Network with the host. For performance, it is recommended to have at least 2GB of Memory and 500-600 MB hard drive space. The client will run on Windows XP, Vista and 7, 32/64 bit Intel or AMD processors 1.5 GHz or greater and with as little memory as 2GB.

When accessing the Internet, the browser icon is clicked which invokes the encapsulated operating system waiting in a pristine state and spawns a web browser. To indicate that an untrusted browser session is initiated, a red box is drawn around the browser window to indicate this type of untrusted browser session is underway. From

here forward, the interactions with any websites are under the view and context of the untrusted browser and the behavior engine which continuously make assessments of each web page viewed during a browser session.

Figure 2 – Untrusted Browser Session Delineated with Visual Red Border

7.4 Server Software, Centralized Logging

For insight into the web based attacks that attempt to exploit the browser, a central logging server should be configured. This will allow for on-demand collection and aggregation of suspicious activity from any protected web browser session. Reviewing the forensic data Invincea captures during its interrogation of the web browser session can be critical in better understanding the severity of the attack and its potential for impact on an environment. To ensure events make it to the appropriate Information Security Staff, events can be configured to be pushed to the Invincea Threat Server. For organizations which run other central logging technology such as a Security Information and Event Management (SIEM) or sys log server, Threat Server can be configured to integrate with several of these other log repositories in their native format.

Figure 3 – Configuration, Centralized Log Reporting

7.5 Isolation through Hardware Virtualization

The Invincea architecture implements complete browser isolation through a dedicated OS kernel via a Type 2 hypervisor, otherwise referred to as Hardware Virtualization. In this type of isolation the hypervisor presents a set of virtual hardware resources (CPU, network, disk, etc.) to a full OS kernel running in the virtual environment. This lightweight operating system fully encapsulates the browser to limit the likelihood of a successful exploit crossing the virtual boundary, allowing code to be executed, but contained within the virtual machine. By design, the browser application session and state will not persist past the given session. Any changes the attacker has made to the target will be disposed of, along with the remainder of the virtual machine state.

7.6 Threat Detection Behavior Engine

The Threat Detection workflow is triggered when the agent detects behavior that is not indicative of a standard request, such as unauthorized modifications to registry keys. The behavior engine in Invincea relies on sensors internal to virtual environment that interact with the host libraries and sensors on the host. Operating off a positive security model, deviations from the designated norm result in the engine making a determination on the validity of such interaction.

7.7 Enterprise Manageability

The virtual browsers can be packaged and deployed through enterprise software deployment with MSI installers for simplified deployment to client machines. A browser icon is located on the desktop which when clicked, invokes the virtual browser that is encapsulated and running in the background. A visual indicator in the form of a red border indicates to the end user that an untrusted browser session is running. To

maximize effectiveness, multiple browser images could be deployed. Each image could have a slightly different codebase and particular end user audience in mind. This tailored customization allows for dialing up or down the aggressiveness of the defenses with various personas. Organizational flow controls such as proxy settings can be configured in the clients as well.

7.8 Reducing the Attack Window

Setting the configuration on the Virtual Machine to automatically restore to a trusted, pristine state at pre-defined intervals can reduce the window of opportunity that an attacker has to work against a given target. With the attack window shortened, even if an undetected compromise did occur, the browser is isolated from the host machine. With the recurring destruction and restoration of the virtual machine to the pristine trusted state, the defenses are considerably improved and the bar significantly raised for detecting and preventing exploitation. This consistent, continuous destruction and restoration to the trusted state can help serve as a disruptive force in repelling the attacker.

7.9 Detecting the Attack

Invincea uses a detection engine to determine whether or not a given interaction with a website has resulted in a compromise of the guest browser. The goal is not to immediately block the exploit; rather it is to allow a normal interaction and exchange to occur, examining the resulting environment events. A positive security model is used with Invincea with it having a bird eyes view of the entire guest operating system environment. With this holistic view, behavior analysis of the http request and the OSI model interactions can be instantly reviewed and a decision made within seconds on whether a page is malicious and attempting to exploit the browser.

Figure 4 – Infection Notification from a Socially Engineered Malicious URL very similar to the IRS.gov web address.

7.10 Logging the Attack

If the engine detects a compromise, a notification to the user is provided and the software automatically initiates reverting back to the pristine trusted browser image and sends the attack activity to the centralized log server. Depending on the goals of the session, through configuration it is possible to allow the exploit to continue in the contained environment for observation or to immediately halt its execution and restore from a pristine image. These policies can be configured across an implementation instance or in select clients.

```
        <infection_restore_timer grace_period_seconds="300" />
-- More  --
```

Figure 5 – Configuration for Pristine Restore Once Detection Occurs

7.11 Keeping Legacy Applications Available

In many cases, custom web applications that are developed and managed by IT and part of the core software set may deviate from standards when they are created. This can result in the use of these applications triggering behavior based defense protections to kick-in and prevent them from running. While they may not meet the standard, they are trusted business applications that are required to conduct business. To allow users to be protected against threats, but still maintain the ability to interact with these custom business applications, Invincea can be configured to allow custom behaviors for these sites. Through configuration, it is possible to allow for interactions with trusted organization applications. A typical use for this configuration is the case where a legacy application that is using a code base that doesn't comply with standards, but is known to be safe, is implemented on an organization's intranet. Only trusted sites for interactions can occur in the trusted zone, with the regex being used to implement rules to put in place. All other untrusted sites will maintain their status in the untrusted zone.

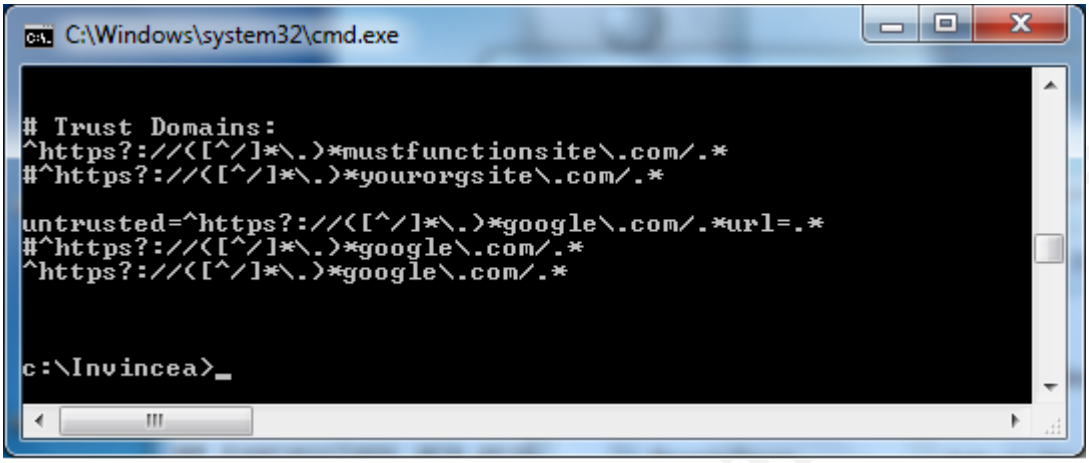

Figure 5 – Configuration, Known Safe Legacy Applications

7.12 Zero Day Attacks

Zero Day Attacks and exploit code remain a challenge for defenses. Although, with behavior based defenses, many Zero Day attacks can be detected and prevented as a positive security model is used. With a positive security model, one does not look for malicious code based on signatures and updates, rather the known validity and deviations from the norm are evaluated. Many Zero-Day attacks take advantage of obfuscation and polymorphic code to slip past radar defenses. With these Zero Days, nothing has been detected in the wild for the security community to build a defensive signature upon. With the positive security model in place, the chances of preventing a zero day from exploiting the client browser target improve.

7.13 Persistent Storage and Inspection

Due to the nature of virtualization and the periodic disposal of operating system state, persistent storage of data valuable to the user needs to be preserved. This is accomplished between a secure communication between the isolated browser and hardened services on the local host. Files which are downloaded during a browser session with Invincea can be saved to dedicated storage outside the guest and heavily

interrogated and inspected to ensure they are free from malware. This directory should be configured to be non-executable. Also for environments which require additional security, through host scripting these files can be immediately swept to an isolated network for client access with only visual display capabilities for review and editing. A solution such as Citrix would accomplish this type of functionality. This would provide additional security when bringing content outside the guest virtual machine environment, such as transferring an Excel or Word document for editing. Blocking downloads in entirety, can be accomplished through configuration as well.

```
erences.xml | findstr downloads
        <log_downloads enabled="true" />
        <block_exe_downloads enabled="false" user_modifiable="false" />
        <remove_all_downloads_on_restore enabled="false" user_modifiable="false"
/>
        <remove_downloads_on_detect enabled="false" />
```

Figure 6 – Configuration, Guest Download Policies

7.14 Reviewing the Tapes

All web pages which trigger the Invincea behavior engine and fall outside of the normal interaction with a typical webpage are alerted to the end user and then immediately the guest virtualization machine is restored to original pristine, trusted state. While knowing that a potential exploitation of the browser has just been prevented is good, the ability to better understand what the exploit has just delivered is important to improving defenses. Learning the exploits methods for potential compromise is a key to gaining insight into the attack and potential classification of the attacker. To obtain this information, the Invincea client log integration can be configured to push down logs to a centralized sys log server in real time, allowing for deeper analysis. Invincea Threat Data Server is also another log repository and server side application which can be configured to serve as a sys log repository. The Threat Analyzer client provides a graphical interface to review and analyze aggregate attacks which have been triggered across client sets.

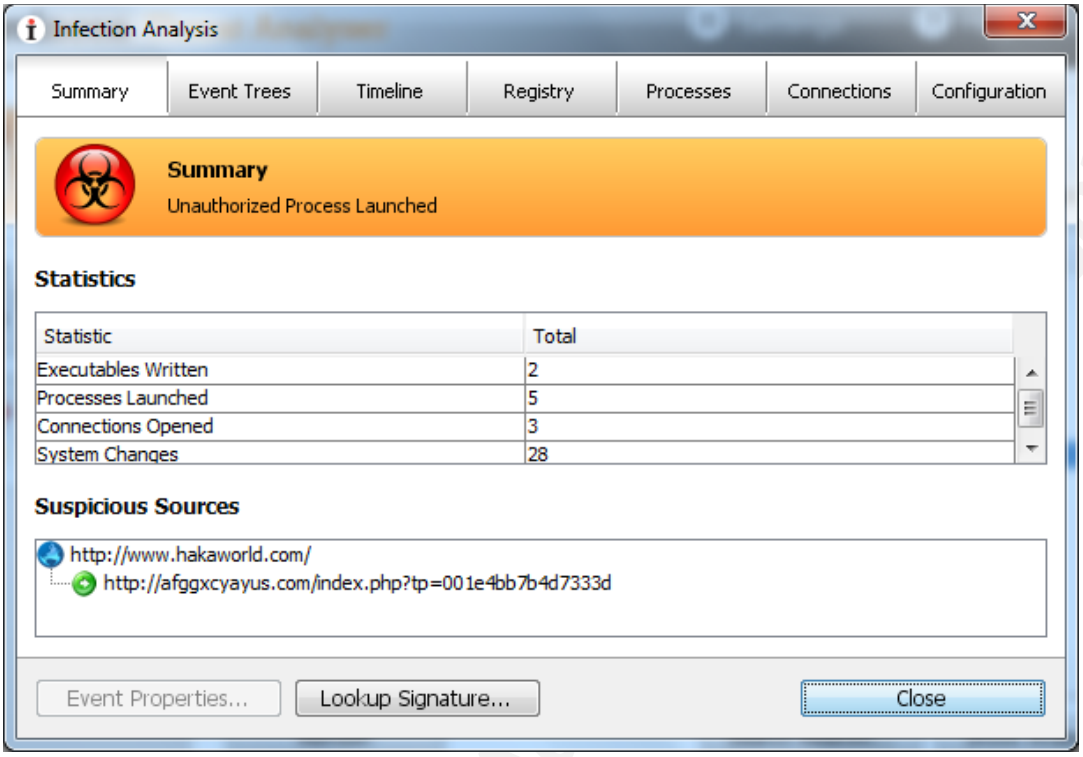

Figure 7 – Threat Center Client, Attack Summary

The composition of the log includes information detailing the attacker's payload, resources attempted to be accessed, files written during a session, processes launched, registry keys modified, network connections initiated and changes attempting to modify the system. This information can prove to be a treasure trove for defense.

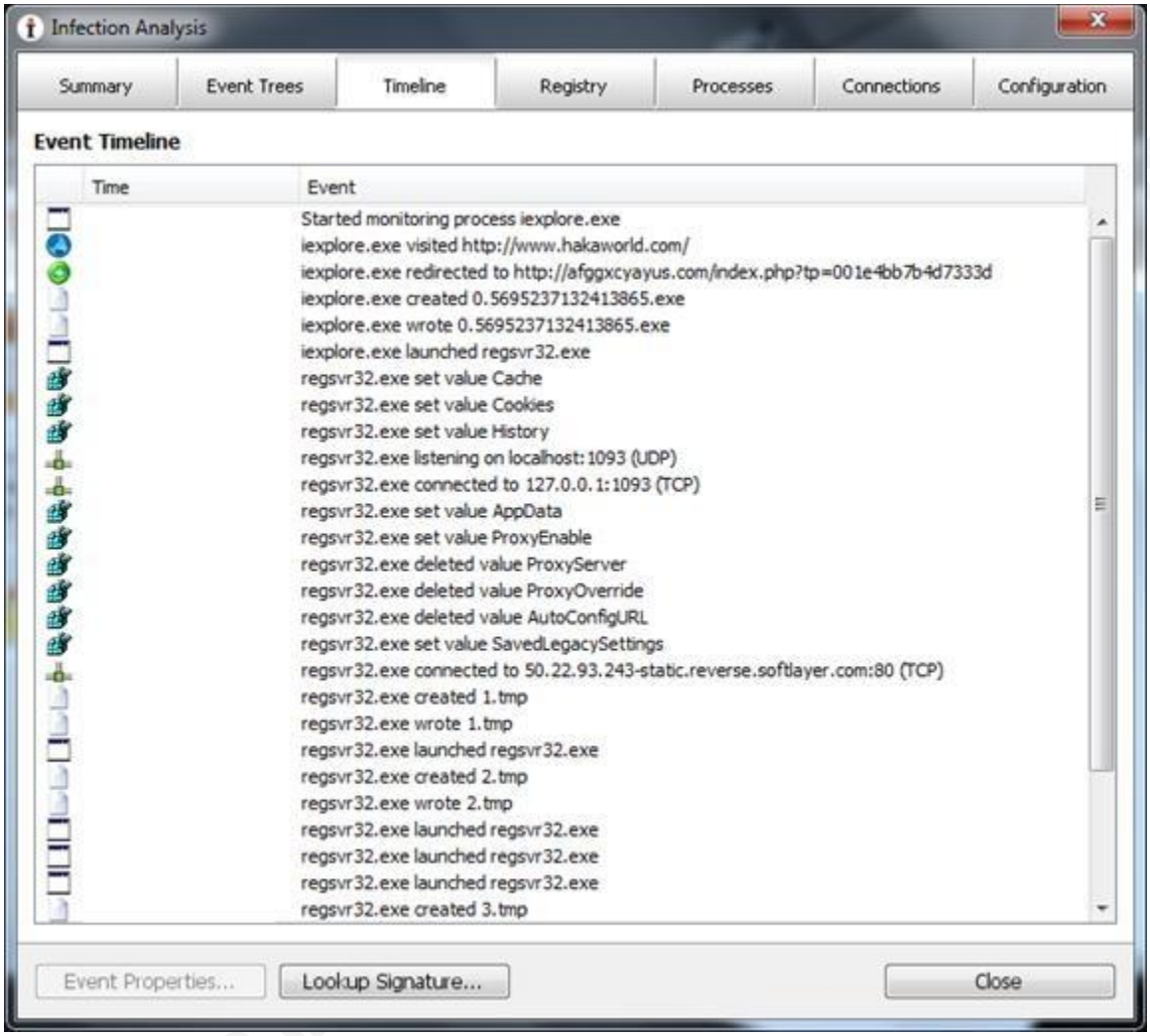

Figure 8 – Threat Center Client, Attack Sequence / Timeline

7.15 Tailored and Custom Invincea Browsers

In order to have the easiest integration and to minimize training gaps in your organization, configuration options do exist to tailor the guest Invincea browser to the desired organization spec. If a nonstandard plug-in or browser helper object is required for a critical line of business application, the browser can be customized to include these libraries. This allows for the full experience without downgrading the capabilities of the web application. Policies and various organization specific configurations can also be controlled through Active Directory, allowing for custom configuration of browser settings. Tailored custom browsers can also be implemented to provide different core sets

of software that accompanies the virtual browser, such as add-ons like NoScript and RequestPolicy, Perspectives, HTTPS Everywhere, etc. Attacks which don't necessarily aim to take over the client browser but rather to steal information persisted on the client side can be very difficult to defend against. By tailoring the Virtualized Browser Client with additional add-ons, and educating users on how to properly use them, one can achieve further levels of protection.

8. Web "Drive-By" - Attempted Exploitation Walkthrough

8.1 A Chance Brush with Malware

Next, we'll examine an end-to-end interaction that takes place when a user visits a website that puts them in direct contact with malware that attempts to exploit the client machine via the browser.

There are various ways that the end user could come into contact with the malware outlined below here, and we will take a look at the scenario where a malicious link is embedded within an iframe of a compromised website. This has been a well-documented method and an effective technique for attackers to exploit and infect client machines (Shinn, 2009). The iframe tag in html allows content from one website to be loaded into the page of another website. When navigating to what appears to be an innocuous site, the existing page content is downloaded and the page appears just as it would normally. Behind the scenes, an additional maliciously planted iframe tag has been embedded into the normal html code. This allows the attacker's malware to download with the regular content and exploit the client machine. This technique was used recently in the compromise of the MySQL.com website, to distribute malware and infect any users which simply navigated to the site (Holwerda, 2011). The only action a user would need to perform in this scenario is to simply visit a website to read an article, and his or her machine would then become infected with malware.

8.2 Isolated Browser Session

A user browsing the web with the Invincea browser, benefits from running their web browser session isolated from the host. When a site is encountered which attempts to load malware, behavior based defenses detect the deviations from the baseline and a notification is displayed on-screen, indicating to the user that a possible infection was detected.

8.3 Defenses Triggered

The following malicious site was retrieved from a malware repository and visited while running the isolated Invincea browser. While the site loads and appears as it normally would to the end user, additional malicious activity behind the scenes is occurring without visible notice.

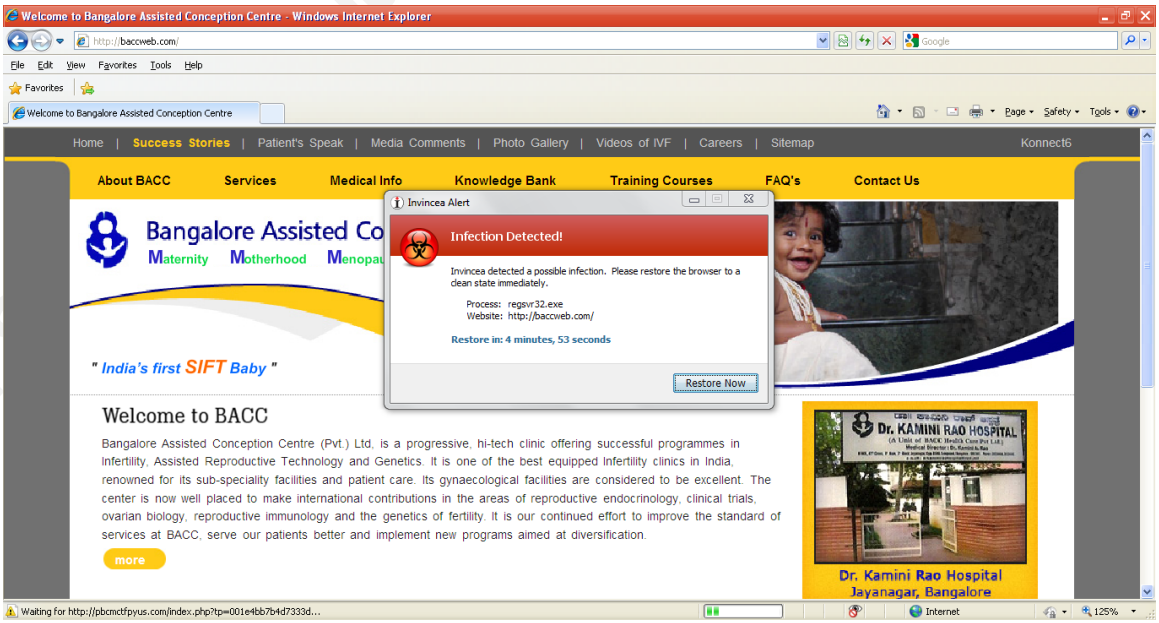

Figure 9 – Infected Website Is Visited

Upon visiting the infected site, an alert is generated once the threshold is crossed in determining what a normal site should perform when interacting with a browser session.

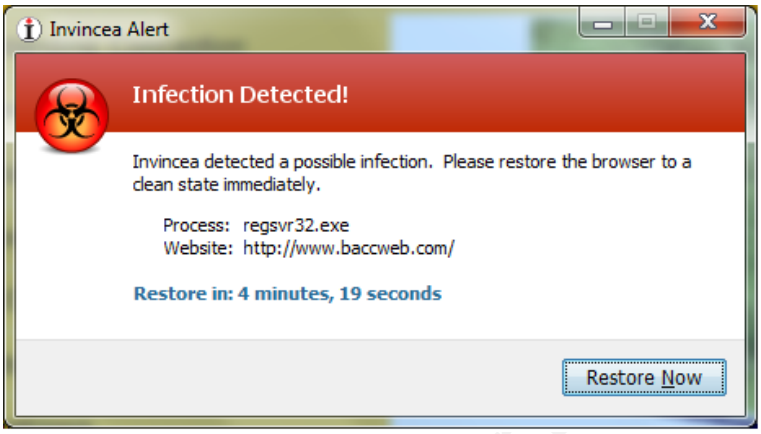

Figure 10 – Defensives Step In, End User Alerted

8.4 Returning to a Healthy State

The alert indicates a possible infection has been detected and instructs the end user to restore their browser session to a clean state immediately. The configurable window before a browser session 'self-destruct' and restore is set to five minutes in this case. This allows for a graceful shutdown by the end user. The configurable and preferred option to 'restore now' would end the interaction immediately and return the browser to the pristine state. At this point the user can throw away the infected browser session which was isolated from their host machine, and start anew with a pristine browser session.

Figure 11 – Restoration to a Healthy State

For the purposes of this walk through, the infected browser session was configured to run and to not immediately halt any infections. This allowed the malware to run for a period of time to gain forensic insight into the exploit. After a delay to allow for the malware execution and the subsequent downloads it attempted, the browser session was ended.

Despite stepping on this "virtual tripwire" by visiting this site that was rigged with malware, normal browsing activity was able to resume once the pristine browser image was restored.

8.5 Reviewing Log Details

To better understand the attempted exploitation and to modify defenses upstream that were not able to detect this malware, a review of the attack can be performed. The attack can be captured and logged to an aggregation server such as syslog for review with any forensic tool of choice. For the purpose of this walkthrough, Invincea Threat Analyzer was used to view the logged attack activity. The Threat Analyzer interface provides details on the attack that was captured and logged. A cursory examination of the forensic detail that is available when an exploit is detected through the Invincea Threat Analyzer will be outlined in the following section.

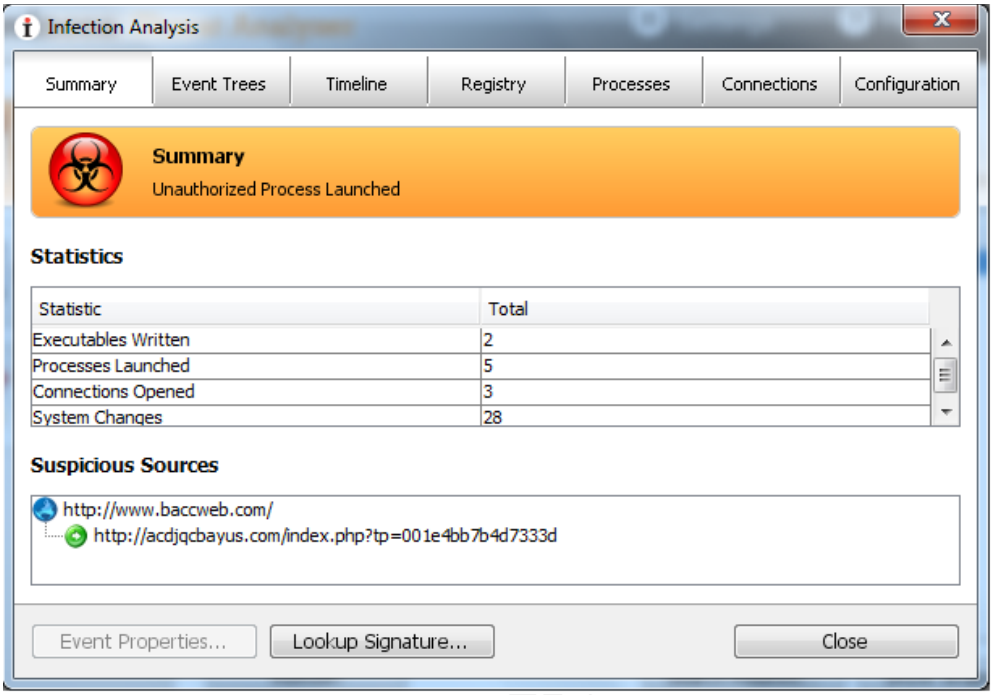

Figure 12– Forensic Summary Detail of Attack

8.5.1 Summary Information

In the Summary tab of the Threat Server Analyzer client, an overview is provided to give high level detail of the exploit which executed in the isolated browser environment. We can see the defense engine identifies the offending source address and subsequent suspect sources that the malware communicated with. Full summary details such as the total number of software executables that were written to the host, subsequent processes launched by the malware, as well as overall system changes are identified. Specific connections that were setup during the malware execution are also available for review. In the link visited during this walkthrough, it is clear from the summary information that a barrage of malicious activity was kicked-off by the malware.

8.5.2 Pin-pointing the Exploit Code

In reviewing the attack summary in the Invincea Threat Analyzer, one can see that the visited site was modified to include a malicious script(s) that would load additional content, compromising the visiting user's browser and host machine. This is evident in the source of the initial page visited, as well as in additional scripts which are loaded from separate files contained in the Scripts directory.

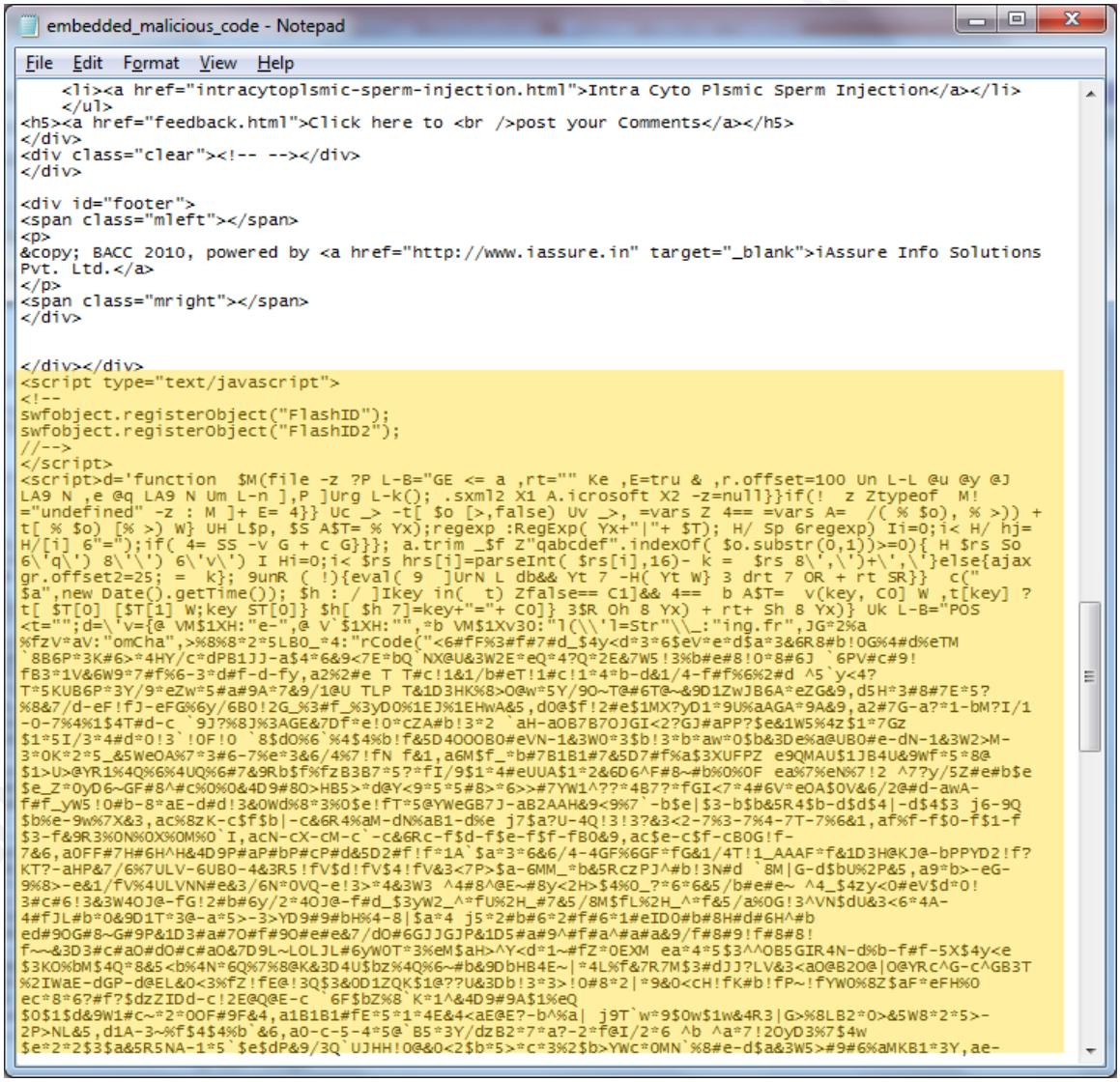

Figure 13 – Obfuscated Malicious Script

To avoid detection from defenses, the attackers embedded script(s) were obfuscated.

8.5.3 Signature Based Detection

To gain a better understanding of where this exploit would register against general signature based detection, given the available data on this malware, a submission of the malicious link is uploaded to Virus Total. Virus Total is a community virus and malware analysis site that performs varying levels of static analysis on uploaded samples. When uploading the malicious link for static signature analyses, a detection rate of nearly thirteen percent is returned.

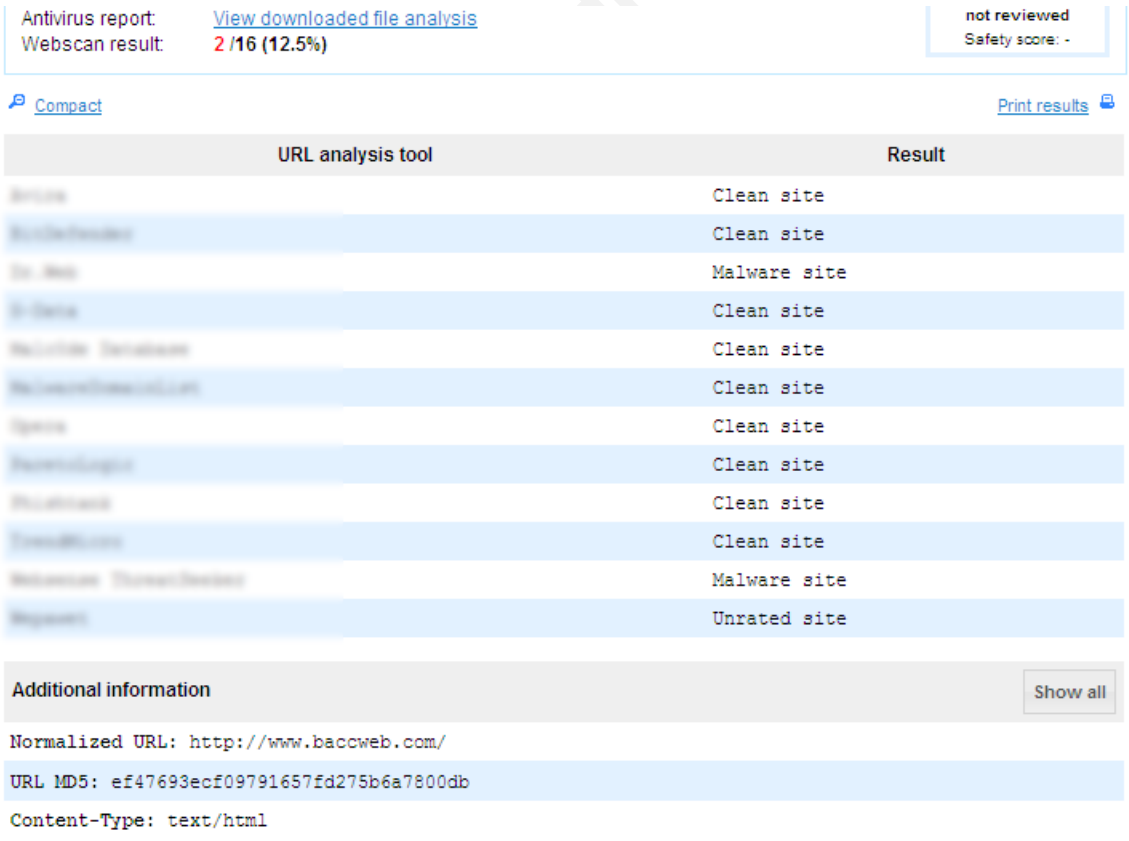

Figure 14 – Virus Total Detection Statistics

The success rate of static signature analysis in this instance is limited - having triggered a twelve and half percent detection rate from the available static signature based anti-virus engines. Each vendor's anti-virus may be configured differently in the cloud based Virus Total service, and will have varying rates of effectiveness against a variety of virus and malware specimens. Although in this particular malware sample, detection through signatures was limited.

8.5.4 Reputation Based Detection

After reviewing the suspect link against Virus Total to gain a sense of the effectiveness of signature based detection, several popular website reputation lists were checked to see if reputation based blocking would have advised or warned the individual browsing, to not proceed to the malicious site. Google Safe Browsing, AVG, and Norton Safe reputation services that track malicious sites also had no record yet of this site on their blacklists at the time the malicious link was visited. While this type of detection and reputation based blacklisting is a day to day moving target, an entry for this site was unfortunately was not yet reported and available to the community at large. This would have flagged the site as untrusted and suspicious, warning other users browsing the web to not proceed with visiting the malicious link.

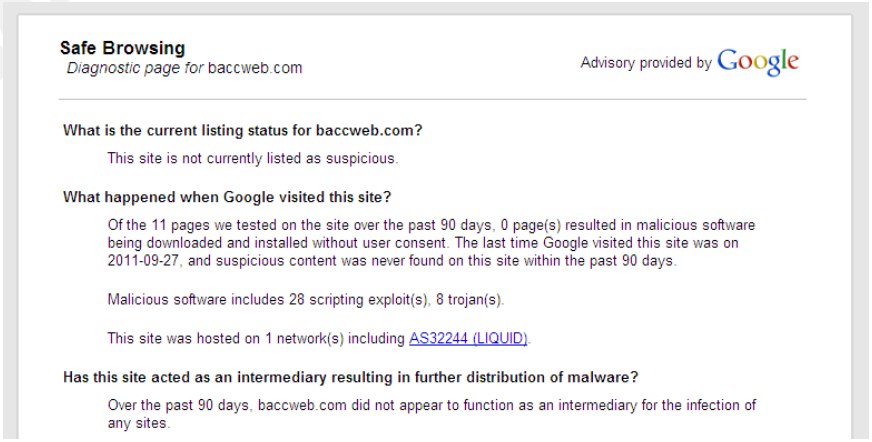

Figure 15 – Google Safe Browsing Result

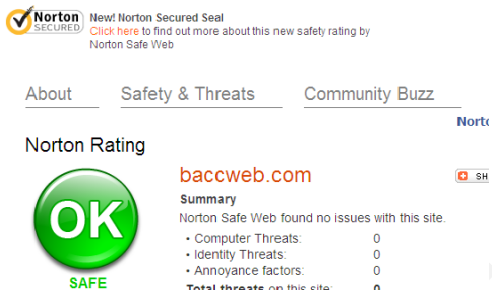

Figure 16 – Norton Safe Web Result

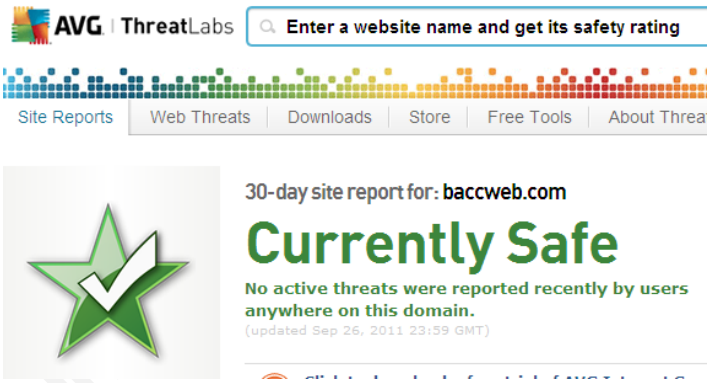

Figure 17 – AVG Site Report Result

8.5.5 Understanding the Exploit Details to Improve Defenses

After gaining a sense of some of the other protections that may not have caught this malware due to its Zero day value or dynamic nature, we can review the additional details of the attack that was detected through behavior based defenses.

When the initial malware site is loaded in the browser, the embedded malicious JavaScript is executed and a request is made to an additional server to deliver the exploit payload.

Joseph Faust, josephfaust.dc@gmail.com

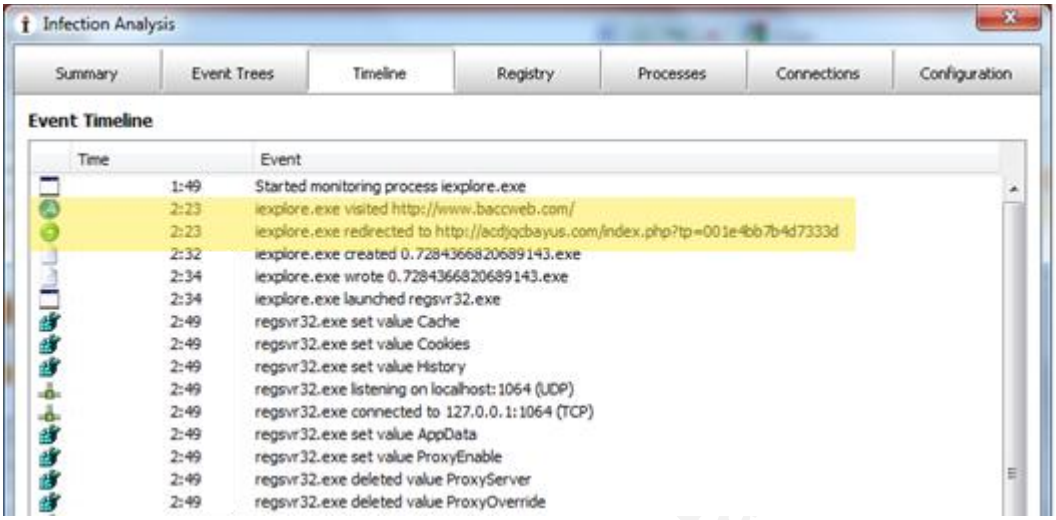

Figure 18 - Malware Redirection for Exploit Payload

From the exploit delivered, we can see several binaries are created. Regsvr32.exe is executed and subsequent changes to the browser cache, cookies and history are observed.

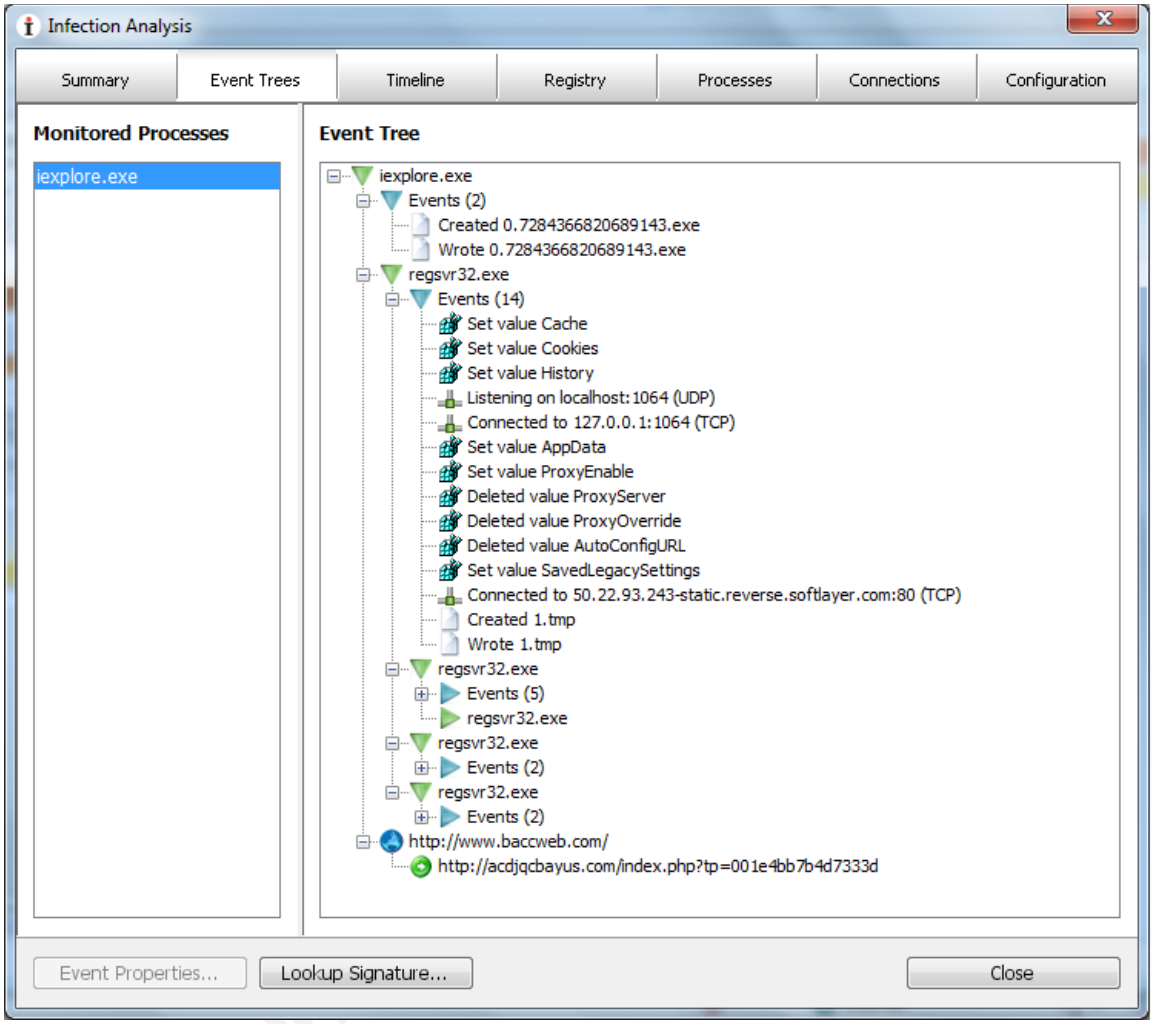

Figure 19 - Suspect Process Spawn Malicious Events

Looking at the Event Tree in the Invincea Threat Analyzer tab, the list of monitored processes that were captured during the exploit session is displayed with the underlying event sequence for each process. One can see a flurry of interesting events that are generated and which ultimately lead to the creation of a backdoor listener over UDP to create a communication channel. The attack goes on to make communication back with the control server.

Joseph Faust, josephfaust.dc@gmail.com

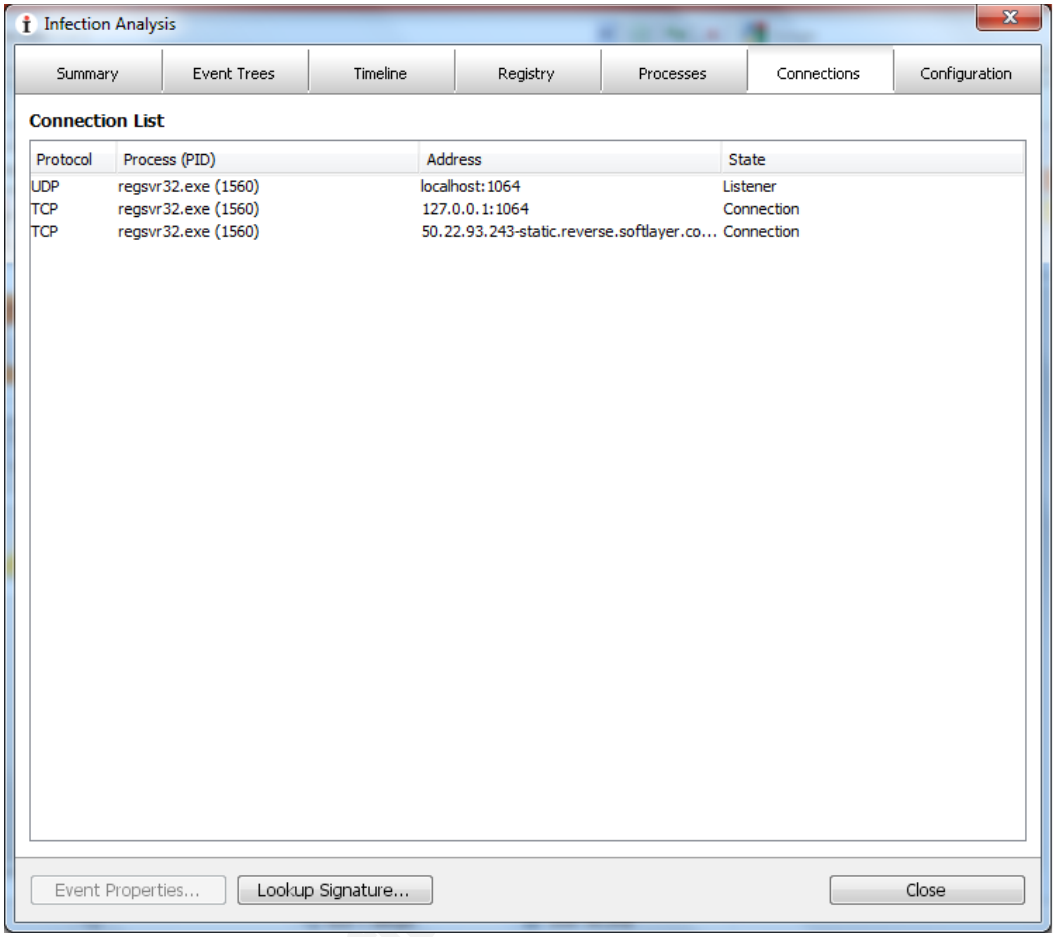

Figure 20 - Malware Initiates Outbound Connections

The Connection Tab of Threat Analyzer shows the connections that the malware has setup in creating a proxy and listener as a command and control channel.

8.5.6 Learning and Improving Defenses

The forensic information gleaned from the attempted exploitation can be filtered back into the overall organizational security defenses. Defenses that exist upstream could benefit greatly from the forensic information captured, and potentially thwart this exploit at the perimeter in the future. In the Firewall, entries could be made to block the domain,

as well in other web filters to shut down any requests to this site. Limiting execution rights of certain types of files would also be applicable. User education would also be prudent. Raising awareness of the specific type of attacks the organization is experiencing would ultimately help increase the ability to provide a more targeted defense.

In this walk through of an isolated browser session with behavior based defenses enabled, the malware was detected and the end user notified of the potential infection. The browser was subsequently restored to a pristine state and browsing activity could resume. Forensic details of the attack were recorded to a central log server for which further examination could occur with additional malware analysis tools. By restoring the machine to the pristine trusted state, normal web browsing was able to resume with minimal disruption, and a possibly damaging and costly breach for an organization is averted.

9. Takeaways and Items to Consider

Behavior Based Browser Defenses and Hardware Virtualization allows for great flexibility in a security architecture and design. It is important to also consider the concepts and possible constraints that may apply when moving to such a model.

9.1 False Positives & Negatives

When deploying the solution en-masse, it is important to tune the configuration accordingly for legacy applications and specific line of business applications. Without tuning the configuration, these legacy applications may otherwise produce false positives. This in many instances for legacy applications is due to the nature of how these programs may have been coded. False positives represent known valid websites which have triggered defenses to halt and shutdown the browser session. Overall, this can be a

minimal disruption, as sites which trigger the behavior engine to block can be reviewed for design issues if they are internal applications. In the event that no change can be made to bring the application up to a standard, then a configuration rule can be put in place to allow interaction with these sites. When applying any security solution, we must account for the fact that certain events may occur which can evade system detection or that are not accounted for in the model, known in this context as a false negative. To quote Donald Rumsfeld, "There are known unknowns. [..] There are things that we know we don't know". Some attacks may make it past a given control – other controls need to be in place to layer the defense and detection accordingly.

9.2 Positive Security Model

With Invincea, a positive security model is implemented. This allows only normal web traffic and web interactions to occur. Web traffic that deviates from the normal behavior is detected and halted, and a pristine trusted state browser session is restored. Through this model, one can study the interactions that a given piece of software should perform and work to evaluate subsequent activities that fall outside of the norm. Modeling after good activity and detecting deviations provides the strongest defenses and improved resistance to attacks.

9.3 Impact on Level of Effort

Through adding Behavior Based Browser Detection and Hardware Virtualization, the chance of successful attack is reduced. Increasing the amount of time required to successfully pull off a given attack can help raise the investment required by the attacker, as well as the overall level of effort needed to successfully exploit a system through a browser session.

9.4 Defense in Depth

The increase in difficulty is not only represented and measured in time, but in security layers as well. In this solution the number of controls which must be compromised compared to a traditional browsing solution is increased. Stacking the controls such as Intrusion Detection Systems / Prevention Systems, Network Intrusion Detection, Endpoint Host Intrusion Detection, Anti-virus, Hardware Virtualization, and Behavior Based Defenses helps appropriately layer security and spread detection and prevention duties across the security controls.

9.5 Limited Attack Window

The defense model with Behavior Based Defense, coupled with Hardware Virtualization can mitigate the effects of an attack. In the case of an exploit that is not detected by the guest and host sensors, the virtualized browser session can be configured for de-construction and restoration from the trusted state on the hour. This limits the information that can be gained by reducing the available window of opportunity for the attacker to work against a target without starting the attack anew.

9.6 Attacks That Don't Exploit the Browser

Attacks that don't attempt to exploit the browser, but still steal important information and credentials from the user are items that users need to consider. Examples of such style of an attack include session hijacking, where an attacker is able to trick the end user into revealing a session id through cross-site scripting. Another example that doesn't attack the browser itself, rather the end user, would be tricking an individual into entering his account credentials into a replica of a banking site. Present technology has its limits in helping users against this particular type of attack (Jakobsson & Meyers, 2006). Ultimately educating users and raising awareness of potential threats in this area is

critical for an effective defense. Organizations must continue evaluate tools and adopt accompanying technologies that will help provide stronger defenses in attacks against the individual human operator.

9.7 Ease of Use

For defenses to be raised in implementing hardware virtualization and behavior based detection /prevention, the software that is put in place must be easy enough for a common user to work. If the secure solution is too complex for an end user to reasonably use and accomplish their job, the end user will work around the technology and ultimately weaken the security posture of the organization. By striving for seamless integration and use in a solution, the technology underpinning the process should be an abstraction to the end user, and work largely just as the original browser software does.

9.8 Escapes

In this solution, defenses are strongly predicated on malware that is able to be contained through hardware virtualization. While practical escapes of the hypervisor may exist, in the wild such escapes have been limited and so far to date, quickly patched to shut down the ability to thwart the hypervisor (2009, Kortchinsky). To limit exposure, systems must be kept patched and up to date, as well as limit the services and ports on the host machine to reduce the attack surface. In this solution, the Invincea Browser software monitors the health of the virtual machine from both within the guest, as well the host. This provides oversight to the general health of the virtual machine guest and any attempts to thwart the hypervisor. As advances and variety in escape techniques increase, monitoring the host for a potential escape will need to be more aggressively considered and addressed through additional desktop controls (NSA, 2010).

10. Conclusion

The Internet has become an integral part of many organizations business. With the adoption of the browser as the ubiquitous interface to web, exploits that target the organization through this attack vector have followed suit in attacking the browser. Advances in technology have allowed for increased efficiency for attackers to exploit targets. Automation and efficiency on the defense mechanisms must evolve as well. Fewer security decisions need be left in the hands of the end user to attempt to guess the correct answer when confronted with a suspicious site or email link.

Isolating the client browser through hardware virtualization and behavior based defense defenses can greatly help limit exposure to malware and help prevent providing the attackers a beach head into a victim's network through exploiting the browser. Data indicates that attacks against the browser continue to accelerate in adoption and practical use, and that this trend will continue for some time. (2011, Symantec)

Through continued end user education and implementation of Behavior Based Defense and Hardware Virtualization to isolate and protect the end user, organizations in the Corporate, Government and Non-Profit sectors can significantly raise the bar of their defenses against web based client side attacks.

Joseph Faust, josephfaust.dc@gmail.com

References

Alperovitch, D. (2011, August 2) *Revealed: Operation Shady RAT.* McAfee. Retrieved from: http://tinyurl.com/3zdyvkv

Avolio, F.(2011). Firewalls and internet security. *The Internet Protocol Journal*, *2*. Retrieved from http://tinyurl.com/644st5r

Barth, Jackson, Reis, Google Inc. (2009, September 29). *The Security Architecture of the Chromium Browser*. Retrieved from http://tinyurl.com/38cb3tm

Barwinski, M., & Irvine C., & Levin T., (2006). In L. Armistead (Ed.), *Proceedings of the International Conference on Internet Warfare and Security* (*ICIW*), University of Maryland Eastern Shore, USA, March 15-16 2006 Reading, UK: Academic Conferences Limited.

Bellis, M. (June 15 2010). *The History of the Internet.* Retrieved from http://tinyurl.com/yfljtjs

CA. (2006, January 10). Best practices: protecting against the growing threat of phishing scams. *Hacker Journals*. Retrieved from http://tinyurl.com/6k4lh3x

Cyveilliance. *(March 2011). Cyber Intelligence Report.* Retrieved from http://tinyurl.com/64qoyhp

Federal Computer Weekly. (August 2011). *Concerns grow as sophisticated attacks, automation evade detection*. Retrieved from http://tinyurl.com/6yar8a9

Federal Computer Weekly. (2011). *Evolving cyber threats demand coordinated defense.* Retrieved from http://tinyurl.com/64czodq

Goodchild, J. (2008, November 3). *Symantec threat report: three ways internet crime has changed*. CSO Magazine, Retrieved from http://tinyurl.com/5txrq79

Goodin, D. (20011, August 8) *Mass WordPress hijack poisons Google Image well*. The

Register. Retrieved from http://tinyurl.com/3qqc6wc

Gross, M. (2011, August 2). *Operation Shady rat—Unprecedented Cyber-espionage Campaign and Intellectual-Property Bonanza*. Vanity Fair. Retrieved from: http://tinyurl.com/3qy8p4z

Grossman, J. (2008, August 27). *Website Security Statistics*. Retrieved from http://tinyurl.com/6fa6dot

Guerra, P. (2009). *How Economics and Information Security Affects Cyber Crime and What It Means in the Context of a Global Recession*. Proceedings of the Black Hat 2009. Retrieved from http://tinyurl.com/6a4b3fm

Harri, H. (2010, February 22). *The Evolution to the Next Generation Firewall*. Retrieved from http://tinyurl.com/6zeq95p

Holwerda, T. (2011, September 26). *MySQL.com hacked to server malware*. OSNEWS, Retrieved from: http://tinyurl.com/625la7k

Invincea. (2011, September 1), *About Invincea*. Retrieved from https://www.invincea.com/company/about/

Jacobsson M., S. Meyers. (2007). *Phishing and Countermeasures: Understanding the Increasing Problem of Identity Theft.* Hoboken: Wiley-Interscience.

Jakobsson & Ramzan (2008). Crimeware: understanding new attacks and defenses. Boston: Pearson Education.

Joch, A (2011, August 5). *What you need to know about the latest cyber foes*. Federal Computer Weekly. Retrieved from http://tinyurl.com/69hbsdj

Kortchinsky, K (2009, June 2) Cloudburst. *Proceedings of the Black Hat 2009*. Retrieved from http://tinyurl.com/lcghml

Joseph Faust, josephfaust.dc@gmail.com

Messmer, E. (2011, January 18). *Mpack, neosploit and zeus top most notorious web attack toolkit list*. Network World Magazine. Retrieved from http://tinyurl.com/5uxgzjf.

MiniWatts Marketing Group. (2011). Internet World Stats - "*Internet Users of the World*". Retrieved from: http://tinyurl.com/65npgn5

M86 (2010 April 27). *Security Lab Report. Web Exploits: There's an App for That* Retrieved from http://tinyurl.com/6ykesw2

NSA. (2010, October 1). *NSA's first Trusted Computing Conference and Exposition* Retrieved from http://tinyurl.com/48fz9n3.

Shiner, D (2004, August 5). *Web Browser Vulnerabilities: Is Safe Surfing Possible?* Retrieved from http://tinyurl.com/3jbbs

Shinn, M (2009, August 25) *Anatomy of an Advanced Persistent Threat.* Retrieved from http://tinyurl.com/68orlju

Symantec (2011). *Symantec Symantec Corp., Internet Security Threat Report.* Retrieved from: http://www.symantec.com/business/threatreport/

Joseph Faust, josephfaust.dc@gmail.com