



# 第13章 安全配置

Jupyter金融应用从入门到实践

## 1. 设置访问域

可以通过修改配置文件 “jupyter\_notebook\_config.py” 中的 “c.NotebookApp.allow\_origin” 参数来实现访问域的设置。 “c.NotebookApp.allow\_origin = ‘ ’” 即表示只允许本地访问。指定 “c.NotebookApp.allow\_origin = 'microsoft.com' ”，这样设置就只允许 “microsoft.com” 域内的用户访问

```
## Set the Access-Control-Allow-Origin header
#
# Use '*' to allow any origin to access your server.
#
# Takes precedence over allow_origin_pat.
# Default: ''
c.NotebookApp.allow_origin = ''
```

```
## Set the Access-Control-Allow-Origin header
#
# Use '*' to allow any origin to access your server.
#
# Takes precedence over allow_origin_pat.
# Default: ''
c.NotebookApp.allow_origin = 'microsoft.com'
```



## 2. 设置访问IP

Jupyter Notebook同样支持配置IP来控制访问权限。通过修改配置文件中的“c.NotebookApp.ip”参数即可实现。如图 13-4 所示，默认情况下，该参数的配置为“c.NotebookApp.ip = 'localhost'”，如图13-5所示，如果设置为“c.NotebookApp.ip = '0.0.0.0'”，则意味着所有IP地址均可以访问。

```
## The IP address the notebook server will listen on.  
# Default: 'localhost'  
c.NotebookApp.ip = 'localhost'
```

图 13-4 Jupyter Notebook 默认 IP 参数 ( 1 )

```
## The IP address the notebook server will listen on.  
# Default: 'localhost'  
c.NotebookApp.ip = '0.0.0.0'
```

图 13-5 Jupyter Notebook 配置 IP 参数 ( 2 )

### 3. 端口配置

Jupyter Notebook同样支持通过端口配置来改变默认的运行端口，从而实现访问控制。由于许多黑客场景通过扫描默认开放端口来快速获取目标，因此建议最好修改默认端口。通过设置 “c.NotebookApp.port”参数可以改变Jupyter Notebook使用的端口，如图13-6所示。

```
## The port the notebook server will listen on (env: JUPYTER_PORT).  
# Default: 8888  
c.NotebookApp.port = 8888
```

图 13-6 Jupyter Notebook 配置端口参数

#### 4. URL前缀配置

Jupyter Notebook提供了 “c.NotebookApp.base\_url”参数来指定访问Jupyter Notebook的URL前缀，默认的URL前缀为 “/”。修改该参数的值，改变Jupyter Notebook运行时的URL前缀，如图13-9所示。

```
## The base URL for the notebook server.  
#  
# Leading and trailing slashes can be omitted, and will automatically be added.  
# Default: '/'  
c.NotebookApp.base_url = '/ipython/'
```

图 13-9 Jupyter Notebook 修改默认 URL 前缀

## 1. Token认证

Jupyter Notebook默认使用令牌（Token）认证。使用基于令牌（Token）的身份验证方法，在服务端不需要存储用户的登录记录。基于令牌（Token）的认证过程如图13-11所示。

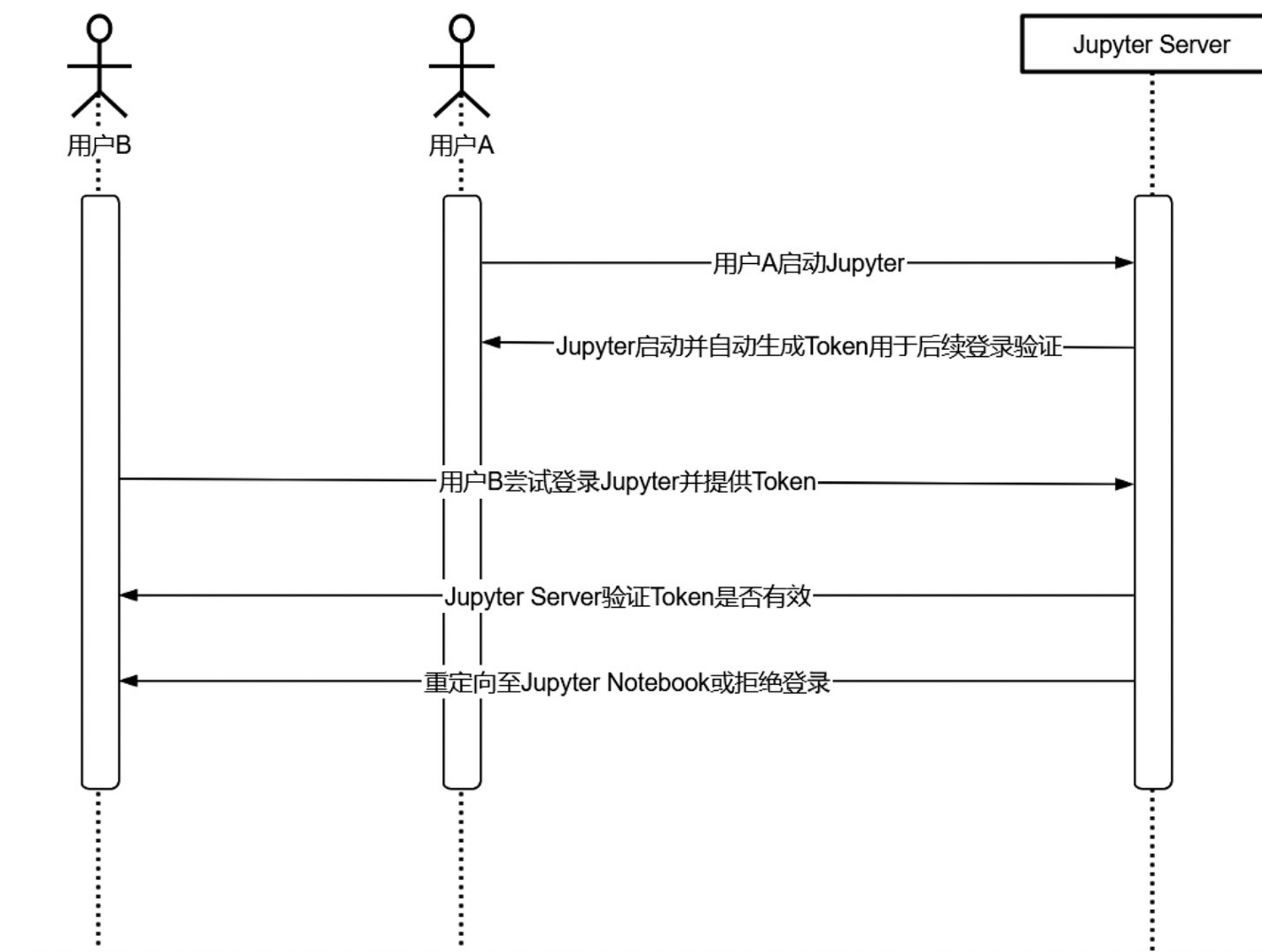


图 13-11 Jupyter Token 认证流程图



## 2. 口令认证

口令（ Password ）身份验证。这是通过设置配置文件中的`c.NotebookApp.password`参数来完成。首先需要输入口令，确认后即可得到一个散列后的密钥，如图13-14所示。然后将生成的口令哈希值填写到上述配置文件的字段中，如图13-15所示。

```
In [1]: from notebook.auth import passwd
```

```
In [2]: passwd()
```

```
Enter password: . . . . .
```

```
Verify password: . . . . .
```

```
Out[2]: 'sha1:834e20e11ef6:44d75198940c262cf0ef68ffd74dadaad1e13c1c'
```

```
In [ ]:
```

```
## Hashed password to use for web authentication.
```

```
#
```

```
# To generate, type in a python/IPython shell:
```

```
#
```

```
# from notebook.auth import passwd; passwd()
```

```
#
```

```
# The string should be of the form type:salt:hashed-password.
```

```
# Default: ''
```

```
c.NotebookApp.password = 'sha1:834e20e11ef6:44d75198940c262cf0ef68ffd74dadaad1e13c1c'
```

### 3. 无认证

如果Jupyter Notebook仅限于局域网内部使用，并且数据没有安全性要求时，那么可以将配置文件中的c.NotebookApp.token字段以及c.NotebookApp.password字段设置为空。

```
## Token used for authenticating first-time connections to the server.
#
# The token can be read from the file referenced by JUPYTER_TOKEN_FILE or set
# directly with the JUPYTER_TOKEN environment variable.
#
# When no password is enabled, the default is to generate a new, random token.
#
# Setting to an empty string disables authentication altogether, which is NOT
# RECOMMENDED.
# Default: '<generated>'
c.NotebookApp.token = ''
c.NotebookApp.password = ''
```





**谢谢！**