

原型链污染漏洞防护方法及装置、介质、设备

技术领域

[01]本说明书一个或多个实施例涉及应用安全技术领域，尤其涉及一种原型链污染漏洞防护方法及装置、介质、设备。

背景技术

[02]在 JavaScript 中通过原型链来实现对象继承，因此如果修改了一个对象的原型，那么会影响到所有来自于这个原型的对象，这就是原型链污染。原型链污染会造成拒绝服务攻击即 DoS、远程代码/命令执行漏洞攻击即 RCE、属性注入漏洞攻击等，威胁 Node 应用安全。

发明内容

[03]本说明书一个或多个实施例描述了一种原型链污染漏洞防护方法及装置、介质、设备，能够避免原型链污染而造成的漏洞攻击。

[04]根据第一方面，提供了一种原型链污染漏洞防护方法，包括：检测请求接口是否接收到请求；若是，则启动在所述请求接口预先注入的切面代理函数；由所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

[05]根据第二方面，提供了一种原型链污染漏洞防护装置，包括：请求检测模块，用于检测请求接口是否接收到请求；函数启动模块，用于若所述请求接口接收到请求，则启动在所述请求接口预先注入的切面代理函数；第一检测模块，用于由所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

[06]根据第三方面，提供了一种计算机可读存储介质，其上存储有计算机程序，当所述计算机程序在计算机中执行时，令计算机执行第一方面提供的方法。

[07]根据第四方面，提供了一种计算设备，包括存储器和处理器，所述存储器中存储有可执行代码，所述处理器执行所述可执行代码时，实现第一方面提供的所述的方法。

[08]本说明书实施例提供的原型链污染漏洞防护方法及装置、介质、设备，各自或者组合后具有如下有益效果：（1）预先在请求接口注入了切面代理函数，当请求接口接收到请求时，切面代理函数会启动检测流程，进而检测请求中是否包含原型属性，如果检

测到原型属性，则拦截该请求。由于原型属性是造成原型链污染的关键属性，如果检测到原型属性就拦截请求，从而避免原型链污染造成后续的各种漏洞攻击的发生。本申请实施例提供一种防护机制，应用无需手动升级 NPM 包，也不依赖于 NPM 包的发布部署流程，因此无需等待官方安全版本的发布，具备业务无打扰的紧急漏洞止血的效果。

[09] (2) 在一个实施例中，在对象递归合并类函数中预先注入切面代理函数，这样当用户输入对象递归合并类函数的所需参数时，就会启动对象递归合并类函数中注入的切面代理函数，进而启动检测逻辑，以检测对象递归合并类函数的输入参数中是否包含原型属性，若包含原型属性，则拦截所述对象递归合并类函数的对象拷贝操作。如果不包含原型属性，则可以使对象递归合并类函数完成正常的对象拷贝操作，这样可以避免使对象递归合并类函数成为造成原型链污染的危险函数。

[10] (3) 在一个实施例中，在按路径定义属性类函数中预先注入切面代理函数，这样当用户输入指定路径时，就会启动按路径定义属性类函数中注入的切面代理函数，进而启动检测逻辑，以检测按路径定义属性类函数的指定路径中是否包含原型属性，若包含原型属性，则拦截按路径定义属性类函数的路径赋值操作。如果不包含原型属性，则可以使按路径定义属性类函数完成正常的路径赋值操作，这样可以避免使按路径定义属性类函数成为造成原型链污染的危险函数。

附图说明

[11] 为了更清楚地说明本说明书实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图是本说明书的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[12] 图 1 是本说明书一个实施例中原型链污染漏洞防护方法的流程示意图。

[13] 图 2 是本说明书一个实施例中原型链污染漏洞防护装置的结构框图。

具体实施方式

[14] 下面结合附图，对本说明书提供的方案进行描述。

[15] 正如背景技术中所说的，原型链污染会造成拒绝服务攻击、远程代码/命令执行漏洞攻击、属性注入漏洞攻击等，威胁 Node 应用安全。为了保证 Node 应用安全，可以通

过代码扫描等方式，检测是否 Node 应用中引入了含有原型链污染漏洞的不安全版本的 NPM 包，然后推动 Node 应用升级 NPM 包成安全版本。但是该方式需要 Node 应用升级代码、重新发布和部署 NPM 包，而且依赖于 NPM 包的发布，整个修复治理周期可能比较长，风险暴露的周期就比较长，不能及时对抗新出现的漏洞攻击。其中，NPM 的全称是 Node Package Manager，是一个 NodeJS 包管理和分发工具，已经成为了非官方的发布 Node 模块的标准。Node 应用是一种服务端脚本语言的应用。

[16]为此，根据第一方面，本说明书实施例提供了一种原型链污染漏洞防护方法，包括：在请求接口接收到请求时，启动在所述请求接口预先注入的切面代理函数，以使所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

[17]下面描述以上构思的具体实现方式。

[18]图 1 是本申请一个实施例中原型链污染漏洞防护方法的流程示意图。可以理解，该方法可以通过任何具有计算、处理能力的装置、设备、平台、设备集群来执行。参见图 1，原型链污染漏洞防护方法包括步骤 S102 至步骤 S106。

[19]S102、检测请求接口是否接收到请求。

[20]S104、若是，则启动在所述请求接口预先注入的切面代理函数。

[21]S106、由所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

[22]在图 1 示出的原型链污染漏洞防护方法中，预先在请求接口注入了切面代理函数，当请求接口接收到请求时，切面代理函数会启动检测流程，进而检测请求中是否包含原型属性，如果检测到原型属性，则拦截该请求。由于原型属性是造成原型链污染的关键属性，如果检测到原型属性就拦截请求，从而避免原型链污染造成后续的各种漏洞攻击的发生。本申请实施例提供一种防护机制，应用无需手动升级 NPM 包，也不依赖于 NPM 包的发布部署流程，因此无需等待官方安全版本的发布，具备业务无打扰的紧急漏洞止血的效果。

[23]下面描述图 1 中各个步骤的执行方式。

[24]S102、检测请求接口是否接收到请求；即，检测请求接口是否接收到请求，在接收到请求时执行 S104。

[25]S104、若所述请求接口接收到请求，则启动在所述请求接口预先注入的切面代理函

数；其中，切面代理函数可以通过安全切面的底座模块预先插桩到请求接口中。切面代理函数的作用是检测原型属性以及在包含原型属性时进行拦截操作。

[26]其中，安全切面是指安全平行切面体系，是下一代原生安全基础设施，通过端—管—云各层次切面使安全管控与业务相互融合且解耦，并依托标准化接口为业务提供精准内视与高效干预能力，具备感知覆盖能力强、应急攻防响应快、安全治理高效和安全布防灵活的核心优势。安全切面将面向切面编程思想推广应用到安全体系建设中，构建与业务正交融合的安全平行空间，在不修改业务正常逻辑的情况下，实现更高纬度的安全防护的体系思想。

[27]其中，安全切面的底座模块的主要功能是程序插桩，程序插桩是在保证被测程序原有逻辑完整性的基础上在程序中插入一些探针，本质上就是进行信息检测的代码段，通过探针的执行而得知是否包含原型属性，并在包括原型属性时进行拦截操作。

[28]S106、由所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

[29]即，在请求接口接收到请求时，启动在所述请求接口预先注入的切面代理函数，以使所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

[30]其中，可以预先在 http 请求接口注入切面代理函数，这样当 http 请求接口接收到 http 请求时就会启动切面代理函数，进而使得切面代理函数对 http 请求执行检测逻辑。

[31]其中，还可以预先在 rpc 请求接口注入切面代理函数，这样当 rpc 请求接口接收到 rpc 请求时就会启动切面代理函数，进而使得切面代理函数对 rpc 请求执行检测逻辑。

[32]可理解的是，http 的全称是 Hyper Text Transfer Protocol，是超文本传输协议。rpc 的全称是 Remote Procedure Call，即远程过程调用。

[33]可见，在一个实施例中，所述请求接口可以为 http 请求接口和/或 RPC 请求接口。

[34]举例来说，JavaScript 只有一种结构即对象，每个对象都有一个私有属性即原型属性，原型属性指向它的构造函数的原型对象，该原型对象也有一个自己的原型对象，这样层层向上直到一个原型对象为空即 null 为止。当访问一个对象的属性时，除了搜索该对象之外，还会沿着对象的原型链层层向上搜索，直到匹配到属性或到达原型链顶端 null。如果修改了一个原型对象，那么会影响到所有来自于这个原型的对象，即产生了原型链污染。

[35]具体的，当污染的属性来自于对象时，原型链污染造成的破坏性更大，例如覆盖对象默认的 toString 方法和 valueOf 方法，可能导致整个程序停止运行，造成 DoS 攻击。如果代码中执行了某个对象被污染的属性，攻击者可执行相关命令，造成 RCE 漏洞攻击。如果原型链污染造成了涉及权限、身份相关的属性注入，例如，污染了 cookie 等，可能导致所有用户公用一个会话。可见会造成各种漏洞攻击。

[36]其中，ToString 方法用于返回一个“以文本方式表示”对象的字符串，结果是一个简明但易于读懂的信息表达式。valueOf 方法用于把对象转换为所需要的类型。

[37]其中，DoS，即 Denial of Service，是指拒绝服务攻击，一种网络攻击方式，目的使计算机或网络无法提供正常的服务，导致正常用户无法访问。

[38]其中，RCE，即 Remote Command/Code Execute，即远程代码/命令执行漏洞攻击，攻击者可以直接在 web 页面向后台服务器远程注入操作系统命令或者代码，从而控制后台系统。

[39]其中，属性注入，例如在 JavaScript 中污染了某个类的原型类型，那么那些没有被显式定义的对象都会受到影响，尤其涉及权限、身份相关的属性污染。

[40]为了避免原型链污染造成的拒绝服务攻击、远程代码/命令执行漏洞攻击、属性注入等漏洞攻击，通过切面技术，将请求接口作为切面代理函数的注入点，注入切面代理函数。当有请求到来时，执行检测流程，以检测请求的主体内容中是否包含原型属性，即“__proto__”属性，如果包含该属性，则拦截该请求。如果不包含该请求，不会拦截该请求，向请求的发送方返回相应的参数。

[41]可见，本申请实施例基于切面技术，在无需升级应用代码业务的前提下，且在无干扰的情况下，实现了一种通用的原型链污染漏洞防护机制，可以有效防护原型链污染漏洞攻击。

[42]其中，切面技术是一种通过预编译方式和运行期动态代理的方式实现在不修改源代码的情况下给程序动态统一添加功能的技术。切面代理函数是需要基于切面技术动态添加增强功能的方法或函数。

[43]在一个实施例中，所述切面代理函数在所述请求接口的注入点可以为所述请求接口接收到请求之后且执行所述请求的逻辑之前的代码位置。

[44]也就是说，切面代理函数的注入点为请求接口接收到请求之后的代码位置，且该代码位置位于执行请求本身的逻辑之前。当请求接口接收到请求后，不执行请求本身对应

的逻辑，而是直接启动切面代理函数进行检测，如果检测不到原型属性，则开始执行请求本身对应的逻辑，进而向请求的发送方返回相应的参数。

[45]在一个实施例中，所述切面代理函数在所述请求接口的注入点可以为执行所述请求的逻辑之后且所述请求接口向所述请求的发送方返回参数之前对应的代码位置。

[46]也就是说，当请求接口接收到请求后，启动切面代理函数执行检测逻辑，且执行请求本身对应的逻辑，得到相关的输出参数。如果切面代理函数检测到原型属性，则请求接口不会向请求的发送方返回相应的输出参数，如果切面代理函数检测不到原型属性，则请求接口向请求的发送方返回相应的输出参数。

[47]可见，所述切面代理函数的注入点可以有多种选择，可以根据具体场景或需要进行选择。

[48]可理解的是，通过切面技术来代理容易造成原型链污染的关键危险函数，检测关键危险函数的参数是否包含造成污染原型链的原型属性，而在实际开发中存在原型链污染漏洞的依赖方法众多，且许多原型链污染漏洞是未可知的，无法将这些方法一一代理进行防护。但实际攻击场景中，攻击者通常会利用 Node 应用的请求接口传递原型属性而造成原型链污染，因此本申请实施例提供的方法将 http/rpc 请求方法作为切面代理检测的关键方法，利用切面技术代理原有方法，增加原型链污染的检测逻辑。

[49]在一个实施例中，本申请实施例还可以包括：在对象递归合并类函数接收到输入参数时，启动在所述对象递归合并类函数中预先注入的切面代理函数，以使所述切面代理函数检测所述输入参数中是否包含原型属性，若包含原型属性，则拦截所述对象递归合并类函数的对象拷贝操作。

[50]也就是说，本申请实施例在请求接口预先注入切面代理函数的基础上，还在对象递归合并类函数中预先注入切面代理函数，进一步避免不安全的对象递归合并引入的原型链污染漏洞。

[51]其中，对象递归合并类函数是一类函数，例如，lodash.merge。

[52]举例来说，在 JavaScript 中进行对象拷贝时，原生的对象拷贝方法——Object.assign 只能实现深拷贝第一层对象属性，第一层之后的对象属性还是浅拷贝，因此开发过程中如果需要深拷贝对象时，可能会引入一个对象递归合并函数 dangerDeepClone。当用户为对象递归合并函数 dangerDeepClone 输入源对象时，如果用户输入的源对象中包括一个具有原型属性的危险对象，那么经过对象递归合并函数 dangerDeepClone 进行对象拷

贝后，就会把对象 car1 的原型属性修改成了{“isAdmin”：true}。因此如果再通过对象构造函数构造一个新的对象 car2 后，就可以发现对象 car2 的原型属性的 admin 权限是通过“isAdmin”控制的，那么此时所有 Car 实例对象都拥有了 admin 权限，这会造成越权攻击。

[53]由于 JavaScript 一切皆对象，因此如果污染的原型是具有原型类型的对象即 Object.prototype 时，原型链污染的范围和破坏性更大，可能导致整个程停止运行，造成上述提到的 DoS 攻击。

[54]为此，本申请实施例还可以在对象递归合并类函数中预先注入切面代理函数，这样当用户输入对象递归合并类函数的所需参数时，就会启动对象递归合并类函数中注入的切面代理函数，进而启动检测逻辑，以检测对象递归合并类函数的输入参数中是否包含原型属性，若包含原型属性，则拦截所述对象递归合并类函数的对象拷贝操作。如果不包含原型属性，则可以使对象递归合并类函数完成正常的对象拷贝操作，这样可以避免使对象递归合并类函数成为造成原型链污染的危险函数。

[55]在一个实施例中，本申请实施例提供的方法还可以包括：在按路径定义属性类函数接收到指定路径时，启动在所述按路径定义属性类函数中预先注入的切面代理函数，以使所述切面代理函数检测所述指定路径中是否包含原型属性，若包含原型属性，则拦截所述按路径定义属性类函数的路径赋值操作。

[56]其中，按路径定义属性类函数是指按照路径定义属性的一类函数。

[57]举例来说，许多 JavaScript 三方库提供按照指定的路径修改或定义对象属性的方法，形式可能如下：Function defineObjectPropByPath(object,path,value){}。

[58]函数 defineObjectPropByPath 用于将对象 object 的指定路径 path 的属性值修改为 value，如果用户输入的制定路径为“__proto__.isAdmin”，那么使用 defineObjectPropByPath 方法后可能将“__proto__.isAdmin”注入到原型类型对象——Object.prototype 中造成大范围的影响。

[59]为此，本申请实施例还可以在按路径定义属性类函数中预先注入切面代理函数，这样当用户输入指定路径时，就会启动按路径定义属性类函数中注入的切面代理函数，进而启动检测逻辑，以检测按路径定义属性类函数的指定路径中是否包含原型属性，若包含原型属性，则拦截按路径定义属性类函数的路径赋值操作。如果不包含原型属性，则可以使按路径定义属性类函数完成正常的路径赋值操作，这样可以避免使按路径定义属

性类函数成为造成原型链污染的危险函数。

[60]可见，本申请实施例在请求接口中注入切面代理函数的基础上，针对不安全的对象递归合并引入的原型链污染漏洞、按路径定义属性引入的原型链污染漏洞两种具体场景，在对应的函数中也注入了切面代理函数，进一步进行防护。

[61]上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者可能是有利的。

[62]根据第二方面，本申请实施例提供一种原型链污染漏洞防护装置。

[63]参见图 2，装置 200 包括：请求检测模块 202，用于检测请求接口是否接收到请求；函数启动模块 204，用于若所述请求接口接收到请求，则启动在所述请求接口预先注入的切面代理函数；第一检测模块 206，用于由所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

[64]在一个实施例中，装置还包括：第二检测模块，用于在对象递归合并类函数接收到输入参数时，启动在所述对象递归合并类函数中预先注入的切面代理函数，以使所述切面代理函数检测所述输入参数中是否包含原型属性，若包含原型属性，则拦截所述对象递归合并类函数的对象拷贝操作。

[65]在一个实施例中，装置还包括：第三检测模块，用于在按路径定义属性类函数接收到指定路径时，启动在所述按路径定义属性类函数中预先注入的切面代理函数，以使所述切面代理函数检测所述指定路径中是否包含原型属性，若包含原型属性，则拦截所述按路径定义属性类函数的路径赋值操作。

[66]在一个实施例中，所述切面代理函数在所述请求接口的注入点为所述请求接口接收到请求之后且执行所述请求的逻辑之前的代码位置。

[67]在一个实施例中，所述切面代理函数在所述请求接口的注入点为执行所述请求的逻辑之后且所述请求接口向所述请求的发送方返回参数之前对应的代码位置。

[68]在一个实施例中，所述请求接口为 http 请求接口和/或 RPC 请求接口。

[69]可理解的是，本申请实施例提供的装置中有关内容的解释、具体实施方式、有益效果、举例等内容可以参见第一方面提供的方法中的相应部分，此处不再赘述。

[70]根据第三方面，本说明书一个实施例提供了一种计算机可读存储介质，其上存储有计算机程序，当所述计算机程序在计算机中执行时，令计算机执行说明书中任一个实施

例中的方法。

[71]具体地，可以提供配有存储介质的系统或者装置，在该存储介质上存储着实现上述实施例中任一实施例的功能的软件程序代码，且使该系统或者装置的计算机（或 CPU 或 MPU）读出并执行存储在存储介质中的程序代码。

[72]在这种情况下，从存储介质读取的程序代码本身可实现上述实施例中任何一项实施例的功能，因此程序代码和存储程序代码的存储介质构成了本申请的一部分。

[73]此外，应该清楚的是，不仅可以通过执行计算机所读出的程序代码，而且可以通过基于程序代码的指令使计算机上操作的操作系统等来完成部分或者全部的实际操作，从而实现上述实施例中任意一项实施例的功能。

[74]此外，可以理解的是，将由存储介质读出的程序代码写到插入计算机内的扩展板中所设置的存储器中或者写到与计算机相连接的扩展模块中设置的存储器中，随后基于程序代码的指令使安装在扩展板或者扩展模块上的 CPU 等来执行部分和全部实际操作，从而实现上述实施例中任一实施例的功能。

[75]可理解的是，本申请实施例提供的计算机可读介质中有关内容的解释、具体实施方式、有益效果、举例等内容可以参见第一方面提供的方法中的相应部分，此处不再赘述。

[76]根据第四方面，本说明书一个实施例提供了一种计算设备，包括存储器和处理器，所述存储器中存储有可执行代码，所述处理器执行所述可执行代码时，实现执行说明书中任一实施例中的方法。

[77]可理解的是，本申请实施例提供的计算设备中有关内容的解释、具体实施方式、有益效果、举例等内容可以参见第一方面提供的方法中的相应部分，此处不再赘述。

[78]可以理解的是，本说明书实施例示意的结构并不构成对本说明书实施例的装置的具体限定。在说明书的另一些实施例中，上述装置可以包括比图示更多或者更少的部件，或者组合某些部件，或者拆分某些部件，或者不同的部件布置。图示的部件可以以硬件、软件或者软件和硬件的组合来实现。

[79]上述装置、系统内的各模块之间的信息交互、执行过程等内容，由于与本说明书方法实施例基于同一构思，具体内容可参见本说明书方法实施例中的叙述，此处不再赘述。

[80]本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于装置实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[81]本领域技术人员应该可以意识到，在上述一个或多个示例中，本申请所描述的功能

可以用硬件、软件、挂件或它们的任意组合来实现。当使用软件实现时，可以将这些功能存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令或代码进行传输。

[82]以上所述的具体实施方式，对本申请的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本申请的具体实施方式而已，并不用于限定本申请的保护范围，凡在本申请的技术方案的基础之上，所做的任何修改、等同替换、改进等，均应包括在本申请的保护范围之内。

权利要求书

1、一种原型链污染漏洞防护方法，包括：

检测请求接口是否接收到请求；

若是，则启动在所述请求接口预先注入的切面代理函数；

由所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

2、根据权利要求1所述的方法，其中，还包括：

在对象递归合并类函数接收到输入参数时，启动在所述对象递归合并类函数中预先注入的切面代理函数，以使所述切面代理函数检测所述输入参数中是否包含原型属性，若包含原型属性，则拦截所述对象递归合并类函数的对象拷贝操作。

3、根据权利要求1所述的方法，其中，还包括：

按路径定义属性类函数接收到用户输入的指定路径时，启动在所述按路径定义属性类函数中预先注入的切面代理函数，以使所述切面代理函数检测所述指定路径中是否包含原型属性，若包含原型属性，则拦截所述按路径定义属性类函数的路径赋值操作。

4、根据权利要求1所述的方法，其中，所述切面代理函数在所述请求接口的注入点为所述请求接口接收到请求之后且执行所述请求的逻辑之前的代码位置。

5、根据权利要求1所述的方法，其中，所述切面代理函数在所述请求接口的注入点为执行所述请求的逻辑之后且所述请求接口向所述请求的发送方返回参数之前对应的代码位置。

6、根据权利要求1所述的方法，其中，所述请求接口为http请求接口和/或RPC请求接口。

7、一种原型链污染漏洞防护装置，包括：

请求检测模块，用于检测请求接口是否接收到请求；

函数启动模块，用于若所述请求接口接收到请求，则启动在所述请求接口预先注入的切面代理函数；

第一检测模块，用于由所述切面代理函数检测所述请求中是否包含原型属性，若包含原型属性，则拦截所述请求。

8、根据权利要求7所述的装置，其中，还包括：

第二检测模块，用于在对象递归合并类函数接收到输入参数时，启动在所述对象递归合并类函数中预先注入的切面代理函数，以使所述切面代理函数检测所述输入参数中是否包含原型属性，若包含原型属性，则拦截所述对象递归合并类函数的对象拷贝操作。

9、一种计算机可读存储介质，其上存储有计算机程序，当所述计算机程序在计算机中执行时，令计算机执行权利要求 1~6 任一项所述的方法。

10、一种计算设备，包括存储器和处理器，所述存储器中存储有可执行代码，所述处理器执行所述可执行代码时，实现权利要求 1~6 任一项所述的方法。