



链滴

Hexo-theme-Sakura 折腾日记 (1)

作者: [lixiang810](#)

原文链接: <https://ld246.com/article/1592060630720>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

为什么要魔改 Sakura

实名与匿名是一种抉择。让利维坦手中握住更多的信息也许能带给大多数人安全，却同样会伤害异见。而事实证明，异见者并不总是出于恶意动机。何况“对事不对人”“针对观点而非质疑动机”也是共说理的基本素养。然而，有一些国家的政府却不允许任何匿名性的存在，无论是纸质出版的匿名性是网络的匿名性。

近年来，在这些国家中的一个国家，新兴的垄断巨头与政府相勾结，垄断巨头负责挤垮小平台进而集用户（这个过程是自由资本主义转为垄断资本主义，也即帝国主义的必然），而政府则可以方便地对那些已经自愿或**因大多数人的自愿而被迫集中起来的人**进行集中管理（这却是偶然——法德等大多数欧国家是反例）。该国网络的匿名程度很低，然而，网络犯罪却不见得因此减少：巨头和政府有办法到公民的个人信息，另一些个人就有办法从巨头和手里把它们购买或盗窃出来。恶俗维基的存在以及俗维基上面满天飞舞的户籍就是证明——小如未成年人大如刘慈欣，只要把自己的信息交予利维坦，不要指望别人不知道，特别是在这单凭手机号就足够把人翻个底朝天的时代。刘慈欣的小号被发现，是从其绑定的手机号着手，发掘出了支付宝号、身份证号等信息，从而让恶俗人士一步步锁定目标。

这一点上，Sakura 采用的 Valine 评论系统令我胆战心惊：部署 Valine 的后端，竟需要进行身份证实名认证！作为亲眼目睹公权力对本人学校里异见者迫害，甚至自己也险遭卷入的人，毫无疑问地，我害怕。万一哪天我心血来潮作了死（如“不存在的人”、“不存在的人”和“不存在的人”），或者意之间说了一些不能说的东西（李医生等人就是例子），那岂不是训诫往上起步？虽然公权力找到我身份易如反掌，但我不想主动投怀送抱。万一触怒了恶俗系的残留势力，那岂不是全家户籍都要被挂去？虽然有心者还是可以从这个博客里找到我真人的蛛丝马迹，但我不希望他们只花5块钱查个社工就做得得到。

为什么使用 gitalk

毫无疑问地，拒绝追踪的评论系统在这个国家已经死绝了。如果没有，那么也不是我能找到的。于是首先考虑了 [disqus](#)。然后我发现它的服务器被墙了。囧rz。

最后我选择了 gitalk，它是一个利用 github issues 作为后端的评论系统。[Hexo-theme-Nexmoe](#) 自带，但我更喜欢 [Sakura](#)，然而 [Sakura](#) 目前仅支持 [gitalk](#)。

使用 gitalk，是因为 github 短期内不会被墙。作为全世界最大开源平台，并且有微软这样与 [这国家](#) 作的公司的加持，github 被墙的可能性不能说没有，但短期内我们仍可拥有“依附的自由”。

于是我开始了魔改。

尝试与失败过程

先咕着。（屑颜）

最终进行的修改

先装 gitalk（不确定这一步是否必须）

```
$ npm install gitalk --save
```

然后修改 themes/Sakura/layout/_partial/comments.ejs，以下是修改后的文本：

```
<!--这一堆是原内容，我全部注释了  
<% if (theme.valine && post.comments) { %>  
<div id="vcomments"></div>
```

```

<script>
window.onload = function(){
  var valine = new Valine();
  valine.init({
    el: '#vcomments',
    appId: "<%= theme.v_appId %>",
    appKey: "<%= theme.v_appKey %>",
    path: window.location.pathname,
    placeholder: "你是我一辈子只会遇见一次的惊喜 ..."
  })
}
</script>
<% } %>
-->

```

```

<!--以下是正文，懒得改别处，就把原来的if照搬了过来-->
<% if (theme.valine && post.comments) { %>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/gitalk@1/dist/gitalk.css">
<div id="gitalk"></div>
<script src="https://cdn.jsdelivr.net/npm/gitalk@1/dist/gitalk.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/blueimp-md5@2.12.0/js/md5.min.js"></script>
<script type="text/javascript">
  var gitalk = new Gitalk({
    clientID: '<%= theme.gitalk.clientID %>',
    clientSecret: '<%= theme.gitalk.clientSecret %>',
    id: md5(location.pathname),
    repo: '<%= theme.gitalk.repo %>',
    owner: '<%= theme.gitalk.owner %>',
    admin: '<%= theme.gitalk.admin %>'
  })
  gitalk.render('gitalk')
</script>
<% } %>

```

最后修改themes/Sakura/_config.yml:

```

# Valine
valine: true
#v_appId: #####
#v_appKey: #####

gitalk:
  admin: 我的名字 # 拥有对该repo进行操作的 GitHub username
  owner: 我的名字 # 持有该 repo 的 GitHub username
  repo: 我的 repo # 存放评论的 issue 所在的 repo
  clientID: 我的 Client ID # GitHub Client ID
  clientSecret: 我的 Secret # GitHub Client Secret

```

保存。

```

$ hexo clean
#以下两条命令默认你是把你的hexo托管在github上:
$ git commit -a
$ git push

```

然后你就拥有了一个 gitalk 匿名评论区。虽说也得注册账号，但只需要程序员们人人都有的 github 号就可以了。

这些方法对于防止公权力的滥用来说是远远不够的，因为你总得在公权力的眼皮子底下传播你的链接。如果你搞的事儿足够大，他们不需要在墙外解决问题，只需要在墙内溯源第一个发布链接的人就行。

但是，这个办法可以稍微提高一点追踪成本，也能让被公权力以及 **被天天作死的你整得神经过敏的境**私人服务提供商稍微松口气，还能让你的评论区不由境内私人服务提供商进行过分严格的审核，何乐不为呢？

黎想

2020年6月13日于家中