Analysis of the adoption of security headers in HTTP

William J. Buchanan, Scott Helme, Alan Woodward

IET Information Security 12 (2018) 118 – 126

HTTP 安全报头的使用分析

William J. Buchanan, Scott Helme, Alan Woodward

IET Information Security 12（2018）118-126

摘 要

随着网络系统内威胁数量的增加，需要采取更加综合的方法确保从服务器到客户端执行安全策略。这些策略旨在阻止中间人攻击，代码注入等。本研究分析了 HTTP 响应中使用的一些最新的安全选项，并通过 HTTP 响应对 Alexa 网站排行前 100 万的站点进行了扫描。这些扫描的选项包括：内容安全性策略，HTTP 的公钥固定扩展，HTTP 严格传输安全性和 HTTP 报头字段 X-frame-options，以便了解这些选项对最受欢迎网站的影响。结果表明，虽然执行参数不断增加，但仍未在许多顶级网站上实施。因此，研究显示了采用 Let's Encrypt 数百万个网站的数字证书，以及评估安全质量的方式头。

关键字：超媒体；公钥加密；传输协议；网站。

1 引言

使用 HTTP，我们有一个请求，比如 GET 请求，和服务器返回的一个响应。 这些响应包含标题包含参数列表中定义的信息键值对[1]。 有许多标准的应用程序层用于交换信息的协议，包括 HTTP[1]，SMTP[2]，FTP[3]和 DNS [4]。 这些规则通常是为了为支持简单的基于文本消息交换而定义的。其中一些是无状态的，例如 DNS 和 HTTP 协议，而 SMTP 和 FTP 则需要创建会话命令和响应。 当时，安全经常是后来才考虑的，通过在应用层协议的位置增加安全套接字来提高安全性层（SSL），例如 HTTPs [5]。

虽然 SSL 和传输层安全（TLS）纯粹保护消息交换的内容，增加了 CSP - 内容安全策略 - [6]集成了一套语言策略对 Web 资源进行内容限制以及定义了服务器如何将该策略传送给客户，以确保该策略被执行。同时还添加了安全扩展以防止中间人（MITM）攻击，例如 HTTP 公钥固定扩展（HPKP）[7]允许站点自行关联与特定的密码公钥并从而防止伪造数字证书。在 HTTP 严格传输安全性内（HSTS）[8] ，Web 服务器可以要求客户端（及其关联的网页浏览器）只能通过安全的方式与它进行交互连接（如使用 HTTPS）。在 2013 年，RFC 7034 定义 HTTP 报头字段 X-frame-options（XFO）[9]，它保护网页针对点击劫持的应用程序，以及针对整合的应用程序来自不可信来源的代码。重点再次放在服务器上通知客户其策略和配置选项。

本文分析了在 Alexa 网站排名 100 万中使用了这些新安全功能的用法的网站，以确定安全响应报头的影响。 可以看出，如图 1 所示，响应头包括：CSP，content-securitypolicy-report-only（CSPRO），public-key-pins（PKP），public-key-pins-report-only（PKPRO），x-content-type-only，XFO 和 X-Xssprotection（XXP）。 同时，采用 Let's Encrypt[10]显示了使用免费数字证书的一个有趣的动机。 因此，分析也将考虑采用它，并查看百万个网站是否使用 Let's Encrypt 权威证书（CA）。

2 背景

虽然 SSL/TLS 纯粹保护数据交换的内容，但 CSP[1]提供了可以设定的语言策略允许对 Web 资源的内容进行限制以及服务器如何将该策略传送给客户的方式，以便该策略能被正确执行。新的安全扩展也被添加进去以防止 MITM 攻击，如 HPKP [11]，它允许一个网站用与自己联系在一起的特定的密码公钥访问以防止伪造数字证书。 通过 HSTS [12]，Web 服务器可以要求 a 客户端（及其相关的网络浏览器）应该只与它交互通过安全连接（如使用 HTTPS）。

RFC 7034 添加了 HTTP 报头字段 XFO [13]，它可以保护针对定点劫持的网络应用程序，以及整合其中来自其他网页的内容。重点再次放在服务器上告诉客户其策略和配置选项。响应头文件包括：CSP，CSPRO，PKP，PKPRO，x-content-type-only，XFO 和 XXP。

总的来说，网络侧重于一个同源策略[14]，其中包含在同一个来源中的脚本仅被允许

访问数据在那个原点之内，因此每个原点都与其他原点隔离。不幸的是，这与开发者以及攻击者过度地限制了开发者使用清晰的技巧从其他域注入恶意代码。许多媒体网站也经常使用其他内容和脚本并且会努力支持限制他们的内容到他们自己的网站。 来自其他站点的代码的集成可能会导致以跨代码脚本（XSS）攻击为代码的问题在网页内往往是完全可信的。 在 CSP 中，我们拥有许多防止 XSS 的方法。 有了这个 CSP 支持资源的多个策略，可以在 Content-Security-Policy 报头或在<meta>元素内，例如[14]：

Content-Security-Policy: default-src https:

<meta http-equiv="Content-Security-Policy’content="default-src https:">

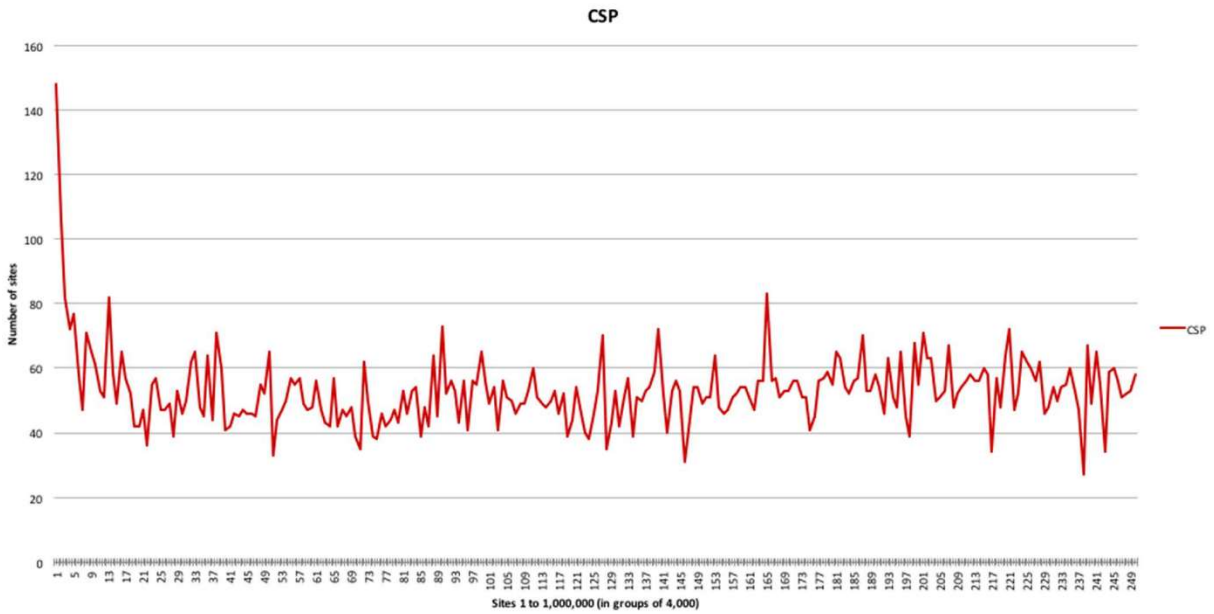

图 1.CSP (2017.5)

CSP 最强大的功能之一是定义白名单供客户使用。 因此常见的问题是 a 浏览器信任页面中的所有代码，以便来自其他站点的内容是可信的，因此，在 CSP 内部，Content-Security-Policy HTTP 标头用于定义可信来源，以及没有其他内容来源可以使用。例如，如果我们只需 https://asecuritysite.com 上的代码我们就可以使用：

Content-Security-Policy: script-src ‘self’

https://asecuritysite.com

在这种情况下，我们相信我们自己的源代码（’self’）和来自 https://asecuritysite.com 的任何脚本。 代码被注入时从另一个来源进入页面，会出现错误。 对于内容，我们也可以限制图片，媒体，插件，样式表和其他对象可以从使用安装指示：

- img-src. 定义图像的来源。
- media-src. 定义音频和视频的来源。
- object-src. 定义插件的来源，例如 AdobeFlash。
- style-src. 定义样式表的来源。
- font-src. 定义字体的来源。

• script-src. 定义所有脚本的来源。

默认情况下，所有指令都是打开的，所以如果指令不是设置它将允许所有的策略的这一部分。 如果有某些插件可能会损害页面，我们可以限制它们：plugintypes：并定义可以信任的插件类型浏览器。 例如，我们可以允许 PDF 和 Flash 插件：

Content-Security-Policy: plugin-types

application/pdf

Content-Security-Policy: plugin-types

application/pdf application/x-shockwaveflash

HTTPS 源内容现在通常比 HTTP 更受欢迎，因为它是可以保护内容免受攻击者的窥探，以及之间创建的安全隧道客户端和服务器。 因此，CSP 支持限制请求，以便 HTTP 请求被自动重写 HTTPS。 这被定义为 CSP 策略的一部分：

Content-Security-Policy: upgrade-insecure-requests

and which will rewrite this code:

<img src="http://asecuritysite.com/good.png">

<img src="https://asecuritysite.com/good.png">

CSP 的一个强大功能是客户端不能执行违反策略的操作，但会将其报告回服务器。为了这我们有一个报告 URI（report-uri :)来定义它的位置将报告发回。 我们也可以限制网页上的网址使用：base-uri:并限制嵌入的帧内容<base>元素；和 child-src:。 例如：

child-src https://youtube.com

可以嵌入来自 YouTube 的视频，但不能从 YouTube 中嵌入视频其他起源。 例如，我们可能会将 iframe 限制为 https：//youtube.com 与：

Content-Security-Policy: child-src https://youtube.com/

以下情况将被阻止：

<iframe src="https://fakeyoutube.com"><iframe>

一个示例攻击是重定向从 a 中发布的参数形成。 对于 CSP，我们可以通过以下方式限制终点：form-action:，这为<form>标签定义了端点。 而白名单有助于检测从不可信的运行的代码来源，XSS 中最大的威胁之一与内联有关脚本注入[15]，例如：

<script>someEvilCode();<script>

CSP 通过禁止任何形式的内联来克服脚本标记以及内联事件处理程序的问题，并且采取以下形式：

javascript: URL

3 文献回顾

3.1 内容安全策略

OWASP 每天都会增加网络安全风险的定义，他们的前 10 个风险[16]包括：破损

的认证和会话管理，XSS，不安全的直接对象引用，安全配置错误，敏感数据暴露，缺失功能级访问控制和跨站请求伪造（CSRF）。 随着这个不同的方法被定义为了定义一个风险评分[17]。 现在许多公司集中精力排名在 OWASP 前 10 位，它们存在其他风险，例如 HTTP 标题可以用作隐藏信息的来源[18]。

Web 应用程序漏洞越来越受到关注，特别是对于 XSS 和 CSRF 来说，已经引发了一个呼吁没有漏洞的应用程序行动[19]。 在真实中世界，这可能是困难的，特别是在松散的前端，中间件和后端 Web 基础架构之间的耦合。 CSP 因此整合到一个分层的方法中, 安全性和使用内容限制以及内容限制执法计划。

研究人员创建了一个 CSP[20] 成功实现了四种削弱 XSS 攻击的范围的浏览器。 他们认为，XSS 攻击可能是：

- ●持久 XSS。 这种类型将脚本存储在服务器上，例如在数据库中，并且每次关联的页面都会运行访问。
- ●非永久 XSS。 这种类型隐藏了一些恶意脚本和欺骗用户运行有害脚本，以窃取认证 cookie 或数据[21]。
- ●基于 DOM 的 XSS。 这种类型的攻击修改了 DOM 一个网页的结构，以便它运行在一个代码中包含的方式。

使用 CSP，他们使用了[21]中的一系列 XSS 来模拟攻击 Chrome,Firefox,Safari 和 Opera 的 50 种不同的攻击类型。他们的结果显示 37 次成功针对 Firefox，19 针对 Chrome，Safari 和 Opera 的攻击，而使用 CSP 降低了所有攻击的成功率。

与 Web 开发中的许多进步一样，CSP 的采用速度缓慢，问题依然存在等与不安全的服务器端 JavaScript 生成一样。 在[22]中，作者提出 PreparedJS 是 CSP 的扩展。 它使用一个安全的脚本模板机制和一个轻量级的脚本校验和方案。 在[23]中，Google 分析了 CSP 识别的缺陷可以绕过 94.72％的所有不同的策略，并使用 a 搜索引擎语料库从超过 1 个大约 1000 亿页十亿个主机名。 这 1,680,867 个部署了 CSP 的主机 26,011 项独特的 CSP 策略。 在他们的论文中，他们确定了三个绕过 CSP 的常用方法。 他们的关键因素之一认定为弱点是 75.81％的不同策略使用 a 白名单过于宽松，可用于绕过 CSP。

在 28 个欧盟国家，发现共同存在脆弱性和弱点，并通过改进防御机制，往往集中在热门网站。另外 Chen [25] 在 2 年期间对 18,000 个欧洲网站进行了分析，并分析了客户端的采用安全方法，并发现财务和教育部门在他们的采用中是最成功的。

3.2 扫描

Alexa 网站的扫描是一个明确的理解请求的变化性质的方法，比如在 HTTP 网页的不断变化的行为，例如，其中[26]发现了包括多媒体在内的大页面的趋势内容和动态内容创作。 在[11]中，作者们有概述了包含 HTTP 请求的消息的和使用模式发现它们在数量，字段名称和字段上差别很大值。 其他协议，例如 SMTP 也已被显示泄漏信息[12]以

及发送时使用的元数据电子邮件可以用来确定可能导致的系统数据妥协。 对于[13]，研究人员分析了 HTTPS 来自 Censys 的证书以及证书透明度（CT）日志。 在这项工作中，他们调查了前 100 万 Alexa 网站，他们发现汇总的 CT 日志和 Censys 快照占 99％。

最近的工作涉及大规模网络基础设施的脆弱性分析，如中国[27]使用 57,112 网络脆弱性事件，并创建新的威胁模型了解现代漏洞[28]。 不幸的是，安全报头的使用似乎还需要一段时间才能完成对最受欢迎的网站产生重大影响。

随着 XSS 攻击的威胁越来越大，Ying 和 Li [29]分析了 100 个最受欢迎网站的 CSP 采用率，发现 2015 年 1 月和 6 月只有 8％的人使用他们发现采用了 0.066 和 0.133％的较大扫描 2015 年 1 月和 6 月，增幅为 73％。

3.3 加密

数字证书有很多风险，包括人工-中间人攻击和假证书[30]。他们的一个主要问题是，通常购买昂贵，因此 Let's Encrypt 计划看起来会创建一个 CA 生态系统的免费使用和自动化证书签名[31]。 它是专注于 CT [10]。 在[32]中，作者分析了 CT 日志中的数据超过 1800 万份证书。这使用诸如 Censys，Alexa 的历史记录，地理位置数据库和 VirusTotal，在他们的结果是只有 54％的域使用了们已经获得的证书，并且发生了很多配置错误的服务器，以及使用 Let's Encrypt 在恶意软件网站中使用的证书。

3.4 *HTTP* 公钥固定

早在 2011 年，一家名为 DigiNotar 的荷兰 CA 就被黑客入侵[33]。 后获得进入他们的系统，黑客设法使他们的系统通往 CA 服务器的路，并发布 500 多个流氓证书指向他们自己。 包含在这些证书中的是一个 google.com 域。 这个证书被用来启动一个 MITM 包括谷歌服务在内的 30 万伊朗用户的攻击 Gmail。 从最终用户的角度来看，攻击是无法察觉的。浏览器收到域名的证书，这是有效的信任链完好无损。 他们拥有安全的所有指标连接，但它已被妥协。

在 2008 年以及 2011 年，StartCom CA 违反[34]。 由安全研究员进行的 2008 年违规行为导致在为 paypal.com 发行的无赖证书 verisign.com 域名。 2011 年的违约行为由一家公司进行不明身份的攻击者导致，非常接近皇冠上的宝石，StartCom 根密钥。 通过访问 StartCom 根密钥，攻击者可以为他们的任何域生成证书像，不仅如此，它还会导致必要的撤销和重新签发每一份 StartCom 证书。

目前，有数百个 CA 能够发布数字证书，并可能导致欺诈性证书。PKP 最初由 Google 声明，使网站所有者能够定义该网站对他们的网站有效的证书，并在 RFC 中定义 7469 [7]。 在[35]中，作者认为这很困难被危险使用，因为它可以很容易地阻止大部分网站访问。 一个问题是记忆效应，一个针一旦设置将在一段时间内保持有效。

在[36]内研究人员概述了实施 HPKP，以及 Web 服务器通知他们的客户他们必须记住（或’插入’）他们的公钥。 他们概述了 HSTS 和 HPKP 是实施 HTTPS 连接并允许的

新功能，证书通过 HTTP 绑定，但通过率一直很差，并且在安全性方面实施也很薄弱。为此，他们已经发现了针对 HSTS 的可能攻击 HPKP。 在 2015 年，Kranch 和 Bonneau [37]分析了 HSTS 和 KPKP，并找到证据表明没有被开发人员和许多网站所理解的非固定资源的加载问题，这可能是用于劫持页面，以及固定域可能泄漏的位置 cookie 值。

4 方法

本文概述了 Alexa 的两个主要扫描 2015 年 8 月和 2017 年 5 月对 CSP 前 100 万进行扫描仪检查以下标题：CSP，CSPRO。 这个回应头文件可以被开发人员用来了解策略执行效果以及返回一个 JSON 文档，以及 HTTP POST 定义的 URI，X-Webkit-CSP（XWC）和 XCSP。

表 1. A Capture from the first few headers for a scan in May 2017

Site from	Site to	Skipped	CSP	CSPRO
1	4000	591	148	45
4001	8000	551	106	18
8001	12,000	622	82	19

```
// The value to award headers.
$values = array("strict-transport-security" => 25,
                "content-security-policy" => 25,
                "public-key-pins" => 30,
                "x-frame-options" => 20,
                "x-xss-protection" => 20,
                "x-content-type-options" => 20);
```

图 2. Value to award headers

表 2. % of the total possible score used as a criteria for each grade awarded

% of total achieved	Grade awarded
0-13	F
14-28	D
19-49	D
50-59	C
60-74	B

75-95	A
95-100	A+

在[20]中，作者发现 CSP 头受 Firefox 版本 23，Chrome 版本 25，Safari 版本 7 和 iOS Safari V7.1 及更高版本支持，而 XCSP 和 XWebKit-CSP 头文件在早期的浏览器版本中使用。通常，为了确保兼容性，管理员配置了 XCSP 和 CSP 两者标题。 对于其他头扫描，以下是检测到：PKP，PKPRO，严格传输安全（STS），X-contenttype-option（XCTO），XFO，XXP，X-download-options（XDO），和 X 许可的跨域策略（XPCDP）。

除此之外，扫描仪还记录了 HTTP 的重定向 HTTPS，例如 google.com 或 facebook.com 的情况，抓取工具将默认为 http://（domain）并跟随重定向直到完成。 其他指标也被捕获：access-control-allow-origin: 这个头文件可以用来定义它只有来自原始网站的内容可以被允许； Let's Entrypt：这个定义证书是否由 Let's Encrypt 生成； 和 securityheaders.io 等级：此提供者为安全提供评分头。

这项工作的一个关键目标是了解当我们通过一百万个站点时安全标题的使用趋势，。因此，我们将其分组到 4000 个服务器并为每个服务器计算标题范围。 表 1 显示了扫描的头几个标题在 2017 年 5 月捕获。

标题的评分基于安全性好处他们提供和易于部署（https：//securityheaders.io/）。 作为安全利益或困难增加，分数增加。 这是奖励的价值（图 2）。

到目前为止，最难以部署的标题是 HPKP，它得分最高。它如果没有部署 HPKP，则无法对 A+进行高价值评分。 接下来是 HSTS 和 CSP，由于有一些部署考虑和提供一个大量的保护它们的得分都相同。 'X'标题的分数都相同因为它们很简单，足以部署并提供有价值的服务保。 这些设计使得网站可以轻松地提升自己从低评分开始鼓励进一步改进，但是 a 需要不同的努力来实现 A，其中除了所有标题，HPKP 是必需的，而 A+则需要所有的头文件目前并妥善部署。

这给 HTTPS 站点的总分为 140，而 HTTP 由于 HPKP 和 HSTS 被忽略，网站总分为 85,浏览器通过不安全的连接传递。 每个网站返回的是被授予一个不区分大小写匹配的分数以及是否找到标题然后检查值的正确语法的测试报头。 如果这些检查通过，那么该网站被授予与该标题相关的分数。如果网站是重复的，则不能再次为同一标题评分标题。 授予该网站的等级取决于他们获得和使用的总可能得分的百分比，标准如表 2 所示。

在这里报告中，每次抓取 Alexa 前 1 百万是用一个单一的约 12 小时完成的服务器。服务器将前 100 万名单分 250 个列出 4000 个站点，然后同时运行 250 个搜寻器线程大大缩短抓取所有 100 万个网站所需的时间。该爬行程序是用 PHP 编写的，并使用 cURL 库进行请求，结果存储在 MySQL 数据库中。这是在模拟托管的 Intel x64 系统的虚拟服务器上执行由 Digital Ocean 运行 Ubuntu 14.04。 Alexa 的数据是作为逗号分隔的文件提供，格式为 rank,hostname，例如 1,google.com 和 2,youtube.com 等等。搜寻器默认通过

HTTP 与网站进行通信,通过简单地连接'http：//'字符串和主机名在 Alexa 数据"http://hostname"中提供。爬虫将会发出这个初始的 GET 请求，然后遵循所有重定向完成时请求设置为 10 秒，并模拟一个现代浏览器通过设置用户代理字符串的值：

'Mozilla/5.0 (Windows NT 10.0; WOW64)

AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/52.0.2743.82 Safari/537.36'.

一旦爬虫收到最终响应，它就会解析该爬虫响应并搜索匹配头部的字符串。 如果响应包含正在搜索的其中一个标题，则为在数据库中的适当条目上标记。

重要的是，一个网站只有被标记为支持 HTTPS 才将这些客户端请求从 HTTP 重定向到 HTTPS,少数用户将以 HTTPS 身份进入部分 URL 被寻找,除非 HTTP 默认为 HTTPS. 作者认为 HTTPS 是无效的，因此不应该被计算在内。在仅报告的版本中，例如 CSPRO 和 PKPRO，响应头可以被开发人员用来理解策略实施的效果以及发送 JSON 文档的位置作为 HTTP POST 返回到定义的 URI。 这项工作确实记录了这些标题为完整性，但我们没有附加重大意义如果他们被使用的话，他们会指出网站的安全级别没有其他适当的标题。

5 结果

2017 年 5 月，Alexa Top One Million 网站的扫描时间<12 小时。表 3 和图 3 概述了结果。 图 3 概述了扫描在 4000 组内的结果。总的来说，我们可以看到一个高点，随后急剧下降，然后在排名下降时保持稳定的尾巴。 在五月 2017 年，CSPRO 和 PKP 都看到了弱势的结果，且被采纳率<1%。 结果显示数量大幅增加部署了 CSP 的网站，并在该网站上有了健康的增长也部署了 PKP 策略的站点数量。 另一个网站数量大幅增加，部署的网站数量增加了 387.2%,增加了测试 CSPRO 策略的网站数量可能显示许多组织现在正在考虑部署完整版 CSP。 除此之外，还有 302%的增长发布 STS 策略的网站数量增加了重定向到 HTTPS 的网站数量超过 236%。

表 3. Results(Auguest 2015 and May 2017)

	August 2015	August 2015,%	May 2017	May 2017,%	% change
CSP	1365	0.1476	13,253	1.5736	870.92
CSPRO	211	0.0228	1028	0.1221	387.20
XWCSP	183	0.0198	362	0.0430	87.81
XCSP	304	0.0329	921	0.1094	202.96
PKP	148	0.0160	6624	0.7856	4375.68
PKPRO	21	0.0023	87	0.0103	314.29
STS	11,308	1.2231	45,527	5.4057	302.61
XCTO	44,315	4.7933	89,053	10.5739	100.95
XFO	55,042	5.9536	93,601	11.1139	70.05
XXSSP	41,948	4.5373	70,032	8.3154	66.95

XDO	192	0.0208	7134	0.8471	3615.63
XPCDP	346	0.0374	6993	0.8303	1921.10
HTTPS	64,043	4.7108	208,710	24.7815	236.40

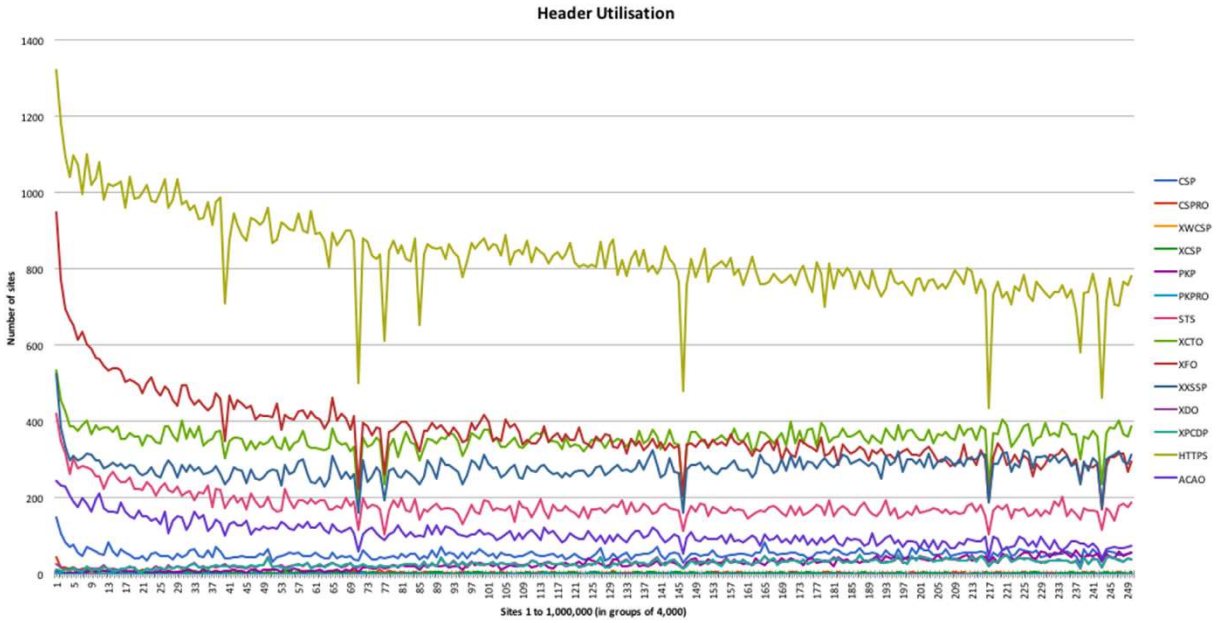

图 3. *Header utilisation (May 2017)*

图 1 显示了 2017 年 5 月扫描的 CSP 图。与模拟器的结果唯一的区别是规模在 Y 轴上有所增加。 可以绘制相同的比较为所有其他头，甚至为部署 HTTPS。

HTTP 公钥固定（HPKP）仅在 6624 处找到的 1,000,000 个网站（图 4），并且是利用率最低的网站之一, 基于安全的 HTTP 响应头（0.79％）。 越来越多对于排名较低的网站采用 PKP 被认为与该网站有关在 Tumblr 网站内采用 PKP。

最广泛使用的非’X’头是严格的运输安全，并在 45,527 个网站上使用，达到 5.41％使用率（图 5）。 对于最多的 100 万人来说，这仍然相当低参观。 有趣的是，在 100 万个网站中，有 208,710 个网站（24.78％）正在其领域积极重定向到 HTTPS，这会留下 163,183 个重定向到 HTTPS 的域，但是不用 STS 强制执行。

对于更常见的’X’标题，有一个明确的标题这些和剩下的标题之间的区别使用它们的网站数量（图 6）。 XFO 标题是迄今为止在 93,601 个网站（11.11％）中使用的最普遍的基于安全性的标题，并且与其他网站相同，显示出相同的下降趋势。有趣的是 XCTO 和 XXSSP 标题都显示出来在 89,053（10.57％）和 70,032（8.31％）的同样高的使用率下，但是，在他们最初的下降后，他们实际上显示了一个上升的趋势，当你下移名单，这是违背了所有其他标题的趋势。

搜索器正在寻找的其他内容之一就是如果您加载的话，有多少域将重定向到 HTTPS 他们通过 HTTP 在第一个请求上（图 7）。 如上所述此前，共有 208,710 个网站重定向到 HTTPS（24.78％），我们可以看到下降趋势与我们下移名单的域名。

绝大多数网站都获得了 F 级成绩 securityheaders.io 扫描。 这意味着他们要么不发

布任何基于安全的 HTTP 头，或者他们做了，而他们不是正确配置。 表 4 强调了差距有多大。你的排名越高，你就越有可能获得 F 级成绩底部可见的更好等级的尖峰。

正如预期的那样，排名越高，加密证书的使用率越高（图 8）。 还有一个因素让我们不会颁发加密 EV 证书，所以可能不能迎合一些网站。 参考文献[32]扫描了 Alexa 网站从 2016 年 7 月 29 日至 8 月 29 日，发现百万的 8%包含 Let's Encrypt 网站

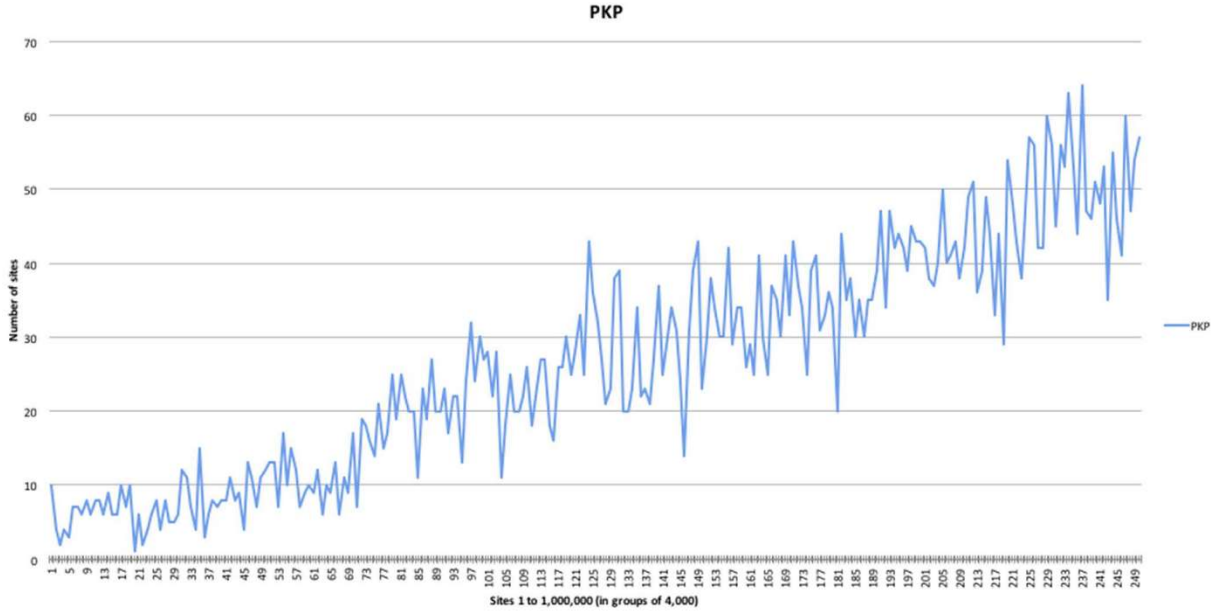

图 4. *PKP*

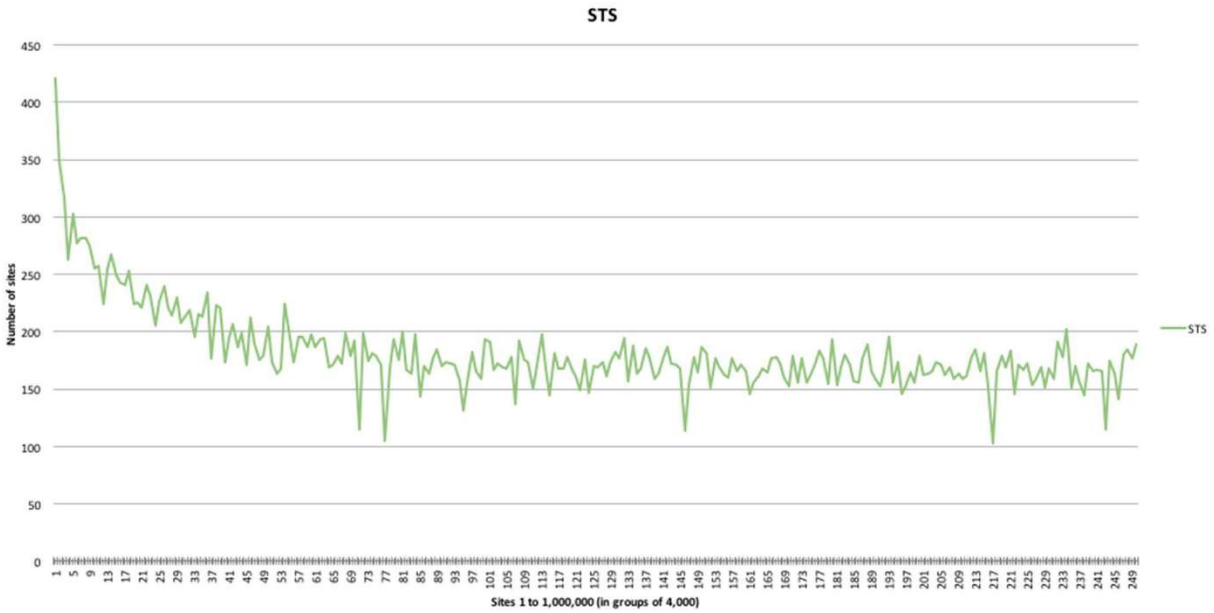

图 5. *STS*

的网站，比例为 30%Comodo 和 17%GeoTrust。 在他们的常见爬网方法中，他们发现在 24%的网站中使用了 Let's Encrypt，与 Comodo 在 21%。 他们发现，Alexa 顶部的 1.2%百万，2017 年 5 月，已经从付费 CA 转移到 Let's 加密，而我们发现 5.29%。 在图 8 中，我们看到采用 Let's Encrypt 证书对于顶级网站来说比较弱。 在表 5 中，我们看到在顶级站点采用 Let's Encrypt 是相当薄弱，只有 0.6%的收养，而附近的网站排名前 100

万的网站已经和通过率在 4.5％和 5.8％之间。

6 结论

随着安全报头使用量的增加，我们会看到 CSP 和 CSPRO 使用量的增加。总的来说，我们在报告中看到了安全报头的使用在增加，去多公司也可能会开始研究安全报头的使用。这也许是 CSP 部署还没用同它自身具有同等的重要性[35]。

重定向到 HTTPS 的网站数量正在增加快速。 securityheaders.io 分数有点差，但有很多机会可以通过简单的改进来完成更简单的基于 X 的头文件。 为了让我们加密证书，我们看到了对顶级站点的影响很大， Top 4000 站点采用率为 0.6％，而排名前 100 万个网站则上涨到 5.3％。 结果与[32]相符，但是作为一般收入增加的趋势， 我们将排名降至 4.5％至 5.8％之间。

未来的工作将继续提高评分系统，并且也可以访问网站上的多个页面。 目前的方法只访问网站的主页，并没有抽样其他页面。

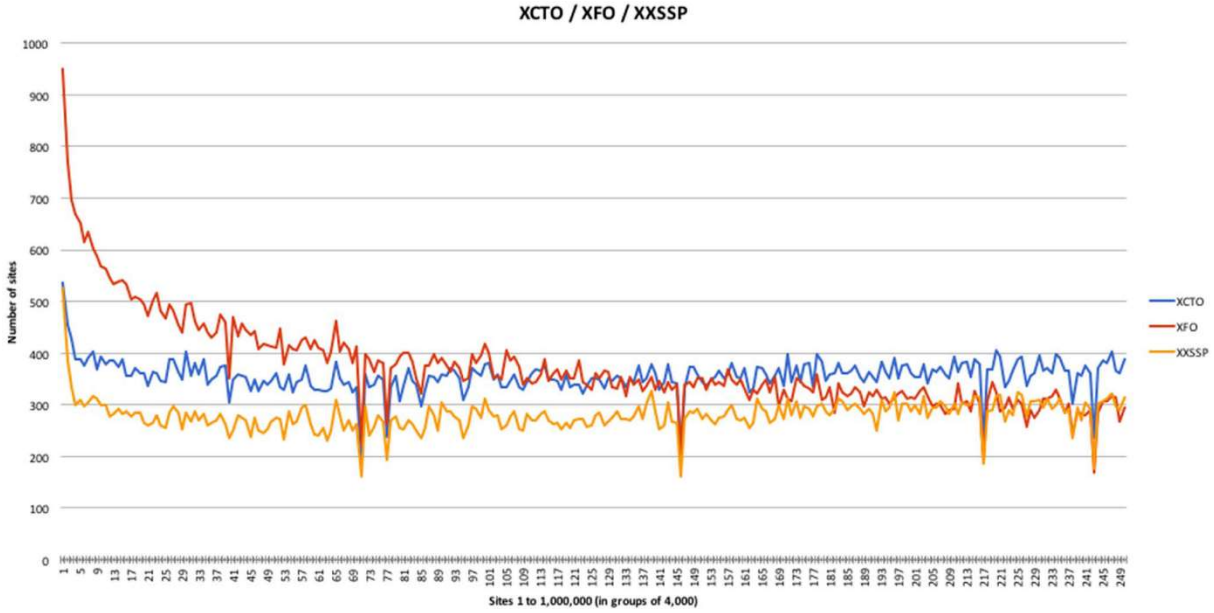

图 6. *X headers*

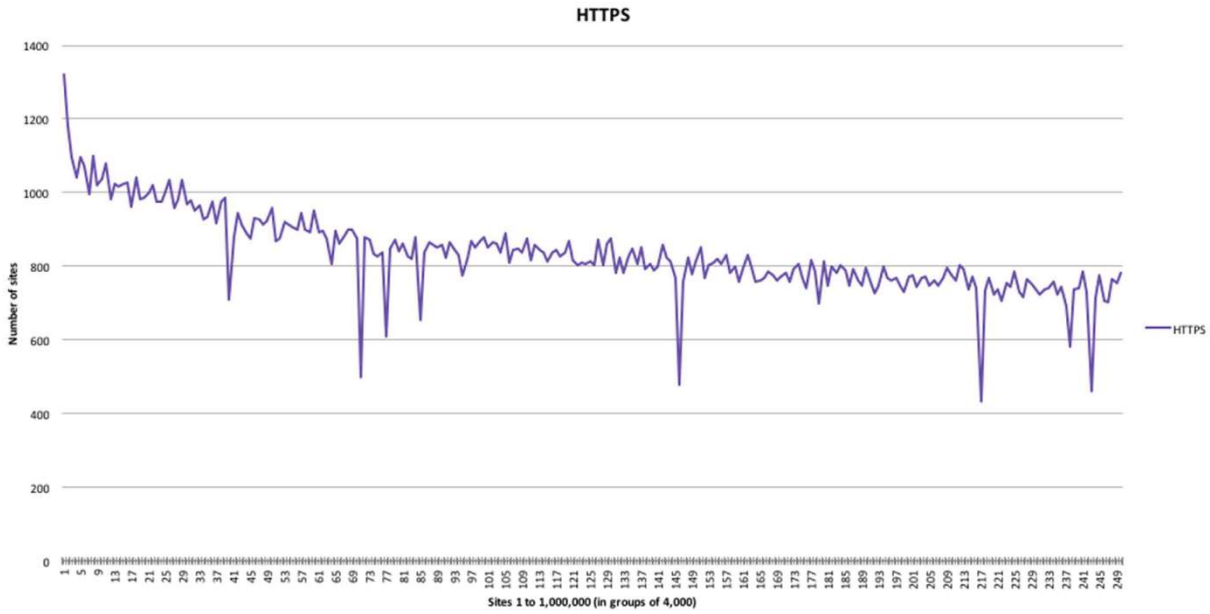

图 7. *HTTPS redirection*

表 4. Security header scores

Security headers	%
A+	0.0071
A	0.1003
B	0.2318
C	3.2153
D	5.6908
E	8.0063
F	82.7401

表 5. Let’s Encrypt certificate distribution

Site position	Number	Percentage	Site position	Number	Percentage
1-4000	25	0.6	900,001-904,000	197	4.9
4001-8000	62	1.6	904,001-908,000	197	4.9
8001-12,000	77	1.9	908,001-912,000	199	5
12,001-16,000	87	2.2	912,001-916,000	195	4.9
16,001-20,000	93	2.3	916,001-920,000	180	4.9
20,001-24,000	103	2.4	920,001-924,000	199	5
24,001-28,000	77	1.9	924,001-928,000	207	5.2
28,001-32,000	103	2.6	928,001-932,000	208	5.2
32,001-36,000	105	2.6	932,001-936,000	210	5.3
36,001-40,000	105	2.6	936,001-940,000	194	4.9
40,001-44,000	125	3.1	940,001-944,000	206	5.2
44,001-48,000	119	3	944,001-948,000	191	4.8
48,001-52,000	124	3.1	948,001-952,000	144	3.6
52,001-56,000	127	3.2	952,001-956,000	232	5.8
56,001-60,000	122	3.1	956,001-960,000	191	4.8
60,001-64,000	109	2.7	960,001-968,000	233	5.8

64,001-68,000	117	2.9	964,001-968,000	189	4.7
68,001-72,000	144	3.6	968,001-972,000	113	2.8
72,001-76,000	152	3.8	972,001-960,000	203	5.1
76,001-80,000	109	2.7	976,001-980,000	224	5.6
80,001-84,000	133	3.3	980,001-984,000	182	4.6
84,001-88,000	110	2.8	984,001-988,000	205	5.1
88,001-92,000	129	3.2	988/,001-992,000	211	5.3
92,001-96,000	124	3.1	992,001-996,000	207	5.2
96,001-100,000	138	3.5	996,001-1,000,000	213	5.3

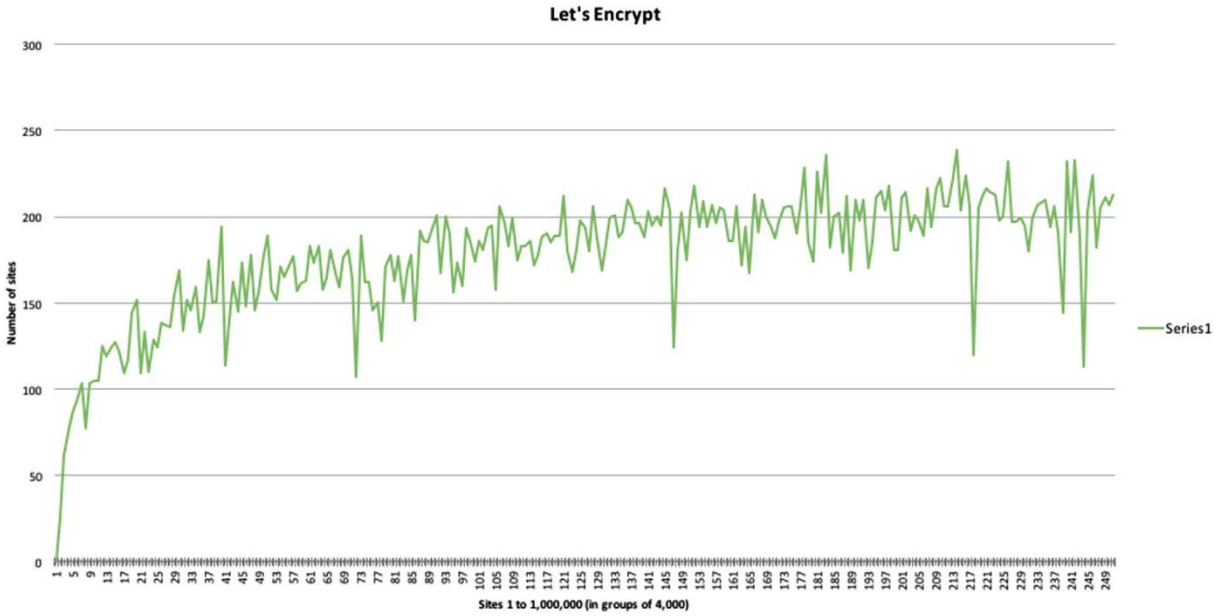

图 8. *Let’s Encrypt*

7 参考文献

[1] Fielding, R., Gettys, J., Mogul, J., et al.: ‘RFC 2616 – hypertext transferprotocol – HTTP/1.1’. Society [Internet], 1999, no. 2616, pp. 1–114.Available at: http://www.ietf.org/rfc/rfc2616.txt

[2] Klensin, J.: ‘RFC 5321 – simple mail transfer protocol’. IETF RFC, 2008

[3] Postel, J., Reynolds, J.: ‘RFC 959 – file transfer protocol’. Rfc 959 [Internet],1985, pp. 1–69. Available at: https://www.ietf.org/rfc/rfc959.txt

[4] Mockapetris, P.: ‘Domain names – implementation and specification[Internet]’. Request for Comments, 1987, pp. 1–55. Available at: https://www.ietf.org/rfc/rfc1035.txt

[5] Rescorla, E.: ‘RFC 2818 – HTTP over TLS’. Network Working Group, IETF,2000. p. pp. 1–8

[6] Sterne, B., Barth, A.: ‘Content security policy 1.0 [Internet]. W3C. 2012’.Available at http://www.w3.org/TR/CSP/

[7] Bash, E.: ‘RFC7469 public key pinning extension for HTTP’. PhD Propos,2015, vol. 1, pp.

1–28

[8] Hodges, J., Jackson, C., Barth, A.: ‘HTTP strict transport security’. Availableat http//tools.ietf.org/html/rfc6797. 2012

[9] Gondrom, T.: ‘HTTP header field X-frame-options’, IETF Standard, 2013.Available at: https://tools.ietf.org/html/rfc7034

[10] Hodson, H.: ‘A little privacy, please’. New Sci [Internet], 2014, vol. 224, no.2997, p. 24. Available at: http://www.sciencedirect.com/science/article/pii/S0262407914622843

[11] Calzarossa, M.C., Massari, L.: ‘Analysis of header usage patterns of HTTPrequest messages’. Proc. – 16th IEEE Int. Conf. on High PerformanceComputing and Communications, HPCC 2014, 11th IEEE Int. Conf. onEmbedded Software and Systems, ICESS 2014 and 6th Int. Symp. onCyberspace Safety and Security, 2014, pp. 847–853

[12] Nurse, J.R.C., Erola, A., Goldsmith, M., et al.: ‘Investigating the leakage ofsensitive personal and organisational information in email headers’, J.Internet Serv. Inf. Secur. [Internet], 2015, 1, (February), pp. 70–84. Availableat: https://www.cs.ox.ac.uk/files/7181/jisis2015_nurse_et_al.pdf

[13] VanderSloot, B., Amann, J., Bernhard, M., et al.: ‘Towards a complete viewof the certificate ecosystem’. Proc. of the 2016 ACM on InternetMeasurement Conf., 2016, pp. 543–549

[14] Mozilla: ‘Content-security-policy – HTTP|MDN [Internet]’. Available athttps://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

[15] Fogie, S., Grossman, J., Hansen, R., et al.: ‘XSS attacks: cross site scriptingexploits and defense [internet]’. Management, 2007, p. 482. Available athttp://portal.acm.org/citation.cfm?id=1534243

[16] Owasp: ‘OWASP top 10 – 2013 [Internet]’. OWASP Top 10. 2013. Availableat http://owasptop10.googlecode.com/files/OWASPTop10-2013.pdf

[17] Owasp: ‘OWASP risk rating methodology [Internet]’. Owasp. 2013, pp. 1–5.Available at https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

[18] Dhobale, D.D., Ghorpade, V.R., Patil, B.S., et al.: ‘Steganography by hidingdata in TCP/IP headers’. ICACTE 2010 – 2010 3rd Int. Conf. on AdvancedComputer Theory and Engineering, Proc., 2010

[19] Stamm, S., Sterne, B., Markham, G.: ‘Reining in the web with contentsecurity policy’, Proc. 19th Int. Conf. World Wide Web WWW 10 [Internet],2010, no. 2, p. 921. Available at http://portal.acm.org/citation.cfm?doid=1772690.1772784

[20] Yusof, I., Pathan, A.S.K.: ‘Mitigating cross-site scripting attacks with acontent security policy’, Computer (Long Beach Calif.), 2016, 49, (3), pp. 56–63

[21] Yusof, I., Pathan, A.S.K: ‘Preventing persistent cross-Site scripting (XSS)attack by applying

pattern filtering approach'. 2014 the 5th Int. Conf. onInformation and Communication Technology for the Muslim World, ICT4M2014, 2014

[22] Johns, M.: 'Script-templates for the content security policy', J. Inf. Secur.Appl., 2014, 19, (3), pp. 209–223

[23] Weichselbaum, L., Spagnuolo, M., Lekies, S., et al.: 'CSP is dead, long liveCSP! on the insecurity of whitelists and the future of content security policy'.Proc. 23rd ACM Conf. on Computer and Communications Security, Vienna,Austria, 2016

[24] Van Goethem, T., Chen, P., Nikiforakis, N., et al.: 'Large-scale securityanalysis of the web: challenges and findings'. Lecture Notes in ComputerScience (including subseries Lecture Notes in Artificial Intelligence andLecture Notes in Bioinformatics), 2014, pp. 110–126

[25] Chen, P.: 'Longitudinal study of the use of client-side security mechanisms onthe European Web'. [cited 2017 May 26]. Available at http://www2016.net/proceedings/companion/p457.pdf

[26] Pries, R., Magyari, Z., Tran-Gia, P.: 'An HTTP web traffic model based onthe top one million visited web pages'. 8th EURO-NF Conf. on NextGeneration Internet, NGI 2012 – Proc., 2012, pp. 133–139

[27] Huang, C., Liu, J., Fang, Y., et al.: 'A study on Web security incidents inChina by analyzing vulnerability disclosure platforms', Comput. Secur., 2016,58, pp. 47–62

[28] Devi, G., Bal, R.S., Priyadarsini Sahoo, P.: 'Threats indentification in webapplication', J. Netw. Commun. Emerg. Technol., 2016, 6, (6), pp. 4557–4573

[29] Ying, M., Li, S.Q.: 'CSP adoption: current status and future prospects. securcommun networks [Internet]'. 2016 Oct 20 [cited 2016 Nov 15]. Available at http://doi.wiley.com/10.1002/sec.1649

[30] Bradbury, D.: 'Digital certificates: worth the paper they're written on?',Comput. Fraud Secur., 2012, 2012, (10), pp. 12–16

[31] Schuster, S., van den Berg, M., Larrucea, X., et al.: 'Mass surveillance andtechnological policy options: improving security of private communications',Comput. Stand. Interfaces, 2017, 50, pp. 76–82

[32] Manousis, A., Ragsdale, R., Draffin, B., et al.: 'Shedding light on theadoption of let's encrypt'. arXiv Prepr arXiv161100469, 2016

[33] Leyden, J.: 'Inside 'Operation black tulip': digiNotar hack analysed[Internet]'. The Register. 2011. Available at http://www.theregister.co.uk/2011/09/06/diginotar_audit_damning_fail/

[34] SecurityWeek: 'StartSSL flaw allowed attackers to obtain SSL cert for anydomain|SecurityWeek.Com [Internet]'. Available at http://www.securityweek.com/startssl-flaw-allowed-attackers-obtain-ssl-cert-anydomain

[35] Ristic, I.: ‘Is HTTP public key pinning dead? – network security blog|Qualys,Inc’. [Internet]. Available at https://blog.qualys.com/ssllabs/2016/09/06/ishttp-public-key-pinning-dead

[36] de los Santos, S., Torrano, C., Rubio, Y., et al.: ‘Implementation state ofHSTS and HPKP in both browsers and servers’. Int. Conf. on Cryptology andNetwork Security, 2016, pp. 192–207

[37] [Kranch, M., Bonneau, J.: ‘Upgrading HTTPS in Mid-Air: an empirical studyof strict transport security and Key pinning’. [cited November 2017].Available at http://www.jbonneau.com/doc/KB15-NDSShsts_pinning_survey.pdf HTTPS in Mid-Air- An Empirical Study of StrictTransport Security and Key Pinning.pdf