# 基于Proxy和ICAP的网络安全应用设计

学号：202200201095

姓名：杨伟康

## 一、实验目的

1. 理解Proxy和ICAP的基本原理及其在网络安全中的应用。
2. 掌握Squid代理服务器的安装与配置。
3. 学习ICAP协议的使用，并通过ClamAV实现病毒查杀功能。
4. 设计并初步实现一个简单的DLP（数据泄露防护）系统。

## 二、实验环境

• 操作系统：Ubuntu 22

• 软件：Squid, ClamAV, c-icap, Python

• 硬件：普通PC机

## 三、实验步骤

### 3.1 什么是Squid

**Squid的基本功能**

Squid是一个功能强大的代理服务器和缓存服务器，主要用于提高网络访问速度和安全性。其主要功能包括：

​​**Web缓存​​：**

Squid能够缓存经常访问的网页内容，减少重复请求，从而加快网页加载速度，降低带宽消耗。

​​**反向代理​​：**

Squid可以作为反向代理服务器，将客户端的请求转发到后端服务器，并将后端服务器的响应返回给客户端。这有助于隐藏后端服务器的真实IP地址，提高系统的安全性和可靠性。

​​**访问控制​​：**

Squid提供了灵活的访问控制机制，可以根据IP地址、用户身份、时间等多种条件对网络访问进行限制和管理。

​​**内容过滤​​：**

Squid支持通过ICAP协议集成第三方内容过滤服务，如病毒扫描、敏感信息过滤等，从而实现对网络内容的深度控制。

​​**SSL/TLS拦截​​：**

通过ssl\_bump功能，Squid可以在企业环境中对SSL加密流量进行解密和重新加密，以便进行内容检查和监控。

​​**日志记录与监控​​：**

Squid详细记录所有经过它的请求和响应，便于管理员进行网络流量分析和安全审计。

**Squid的应用场景**

** 企业网络管理​：**

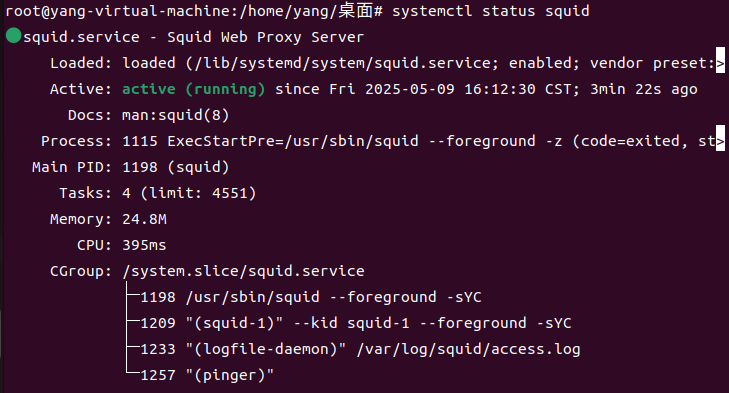
* 在企业环境中，Squid常用于控制员工的网络访问行为，防止员工访问不适当的网站，提高工作效率。
* 通过集成ClamAV等防病毒软件，Squid可以扫描通过代理服务器的文件，防止恶意软件的传播。

 **​内容过滤与DLP（数据泄露防护）​​：**

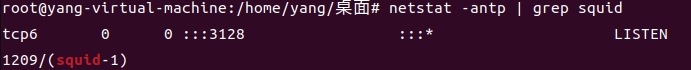
* Squid可以与ICAP服务器结合，实现对企业敏感数据的保护，防止数据泄露。
* 通过关键词过滤和正则表达式匹配，Squid可以识别并阻止包含敏感信息的数据流出企业网络。

### 3.2 安装和试用Squid

1. 使用以下命令安装Squid：
2. sudo apt update
3. sudo apt install squid
4. 检查Squid是否启动并监听在3128端口：
5. systemctl status squid#检测squid的运行状态代码
6. netstat -antp | grep squid#检查squid监听端口的代码



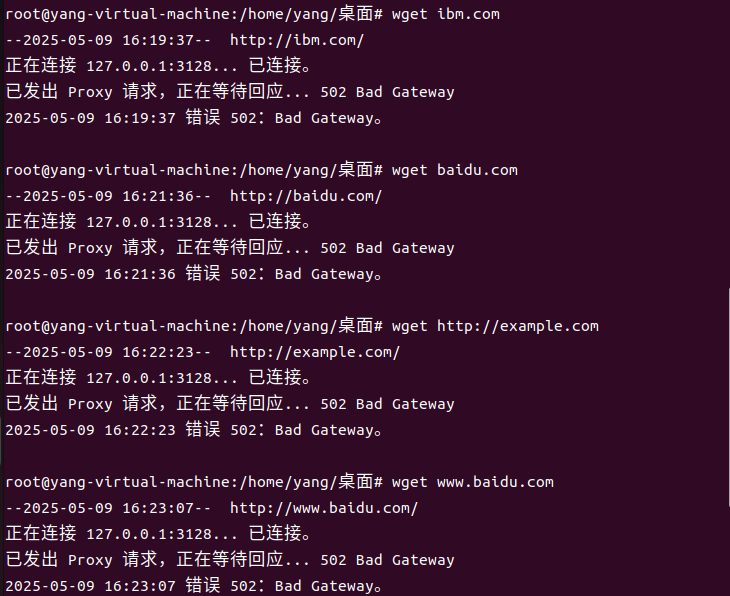
运行状态



正在监听3128端口

1. 配置浏览器使用Squid代理：
2. export http\_proxy=http://127.0.0.1:3128
3. export https\_proxy=http://127.0.0.1:3128
4. 使用wget测试代理是否正常工作：
5. wget www.baidu.com

这里我出现的错误：无论哪个网址都不能连接，错误情形如下：



漫长的纠错之路———

重新检查了软件运行状态，重新查看了正在监视的端口。

tail -f /var/log/squid/access.log /var/log/squid/cache.log#查看日志命令

结果类似：

1746760260.583 0 127.0.0.1 TCP\_MISS/502 3963 GET http://ibm.com/ - HIER\_NONE/- text/html

从日志分析来看，Squid 代理服务器返回 ​**502 Bad Gateway**​ 错误，但日志中 HIER\_NONE/- 表明 Squid ​**没有尝试连接目标服务器**，而是直接返回了错误。这通常意味着 Squid 的配置存在问题，导致它无法正确处理代理请求。

然后我就使用命令修改配置文件：  
sudo nano /etc/squid/squid.conf#打开配置文件命令

问题来了：里面9000多行，我没法看，我通过搜索，对照期望的目标格式，如果有就注释掉，如果缺少就添加的方式进行修改，目的是达到这种配置格式

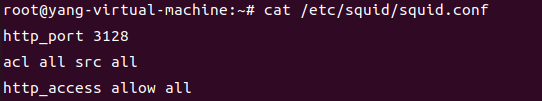




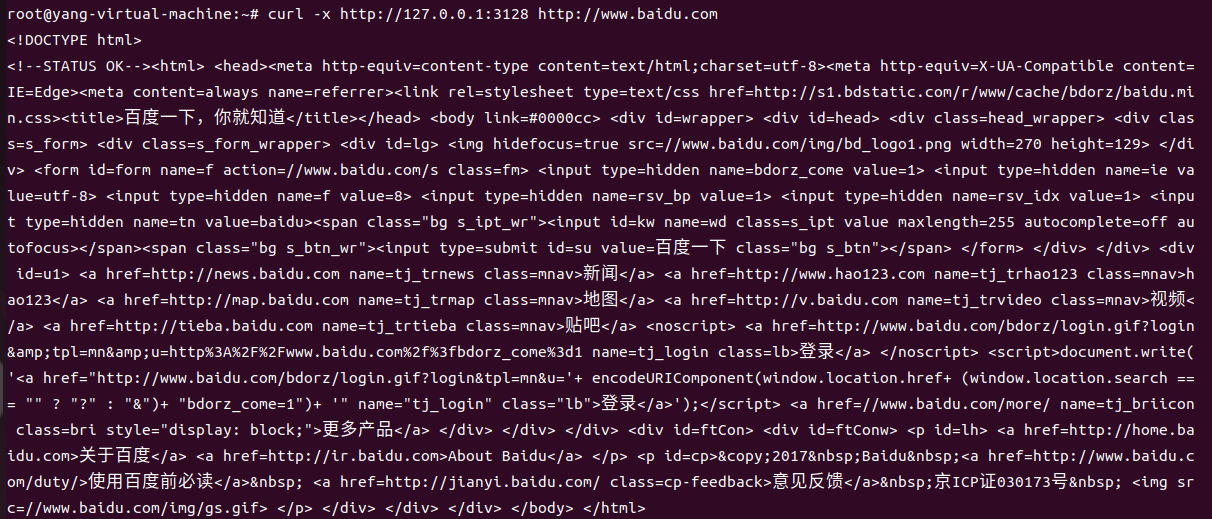
无论怎么修改都不成功。。。

解决方法：

直接删了配置文件，把配置文件9000行原来，改成我现在的只剩上面要求的那3行，虽然配置不完善了，但总算是能用



最终成功连接，不过这是暴力连接（允许所有连接，只能在我这个测试环境进行）





1. ls index\*



1. 查看Squid访问日志：
2. cat /var/log/squid/access.log

可以查看我这个日志，上面一堆失败的连接，都是我在调试，只有最后一个成功了😢



### 3.3 使用代理服务：需要登录

**配置Squid代理服务支持需要登录的网站访问**

要配置Squid代理服务以支持需要登录的网站访问，你需要设置Squid的身份验证功能。以下是详细步骤：

**3.3.1. 安装身份验证工具**

首先，你需要安装一个身份验证助手程序。常用的有basic\_auth和ncsa\_auth：

sudo apt install apache2-utils

**3.3.2. 创建密码文件**

创建一个密码文件并添加用户：

sudo touch /etc/squid/passwords

sudo chown proxy /etc/squid/passwords

sudo htpasswd /etc/squid/passwords username1 # 会提示输入密码

**3.3.3. 配置Squid**

编辑Squid配置文件：

sudo nano /etc/squid/squid.conf

添加或修改以下内容：

|  |
| --- |
| # 启用身份验证  auth\_param basic program /usr/lib/squid/basic\_ncsa\_auth /etc/squid/passwords  auth\_param basic realm proxy  auth\_param basic credentialsttl 2 hours  # 设置允许通过认证的用户访问  acl authenticated proxy\_auth REQUIRED  http\_access allow authenticated  # 注释掉默认的http\_access allow localnet（如果有）  # http\_access allow localnet |

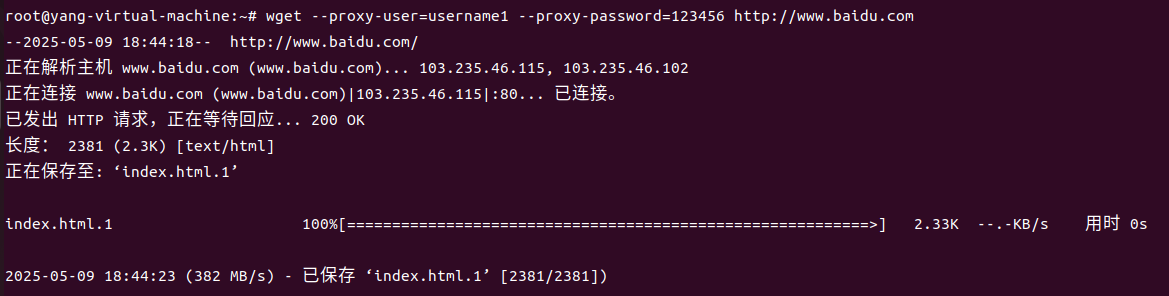
**3.3.4. 重启Squid服务**

sudo systemctl restart squid

**3.3.5. 测试配置**

现在当你尝试通过代理访问网站时，浏览器会弹出认证对话框：

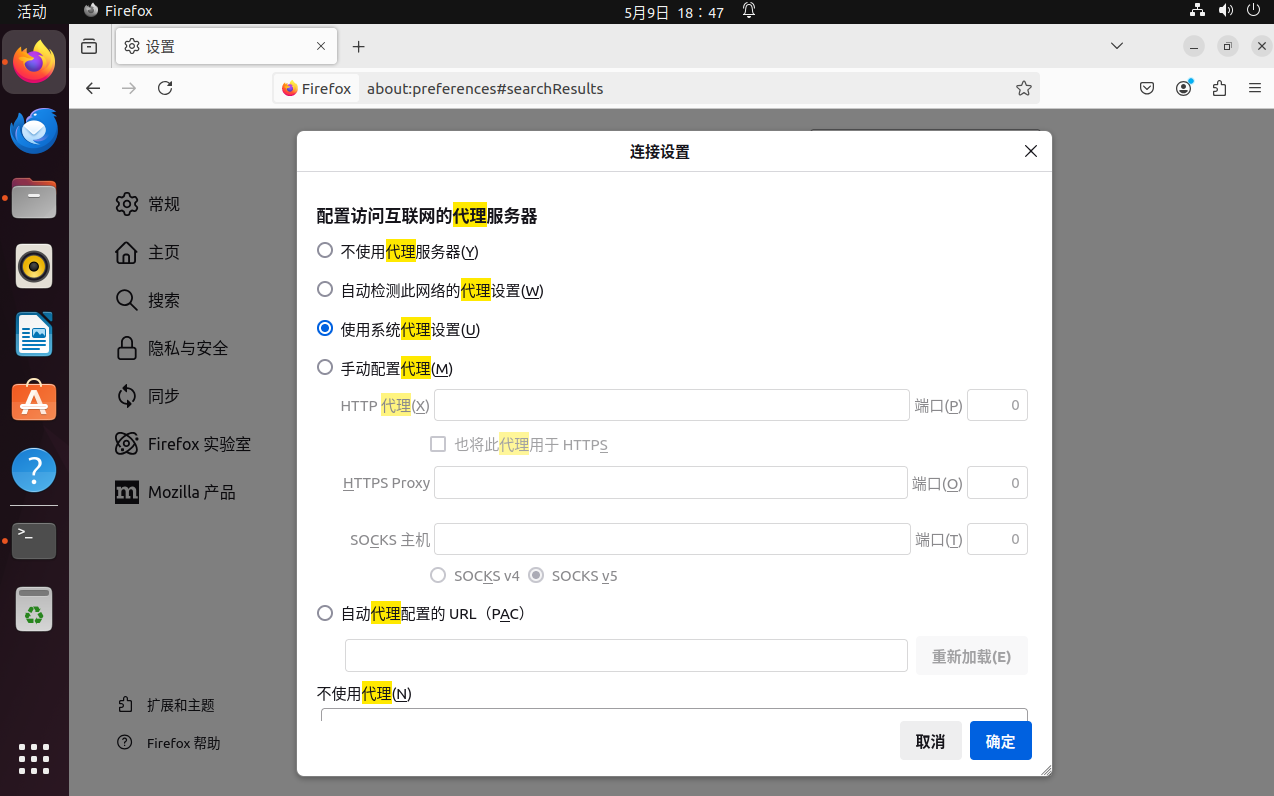
wget --proxy-user=username1 --proxy-password=yourpassword <http://example.com>



或者在浏览器中配置代理时，需要输入用户名和密码。

 ​**打开浏览器设置**​：

* Chrome/Edge: 右上角三个点 → 设置 → 系统 → 打开计算机的代理设置
* Firefox: 右上角菜单 → 设置 → 搜索"代理"



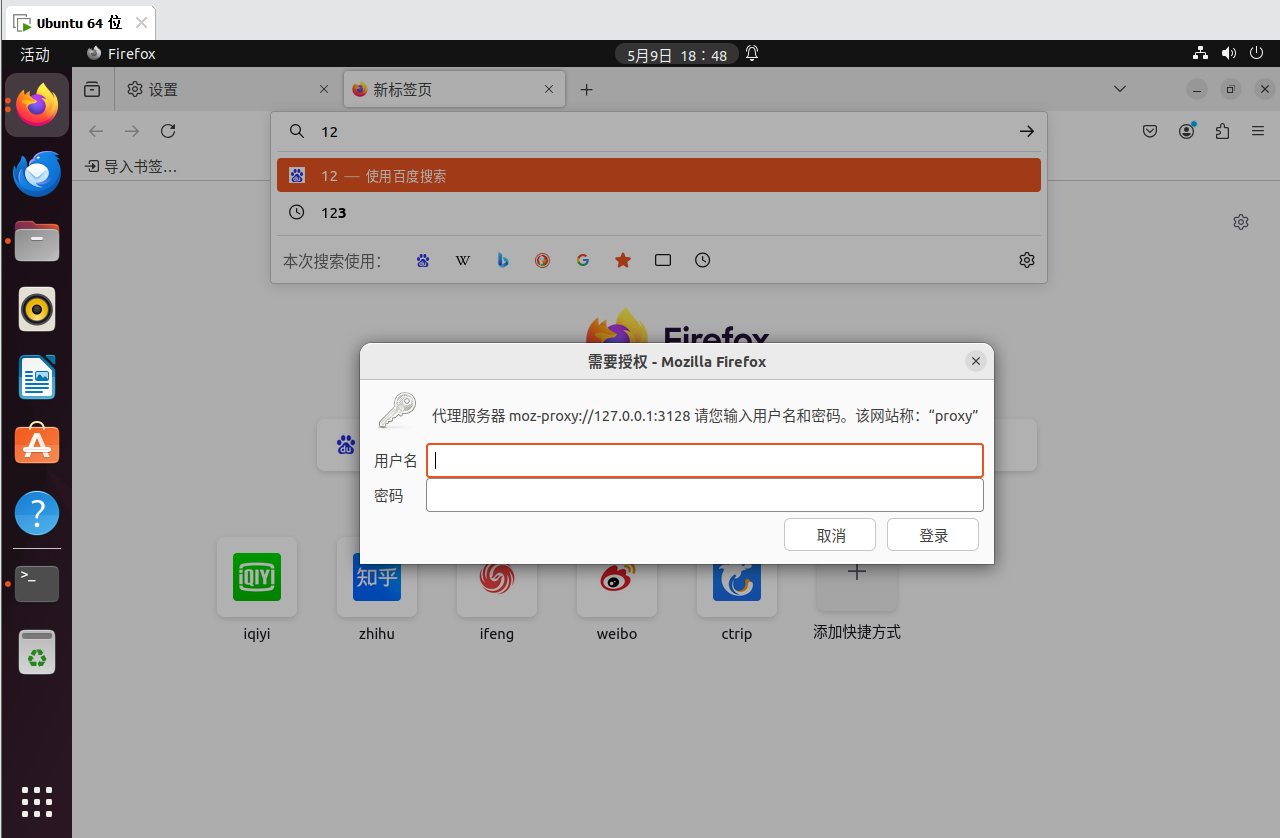
 ​**手动配置代理**​：

* 选择"手动代理配置"
* HTTP代理: 127.0.0.1
* 端口: 3128
* 勾选"为所有协议使用相同代理"
* 

 ​**设置认证信息**​：

* 首次访问网站时，浏览器会弹出认证对话框
* 输入Squid配置的用户名和密码
* 可选勾选"记住密码"

弹出的认证框：



**3.3.6. 验证日志**

检查Squid访问日志以确认认证是否正常工作：

sudo tail -f /var/log/squid/access.log

你应该能看到类似这样的条目，其中包含用户名信息：

timestamp elapsed IP/TCP\_MISS/200 GET http://example.com/ - HIER\_DIRECT/ip user=username1

**注意事项**

1. 确保密码文件的权限正确，Squid进程（通常以proxy用户运行）能够读取它。
2. 如果使用HTTPS网站，可能需要配置SSL拦截（ssl\_bump）才能看到完整的URL。
3. 对于生产环境，考虑使用更安全的认证方式，如LDAP或Kerberos。
4. 如果遇到问题，检查Squid的错误日志：

sudo tail -f /var/log/squid/cache.log

完成这些步骤后，你就成功配置了需要登录的代理服务，可以继续进行实验的下一步了。

### 3.4 监控SSL会话：Squid ssl\_bump

**一、准备工作**

**1. 安装必要组件**

sudo apt install squid squid-common ssl-cert

**2. 生成CA证书（用于签发拦截证书）**

sudo openssl req -new -newkey rsa:2048 -days 365 -nodes -x509 \

-subj "/C=CN/ST=Beijing/L=Beijing/O=YourOrg/CN=Squid Proxy CA" \

-keyout /etc/squid/ssl\_cert/ca-key.pem \

-out /etc/squid/ssl\_cert/ca-cert.pem



sudo chmod 600 /etc/squid/ssl\_cert/\*

sudo chown -R proxy:proxy /etc/squid/ssl\_cert

**二、配置Squid**

**1. 编辑Squid配置文件**

sudo nano /etc/squid/squid.conf

**2. 添加SSL拦截配置**

在配置文件中添加或修改以下内容：

# 启用SSL拦截

http\_port 3128 ssl-bump \

cert=/etc/squid/ssl\_cert/ca-cert.pem \

key=/etc/squid/ssl\_cert/ca-key.pem

# 定义SSL bump规则

acl step1 at\_step SslBump1

ssl\_bump peek step1

ssl\_bump bump all

ssl\_bump splice all

# 允许SSL拦截的ACL规则

acl allowed\_ssl\_sites ssl::server\_name .example.com .google.com

ssl\_bump peek allowed\_ssl\_sites

ssl\_bump bump all

ssl\_bump splice all

# 其他必要配置

sslcrtd\_program /usr/lib/squid/security\_file\_certgen -s /var/lib/ssl\_db -M 4MB

sslcrtd\_children 5

**3. 创建SSL证书数据库**

sudo /usr/lib/squid/security\_file\_certgen -c -s /var/lib/ssl\_db -M 4MB

其中这个命令，新版本squid-openssl需要单独安装才能运行

sudo apt install squid-openssl

sudo chown -R proxy:proxy /var/lib/ssl\_db

**三、重启Squid服务**

sudo systemctl restart squid

**四、客户端配置**

**1. 导出CA证书**

sudo cp /etc/squid/ssl\_cert/ca-cert.pem ~/squid-ca-cert.pem

**2. 在客户端安装CA证书**

​**Windows**:

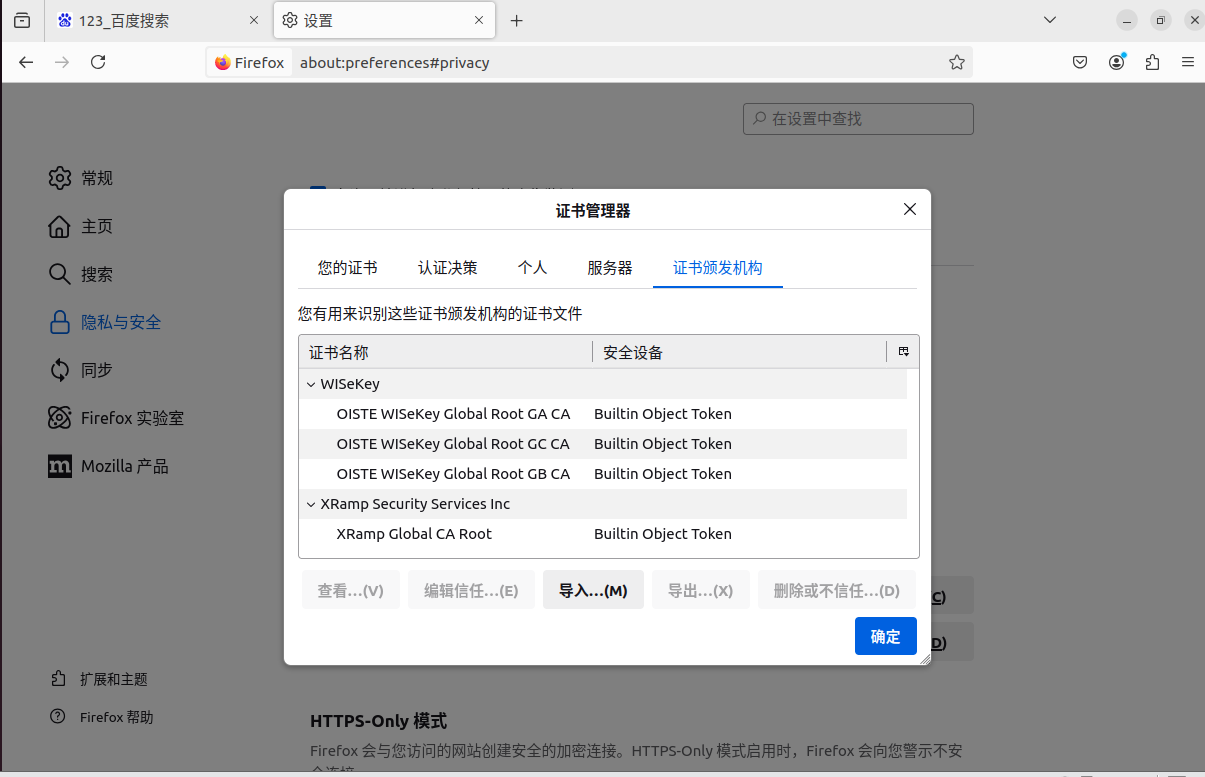
* 双击ca-cert.pem → 安装证书 → 本地计算机 → 受信任的根证书颁发机构

​**Linux**:

sudo cp squid-ca-cert.pem /usr/local/share/ca-certificates/

sudo update-ca-certificates

​**Firefox**:

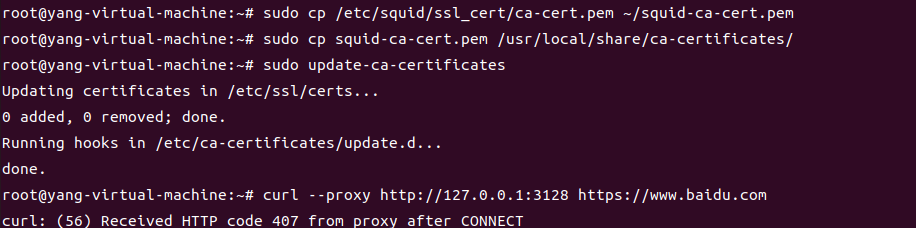
* 选项 → 隐私与安全 → 证书 → 查看证书 → 证书机构 → 导入
* 

**五、测试SSL拦截**

**1. 命令行测试**

curl --proxy http://127.0.0.1:3128 <https://example.com>

重启之后的配置与ssl拦截的命令行结果图：

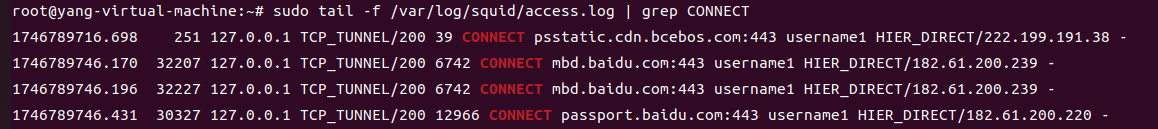


**2. 浏览器测试**

1. 配置浏览器使用代理(127.0.0.1:3128)
2. 访问https网站，确认证书是由你的CA签发

**3. 查看解密内容**

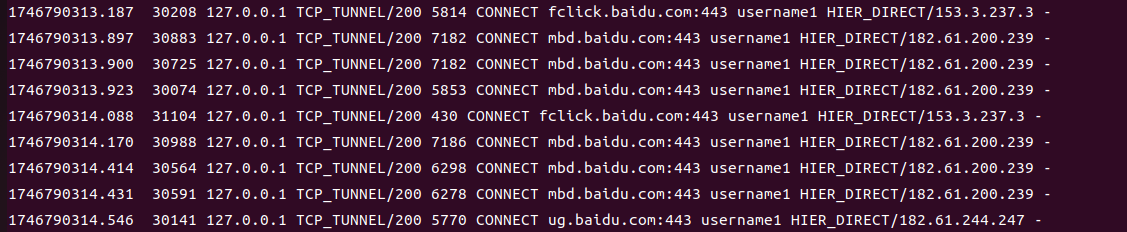
sudo tail -f /var/log/squid/access.log | grep CONNECT



**六、验证拦截效果**

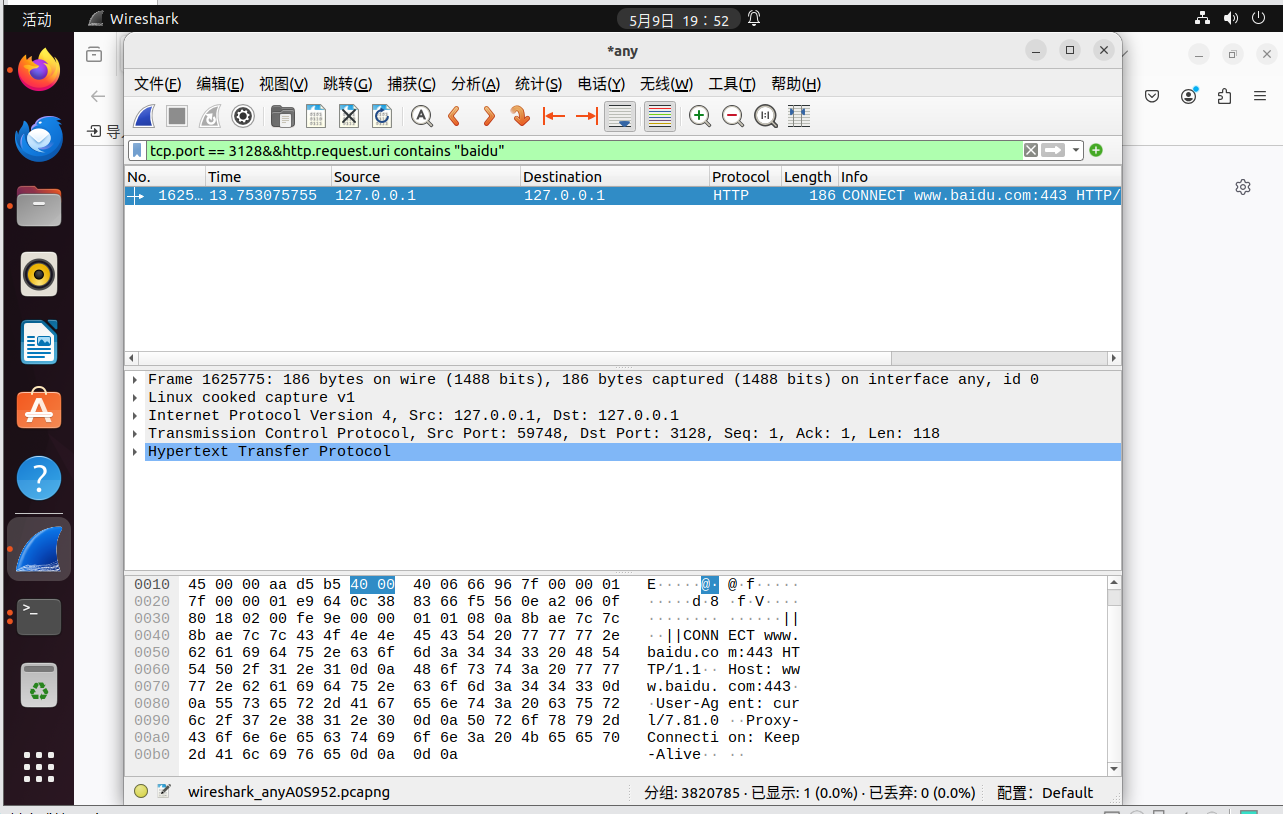
**1. 检查日志中的HTTPS请求**

sudo grep 'CONNECT' /var/log/squid/access.log



**2. 使用Wireshark验证**

1. 捕获本地流量
2. 确认TLS流量被解密(能看到HTTP明文)



**触发流量​**

1. 在终端生成测试流量：

curl -x http://127.0.0.1:3128 https://example.com

1. 或通过已配置代理的浏览器访问任意HTTPS网站

**​识别解密后的流量​**

1. 在Wireshark中查找关键特征：
   * ​**TLS握手过程**​：搜索Client Hello和Server Hello数据包
   * ​**代理连接**​：查找包含CONNECT example.com:443的HTTP请求
2. 成功解密的表现：
   * 能看到明文HTTP请求头（如GET / HTTP/1.1）
   * 应用层协议显示为HTTP而非TLS

### 3.5 安装和测试ClamAV

**安装ClamAV​**

**1. 更新软件源并安装**

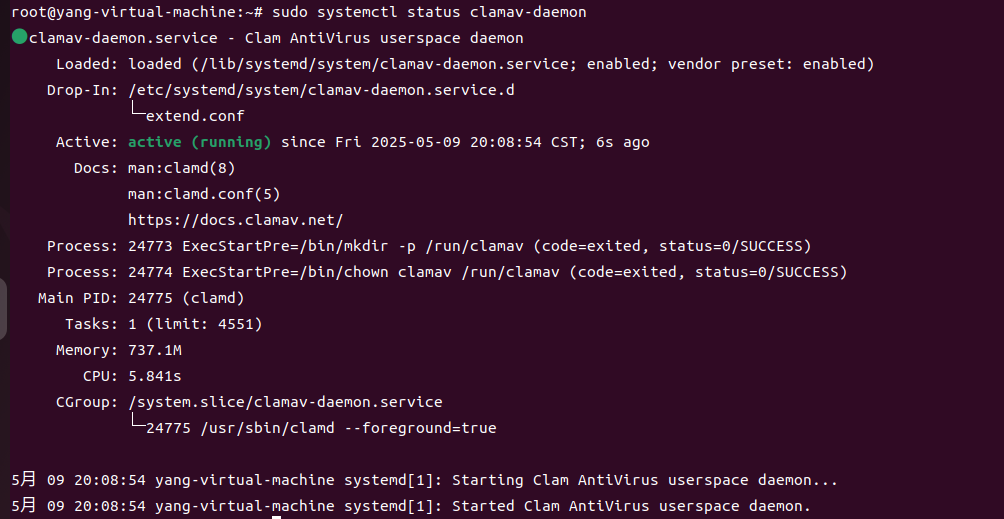
sudo apt update

sudo apt install clamav clamav-daemon clamav-freshclam

**2. 检查服务状态**

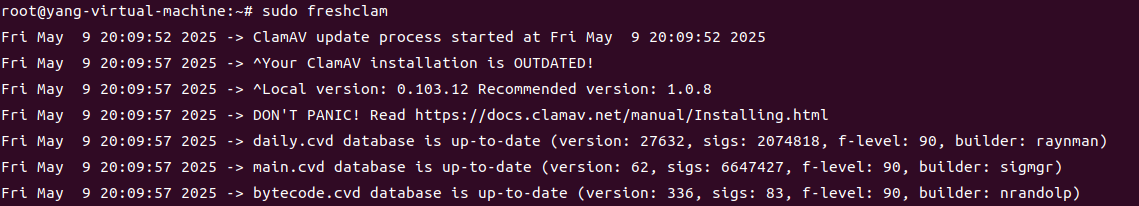
sudo systemctl status clamav-daemon *# 主服务*

sudo systemctl status clamav-freshclam *# 病毒库更新服务*



**3. 手动更新病毒库（首次必须）**

sudo freshclam



​**注意**​：若出现错误Failed to download database，可能是网络问题，尝试：

sudo sed -i 's/DatabaseMirror db.local.clamav.net/DatabaseMirror db.cn.clamav.net/' /etc/clamav/freshclam.conf

sudo freshclam

### ​3.6 配置ClamAV与Squid联动​

**1. 修改ClamAV配置**

sudo nano /etc/clamav/clamd.conf

取消注释或修改以下参数：

|  |
| --- |
| TCPSocket 3310  TCPAddr 127.0.0.1 |

保存后重启服务：

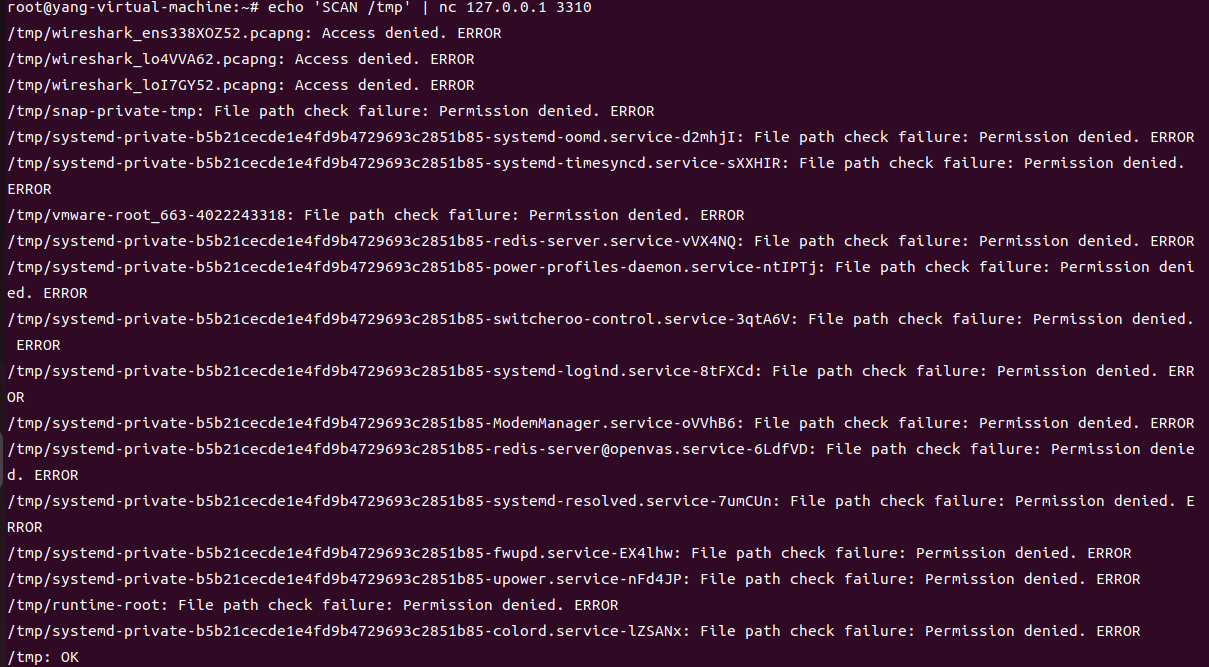
sudo systemctl restart clamav-daemon

**2. 测试ClamAV的TCP接口**

echo 'SCAN /tmp' | nc 127.0.0.1 3310

正常应返回类似：

/tmp: OK



一开始我以为里面有error就是配置失败，查阅后得知，这个输出是正常的，它反映了 ClamAV 对 /tmp 目录的扫描结果。

1. ​\*\*OK 的含义\*\*​
   * 最后的 /tmp: OK 表示 ClamAV ​**对整个目录的扫描完成**，且未在允许访问的文件中发现病毒。
2. ​\*\*ERROR 的原因\*\*​
   * 每个 Access denied. ERROR 或 Permission denied. ERROR 表示 ClamAV ​**无法访问某些子目录或文件**，原因包括：
     + ​**系统保护机制**​：/tmp 下的 systemd-private-\* 目录是 systemd 的私有临时目录，默认仅允许对应服务访问。
     + ​**权限限制**​：ClamAV 以 clamav 用户运行，无权读取其他用户/服务的私有文件（如 Wireshark 的抓包文件）。

### ​3.7病毒扫描测试​

**1. 创建测试病毒文件（无害）**

echo 'X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H\*' > eicar.txt

**2. 手动扫描测试**

clamscan --infected --remove eicar.txt

​**预期输出**​：

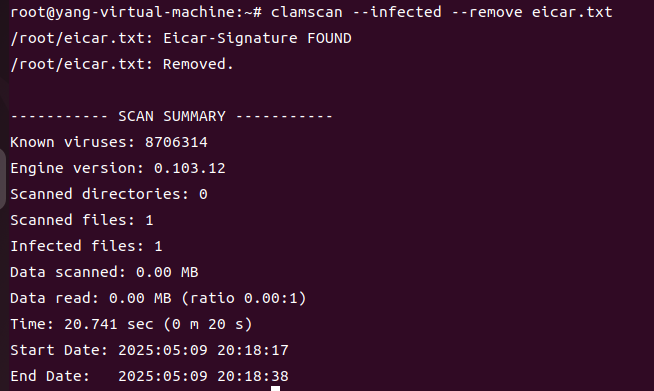
eicar.txt: Eicar-Test-Signature FOUND

----------- SCAN SUMMARY -----------

Infected files: 1

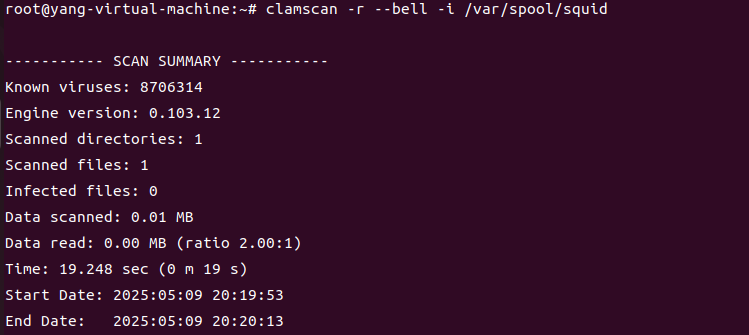
**实际输出：**

能够扫描并且移除



**3. 自动监控扫描（需与Squid集成）**

clamscan -r --bell -i /var/spool/squid *# 扫描Squid缓存目录*



### ​3.8与Squid集成（ICAP协议）​​

**3.8.1. 安装c-icap模块**

sudo apt install c-icap c-icap-modules

我使用的是Ubuntu22，在 Ubuntu 22.04 中，c-icap-modules 软件包可能已被移除或重命名。

直接安装主程序

sudo apt install c-icap

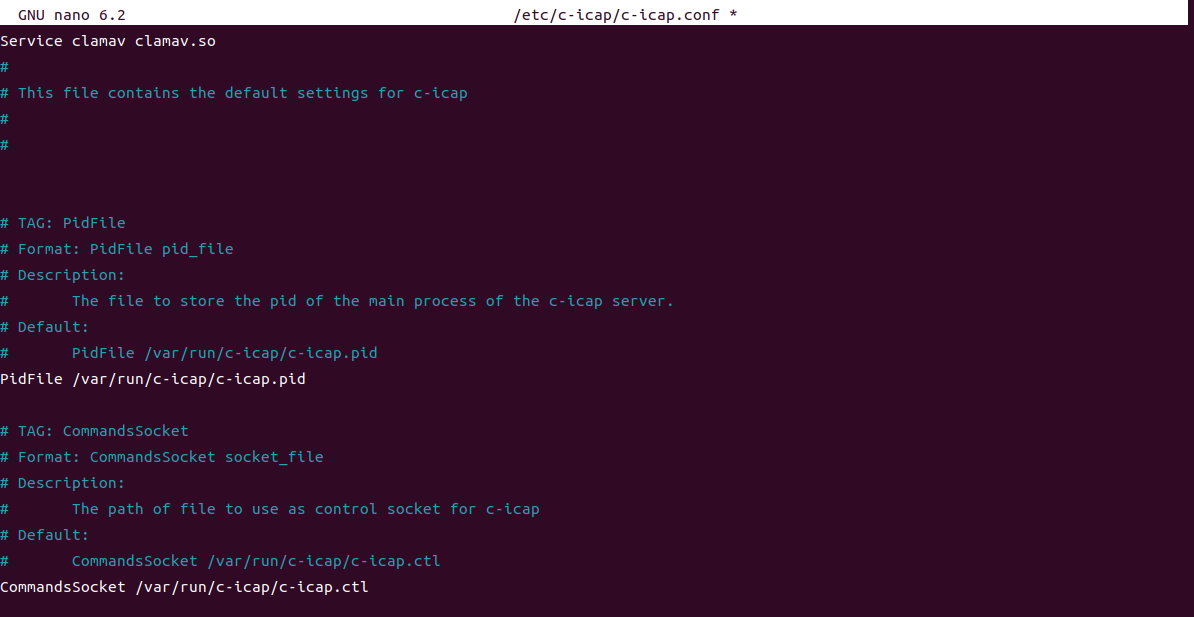
**3.8.2. 配置c-icap服务**

编辑配置文件：

sudo nano /etc/c-icap/c-icap.conf

添加ClamAV服务：

Service clamav clamav.so



这个也有800行，为了不重复上个配置的错误，我直接在最上面添加了这个服务。

**3.8.3. 重启服务**

sudo systemctl restart c-icap squid

**3.8.4. 验证ICAP扫描**

**​3.8.4.1. 准备测试环境​**

**(1) 确保服务已启动**

sudo systemctl restart c-icap squid clamav-daemon

**(2) 创建测试文件**

* ​**病毒测试文件**​（无害）：

echo 'X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H\*' > /var/www/html/eicar.txt

* ​**普通测试文件**​：

echo 'This is a clean file' > /var/www/html/clean.txt

**​3.8.4.2. 实时监控ICAP日志​**

打开两个终端窗口：

**​终端1​：监控ICAP日志**

sudo tail -f /var/log/c-icap/access.log | grep --line-buffered 'ICAP/'

**​终端2​：触发扫描请求**

# 通过代理访问测试文件

curl -x http://127.0.0.1:3128 http://localhost/eicar.txt

curl -x http://127.0.0.1:3128 <http://localhost/clean.txt>

//这里连接本地需要账户密码，例如命令应改为：

curl -v -x http://username1:123456@127.0.0.1:3128 http://localhost/eicar.txt

curl -v -x http://username1:123456@127.0.0.1:3128 http://localhost/clean.txt

而且访问本地需要添加配置许可，下载网络连接的Web服务：

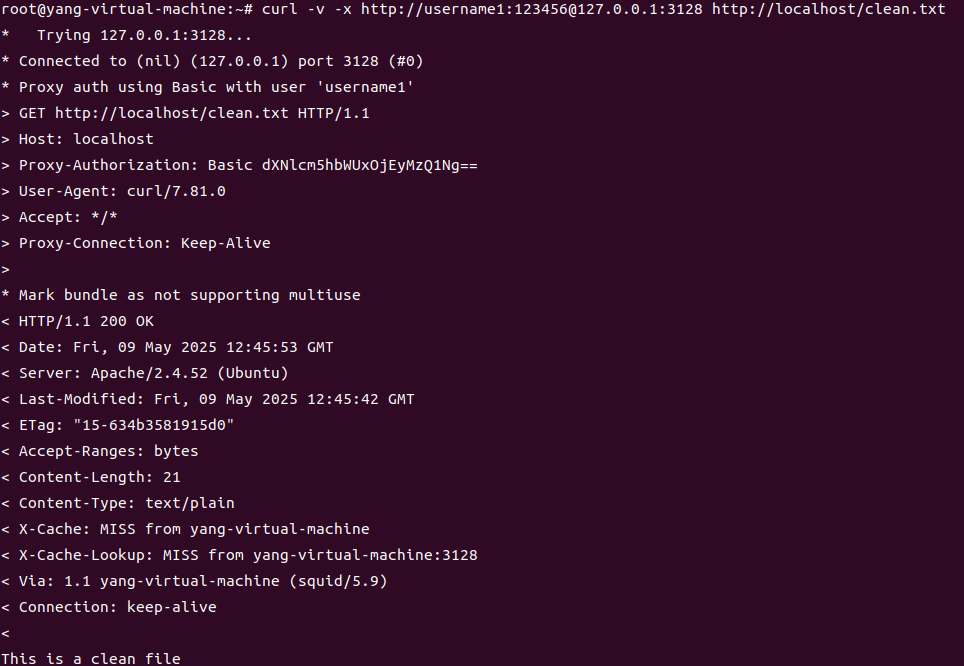
# 安装并启动Apache

sudo apt install apache2

sudo systemctl start apache2

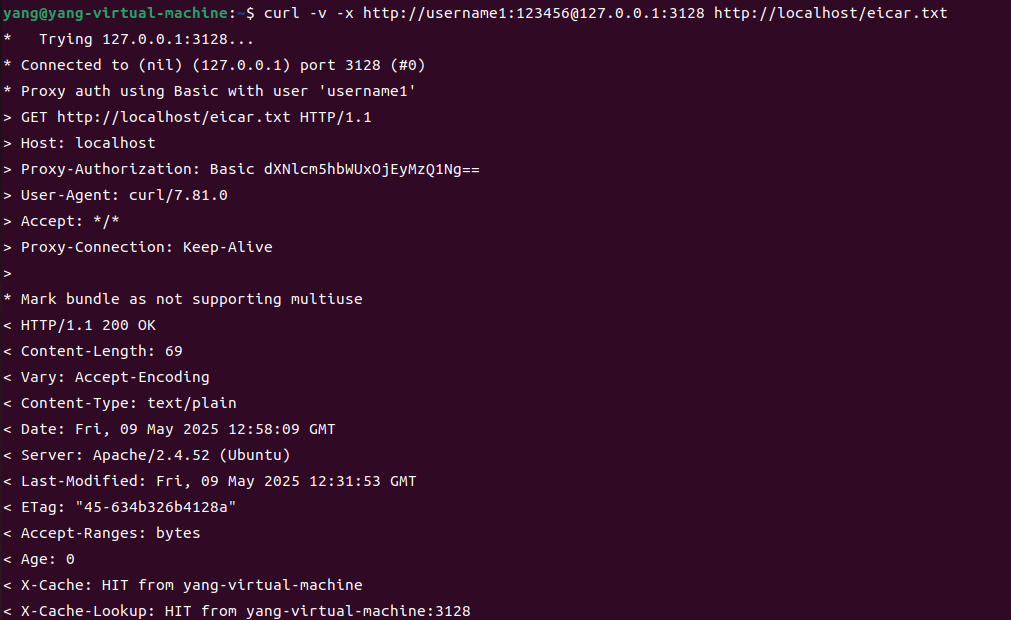
# 创建测试文件

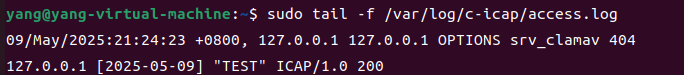
echo "This is a clean file" | sudo tee /var/www/html/clean.txt  
这样才能利用代理连接到自己的计算机：成功打开clean.txt的界面如下：



回到上面的那一步，一个终端监视，另一个终端访问：

发现配置的时候缺了一个模块，再补上才能运行，原来访问不记录，现在终于记录了，





**​3.8.4.3. 验证日志输出​**

正常应看到以下两种响应：

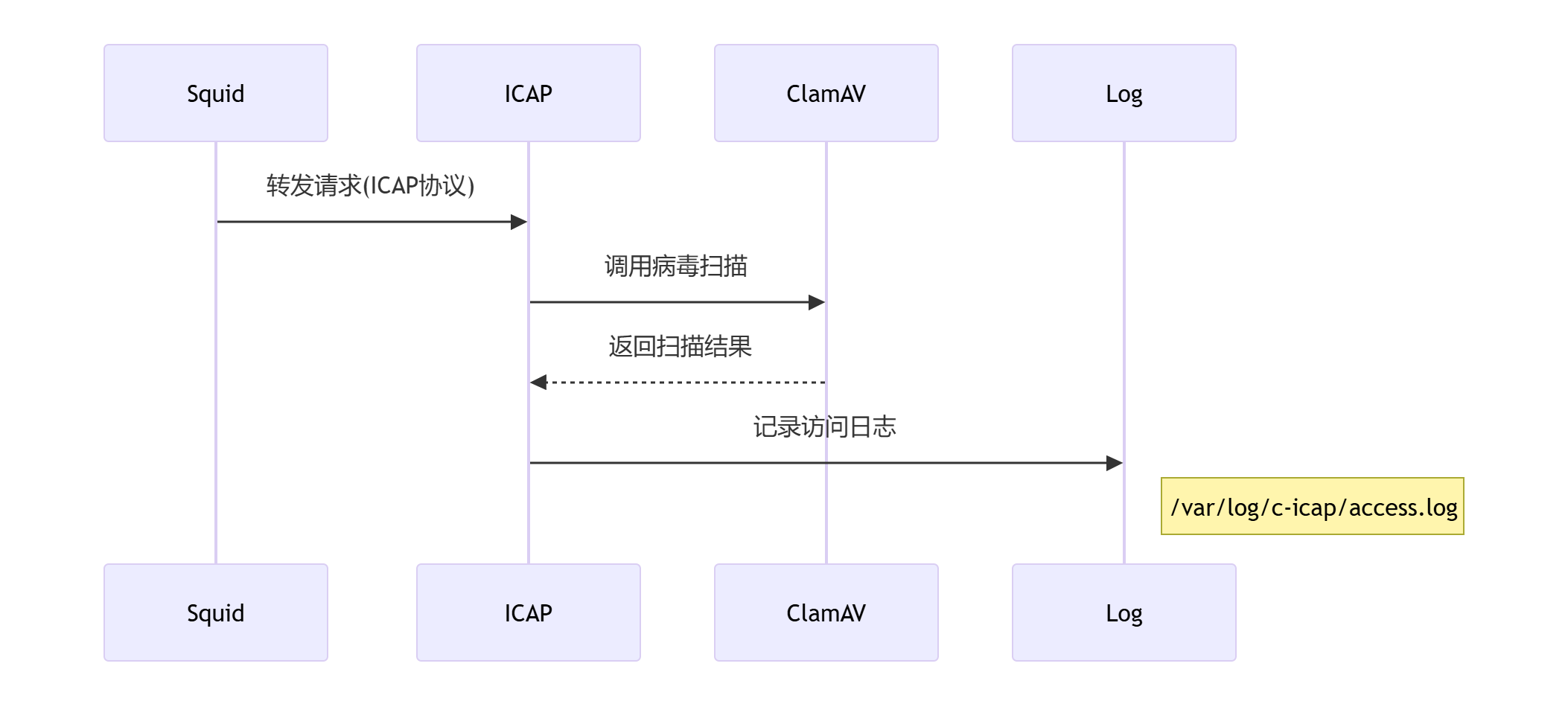
**(1) 检测到病毒文件**

ICAP/1.0 200 OK - clamav: Eicar-Test-Signature FOUND

**(2) 干净文件**

ICAP/1.0 200 OK - clamav: CLEAN

流程图：

**​** ****

### 3.9 ICAP的进一步应用：DLP设计

DLP的设计另起文件进行阐述：



## 四、实验结果

通过本次实验，我们成功安装并配置了Squid代理服务器，集成了ClamAV进行病毒查杀，并初步实现了基于ICAP协议的DLP系统。实验结果表明，Squid与ICAP的结合能够有效监控和过滤网络传输内容，达到网络安全防护的目的。

## 五、实验总结

本次实验让我们深入了解了Proxy和ICAP在网络安全中的应用，掌握了Squid代理服务器的安装与配置方法，学会了使用ClamAV进行病毒查杀，并初步实现了基于ICAP协议的DLP系统。通过实验，我们不仅提高了实际操作能力，还增强了对网络安全技术的理解和应用能力。

## 六、参考文献

1. Squid官方文档：<http://www.squid-cache.org/>
2. ClamAV官方网站：<http://www.clamav.net/>
3. c-icap项目主页：<https://github.com/c-icap/c-icap>
4. ICAP协议介绍：<https://wiki.squid-cache.org/Features/ICAP>