Interested in learning more about security?

SANS Institute InfoSec Reading Room

Preventing Incidents with a Hardened Web Browser

There is substantial industry documentation on web browser security because the web browser is currently a frequently used vector of attack. This paper investigates current literature discussing the threats present in today's environment as well as weaknesses of the browser, including PKI and plugins. To help the organization harden the browsers deployed in its environment, comparative studies of browser security ratings are reviewed and hardening suggestions for Internet Explorer and Firefox, the most popular b...

Preventing Incidents with a hardened web browser

GIAC (GCIH) Gold Certification

Author: Christopher Crowley, crowleigh@gmail.com
Advisor: Stephen Mathezer

Accepted: November 17th 2009

Abstract

There is substantial industry documentation on web browser security because the web browser is currently a frequently used vector of attack. This paper investigates current literature discussing the threats present in today's environment as well as weaknesses of the browser, including PKI and plugins. To help the organization harden the browsers deployed in its environment, comparative studies of browser security ratings are reviewed and hardening suggestions for Internet Explorer and Firefox, the most popular browsers, are provided. Security Enhancements in Internet Explorer 8 are also identified. Options for browsing the internet via a bastion host are explored as an advanced method for enhancing security. Based on current literature it is safe to say no web browser is truly secure. By adapting and implementing these suggestions most organizations will be more resistant to compromise.

Chris Crowley, crowleigh@gmail.com

1. Introduction

Incident Handling follows a six phase cycle: Preparation, Identification, Containment, Eradication, Remediation, Lessons Learned. This document is firmly grounded in the Preparation phase. It sets out to prevent incidents from happening by configuring web browsers securely.

A web browser is a ubiquitous application for client access to resources. The dominance of Hyper Text Markup Language (HTML) as a medium for delivery of internet content to individuals is nearly unchallenged. Most organizations provide a web browser for users to connect directly to resources outside of the organization's control. The web browser is one of the few applications specifically intended for the purpose of connecting to arbitrary sites. The EU anti-trust case against Microsoft regarding Internet Explorer states, "The Web is a distributed system that supports a massive collection of digital information resources stored throughout the Internet in documents called "Web Pages". Users of client PCs can access Web pages and display them by means of applications called "Web browsers"." (EU, 2004). According to this commission decision, a web browser is really the only application available for users of personal computers to access Web pages.

Web browsers are designed to access arbitrary resources, whereas other applications can access Internet resources but their primary function is something different. Media Players are designed to process local and network sound or video files. The media player is spawned by the browser to handle specific file types and is not the application used to locate available resources. Think of common types of applications: Office Suites are mostly intended for processing of local resources. Games are either local only, or network centric, but the network centric games typically connect to a specific server to coordinate communication among participants. Chat clients mostly connect to specific servers to make user connections. The messages are typically routed through these servers. E-mail clients connect to a small number of servers, usually only one, and this server is often within the LAN.

It is unrealistic that an organization would expect to have any sort of collaborative security agreement with all of the web sites that users can contact, leaving the organization to harden the web browser against the malicious content that users will encounter. A review of current security literature turns up a large number of attacks specifically against web browsers and associated features (Alme, 2009).

The prevalence of web browsers means that securing the browser presents a common challenge

for security teams. The intent of this paper is to investigate security features of some of the most common web browsers available today. The paper represents an effort to provide general recommendations applicable to all types of web browsers as well as guidance for specific browsers where possible.

This general guidance must be adjusted depending on the requirements of a given environment. Further, supporting policy documents are important to educate and provide expectations for users. In short, rules of behavior and user responsibilities are a contract between the organization and its users. This document presupposes the existence of these documents in the organization. If these do not exist in your organization, jump start their creation by looking at SANS Policy Project (SANS, 2009). For example, Employee Internet Use Monitoring and Filtering Policy provides a sample policy which identifies expectations and affords legal protection for the company from allegations of improper monitoring (Bong, 2007).

Further, this guidance is intended as a loss prevention measure. Ultimately the security practitioner must justify the expenditure of resources to develop hardened browsers to prevent incidents. Cenzic reports that a single data loss breach can cost an organization $500,000 or more, not including reputation damage. (Cenzic, 2009) The Verizon 2009 Data Breach Report identified web browsing in seven (7) of the thirty four (34) malware related data breaches out of ninety (90) total data breach incidents studied in the report. The web browser incident was a foothold toward a larger impact data breach. (Baker, 2009)

One strategy for estimating the single loss expectancy (SLE) of a browser compromise is the cost in lost productivity for your organization to have a user workstation out of service for one day, and the cost of one IT technician to spend one day re-imaging and reinstalling that workstation. The challenge of quantifying loss is the subject of much debate. (Bejtlich, 2009) Having a reasonable and justified figure of the cost the organization will incur allows the security practitioner to make an informed recommendation for organization appropriate countermeasures.

A presumption of this paper is that centralized management, consistent configuration, and reducing user based security decisions (such as allowing a user to decide to trust content) enhances security. User awareness is an important component of computer security (Pruitt-Mentle, 2008). The implementation of user awareness and security training is beyond the scope of this paper, however many resources can easily be found with Google, such as the NIST document: Building an Information

Technology Security Awareness and Training Program. (Wilson, 2003)

1.1 Intended Audience

Based on the assumption that Security staff will be responsible for configuration decisions for the organization, the reader of this paper is assumed to be a Network or Security engineer who manages systems for an enterprise and provides these systems to users.

An independent user functioning as administrator of his own system should be able to adapt the advice provided herein. An excellent reference for browser specific configuration is the US-CERT browser resource (http://www.us-cert.gov/reading_room/securing_browser/).

2. Main Section

2.1 Threat Environment

Let's look at the threats the web browser may encounter. There are several major categories of threats to web browsers which exist in the wild.

Unauthorized disclosure of content stored on the system running the web browser is a threat which must be addressed. One example is Apple Security Update 2009-001. 2009-001 fixed a flaw in Safari which allowed access to files on the local hard drive due to execution of arbitrary Javascript in the local security zone (Mastenbrook, 2009).

The browser is capable of retrieving content, so directly downloading malicious programs is clearly a threat. In fact, Trend Micro 2008 threat data indicates that more than half of the most common infections were due to direct download of malware from the internet (Cruz, 2008).

Similar, but more pernicious is the threat of accessing a known, legitimate site, but to have content delivered from another party which is malicious. Direct compromise of the hosting server and inclusion of malicious content is one method of accomplishing this. Or attackers can potentially utilize the web server's application flaws to inject references to malware. Currently a great deal of malicious content is delivered via a number of vectors, including cross site scripting (XSS) or cross form scripting (XFS), as well as SQL Injection. Any of these content injection methods can be used to deliver iFRAMEs or Javascript pointing to malicious code (White Hat Security, 2009).

Another class of threats is deceiving the browser's security mechanisms to afford greater trust to a site than the user had authorized. A classic example of this is unicode domain names. The browsers

render a domain name to the user which appears legitimate, but is in fact a malicious duplicate. (Dorman, 2005)

Finally, the browsers themselves have vulnerabilities. Firefox 3.5.4 was released to address memory corruption issues. Mozilla advisory 2009-064 states, "Some of these crashes showed evidence of memory corruption under certain circumstances and we presume that with enough effort at least some of these could be exploited to run arbitrary code." (Vukicevik, 2009)

2.2 Plugins

Browsers often depend on plug-in applications to deliver an additional capability, for example enhanced media functions such as a video. There exists the possibility of exploiting the transfer of data and control from the browser to the plug-in. For example, CVE-2007-6242 was the result of the Adobe Flash plugin trusting data contained in embedded JPG files, and allocating memory based on that information. (Portony, 2007)

A web browser plugin is a utility or program that is not considered part of the original web browser application, but provides some additional functionality to the web browser via a sanctioned application program interface (API) provided by the browser (Koziol, 2004, p. 106). Browser providers enable APIs to facilitate community development. Add on browser capabilities are additional vectors of attack requiring protection.

Delivery of dynamic, high quality content often depends on browser plugins and helper applications, for example, Adobe's Flash player. Flash is nearly ubiquitous (there are versions for Solaris, Linux, OS X, and Windows) and had four security related updates between January and October 2009 (Adobe, 2009). The web browser is typically configured to seamlessly start appropriate applications. For example, the Adobe browser plugin automatically configures Internet Explorer to open downloaded PDF files in Acrobat reader. (Adobe, 2008). An attacker uses this to exploit a vulnerability in the helper application, Acrobat in this case, by embedding a malicious document in a web page. This increase of attack surface makes the browser difficult, yet important to protect.

It is not just Adobe Flash and Acrobat. Other potentially vulnerable plugins and helper applications include other media players: Windows Media Player, Quicktime, and Real Player. There are also language processors specifically intended to process code delivered from an external server including: ActiveX, Javascript, and Java. (Maone, 2009)

Firefox has a substantial and active community of addon developers. Add-ons in Firefox

parlance are components that become part of the browser's functionality. The Firefox Add-on site (https://addons.mozilla.org/en-US/firefox) had roughly eleven thousand (11,000) downloads available at the time this paper was written. The Mozilla Foundation provides the infrastructure within Firefox to install and update the plugins, as well as the ability to universally disable a plugin. It does not however take responsibility for the content or activity of the addons themselves. "Mozilla is providing links to these applications as a courtesy, and makes no representations regarding the applications or any information related there to. Any questions, complaints or claims regarding the applications must be directed to the appropriate software vendor." (Mozilla Foundation, 2009)

Internet Explorer has this same functionality, but the added functional units are called plugins. This unsupported framework is called spicIE and is available for IE7 and IE8. "Important: SpiceIE is released independently of Internet Explorer and related products. It is not an officially supported product by the Internet Explorer team. While SpicIE is a valuable tool for prototyping and testing Internet Explorer extensions, it is not recommended for use with released software."(Microsoft, 2009)

2.3 SSL and Trust

There is a global web of trust that is established for the benefit of providing confidentiality and integrity of web browser traffic. There is substantial cost involved in maintaining this trust. Verisign's 2009 SEC filing sites $300M USD for 6 months ending June 2009 (Verisign, 2009). This trust system utilizes public key cryptography and disseminates certificates to vouch for the identity of the server the web browser connects to. PKI certificates can also vouch for the identity of the user of the web browser (Verisign, 2009). A critical component of the trust model is that the certificates of Certification Authorities are distributed with the web browser.

The Certification Authorities pay to include their root certificates in the web browser. Individual businesses pay the certification authorities for the certificates which vouch for the identity of their servers. The Certification Authorities vet the certificate requester to verify the requester is actually part of the organization represented by the proposed certificate(Harris, 2005, p. 653).

This trust model is subject to exploitation, for example by social engineering and race conditions. In one high profile incident an unauthorized group secured web server certificates within the microsoft.com domain. "According to Microsoft, someone posing as a Microsoft employee tricked VeriSign, which hands out so-called digital signatures, into issuing the two certificates in the software giant's name on Jan. 30 and Jan. 31." (Lemos, 2001)

Chris Crowley, crowleigh@gmail.com

More recently, attacks against Verisign's RapidSSL service exploited collisions in md5 hashing utilized by some SSL certificates. The Sotirov et al presentation *MD5 Considered Harmful Today: Creating a Rogue CA Certificate* details the man in the middle attack on certificates signed with MD5 hashing algorithm. For the presentation thirty thousand (30,000) SSL web server certificates were downloaded from the internet. Nine thousand (9,000) of these certs were using MD5. (Sotirov, 2008)

A race condition, specifically "time of check / time of use" exists when revocation lists are out of date. (MITRE, 2009) If a certificate is known to be compromised, the certification authority publishes revocation of the certificate. Often however, the time between the compromise and delivery of that information to the browser is susceptible to exploit. A proposed fix for this problem, a system of real time certificate verification called Online Certificate Status Protocol (OCSP) is not widely implemented. This is similar to DNSSEC – an available security extension to the Domain Name Services (DNS) protocol not widely utilized because the underlying infrastructure has not achieved widespread adoption. (Mimoso, 2009) Browsers warn users if a certificate revocation list (CRL) is not available. While these warnings are sufficient gestures to transfer culpability to the user of the browser, users simply dismiss them to allow the access the site, accepting the risk that the certificate is not in fact valid.

Each web browser provides the user the capability to configure the trusted Certification Authorities (CA). This is required, but subject to abuse. In an enterprise environment, changes to the CAs should be prohibited to standard users. Individual users managing their own systems should be cautious when adding additional CAs. The web browser affords trust to CAs in the CA store as authorities on the identity of other organizations. The enterprise should manage this trust granting power to prevent fraud, abuse and malicious activity. It is analogous to checking the identification of an individual before hiring him. Only trained Human Resources staff are entrusted by the organization with the authority to verify identity. Likewise, informed IT staff need to manage the organization's CA repository.

An additional complication of SSL is that it renders the security capabilities of network inspection (IDS / IPS) blind to attacks delivered within the SSL stream. In some environments, this must be addressed by terminating outbound SSL connections at a proxy for content inspection. This typically involves configuration on the client computer running the web browser to facilitate termination of the SSL connection at the proxy. For example, the BlueCoat proxy configuration

requires a new Certification Authority be installed on the client web browser, or assigning a certificate to the proxy server signed by an already present CA. (Bluecoat, 2007)

2.4 Web Browsers

2.4.1 Popularity

The most commonly used browser worldwide as of September 2009 is Internet Explorer with about 70% of the browser market. The other browsers in order of most to least used: Firefox (about 20%), Safari, Chrome and Opera have less than 5% each. (Hitslink, 2009). The numbers are rough estimates, and other sites provide other numbers. To focus this document, I will only consider the top 2 web browsers: Internet Explorer and Firefox. There are a number of other potentially useful and potentially vulnerable web browsers available: Safari, Chrome, Opera, Konqueror, Lynx, wget, Songbird. Unfortunately they are out of scope for this paper. Configuration guidance provided for the more common browsers should help guide the secure configuration of those browsers not covered.

The popularity of Internet Explorer may have negative security implications. As attackers leverage the law of averages and target the largest population. On the other hand, Internet Explorer's maturity and Microsoft security initiatives have earned Internet Explorer fewer security vulnerabilities than Firefox. (Jones, 2007).

2.4.2 Web Browser configuration reference

There are multiple online reference points for advice on secure browser configuration. The previously mentioned US-CERT is the best overview with browser specific suggestions (http://www.us-cert.gov/reading_room/securing_browser/).

2.4.3 Is there a safe Browser?

We know the browsers are vulnerable to exploit and that there are multiple attack vectors to get the browser to load attacks to potentially vulnerable helpers or plugins. We know the environment may contain hostile code, from even supposedly trustworthy sources. We also know that the PKI mechanism of trust used throughout the internet is vulnerable to exploitation. The question remains, which web browser is the most secure? The answer is, it depends. There is no browser that is clearly more secure than another. Each browser has security flaws. During a time when there is an unpatched Internet Explorer vulnerability and active exploits in the wild, Firefox would clearly be a more secure browser option. For an organization that patches frequently, the more frequent and automatic updates

of Firefox may be preferable. For an organization that uses a Microsoft based patching services such as WSUS, Internet Explorer is more likely to be maintained with the most up to date patches.

2.4.4 Internet Explorer

The best source for Internet Explorer 7 configuration is the Federal Desktop Core Configuration (FDCC) specifications. These specifications provide a full life-cycle of configuration, including specifications and content for auditing conformance to these settings.

The US Office of Management and Budget (OMB) recently commissioned National Institute of Standards and Technology (NIST) to create and curate FDCC. (Evans, 2008) The intent of FDCC is to work with suppliers to produce a standardized configuration for the purchase of software resources. One benefit of this consistency is to reduce the cost of purchasing computing equipment by the US Federal Government. Another benefit is the reduction of cost of securely configuring these resources. Previous to FDCC, each Agency independently analyzed, reviewed, and proposed specific workstation configurations. OMB also requires that software purchased by US Agencies not require deviation from the standard secure configuration.

To realize this cost reduction and standard secure configuration OMB chartered NIST to develop a method of automated testing for the required FDCC configuration. The Security Content Automation Protocol (SCAP) will likely be a more lasting contribution to computer security than the FDCC content. SCAP framework provides a lexicon to express configuration, and use the same content to audit implementation. (Mell, 2009)

Thus, FDCC requires vendors to conform to a standard configuration, and prove conformance by standardized audit content. SCAP also defines a validation program which encourages vendors to comply with SCAP by including the requirement in OMB procurement specifications. Part 39 of the Federal Acquisition Register (FAR) now reads "In acquiring information technology, agencies shall include the appropriate IT security policies and requirements, including use of common security configurations available from the NIST's website at http://checklists.nist.gov. Agency contracting officers should consult with the requiring official to ensure the appropriate standards are incorporated." (Evans, 2008)

The scope of FDCC and SCAP are constrained by resources and politics. Based on the abundance of desktop PCs running Microsoft Windows operating systems in US Agencies, most of the NIST produced FDCC content is specific to Microsoft OSes. OS settings for several Unix variants are

also available. For web browsers, only Internet Explorer security configurations are provided. (NVD, 2009)

Using the FDCC content provides a number of benefits. The primary benefit is the ability to implement and audit the implementation using the same, standardized specification.

How to use the FDCC content for Internet Explorer 7 via GPO? Start by downloading the GPO .adm templates from NIST (http://nvd.nist.gov/fdcc/download_fdcc.cfm). Check the checksum listed on that page against your download to assure the file has not been corrupted or tampered with in transit. If you don't already have a utility to calculate checksum and are using a windows system, download Microsoft's File Checksum Integrity Verifier from http://support.microsoft.com/kb/841290 . This is a simple command line utility which calculates md5 and sha1 hashes of files. NIST does not provide MD5 checksums on its site because MD5 is not FIPS 140-2 compliant.

Once verified, unzip the file. It is organized into directories for Vista and XP. Select the appropriate directory, or use the "Both" directory. The Internet Explorer specific configurations are located in the "inetres.adm" files. For XP, these files are:

- FDCC v1.0 Q1 2009 Revised GPO's/XP/{51419B81-0B58-4005-990B-00D23627B453}/DomainSysvol/GPO/Adm/inetres.adm
- FDCC v1.0 Q1 2009 Revised GPO's/XP/{1ACE0B04-3907-4A1C-B12B-1A1761E74AD5}/DomainSysvol/GPO/Adm/inetres.adm
- FDCC v1.0 Q1 2009 Revised GPO's/XP/{60552F64-F962-4393-8392-887D7948A9A5}/DomainSysvol/GPO/Adm/inetres.adm

It is very important to test these settings before deploying them to a production environment as they are very strict. For example, the settings (ex. HKCU\Software\Policies\Microsoft\Internet Explorer\Main!FormSuggest Passwords, HKCU\Software\Policies\Microsoft\Internet Explorer\Control Panel!FormSuggest Passwords) to disable auto-complete for all usernames and passwords may not be acceptable in your environment.

Test these settings before deploying them to a production environment! Document deviations from the standard, and the rationale for the deviations. Anticipate that for some settings, individual use cases may require an exception. Document these exceptions. In this documentation include expiration dates, rationale, authorized individuals, and who authorized the exception. The preferable fashion for handling exceptions is on a granular basis. For example, assume your organization disables the option

to remember passwords. However, an individual with a disability has management approval to allow the browser to remember passwords. Enable the option to remember passwords for this user but not any of the other configuration settings. This is done by maintaining multiple containers within the active directory (AD) hierarchy with varying GPOs.

Deploy a subset of the settings to a limited user group initially. The ideal pilot user group includes sophisticated, technology savvy, power users. Engage a sampling of individuals from all parts of the organization, because usage patterns are different for different groups. By deploying a subset of the settings the root cause of problems can be isolated quickly. Allow at least one full week to elapse between deployments to flush out potential problems. If problems are found, consider excluding the settings as either deviations or exceptions. Repeat the deployment of subsets with a larger group. Depending on the size of your organization, this may take multiple passes. As the iteration recurs, increasing the number of settings in the subset and the number of users is reasonable. (Microsoft, 2003)

FDCC settings have the added benefit of allowing for the auditing of content. There are multiple products which are SCAP validated FDCC scanners, which means they have "The capability to audit and assess a target system to determine its compliance with the FDCC requirements." (NVD, 2009) This affords the IT department the ability to audit the deployment of the mandated settings. After deployment, this should be done on a regular basis, usually in conjunction with the periodic vulnerability scanning. Or a SCAP validated Misconfiguration Remediation product such as BigFix Security Configuration and Vulnerability Management Suite (SCVM), could check daily and correct inappropriate settings. (NVD, 2009)

NSS Labs performed a comparison of the built in reputation systems of the five most popular web browsers. The concept of a reputation system is to leverage Internet wide intelligence to prevent web browsers from connecting to known bad sites. If half of the malware is directly downloaded from sites as the previously cited Trendmicro study claims, then arming the web browser with intelligence to avoid bad sites is a reasonable defensive action. A honeyclient approach was taken, connecting the browsers to know bad sites. The results were that the Internet Explorer 8 RC1 browser had a protection rating of 69% (Internet Explorer 7 had only 4%), and Firefox v3.07 had a protection rating of 30%. (NSS Labs, 2009)!

The reputation system isn't a fully effective resolution. The best reputation type browser (IE 8) had a success rate of only 70%.

Chris Crowley, crowleigh@gmail.com

Internet Explorer 8 has some security enhancements that update it for the current threat environment. There is the SmartScreen filter which is what was lauded by the NSS Labs white paper. There is a cross site scripting (XSS) filter which helps to prevent inclusion of the targeted site by a frame. To help users discern the actual site contacted, the domain is presented in contrasting text from the hostname and URI (the rest of the URL). (Microsoft, 2009)

There is additional memory protection called Data Execution Protection (DEP). This functions in conjunction with the OS. In summary, the kernel monitors programs access to memory. In the event that a non-executable memory page is executed by a program, the kernel halts execution of that program. This is great, but time will tell if a lot of people will turn this off once it interferes with a needed capability, or interferes with a legacy application in the organization. (Microsoft, 2009)

The "InPrivate Browsing" feature enables reducing storage of browsing history information. According to Microsoft's website this is a benefit for browsing on untrusted or public computers. Of course, if the environment is untrusted it is probably best not to use the computer anyway. For kiosks and shared workspaces this is better than having to clear all the cookies and browser history manually. There is also a new privacy capability called "InPrivate Filtering." This allows customization of the access to data that sites use to correlate personal information. (Microsoft, 2009)

More information for specific Internet Explorer 8 configuration is available at http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=44405777-51b4-4376-9cef-f0341b13fcde

It is best to begin work to update to Internet Explorer 8 to take advantage of the new protections afforded by this browser.

2.4.5 Firefox

The 2007 Honeynet project study "Know Your Enemy: Malicious Web Servers" states that there were a higher number of vulnerabilities in Firefox than in Internet Explorer, but that there were fewer compromises as a result of these vulnerabilities when browsing known malicious websites. There was no conclusion derived as to why this is, since it depends partially on the decision by the attacker to target a specific platform. The inference drawn from the honey client data is very interesting. It is that attackers have a higher success rate of compromise for Internet Explorer because its user population tends to be deficient in patches. (Riden, 2007)

The Jones paper cited earlier, " Browser Vulnerability Analysis of Internet Explorer and

Firefox" corroborates the Honeynet project's study finding that Firefox had more *vulnerabilities* than Internet Explorer. The Jones's paper and the NSS paper didn't cite the success rate of malicious web sites in *exploitation* of the vulnerabilities present. The implication is that browsing with Firefox has a lower likelihood of resulting in a compromise than browsing with Internet Explorer.

Firefox lacks the enterprise level administration capabilities of Active Directory configuration of Internet Explorer relevant registry settings. Also lacking is the substantial investment in infrastructure by an organization like NIST to provide validation programs for its configuration, "Currently, US government SCAP content is primarily focused on Windows operating systems." (NVD, 2009) Note that Internet Explorer is part of the Windows operating system.

So, vendor and community support is what is available to assist in hardening Firefox. Fortunately, there is a good deal of this support available.

The first thing to do for Firefox in an enterprise setting is to manage specific user settings. There is a method for locking specific user preferences. It involves defining a set of key / value pairs, encoding the file and setting it in the appropriate location with restrictive permissions to prohibit a user from modifying the settings. (Ilias, 2005)

Use the US-Cert.gov settings for Firefox. These recommendations will be defined in a fashion that cannot be changed by individual users. Per Chris Ilias's instructions the first step is to create a text file, we'll call it lockPref.txt with key – value pairs of the settings to constrain.

```
//
// begin lockPref.txt - cc
//ask where to save downloads – US-Cert.gov
lockPref("browser.download.useDownloadDir",false);
// remember forms filled – US-Cert.gov
lockPref("browser.formfill.enable", false);
// ask for cookies – US-Cert.gov
lockPref("network.cookie.lifetimePolicy", 1);
// warn for addon install – US-Cert.gov
lockPref("xpinstall.whitelist.required", true);
// disable java – US-Cert.gov
lockPref("security.enable_java," false);
```

Chris Crowley, crowleigh@gmail.com

```
// manage media file types
// automatically install application updates
lockPref("app.update.autoInstallEnabled", true);
```

These settings will be locked, and the user cannot change them. It is extremely important to note that locking settings in Firefox depends on system permissions which prohibit the user from changing the installed configuration files. The end of this section (after the NoScript locked settings) includes details of these permissions.

US-Cert's site suggests using NoScript for granular sitewise configuration because NoScript provides default deny posture for active content in Firefox. It is an excellent plugin but has an Achilles heel: the end user is required to authorize content. For an enterprise wide deployment, the user training required to effectively deploy NoScript in Firefox could be enough to prohibit a new deployment. If Firefox is already available in your environment, serious consideration should be given to deploying NoScript along with it. Based on an e-mail from Giorgio Maone, the author and maintainer of NoScript, there will likely be an enterprise version of NoScript available sometime in 2010 (Maone, personal communication, November 10, 2009).

NoScript blocks Javascript, Flash, Java, and other plugin content by default. The user must elect to trust a site, or in some cases specific content. For example, clicking on link which opens a PDF would result in a page with a large NoScript blocked element. Click the element, allow it, and the browser launches the PDF viewer of the system. It is simple, intuitive, and extremely effective. NoScript also provides Cross Site Scripting protection by blocking inclusion of code from a site into a NoScript trusted site. (Maone, 2009)

There are a number of NoScript's options which should not be changed by a regular user. "Options – General" and selecting "Scripts Globally Allowed (dangerous)" is a good example. These should be added to our developing lockPrefs file.

```
// noscript specific lockPref
// from http://hackademix.net/2007/12/05/plugin-security-plug-insecurity/#comment-2209
// you'd better copy the site list from the
//   "capability.policy.maonoscript.sites"
//   key in the prefs.js file found in a test profile
```

Chris Crowley, crowleigh@gmail.com

```
    lockPref("noscript.default", "informaction.com
http://informaction.com https://informaction.com flashgot.net http://
flashgot.net https://flashgot.net noscript.net http://noscript.net
https://noscript.net");
    // hide context menu
    lockPref("noscript.ctxMenu", false);
    // hide statusIcon
    lockPref("noscript.statusIcon", false);
    // hide notification bar
    lockPref("noscript.notify", false);
    // disable DOM Inspector and Error Console
    //   (which may be used to programmatically unlock the prefs)
    lockPref("noscript.lockPrivilegedUI");
    // disable global scripts
    lockPref("noscript.showGlobal", false);
    // end lockPref 2009-11-06 cc
```

See Appendix A for an easy copy / paste content of this file. This file is then encoded using a byte shift. For this paper it was done using moz-byteshift.pl, but can be accomplished by a number of other methods. (Knaff, 2005).

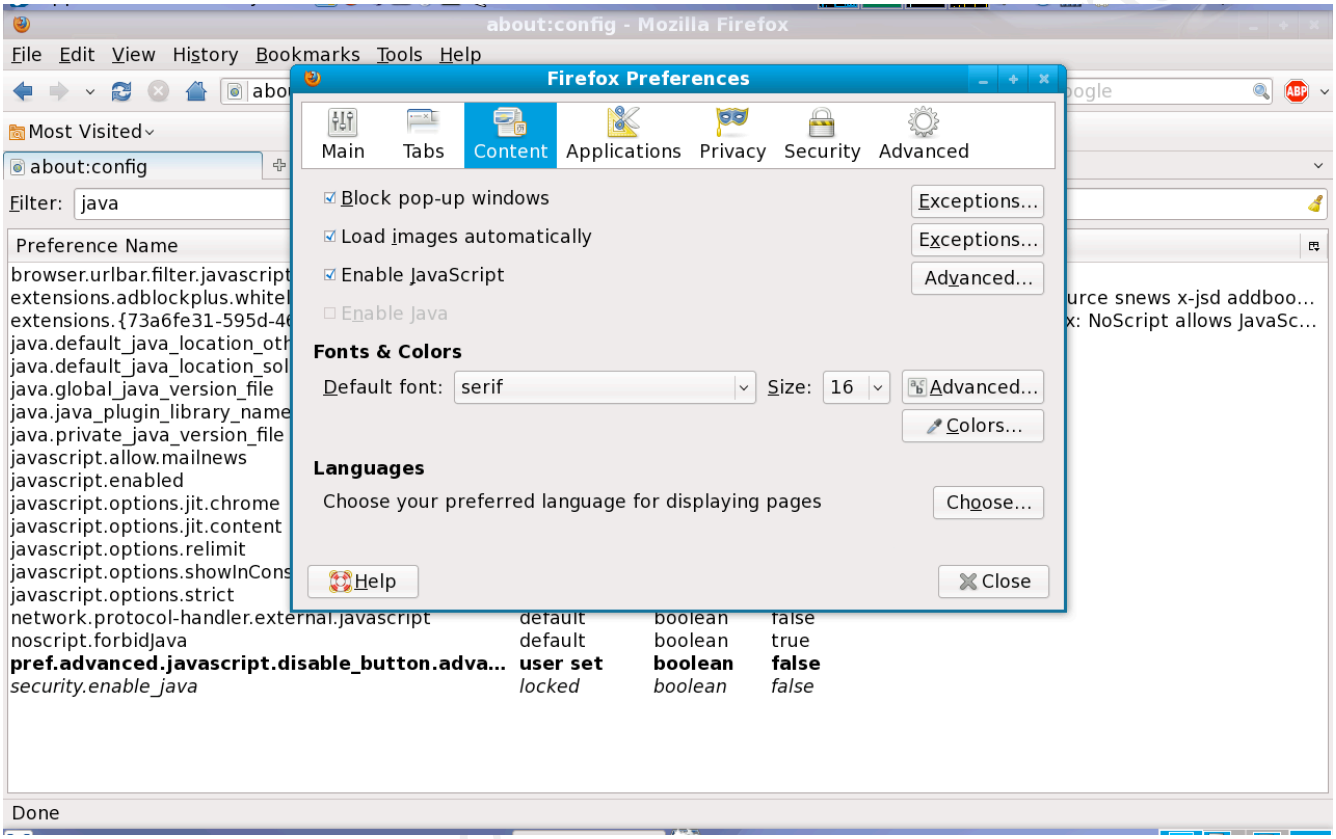
```
    $ ./moz-byteshift.pl -s -13 < lockPref.txt > mozilla.cfg
```

This is not actually a requirement, in order not to use the encoding specification these change would be required in the all.js file.

```
    // pref("general.config.obscure_value", 13);
    // for plain text
    pref("general.config.obscure_value", 0);
```

The resulting rotated file (or plain text version) is copied to the Mozilla directory which for Windows is "C:\Program Files\mozilla.org\Mozilla\defaults\pref\". The location on Linux varies, but usually is "/usr/lib/firefox-*version*/". In Linux this file is dropped in the top of the Firefox directory, not in the greprefs sub-directory. It is included by adding the following line to the all.js file in the greprefs sub-directory:

```
pref("general.config.filename", "mozilla.cfg");
```



Mozilla with Java locked in a disabled state

It is extremely important to prohibit changes to all.js and mozilla.cfg by restricting write permissions to only administrative users and groups. On a windows system, this is done by selecting file properties, and marking the file read only. On a domain joined system, the administrator group should be given write permission, and everyone granted read only permissions. On a linux system, the permission should be root ownership, with file permissions rw-r--r-- (644).

The testing and deployment scenarios for Firefox should be as extensive as that described in the Internet Explorer section above. An organization approved statement of what settings are required is first, since the FDCC standards were already defined that for Internet Explorer but no authoritative document exists for Firefox. A method for authorizing, implementing, and tracking exceptions to the organization wide standards is required as was the case for Internet Explorer. Implementing this may be possible via GPO, but would require scripting to produce a suitable logon script that copies the appropriate mozilla.cfg into place.

Chris Crowley, crowleigh@gmail.com

So we have hardened the configuration settings for the browser. How else can we isolate the browser from impacting our systems?

2.5 Bastion Browsing

2.5.1 The Next Level in Protection

"The adoption of the web browser as the medium to provide application access and Internet communication has completely permeated the enterprise computing environment. " (Bluecoat, 2009) Often a web proxy provides a technical control point where content passing through the network is subject to inspection but browsers with Internet access are maintained on a relatively large number of systems. The typical business justification for this configuration is that users must connect to resources outside of the organization to accomplish core duties. Of course, personal productivity is also a requirement since a work / life balance tends to depend on ongoing access to Internet sites.

The question each organization must consider is whether every workstation in the organization actually needs access to external resources. If the actual business requirement is access to content available on the Internet, I propose that all systems do not individually need access. Rather, users need a way to review the content. This is in keeping with the concept of least privilege – accomplish the intent, but nothing more.

This scenario is similar to the "remote access" situation that most environments currently support for external users accessing enterprise LAN resources. One option to minimize exposure of workstation computers is to utilize a remote access scenario for user Internet access. Particularly in environments requiring higher security, this scenario presents a method for enabling users with access to Internet resources, but minimizing exposure of internal systems.

One problem with this scenario is the delivery of download content to the user workstation. For example if an individual would like to download a PDF file, a share (or some method of access) from the bastion host would be required for the user to access that downloaded file.

The benefit of this configuration is an "air gap" between enterprise workstations and the Internet. An additional benefit is in addressing the SSL inspection issue. The configuration creates a single termination point of SSL traffic. This means that all other encrypted traffic on the workstation portion of the network may be suspect.

2.5.2 Current Commercial Bastion methods

There are multiple available commercial solutions which provide this capability. HP,

Symantec, and Mozilla provide a "Mozilla Firefox for HP Virtual Solution." This is essentially a chrooted Firefox which runs within Software Virtualization Solution (SVS). This utilizes the individual workstation as the host running the browser. (Symantec, 2009)

Another example, Citrix Access Gateway (CAG) could be configured to provide this type of access. The current intent for CAG is to provide access to corporate resources from outside the corporate network. But, there is no reason that the configuration of network spaces could not be inverted (from the standard perspective of CAG implementation) to allow access to the Internet from the internal corporate network. A full desktop environment for web browsing is probably overkill, as it would confuse users about which desktop they are currently operating. But, execution of the browser based SSL VPN session from the desktop to the CAG, and CAG execution of the browser which actually connects out affords a centralized place for the organization to manage and harden web browsers. The Citrix device would map a drive for each user who connects , this is also a capability of the CAG. The user workstation would map that same drive, and resources downloaded would be readily available. Implementation would include firewall blocks for all outbound HTTP and HTTPS (tcp ports 80 and 443 respectively) except from the CAG. (Citrix, 2009)

There are some problems with this. First, it is adapting a tool from its initial intended use. Surely Citrix would support the implementation, but it would take customization and development. That equates to time, which equates to money. Additional cost is the licensing this software from Citrix, as well as hardware to run it. Expenditure justification would obviously be required to sell the idea to decision makers in the organization.

Imagine, if you will, the elevator speech for this proposal. “The idea is to license CAG for all outbound connections.” The response might be, “But we don't have to pay anything to connect the workstations now, and we already own web browsers.”

The expense justification would entail a proposed reduction in incidents, and the associated reduction is cost for remediating those incidents. The costs and loss reduction would be environment specific. Loss expectancy and the cost of a data breach, previously mentioned in the introduction, are good figures for deciding what security measures should be taken. There may be less expensive approaches which accomplish the same protection.

2.5.3 Low Cost Bastion method

A low cost version of a bastion browser host would be a Linux server running Mozilla browsers

displayed on workstations via X windows. This server would also run a SMB server (or NFS for a Unix workstation environment) so that clients can map drives to the downloads directory. The mounts should be mapped with no executable permissions, and ClamAV or similar Linux based anti-virus should be employed to inspect the files being delivered.

This has the benefit of concentrating security policy into a single host, as in the CAG method discussed above. Extensive hardening, mandatory access control via SELinux for example, can be applied. (Red Hat, 2009)

2.5.4 Individual Workstation Bastion method

A simple way to decrease the likelihood of a host compromise is to provide a layer between the host and the Internet. An inexpensive way to replicate the SVS offering would be to establish a Virtual Box virtual machine on the host and run the web browser in it. (Sun Microsystems, 2009)

This is only recommended selectively within the enterprise environment, since it would be laborious to maintain and would require user training. However, laptops of high value targets and users with risky browsing behavior would be good candidates for this solution in the absence of a centralized browser proxy server. Provide a check point to revert the virtual system to a known good state after each web browsing session. If necessary, provide a drive map between the virtual system and the host system. If this is done, executable programs should be prohibited from executing on the host system from the mapped drive.

Train the user on how to start the virtual system and utilize a sophisticated host based firewall which prohibits outbound connections on TCP ports 80 and 443 for all applications, except Virtual Box.

3. Conclusions

In conclusion, the threat to web browsers is severe currently. Flaws in the browsers and flaws in browser plugins are numerous and of high impact. There is extensive documentation, government support, and industry utilities for configuring Internet Explorer from an enterprise perspective. Internet Explorer should be the most secure browser based on vulnerability data. However it is more likely to be successfully exploited and result in system compromise than Firefox according to one study. Firefox has some enterprise level lock-down capability and its security posture is substantially enhanced by the NoScript addon. Browser virtualization, be it directly on the workstation or via a

centralized service is likely to be the next step for enterprise security to protect its environment.

4. References

Adobe Corporation (2009). Security bulletins and advisories. Retrieved October 26, 2009, from http://www.adobe.com/support/security/#flashplayer

Adobe Corporation (2008). *Configure internet explorer or AOL to display PDF files.* Retrieved October 26, 2009, from http://kb2.adobe.com/cps/331/331025.html

Alme, Christoph (2009). *Web browsers: an emerging platform under attack.* Retrieved October 26, 2009, from http://newsroom.mcafee.com/images/10039/wp_webw_browsers_w_en.pdf

Baker, Wade H. (2009). *2009 Data breach investigations report.* Retrieved November 14, 2009, from http://www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf

Betjlich, Richard (2009). *How much to spend on digital security.* Retrieved November 14, 2009, from http://taosecurity.blogspot.com/2009/06/how-much-to-spend-on-digital-security.html

Blue Coat Systems, Inc (2007). *Solution brief: accelerating SSL applications across the WAN.* Retrieved October 26, 2009, from http://www.bluecoat.com/doc/800

Blue Coat Systems, Inc (2007). *Secure and accelerate web applications and public websistes: blue coat SG for reverse proxy.* Retrieved October 26, 2009 from http://www.bluecoat.com/doc/680

Bong, Kevin (2007). *Employee internet use monitoring and filtering policy*. Retrieved October 26, 2009, from http://www.sans.edu/resources/student_projects/200711_004.pdf

Citrix, Inc. (2009). *Access gateway advanced edition delivers innovative application security.* Retrieved November 14, 2009, from http://www.citrix.com/English/ps2/products/feature.asp?contentID=26143.

Commission of the European Communities (2004). *COMMISSION DECISION of 23.04.2004 relating to a proceeding under article 82 of the EC treaty (Case COMP/C-3/37-792 Microsoft).* Retrieved October 26, 2009, from http://ec.europa.eu/competition/antitrust/cases/decisions/37792/en.pdf

Cruz, Mackey (2008). *Most abused infection vector*. Retrieved October 26, 2009, from http://blog.trendmicro.com/most-abused-infection-vector/

Cenzic (2009). *Cenzic web application security trends report shows increase in hacker attacks on web sites exploiting faults in popular web browsers and software.* Retrieved November 14, 2009, from http://www.earthtimes.org/articles/show/cenzic-web-application-security-trends,1034968.shtml

Chris Crowley, crowleigh@gmail.com

Dorman, Will (2005). *Vulnerability note VU#273262 multiple web browsers vulnerable to spoofing via internationalized domain name support*. Retrieved October 26, 2009, from http://www.kb.cert.org/vuls/id/273262

Evans, Karen (2008). *M-08-22 Memorandum for the chief information officers*. Retrieved October 26, 2009, from http://www.whitehouse.gov/omb/memoranda/fy2008/m08-22.pdf

Harris, Shon (2005). *All in one CISSP exam guide*. Emeryville: McGraw-Hill/Osbourne

Hitslink, Inc (2009). *Market share for browsers, operating systems and search engines*. Retrieved October 26, 2009, from http://marketshare.hitslink.com/report.aspx?qprid=0

Ilias, Chris (2005). *Locking mozilla firefox settings*. Retrieved October 26, 2009, from http://ilias.ca/blog/2005/03/locking-mozilla-firefox-settings/

Jones, Jeffrey (2007). *Browser vulnerability analysis of internet explorer and firefox*. Retrieved October 26, 2009, from http://i.zdnet.com/blogs/ie-firefox-vuln-analysis.pdf

Knaff, Alain (2005). *Automatic mozilla configurator*. Retrieved October 26, 2009, from http://www.alain.knaff.lu/howto/MozillaCustomization/locked.html

Knaff, Alain (2005). *moz-byteshift.pl*. Retrieved October 26, 2009, from http://www.alain.knaff.lu/howto/MozillaCustomization/moz-byteshift.pl

Koziol, Doug (2004). *The shellcoder's handbook: discovering and exploiting security holes*. Indianapolis: Wiley Publishing, Inc.

Lemos, Robert (2001). *Microsoft warns of hijacked certificates*. Retrieved October 26, 2009, from http://news.cnet.com/2100-1001-254586.html&tag=tp_pr

Mastenbrook, Brian (2009). *Safari RSS vulnerability: what went wrong?*. Retrieved October 26, 2009, from http://brian.mastenbrook.net/display/28

Maone, Giorgio (2009). *NoScript – javascript/java/flash blocker for a safer firefox experience!*. Retrieved October 26, 2009, from http://noscript.net/

Mell, Peter (2009). *Security content automation protocol (SCAP) version 1.0 validation program test requirements (DRAFT)*. Gaithersburg: National Institute of Standards and Technology

Microsoft Corporation (2003). *Creating a pilot plan*. Retrieved November 6, 2009, from http://technet.microsoft.com/en-us/library/cc781659(WS.10).aspx

Microsoft Corporation (2009). *Internet explorer 8: features*. Retrieved October 26, 2009, from http://www.microsoft.com/windows/internet-explorer/features/safer.aspx?tab=1

Chris Crowley, crowleigh@gmail.com

Microsoft Corporation (2009). *Internet explorer 8 – data execution protection /nx*. Retrieved October 26, 2009, from http://msdn.microsoft.com/en-us/library/dd371730(VS.85).aspx

Mimoso, Michael (2009). *Kaminsky interview: DNSSEC addresses cross-organizational trust and security*. Retrieved October 26, 2009, from http://searchsecurity.techtarget.com/news/interview/0,289202,sid14_gci1360143,00.html

Mozilla Foundation (2009). *Add-ons for firefox*. Retrieved November 6, 2009, from https://addons.mozilla.org/en-US/firefox/

The MITRE Corporation (2009). *CWE-362: Race condition*. Retrieved October 26, 2009, from http://cwe.mitre.org/data/definitions/362.html

NSS Labs, Inc (2009). *Web browser security test - socially engineered malware protection comparative test results*. Retrieved October 26, 2009, from http://nsslabs.com/test-reports/NSS%20Labs%20Browser%20Security%20Test%20-%20Socially%20Engineered%20Malware.pdf

National Vulnerability Database (2009). *Checklist details for FDCC IE7 1.2*. Retrieved October 26, 2009, from http://web.nvd.nist.gov/view/ncp/repository/checklistDetail?id=170

National Vulnerability Database (2009). *National checklist program repository*. Retrieved November 6, 2009, from http://web.nvd.nist.gov/view/ncp/repository

National Vulnerability Database (2009). *Security content automation protocol validated products.* Retrieved on 11/06/2009 from http://nvd.nist.gov/scapproducts.cfm#scapproducts

Phillips, David (2008). *Malware in the virtual world*. Retrieved October 26, 2009, from http://webmedia.company.ja.net/content/documents/shared/networkshop080408/phillips-malwareinthevirtualworld.pdf

Portony, Aaron (2007). *Adobe flash player JPG processing heap overflow vulnerability*. Retrieved October 26, 2009, from http://dvlabs.tippingpoint.com/advisory/TPTI-07-21

Pruitt-Mentle, Davina (2008). *2008 National cyberethics, cybersafety, cybersecurity baseline study.* Retrieved October 26, 2009, from http://www.whitehouse.gov/files/documents/cyber/National%20Cyber%20Security%20Alliance%20-%202008%20National%20Cyberethics,%20Cybersafety,%20Cybersecurity%20Baseline%20Survey_11_14_08_Final.pdf

Red Hat, Inc (2009). *Introduction to SELinux.* Retrieved November 14, 2009, from http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/ch-selinux.html

Riden, Jamie (2007). *Known your enemy: malicious web servers.* Retrieved October 26, 2009, from http://www.honeynet.org/node/154

Chris Crowley, crowleigh@gmail.com

SANS Institute (2009). *SANS: Information security policy templates.* Retrieved October 26, 2009, from http://www.sans.org/security-resources/policies/

Sotirov, Alexandar (2009). *MD5 considered harmful today: creating a rogue CA certificate*. Retrieved October 26, 2009, from http://www.win.tue.nl/hashclash/rogue-ca/

Sun Microsystems, Inc. (2009). *Open source desktop virtualization.* Retrieved November 14, 2009, from http://www.sun.com/software/products/virtualbox/

Symantec Corporation (2009). *Welcome to Mozilla Firefox for HP Virtual Solutions*. Retrieved October 26, 2009, from http://www.symantec.com/virtualfirefox/welcome.jsp

Verisign, Inc (2009). *Form 10-Q for verisign inc/CA.* Retrieved October 26, 2009, from http://biz.yahoo.com/e/090806/vrsn10-q.html

Verisign, Inc (2009). *Digital ID a brief overview*. Retrieved October 26, 2009, from http://www.verisign.com/static/005326.pdf

Vukicevic, Vladimir (2009). *Mozilla foundation security advisory 2009-64 (credit for exploit).* Retrieved October 26, 2009, from http://www.mozilla.org/security/announce/2009/mfsa2009-64.html

Westervelt, Robert (2009). *DNSSEC deployment challenges can be overcome*. Retrieved October 26, 2009, from http://searchsecurity.techtarget.com/news/interview/0,289202,sid14_gci1367915,00.html

White Hat Security (2009). *WhiteHat website security statistic report*. Retrieved October 26, 2009, from http://www.whitehatsec.com/home/assets/WPstats_spring09_7th.pdf

Wilson, Mark (2003). *Building an information technology security awareness and training program.* Gaithersburg: National Institute of Standards and Technology.

Appendix A – lockPref.txt

```
//
// begin lockPref.txt - cc
//ask where to save downloads – US-Cert.gov
lockPref("browser.download.useDownloadDir",false);
// remember forms filled – US-Cert.gov
lockPref("browser.formfill.enable", false);
// ask for cookies – US-Cert.gov
lockPref("network.cookie.lifetimePolicy", 1);
// warn for addon install – US-Cert.gov
```

Chris Crowley, crowleigh@gmail.com

```
lockPref("xpinstall.whitelist.required", true);

// disable java – US-Cert.gov

lockPref("security.enable_java", false);

// manage media file types

// automatically install application updates

lockPref("app.update.autoInstallEnabled", true);

// noscript specific lockPref

// from http://hackademix.net/2007/12/05/plugin-security-plug-insecurity/#comment-
2209

// you'd better copy the site list from the

//   "capability.policy.maonoscript.sites"

//   key in the prefs.js file found in a test profile

lockPref("noscript.default", "informaction.com http://informaction.com
https://informaction.com flashgot.net http://flashgot.net https://flashgot.net
noscript.net http://noscript.net https://noscript.net");

// hide context menu

lockPref("noscript.ctxMenu", false);

// hide statusIcon

lockPref("noscript.statusIcon", false);

// hide notification bar

lockPref("noscript.notify", false);

// disable DOM Inspector and Error Console

//   (which may be used to programmatically unlock the prefs)

lockPref("noscript.lockPrivilegedUI");

// disable global scripts

lockPref("noscript.showGlobal", false);

// end lockPref 2009-11-06 cc
```

Appendix B – byte shifted mozilla.cfg

```
""�""#UXZ\a#_bV^CeXY!gkg# #VV�""Tf^#j[XeX#gb#fTiX#Wbja_bTWf#�s�#HF 6Xeg!
Zbi�_bV^CeXY##UebjfX

e!Wbja_bTW!hfX7bja_bTW7\e##YT_fX#.�""#eX`X`UXe#Ybe`f#Y\__XW#�s�#HF 6Xeg!
Zbi�_bV^CeXY##Uebj

fXe!Ybe`Y\__!XaTU_X###YT_fX#.�""#Tf^#Ybe#Vbb^\Xf#�s�#HF 6Xeg!Zbi�_bV^CeXY##aXgjbe^!
Vbb^\X!_

\YXg\`XCb_\Vl###$#.�""#jTea#Ybe#TWWba#\afgT__#�s�#HF 6Xeg!Zbi�_bV^CeXY##kc\afgT__!j[\
gX_\fg!

eXdh\eXW###gehX#.�""#W\fTU_X#]TiT#�s�#HF 6Xeg!Zbi�_bV^CeXY##fXVhe\gl!
```

Chris Crowley, crowleigh@gmail.com

```
XaTU_XR]TiT###YT_fX#.

TaTZX#`XW\T#Y\_X#glcXf@""#Thgb`Tg\VT__l#\afgT__#Tcc_\VTg\ba#hcWTgXf@_bV^CeXY##Tcc!
hcWTgX!

Thgb<afgT__8aTU_XW###gehX#.@""#abfVe\cg#fcXV\Y\V#_bV^CeXY@""#Yeb`#[ggc-""[TV^TWX`\k!
aXg"%

##*"$%"#("c_hZ\a fXVhe\gl c_hZ \afXVhe\gl"#Vb``Xag %%#,@""#lbh@s@W#UXggXe#Vbcl#g[X#f\
g

X#_\fg#Yeb`#g[X#@""####VTcTU\_\gl!cb_\Vl!`TbabfVe\cg!
f\gXf#@""###^Xl#\a#g[X#ceXYf!]f#Y\_X#YbhaW#\a

#T#gXfg#cebY\_X@""_bV^CeXY##abfVe\cg!WXYTh_g####\aYbe`TVg\ba!Vb`#[ggc-""\aYbe`TVg\ba!
Vb`#[g

gcf-""\aYbe`TVg\ba!Vb`#Y_Tf[Zbg!aXg#[ggc-""Y_Tf[Zbg!aXg#[ggcf-""Y_Tf[Zbg!aXg#abfVe\c

g!aXg#[ggc-""abfVe\cg!aXg#[ggcf-""abfVe\cg!
aXg##.@""#[\WX#VbagXkg#`Xah@_bV^CeXY##abfVe\cg

!Vgk@Xah###YT_fX#.#@""#[\WX#fgTghf<Vba@_bV^CeXY##abfVe\cg!
fgTghf<Vba###YT_fX#.@""#[\WX#abg\Y\VT

g\ba#UTe#@_bV^CeXY##abfVe\cg!
abg\Yl###YT_fX#.#@""#W\fTU_X#7B@#<afcXVgbe#TaW#8eebe#6bafb_X#@""####j[\

V[#`Tl#UX#hfXW#gb#cebZeT``Tg\VT__l#ha_bV^#g[X#ceXYf#@_bV^CeXY##abfVe\cg!
_bV^Ce\i\_XZXWH<###YT_

fX#.@""#W\fTU_X#Z_bUT_#fVe\cgf@_bV^CeXY##abfVe\cg!
f[bj:_bUT_###YT_fX#.@""#XaW#_bV^CeXY#%##, $

$ #)#VV
```

Chris Crowley, crowleigh@gmail.com

Last Updated: January 16th, 2018

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Las Vegas 2018	Las Vegas, NVUS	Jan 28, 2018 - Feb 02, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MDUS	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS Miami 2018	Miami, FLUS	Jan 29, 2018 - Feb 03, 2018	Live Event
SANS London February 2018	London, GB	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Scottsdale 2018	Scottsdale, AZUS	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS SEC455: SIEM Design Beta One 2018	Arlington, VAUS	Feb 12, 2018 - Feb 13, 2018	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CAUS	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Secure India 2018	Bangalore, IN	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Brussels February 2018	Brussels, BE	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Secure Japan 2018	Tokyo, JP	Feb 19, 2018 - Mar 03, 2018	Live Event
Cloud Security Summit & Training 2018	San Diego, CAUS	Feb 19, 2018 - Feb 26, 2018	Live Event
SANS Dallas 2018	Dallas, TXUS	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS New York City Winter 2018	New York, NYUS	Feb 26, 2018 - Mar 03, 2018	Live Event
CyberThreat Summit 2018	London, GB	Feb 27, 2018 - Feb 28, 2018	Live Event
SANS London March 2018	London, GB	Mar 05, 2018 - Mar 10, 2018	Live Event
SANS Secure Singapore 2018	Singapore, SG	Mar 12, 2018 - Mar 24, 2018	Live Event
SANS San Francisco Spring 2018	San Francisco, CAUS	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Secure Osaka 2018	Osaka, JP	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Paris March 2018	Paris, FR	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	McLean, VAUS	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Munich March 2018	Munich, DE	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Pen Test Austin 2018	Austin, TXUS	Mar 19, 2018 - Mar 24, 2018	Live Event
ICS Security Summit & Training 2018	Orlando, FLUS	Mar 19, 2018 - Mar 26, 2018	Live Event
SANS Secure Canberra 2018	Canberra, AU	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Boston Spring 2018	Boston, MAUS	Mar 25, 2018 - Mar 30, 2018	Live Event
SANS 2018	Orlando, FLUS	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, AE	Apr 07, 2018 - Apr 12, 2018	Live Event
Pre-RSA® Conference Training	San Francisco, CAUS	Apr 11, 2018 - Apr 16, 2018	Live Event
SANS Dubai 2018	OnlineAE	Jan 27, 2018 - Feb 01, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced