doi :10.3969/j.issn.1671-1122.2012.12.008

Android 平台上基于 WebKit 引擎的安全浏览器的设计与实现

吴昊[1]，文伟平[1]，严寒冰[2]

（1. 北京大学软件与微电子学院信息安全系，北京 102600 ；
2. 国家计算机网络应急技术处理协调中心，北京 100029）

摘 要 ：文章针对当前网络攻击技术尤其是网络钓鱼攻击的技术，提出并建立了攻击模型。针对攻击模型，文章设计了移动终端安全浏览服务的模型和恶意网址黑名单更新策略，对安全浏览服务及其客户端 / 服务器端的通信协议及相关数据结构进行了详细的阐述。在安全浏览服务协议的基础上，文章设计并实现了 Android 平台上的安全浏览器，为 Android 平台用户提供了有价值的安全浏览应用服务。

关键词 ：Android 操作系统 ；WebKit ；安全浏览器 ；反钓鱼

中图分类号 ：TP393.08 文献标识码 ：A 文章编号 ：1671-1122（2012）12-0025-04

Design and Implementation of a Safe Browser based on WebKit Engine on Android Platform

WU Hao[1], WEN Wei-ping[1], YAN Han-bing[2]

(1. *Department of Information Security, SSM, Peking University, Beijing* 102600*, China;*
2. *National Computer Network Emergency Response Technical Team / Coordination Center, Beijing* 100029*, China*)

Abstract: Cyber-attacks, especially phishing attacks were studied, and a phishing attack model was established. A model of secure mobile terminal browsing service and the updating policy of phishing URL blacklist were designed. The details of secure browsing service, the client/server communication protocols and the data structures were described. This part is the important foundation to achieve a secure browser application. On the basis of secure browsing protocol, a complete secure browser application was implemented. The browser provides a very useful secure browsing application service for Android platform users.

Key words：android operation system; webKit; secure browser; anti-phishing

0 引言

随着移动通信技术的发展，使用手机浏览器访问互联网已成为潮流和趋势。浏览器的核心也叫浏览器的内核，或者是引擎。目前主流的浏览器引擎技术有如下几种 ：Chrome 和 Safari 所使用的 WebKit ；FireFox 使用的 Geeko ；IE 的 Trident 引擎以及 Opera 浏览器使用的 Presto。

其中，WebKit 是最为优秀的一种浏览器引擎，桌面电脑操作系统的浏览器 Chrome 和 Safari 都使用 WebKit 作为内核 ；此外，很多嵌入式设备，例如智能手机、PDA、电视机顶盒、掌上游戏机等，也越来越多的倾向于使用 WebKit 作为自己的浏览器内核。WebKit 是由苹果公司发布的一个开源的浏览器引擎，拥有清晰的源代码结构、极快的渲染速度等特点，不仅能更快地打开显示网页，而且也能支持最新的 HTML5 标准草案。手机上的浏览器大部分都是采用 WebKit 作为引擎，例如 Google 的智能手机操作系统 Android OS 内置的 Chrome Lite 浏览器，Apple 的 iPhone，Nokia 的 S60 浏览器等。可以看到，WebKit 在现在的手机浏览器占据着市场主导位置。

另一方面，随着移动互联网技术的飞速发展，传统的针对桌面操作系统的攻击手段也逐渐在移动设备上进化开来。总结来看主要有以下方面 ：1）数据丢失或数据泄露导致的信息安全问题 ；2）各种恶意软件攻击的风险，黑客利用用户安全意识淡薄，在其手机上安装可窃取信息的恶意软件盗取用户的各种隐私信息 ；3）个人移动应用面临恶意站点的威胁，进一步带来信息安全威

收稿时间 ：2012-08-12

基金项目 ：国家自然科学基金资助项目 [61170282]

作者简介 ：吴昊（1987-），男，硕士研究生，主要研究方向 ：系统与网络安全 ；文伟平（1976-），男，湖南，副教授，博士，主要研究方向 ：网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术等 ；严寒冰（1975-），江西，男，高级工程师，博士，主要研究方向 ：计算机网络安全等。

胁；4）Android 平台的开放性也带来了恶意应用程序增多的结果。随着 HTML5 技术标准的不断改进完善，尤其是 HTML5 的本地存储、离线缓存、地理位置信息、传感器控制以及 JavaScript 的广泛应用，移动浏览器的 Web 页面对移动终端自身的潜在安全威胁也越来越严重。在 Android 平台上，各种钓鱼攻击、跨站脚本攻击、恶意应用下载、用户隐私信息非法截获等攻击层出不穷。

综上所述，研究 Android 平台上浏览器的安全技术，开发出能有效预防常见 Web 攻击、保障用户信息安全的浏览器具有非常显著的价值。论文首先对 WebKit 引擎和浏览器技术进行分析总结，对常见的网络钓鱼技术进行研究分析，总结网络钓鱼攻击的手段和在移动终端平台上的攻击特点，提出并建立了攻击模型。针对攻击模型，论文设计了移动终端安全浏览服务的模型和恶意网址黑名单更新策略，对安全浏览服务及其客户端 / 服务器端的通信协议及相关数据结构进行了阐述。在安全浏览服务协议的基础上，论文设计并实现了Android 平台上的安全浏览器。

1 WebKit 浏览器引擎

WebKit 源自于 KDE 小组开发的 KHTML。Apple 凭借自身的技术和资金实力，开发了以 WebKit 引擎为核心的 Safari 浏览器，打下了一定的用户基础。

简要的说，WebKit 由三个模块组成：JavaScriptCore、WebCore 和 WebKit。WebKit 作为了整个项目的名称。图 1 为 WebKit 的结构图。

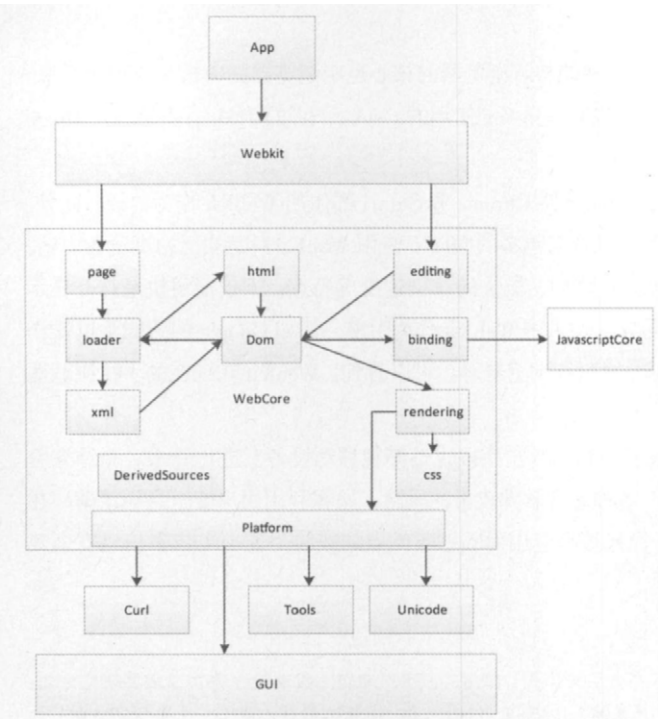

图1 WebKit结构示意图

WebKit 进行解析的一般过程如下：1）CURL 获得网站的 stream ；2）解析划分字符串 ；3）通过 Dom Builder 按合法的 html 规范生成 Dom 树 ；4）如果有 JavaScript，JSEngine 就通过 ECMA-262 标准完善 Dom 树 ；5）把 Dom 传给 Layout Engine, 进行布局，如果有 CSS 样式，就通过 CSS Parser 解析 ；6）最后解析渲染出来。

2 移动终端钓鱼攻击技术分析

钓鱼式攻击（Phishing，与钓鱼的英语 Fishing 发音一样）是指不法分子利用各种手段，仿冒真实网站的 URL 地址以及页面内容，或者利用真实网站服务器程序上的漏洞在站点的某些网页中插入危险的 HTML 代码，以此来骗取用户银行或信用卡账号、密码等私人资料 [1]。带有该种诱骗色彩的钓鱼式攻击方式的网站统称为钓鱼网站，该类型网站的特点是：内容虚假，常伴随虚假的用户中奖类信息 ；仿冒内容，常冒充正规网站的登录页 ；含有病毒、恶意脚本等有害信息。

研究总结 Phishing 的攻击手段是防范和打击这类网络攻击的基础。目前互联网上常见的 Phishing 欺骗攻击手段主要有如下几种：

1）利用网站界面和域名极其类似，诱骗粗心用户“上钩”[2] ；2）使用键盘监控程序窃取信息 ；3）利用虚假的电子商务网站进行诈骗 ；4）利用搜索引擎引诱、欺骗受害者 [3] ；5）利用浏览器漏洞窃取个人信息 ；6）假冒系统升级和安装补丁程序 ；7）使用“社会工程”（Social Engineering）手段进行攻击 [4]。

移动终端设备，尤其是智能手机和平板电脑设备的性能和功能越来越强，使用群体越来越大，攻击者也逐渐盯上了这一新型用户群体。但是同传统 PC 设备上 Windows 操作系统一家独大有所不同，移动终端目前主要采用苹果公司的 iOS、谷歌公司的 Android 和微软公司的 Windows Phone 等几大平台，攻击者难以使用单一的恶意程序或攻击手段就覆盖大多数用户，因此攻击成本相对较高，攻击案例呈现出相对单一化的特点。这也促使安全研究人员要更早涉足移动设备安全性的研究，防患于未然。在前面章节对网络钓鱼攻击研究的基础上，本节总结出在移动终端设备上的钓鱼攻击的一般流程。

这个过程的具体描述如下：1）钓鱼攻击者通过短信息（SMS）、彩信、电子邮件、WAP PUSH 消息等手段或者利用社交工程的手段将欺诈信息传递给用户的手机、平板电脑等移动设备 ；2）用户收到并阅读移动智能设备上收到的带有欺诈性质的信息后，产生误解 ；3）用户对欺骗信息做出错误反应，执行可能使自己掉入钓鱼陷阱的操作，比如点击电子邮件中的假冒网站、向钓鱼攻击者填写回复隐私信息等 ；4）根据用户的反应，用户的移动终端向恶意站点发出相应的访问请求 ；5）恶意站点服务器根据用户移动设备的访问请求，响应虚假

钓鱼攻击内容；6）用户移动设备收到恶意站点回馈的钓鱼信息，提示用户输入具有经济价值的隐私信息；7）用户输入个人隐私信息。比如个人姓名、银行账户信息、购物网站账户信息，以及手机号码均属于具有经济价值的隐私信息；8）用户的隐私信息回传给钓鱼网站服务器，甚至直接回传给攻击者；9）攻击者获取用户隐私信息，从而获取商业利益并进行下一步的攻击，以获得更有价值的信息[5]。

3 安全浏览服务及客户端的设计与实现

3.1 安全浏览服务设计

从前面章节对钓鱼攻击等网络攻击手段的研究，我们可以提出一种安全浏览服务，在用户使用浏览器或其它客户端程序访问恶意网址和钓鱼网站时，为终端用户提供安全警告，以及为客户端程序采取进一步的防护操作提供安全判断凭据。

3.1.1 协议设计

从上层来看，安全浏览服务应该维护着动态更新的钓鱼网站和恶意站点的网址列表，在浏览器客户端程序将要加载网站内容时，检查目标站点地址是否在此恶意网址数据库中。为了最大限度地精确区分任意的恶意 URL 网址，使用“主机名后缀 / 路径前缀”的形式来代表恶意网址。这种方式能够有效地应对攻击者使用大量不同网站域名和子页面的攻击手段。例如“ google.com/ ”、“ some.host.com/123/ ”以及“ otherhost.net/some/url.html?q=123 ”都是有效的“主机名后缀 / 路径前缀”格式。需要注意的是，主机名后缀必须是完整的主机地址，所以“ host.com/ ”并非是“ otherhost.com ”的后缀；此外，如果目标 URL 中包含了查询参数（Query Parameters），比如上述的第三个示例，表达式也必须完整匹配目标 URL。如果每次访问 URL 都将 URL 串发送给安全浏览服务器进行检查，这样做既不高效也不经济（尤其在移动数据通信费用高昂的情况下），还会有侵犯用户隐私之嫌。因此安全浏览协议采取了将恶意网址黑名单数据库在客户端保存备份的方法，浏览器客户端程序定期与服务器同步，更新黑名单数据库。更新数据库的流程将在后面详细说明。

为了压缩黑名单数据库的体积，在进行黑名单更新时，浏览器客户端程序实际上并不接受传输包含完整的“主机后缀 / 路径前缀”的表达式列表，而是在服务器端对每个表达式进行 Hash 散列，在得到的 32 个字节 (256 bits) 大小的 Hash 结果中，截取前 4 个字节传输给浏览器客户端。浏览器客户端程序在需要检查目标 URL 是否在黑名单中时，先对 URL 地址的所有“主机后缀 / 路径前缀”表达式进行 Hash 计算。比如，如果浏览器即将加载 URL 地址为“ http://www.host.com/service/login.html ”的页面，该 URL 所对应的表达式“ host.com/ ”和“ host.com/service ”均需进行 Hash 计算。在对表达式进行 Hash 计算获得 32 字节大小的结果串后，分别截取前 4 字节，与保存在客户端本地的黑名单数据库进行匹配。

因为在进行匹配时使用 4 字节的 Hash 值前缀可能会发生碰撞（Collision），即不同的 URL 对应的前 4 个字节的 Hash 值可能是相同的。所以，在 4 字节 Hash 值前缀 - 黑名单发生匹配时，浏览器客户端必须去请求安全浏览服务的服务器端来获取相应的完整的 32 字节 Hash 值，进行第二次匹配。如果第二次匹配成功，那么说明目标 URL 的确是黑名单里的恶意网址，浏览器根据匹配结果进行访问拦截或对用户发出警告。

3.1.2 数据格式设计

在前面的协议设计中已经说明，浏览器客户端从服务器端下载的黑名单数据被分成了数据块，每个块包含 4 字节的 Hash 前缀。这里设计出两种类型的数据块：“ Add ”型和“ Sub ”型，分别用来向客户端添加新的 Hash 前缀以及删除无效的或误报的 Hash 前缀，形象地说，这两种类型的数据块分别作“加法”和“减法”。分块式黑名单设计有两大优点：

1）浏览器客户端能够增量式地下载黑名单数据，完整的恶意网址黑名单的数据体积可能多达几十兆字节大小，这给使用相对较慢的移动通信网络的用户带来了比较大的数据通信开销和下载时间开销，而增量式下载是按需下载，能够在完全满足用户检测需求的情况下，将数据传输的开销降到最小。

2）服务器端能够更加灵活地向浏览器客户端推送数据块。众所周知，钓鱼攻击者所使用的网址 URL 通常寿命很短，而且多变。使用分块数据，服务器就能够及时地将最新的黑名单数据传送给客户端。

3.1.3 恶意网址黑名单更新策略设计

当浏览器需要更新存储在本地的安全浏览服务数据时，就使用 HTTP 连接先向安全浏览服务更新服务器发送当前本地已有的数据块的编号。下面是一个更新请求的示例：

phish-shavar:a:20-30,49

phish-shavar:s:10-15

这个请求表明，当前浏览器客户端拥有 phish-shavar 黑名单列表中的“ add ”类型的第 20 至第 30 个，以及第 49 个数据块；另外也包含“ sub ”类型的第 10 至第 15 个数据块。

如果浏览器客户端当前还没有任何数据块（比如浏览器安装后第一次启动运行时），就需要使用下面的请求格式：

phish-shavar:

更新服务器在收到浏览器客户端的请求后，对其响应。但是，更新服务器的响应并不直接包含黑名单数据块，而是响应一系列的包含黑名单数据下载地址的位于内容服务器上的跳转 URL(Redirect URL)，在这个内容服务器上才存放着“ add ”和“ sub ”类型的数据块。此外，更新服务器也能够通过响应，来引导浏览器客户端删除那些失效的黑名单数据。

为了增强安全性，更新服务器对浏览器客户端的响应数据 (基于HTTP) 和发回给浏览器的跳转URL数据都会进行数字签名，浏览器客户端会对接收到的数据的数字签名进行认证，从而预防数据篡改 (比如中间人攻击)。

在上面的具体步骤中 ：1) 浏览器客户端向更新服务器发出更新请求，请求中包含了当前客户端本地的黑名单数据包状态 ；2) 更新服务器进行响应，响应数据中包含了数据块实际存放地址的 URL 列表，以及删除当前本地无效数据的命令 ；3) 浏览器客户端收到响应数据后，依次访问数据块URL，从内容服务器上请求最新的黑名单数据块 ；4) 内容服务器响应浏览器的下载请求，向浏览器客户端传输最新的黑名单数据块。

3.2 安全浏览器客户端的设计与实现

3.2.1 浏览器客户端的架构

浏览器系统结构共分为四层 ：

第一层为应用层，在此层面构建实现面向用户的应用 ；第二层为插件及第三方库 ；第三层为 WebKit 平台移植所需的接口层 ；第四层为平台和操作系统支持层，提供 WebKit 平台所需的软硬件资源。

根据移动终端用户一般的使用场景，对安全浏览器的功能进行了设计和模块划分，主要包含以下功能模块 ：书签管理、黑名单拦截报警、主页面、网页加载、下载管理、历史记录。

3.2.2 URL检查模块

URL 检查模块是实现安全浏览服务的核心部分，此模块的功能和实现也相对比较复杂，主要有以下功能单元 ：

1) URL 规范化处理

进行 URL 检查前需要先对 URL 进行规范化处理，转化成“ 主机名后缀 / 路径前缀 ” 形式的表达式。规范化的处理过程如下 ：(1) 删除 URL 字符串中所有的“ tab ” (0x09)、“ CR ” (0x0d)、“ LF ” (0x0a), 注意对这些特殊符号转码而产生的“ %0a ” 要保留 ；(2) URL 尾部的锚标签需要被删除，比如“ http://abc.com/#frag ” 需要被处理成“ http://abc.com/ ” ；(3) 对 URL 字符串反复进行解码，确保字符串中不再有十六进制字符 ；(4) 规范化主机名字符串 ；(5) 规范化路径字符串 ；(6) 对经过上述步骤处理得到的字符串中包含的非常规字符 (ASCII 值小于 32 或大于 127, “ # ”, “ % ”) 进行 UTF-8 转码。

规范化处理完成后，为了不漏掉任何潜在的恶意网址，还需要额外地枚举查询所有符合正则匹配条件的表达式。

2) 对规范化 URL 进行 MD5 Hash 计算

Hash 散列计算是 URL 检查模块中进行黑名单匹配的判断标准，在应用中借助 Java 语言中的报文摘要类 (java.security.MessageDigest) 进行实现。

3) 与本地数据库匹配

在 URL 查询的过程中可能有二次匹配的过程。

客户端应用使用了开源项目 greenDAO[6] 来实现在 Android 上的对 SQLite 数据库的 DAO (Data Access Objects) 操作，从而进行数据库查询。

3.2.3 恶意网址处理模块

如果确认用户即将访问的目标网址 URL 位于黑名单数据库中，浏览器客户端就需要及时进行处理。一般情况下，为了提高用户体验，浏览器需要遵循用户的选择或偏好设置。因此，在本项目中浏览器首先查询用户的配置文件，判断用户对恶意网址处理的设置 ；如果用户没有进行配置，就在发生危险情况时，浏览器及时对用户发出警告 ；然后根据用户的选择进行下一步的处理，继续加载恶意网址或拦截恶意网址。

由于对恶意网址拦截与否直接关系到移动终端用户的信息安全，对警告信息的表述就至关重要，在浏览器应用中采取了目前 FireFox 浏览器、Chrome 浏览器也采用的危险警告 ：“ 警告 ：可疑的钓鱼网站页面。当前页面可能是其它网站的仿冒站点，这通常被用来欺骗访问用户，并以此套取其个人隐私和财产信息。在此页面上输入任何个人信息都可能导致身份泄露或其它不可预料的信息滥用事件的发生 [7]。”

4 结束语

论文研究了浏览器的 WebKit 引擎的相关技术，研究了当前网络攻击技术尤其是网络钓鱼攻击的技术。进一步设计了移动终端安全浏览服务的模型，对安全浏览服务及其通信协议进行了阐述，设计了相关的数据结构和恶意网址黑名单更新策略。在此基础上，论文设计并实现了安全浏览器客户端应用及服务器端环境，为 Android 平台用户提供了安全浏览的应用服务。（责编 杨晨）

参考文献 ：
[1] 陈达 . 网络钓鱼的现状、方式及防范初探 [J]. 网络安全技术与应用，2006，(07) ：35-37.
[2] 焦永鑫 . 基于浏览器插件的网络钓鱼防范研究 [D]. 吉林大学硕士学位论文，2011.
[3] 梁雪松 . 基于浏览器的钓鱼网站检测技术研究 [J]. 信息安全与通信保密，2007，(11) ：53-55.
[4] 任传伦，杨义先，冯朝辉 . 网络钓鱼攻击的发展趋势及法律对策考虑 [J]. 网络安全技术与应用，2007，(06) ：86.
[5] 焦永鑫 . 基于浏览器插件的网络钓鱼防范研究 [D]. 吉林大学硕士学位论文，2011.
[6] GreeenDAO team. Green DAO Introduction[EB/OL].http://greendao-orm.com/documentation/introduction/, 2011.
[7] Anti-Phishing Working Group. Active Phishing Warning[EB/OL]. http://www.antiphishing.org/, 2012.