The Art of Secure Web Browsing

Security and Privacy in Web Browsers

Ananth Jillepalli

July 31, 2017
Version 2.2

University of Idaho

CS 539: Applied Security Concepts

Summary

Web Browsers are a significant threat to system security and individual privacy. The flexibility of functionality offered by Web Browsers comes at a cost of rampantly sheer number of security vulnerabilities and privacy exploitations. In the oncoming tutorial, we will detail several steps to secure Web Browsers from most prevalent attacks and also to protect your identity and data privacy, when on-line.

Contents

1 Objectives of this Tutorial >

1. Browser Hardening:
   (a) Keeping Browsing Data Secure.
   (b) Enhancing Privacy by Mitigating Tracking.
   (c) Increasing Security by Cookie Management.
   (d) Protecting both Security and Privacy by managing Scripts, Images etc.,
   (e) Secure browsing habits [Cache and Credential management, etc.,]

1. Data not recorded is data not stolen. As such, not allowing the browser to record browsing history is the best possible mitigation. There are other alternatives like encryption and obfuscation, but none are effective if your browser is compromised as it will always have unrestricted access to browsing data.

2. It's quite common knowledge these days, that if you are not paying for a service, then it is highly possibly that you are the product or byproduct and not the customer [7]. This is done through a series of third-party trackers, who gain information about you and sell that data to advertisement companies, leading to an infringement of privacy. We will see a series of mitigations in this tutorial to enhance privacy of ourselves.

3. Cookies are one of the most vulnerable features of a web browser, with a potential to compromise the entire system of user through the cookie injection exploitation [6]. As such, in this tutorial, we will learn about better cookie management and thereby, increasing security of browsers.

4. Languages like JavaScript and Steganographic scripting through images is one of the other primary avenues for vulnerability exploitation. Also, Phishing is one of the major security problems related to scripting. As such, in this tutorial, we will see management of scripts and images as well, with the use of add-ons in Mozilla Firefox, natively in Google Chrome and in Internet Explorer.

5. Last, but not the least, we will also have a quick look at secure browsing habits, which includes, but is not limited to cache and credential management. On a whole, at the end of this tutorial, one can expect to gain sufficient knowledge to keep their browsers secure.

2 Required Background <>

We assume that the reader of this tutorial has an extent of background knowledge in the following areas:

1. Working experience on usage of computers and software applications, like web browsers, and virtualization software.
2. Basic overall idea of computer networks and Internet.
3. Fundamentals of website features like cookies, javascript, image rendering, etc.,
4. An overall idea on general issues like data privacy, computer security, etc.,

Due to restrictions on time and manpower resources, we are not able to make the ensuing tutorial to be completely self-contained from the perspective of a user. As such, the tutorial is best used when the user already has certain background skills and knowledge. The following are some areas where we expect the users of this tutorial to have some previous skills/knowledge:

1. Practical experience on using computers, installing and using common software applications (particularly web browsers, and virtualization software platforms). The tutorial does not explain how to navigate within the operating system's graphical user interface (GUI). Similarly, the tutorial also does not explain how to browse the Internet, how to install software applications, and how to use/navigate software applications.
2. An overall idea on the working of computer networks and Internet. The tutorial expects a user to understand common terms like "web server", "links", "websites", and "web browser / web engines" etc.,
3. Fundamental or very basic idea on features of a web-page or any website. The tutorial expects a user to understand technical terms like "web cookies / cookies", "javascript", "image rendering", and "graphics scalability" etc.,
4. Also bit of exposure to computer programming would be very helpful in tackling the challenges efficiently. An overall brief idea on general computer-related issues like "data privacy", "computer security", "network security", "attacks on computer or network" etc., will help a lot.

3 Hardware and Software Requirements <>

We recommend having at least the following hardware and software specifications for smooth execution of this tutorial's activities and challenges:

1. A computer which can at least boot 1 virtual machine (VM) smoothly, with no noticeable lag and delay
2. A functional virtualization software platform. For example, VMWare or VirtualBox.
3. A vanilla Windows 10 VM with Mozilla Firefox and Google Chrome installed and functioning.

For the purpose of getting the best experience out of this tutorial, there are certain minimum hardware and software requirements that we recommend users have at their disposal. However, this tutorial can also be carried out in lower specifications than what is recommended.

1. A computer powerful enough to be able to boot 1 virtual machine without any noticeable delay or log. That would mean at least a quad-core processor, 8 GB RAM, optimally two monitors (can be managed with one), and a functional keyboard and mouse.
2. It is always recommended to use a virtual machine to run tutorials like the one available in this document so as to not destabilize one's own workstation or personal machine's environment and software configurations. To that extent, we recommend having a virtualization software installed on machine, which can be used to generate virtual machines as needed.
3. Though the present tutorial can be followed with any operating system which supports Mozilla Firefox and Google Chrome, to be consistent with file paths and locations, we recommend using a vanilla (default installation) Windows 10 virtual machine with Mozilla Firefox and Google Chrome installed.

   Note: Please go to part **2** of the **Appendix** section for information on resources and website links to access some of the required software for the tutorial.

4 Introduction: Web Browsers <>

1. Evolution of Modular Web Browsers
2. Usage Statistics
3. Reasons for Web Browser Success
   (a) Ease of use & Module-based Design
   (b) Scalable, GUI-based & Advanced Graphics Capability
   (c) Network accessible and immediately deployable
   (d) Turing-complete virtual execution capabilities

1. Web browsers started out as monolithic (single operational functionality), without any additional capacity apart from accessing plain websites and producing the data to the users. After few years down the lane in development, "Web Engines" came into existence, which enhance the operational scope of web browsers to include native e-mail clients, document readers, markup language formatting, image rendering and object interaction.

2. One of the foremost web browsers to have started usage of "web engine" functionality was Mosaic, developed in 1993. Based on Mosaic, Netscape Navigator was developed. Later on, Internet Explorer and many other browsers like Mozilla Firefox, Google Chrome, Safari and etc., started using different web engines, like: *Gecko* for Firefox, *Trident* for Internet Explorer and *Blink* for Google Chrome. Now-a-days, web browsers are almost full virtual machines in themselves.

3. According to StatCounter [statistics derived from hits, which are not unique, from 3 million sites, totaling more than 15 billion hits per month], by the end of 2015 - Microsoft Internet Explorer had 17.3%, Google Chrome had 54.2%, Mozilla Firefox had 14.61%, Safari had 9.42% and others constituted the remaining percentage of web browser usage share, globally, across all devices except non-Android phones [1]. Human usage of Internet and subsequently, usage of web browsers to access networks, has become so enormous that 47% of entire world population and 88% of United States population use at least one web browser in their lives [2].

4. As most of the service content is moving towards web-service environments, necessity and dependence on web browsers is growing. This immense dependency of humans on web browsers ultimately creates problems on an international scale.

5 Problem Statement: Secure Configuration <>

1. Problems related to security and privacy
   (a) Phishing
   (b) Cross Site Scripting (XSS)
   (c) Cross Site Request forgery (CSRF)
   (d) Execution After Redirect (EAR)
2. Can be effectively countered through secure configuration
3. Lack of awareness among people about such configurations

1. Because of the flexibility, scalability, modularity, and ease-of-use provided by web browsers, they are susceptible to various kinds of attacks, both on security of the user machine and on the privacy of the user. One most popular of such attacks is Phishing attack, which is very efficient in targeting both security and privacy of a user and the machine. For example, a most common phishing attack is carried out in the form of having a dubious link asking banking and personal information, disguising the link as target's bank web site, sent out to people at a large scale.

2. Similarly, attacks like XSS, CSRF and EAR are also common, exploiting the weakness from web application side and not from user side, for the most part. XSS is an attack where JavaScript functionality of a web browser is exploited to perform unauthorized script requests. CSRF involves duping a server into thinking an attacker is a regular user and finally, EAR deals with sending redirect links which have malicious scripts embedded in them, which execute after being redirected.

3. The attacks explained above and stated in the slide are a mere number, scratching the surface of possible attacks on web browsers. Though most common, the above mentioned attacks are not very complicated and can be easily blocked through the use of secure configurations. Secure configuration of web browser, by no means, makes it invincible against attacks, but they provide a very good defense and help secure web browsers against many common attacks. However, lack of awareness and training amongst people regarding such secure configuration of web browsers causes many people to be victims of such attacks and compromise their privacy and/or data integrity.

6 Related News: Recent Exploits <>

1. Microsoft Edge Storing Private Data (29th Jan. 2016) [3]
2. Firefox Leads Private Browsing with In-built Tracking Protection (4th Nov. 2016) [4]
3. Web Browsers are a Goldmine for Attackers (13th Oct. 2015) [5]
4. Cookies Can be Used to Access Private Data, Even Over HTTPS Connections (25th Sept. 2015) [6]

1. In a very recent news article [3], a vulnerability in Microsoft Edge's incognito "InPrivate" mode has been exposed, where the data of websites visited while browsing in *InPrivate* mode can be obtained by examining Web Cache files stored on the hard disks of users. Visited sites can be found in same "Container_n" table that stores tab history from normal browsing.

2. During November of 2015 [4], Mozilla Firefox became the first Web Browser to have taken a step forward in having inbuilt protection against tracking by automatically blocking a list of trackers. This is resulted in up to 44% faster page loading time.

3. In a separate article published on 13th October of 2015 [5], Madelyn Bacon of TechTarget wrote that browsers using Flash, Java and Silverlight specifically are often targeted, though the fault often lies not with browsers, but with third-party software extensions. She quotes that, in most cases, even if browser developers knew about the vulnerability, they cannot do anything as the fix should be carried out by respective third party developers.

4. During September of 2015 [6], United States Computer Emergency Response Team (CERT) issued a vulnerability note warning that cookies in web browsers could allow remote attackers to bypass secure sessions so as to gain access to private information, with the help of cookie injection attacks.

5. All of these and many other events strongly suggested one thing: **Users** should be more aware about security and privacy configurations, for their own sake. To that end, we will be going through a "Browser Hardening" Tutorial in this presentation.

7 Outline of Tutorial <>

1. Configuring Google Chrome for Security & Privacy
2. Configuring Mozilla Firefox for Security & Privacy
3. Configuring Internet Explorer and Microsoft Edge
4. Challenges

1. Due to a certain degree of redundant actions and for efficient organization of this tutorial, we have laid out the delivery of content as depicted above. That is, we will demonstrate configuring a single web browser before moving onto another one, instead of handling two browsers at once and increasing the confusion.

2. At first, we will have a thorough understanding of configuring Google Chrome. Since Google Chrome is based on Project Chromium, this configuration process can be applied for all Chromium based browsers. In addition to native configuration, we will also see how to enhance privacy and security in Google Chrome using add-ons: Ghostery and Adblock Plus.

3. Second, we will learn through an add-on in Firefox called Lightbeam, how websites are filled with trackers and how dangerous the tracking entities can be if used by exploiting attackers. Then we will walk-through how to configure security and privacy using native settings as well as with add-ons such as NoScript, Ghostery, and Cookie Manager+.

4. Third, we will have a look at security and privacy enhancing configurations for Internet Explorer and Microsoft Edge. Since both of these browsers share the same set of nomenclature, we figured it would be easier to configure them one after the other in a succession. Windows 10 oddly enough, has both Internet Explorer 11 and Microsoft Edge.

8 Questions 1,2 <>

1. What is the name of web engine used by Internet Explorer?
2. What type of attack technique can be used to exploit a web browser feature to access private data over HTTPS connection?

Answer to Q1: Trident

Answer to Q2: Cookie Injection

9 Background: Web Browser Add-Ons <>

- There are thousands of add-on modules for most of the web browsers these days.
- While add-ons provide additional functionality and features, they also pose a risk.
- There is a significant trade-off between what we gain and what we compromise for the additional functionality of add-ons
- There are some add-ons which are relatively a lot more malicious than others.

Trade offs for Add-Ons:
In this tutorial, we will be using some add-ons or extensions for default web browsers. However, there are some trade-offs while using web browser add-ons or extensions. They are:
Pros:

+ Increased flexibility.
+ Enhanced functionality.
+ More customization.

Cons:

– Trust Assumption. We have to assume the creators of add-ons are not of malicious intent.

– Third-party factoring issues. Add-on/extension creators are third party and might not always comply with respective browser policies or software development philosophy.

– Increases avenue for vulnerability. Add-ons are usually not as quick in being updated as web browsers, leading to vulnerabilities which are patched in browser but are existent in add-ons.

Bottom line: All the add-ons, which will be used in this tutorial are trusted by open-source community and are not known to cause any specific vulnerability. However, Adblock Plus has been stated to consume heavy CPU resources and it is also rumored to have allowed ads by organizations who pay Adblock Plus for their ads to be white-listed [16] [17]. A better alternative seems to be **uBlock Origin** [18].

10 Activity: Configuring Google Chrome - I <>

1. Cookie Management
2. JavaScript, Images and Content Settings
3. Search Engines, Credentials and Certificates

1. **Cookie Management**:
   (a) Open Google Chrome.
   (b) Click on `Customize and Control Google Chrome` (the three horizontal bars in the upper right corner of the browser) and select `Settings`.
   (c) Click on `Show Advanced Settings` hyper-link. Click on `Content Settings` button.
   (d) Tick the `Block Sites from Setting Any Data` check-box; Also tick `Block third-party cookies and site data`.
   (e) Make sure to delete all unwanted exceptions present under `Manage Exceptions` and to remove all unwanted cookies in `All cookies and site data`.
   (f) A better way of managing cookies is through add-on "**Cookies**". This add-on makes it easier to encrypt cookie data and/or to edit existing cookies.

2. **Javascript, Images and Content Settings**:
   (a) Open Google Chrome.
   (b) In dealing with Javascript and images, we recommend using the policy of "white-listing" instead of blacklisting.
   (c) Click on `Customize and Control Google Chrome` and select `Settings`.
   (d) Click on `Show Advanced Settings` hyper-link. Click on `Content Settings` button.

(e) Go ahead and choose `Do not show any images` and `Do not allow any site to run JavaScript`.

(f) As per need, one can go on adding exceptions by clicking on `Manage Exceptions` button and adding the domains to list. This is an effective policy in combating against phishing, the attack where an unwanted and a mostly malicious link is scripted into seemingly harmless link placed in a seemingly legitimate context.

(g) In `Content Settings`, scroll down to `Handlers` and select `Do not allow any site to handle protocols`. Coming to `Plugins`, select `Let me choose when to run plugin content`.

(h) Scroll down to `Location` and select the `Do not allow any site to track your physical location` radio option.

(i) Scroll down to `Microphone` feature and select `Do not allow sites to access your microphone`. When you select that, you will be presented with an option to change Adobe Flash microphone settings as well. Do it by clicking on `Change` hyper-link [requires Internet access].

(j) Similarly, scroll down to `Camera` feature and select `Do not allow sites to access your camera`. When you select that, you will be presented with an option to change Adobe Flash camera settings as well. Do it by clicking on `Change` hyper-link [requires Internet access].

3. **Search Engines, Credentials and Certificates**:

(a) Open Google Chrome.

(b) Click on `Customize and Control Google Chrome` and select `Settings`.

(c) Under `Search` feature, click on `Manage search engines`. In here, delete every existing and unnecessary search engines.

(d) Add "**Startpage**" and/or "**DuckDuckGo**" search engines.

(e) Under `Passwords and Forms` feature, un-tick `Enable Autofill to fill out web forms in a single click` and also un-tick `Offer to save your web passwords`

(f) Under `HTTPS/SSL` feature, click on `Manage Certificates`. Click on `Advanced`. In the subsequent dialogue box, tick everything.

11 Activity: Configuring Google Chrome - II <>

1. Ghostery
2. Adblock Plus
3. Intro to Configuration File editing

1. **Ghostery**:
   Ghostery provides information to users, helping them make informed decisions about allowing trackers to collect personal footprints, from the sites visited by users. Ghostery also provides users with controls, which can block unrequited trackers subsequently, increasing page load times and enhancing user privacy.

   (a) Open Google Chrome.
   (b) Click on the **Ghostery** side-heading above to be redirected towards official distribution of Ghostery add-on for Google Chrome.
   (c) Click on `Add to Chrome` in the subsequent dialogue box.
   (d) You will be redirected to Ghostery configuration page. Here, click `Next`.
   (e) In the ensuing screen, select `No Thanks`.
   (f) Click `Next` in the `Notification` section.
   (g) In the `Blocking` section, click on `Select all` hyper-link. Click `Next` and configuring Ghostery is done.

2. **Adblock Plus**:
   Adblock Plus is a one-stop filter for blocking JUST the annoying ads and not breaking sites completely.

   (a) Open Google Chrome.

(b) Click on the **Adblock Plus** side-heading above to be redirected towards official distribution of Adblock Plus add-on for Google Chrome.

(c) Click on `Add to Chrome` in the subsequent dialogue box. Click on the `Add extension` button to confirm.

(d) You will be redirected to Adblock Plus configuration page.

(e) In this page, scroll down to find a section titled `Adblock Plus can do more than block ads`.

(f) Under that section, turn on all three options of `Malware Blocking`, `Remove Social Media Buttons`, and `Disable Tracking`. Feel free to close the tab.

(g) Click on ABP octagon on the right top corner of the browser. In the drop-down box, select `Options`.

(h) In the options page, under `Filter lists`, tick every default list and click on *Update now*. Also, un-tick `Allow some non-intrusive advertising`.

3. **Intro to Configuration File editing**:
All of the above-mentioned settings and even further advanced customization of configurations can be done through editing `Preferences` file of Google Chrome, which can be found in `default` folder of user data under `Google Chrome` in `AppData`'s `Local folder`. The language used by Google Chrome's setting is very verbose and not easily understandable by humans. As such, it is strongly suggested to not make any modifications in it unless ABSOLUTELY necessary, to avoid troubles.

   As an example, we can disable cookies by adding string phrase: `"default_content_setting_values":"cookies":2,` after the partial string `"created_by_version":"48.0.2564.97",` (or whatever version number is current). Location is critical to Google Chrome's file configuration functionality, it will not work if you place the string wherever you want in the file, unlike Firefox.

12 Activity: Configuring Mozilla Firefox - I <>

1. Lightbeam Demonstration
2. Search Engines, Content Settings, and General Security
3. Privacy and Cookies

1. **Lightbeam Demonstration**:
   Lightbeam enables users to see the second-party (destination sites) and third-party (unwanted sites) websites with whom, the first-party (users) interact. As the users browse, Lightbeam records every interaction and footprint tracking attempt. These records are then visualized in the form of interactions using a mind-map type of graph. The visualization data is temporarily stored in user's storage and the user has the option to permanently save the data. To test a case:

   (a) Open Mozilla Firefox.

   (b) Click on the **Lightbeam** side-heading above to be redirected towards official distribution of Lightbeam add-on for Mozilla Firefox.

   (c) In the page, click on `Add to Firefox` button. Click on the `Install` button to confirm.

   (d) Click on the diamond-shaped Lightbeam icon in the top right corner of browser.

   (e) This is the visualization display page and main dashboard of Lightbeam add-on.

   (f) To start visualization, let's open new tabs and visit some sites. For example: New York Times, MSN, Yahoo, Amazon, Weather Underground, and Facebook. On every site, scroll down a bit to load the pages properly.

   (g) Observe the number of sites `You Have Visited` and third-party sites which `You Have Connected With`.

2. **Search Engines, Content Settings, and General Security**:

(a) Open Mozilla Firefox.

(b) Click on the `Open menu` button in the top-right corner of browser.

(c) In the drop-down pane, click on `Options` button.

(d) In the left navigation pane, click on `Search` option.

(e) Under the `One-click search engines` section, remove all unnecessary search engines. You can add **Startpage** as a search engine if privacy is your concern.

(f) In the left navigation pane, click on `Content` option. Here, make sure there are no exceptions under pop-up blocking by clicking on `Exceptions...` button under `Pop-ups` subsection.

(g) In the left navigation pane, click on `Security` option. Here, make sure there are no exceptions under add-ons installation warning by clicking on `Exceptions...` button beside `Warn me when sites try to install add-ons` check box (which should be checked). In the same page, under `Logins` subsection, un-tick both check-boxes.

(h) In the left navigation pane, click on `Advanced` option, click on `Data Choices` tab. Here, it is usually safest to un-tick both *Enable Firefox Health Report* and *Enable Crash Reporter*. Under `Certificates` tab, select the `Ask me every time` radio button for personal certificate requests.

3. **Privacy and Cookies**:

(a) Open Mozilla Firefox.

(b) Click on the `Open menu` button in the top-right corner of browser.

(c) In the drop-down pane, click on `Options` button.

(d) In the left navigation pane, click on `Privacy` settings.

(e) Under `History` subsection, select the option `Use custom settings for history`, from the drop-down list of `Firefox will:` and un-tick `Remember search and form history`. `Accept third-party cookies` should be set to `Never` and `Keep until` should be `I close Firefox` from their respective drop-down lists.

A better way to manage unrequested cookies is to either use native Firefox cookie manager, which is time consuming to access (and which does not have any advanced features of cookie management) or **Cookie Manager+**. This add-on is a highly advanced cookie manager, which can import, export, modify, inject and remove cookies.

13 Questions 3,4 <>

1. What is the most feasible method to increase browser security in an Organization?
   A. Uninstall Internet Explorer.
   B. Disable JavaScripts.
   C. Create safe-browsing-configuration awareness in people.
   D. Disconnect Organization's computers from networks.
2. What is an effective browser security policy in combating "phishing"?

Answer to Q3: Option C - Create safe-browsing-configuration awareness in people.

Answer to Q4: Educating people and spreading awareness about phishing and white-listing any scripting functionality.

14 Activity: Configuring Mozilla Firefox - II <>

1. NoScript
2. Ghostery
3. Intro to "about:config"

Firefox does not have a native method of whitelisting or blacklisting JavaScripts through GUI. It can be done through NoScript extension.

1. **NoScript**:

   (a) Open Mozilla Firefox.

   (b) Click on the **NoScript** side-heading above to be redirected towards official distribution of NoScript add-on for Mozilla Firefox.

   (c) In the page, click on `Add to Firefox` button. Click on the `Install` button to confirm.

   (d) Hover your mouse over the `S` NoScript icon. In the drop-down list, click on `Options`.

   (e) Under the `General` tab of `Options`, check the `Reload the current tab only` check box.

   (f) Under the `Embeddings` tab of `Options`, check all the check-boxes under `Additional restrictions for untrusted sites` subsection.

   (g) Under the `Appearance` tab of `Options`, check all the check-boxes except `Status bar label`, `Full Domains`, and `Full Addresses` check-boxes.

   (h) Under the `Advanced` tab of `Options`, there are six sub-tabs. Under `Untrusted` tab, check everything except `Hide <NOSCRIPT> elements` option. Under `Trusted` sub-tab un-check `Show the <NOSCRIPT> element...` option.

(i) These settings can be exported and imported through respective buttons in `Options` window. No need to memorize all of these settings.

If NoScript seems overkill or unmanageable, there is always **Adblock Plus**(link embedded into the word), for providing basic safety against scripts. Adblock Plus fulfills its functionality as it does for Google Chrome ^ (please click on ^ symbol to go to the respective section).

2. **Ghostery**

   (a) Open Mozilla Firefox.

   (b) Click on the **Ghostery** side-heading above to be redirected towards official distribution of Ghostery add-on for Mozilla Firefox.

   (c) In the page, click on `Add to Firefox` button. Click on the `Install` button to confirm.

   (d) Ghostery satisfies and fulfills its functionality as it does for Google Chrome ^(please click on ^ symbol to go to the respective section). Firefox version of Ghostery can in addition, even block third-party cookies.

3. **Intro to "about:config"**: Using this method, we can alter advanced web browser settings, which would negatively impact web browser stability if improperly handled. As such, it is preferable to not use this method unless either it is necessary or user is in a sandbox environment. This configuration terminal can be accessed through typing `about:config` in the address location bar.

   In here, we can modify configurations by changing either integer values or boolean flags. For example, to keep cookies until Firefox is closed, we can search for cookie in the search bar, find `network.cookie.lifetimePolicy` setting and set the value to 2. This change can be noted in GUI immediately after it is set, because the setting changes display to bold whenever changed from default value.

   A further advanced alternative method is to browse for "prefs.js" file. The file can be located under `C -> Users -> <UserName> -> AppData -> Roaming -> Mozilla -> Firefox -> Profiles -> <random eight characters string>.default`. Append "`user_pref("network.cookie.lifetimePolicy", 2);`" to the Prefs file. This action does exactly the same thing as changing the value of setting to 2 in "about:config", but in a more non-GUI kind of style.

15 Activity: Configuring Microsoft Browsers <>

1. Internet Explorer (Yes, Windows 10 still has Internet Explorer 11 installed in it, by default.)
2. Microsoft Edge

1. **INTERNET EXPLORER (IE):**

   (a) **Managing Browsing History:**
   We can delete the Browsing History by pressing `Ctrl+Shift+Del` or by going to `Tools->Safety->Delete Browsing History`. `Tools` is a gear-shaped icon at the top-right of web browser. For better privacy, select all the options and click on `Delete` button.

   (b) **InPrivate Mode:**
   Browsing in InPrivate does not store any data about the browsing session in the computer (example: temporary internet files, cookies and browsing history). We can browse in InPrivate session by pressing `Ctrl+Shift+P`, or going to `Tools->Safety->InPrivate Browsing`, or right click the Internet Explorer icon anywhere and click on `Start InPrivate Browsing` option.

   (c) **Turn Off Do Not Track Request:**
   To turn on or off `Do Not Track requests`, go to `Tools->Safety->Turn On Do not track request`. If that action is checked, web browser has turned on do not track requests. If the action is not checked, web browser has turned off no-tracking requests.

   (d) **ActiveX Filtering:** To turn on or off ActiveX Filtering Request, go to `Tools->Safety->ActiveX Filtering.`

(e) **Privacy Report:**
To get details about a webpage privacy, go to `Tools->Safety->Webpage privacy report`. This particular list gives us information regarding what sites' cookies are accepted and which ones are not (if a site is not mentioned as accepted, then cookies are not allowed on that site).

(f) **Check This Website:**
IE provides a feature called `Check this website`, which when clicked, reports about phishing and malware content in the current website. This feature can be accessed by going to and click on: `Tools->Safety->Check this website`.

(g) **SmartScreen:**
Check this website, uses SmartScreen filtering to find the malicious content. To turn off smart screen filtering, to go `Tools->Safety->Turn Off SmartScreen Filter`. To turn it back on, just click on the same thing and it will be turned on.

(h) **Report Unsafe Website:**
To report an unsafe website, go to `Tools->Safety->Report Unsafe Website`. This feature is useful towards alerting other users of IE about malicious or unwanted content on any page in which this feature is invoked.

(i) **Location Settings:**
It is highly suggested to turn off zeroing of user's location, unless and until necessary. To turn off the location setting, go to `Tools->Internet Options->Privacy` tab, and in here, check on `Never allow websites to request your physical location`. Sometimes, this option maybe grayed out. If so, location services are anyway disabled from control panel.

(j) **Pop-up Blocking:**
To turn on pop-up blocker, go to `Tools->Internet Options->Privacy`, and check on `Turn on Pop-up Blocker`. Next, click on `Settings`. Make sure there are no allowed sites, unless you explicitly allowed a site to throw out pop-ups. Also, at the bottom of this dialogue box, it is suggested to change `Blocking level` to `High: Block all pop-ups (Ctrl+Alt to override)`.

(k) **Managing Sites and Cookies:**
To manage cookies with respect to sites, go to `Tools->Internet Options->Privacy`, and click on `Sites` for managing exceptions per site. Click on `Advanced` for `First-party Cookies` and `Third-party Cookies`. It is advised to always use `Prompt` and `Block` values respectively for first and third party cookies.

(l) **Managing Scripting:**
To manage Scripting, go to `Tools->Internet Options->Security`. In here, click on `Custom Level` button. In the resultant box, scroll to `Scripting` section and choose either `Disable` or `Prompt` radio button for different kinds of scripting.

2. MICROSOFT EDGE:

(a) **Browsing history:** To clear web browser's stored data like cache and etc., go to More actions->Settings (More actions button can be found on top-right corner of the web browser) and in the resultant pane, click on `Choose what to clear` under `Clear browsing data` subcategory. Check all desired/necessary checkboxes and click on `Clear` button.

(b) **InPrivate Browsing:** To browse without allowing browser to store any data after it is closed, use InPrivate browsing. To do that, click on `More actions` and select `New InPrivate window`. Beware, as Microsoft Edge was found to be storing private data even while browsing in InPrivate mode. [3]

(c) **Turn on do not track request:** Navigate to `More actions->Settings` and in this pane, click on `View advanced settings` scroll to find `Send Do Not Track requests` from `Privacy and services` subcategory.

Bottom Line:

- Many "Advanced Settings" of Microsoft Edge, like disabling JavaScript, needs to be done through Active Directory and Administrative Templates, which is not convenient for non-CS/IT people.
- Internet Explorer had a great extent of flexibility in changing settings through GUI, Microsoft Edge does not have that extent of flexibility.
- Microsoft Browsers are becoming, heavily non-configuration friendly in terms of customization and custom configuration, as development goes on.
- Many safety settings are dependent on sending back data to Microsoft, which is known to have helped PRISM program along with Google and several major software companies [15].

Interesting to know: PRISM is a secret surveillance programme carried out by NSA [National Security Agency] of United States. Through PRISM, NSA snooped on data [including e-mails, photos, documents, etc.,] of citizens and politicians in many countries. Mozilla Foundation is not known to have helped NSA with its' PRISM program.

16 Questions 5,6 <>

Q5. How can we effectively disable JavaScript in Microsoft Edge?

A. Ask Cortana to do it.

B. Not possible natively, use Administrative Templates.

C. Install an extension/add-on for it.

D. None of the above.

Q6. Can Mozilla Firefox natively White-list or blacklist JavaScripting?

Answer to Q5: Option B - Not natively possible, use Administrative Templates.

Answer to Q6: Mozilla Firefox cannot natively whitelist or blacklist JavaScripting.

17 Challenge I: JavaScript White-listing <>

1. Through GUI of Google Chrome, using one rule per domain, white-list the following websites, after disabling JavaScript globally(*)(Easy Challenge).

Websites:

1. University of Idaho
2. Google Mail
3. Wikipedia
4. Outlook

* Not just creating the rule, but make sites work without breaking any necessary feature.

Hint: To make the sites really work without breaking anything, global white-listing of entire site domains is the efficient option.

18 Challenge II: Config File Study <>

1. White-list websites from Challenge I in NoScript GUI. Study the configuration code in "pres.js" file.
2. Observe how the config file documents the white-listed sites (*)(Medium Challenge).

* The purpose of this challenge is to learn a crucial skill required to tackle next two challenges.

Hint: If you have cleared Challenge I, you should be able to easily perform the 1st task of this challenge. For the second task, seeing the difference of "prefs.js" file before and after the modifications to NoScript GUI will be beneficial.

19 Challenge III: Config. File Editing <>

- For Firefox, using configuration file editing method, do the following:

  1. Disable cookies completely and disable JavaScript globally in native Firefox.
  2. Add a command enabling JavaScript globally, while not removing the disable command from the configuration file. Swap the order of command placement. Report what behaviour you can observe.

(Hard Challenge).

Hint: Apply mapping skill learned in previous challenge

20 Challenge IV: Config. File Editing <>

1. Using one rule per domain, White-list following websites in Google Chrome (after disabling JavaScript globally) using Preferences file editing, not the GUI of browser (*).

Websites:

1. Facebook
2. LinkedIn
3. Schweitzer Engineering Laboratory
4. Youtube

* Not just creating the rule, but make sites work without breaking any necessary feature.

Hint: If the previous challenges were completed, the final challenge is an extension of skills learned during all previous challenges.

21 Conclusion <>

1. Secure configuration of browsers is important.
2. Most widely exploited features of a Web Browser include scripting, cookies, and ease of use.
3. We have learned how to configure four web browsers [Google Chrome, Mozilla Firefox, Internet Explorer, and Microsoft Edge] for fundamental security.
4. Configuring browsers in a technologically diverse enterprise is not very easy with existing group policy tools.

1. In the background section of this tutorial, we have seen how secure configuration of web browsers can prove useful against mitigating common attacks like XSS, CSRF, and EAR. Secure configuration of web browsers is also important, not just because the privacy and security of users is at risk, but also because enterprises are at an even greater risk, as the large-scale financial repercussions are higher than individual users.
2. Secondly, we have also seen how difficult it can get for an administrator to deploy configurations across a large-scale enterprises' systems. Currently, there are no group-configuration tools which can run across multiple operating systems and with different web browser applications. As such, we have started an endeavour to solve the problem through designing a framework which can be used to specify security policies and these policies will be automatically transformed into configurations, irrespective of operating system and web browser application.
3. To conclude, this tutorial provides a fundamental knowledge of configuring Google Chrome, Mozilla Firefox, Internet Explorer, and Microsoft Edge. However, this tutorial will not and does not claim to make the browsers invincible against all forms of attacks.
4. There are many forms of advanced attacks which can compromise any web browser, irrelevant of the configurations being used. As such, it is always better to be safe than sorry, because nothing can ever be too-secure, but anything can be too vulnerable.

22 Appendix: Solutions and Change-log <

1. Solutions to the challenges
   (a) Challenge I
   (b) Challenge II
   (c) Challenge III
   (d) Challenge IV
2. Tutorial-Related Resources
3. Change-Log

1. **Solutions to the Challenges:**

   (a) **Challenge I:**
      i. Open Google Chrome.
      ii. Click on `Customize and Control Google Chrome` and select `Settings`.
      iii. Click on `Show Advanced Settings` hyper-link. Click on `Content Settings` button.
      iv. In the `Content Settings` dialogue box, under `JavaScript` subsection, click on `Do not allow any site to run JavaScript` radio button.
      v. Next, click on `Manage Exceptions...` and type in [*.]uidaho.edu, setting the Behavior to `Allow` for allowing University of Idaho's web-pages run JavaScripts.
      vi. Similarly, use [*.]gmail.com, [*.]wikipedia.org, and [*.]live.com for allowing execution of JavaScripts in Google Mail, Wikipedia, and Outlook respectively.

   (b) **Challenge II:**
      i. Open Mozilla Firefox.
      ii. Click on `S` icon of NoScript in the top-right corner of the page and select `Options`.
      iii. In the resultant pop-up box, click on `Whitelist` tab.
      iv. Under the `Address of web site:` subsection, there is an entry bar. Using it, specify the website domains in a format as given in the previous challenge's solution.

v. Notice the changes to `prefs.js` file found at: `C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\<randomstring>.default\prefs.js`. Here, audience learn the configuration code to GUI command mapping skill, crucial for next challenges.

(c) **Challenge III:**
Navigate to and open the file at:
`C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\<randomstring>.default\prefs.js`

At the end of this file, append the lines:
```
user_pref("network.cookie.cookieBehavior", 2);
user_pref("javascript.enabled", false);
```
Save the file

Ordering of configurations' commands plays a significant role in resolving conflicts. First placed configuration command takes priority, that means if two conflictive commands like disabling and enabling javascript are in the same configuration file, the one which is above the other will override the later command.

(d) **Challenge IV:**
Navigate to and open the files at:
`C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\Preferences`

Append the following, after string phrase `"images":{},`
```
"javascript":{"[*.]facebook.com,*":{"setting":1},"[*.]
linkedin.com,*":{"setting":1},"[*.]selinc.com,*":{"setting":1},
"[*.]youtube.com,*":{"setting":1}},
```

2. **Tutorial-Related Resources:**

   A free virtualization software platform - VirtualBox - CLICK ME.

   Mozilla Firefox - CLICK ME

   Google Chrome - CLICK ME

   Internet Explorer - CLICK ME

   A Windows 10 VM with Microsoft Edge - CLICK ME

3. **Change-Log:**

The Art of Secure Web-Browsing tutorial			
Ver.	**Date**	**Authors**	**Changes**
v1	Feb. 2nd 2016	Ananth Jillepalli	First draft of tutorial.
v2	May 24th 2016	Ananth Jillepalli	Major content additions and remodeled the structure.
v2.1	May 31st 2016	Ananth Jillepalli	Added appendix and other minor enhancements.
v 2.2	July 31st 2017	Ananth Jillepalli	Changed the licensing from CC BY-NC-ND 4.0 to CC BY-NC-SA 4.0

References

[1] StatCounter Global Stats, "January 2015 to December 2015", last accessed on 29th Jan. 2016, http://gs.statcounter.com/#browser-ww-monthly-201501-201512.

[2] Internet World Stats, "Internet Usage Statistics, The Big Picture", last accessed on 29th. Jan 2016, http://www.internetworldstats.com/stats.htm.

[3] Russell Brandom, "Microsoft's Edge Browser maybe Storing Private Browsing Data", last accessed on 29th. Jan 2016, http://www.theverge.com/2016/1/27/10845448/microsoft-edge-inprivate-storing-data-incognito, January 27th 2016.

[4] Lucian Armasu, "Firefox Leads Other Browsers With New Tracking Protection For Private Browsing", last accessed 29th Jan. 2016, http://www.tomshardware.com/news/firefox-tracking-protection-private-browsing,30484.html, November 4th 2015.

[5] Madelyn Bacon, "Why Web Browser Security is a Goldmine for Attackers", last accessed 29th Jan. 2016, http://searchsecurity.techtarget.com/video/Why-Web-browser-security-is-a-goldmine-for-attackers, September 25th 2015.

[6] Juha Saarinen, "Cookies Can Bypass HTTPS in Web Browsers", last accessed 29th Jan. 2016, http://www.itnews.com.au/news/cookies-can-bypass-https-in-web-browsers-409617, September 25th 2015.

[7] Alan Henry, "Everyone's Trying to Track What You Do on the Web: Here's How to Stop Them", last accessed 30th Jan. 2016, http://lifehacker.com/5887140/everyones-trying-to-track-what-you-do-on-the-web-heres-how-to-stop-them, February 22nd 2014.

[8] David Cancel, "Ghostery", last accessed 31st Jan. 2016, https://www.ghostery.com/, April 2009.

[9] The Tor Project and Electronic Frontier Foundation, "HTTPS Everywhere", last accessed 31st Jan. 2016, https://www.eff.org/https-everywhere, June 2010.

[10] Wladimir Palant - Eyeo GmbH, "Adblock Plus", last accessed 31st Jan. 2016, https://adblockplus.org/, April 2006.

[11] Atul Varma, "Lightbeam for Firefox", last accessed 31st Jan. 2016, https://www.mozilla.org/en-US/lightbeam/, July 2011.

[12] Inform Action - Open Source Software, "NoScript Suite", last accessed 31st Jan. 2016, https://noscript.net/, April 2006.

[13] HotCleaner, "Cookies", last accessed 31th Jan. 2016, https://chrome.google.com/webstore/detail/cookies/iphcomljdfghbkdcfndaijbokpgddeno?hl=en, October 8th 2015.

[14] Vano, "Cookies Manager+", last accessed 31th Jan. 2016, https://addons.mozilla.org/en-US/firefox/addon/cookies-manager-plus/, October 11th 2010.

[15] K. Johnson, S. Martin, J. O'Donnell and M. Winter, "Secret Program that began in 2007 extracts e-mail, audio, video, photos, documents, search history and logs.", last accessed 1st Feb. 2016, www.usatoday.com/story/news/2013/06/06/nsa-surveillance-internet-companies/2398345/, June 06 2013.

[16] Guardian Media Group, "Adblocking has an 'unsavoury' business model, says Trinity Mirror chief", last accessed 24th May 2016, http://www.theguardian.com/media/2016/mar/09/adblocking-business-model-trinity-mirror-guardian-media-group, March 09 2016.

[17] Wladimir Palant, "Allowing acceptable ads in Adblock Plus", last accessed 24th May 2016, https://adblockplus.org/development-builds/allowing-acceptable-ads-in-adblock-plus, December 05 2011.

[18] Raymond Hill, "Blocking Modes and Efficiency", last accessed 24th May 2016, https://github.com/gorhill/uBlock/wiki/Blocking-mode, March 16 2016.