STEPS TO SECURE WEB BROWSING

BACKGROUND

Web browsers pose a unique risk to enterprise infrastructure because of their frequent exposure to untrusted dynamic content. Configuring browser security settings is challenging due to uncertainty of both attack mitigation effectiveness and impact on end users. A key goal of this paper is to avoid impact to users while also mitigating as many attacks as possible. The following guidance uses a statistics-based approach to identify three mitigations in commonly-used web browsers that will ward off nearly all publicly known attacks. Further mitigations are provided in the advanced mitigations section for administrators seeking to defend against adversaries with significant resources.

ENABLE AUTOMATIC UPDATES

Research shows that 88% of publicly disclosed vulnerabilities are exploited within a day of release[1]. Administrator-driven manual patching often incurs significant lag time before patches are deployed. Adversaries take advantage of this lag time to exploit known vulnerabilities. Automatic updating limits the time available for attackers to exploit publicly disclosed vulnerabilities. Based on data calculated from the 2017 Common Vulnerabilities and Exposures (CVE®) database[2], automatic updating mitigates 91% of publicly known browser vulnerabilities[3].

Administrators are often hesitant to enable automatic updating out of fear that patches will break existing functionality. Browser patches are thoroughly tested by vendors before deployment making them unlikely to disrupt core functionality. Additionally, for administrators unwilling to enable automatic updates, administrators can internally test updates with a small subset of users in order to detect compatibility issues before performing a full deployment to their environment[4].

ENABLE REPUTATION SERVICES

Reputation-based blocking services (such as Microsoft SmartScreen®[5] or Google®[6] Safe Browsing) block browsers from accessing sites known to deliver malware. Most browsers offer reputation-based blocking as a core component, but blocking can also be achieved through the use of a trusted browser extension available through each browser's official extension repository. Reputation services continuously update from a variety of sources enabling adaptive protection against emerging threats. Reputation-based blocking prevents an average of 87.7% of socially engineered malware and phishing attempts (calculated from data produced by NSS Labs[7].) Third-party content networks are increasingly used to deliver malware from otherwise legitimate websites. Selective content blocking (e.g., ad-blocking through a browser extension or at the network boundary) should be used to prevent communication with malicious scripts and domains. Properly configured selective content-blockers can have a very low impact on daily web browsing and offer significant security and usability advantages.

-

DISABLE UNSAFE PLUGINS AND EXTENSIONS

Web browser plugins and extensions enrich web browsers by embedding extra features. Based on data calculated from the 2017 CVE database, browser plugins accounted for 34.5% of browser-related vulnerabilities[8]. Plugins and extensions expand the browser's attack surface and provide plugin vendors with access to sensitive browser information. Administrators should carefully consider the risks associated with each plugin and extension installed in their environment.

Browser plugins are third-party software that enable embedding proprietary content in the web browser. Administrators should also disable automatic plugin execution and remove all unnecessary plugins. Rather than automatic execution, administrators should require manual execution confirmation (e.g., click-to-play).

Browser extensions are installed within the web browser and often add new functionality. Unlike plugins, browser extensions cannot be easily selectively enabled/disabled. Administrators should investigate the impact of preventing unauthorized user installation of extensions and remove any unnecessary extensions. Most web browsers support extension whitelisting as well, which allows administrators to pre-approve a set of extensions that users are allowed to install.

ADVANCED MITIGATIONS

Well-resourced adversaries have the advantage of time and skill, enabling them to target weaknesses in even properly configured systems. Zero-day vulnerabilities can often defeat most web browser defenses, so administrators must add additional defensive layers in order to slow down exploitation and increase chances of detection. Additional mitigations for web browsers include browser isolation, disabling unnecessary features, and enabling operating system level mitigations.

ENABLE BROWSER ISOLATION

Browser isolation is a strategy that creates a logical barrier between the web browser and the operating system. This barrier decreases the impact of exploits by limiting malicious code to an ephemeral environment. Several browser isolation products exist, and administrators should consider how they can be implemented in their environments. In Windows 10®[9] Enterprise environments, Application Guard for Microsoft Edge can be used to enable browser isolation. Administrators should evaluate if the feature can be enabled in their environment and configure Application Guard using Group Policy settings[10]. Other browser virtualization products also exist. Cloud Browsers completely separate the web browser from the user's operating system by hosting the browser in a remote cloud environment and by then displaying the web content into a user's window on their system. While cloud browsers can offer enhanced security, they do come with increased costs and usability concerns, such as creating a single point of failure and throughput capacity. Administrators should evaluate if cloud browsers fit the requirements of their environment.

Site isolation loads each website in its own process which prevents malicious websites from accessing unauthorized data in the browser. Currently, the Google Chrome™[11] Site Isolation feature can be enabled through the Chrome Policy feature[12]. As more web browsers support this feature, administrators should analyze the impact of enabling site isolation in their environments.

DISABLE UNNECESSARY FEATURES

Some web browser features are not intended for wide spread use in a production environment, resulting in an unnecessarily large attack surface. Features which provide no benefit to the end user can sometimes be disabled to reduce overall risk. Determining necessary features is dependent upon each specific environment. Examples of unnecessary features may include hardware acceleration, rendering of untrusted fonts, and credential management. Additionally, legacy features such as VBScript and ActiveX®[13] controls give web browsers deeper system level functionality and should be disabled if administrators determine that they are not needed.

ENABLE OPERATING SYSTEM LEVEL MITIGATIONS

Protecting the browser should extend beyond the browsing environment itself. Building a layered defensive approach is critical to both limiting the impact of intrusions and preventing attacks. Web browsers are frequently exposed to untrusted content, so the operating environment must be configured with the assumption that defensive mitigations can fail. Automatic updates should be enabled for the host operating system as well as any available anti-exploitation features. Administrators should ensure that other defensive guidance, such as NSA's Top Ten Cybersecurity Mitigation Strategies[14], is enforced in their environment.