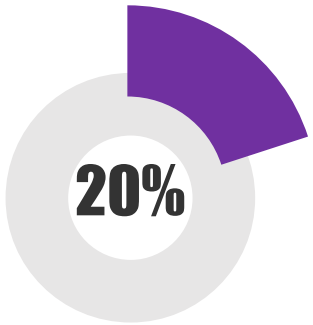
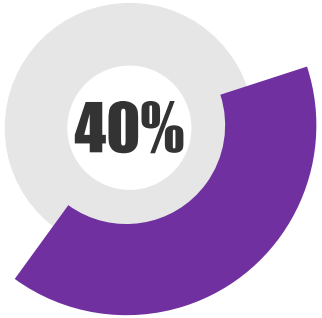
实验安排

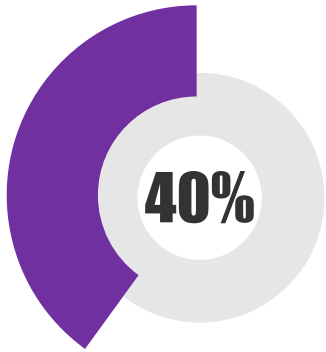
考核要求



作业



实验



期末考试（开卷）

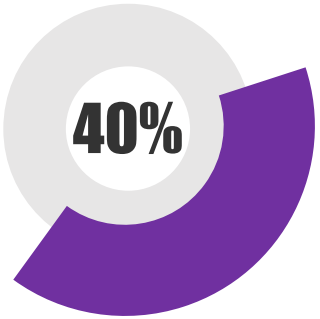
作业：实验：期末考试=1:2:2

平时作业：20分

三次实验：★5分，★★10分，★★★15分

期末考试：40分

实验选题要求



实验

1. **必须选择2个★★★ 实验 (30分)**
2. **另选1个★★ 实验或者2个★实验 (10分)**

实验对应章节	实验名称	难度
4. 数据加密	MD5散列碰撞试验	★
	基于口令的身份验证协议	★★★
5. 隐私保护	同态加密匿名投票	★
6. 硬件安全	微处理器安全漏洞: Spectre	★★★
7. 操作系统安全	模拟栈溢出攻击	★★
	Nginx栈溢出漏洞	★★★
8. 网络安全	基于IPID的TCP侧信道漏洞	★★★
	SYN洪范攻击	★
	基于可编程交换机的DDoS防御	★★
9. DNS安全	本地DNS缓存中毒	★★
10. 真实源地址	模拟域间源地址验证技术	★
11. 公钥基础设施	数字证书综合使用	★
12. 分布式系统安全	容错算法验证和模拟	★★
13. 应用安全	常见Web漏洞演示	★★★
14. 人工智能安全	神经网络后门攻击与防御	★★

第四章 MD5散列值碰撞

难度: ★☆☆

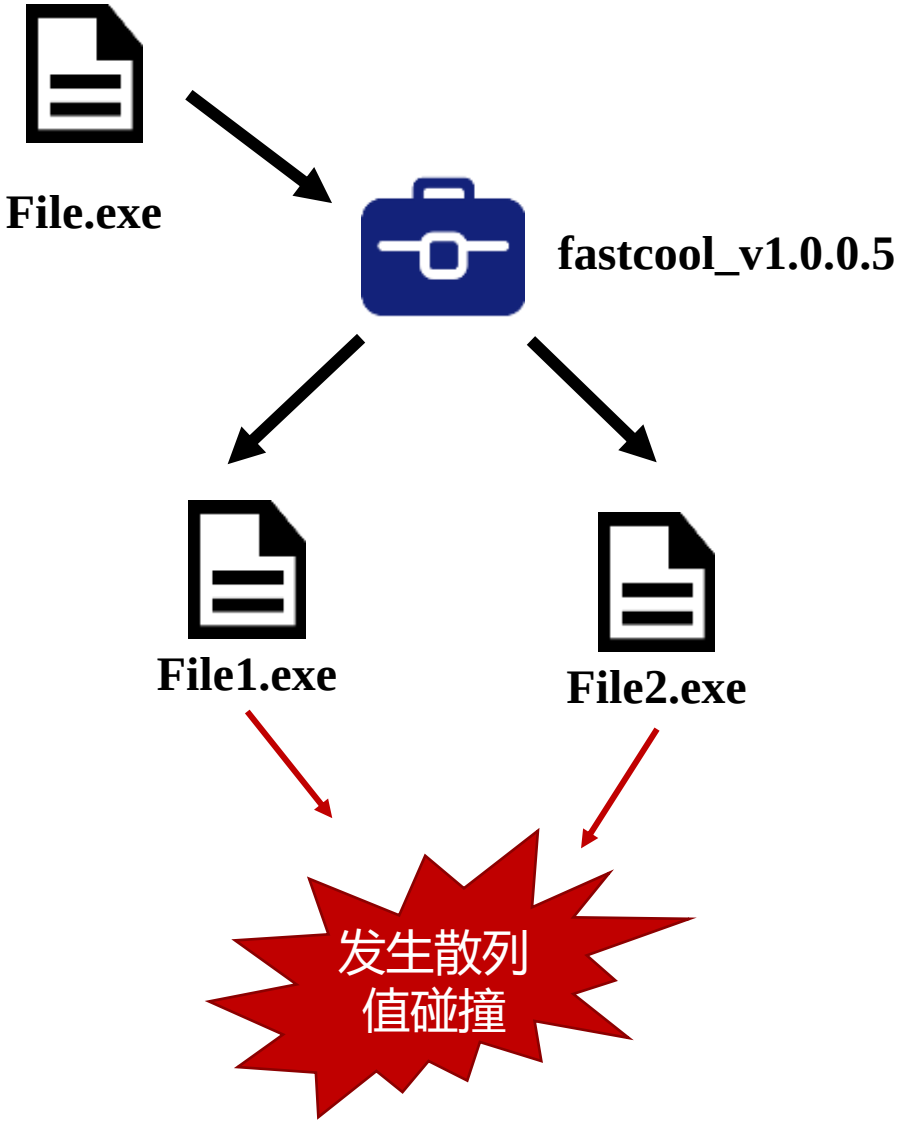
- **实验目的：**利用散列校验文件的完整性，并利用散列碰撞工具产生散列碰撞现象

- **实验要求：**

在Windows环境下调用散列碰撞生成工具，生成散列码相同的两个文件

- **实验资料：**

1. 教材 P147（配套工具下载地址见教材）
2. 网络学堂第四章课程讲义



第四章 基于口令的身份验证协议

难度: ★★★

- **实验背景:**

口令是最原始的登录方式，虽然手机验证码登录盛行，但口令验证协议仍被京东、淘宝等主流电商平台采用

Bellovin-Merritt 协议诞生于贝尔实验室，被发表于92年的IEEE S&P，是早期口令安全协议的代表

- **实验目的:**

通过完整实现该身份认证协议，体验实现密码学库的乐趣，加深对身份验证安全协议的理解

Encrypted Key Exchange:
Password-Based Protocols
Secure Against Dictionary Attacks

Steven M. Bellovin
AT&T Bell Laboratories
Murray Hill, NJ 07974
smb@ulysses.att.com

Michael Merritt
AT&T Bell Laboratories
Murray Hill, NJ 07974
mischu@research.att.com

Abstract

Classical cryptographic protocols based on user-chosen keys allow an attacker to mount password-guessing attacks. We introduce a novel combination of asymmetric (public-key) and symmetric (secret-key) cryptography that allow two parties sharing a common password to exchange confidential and authenticated information over an insecure network. These protocols are secure against active attacks, and have the property that the password is protected against off-line "dictionary" attacks. There are a number of other useful applications as well, including secure public telephones.

1 Introduction

People pick bad passwords, and either forget, write down, or reuse good ones. We present a protocol that affords a reasonable level of security, even if resources are protected by bad passwords. Using a novel combination of asymmetric (public-key) and symmetric (secret-key) cryptography — a secret key is used to encrypt a randomly-generated public key — two parties sharing a secret such as a password use it to exchange authenticated and secret information, such as a session key or a "ticket" for other services, a la Kerberos [1]. This protocol, known as encrypted key exchange, or *EKE*, protects the password from off-line "dictionary" attacks.

EKE can be used with a variety of asymmetric cryptosystems and public key distribution systems, subject to a few constraints detailed below. It works especially well with exponential key exchange [2]. Section 2 describes the asymmetric cryptosystem variant and implementations using RSA[3] and ElGamal[4]. Each of these two systems presents unique problems. Section 3 generalizes EKE, and shows how most public key distribution systems can be used. Section 4 considers general issues related to the choice and use of symmetric and asymmetric cryptosystems in EKE. Finally, in Section 5, we describe applications for EKE, and discuss related work in Section 6.

1.1 Notation

Our notation is shown in Table 1. To avoid confusion, we use the word "symmetric" to denote a conventional cryptosystem; it uses secret keys. A public-key, or asymmetric, cryptosystem has public encryption keys and private decryption keys.

1.2 Classical key negotiation

Suppose A (Alice) and B (Bob) share a secret, the password P. In order to establish a secure session key, A could generate a random key R, encrypt it with P as key and send the result, $P\{R\}$, to B. (This is essentially the mechanism used to obtain the initial ticket in the Kerberos authentication system [1].) Now A and B share R and can use it as a session key; perhaps B replies to A with

$$R\{\text{Terminal type}?\}$$

But an eavesdropper could record these messages, and run a dictionary attack against P by first decrypting $P\{R\}$ with candidate password P', and then using the resultant candidate session key

$$R' = P'^{-1}(P\{R\})$$

to decrypt $R\{\text{Terminal type}?\}$, examining the result for expected redundancy.

0-8186-2825-1/92 $3.00 © 1992 IEEE

第四章 基于口令的身份验证协议 难度: ★★★

- **实验要求:**

根据交互图完全实现该协议:

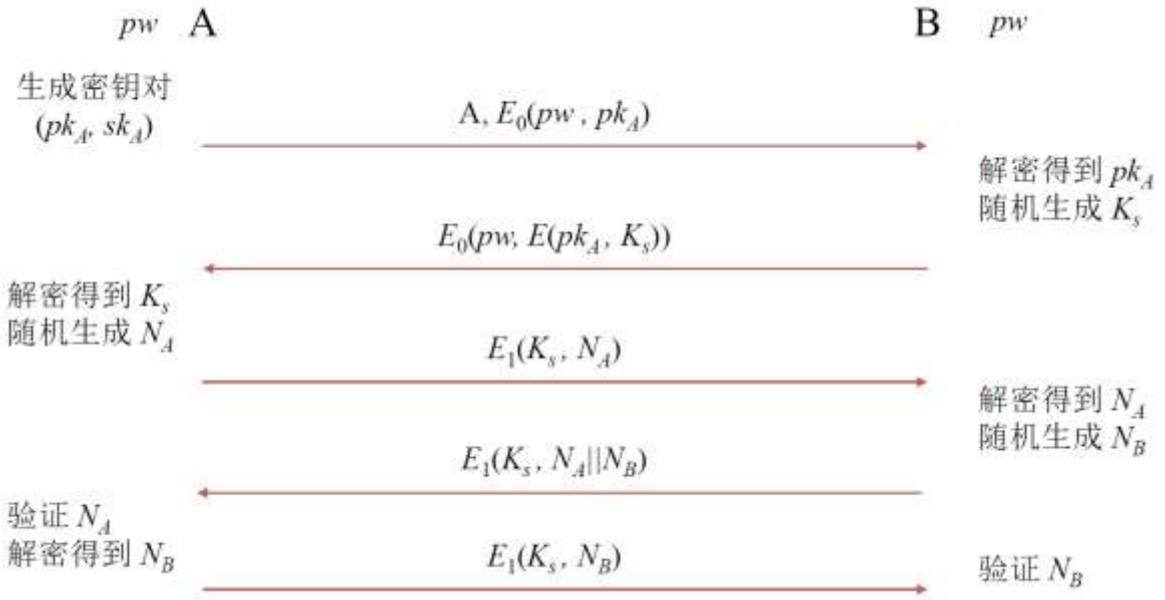
1. 通过TCP套接字进行传输， 用2个进程模拟交互流程，最建立安全信道，并加密传输信息
2. 不限定编程语言，具体对称/非对称加密方案自选，可以使用AES/RSA等基础密码学库中的实现

- **实验资料:**

1. 教材 P149
2. Bellovin-Merritt 协议原始文章:

https://ieeexplore.ieee.org/document/213269

Bellovin-Merritt 协议的交互流程图



第五章 同态加密匿名投票

难度: ★☆☆

- **实验背景:**

在电子投票系统中，由于统计票数是使用加法累加票数进行统计的，因此具有加法同态性质的 Paillier 算法可被应用于实现匿名的电子投票系统，以保护投票人的投票信息

- **实验目的:**

基于Paillier算法的匿名电子投票系统可实现匿名的投票本地模拟程序

- **实验要求:** 实现方式不限
- **实验资料:** 教材 P186

- Paillier算法具有加法同态性
- 密文相乘结果解密与明文相加结果相同

密钥生成

随机选取两个大素数p和q

计算$N = pq$和$\lambda = lcm(p-1, q-1)$

随机选取一个小于N^2的正整数g，并保证

$\gcd(L(g^\lambda \bmod N^2), N) = 1$，其中$L(x) = \frac{x-1}{N}$

公钥为(N, g)，私钥为λ

加密

已知明文$m < N$，随机选取$r < N$

密文$c = g^m r^N mod N^2$

解密

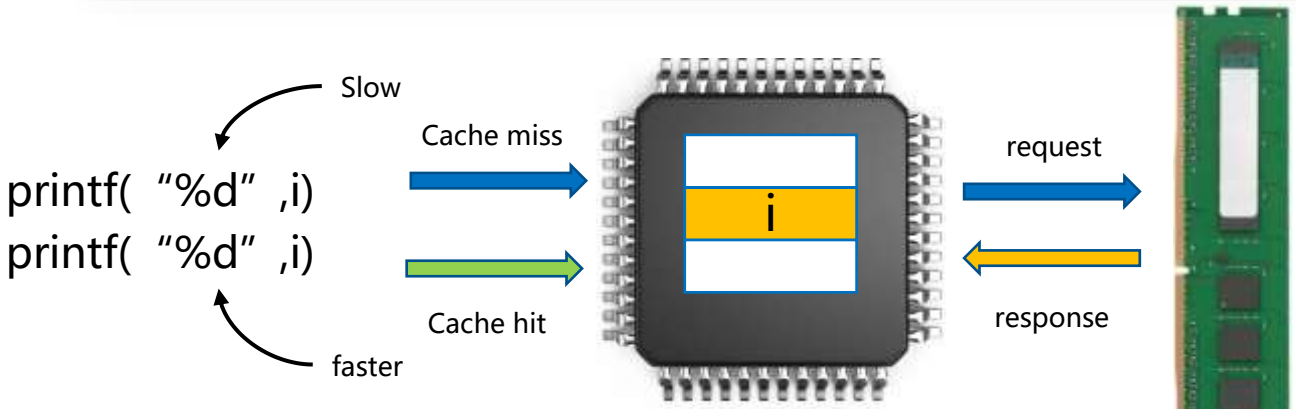
已知密文$c < N^2$，明文$m = \frac{L(c^\lambda mod N^2)}{L(g^\lambda mod N^2)} mod\ N$

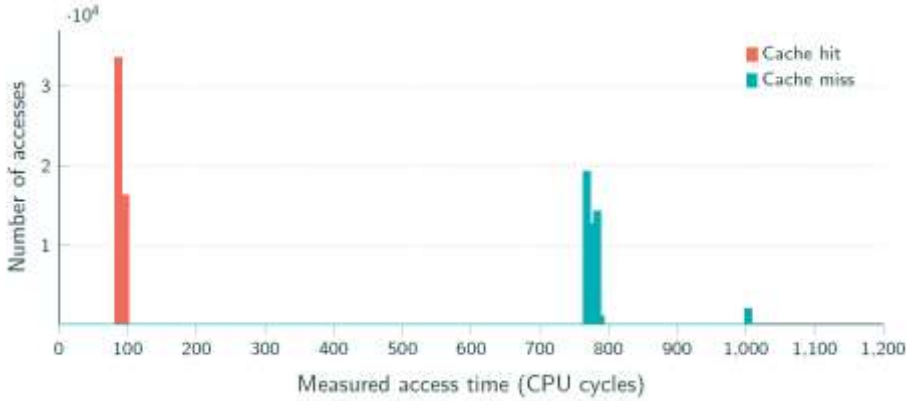
```
请输入第一个明文：123
请输入第二个明文：456
对第一个明文加密后得到密文：21074939849326247864279619
对第二个明文加密后得到密文：94145804565136665960095689
两密文相乘得到：19841171682766797052436847925323719
密文相乘后解密得到的明文为：579
```

第六章 微处理器安全漏洞 Spectre 难度: ★★★

- **实验背景：**微处理器架构侧信道严重损害内存的隔离性，泄露用户隐私信息

- **实验目的：**通过实现Spectre攻击样例，学习了解CPU性能优化技术（分支预测）带来的安全问题，进一步理解CPU的工作进程，加深对处理器硬件安全的认识





微处理器缓存侧信道

第六章 微处理器安全漏洞 Spectre 难度: ★★★

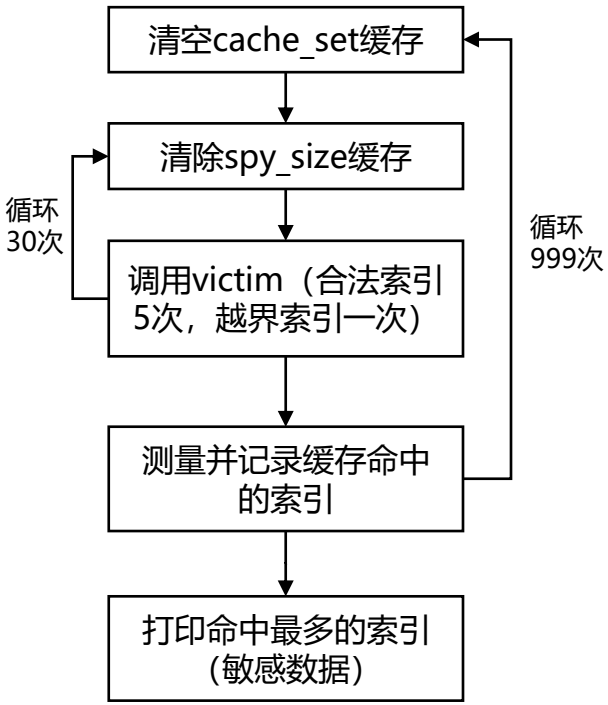
- **实验要求：**

1. 根据发现Spectre的文章附录当中的源代码复现文中 Section IV 中的实验，窃取内存当中的机密信息
2. 要求为C语言版本，可以参考其他源码但必须标明出处，且实验效果需要与教材一致

- 实验资料

1. 教材 P218
2. IEEE S&P上的Spectre原始文章：

https://ieeexplore.ieee.org/abstract/document/8835233



攻击流程

```
Reading 40 bytes:
0x54='T' score=999
0x68='h' score=999
0x65='e' score=999
0x20=' ' score=999
0x4D='M' score=999
0x61='a' score=999
0x67='g' score=999
0x69='i' score=999
0x63='c' score=999
0x20=' ' score=999
0x57='W' score=999
0x6F='o' score=999
0x72='r' score=999
0x64='d' score=998
0x73='s' score=999
0x20=' ' score=999
0x61='a' score=999
0x72='r' score=999
0x65='e' score=999
0x20=' ' score=998
0x53='S' score=999
0x71='q' score=999
0x75='u' score=999
```

实验结果

第七章 模拟栈溢出攻击

难度: ★★☆

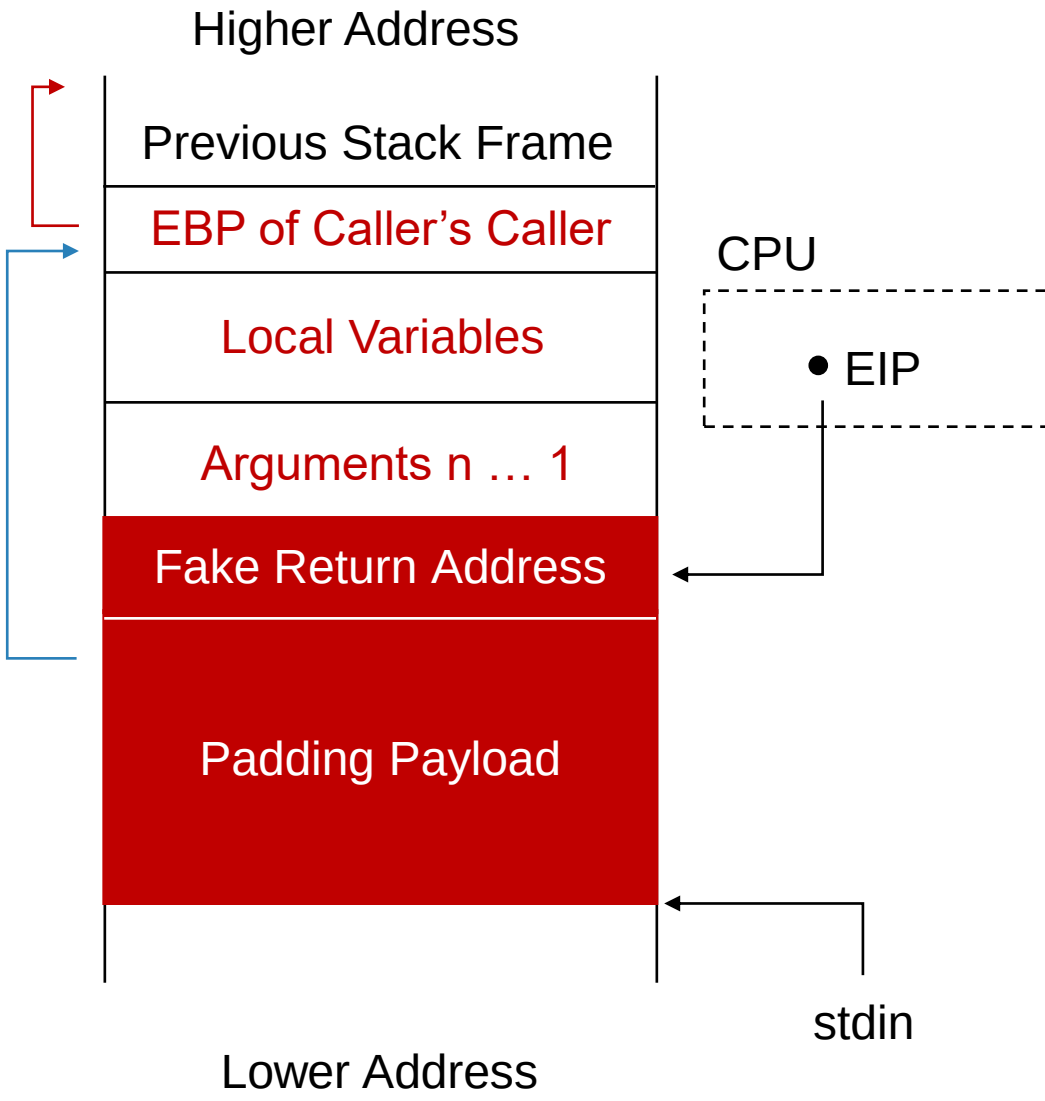
- **实验背景:**

栈区溢出攻击，是最常见的缓冲区溢出攻击方式，是多种攻击的基础

攻击者构造恶意的程序输入覆盖栈当中的返回地址，不正当触发函数执行，达到修改进程行为的目的

- **实验目的:**

本实验模拟一次朴素的栈区溢出攻击，核心在于掌握“如何构造覆盖栈帧的恶意输入”



第七章 模拟栈溢出攻击

难度: ★★☆

- **实验要求:**

实验需要完成如下步骤:

1. 给出一个C语言受害程序，包含一个要被恶意触发的目标函数，和不安全的输入语句
2. 关闭内存防御方案（ASLR、Stack Canary等）的条件下编译受害程序
3. 利用GDB观察受害函数的地址，并计算覆盖栈帧返回地址需要的“偏移量”
4. 编写恶意程序，构造非法输入
5. 执行恶意程序，恶意执行目标函数

- **实验资料:**

1. 教材 P254

第七章 Nginx栈溢出漏洞

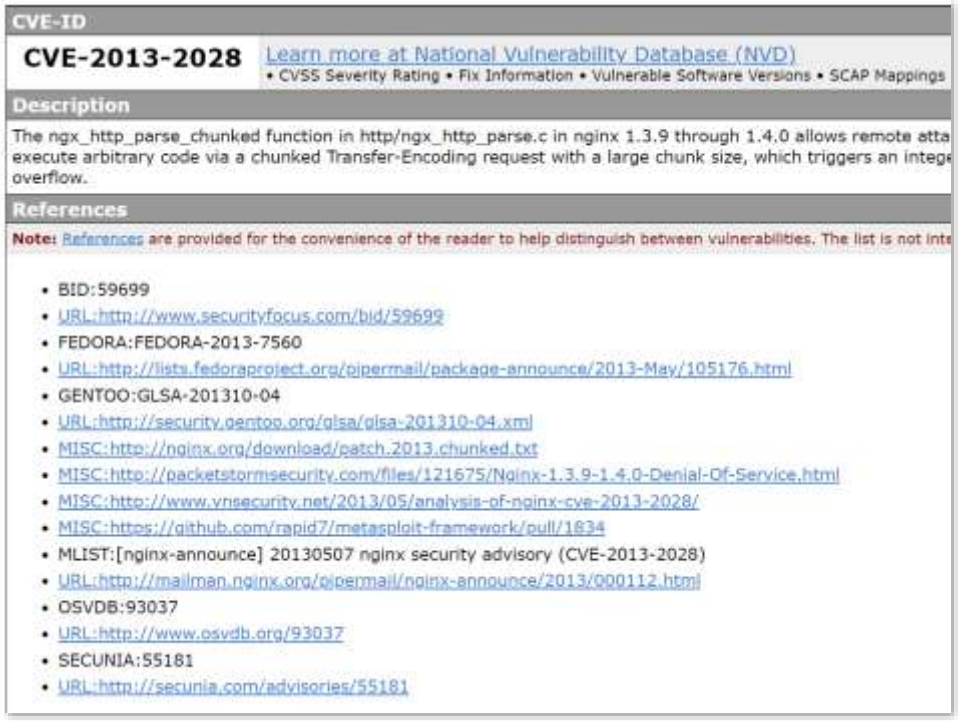
难度: ★★★

- **实验背景:**

上一个实验是模拟环境下的栈溢出漏洞，下面尝试复现真实世界当中的Nginx的栈溢出漏洞：CVE-2013-2028

- **实验目的:**

掌握在现实世界中可行的漏洞利用技巧：BROP，理解BROP如何绕过Canary、ASLR等防御机制，体验控制工业产品级别服务器的快乐



CVE-ID

CVE-2013-2028 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings

Description

The ngx_http_parse_chunked function in http/ngx_http_parse.c in nginx 1.3.9 through 1.4.0 allows remote atta execute arbitrary code via a chunked Transfer-Encoding request with a large chunk size, which triggers an intege overflow.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not inte

- BID:59699
- URL:http://www.securityfocus.com/bid/59699
- FEDORA:FEDORA-2013-7560
- URL:http://lists.fedoraproject.org/pipermail/package-announce/2013-May/105176.html
- GENTOO:GLSA-201310-04
- URL:http://security.gentoo.org/glsa/glsa-201310-04.xml
- MISC:http://nginx.org/download/patch.2013.chunked.txt
- MISC:http://packetstormsecurity.com/files/121675/Nginx-1.3.9-1.4.0-Denial-Of-Service.html
- MISC:http://www.vnsecurity.net/2013/05/analysis-of-nginx-cve-2013-2028/
- MISC:https://github.com/rapid7/metasploit-framework/pull/1834
- MLIST:[nginx-announce] 20130507 nginx security advisory (CVE-2013-2028)
- URL:http://mailman.nginx.org/pipermail/nginx-announce/2013/000112.html
- OSVDB:93037
- URL:http://www.osvdb.org/93037
- SECUNIA:55181
- URL:http://secunia.com/advisories/55181

CVE-2013-2028 检索结果

第七章 Nginx栈溢出漏洞

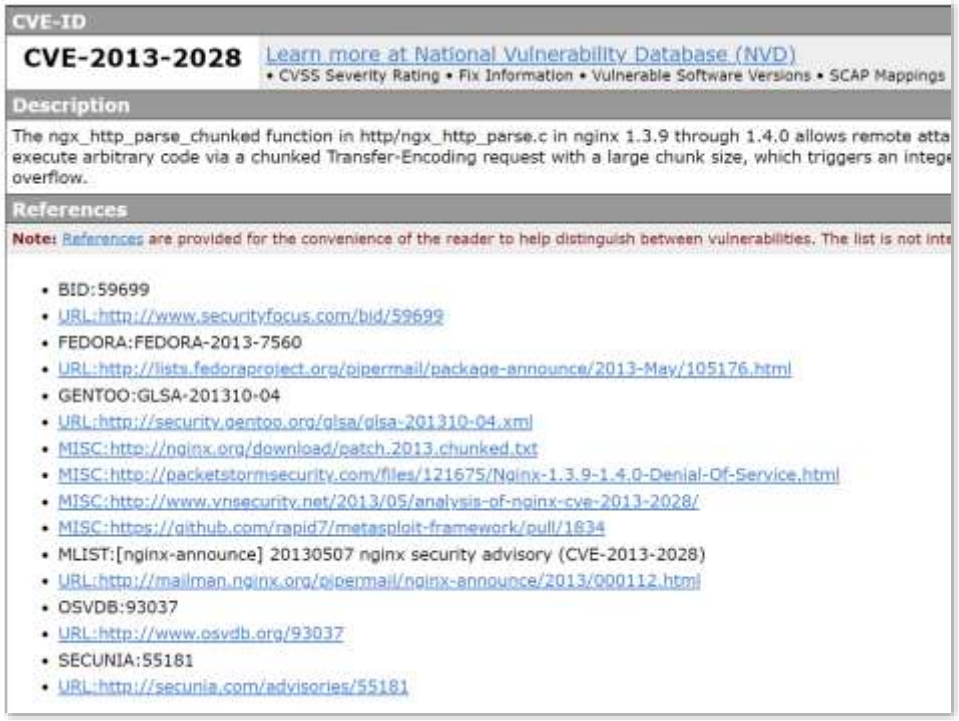
难度: ★★★

- **实验要求：**

1. 去CVE中检索该漏洞相关的信息
2. 部署存在漏洞的Nginx版本
3. 编写漏洞利用源代码，可使用现有工具库，亦可参考现有源代码，但需要在实验报告中标明出处，禁止抄袭源代码
4. 连接Shell，完全控制Nginx服务器，并尝试访问受害服务其文件系统

- **实验资料：**

CVE：*https://cve.mitre.org/*



CVE-ID

CVE-2013-2028 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings

Description

The ngx_http_parse_chunked function in http/ngx_http_parse.c in nginx 1.3.9 through 1.4.0 allows remote atta execute arbitrary code via a chunked Transfer-Encoding request with a large chunk size, which triggers an intege overflow.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not inte

- BID:59699
- URL:http://www.securityfocus.com/bid/59699
- FEDORA:FEDORA-2013-7560
- URL:http://lists.fedoraproject.org/pipermail/package-announce/2013-May/105176.html
- GENTOO:GLSA-201310-04
- URL:http://security.gentoo.org/glsa/glsa-201310-04.xml
- MISC:http://nginx.org/download/patch.2013.chunked.txt
- MISC:http://packetstormsecurity.com/files/121675/Nginx-1.3.9-1.4.0-Denial-Of-Service.html
- MISC:http://www.vnsecurity.net/2013/05/analysis-of-nginx-cve-2013-2028/
- MISC:https://github.com/rapid7/metasploit-framework/pull/1834
- MLIST:[nginx-announce] 20130507 nginx security advisory (CVE-2013-2028)
- URL:http://mailman.nginx.org/pipermail/nginx-announce/2013/000112.html
- OSVDB:93037
- URL:http://www.osvdb.org/93037
- SECUNIA:55181
- URL:http://secunia.com/advisories/55181

CVE-2013-2028 检索结果

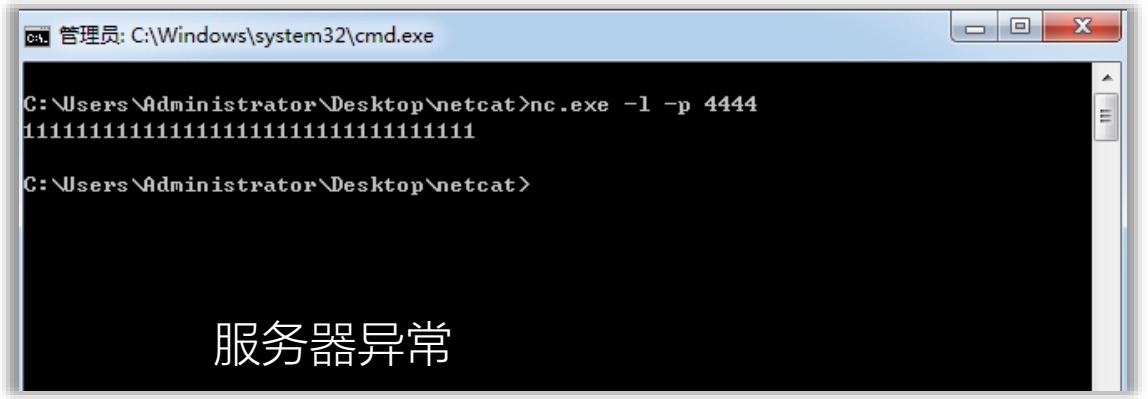
第八章 基于IPID的TCP侧信道漏洞 难度: ★★★

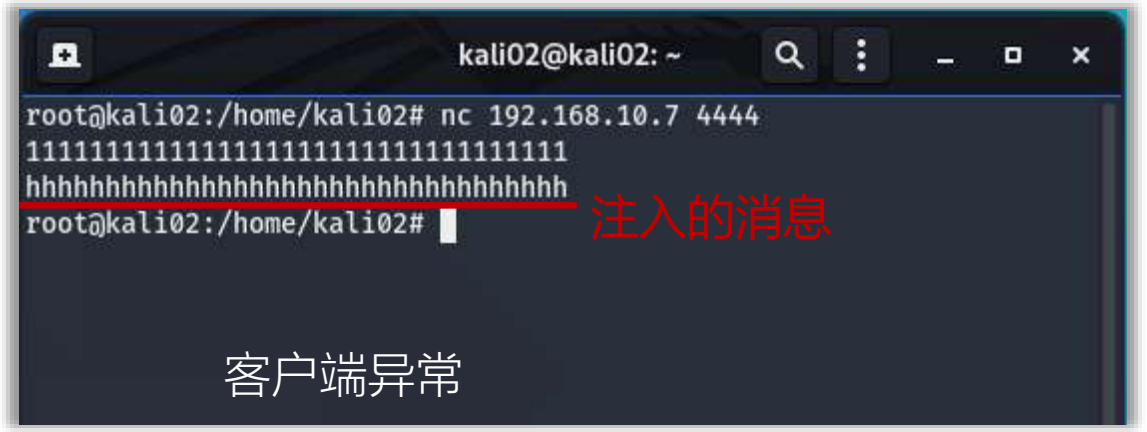
- **实验背景:**

网络侧信道攻击借助通讯内容无关信息干涉通讯的正常进行，可实现链接的强制中断或虚假信息注入

- **实验目的:**

复现基于全局IPID计数器的TCP连接注入攻击，了解网络协议栈常见安全威胁及漏洞，加深对网络安全问题的认识





第八章 基于IPID的TCP侧信道漏洞 难度: ★★★

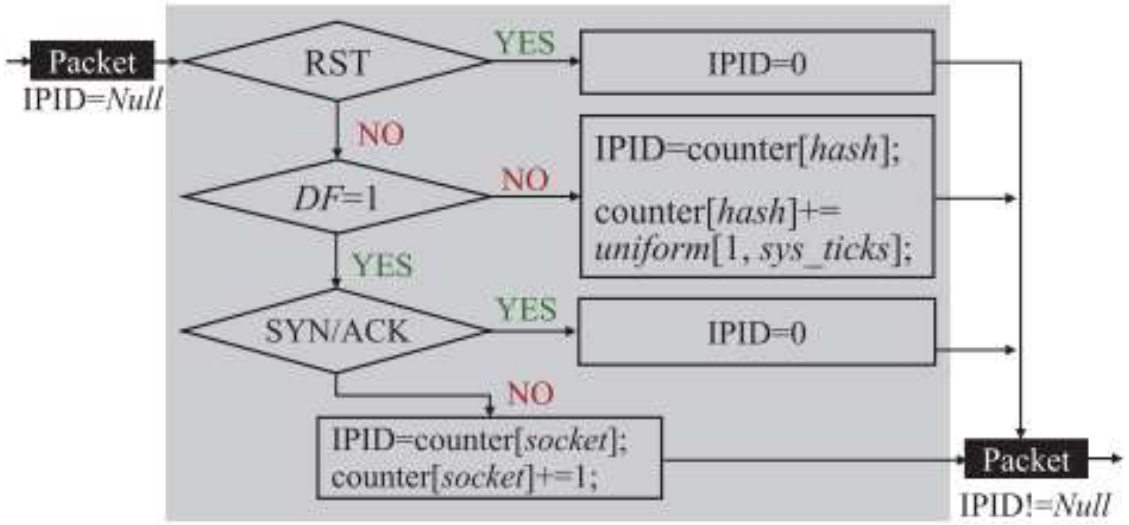


Fig. 1. IPID assignment in Linux version 4.18 and beyond.

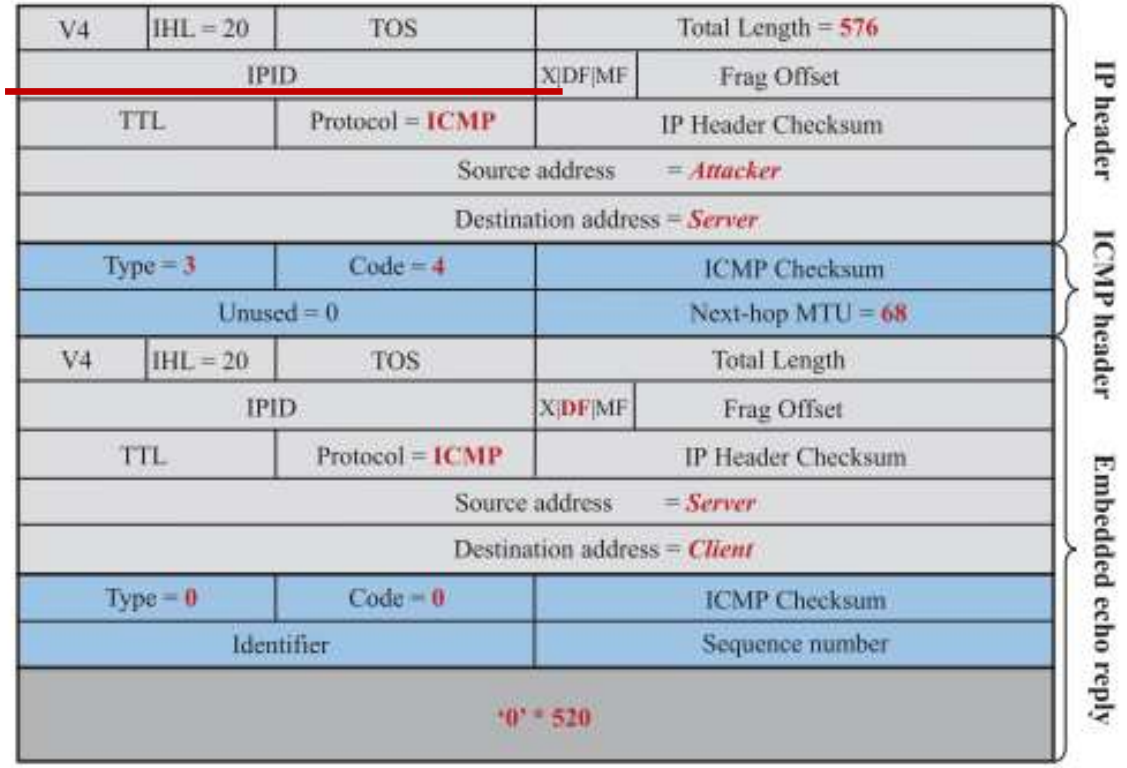
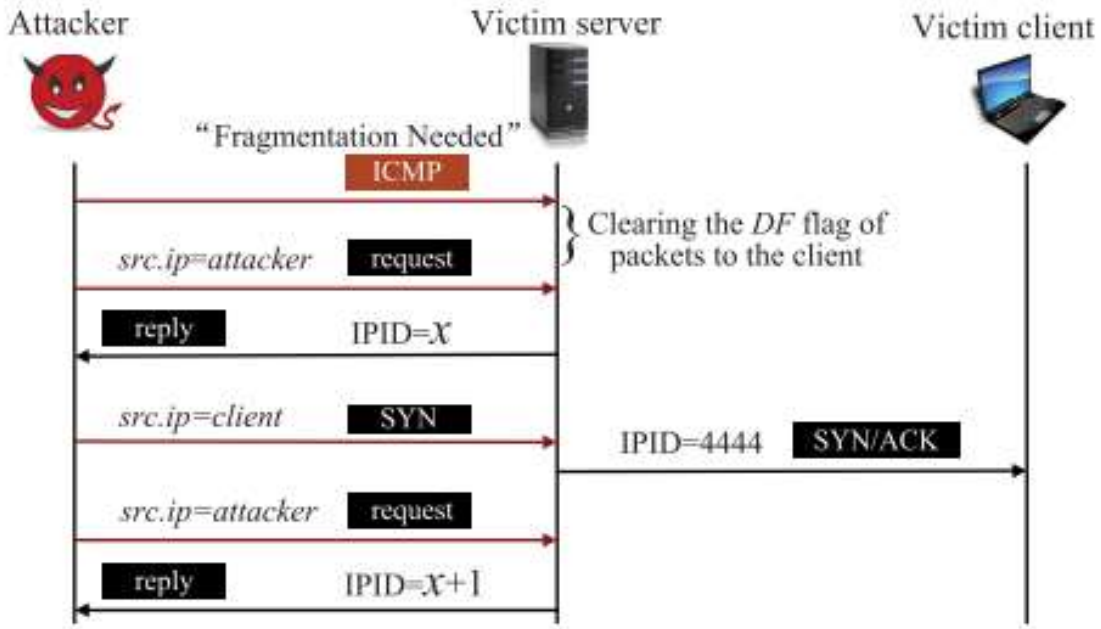
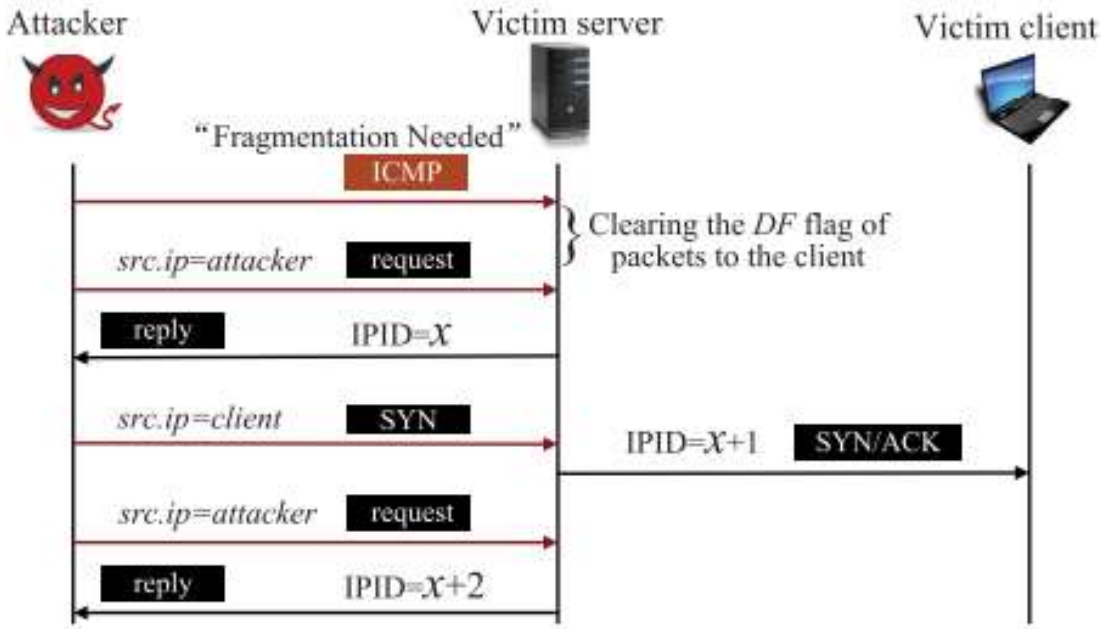


Fig. 3. Structure of the forged ICMP error message.

第八章 基于IPID的TCP侧信道漏洞 难度: ★★★



(a) No hash collisions with the client



(b) Hash collisions with the client

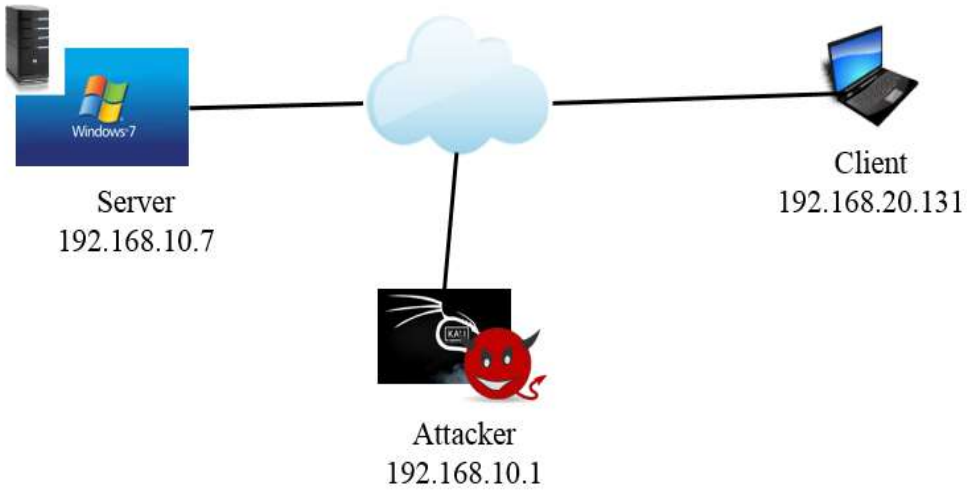
第八章 基于IPID的TCP侧信道漏洞 难度: ★★★

- **实验要求：**

1. 根据实验环境设置通讯环境，配置正确的内核版本，建立客户端和服务器之间的受害链接
2. 利用**IPv6地址**发起基于IPID的TCP侧信道攻击
3. 向TCP通讯信道当中注入恶意信息，可以在客户端注入或服务器端，**注入前链接不可中断**

- **实验参考：**

1. 教材 P288
2. IPID侧信道漏洞原始文章：
   https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9556515
3. 参考源代码：
   https://github.com/fuchuanpu/TCP-exploit



实验环境设置

如果你觉得这项工作还不错，请留给我们一个 Star☆

第八章 SYN洪范攻击

难度: ★☆☆

- **实验背景:**

DDoS攻击天天都在发生，简单洪范式DDoS攻击的代表是SYN洪范攻击

- **实验目的:**

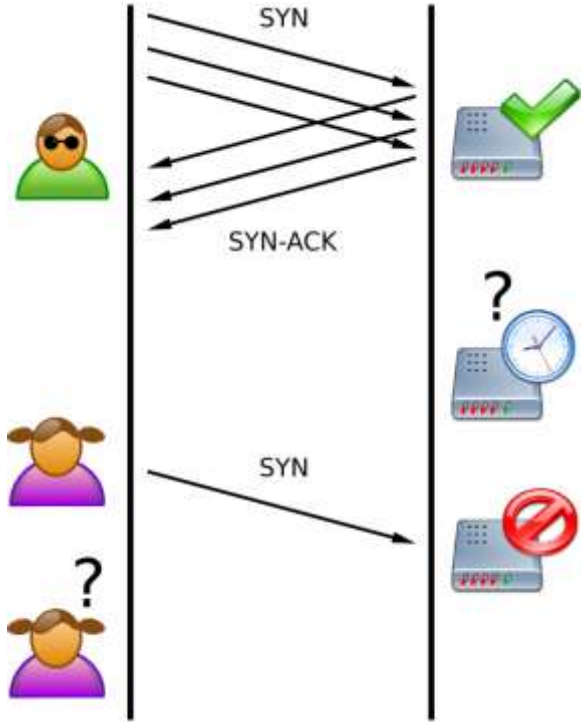
1. 部署一台受害Web服务器，测量一次访问时延

2. 利用**原始套接字**编写程序洪范TCP SYN，再次测量并记录访问延迟

本实验千万要注意安全

- **实验资料:**

教材 P287



SYN 洪范攻击 示意图

第八章 基于可编程交换机的DDoS防御 难度: ★★☆

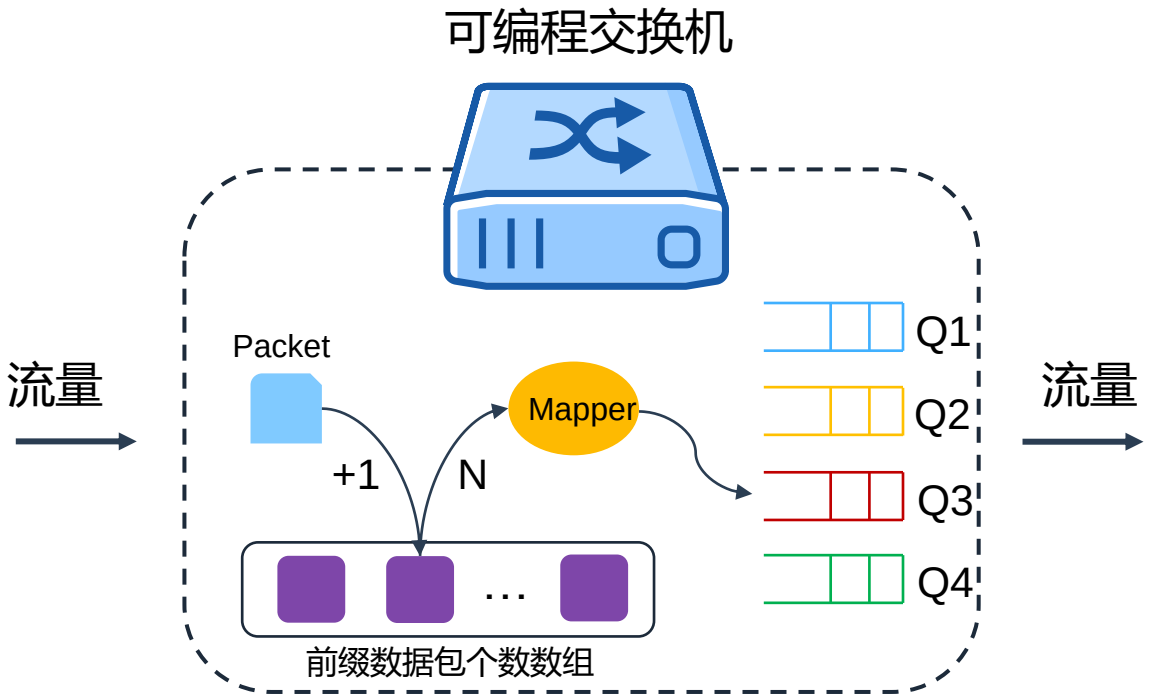
- **实验背景:**

DDoS攻击是目前网络空间中最广泛的攻击方式，其通过不同位置的多个攻击者同时向一个或数个目标发动攻击，使目标主机不胜负荷以至于瘫痪而停止提供正常的网络服务。

- **实验目的:**

在可编程交换机上实现DDoS流量简易防御，主要思想是统计不同前缀到达的数据包个数，根据数据包所属前缀已到达的数据包个数决定其出队优先级。最终实现到达数据包越多的前缀出队优先级越低，避免由于DDoS流量导致正常流量丢包。

通过本次实验了解可编程交换机技术，进一步认识DDoS攻击和防御。



N: 对应前缀到达的数据包个数

Q: 不同优先级队列

第八章 基于可编程交换机的DDoS防御 难度: ★★☆

- **实验要求：**

实验需要完成如下步骤：

1. 数据包进入后得到其源地址前缀（前16位）
2. 根据前缀找到对应寄存器位置
3. 更新对应寄存器数值并读出该前缀已到达数据包个数N
4. 根据N查询流表得到对应队列（流表中实现N越大，对应出队列优先级越低）
5. 设定数据包出队列为对应队列

要求：实验中将N和对应队列值嵌入数据包头中，对出流量包头嵌入值分析验证功能正确性

- **实验参考：**

1. ACC-Turbo 实现

第九章 本地DNS服务器缓存中毒 难度: ★★☆

- **实验背景：**

DNS是多种服务的基础，当DNS发生安全故障时，全部依赖DNS的服务都将被影响，最终危害大量使用者的安全

- **实验目的：**

通过实验，掌握DNS缓存中毒实施过程及原理，从而加深对网络安全问题的认识，并了解相应的安全防御手段

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1367
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.org.                IN      A

;; ANSWER SECTION:
www.example.org.        172800  IN      A       108.109.10.66

;; Query time: 19 msec
;; SERVER: 10.0.10.5#53(10.0.10.5)
;; WHEN: Fri Jan 22 12:27:52 EST 2021
```

攻击成功后，用户查询域名时，得到“攻击者”设定的伪造地址

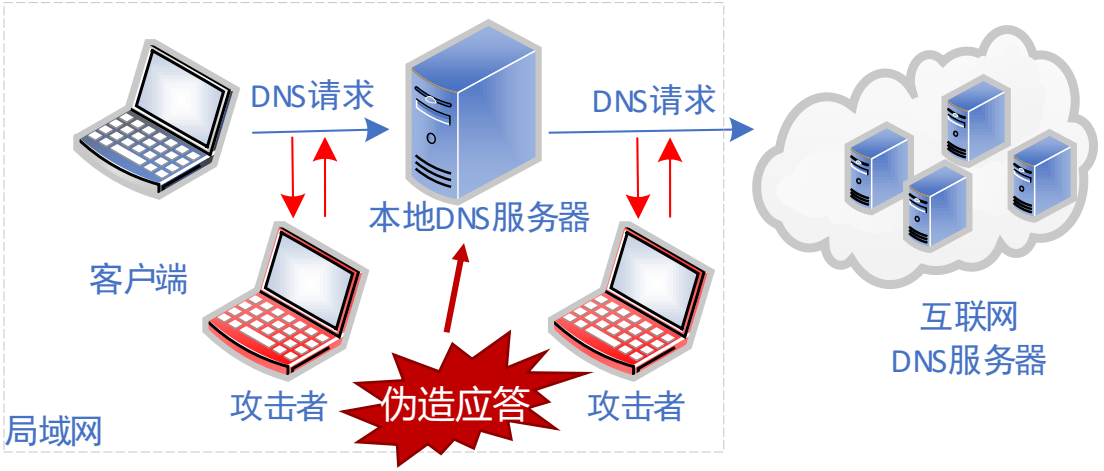
第九章 本地DNS服务器缓存中毒 难度: ★★☆

- **实验要求:**

1. 配置bind9本地DNS缓存服务器
2. 受害者访问某一域名，缓存服务器产生缓存项
3. 编写并运行程序监听DNS请求，并在监听到请求后注入恶意DNS响应
4. 客户端再次进行查询，得到错误DNS解析

- **实验参考:**

教材 P327



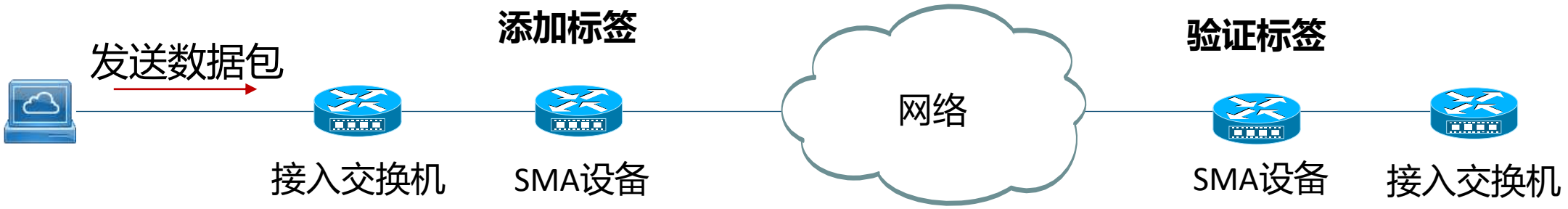
第十章 域间源地址验证协议

难度: ★☆☆

- **实验背景:**

域间源地址验证技术SMA中，AS之间通过建立状态机，并在AS的边界路由器进行标签的添加、验证和移除，实现自治域间的源地址验证

发送方边界核心路由器在分组添加标签，以传递源地址前缀的真实性。目的域在边界路由器检查标签正确性，以验证所关联源地址的真实性，过滤伪造分组



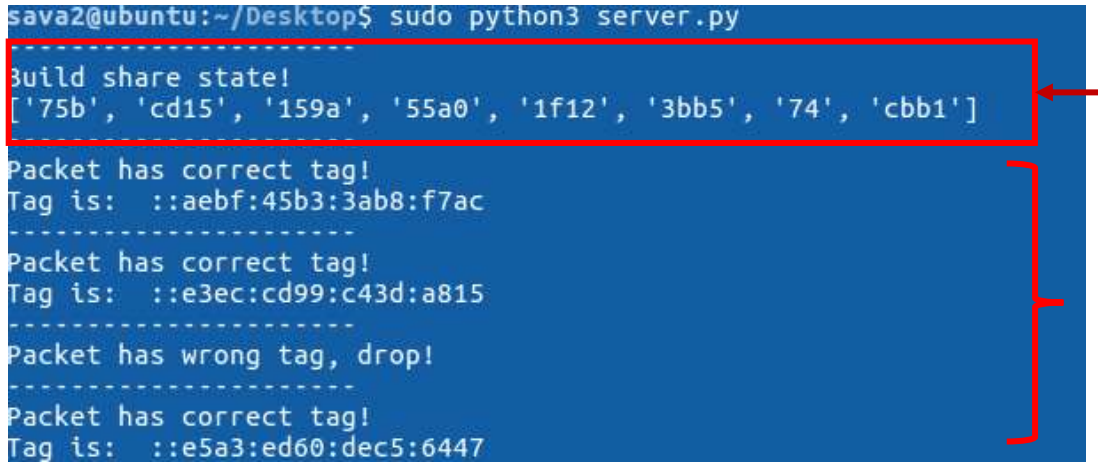
第十章 域间源地址验证协议

难度: ★☆☆

- **实验要求与参考资料：**

实验环境配置两台计算机，两台计算机分别充当发送方边界路由器以及目的域边界路由器，进行状态机建立、添加标签、验证标签三项功能的模拟。

可将两台计算机作为一个虚拟机运行在同一物理计算机中，详细资料见教材 P360



协商建立状态机

检验标签，同步状态机

第十一章 数字证书综合使用

难度: ★☆☆

- **实验目的：**

掌握PKI的原理和证书使用，加深对网络安全问题的认识，了解如何增加通信安全性

- **实验要求：**

1. 通过OpenSSH工具生成公私钥对，并能将公钥传到自己的远程服务器，使用密钥对通过ssh的方式安全访问远程服务器

2. 编写一个属于自己的简单网站，并通过nginx或者apache添加https协议，如果有域名可以通过阿里云申请对应的证书

- **实验资料：** 教材 P394

```
[➜  key01 ls
id_rsa.server      id_rsa.server.pub
[➜  key01 cat id_rsa.server.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCt8bSpltaViz+haggLZi2PNAm
Fwi4cwfm+eG3auut52OYmSMF3RbB5GzBeEOaevohBP7qWX5ohCAB5JBI8fdGvIJ
iiS+vUBat4LzgwMSgYV/RfJVRP1wz4eQQxf4QWOaBfajPwVDB2nnd5Zb4NrYkE3
T7IyEdOtGqAUrncxHG/dmeIKOXtuJLulaFJGSOOwEevHZkD8JJv2lzcb0oI8fd0
xIzP1EWxtML/Aq1Tk0C02A1VCZELHFBx+puSvrkVXqbuxekF7KRGO5hwjkPP ha
.com
```

第十二章 容错算法的模拟与验证 难度: ★★☆

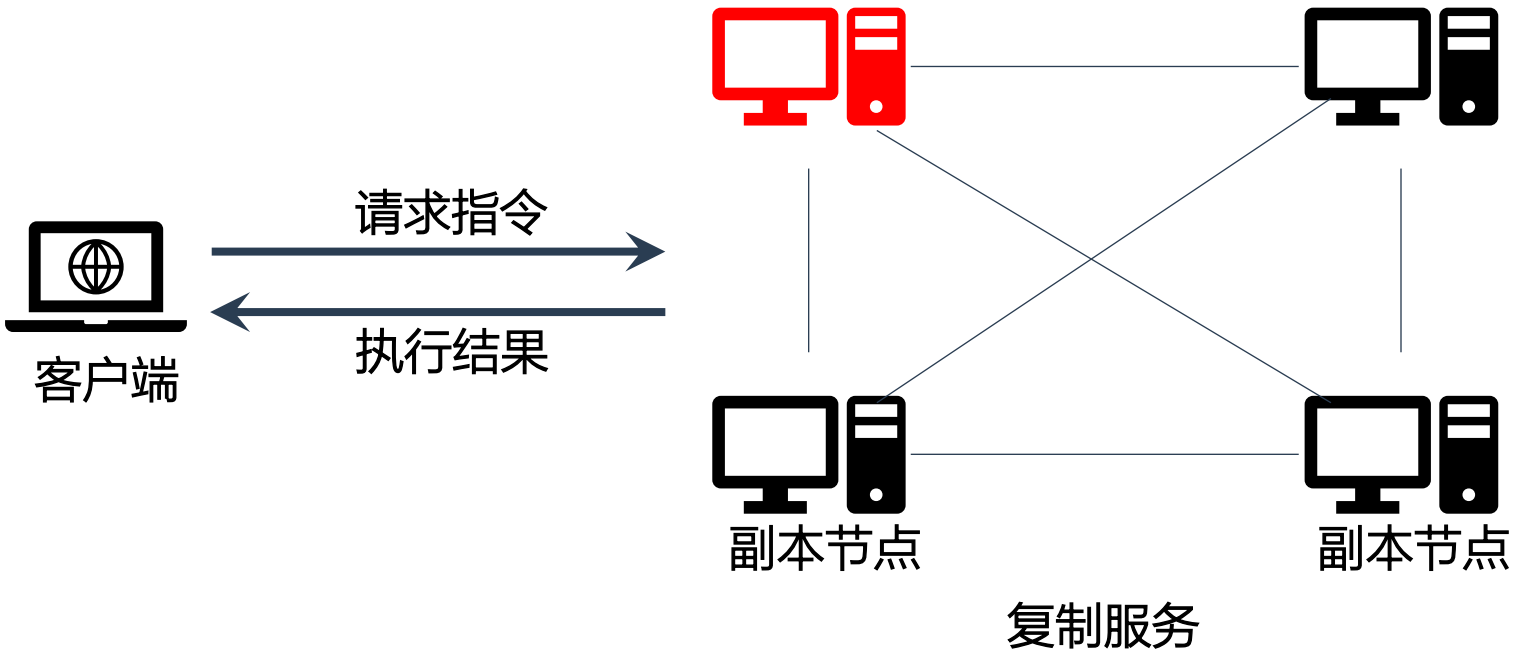
- **实验背景:**

容错算法可以用于构建一个分布式复制服务，能容忍少部分节点出现错误，持续稳定运行

- **实验目的**

了解实际运行一个分布式系统需要配置的参数信息，加深对容错算法的认识

即使服务中的一个节点出现故障，服务依然能稳定运行



第十二章 容错算法的模拟与验证 难度: ★★☆

- **实验要求：**

用4个终端（进程）来构建一个四节点容错服务，一个终端充当客户端，发送服务请求

借助一个拜占庭容错共识开源库bft-smart完成

- 实验大致包含了如下步骤：

1. 开启由四个共识节点组成的映射表服务，并运行客户端使用该服务
2. 手动关闭其中一个节点，然后查看各共识节点的运行状态，并通过客户端判断系统是否能够正常运行
3. 将步骤2中关闭的节点重新开启，查看是否各共识节点的运行状态，判断是否可以恢复正常运行
4. 尝试手动关闭Leader节点对应的机器，再通过客户端发送请求，观察Leader节点更换过程

- **实验资料：**

教材 P435

第十三章 常见Web漏洞演示

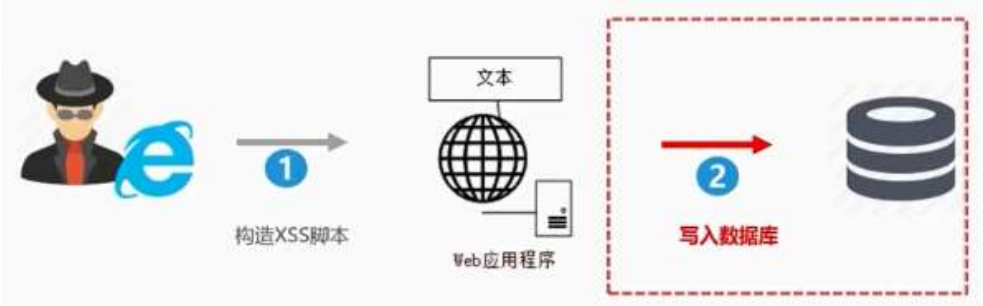
难度: ★★★

- **实验目的:**

演示三种常见的 Web 攻击：的 XSS 、CSRF 、SQL 注入，并体会现代浏览器如何防御这些漏洞。

- **实验要求:**

1. 在2018年后的浏览器上完成本实验，**需要十分小心地关闭一系列防御机制**
2. 编写并部署存在漏洞的示例网站程序
3. 进行持久/非持久XSS，CSFR，以及SQL注入攻击



持久型XSS

非持久型XSS

第十三章 常见Web漏洞演示

难度: ★★★

XSS实验步骤举例

1. 安装Flask框架并运行Flask服务器
2. 访问受害服务器

```
$ flask run
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

3. 在搜索框内输入:

<script>alert(‘反射型XSS’)</script>

4. 在评论框内输入:

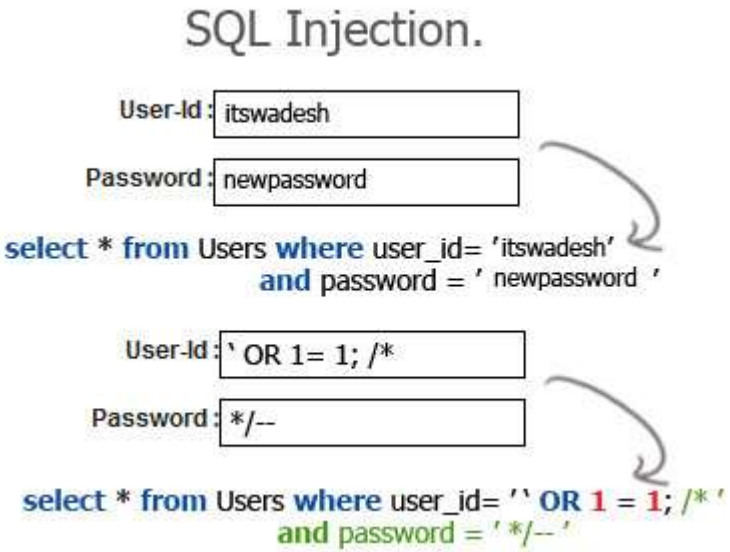
<script>alert('持久型XSS')</script>

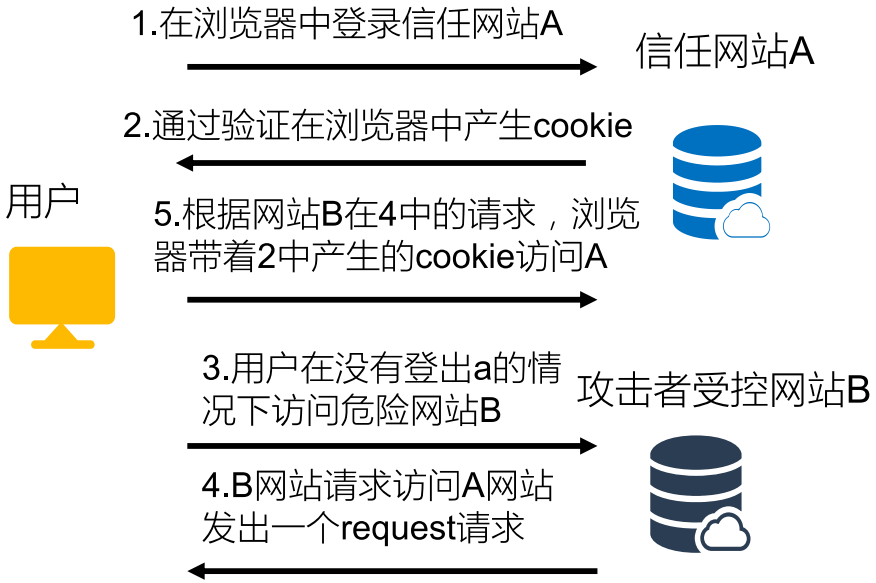
第十三章 常见Web漏洞演示

难度: ★★★

类似地可以完成CSFR和SQL注入攻击

详细的实验教程参考教材





第十四章 神经网络暗门攻击与防御 难度: ★★☆

- **实验背景:**

加深对人工智能安全问题的认识，了解后门攻击的实现逻辑以及防御细节，以及相应的防御

- **实验目的:**

采用成熟的神经网络后门攻击与防御方案，对常见神经网络结构植入后门，并部署后门防御机制

- **实验资料:**

1. 教材 P504
2. BadNets 文章和源代码:

https://arxiv.org/abs/1708.06733

https://github.com/Kooscii/BadNets

3. Neural Cleanse 文章和源代码:

https://ieeexplore.ieee.org/abstract/document/8835365

https://github.com/bolunwang/backdoor

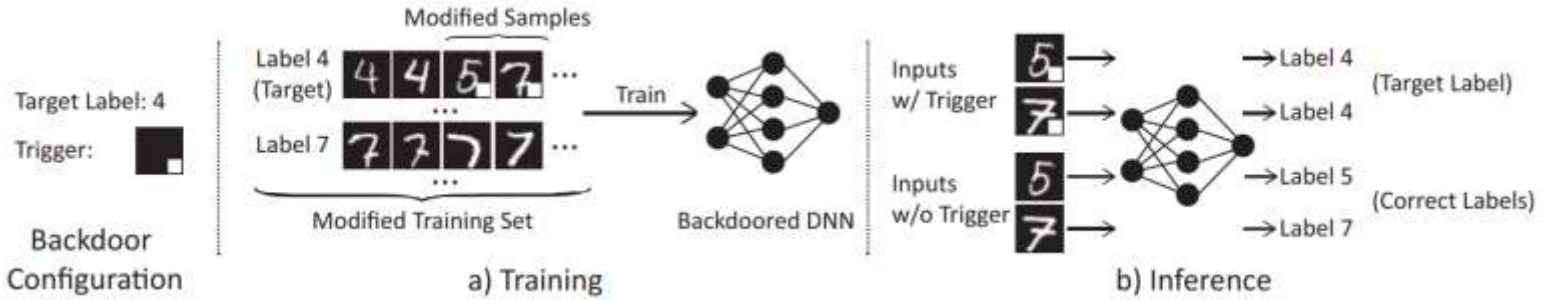
神经网络暗门示意

第十四章 神经网络暗门攻击与防御 难度: ★★☆

- **实验要求：**

1. 复现BadNets暗门方案：不改变原有深度学习模型结构，向训练数据增加特定模式噪音，修改训练数据标签，没有遇到特定模式噪音时正常工作，遇到特定模式噪音数据输出与预定规则相匹配的错误行为

2. 复现Neural Cleanse暗门防御方案：指的是通过利用数据的独特属性或者精心设计的防御机制，来降低后门攻击的成功率



如何获取帮助

1. 《网络空间安全原理与实践》 → 2. 网络学堂当中的实验讲义 → 3. 人工服务

讲义最后有实验讲解内容
与实验附加材料

欢迎实验上机
时间段交流

如何提交实验

- **当我完成了实验之后：**

1. 撰写实验报告

(i) 完整描述实验步骤

(ii) 包含关键实验现象的截图

(iii) 注意保持简洁性

(iv) 期末在网络学堂上提交

2. 向助教演示实验

可通过如下三种方式：

(i) 上机时间现场演示

(ii) 录制视频，连同实验报告上传网络学堂

(iii) 预助教预约远程视频演示

两项均完成时则表明成功提交实验

- **实验截至时间为考试周结束**