



《网络攻防基础》

实验 1A & 1B 讲解

(Rev: for 软件与系统安全 2023)

刘鹏

中国科学院大学 · UCAS

Contents

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 实验内容介绍 (A/B 二选一即可)

- 1A：嗅探器设计与实现
 - 1B：安全文件传输软件设计与实现

- 实验实施步骤

- 作业提交方式 (2023年10月29日23:55前提交到课程网站)

- 不要压缩为 rar、7zip 等格式，只要 ZIP，不要加密。
 - 文件组织与命名方式见下页
 - 所有代码必须开源，GitHub/Gitee/GitLab 有完整记录。担心代码被抄袭的同学，可以选择在提交之后将 repo 设置为 public.

- 实验考核及评分准则

- 共有 55 人选课，最终将根据完成情况按排名打分。

Commit Guides

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

1A: 嗅探器设计与实现

文件夹	张三_202318008829001_EX1A
1. 源码文件夹	张三_202318008829001_ex1A_src
2. 演示视频	张三_202318008829001_ex1A_play.mp4
3. 文档	张三_202318008829001_ex1A_report.pdf/docx

1B: 安全文件传输软件设计与实现

文件夹	张三_202318008829001_EX1B
1. 源码文件夹	张三_202318008829001_ex1B_src
2. 演示视频	张三_202318008829001_ex1B_play.mp4
3. 文档	张三_202318008829001_ex1B_report.pdf/docx

无需 README, 想说的全放在文档里即可。Git 链接必须写进去。

温馨提示

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 在本课程内容方面，学生本身水平参差不齐
 - 学科跨度大
 - 学科基础薄弱
- 科研压力大
 - 同时兼具一些科研任务、杂活儿，有能力但没心力
 - 无法分清科研和学习的差别，侧重点模糊
- 一心找工作
 - 把实验做好，是找工作项目介绍时的 highlight

本节课的目的就是带领同学们开阔眼界！

实验 1A：Introduction

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

1A：带 GUI 的网络嗅探器 设计与实现

实验 1A: Introduction

Contents

Exp I

ISTs I

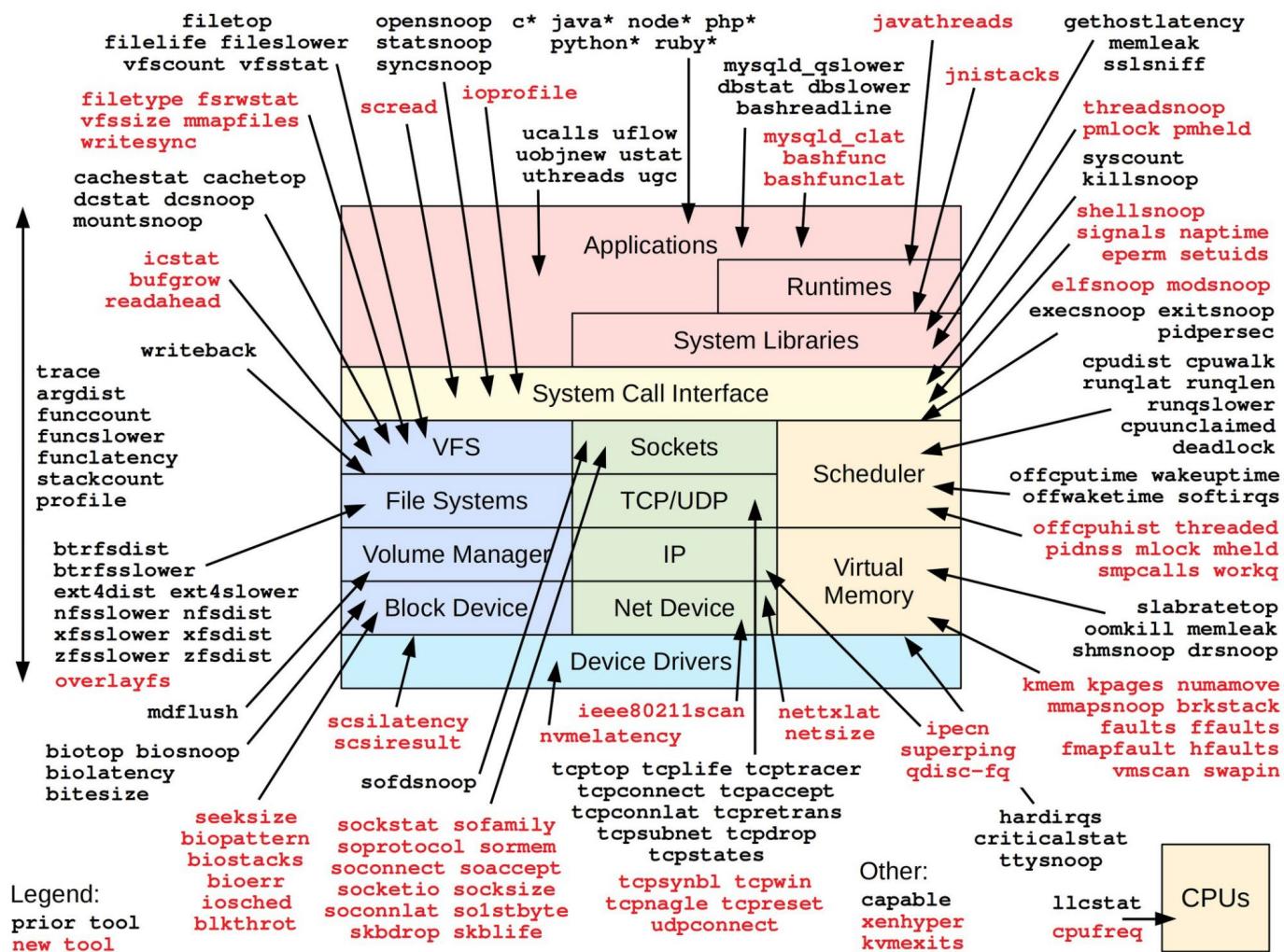
Prompt I

Exp II

ISTs II

Prompt II

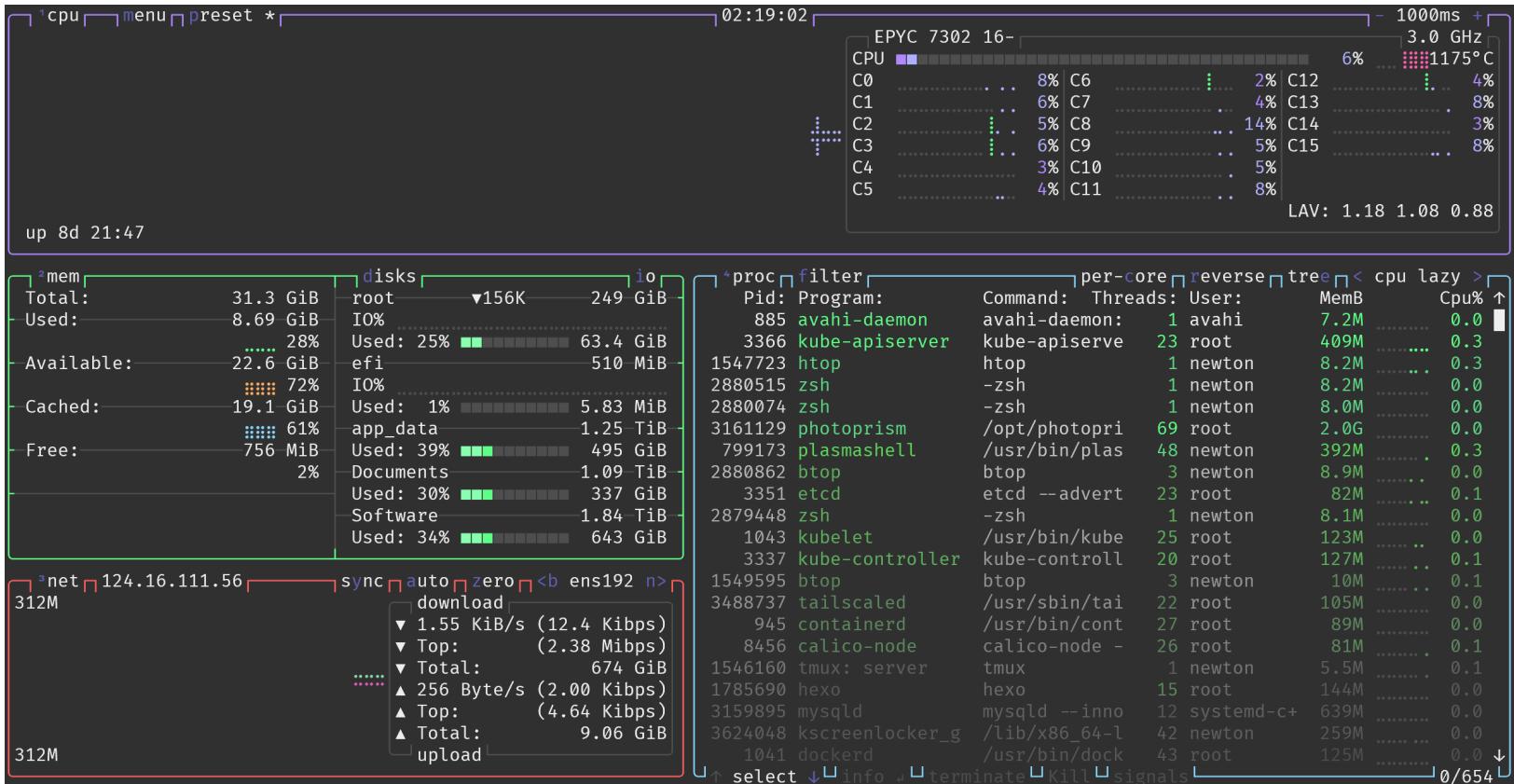
Q&A



实验 1A：Introduction

Contents

Exp I



Linux `btop` 命令输出
(使用 libncurses 开发 TUI 界面)

实验 1A：Introduction

Contents

Exp I

ISTs I

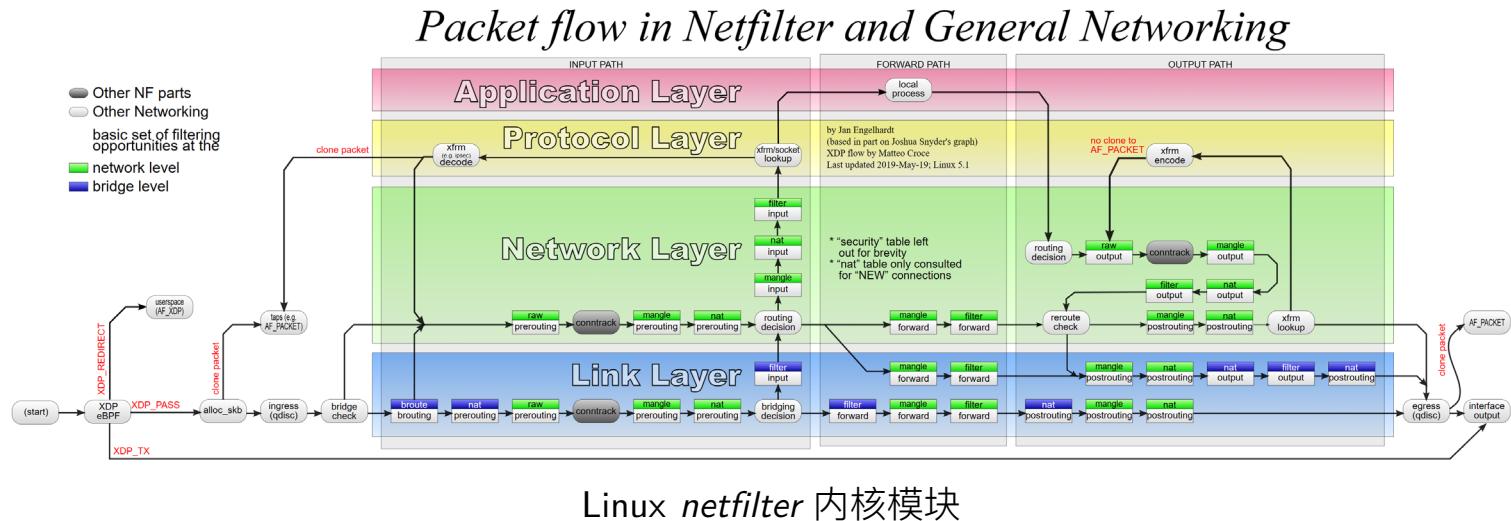
Prompt I

Exp II

ISTs II

Prompt II

Q&A



实验 1A：嗅探器设计与实现

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

• 网络嗅探器

- 介绍：网络嗅探器，Network Packet Sniffer (NPS)，是一种专门用来进行网络流量监听的工具。数据包嗅探器是研究网络行为学的基础工具。
- 本次实验中的嗅探器与常见的 Wireshark 等工具不同
 - Wireshark 等工具实现了强大的数据包协议分析功能，属于 Network Protocol Analyzer (NPA).
 - 请各位思考，如何识别某种特定类型的应用层报文。

• 温馨提示

- 实验中必须实现 NPS 功能（否则不及格）
- 实验中尽量完善 NPA 功能（否则分不高）
- 如果有同学可以基于 Wireshark 做良好的二次开发，也会有加分。

Demo: Wireshark

Contents

Exp I

ISTs I

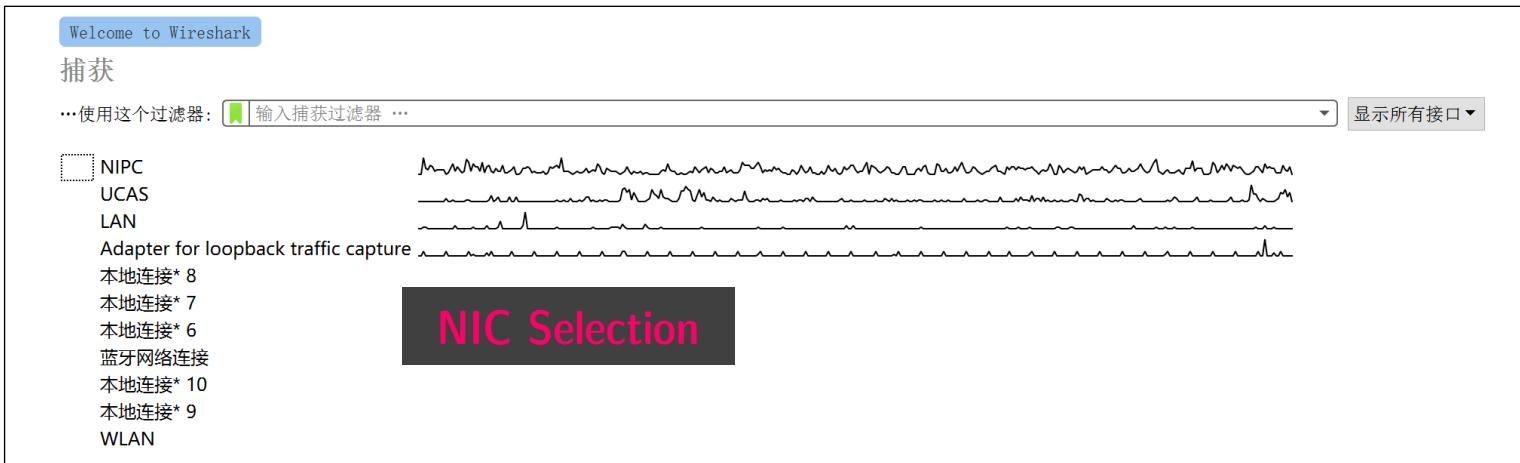
Prompt I

Exp II

ISTs II

Prompt II

Q&A



Demo: Wireshark

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

Standard 3-pane Packet Browser

Packet List

Packet Details

Packet in Binary

Frame 68: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{295CF37B-0794-4165-9C0B-03B9D06D6F52}, id 0
Ethernet II, Src: Giga-Byt_ad:01:74 (b4:2e:99:ad:01:74), Dst: OpenWRT.lan (00:0c:29:c6:8d:de)
Internet Protocol Version 4, Src: 10.0.0.3 (10.0.0.3), Dst: n169-1.mail.139.com (120.232.169.1)
Transmission Control Protocol, Src Port: 14220, Dst Port: 993, Seq: 490, Ack: 4839, Len: 0

0000 00 0c 29 c6 8d de b4 2e 99 ad 01 74 08 00 45 00 ..)....
0010 00 28 b1 24 40 00 40 06 00 00 0a 00 00 03 78 e8 .(\$@ @
0020 a9 01 37 8c 03 e1 13 ec 47 c1 86 20 55 89 50 10 ..7....
0030 03 ff 2c 07 00 00 ..,....

wireshark_eth0UBQW00.peapng | 分组: 729 • 已显示: 729 (100.0%) • 已丢弃: 0 (0.0%) | 配置: Default

Demo: CommView

Contents

Exp I

ISTs I

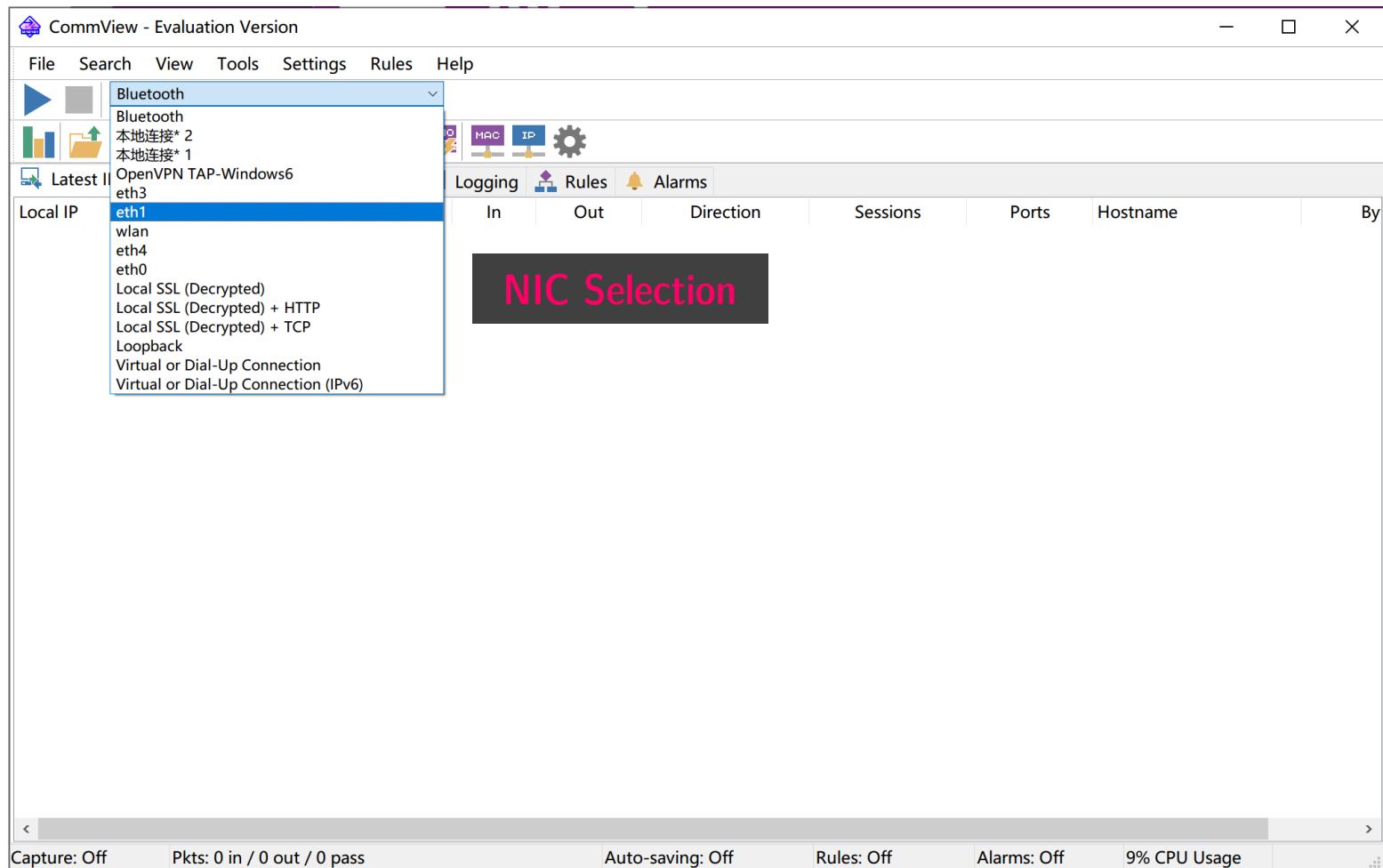
Prompt I

Exp II

ISTs II

Prompt II

Q&A



Demo: CommView

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

The screenshot shows the CommView - Evaluation Version application window. The menu bar includes File, Search, View, Tools, Settings, Rules, and Help. The toolbar features icons for capturing (blue triangle), stopping (grey square), and saving (floppy disk). A dropdown menu shows 'eth0' selected. Below the toolbar is a status bar with 'Capture: Off', 'Pkts: 758 in / 832 out / 385 pass', 'Auto-saving: Off', 'Rules: Off', 'Alarms: Off', '3% CPU Usage', and a settings gear icon.

The main pane displays a table of 'Latest IP Connections'. The columns are Local IP, Remote IP, In, Out, Direction, Sessions, Ports, Hostname, Bytes, and Process. The table lists numerous connections, such as 10.0.0.3 to 192.168.137.1 (Inbound, 151 bytes), 10.0.0.88 to 239.255.255.250 (Outbound, 156 bytes), and various loopback and broadcast addresses. The 'Ports' column includes entries like 36959,ssdp, 5353, domain, public1.alidns.com, https, netbios-ns, 9999, 14674, 14005, 14608, 14609, 14111, 14638, 14681, imaps, and https. The 'Process' column indicates the number of system processes involved in each connection.

A large black box with pink text 'OS Process tracing' is overlaid on the bottom left of the main pane.

实验 1A 要求

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

• 理清 OSI 五层模型的概念

■ 能深入理解抓包层次与分析层次

➤ 思考题：iOS 的 Thor App 可以在安装了系统描述文件之后，抓取 HTTP/HTTPS 的应用层包。同样因为安装了根证书性质的描述文件，故 Thor 能解析 HTTPS 中的内容。但具体到某些传输控制协议的内容，Thor 无能为力。这告诉了我们什么道理？

■ 软件要有一定的协议过滤能力

➤ Wireshark 等软件的协议过滤器支持逻辑演算，所以该软件里含有逻辑推导的组件。基本的协议过滤需要支持筛选 HTTP、TCP/UDP、IPv4/v6、ICMP 等不同类型、层次的数据包。libpcap 提供了数据包筛选功能。

■ 软件有一定的流追踪能力（加分项）

➤ 基于 IP + Port 的 TCP 流

➤ 某进程产生的所有 TCP 流

实验 1A 提示

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

• 基于图形化编程的重要性

- 现代化的软件基本都提供图形化交互界面。
- 图形化的界面对于分析流的时序图具有重要帮助作用。
- 在现代图形 API 的帮助下开发图形化程序并不难。
- 有关现代图形库的帮助，参见 Prompt II.

• 推荐使用的抓包系统 API

- libpcap
- winPcap: 基于 Windows NT 内核定制的 libpcap
- 请大家自行学习上述 lib/dll 的用法，然后在代码中实现调用。

Linux 协议栈

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

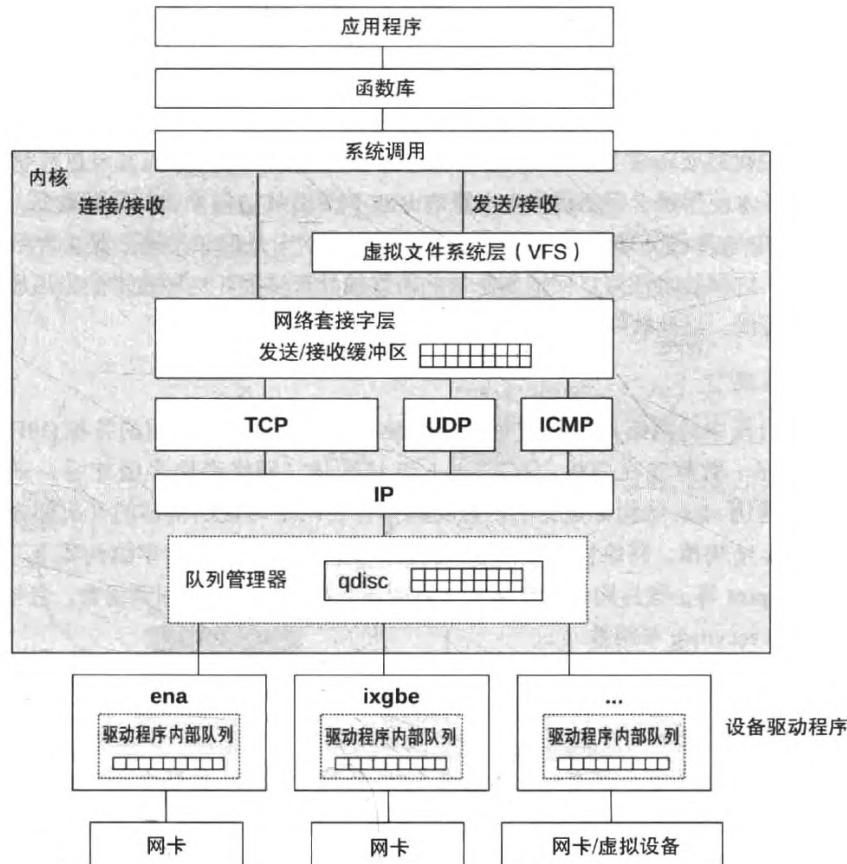


图. Linux 网络协议栈

VFS 抽象层会在 1B 中用到

“一切皆文件”是Linux的基本哲学之一，不仅是普通的文件，包括目录、字符设备、块设备、套接字等，都可以以文件的方式被对待。实现这一行为的基础，正是Linux的虚拟文件系统机制。

Linux 收包过程图解

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

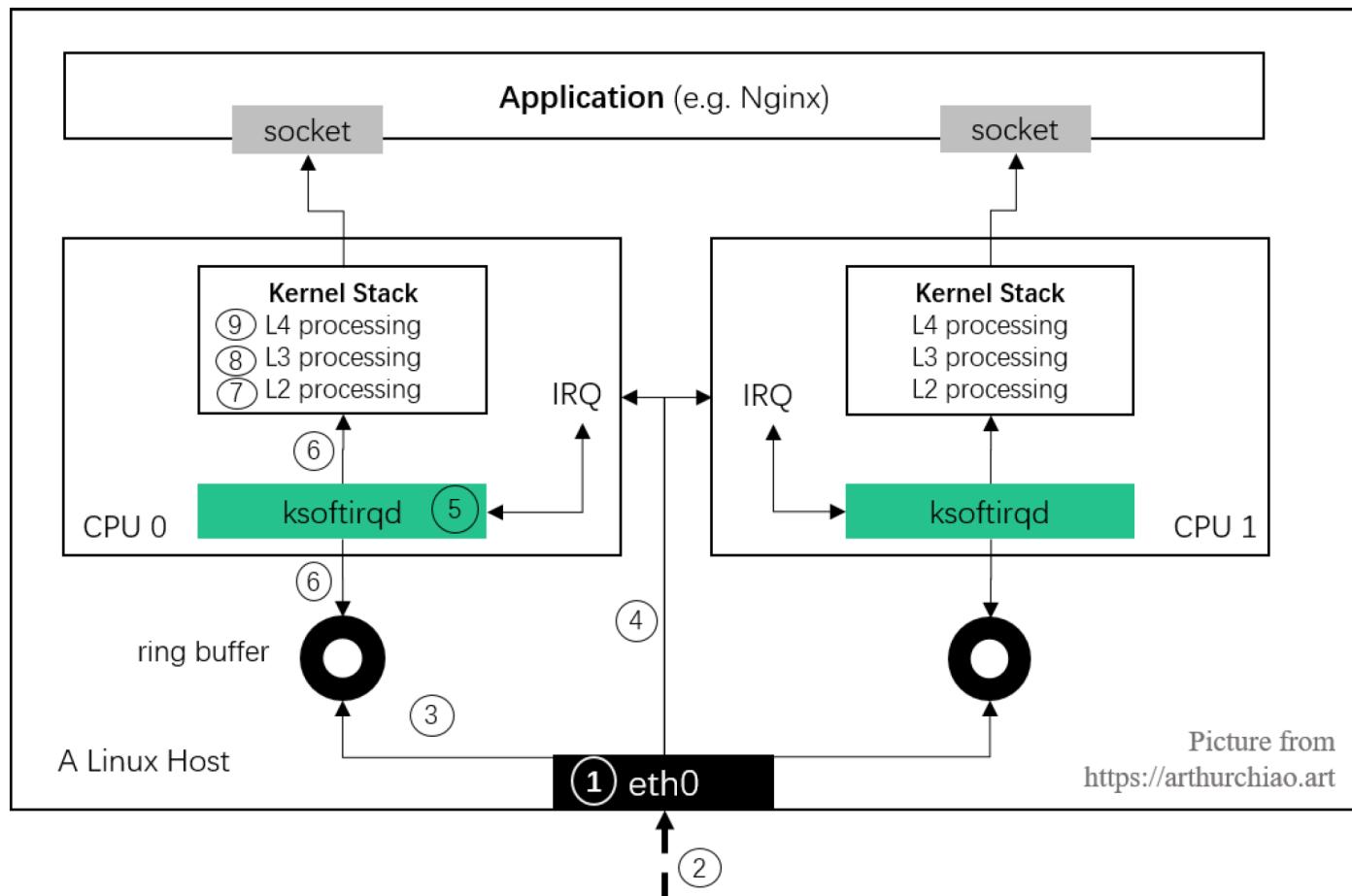


图. Linux 网络收包 RX 过程，内核为先。

Linux netfilter hooks

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

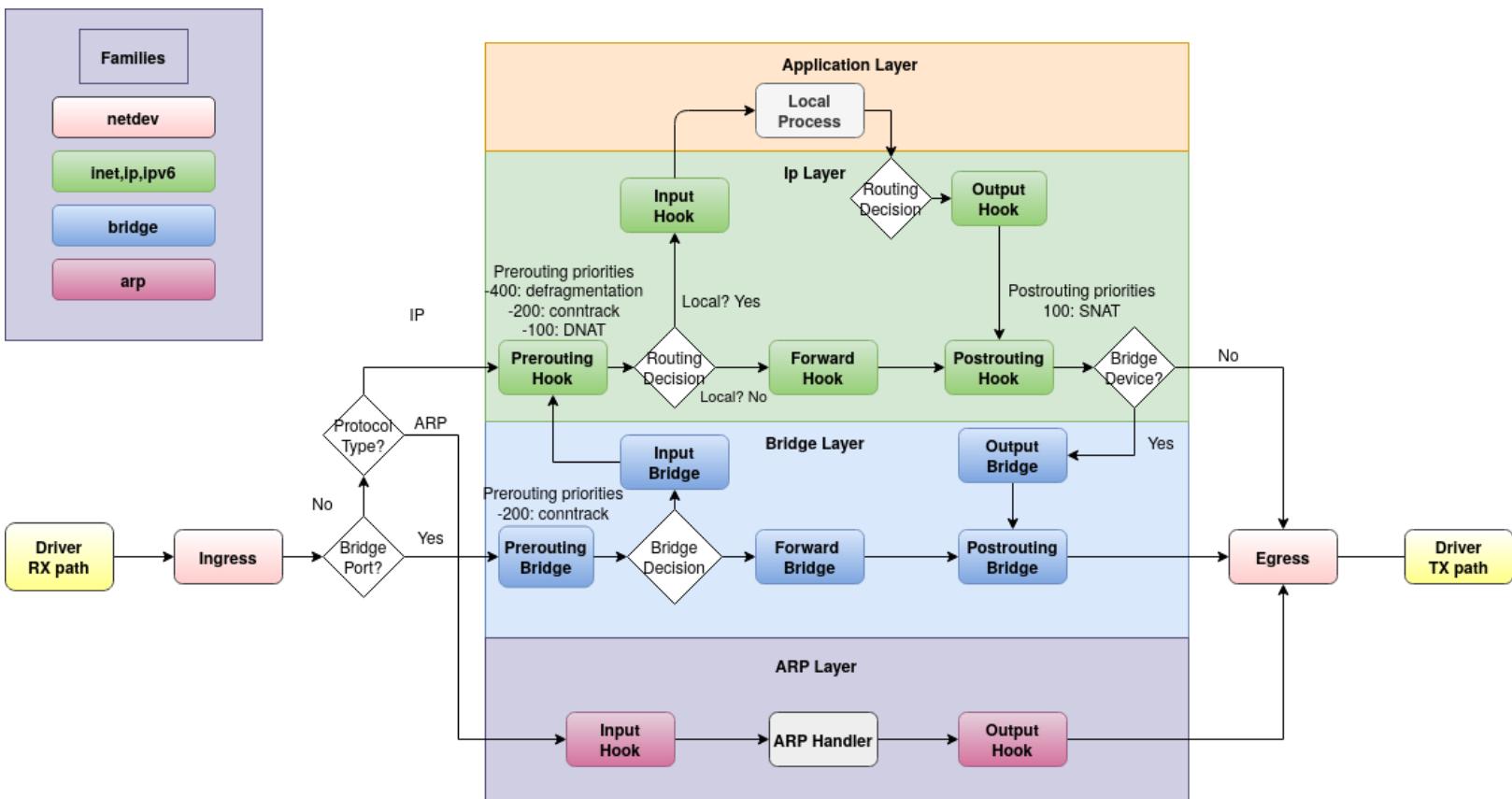


图. Linux 网络收包 RX 过程

Linux Netfilter

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

• Linux 内核注册的 hook

- *NF_IP_PRE_ROUTING* 接收到的包进入协议栈后立即触发此 hook，先于路由判断
- *NF_IP_LOCAL_IN* 接收到的包经过路由判断，如果目的是本机，将触发此 hook
- *NF_IP_FORWARD* 接收到的包经过路由判断，如果目的是其他机器，将触发此 hook
- *NF_IP_LOCAL_OUT* 本机产生的准备发送的包，在进入协议栈后立即触发此 hook
- *NF_IP_POST_ROUTING* 本机产生的准备发送的包或者转发的包，在经过路由判断之后，将触发此 hook推荐使用的抓包系统 API

Linux netfilter hooks

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

接口	Bridge	添加绑定	添加网桥	添加 VLAN
名称	IP 地址	发送中	接收中	
br-5c36015c65b8	172.19.0.1/16	0 bps	0 bps	
br-7ed6212d8f3e	172.18.0.1/16	76.3 Kbps	10.6 Kbps	
docker0	172.17.0.1/16, fd00:0:0:0:0:0:1/80	0 bps	0 bps	
ens160	10.2.1.21/16	3.89 Mbps	586 Kbps	
ens192	124.16. [REDACTED] /23, 2400:dd01:103a:4007 [REDACTED] /64	9.62 Kbps	357 Kbps	

图. Linux net interface

Linux BPF Internals

Contents

Exp I

ISTs I

Prompt I

Exp II

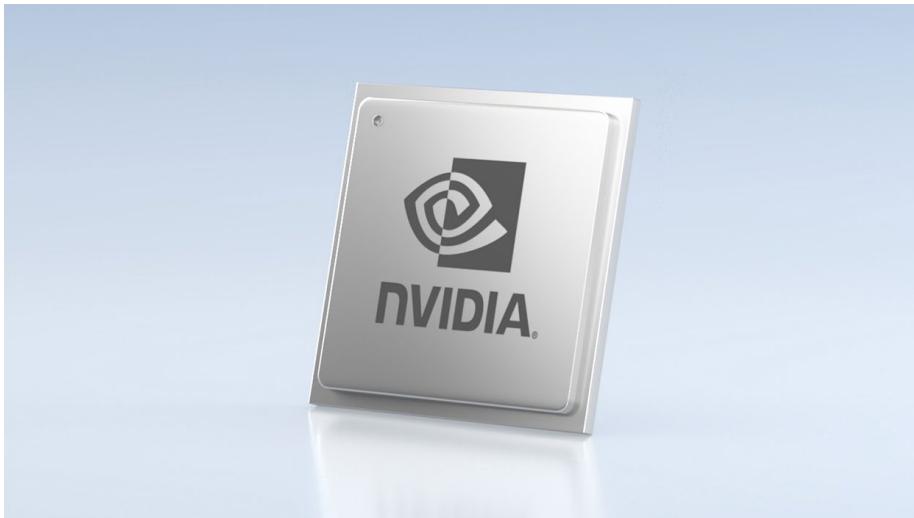
ISTs II

Prompt II

Q&A

• 网络传输速度不断提升

- 商用以太网已经实现 100Gbps 甚至是 400Gbps 的速率
- 如何让抓包工具能匹配网速呢？性能是个关键问题。



PRODUCT SPECIFICATIONS

Max Total Bandwidth	400Gb/s
Supported Ethernet Speeds	10/25/40/50/100/ 200/400GbE
Number of Network Ports	1/2/4
Network Interface Technologies	NRZ (10/25G) / PAM4 (50/100G)
Host Interface	PCIe Gen5.0 x16/ x32
Cards Form Factors	PCIe FHHL/ HHHL, OCP3.0 SFF
Network Interfaces	SFP56, QSFP56, QSFP56-DD, QSFP112, SFP112

Linux BPF Internals

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- GNU/Linux 6.2 为 800Gbps 以太网做准备

■ 消息来源：Phoronix

Linux 6.2 Begins Making Preparations For 800 Gbps Networking

Written by Michael Larabel in Linux Networking on 27 October 2022 at 05:22 AM EDT. 22 Comments



The Linux kernel's networking subsystem is beginning to make preparations for 800 Gbps Ethernet networking.

The IEEE-802 LAN/MAN Standards Committee has published standards for 800 Gbps networking and NVIDIA with continuing to build from their acquired Mellanox IP is working on 800 Gbps networking hardware. NVIDIA engineers working on their Linux driver support have confirmed that their next-generation NVIDIA Spectrum ASIC will indeed support 800 Gbps speeds.

Initially NVIDIA is going to support 800 Gbps speeds using eight lanes at 100 Gbps each while further out they are looking at four lanes at 200 Gbps.



Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A



抓包工具在后台执行会影响性能吗？



抓包工具，如Wireshark、tcpdump等，用于捕获、分析和查看网络流量。当这些工具在后台执行时，是否会影响性能取决于以下几个因素：

- 流量量级**：如果网络流量非常大，抓包工具需要处理和保存大量数据，这可能会对系统性能产生影响。
- 存储方式**：抓包工具通常会将捕获的数据包保存到硬盘上。如果数据包捕获速率很高，并且磁盘I/O速度跟不上，可能会导致性能问题。
- 系统资源**：运行抓包工具需要CPU、内存和磁盘资源。如果系统资源已经很紧张，再运行抓包工具可能会进一步影响性能。
- 过滤条件**：如果设置了合适的过滤条件，只捕获关心的特定流量，那么工具需要处理的数据量就会减少，对性能的影响也会相应减少。
- 工具实现和优化**：不同的抓包工具可能在性能方面有所不同。例如，一些工具可能针对高流量环境进行了优化，而其他工具可能没有。

总的来说，虽然抓包工具在后台执行可能会对性能产生一些影响，但在大多数常规应用中，这种影响是可以接受的。然而，在高流量、资源受限或其他特定场景中，建议在使用抓包工具时进行适当的评估和监控，以确保不会对系统性能产生不良影响。

Ask ChatGPT



Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A



抓包工具，如Wireshark，能否利用Linux内核特性实现高速抓包，因为我的网络流量很大，普通方式难以应付。



是的，为了应对大流量环境，可以结合Linux内核的某些特性或使用特定的库来实现高速抓包。以下是一些建议和方法：

1. **PF_RING**: PF_RING 是一种Linux内核的扩展，它提供了一个更加高效的方式来捕获数据包。PF_RING 可以显著提高数据包捕获的性能，并减少丢包率。它还支持多核处理，可以在多个CPU核心上并行处理数据包。Wireshark 可以与 PF_RING 结合使用。
2. **使用 AF_PACKET 的 TPACKET_V3 模式**: Linux的AF_PACKET套接字提供了一个TPACKET_V3模式，它可以批量接收和发送数据包，提高处理效率。
3. **网卡硬件支持**: 某些现代网卡支持硬件级的流量平衡和多队列功能，这允许多个CPU核心并行处理网络流量，从而提高捕获性能。
4. **NUMA 架构优化**: 在多核、多CPU的系统上，考虑到NUMA（非统一内存访问）的特性，合理地分配和绑定内存和CPU核心可以提高数据包处理的性能。
5. **BPF (Berkeley Packet Filter) 过滤**: 在内核级使用BPF过滤器可以在数据包到达用户空间之前就对其进行过滤，这可以减少不必要的数据包传输和处理。
6. **eBPF/XDP**: eBPF 和 XDP (Express Data Path) 为Linux提供了一个在内核空间执行自定义程序的机制。特别是XDP允许在数据包到达网络栈之前进行处理，这可以用于高效的数据包过滤和转发。
7. **使用DPDK (Data Plane Development Kit)**: 虽然DPDK不是Linux内核的一部分，但它提供了一种绕过内核，直接在用户空间处理数据包的方法。这可以显著提高数据包处理的速度，但可能需要特定的硬件支持。

如果你的环境流量真的非常大，那么可能需要结合上述多种技术并进行适当的调优以满足需求。此外，选择和配置适当的硬件，例如高性能的网卡和服务器，也非常关键。

Linux BPF

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- BPF/eBPF 可解决抓包开销问题

- BPF 已经是一个技术名称综合，与 LLVM 不是底层虚拟机一样，不可“望文生义”。利用 BPF 可以在内核态编程，并且有以下主要优势：

- User-defined programs. 用户可编程

- Limited and secure kernel access. 内核依旧安全

- A new type of software. 与 VM、Docker 类似，已经为一种新技术

- 网络抓包工具 tcpdump 和 Wireshark 就是 BPF 经典案例

<https://www.usenix.org/conference/lisa21/presentation/gregg-bpf>

利用 Linux BPF 开发监控软件

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

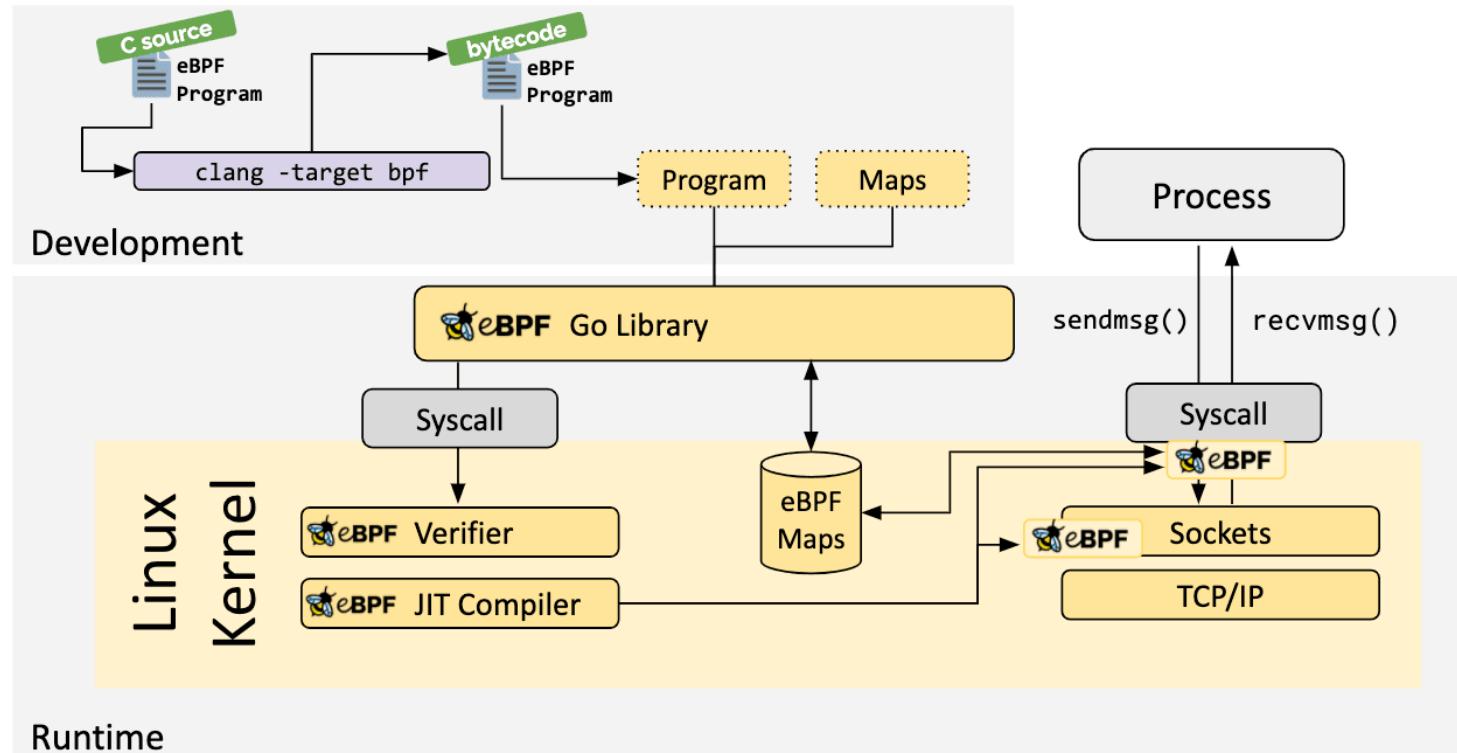


图. BPF 概览

利用 Linux BPF 开发监控软件

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

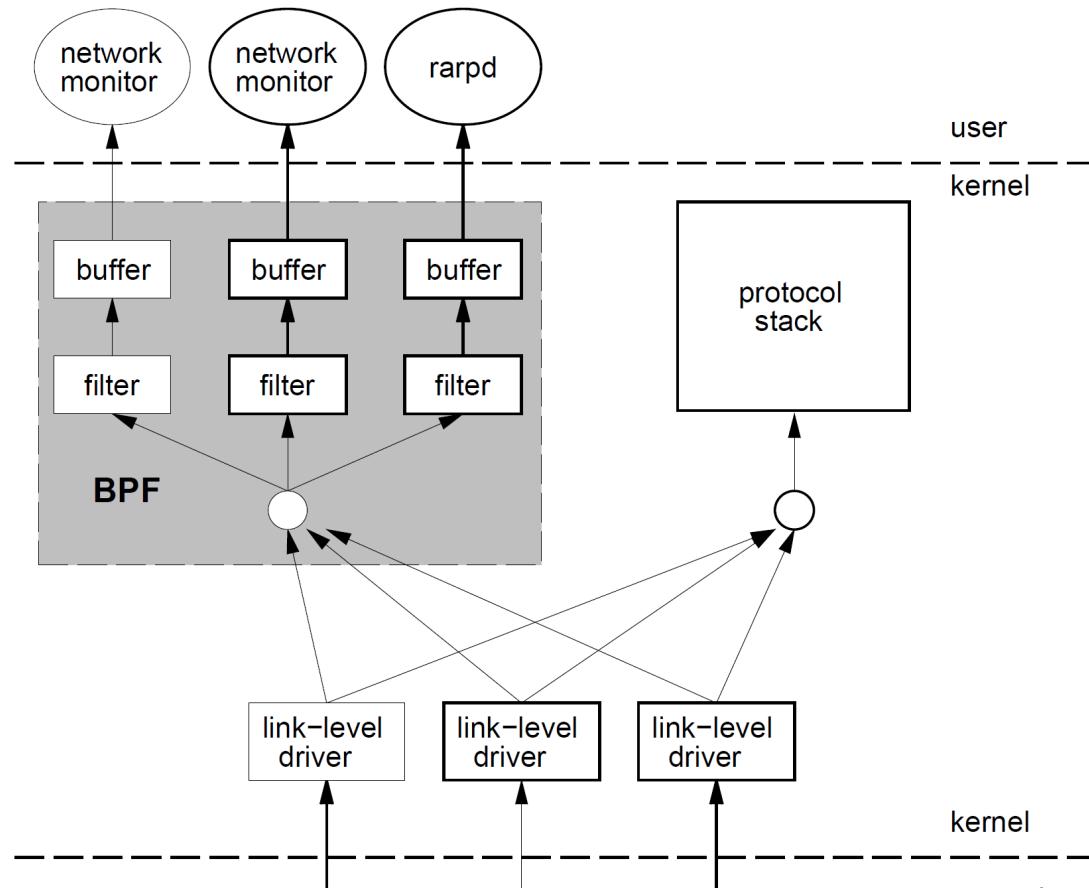


图. BPF 概览

BPF 与 传统编程对比

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

		Execution model	User defined	Compilation	Security	Failure mode	Resource access
	User	task	yes	any	user based	abort	syscall, fault
	Kernel	task	no	static	none	panic	direct
	BPF	event	yes	JIT, CO-RE	verified, JIT	error message	restricted helpers

图 3. 用户态、内核态与 BPF 编程对比

BPF kernel Verifier

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

check_subprogs	check_helper_mem_access
check_reg_arg	check_func_arg
check_stack_write	check_map_func_compatibility
check_stack_read	check_func_proto
check_stack_access	check_func_call
check_map_access_type	check_reference_leak
check_mem_region_access	check_helper_call
check_map_access	check_alu_op
check_packet_access	check_cond_jmp_op
check_ctx_access	check_ld_imm
check_flow_keys_access	check_ld_abs
check_sock_access	check_return_code
check_pkt_ptr_alignment	check_cfg
check_generic_ptr_alignment	check_btf_func
check_ptr_alignment	check_btf_line
check_max_stack_depth	check_btf_info
check_tp_buffer_access	check_map_prealloc
check_ptr_to_btf_access	check_map_prog_compatibility
check_mem_access	check_struct_ops_btf_id
check_xadd	check_attach_modify_return
check_stack_boundary	check_attach_btf_id

Reference 1A

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- https://www.youtube.com/watch?v=5Z2AU7QTH4&ab_channel=USENIX
- (开发必备) libpcap官方仓库: <https://github.com/the-tcpdump-group/libpcap>
- (隆重推荐) 深度网络知识: <http://arthurchiao.art/articles-zh/>
- BPF 官方网站: <https://ebpf.io/>
- BPF 中文教程: <https://github.com/eunomia-bpf/bpf-developer-tutorial>

实验 1B：安全文件传输软件

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 文件传输是网络的基本功能之一

- 文件传输属于抽象需求，可细分为多种场景

- HTTP：通过构造 HTTP 请求实现文件传输（非典型）

- WebDAV：Http 的扩展，用于在 80 端口进行文件共享

- FTP：远比 HTTP 复杂，专用于文件传输，不加密

- SFTP：FTP 的扩展，保持同样的接口，但是支持加密

- SCP：Secure Copy，基于 SSH 协议在两台主机直接传输

- SMB：微软独家的 Windows 电脑间的文件共享、打印共享协议，性能安全优异

- NFS：Unix 系统间的文件系统级共享，权限控制比较弱

- Note：文件共享有别于块存储共享！

- 块存储常见于 IDC，与本实验不同，但是功能比较相似

- IP-SAN、FC-SAN：基于 SAN 存储网络的高性能块共享

- iSCSI：一种广泛应用的 IP-SAN 技术

Demo: WinSCP

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

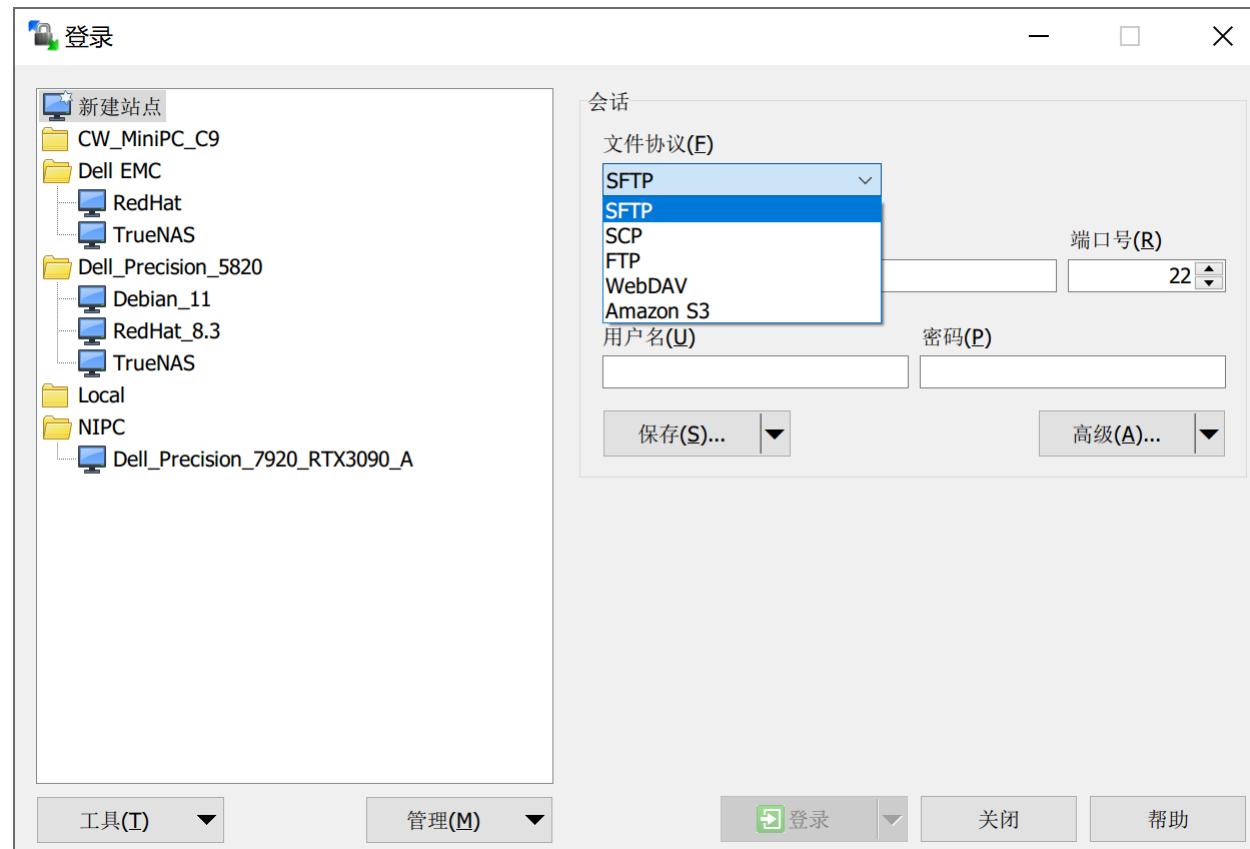


图. WinScp 软件配置界面

实验 1B：要求与提示

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 要求实现一种简单但安全的文件传输协议
 - 能多用户并发访问，文件可完整地传输
 - 通信全过程可抵御中间人、DDoS、SQL 注入等常见攻击
- 由于用来判卷的电脑是 Windows 系统，因此用其他 OS 的同学，需要提供编译成功、运行成功的视频。
 - 视频里可以讲解一下自己的代码与思路，注意录制时长
 - 视频要展示程序编译、部署、测试的全过程
- 建议 (1) 使用 Visual Studio 2022 实现客户端图形化
 - Windows 的 WinForm、WPF 图形库
 - macOS 的 Cocoa 图形库
 - Linux 的各种图形库 (xxQT)
 - 基于 Native API 的图形库封装：wxWidgets

实验 1B：要求与提示

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 建议 (2) 使用 Web 进行开发

- SaaS 架构

- 通过 HTTPS 做加密通信

- 自定义认证方式，可参考 OAuth, 2FA 方案

- 建议使用 Java, Node.JS 做后端

- Vue 等前端框架

- 使用容器做部署，比如 Docker + K8S

实验 1B：要求与提示

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

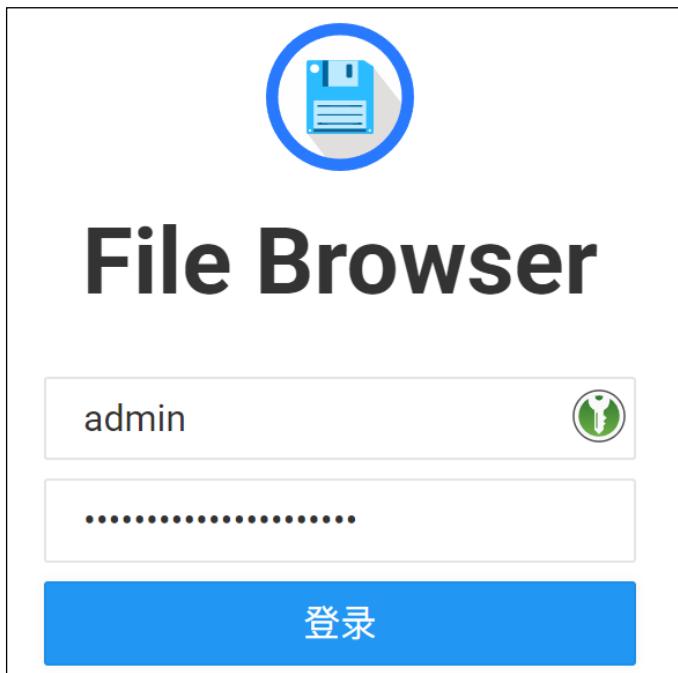


图. 认证界面，使用 Token 认证

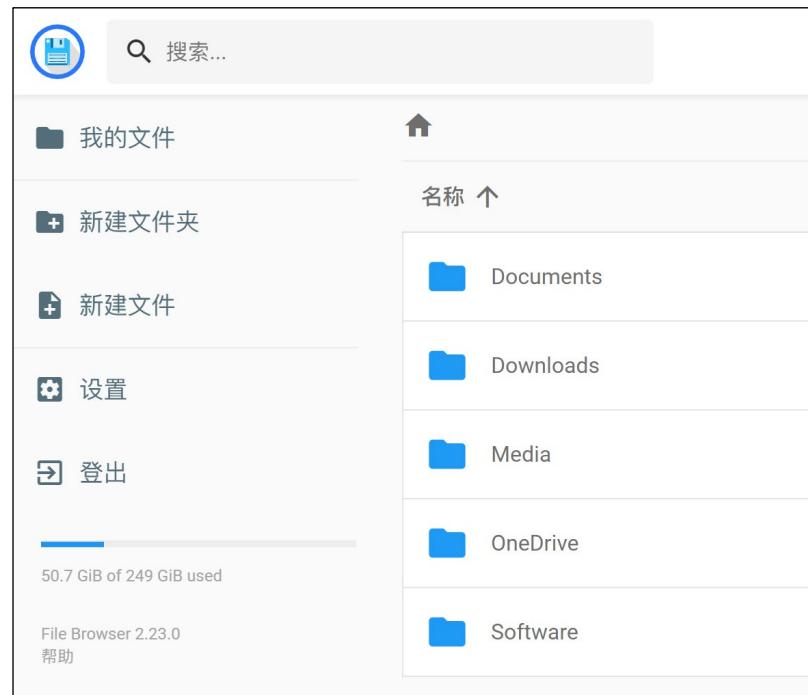


图. 文件浏览



图. 使用容器部署

实验 1B：网络应用设计

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 需要建立长连接，代码中要明确指示 keep alive
 - 以下行为是不允许的：Client ~~发起一次请求并得到响应之后，即关闭 TCP 连接。~~ 需要有守护进程维护连接，直至 Client 被手动关闭。
- 需要在 Report 明确写出应用层协议的设计原理
 - 图文并茂地叙述协议设计的思路
 - 时序图（协议的每个模块都要画出时序图）
 - 协议头部信息描述图（类似 TCP 头部）
 - 解释协议能准确运行的合理性

实验 1B：应用层协议设计

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 需要建立长连接，代码中要明确指示 keep alive
 - 以下行为是不允许的：Client ~~发起一次请求并得到响应之后，即关闭 TCP 连接。~~ 需要有守护进程维护连接，直至 Client 被手动关闭。
- 需要在 Report 明确写出应用层协议的设计原理
 - 图文并茂地叙述协议设计的思路
 - 画出时序图（协议的每个模块都要画出时序图）
 - 给出协议头部信息描述图（类似 TCP 头部）
 - 解释协议能准确运行的合理性

功能流程图: Server

Contents

Exp I

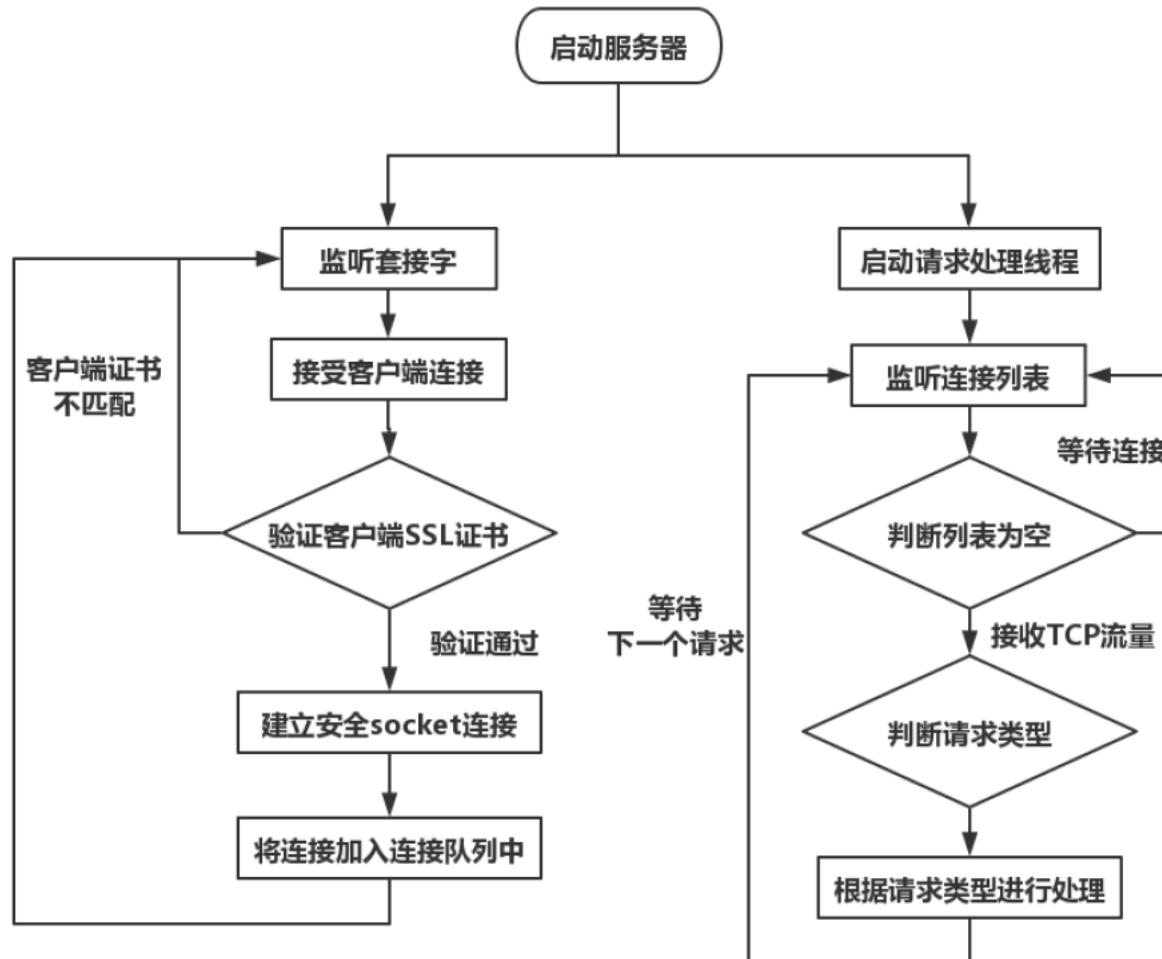
ISTs I

Prompt I

Exp II

ISTS II

Prompt II



协议时序图

Contents

Exp I

ISTs I

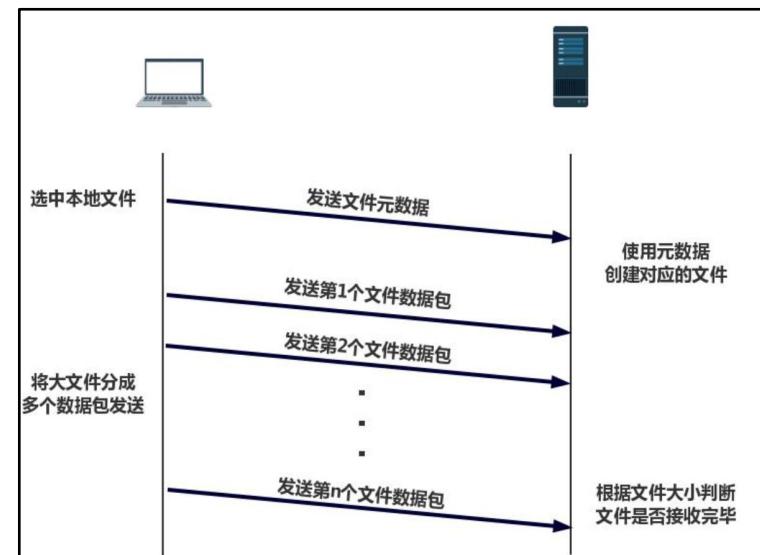
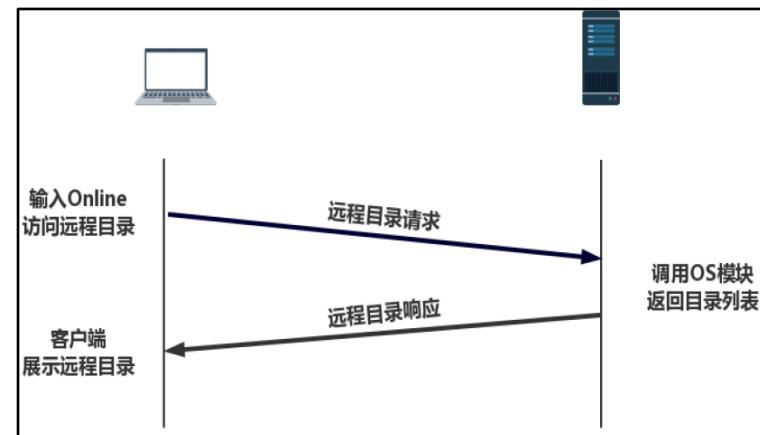
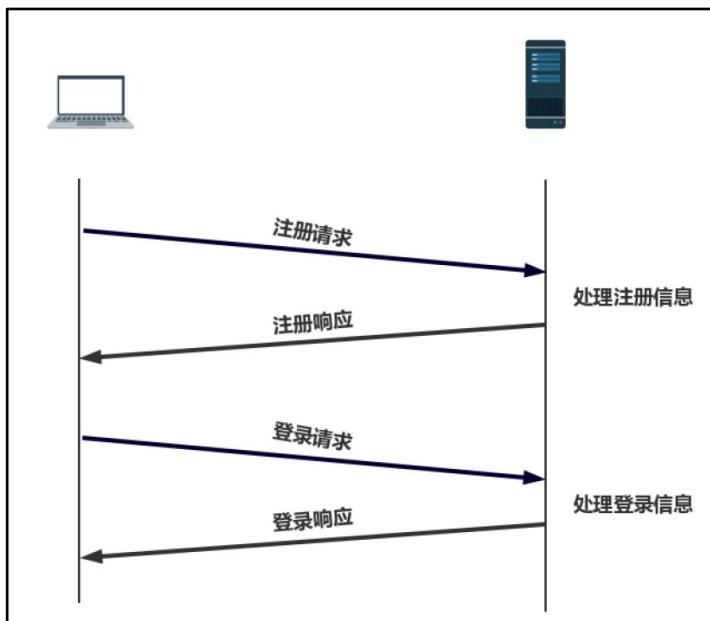
Prompt I

Exp II

ISTs II

Prompt II

Q&A



加分项

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 文件断点续传
- 基于公钥的身份验证
 - 双向证书验证亦可
- 支持同时上传、下载多个文件
 - GUI 支持
- 支持服务端根据用户身份，控制传输带宽
 - VIP

什么是协议

Contents

Exp I

ISTs I

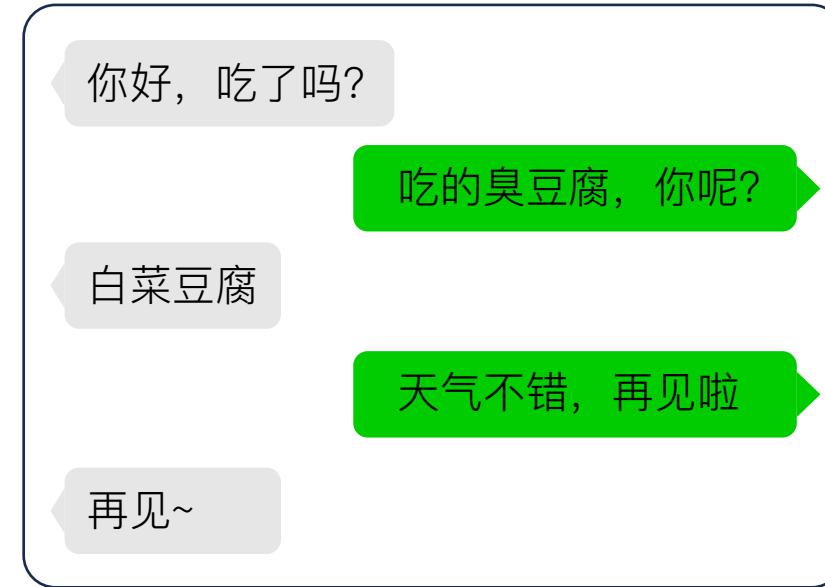
Prompt I

Exp II

ISTs II

Prompt II

Q&A



- 协议就是一组预定义的规范
- 协议实现：用报头表征控制，用载荷表征数据。

头部

载荷

从 TCP 报头谈协议是什么

Contents

Exp I

ISTS I

Prompt I

Exp II

ISTS II

Prompt II

Q&A

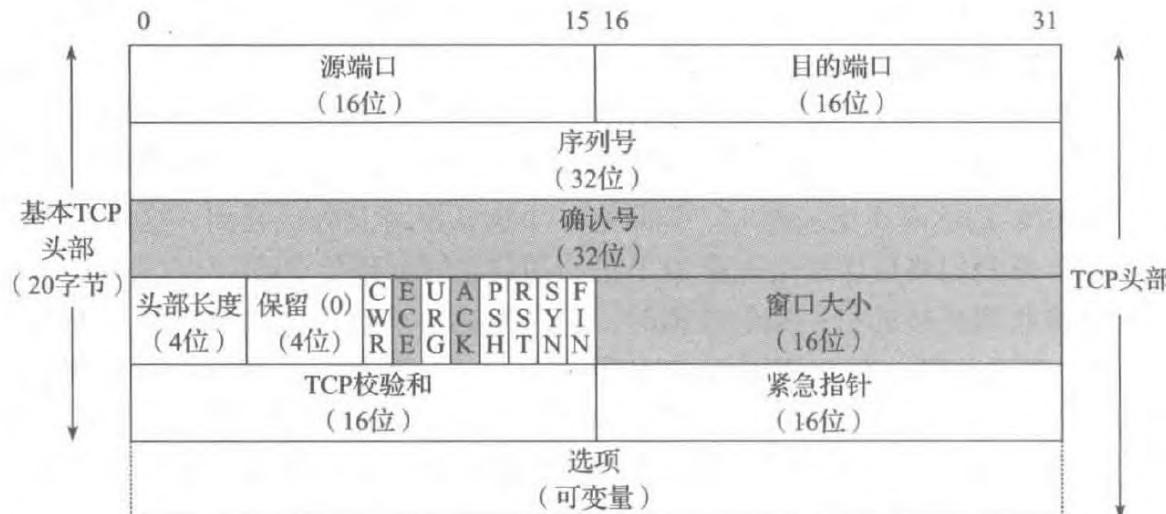


图 12-3 TCP 头部。它的标准长度是 20 字节，除非出现选项。头部长度 (Header Length) 字段以 32 位字为单位给出头部的大小 (最小值是 5)。带阴影的字段 (确认号 (Acknowledgment Number)、窗口大小 (Window Size) 以及 ECE 位和 ACK 位) 用于与该报文段的发送方关联的相反方向上的数据流

下知地理：TCP 可靠字节流

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- TCP 协议本质是带累积正向确认的滑动窗口协议

- TCP 虽然会给网络层提供 Segment，即报文段，但 TCP 本身是面向字节流的，并非是传输片段。
- TCP 会自动分片
- TCP 利用反馈、重传机制，在有损信道上实现可靠传输。

- 如何利用 TCP 传输一个固定大小的文件？

- 首先建立 TCP 连接：三次握手
- 发送方用 Socket 发送文件：以 4 KiB 为单位读取本地文件，在读了 N 个 4KiB 之后，终于读取到了 EOF 标志，则最后发送 $(FileSize - 4 * N)$ KiB 的数据
- 发送 FIN，关闭连接：四次挥手
- 接收方只需要傻傻接收，然后把数据包组合起来即可恢复出原文件。

如何建立完善的应用层协议

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 上述场景的局限性在于传输的终结由发送方控制
 - 在传输完成之前，只有发送方知道文件的具体大小。
 - 文件是否接收完是由发送方关闭连接所确定的。
- 如何克服上述弊端？
 - 发送方通过自定义应用层协议，提前告知接收方需准备多少磁盘空间（比如 N kiB）来存储要接收的数据。
 - 接收方根据 N 的值，从某一个包开始，截取包内容，并往本地的文件描述符写入数据，一旦接收到了 N kiB 的数据，就完成接受，并回复接收完成的确认。此后也不需要关闭连接。
 - 接收方回复已完成接收后，发送方关闭打开的文件，并对其文件描述符引用计数减一。

安全要点：Server 端 ACL 配置

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

• 自定义服务端业务代码

- 业务端进程需具备严格的访问控制。服务端只允许某个目录被 Client 读取、写入，以减少 Server 进程被黑客控制后产生进一步的损失
- ACL 主要针对以下项目进行权限控制
 - 用户 User
 - 用户组 Group
 - 默认属性 Mask
- 另有 setfacl 之外的方法可实现服务器端的访问控制。
- Server 端的可读写目录配置需以 shell 脚本或其他形式给出明确体现。
- 思考：能否使用 chroot 或直接容器化应用实现隔离？

提示： 协议的实现方法

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 仿照 TCP 报头的样式进行实现

- 划分字段，不同的值代表不同的行为
- 收到报头，按照 struct 进行解析，然后按照对应的逻辑完成对应的任务。
- 特别注意，报头将出现在每一个应用层 packet 里，要注意协议的传输开销。

- 用 JSON 编码/解析器完成

- 这种方法与 TCP/IP 的 bitmap + struct 报头方案在理论上是等价的，但是更加现代化、更灵活
- 优势：JSON 是服务器端和客户端都理解的描述语言，因此只要定义了双方的通信解析方法，就可以通过 JSON 格式的数据来传输控制报文，无需做字节级的包头解析。

仿 TCP 等报文头的设计

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

包类型名称	类型解释	包类型定义
REGISTER_REQUEST	注册请求	00
REGISTER_RESPONSE	注册响应	01
LOGIN_REQUEST	登陆请求	11
LOGIN_RESPONSE	登陆响应	12
CATALOG_REQUEST	远程文件目录请求	21
CATALOG_RESPONSE	远程文件目录响应	22
FILE_REQUEST	文件下载请求	31
FILE_METADATA	文件元数据通知	32
FILE_CONTENT	文件内容	33

PKI Internals

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

• PKI 技术：解决公钥密码算法中的信任问题

- 要使用公钥算法，必须明确公钥的归属问题
- Diffie and Hellman 首次公开公钥密码算法时便已意识到此问题，他们设想了一种 online 的可信资料库，但面临许多问题：
 - 频繁查询带来的性能开销问题
 - 离线使用问题
- 1978 年，Kohnfelder 提出数字证书 (Certificate) 的概念，由证书认证中心 (Certification Authority, CA) 签发证书来解决公钥归属问题。PKI 中的三个实体：
 - CA：负责为他人签发证书 (CA 的私钥至关重要！)
 - 证书持有者 (Certificate Holder)：有 CA 签发的证书和自己的对应私钥 (而非 CA 的私钥)
 - 依赖方 (Relying Party)：安装了 CA 的证书 (CA 的公钥)

CA 签名过程

Contents

Exp I

ISTs I

Prompt I

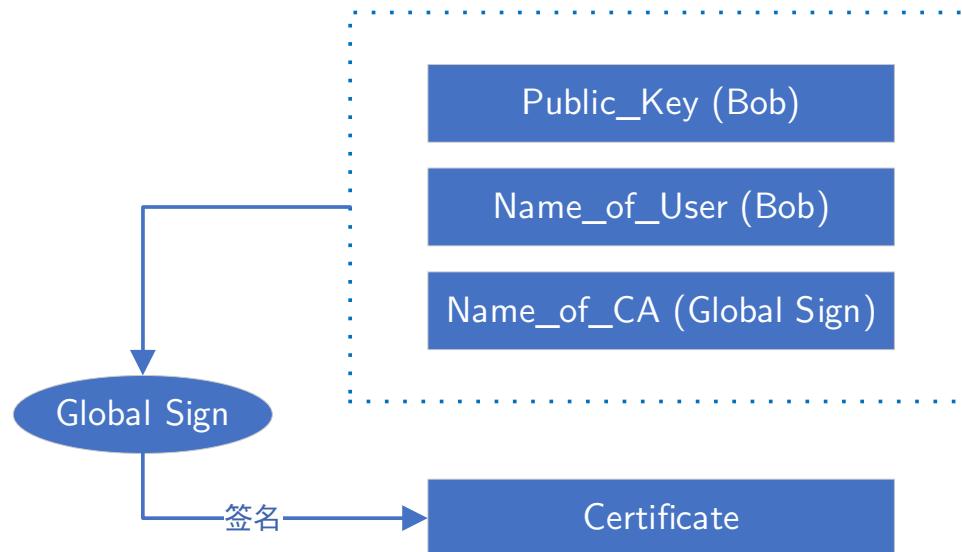
Exp II

ISTs II

Prompt II

Q&A

- 签名过程很严格、很繁琐，需要大量人工参与
 - CA 保证证书申请者与证书所标识的是同一个实体
 - 证书的身份信息也要进行仔细审查
 - 检查 Optional 的信息：电子邮件、IP 地址、DNS 域名



CA 给自己签发证书的过程

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

创建 CA 的信息

生成 CA 私钥

```
openssl genrsa -out ca-key.pem 4096
```

生成 CA 请求文件

```
openssl req -out ca-req.csr -key ca-key.pem -new
```

自签署 CA 的根证书

```
openssl x509
```

```
    -out ca-cert.cer
```

```
    -req -in ca-req.csr
```

```
    -signkey ca-key.pem
```

```
    -days 365
```

CA 给 Server 签发证书的过程

Contents

CA 签发服务器的证书

Exp I

生成服务器私钥

ISTs I

```
openssl genrsa -out server-key.pem 1024
```

Prompt I

生成服务器请求文件 (公钥 + 信息)

Exp II

```
openssl req -out server-req.csr -key server-key.pem -new
```

ISTs II

用 CA 的根证书、CA 的私钥签署服务器证书

Prompt II

```
openssl x509
```

```
    -out server-cert.cer
```

```
    -req -in server-req.csr
```

```
    -signkey server-key.pem
```

```
    -CA ca-cert.pem
```

```
    -CAkey ca-key.pem
```

```
    -CAcreateserial
```

```
    -days 365
```

Q&A

CA 给 Client 签发证书的过程

Contents

创建客户端证书

Exp I

生成客户端私钥

ISTs I

```
openssl genrsa -out client-key.pem 1024
```

Prompt I

生成客户端请求文件

Exp II

```
openssl req -out client-req.csr -key client/client-key.pem -new
```

ISTs II

用 CA 的根证书、CA 私钥签发客户端证书

Prompt II

```
openssl x509
```

```
    -out client-cert.cer
```

```
    -req -in client-req.csr
```

```
    -signkey client-key.pem
```

```
    -CA ca-cert.cer
```

```
    -CAkey ca-key.pem
```

```
    -CAcreateserial
```

```
    -days 365
```

Q&A

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

创建密钥



Contents

Exp I

ISTs I

Prompt I

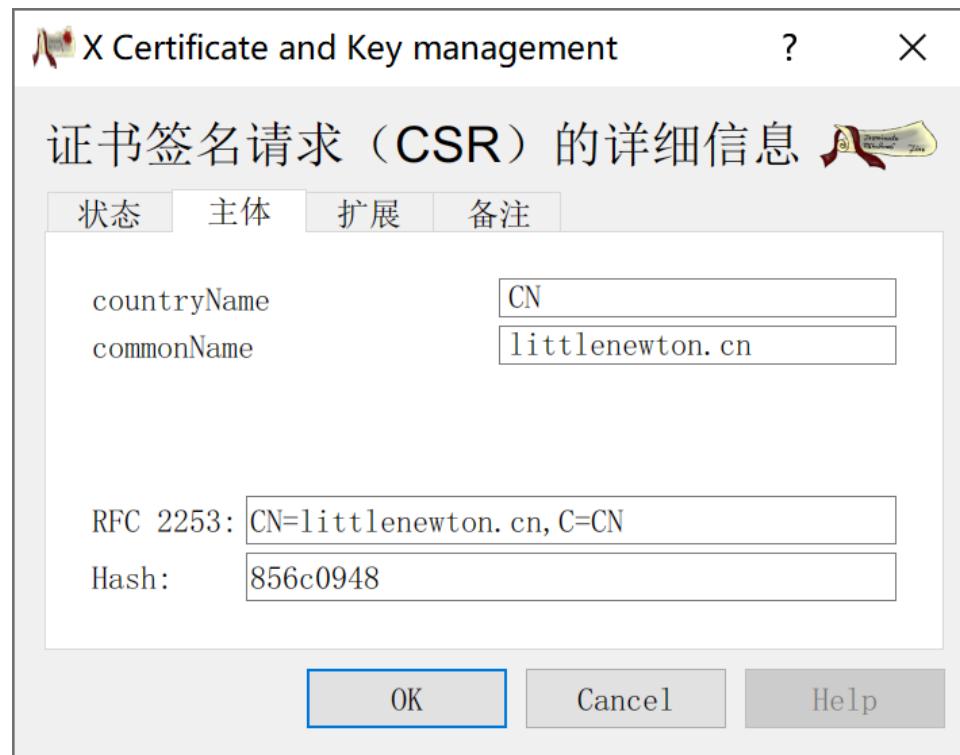
Exp II

ISTs II

Prompt II

Q&A

创建签名请求



Contents

Exp I

ISTs I

Prompt I

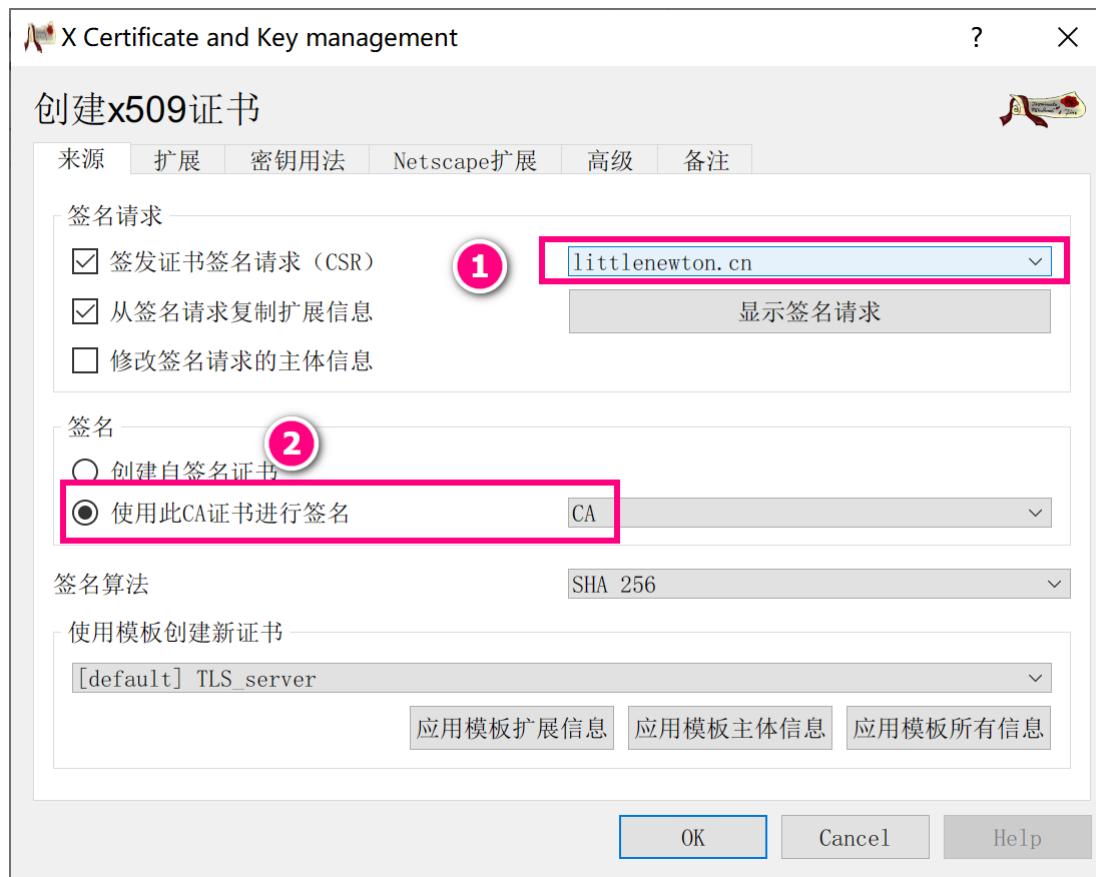
Exp II

ISTs II

Prompt II

Q&A

签发证书



Contents

Exp I

ISTs I

Prompt I

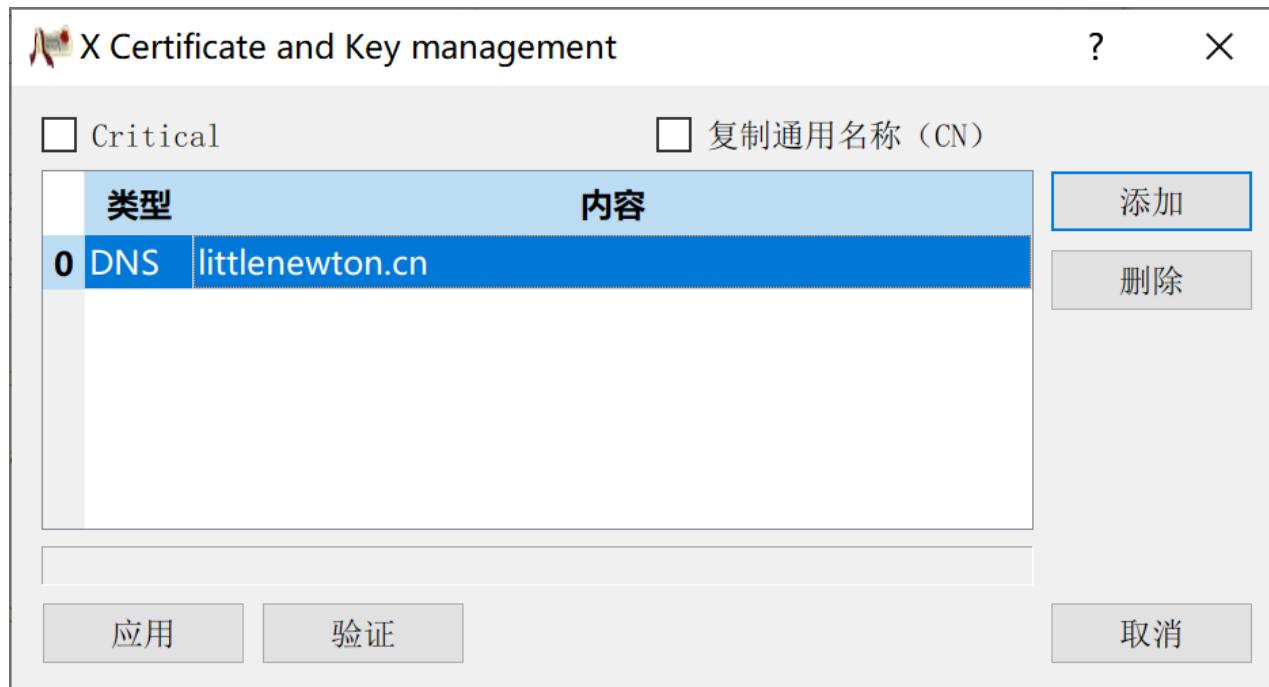
Exp II

ISTs II

Prompt II

Q&A

签发证书



Contents

Exp I

ISTs I

Prompt I

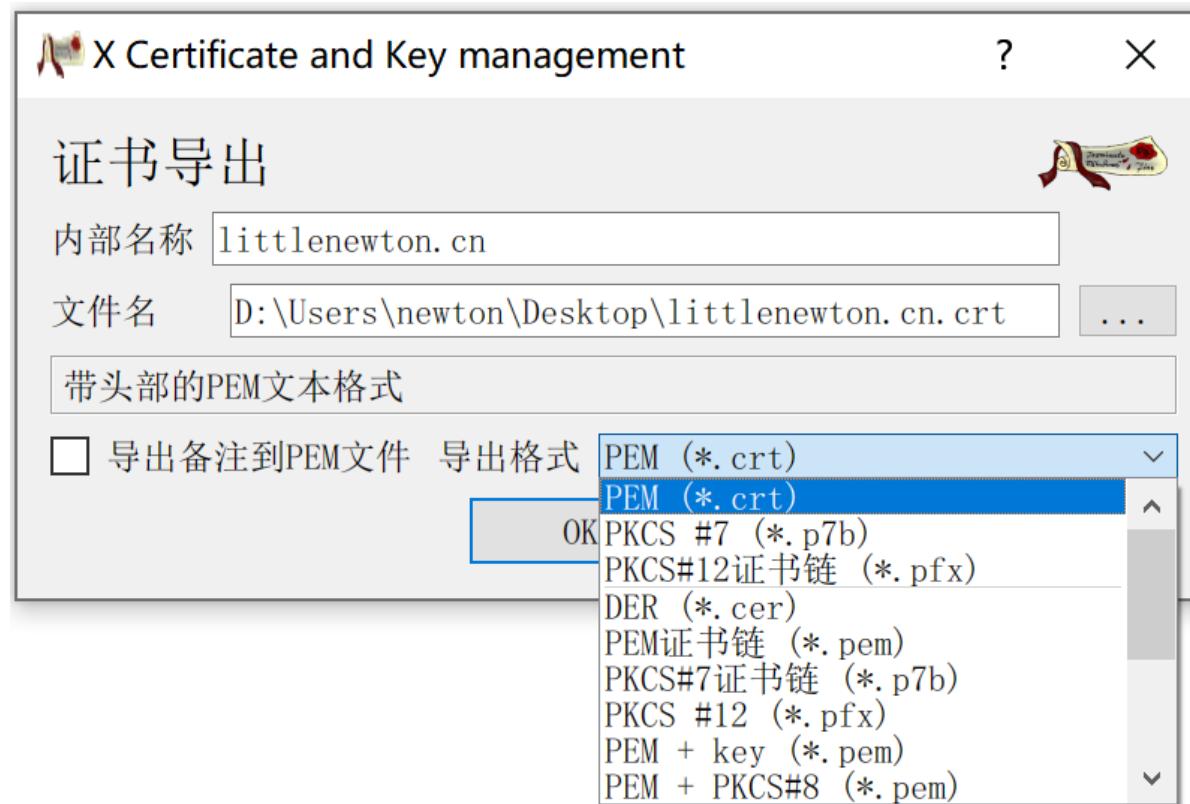
Exp II

ISTs II

Prompt II

Q&A

导出并使用证书



检查 HTTPS

Contents

Exp I

ISTs I

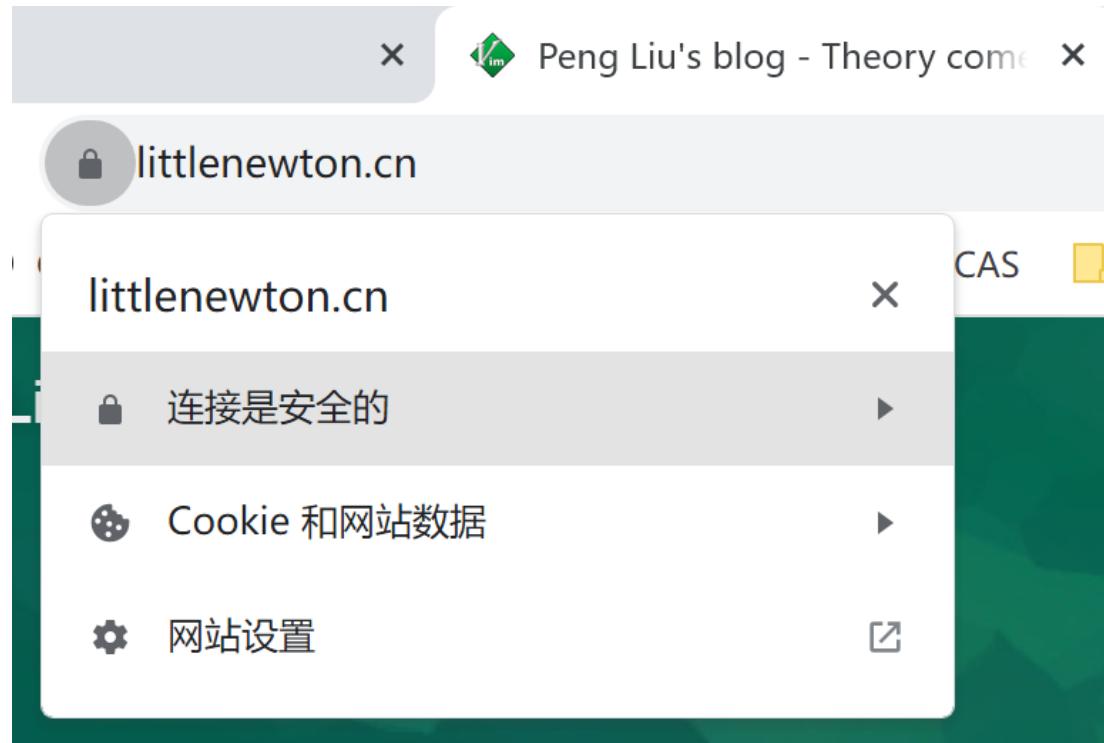
Prompt I

Exp II

ISTs II

Prompt II

Q&A



证书使用说明

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 证书使用者不限于计算机

- 进程、线程等运行时对象之间也可以用证书校验身份

- 以邮件程序为例：

- Alice 给 Bob 发送邮件时，在写好邮件之后，用自己的证书给邮件进行签名（证书持有者）

- Alice 准备发送时用 Bob 的证书给邮件加密（证书依赖者）

- 证书需要能撤销

- 证书持有者丢失了自己的私钥

- 需要重新申请新证书

- 废除旧的证书

- 绕过 CA 撤销别人的证书将直接导致服务不可用

Let's Encrypt 平台

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A



证书使用说明

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

SSL Certificates

Search Certificate...

NAME	CERTIFICATE PROVIDER	EXPIRES
 [REDACTED] Created: 27th January 2023	Let's Encrypt	27th April 2023, 6:03 am
 [REDACTED] cn Created: 27th January 2023	Let's Encrypt	27th April 2023, 6:44 am
 [REDACTED] cn Created: 27th January 2023	Let's Encrypt	27th April 2023, 6:03 am
 [REDACTED] .cn Created: 8th February 2023	Let's Encrypt	9th May 2023, 5:31 pm
 [REDACTED] .cn Created: 14th March 2023	Let's Encrypt	12th June 2023, 2:46 am
 [REDACTED] .cn Created: 8th February 2023	Let's Encrypt	9th May 2023, 5:32 pm
 [REDACTED] .cn Created: 8th February 2023	Let's Encrypt	9th May 2023, 5:33 pm

图. Let's Encrypt 证书列表

云时代 DevOps

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 在云时代，以 HTTPS 协议为核心
 - Google QUIC 重新定义传输层
 - DoT/DoT 重新定义 DNS
 - 以浏览器为主要客户端，Electron 等架构为辅
- 厂商以云计算中心为核心，保证高可用
 - 使用 RDMA 网络构建高性能集群
 - Google Cloud、Amazon AWS、Microsoft Azure
 - 华为云、腾讯云、阿里云
- 服务以容器形式部署、以镜像形式分发
 - 软件供应链安全

技术分享建议

Contents

Exp I

ISTs I

Prompt I

Exp II

ISTs II

Prompt II

Q&A

- 实验完成，将会邀请同学展示（择优、加分）
 - 先描述好一个问题。
 - 描述一个能让大家感同身受的编程、设计问题，不要一上来就讲你怎么做的
- How 比 What 重要
 - 要有不同技术的比较
 - 你是如何解决、排查问题所在的
- 一定要有 Best Practice 或方法论总结
 - 研究一个技术点，要总结出最佳实践

Thanks

- Thank you for your listening!
- contact information
 - Email: liupeng19@mails.ucas.ac.cn