



(12)发明专利

(10)授权公告号 CN 106228040 B

(45)授权公告日 2018.11.13

(21)申请号 201610550722.3

(22)申请日 2016.07.13

(65)同一申请的已公布的文献号
申请公布号 CN 106228040 A

(43)申请公布日 2016.12.14

(73)专利权人 成都知创信息技术有限公司
地址 610000 四川省成都市高新区天府大道中段801号5幢8层

(72)发明人 罗意 王小虎 石涵 王春鹏
赵晨晖

(74)专利代理机构 成都信博专利代理有限责任公司 51200
代理人 张辉 崔建中

(51)Int. Cl.
G06F 21/12(2013.01)

(56)对比文件

CN 103150493 A,2013.06.12,
CN 1967553 A,2007.05.23,
US 2011/0167121 A1,2011.07.07,
CN 103377326 A,2013.10.30,

审查员 贾东曜

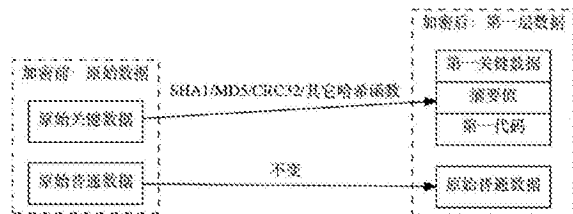
权利要求书1页 说明书4页 附图1页

(54)发明名称

一种网页源码三层加密方法

(57)摘要

本发明公开了一种网页源码三层加密方法,主要包括以下步骤:算出原始关键数据的摘要值,抹除原始关键数据中几个字符成为第一关键数据;改写第一层关键数据成第二关键数据;用正则表达式将其中有意义的词语片段替换成无意义的占位字符;将第三层数据传给浏览器;浏览器收到第三层数据后,通过转换对应关系用正则表达式转换为第二层数据;浏览器自身的Javascript引擎将第二层数据中的第二关键数据解析为第一层关键数据;通过第一代码根据摘要值与第一关键数据穷举得到原始关键数据。本发明使用高效灵活的混淆策略、动态生成的极小化字符集以及强制穷举三层技术,在正常访客无感知的前提下,极大增加攻击者分析与破解网页源代码的难度。



1. 一种网页源码三层加密方法,其特征在于,包括以下步骤:

页面中的数据分为原始关键数据和原始普通数据,对页面中的原始关键数据,用摘要算法算出摘要值,抹除原始关键数据中几个字符成为第一关键数据;编写通过第一关键数据与摘要值能穷举出原始关键数据的第一代码;此时,经处理后的页面中的数据称为第一层数据;将第一关键数据、摘要值、通过第一关键数据与摘要值能穷举出原始关键数据的第一代码,以及页面原始普通数据传入下一层;

对传入的第一层数据,区分出第一层关键数据和第一层普通数据,用极小化的JavaScript字符集改写第一层关键数据成第二关键数据;此时,经处理后的页面中的数据称为第二层数据;第二关键数据与第一层普通数据传入下一层;

对传入的第二层数据,用正则表达式将其中有意义的词语片段替换成无意义的占位字符,得到混淆数据,并保存这种转换对应关系为转换表;整个页面中的第二层数据经过替换变成了第三层数据;所述第三层数据包括混淆数据和转换表;

三层加密完成后,将第三层数据传给浏览器;

浏览器收到第三层数据后,通过转换对应关系用正则表达式转换为第二层数据;浏览器自身的Javascript引擎自动将第二层数据中的第二关键数据解析为第一层关键数据,与第一层普通数据合并为第一层数据;在第一层数据中,通过第一代码根据摘要值与第一关键数据穷举得到原始关键数据,与原始普通数据合并为原始数据。

2. 如权利要求1所述的一种网页源码三层加密方法,其特征在于,所述摘要算法为SHA1、MD5或者CRC32。

3. 如权利要求1所述的一种网页源码三层加密方法,其特征在于,所述摘要算法为自定义哈希算法。

4. 如权利要求1或2或3所述的一种网页源码三层加密方法,其特征在于,在穷举出原始关键数据的第一代码时,将穷举难度控制在平均时间1秒内,抹除的原始关键数据的字符为3~4个。

一种网页源码三层加密方法

技术领域

[0001] 本发明涉及网络中网页加密领域,特别涉及一种网页源码三层加密方法。

背景技术

[0002] 网页代码是指在网页制作过程中需要用到的一些特殊的“语言”,设计人员通过对这些“语言”进行组织编排制作出网页,然后由浏览器对代码进行“翻译”后才是最终看到的效果。通常,使用的保护源代码的技术是通过一些软件,将源代码在不改变执行结果的前提下,以一定方式,改写为对真人来说不便于阅读的形式,以此增加攻击者分析代码的难度,达到保护源代码的效果。

[0003] 网页源代码要交由访客的浏览器来执行,一个网站的页面代码能轻松地被恶意攻击者获取、分析,进而产生出能自动攻击网站的程序,对网站的正常访问造成极大威胁。因为方式单一且通常长期不更新算法,遇到针对性的攻击时,防御效果非常有限。

发明内容

[0004] 本发明所要解决的技术问题是提供一种网页源码三层加密方法,使用高效灵活的混淆策略、动态生成的极小化字符集以及强制穷举三层技术,在正常访客无感知的前提下,极大增加攻击者分析与破解网页源代码的难度。

[0005] 为解决上述技术问题,本发明采用的技术方案是:

[0006] 一种网页源码三层加密方法,包括以下步骤:

[0007] 页面中的数据分为原始关键数据和原始普通数据,对页面中的原始关键数据,用摘要算法算出摘要值,抹除原始关键数据中几个字符成为第一关键数据;编写通过第一关键数据与摘要值能穷举出原始关键数据的第一代码;此时,经处理后的页面中的数据称为第一层数据;将第一关键数据、摘要值、通过第一关键数据与摘要值能穷举出原始关键数据的第一代码,以及页面原始普通数据传入下一层;

[0008] 对传入的第一层数据,区分出第一层关键数据和第一层普通数据,用极小化的JavaScript字符集改写第一层关键数据成第二关键数据;此时,经处理后的页面中的数据称为第二层数据;第二关键数据与第一层普通数据传入下一层;

[0009] 对传入的第二层数据,用正则表达式将其中有意义的词语片段替换成无意义的占位字符,得到混淆数据,并保存这种转换对应关系为转换表;整个页面中的第二层数据经过替换变成了第三层数据;将混淆数据和转换表一起传入下一层;

[0010] 三层加密完成后,将第三层数据传给浏览器;

[0011] 浏览器收到第三层数据后,通过转换对应关系用正则表达式转换为第二层数据;浏览器自身的Javascript引擎自动将第二层数据中的第二关键数据解析为第一层关键数据,与第一层普通数据合并为第一层数据;在第一层数据中,通过第一代码根据摘要值与第一关键数据穷举得到原始关键数据,与原始普通数据合并为原始数据。

[0012] 进一步的,所述摘要算法为SHA1、MD5或者CRC32。

[0013] 进一步的,所述摘要算法为自定义哈希算法。

[0014] 进一步的,在穷举出原始关键数据的第一代码时,将穷举难度控制在平均时间1秒内,抹除的原始关键数据的字符为3~4个。

[0015] 与现有技术相比,本发明的有益效果是:1)混淆能增加人工分析代码的难度,挡住网络上大量使用的没有针对性的自动程序;2)通过极小化JavaScript字符集改写代码中重要数据后,攻击者很难写出针对性的程序让攻击自动化;3)强制客户端对关键数据进行短时间的穷举,能大量消耗攻击者的硬件资源,拖慢攻击速度;4)三层加密中都有可以动态随机生成的部分,攻击者无从分析,分析了也会很快过时。

附图说明

[0016] 图1是本发明方法中第一层加密强制穷举示意图。

[0017] 图2是本发明方法中第二层加密最小化Javascript字符集示意图。

[0018] 图3是本发明方法中第三层加密混淆示意图。

具体实施方式

[0019] 下面结合附图和具体实施方式对本发明作进一步详细的说明。尽管本发明将结合一些具体实施方式进行阐述和说明,但需要注意的是本发明并不仅仅只局限于这些实施方式。相反,对本发明进行的修改或者等同替换,均应涵盖在本发明的权利要求范围当中。

[0020] 本发明方法不但沿用现有技术的混淆,抵御无针对性的自动攻击,还使用一种极小化的字符集改写代码中部分重要数据,这种改写不但对人不友好,对除JavaScript以外的计算机语言也极不友好;对代码中小部分的关键数据,可以故意擦除几个字符,但提供数字摘要强制客户端进行短时间的穷举。

[0021] 页面中的数据分为原始关键数据和原始普通数据,对页面中的原始关键数据,用摘要算法算出摘要值,抹除原始关键数据中几个字符成为第一关键数据;编写通过第一关键数据与摘要值能穷举出原始关键数据的第一代码;此时,经处理后的页面中的数据称为第一层数据;将第一关键数据、摘要值、通过第一关键数据与摘要值能穷举出原始关键数据的第一代码,以及页面原始普通数据传入下一层;

[0022] 对传入的第一层数据,区分出第一层关键数据和第一层普通数据,用极小化的JavaScript字符集改写第一层关键数据成第二关键数据;此时,经处理后的页面中的数据称为第二层数据;第二关键数据与第一层普通数据传入下一层;

[0023] 对传入的第二层数据,用正则表达式将其中有意义的词语片段替换成无意义的占位字符,得到混淆数据,并保存这种转换对应关系为转换表;整个页面中的第二层数据经过替换变成了第三层数据;将混淆数据和转换表一起传入下一层;

[0024] 三层加密完成后,将第三层数据传给浏览器;

[0025] 浏览器收到第三层数据后,通过转换对应关系用正则表达式转换为第二层数据;浏览器自身的Javascript引擎自动将第二层数据中的第二关键数据解析为第一层关键数据,与第一层普通数据合并为第一层数据;在第一层数据中,通过第一代码根据摘要值与第一关键数据穷举得到原始关键数据,与原始普通数据合并为原始数据。

[0026] 对本发明方法举例说明如下:

[0027] 1、对页面中的关键数据K,用任一摘要算法(包括SHA1、MD5、CRC32等常见摘要算法,甚至自定义的哈希算法也可以)算出摘要值D,抹除关键数据K中几个字符(具体几个取决于想要的穷举的难度,为了不影响用户体验,将穷举难度控制在平均时间1秒内比较好,这时抹除3~4个字符就可以)成为数据K2,将K2、D、通过K2与D能穷举出K的代码,以及页面其它数据传入下一层。

[0028] 2、对传入数据中的部分重要数据,用极小化的JavaScript字符集改写后成为数据F,与页面其它数据传入下一层。

[0029] 3、对所有传入数据,用正则表达式将其中有意义的词语片段替换成无意义的占位字符,并将这种转换的对应关系保存为T,此时整个页面的数据经过替换变成了R。将R、T以及通过T还原R的代码一起传入下一层(这一步对数据还有压缩效果)。

[0030] 4、三层加密完成,将数据传给浏览器。浏览器通过转换表T用正则表达式转换得到大部分数据;用极小化JavaScript字符集改写的代码因为本身就是JavaScript代码,所以也能被浏览器正确得到;最后根据数字摘要与部分关键数据穷举还原出整个关键数据。这样就得到了全部数据。

[0031] 另外,可以按一定条件,如时间、访问者、被访问次数等,重新随机生成或选择摘要算法、极小化JavaScript字符集和正则表达式。

[0032] 关于极小化JavaScript字符集的说明:因为JavaScript语法强大的兼容性,我们可以只用极少的几个字符写出与原JavaScript代码等效的代码。表1中就展示了使用“+”“-”“~”“[”“]”“{”“}”“/”“!”9个字符表示JavaScript中一些常见值的一种方式(最少可以只用6个字符)。能表示0和1就能表示所有数字,有数字就能用ASCII码转化出所有字符,因此任意JavaScript代码都可以用这种极小化JavaScript字符集表达。

[0033] 表1JavaScript中的值与用极小化字符集表示后的对应关系

[0034]

JavaScript中的值	用极小化字符集表示
0	~~[]
1	--[]
true	!+[]
false	![]
NaN	+{ }
undefined	[] [{ }
Infinity	~[] / -[]
[object object]	{ }

[0035] 这种表示方法非常多变,能生成大量不重复的转换方案供随机选择,且利用了JavaScript的语言特性,其它程序语言根本分析不了。

[0036] 以上具体实施方式和附图仅为本发明之常用实施例。显然,在不脱离权利要求书所界定的本发明精神和发明范围的前提下可以有各种增补、修改和替换。本领域技术人员应该理解,本发明在实际应用中可根据具体的环境和工作要求在不背离发明准则的前提下在形式、结构、布局、比例、材料、元素、组件及其它方面有所变化。因此,在此披露之实施例仅用于说明而非限制,本发明之范围由后附权利要求及其合法等同物界定,而不限于此前

之描述。



图1

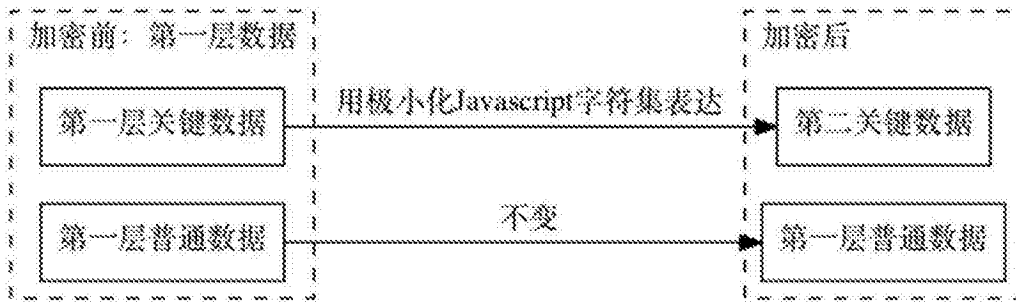


图2

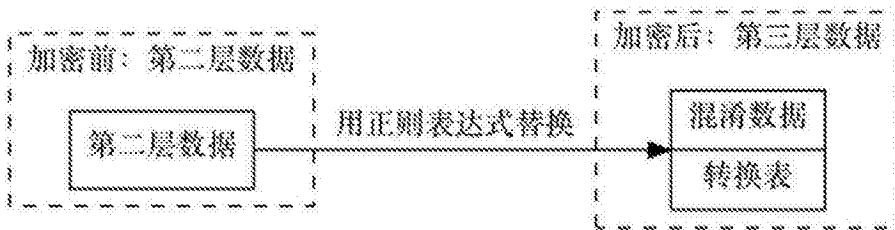


图3