网络安全

1. 模拟测试

1、以下关于风险评估模型的说法正确的是

网络和系统防护策略设计应围绕资产价值和资产特点

网络和系统加固的目标是消除所有风险

安全措施虽然可以降低风险，但是在实践中也要考虑实施成本和企业接受能力、资产价值、政策法规等因素

资产价值是吸引攻击者和攻击行为的根本原因

2、同一台宿主机上使用虚拟机软件Virtualbox运行的2台VM之间需要相互直连访问，请问以下哪种虚拟网络连接方式可以实现

NAT

NAT-网络

桥接网络

Host-only

3、以下关于CVSS的说法正确的是

CVSS基础评分高的漏洞在实践中需要引起更高优先级的关注

CVSS基础评分高的漏洞无论在任何情况下都应比评分低的漏洞优先修补

CVSS 2.0和3.0的基础评价组里的指标定义有差异

CVSS评分标准以技术事实为依据对部分单项指标进行主观打分

4、以下关于网络代理的说法正确的是

网络代理可以实现访问者匿名

使用匿名通信网络作为网络访问代理就可以躲避网络攻击溯源和入侵取证了

谨慎使用第三方提供的网络代理服务避免通信数据被监听和篡改

访问来源IP地址无法伪造，因此可以作为攻击溯源的可靠性指标

5、以下哪些手段可以实现中间人攻击

ARP欺骗

DNS缓存投毒

ICMP重定向

代理服务器

6、以下哪些行为特征可以用于判定为典型的网络扫描行为

短时间内来自同一IP地址对另一IP地址大量不同端口发送数据

短时间内来自同一IP地址对另一IP地址单一端口发送大量数据

短时间内来自同一IP地址对另一IP地址单一端口的反复建立TCP连接又迅速关闭TCP连接的行为

短时间内来自不同IP地址的随机端口对单一IP单一端口发送大量数据

7、关于URL： https://sec.cuc.edu.cn/huangwei/wiki/teaching\_cns.html#toc\_2 以下说法正确的是

这是一个使用加密通信协议的网址

这有可能是一个静态网页

如果该网址可以正常打开则说明该网站服务器监听了TCP 443端口

#toc\_2是URL中的请求参数

8、以下关于入侵检测和蜜罐的关键技术原理说法正确的是

入侵检测和蜜罐均可以使用异常检测算法实现入侵分析

入侵检测分析的数据中正常流量和异常流量的比例是未知的，而蜜罐中捕获的所有流量都是异常的

入侵检测可以联动防火墙阻断入站流量，蜜罐不能联动防火墙对入站流量进行控制

入侵检测系统可以部署在蜜罐和蜜网中

\*题解：Honeyfarm就是一个典型的可以利用防火墙来控制流量重定向到其他蜜罐的安全机制。

9、以下关于Bro和Snort的关系说法正确的是

Snort和Bro都支持读取并分析pcap文件

Snort和Bro都可以部署为IDS模式

Snort和Bro都可以用于入侵取证

Snort无法用于检测DoS攻击，Bro可以

10、以下关于风险评估模型说法正确的是

XSS攻击可能会破坏网站用户数据的机密性、完整性和可用性

任意文件上传漏洞如果能上传服务端脚本程序并能执行，则该类漏洞通常会被评级为高危漏洞

WEB应用程序的威胁来源是应用程序代码漏洞，用户安全意识差、网站和网络配置缺陷等不属于WEB应用程序的威胁来源

对于一个网站来说，网站源代码、用户数据、网站服务器等都属于网站的资产范畴

11、以下有关漏洞的说法正确的是

漏洞只有被利用了才能被定义为漏洞，否则只能被称之为威胁

漏洞只要发现了就能被称之为威胁

漏洞利用行为又被称为攻击行为

漏洞利用行为可能会因为网络和系统防御措施而失败

12、以下关于等级安全保护概念正确的是

等级安全保护的设计是为了有效解决漏洞防御的全面性和网络与系统安全的短板问题

等级安全保护的理念是围绕资产价值的大小和分布情况进行“尽力而为”防御

高安全等级的防护措施不能用于低安全等级资产的保护

个人门户网站的安全保护等级通常要低于金融交易系统的安全保护等级

13、以下关于融合网安全的说法正确的是

融合网安全是个伪命题，实质上就是计算机网络安全

融合网安全的研究内容取决于融合网技术特点，当前融合网的趋势之一是IP网络，因此计算机网络安全是融合网安全的核心研究内容之一

由于网络融合的演进和发展使得电信网络和广电网络将面临大量来自计算机网络的威胁

物联网技术的发展对现有网络增加了更多的攻击入口点和威胁来源

14、以下关于网络安全分层模型的说法正确的是

局域网的安全治理是互联网安全的基础这符合“下一层的安全机制是上一层安全机制的基础”理论

在不安全的局域网中使用VPN和应用层数据加密进行互联网通信可以改进和提高直接进行互联网通信的安全等级

使用安全操作系统并不能杜绝WEB应用程序已有的XSS漏洞造成的风险

基于文件系统的强制访问控制机制不是设计用来阻止基于硬盘数据直接拷贝方式的非授权数据访问的

15、以下关于基于生物特征识别的身份认证机制的说法正确的是

在可以物理接触手机的场景下，手机的指纹解锁机制安全等级是低于基于复杂口令策略的认证机制的

指纹解锁由于是基于生物特征的复杂性和唯一性，因此基于指纹方式的身份认证安全保护等级明显高于口令方式

可以同时使用刷脸认证+指纹认证这种双因素认证方式来取代基于口令认证+指纹认证方式获得更高的身份认证强度

基于生物特征识别的身份认证方式存在隐私泄漏、取消授权困难、被活体克隆复制等风险

16、以下关于访问控制理论的说法正确的是

在非强制访问控制模型中，属主权限的所有者可以撤销自己所拥有的文件读写执行访问权限后再重新授权

没有身份认证就无法实现访问授权，换句话来说如果身份认证失败则没必要再进行访问授权检查

使用OAuth协议实现的第三方账号登录功能属于单点登录技术，所以OAuth协议是一个身份认证协议

CSRF漏洞的利用原理是一种仿冒身份威胁，如果被攻击者使用的是管理员权限执行操作，则CSRF漏洞的利用效果相当于同时实现了提升权限

17、以下关于身份认证机制的说法正确的是

网站限制用户使用超过16位字符长度的口令是一种不负责任的安全策略，在实践中不应限制用户口令长度

网站提供找回口令功能就是要把用户的旧口令通过安全通信渠道发送给用户

“忘记密码”和“找回口令”功能的实现需要防御用户名探测和枚举威胁

“忘记密码”和“找回口令”功能如果是基于查收电子邮件或短信验证码方式则也可以作为现有用户名+口令认证方式的一个辅助认证手段，也属于一种双因素认证方式

18、以下关于数字证书的说法正确的是

大多数HTTPS网站都只实现了单向认证，即：客户端会验证服务器的合法性，而服务器不会验证客户端的合法性

网银支付中使用的U盾相当于用于服务器验证用户真实身份的一种凭据

现有的基于证书的互联网PKI体系相当于是一种“森林”结构，即没有一个集中的“信任根”，而是同时有多个“信任根”

用于网站HTTPS的数字证书必须确保CN字段与访问网站使用的主机名字段匹配，否则现代浏览器也会报告证书设置错误警告或阻止用户继续访问

19、以下关于匿名通信技术的说法错误的是

使用匿名通信技术就是为了实现通信地址的隐藏

洋葱路由技术对通信数据进行了多层加密

使用透明HTTP代理可以实现通信关系匿名

加密应用层负载是实现通信地址隐藏的必要技术

20、以下关于互联网通信过程中的IP地址识别与隐藏技术的说法正确的是

使用洋葱路由代理技术访问互联网中的一个普通HTTPS网站依然存在被网站追踪和定位的风险

使用nmap对互联网中的一个IP进行源IP地址欺骗扫描可以在不暴漏扫描源IP的前提下实现探测目标IP的端口开放情况

使用nmap对局域网中的一个IP进行源IP地址欺骗扫描可以在不暴漏扫描源IP的前提下实现探测目标IP的端口开放情况

只要使用了权威准确的IP归属地数据库就可以较为准确的根据网站访问日志中的请求来源IP地址确定真实的网站攻击者地理位置

21、以下关于网络嗅探的说法正确的是

ARP欺骗既可以劫持局域网主机到网关之间的通信流量，也可以劫持局域网主机之间的通信流量

对交换机进行DoS攻击迫使交换机退化到广播（集线器）模式这种攻击方式可以通过配置交换机限制单个端口允许的源MAC地址数量的方法来缓解风险

安装桌面ARP防火墙无法阻止针对交换机DoS和CAM表毒化方式的网络嗅探攻击

如果局域网中的主机被通过网络扫描方式发现网卡工作在混杂模式则说明该主机当前正在进行ARP欺骗攻击

22、以下关于渗透测试的说法正确的是

虽然没有取得被测试目标的授权但只要没有进行破坏性活动这就是一次合法的渗透测试活动

渗透测试与网络攻击的唯一区别就是是否取得合法授权

渗透测试过程中即使为了验证目标系统的漏洞可利用性，也不能无限制的利用漏洞获取用户数据

渗透测试报告中一定要包含对于发现漏洞的修补建议

23、以下关于Google Hacking技术的说法正确的是

网站资源一旦被Google收录就意味着全世界范围内所有人都可能在网站所有者不知情的情况下访问到该资源

网站资源一旦被Google收录即使网站所有者删除了自己网站上的原始资源，其他人依然可以通过Google的网页缓存功能访问到，网站所有者只能被动等待该资源缓存超时后自动删除

使用robots.txt可以限制Google的爬虫爬取和收录哪些网站资源

Google Hacking技术仅限于Google搜索引擎，其他搜索引擎没有类似功能

24、以下关于网络流量捕获技术的说法正确的是

使用tcpdump、snort、wireshark和bro都可以捕获网络流量

网络流量捕获过程中如果不设置抓包过滤条件则不会丢包

网络流量保存到文件之后无法进行数据包内容编辑

使用snort和bro都可以对pcap文件进行流量分析和应用层负载检测

25、以下关于计算机取证的说法正确的是

计算机取证过程为了保证数据完整性必须严格断开网络

关机状态下拆卸下硬盘后直接挂载到一台物理隔离、安全可靠的电脑上进行数据恢复是正确的取证操作方法

云取证技术和服务器取证技术完全相同

入侵检测、蜜罐和蜜网均可以用于计算机取证

26、以下关于P2DR模型的说法正确的是

安全策略的设计是整个模型的内核和中心，安全策略不是一成不变的，会随着安全防护、检测和响应技术的实施经验积累而不断完善

蜜罐和蜜网既可以应用于安全检测，也可以用于安全防护，还可以作为一种应急响应手段

系统安全保护过程从防护开始，到响应结束

对于防御一方来说对零日漏洞的防护、检测和应急响应能力建设通常也要服从和匹配于企业的总体安全策略设计，例如等级安全保护原则

27、以下关于电信诈骗的说法正确的是

+95-555和95555都是招商银行的官方电话号码

通过百度搜索出来的官方认证网站上发布的信息都是真实可靠的

网上订机票后立刻接到电话通知航班取消和要求改签的电话通知说明，在电话中对方可以准确的报出你的姓名、航班号信息说明这确实是官方通知应按照通知说明去操作退改签

微博蓝V、黄V认证账号发布的微博中包含的链接也不能随便点击

28、以下关于主机入侵防御系统fail2ban的说法正确的是

只能用于WEB服务器和SSH服务器的暴力破解类攻击检测和防护

一旦检测到攻击行为除了记录日志只能通过联动iptables设置规则来对攻击源进行屏蔽

基本检测原理是通过轮询日志文件的方式统计轮询周期内特定模式行为出现次数是否超过预设置阈值来报告异常

可以通过内置的小工具fail2ban-regex调试检测规则的正确性和鲁棒性

29、以下关于HSTS（HTTP Strict Transport Security）的说法正确的是

服务器通过实现HSTS机制，可以帮助客户端（例如浏览器）强制客户端和服务器之间的所有通信流量必须使用HTTPS

使用HSTS就可以保护Windows XP用户在使用IE 6浏览器访问HTTPS站点时避免被替换证书方式实现中间人劫持攻击

一旦浏览器访问过设置了HSTS机制的HTTPS站点后，即使该站点同时支持HTTPS站点，则该浏览器将永远无法再通过HTTP方式访问该站点，只能被强制以HTTPS方式访问该站点

浏览器厂商也可以在浏览器发布时内置一份启用HSTS站点的名单而不用等待服务器设置HSTS标志

30、以下哪些概念存在于CVSS 2.0标准中但不属于CVSS 3.0标准

Access Vector

Attack Vector

Privileges Required

User Interaction

31、以下哪些概念是CVSS 3.0标准中但不属于CVSS 2.0标准

Access Vector

Attack Vector

Privileges Required

User Interaction

32、以下关于域名解析服务DNS的说法正确的是

域传送是域名管理员用于备份DNS配置记录数据库的一种手段，如果没有设置访问来源IP白名单则可能被攻击者用来快速获取子域名信息

域名管理的组织架构机制是一个典型的树状层次结构，顶级域名的解析由根域名解析服务器负责

全球共有13台根域名服务器。这13台根域名服务器中名字分别为“A”至“M”，其中10台设置在美国，另外各有一台设置于英国、瑞典和日本

sec.cuc.edu.cn这个域名的递归解析过程总是完全由中国国家境内的域名解析服务器负责解析

33、以下哪些安全防御措施对于文件上传漏洞的防御或修补有实质性加固作用

限制特定文件扩展名（例如.php、.php5、.exe、.html、.swf、.js）作为上传文件的文件扩展名

文件上传目录禁止脚本解析和执行功能

对于图片类文件上传使用ImageMagick库对上传图片进行重新渲染后保存到文件系统

上传后文件统一重命名

34、以下哪些安全防御措施对于SQL注入漏洞的防御或修补没有实质性加固作用

禁止用户输入字符包含SQL关键词（例如：UNION/SELECT/DELTE/UPDATE/INSERT/FROM等）及其所有变形形式（大小写不区分匹配、多种等价字符集编码方式匹配）

使用最小授权的数据库连接账号，删除数据库中的高风险扩展功能

使用预编译的SQL来处理用户输入的数据、动态构造SQL语句

对于关键和重要的数据库操作使用图形验证码和短信验证码进行多因素认证

35、以下哪些安全防御措施对于CSRF漏洞的防御或修补有实质性加固作用

后端服务器API一律使用POST方法代替GET方法

对于关键和重要的后端服务器操作使用图形验证码和短信验证码进行多因素认证

在链接中使用时间戳和加密（防⽌简单重放）

使用htmlspecialchars和escapeshellcmd对用户请求参数进行过滤和转义

模拟测试（是非题）

Web应用程序的文件上传漏洞主要针对PHP、JSP编写的网站程序，使用Python语言开发的文件上传应用不存在此类漏洞。

Yes

No

使用Burpsuite可以篡改HTTP请求数据无法修改HTTP响应数据。

Yes

No

使用HTTP代理服务器访问HTTPS站点可以杜绝代理服务器查看通信内容。

Yes

No

一个域名可以解析到多个IP，一个IP也可以被多个域名解析指向，一个IP地址上还可以有多个网站实例运行，一个网站也可以在多个IP地址上运行。

Yes

No

使用最新版本浏览器点击一个网页链接对点击者来说面临的风险主要包括：XSS、CSRF、钓鱼站点、下载可执行文件并自动运行、格式化硬盘。

Yes

No

授权检查检查的前提是先通过身份认证、确定用户唯一标识，如果没有通过身份认证就不需要做授权检查。

Yes

No

信息安全的本质是持续对抗，但对抗的双方存在信息不对称现象，因此防御一方设计出了蜜罐、蜜网等手段来收集攻击者的攻击行为和攻击工具，并基于此分析攻击意图和可能的攻击影响。

Yes

No

使用Google Hacking技术可以绕过目标系统和网络的防火墙和入侵检测系统实现信息收集目标。

Yes

No

防火墙是在两个信任程度不同的网络边界实现访问控制的手段之一，常用的防火墙响应处置策略包括：拒绝、重定向、静默丢弃和记录日志等。

Yes

No

“当智能手机系统处于锁屏状态下，可以使用相机拍照但不能查看相册中的照片”，这是符合Biba强制访问控制策略模型的。

Yes

No

模拟测试（排序题）

1、请排列组合出Nmap扫描一般流程顺序：主机可达性扫描——>端口扫描——>主机详细信息扫描——>脚本扫描

2、请排列组合出渗透测试一般流程顺序：取得北侧是目标的法律授权——>信息收集——>目标踩点——>网络扫描——>漏洞发现——>漏洞利用——>提供测试报告

1. 各章实验
2. VirtualBox虚拟网络详解

VirtualBox的提供了四种网络接入模式，它们分别是：

1、NAT 网络地址转换模式(NAT,Network Address Translation)

2、Bridged Adapter 桥接模式

3、Internal 内部网络模式

4、Host-only Adapter 主机模式

第一种 NAT模式

解释：

NAT模式是最简单的实现虚拟机上网的方式，你可以这样理解：Vhost访问网络的所有数据都是由主机提供的，vhost并不真实存在于网络中，主机与网络中的任何机器都不能查看和访问到Vhost的存在。

虚拟机与主机关系：

只能单向访问，虚拟机可以通过网络访问到主机，主机无法通过网络访问到虚拟机。

虚拟机与网络中其他主机的关系：

只能单向访问，虚拟机可以访问到网络中其他主机，其他主机不能通过网络访问到虚拟机。

虚拟机与虚拟机之间的关系：

相互不能访问，虚拟机与虚拟机各自完全独立，相互间无法通过网络访问彼此。

IP:10.0.2.15

网关：10.0.2.2

DNS：10.0.2.3

一台虚拟机的多个网卡可以被设定使用 NAT， 第一个网卡连接了到专用网 10.0.2.0，第二个网卡连接到专用网络 10.0.3.0，等等。默认得到的客户端ip（IP Address）是10.0.2.15，网关（Gateway）是10.0.2.2，域名服务器（DNS）是10.0.2.3，可以手动参考这个进行修改。

NAT方案优缺点：

笔记本已插网线时： 虚拟机可以访问主机，虚拟机可以访问互联网，在做了端口映射后（最后有说明），主机可以访问虚拟机上的服务（如数据库）。

笔记本没插网线时： 主机的“本地连接”有红叉的，虚拟机可以访问主机，虚拟机不可以访问互联网，在做了端口映射后，主机可以访问虚拟机上的服务（如数据库）。

第二种 Bridged Adapter模式

解释：

网桥模式是我最喜欢的用的一种模式，同时，模拟度也是相当完美。你可以这样理解，它是通过主机网卡，架设了一条桥，直接连入到网络中了。因此，它使得虚拟机能被分配到一个网络中独立的IP，所有网络功能完全和在网络中的真实机器一样。

虚拟机与主机关系：

可以相互访问，因为虚拟机在真实网络段中有独立IP，主机与虚拟机处于同一网络段中，彼此可以通过各自IP相互访问。

虚拟机于网络中其他主机关系：

可以相互访问，同样因为虚拟机在真实网络段中有独立IP，虚拟机与所有网络其他主机处于同一网络段中，彼此可以通过各自IP相互访问。

虚拟机于虚拟机关系：

可以相互访问，原因同上。

IP：一般是DHCP分配的，与主机的“本地连接”的IP 是同一网段的。虚拟机就能与主机互相通信。

笔记本已插网线时：（若网络中有DHCP服务器）主机与虚拟机会通过DHCP分别得到一个IP，这两个IP在同一网段。 主机与虚拟机可以ping通，虚拟机可以上互联网。

笔记本没插网线时：主机与虚拟机不能通信。主机的“本地连接”有红叉，就不能手工指定IP。虚拟机也不能通过DHCP得到IP地址，手工指定IP后，也无法与主机通信，因为主机无IP。

这时主机的VirtualBox Host-Only Network 网卡是有ip的，192.168.56.1。虚拟机就算手工指定了IP 192.168.56.\*，也ping不能主机。

第三种 Internal模式

解释：

内网模式，顾名思义就是内部网络模式，虚拟机与外网完全断开，只实现虚拟机于虚拟机之间的内部网络模式。

虚拟机与主机关系：

不能相互访问，彼此不属于同一个网络，无法相互访问。

虚拟机与网络中其他主机关系：

不能相互访问，理由同上。

虚拟机与虚拟机关系：

可以相互访问，前提是在设置网络时，两台虚拟机设置同一网络名称。如上配置图中，名称为intnet。

IP: VirtualBox的DHCP服务器会为它分配IP ，一般得到的是192.168.56.101，因为是从101起分的，也可手工指定192.168.56.\*。

笔记本已插网线时：虚拟机可以与主机的VirtualBox Host-Only Network 网卡通信

这种方案不受主机本地连接（网卡）是否有红叉的影响。

第四种 Host-only Adapter模式

解释：

主机模式，这是一种比较复杂的模式，需要有比较扎实的网络基础知识才能玩转。可以说前面几种模式所实现的功能，在这种模式下，通过虚拟机及网卡的设置都可以被实现。

我们可以理解为Vbox在主机中模拟出一张专供虚拟机使用的网卡，所有虚拟机都是连接到该网卡上的，我们可以通过设置这张网卡来实现上网及其他很多功能，比如（网卡共享、网卡桥接等）。

虚拟机与主机关系

默认不能相互访问，双方不属于同一IP段，host-only网卡默认IP段为192.168.56.X 子网掩码为255.255.255.0，后面的虚拟机被分配到的也都是这个网段。通过网卡共享、网卡桥接等，可以实现虚拟机于主机相互访问。

虚拟机与网络主机关系

默认不能相互访问，原因同上，通过设置，可以实现相互访问。

虚拟机与虚拟机关系

默认可以相互访问，都是同处于一个网段。

虚拟机访问主机 用的是主机的VirtualBox Host-Only Network网卡的IP：192.168.56.1 ，不管主机“本地连接”有无红叉，永远通。

主机访问虚拟机，用是的虚拟机的网卡3的IP： 192.168.56.101 ，不管主机“本地连接”有无红叉，永远通。

虚拟机访问互联网，用的是自己的网卡2， 这时主机要能通过“本地连接”有线上网，（无线网卡不行）

第三章 网络安全应用基础

1. HTTP代理服务器实验

Q：使用http代理服务器访问HTTPS站点时，通信传输内容是否会被代理服务器“看到”？

A：结论是代理服务器不知道客户端和服务器的HTTPS通信内容，但代理服务器知道客户端访问了哪个HTTPS站点，这是由http代理的协议机制决定的：代理客户端会发送Connect请求到http代理服务器。

实验验证：在Kali Linux中安装tinyproxy，然后用主机设置浏览器代理指向tinyproxy建立的HTTP正向代理，在Kali中用wireshark抓包，分析抓包过程，理解HTTP正向代理HTTPS流量的特点。

提醒注意：

HTTP代理服务器在转发客户端请求时，可能会添加Via字段，从而向目标站点暴露客户端正在使用代理访问。类似的，匿名通信应用tor的部分出口节点也会在http请求中自动加入via字段，向被访问站点宣告：当前请求正在使用匿名通信网络tor提供的匿名通信服务。

Q: 了解什么是HSTS

A: HSTS（HTTP Strict Transport Security）机制：

· 服务器通过实现HSTS机制，可以帮助客户端（例如浏览器）强制

客户端和服务器之间的所有通信流量必须使用HTTPS

SSL证书合法性校验

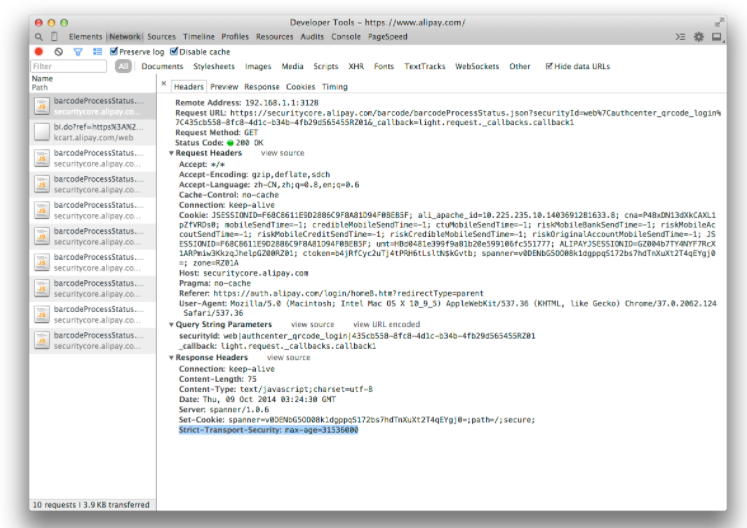


图1 在Chrome的开发者工具中查看启用HSTS站点的通信数据

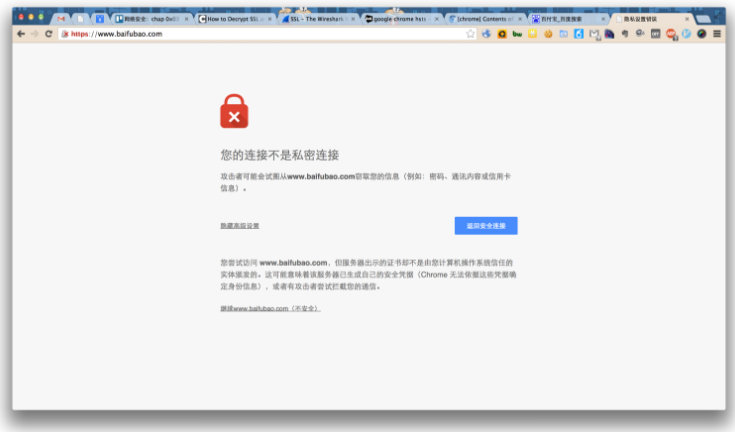
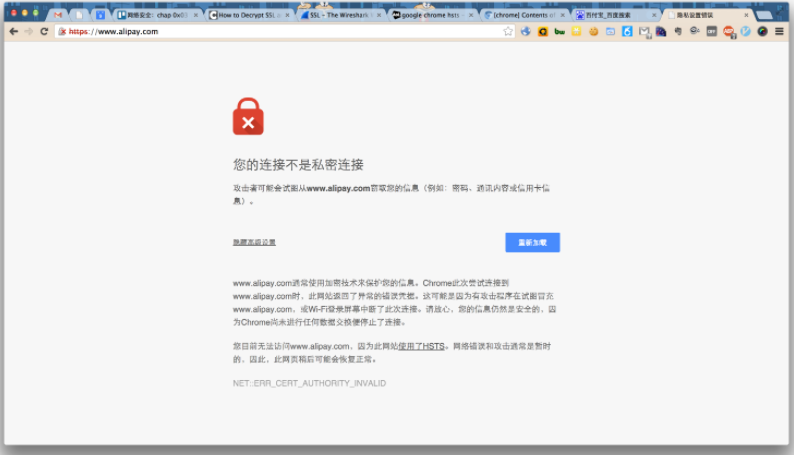
图2 未启用HSTS站点被SSL中间人攻击时Chrome浏览器的警告提示

图3 未启用HSTS站点被SSL中间人攻击时Chrome浏览器在用户允许继续访问情况下

图4 启用HSTS站点被SSL中间人攻击时Chrome浏览器禁止用户继续访问

Ref:

· 使用HTTP代理服务器的安全性简评

由于各种原因，在国内上网，免不了要和HTTP代理服务器打交道。HTTP代理服务器分加密代理服务器和普通代理服务器。如果我们在自己的电脑上运行了某些软件之后就能访问之前不能访问的网站，那么很可能是加密代理；如果我们直接在浏览器中设置代理服务器，不用运行任何软件，那么这种类型的代理服务器就是普通代理。一般来说，如果加密代理软件没有问题，加密代理的安全性要好于普通代理。在这里，本文主要还是讲述普通代理的安全性。

　　1、HTTP代理协议简介

　　浏览器和HTTP代理服务器之间是通过HTTP代理协议进行通讯的。如果我们没有使用运行在自己的电脑上的加密代理，而是直接在浏览器中设置了在互联网上的代理服务器地址，那么浏览器就和代理服务器之间进行的通讯就会发送到互联网上。代理服务器常见的请求有GET、POST和CONNECT，下面分别做介绍：

　　1.1、GET

　　如果我们通过代理服务器访问某个http协议的网站网页，那么浏览器就会向HTTP代理服务器发送GET请求。比如我们通过代理服务器访问 http://www.microsoft.com/ ，那么浏览器就会向HTTP代理服务器发送数据：

　　GET http://www.microsoft.com/ HTTP/1.1

　　……

　　之后代理服务器取得数据后把网页返回给浏览器：

　　HTTP/1.1 200 OK

　　……

　　<html>

　　……

　　我们来和没有使用代理服务器的情况对比一下，如果没有使用代理服务器，那么我们访问 http://www.microsoft.com/ ，浏览器就会向 www.microsoft.com 这台服务器发送以下请求：

　　GET / HTTP/1.1

　　……

　　服务器返回：

　　HTTP/1.1 200 OK

　　……

　　<html>

　　……

　　我们看到，使用代理服务器之后，GET请求和返回的格式基本没有变化，是否使用代理服务器的安全性上基本没有差异。

　　1.2、POST

　　当我们在使用http协议的网站上提交表单时（比如网站上的用户登陆表单，填写好用户名和口令，然后点登陆，就把表单提交到服务器了），如果表单类型是POST（大多数表单类型都为POST，不过搜索引擎的搜索表单好像都是GET类型的），那么浏览器就会向代理服务器发送POST请求。比如在 http://www.williamlong.info/archives/2209.html 中进行评论，数据会提交到 http://www.williamlong.info/cmd.asp?act=cmt&key=a666b083 ，如果我们使用HTTP代理服务器，以下请求会发送到代理服务器：

　　POST http://www.williamlong.info/cmd.asp?act=cmt&key=a666b083 HTTP/1.1

　　……

　　[提交的评论数据]

　　如果没有使用代理服务器，那么以下POST请求直接发送到 www.williamlong.info 服务器：

　　POST /cmd.asp?act=cmt&key=a666b083 HTTP/1.1

　　……

　　[提交的评论数据]

　　使用代理服务器和没有使用代理服务器的返回结果也是基本一致的。所以POST请求在是否使用代理服务器的安全性上也是基本没有差异的。

　　1.3、CONNECT

　　当我们访问https协议的网站时，浏览器会向代理服务器发送CONNECT请求。比如我们访问 https://mail.google.com/mail/?shva=1#inbox 时，浏览器会向代理服务器发送以下请求：

　　CONNECT mail.google.com:443 HTTP/1.0

　　……

　　之后服务器返回：

　　HTTP/1.1 200 Connection established

　　……

　　然后浏览器开始向代理服务器发送加密的数据，使用的是 mail.google.com 的SSL证书。代理服务器把 mail.google.com 返回的加密数据原原本本的返回给浏览器。

　　如果没有使用代理服务器，那么浏览器直接连接 mail.google.com:443 ，然后开始发送和接收加密数据。

　　我们看到，使用了HTTP代理服务器之后，HTTPS协议传输的内容仍旧是端到端加密的，HTTPS仍旧保持了传输的内容不被任何负责传输数据的设备（包括代理服务器）看到的特性。是否使用代理服务器对HTTPS协议的安全性没有变化。

　　2、深入讨论

　　Q：我们看到，使用了代理服务器之后，都会把需要访问网站的域名（包括https的）发送给代理服务器，这样如果网络受到监视，就能知道我们在访问什么网站（但https协议监控不到传输的内容，只能知道我们访问了这个网站），这样是否降低了安全性？

　　A：如果我们的网络受到监视，所有的网络数据包都被记录，那么如果不用代理服务器，浏览器也会首先去发送查询域名所对应的IP的DNS请求，还会在得到IP后发送TCP连接请求到需要访问的网站，通过网络数据包监视一样可以知道我们在访问什么网站。

　　Q：有些代理服务器产品声称能够记录HTTPS传输的内容，是怎么做到的呢？

　　A：使用了SSL劫持的手段。这些代理服务器产品无一例外需要安装对应的客户端，一旦安装了这些客户端，就能控制客户端电脑，进而安装代理服务器自己的证书，进行SSL劫持。有些产品不需安装客户端，但客户端浏览器访问所有的HTTPS网站时都通不过浏览器的SSL证书验证，浏览器都会给出证书无效的警告。

　　Q：如果我们原先不能访问某个网站，使用了代理服务器可以放问这个网站了，这个网站是http协议的，这样访问这个网站所有的行为在受监控的网络环境内都会被记录吧？

　　A：是的。因为代理服务器不改变HTTP协议的安全性，所以所有明文传输的内容都能被监控系统记录下来。所以，我们使用代理服务器提交重要信息（比如用户名和口令）时，都要选择HTTPS协议，在输入用户名和口令时需要注意浏览器的地址栏是不是以https开头的。

　　Q：使用了代理服务器后，网络数据包所经过的路和没有使用代理服务器时不同，对安全性有什么影响呢？

　　A：这要取决于使用了代理服务器前后中所经过的路中的每一个节点了。即使不使用代理服务器，国内上网仍旧会面临各种人为网络故障，使用代理服务器后因为经过的路不同，所以有可能还是需要面对这些人为网络故障，也有可能不需要面对这些人为网络故障了（如果代理服务器本身特意解决了这些故障）。

　　3、总结

综合上述，使用远程HTTP代理服务器的安全性和不使用代理服务器相比基本没有什么差别，我们面对的人为网络故障在使用了代理服务器后也有可能依然存在，也有可能减少甚至消失了。如果有人在国内提供解决了人为网络故障的代理服务器，给一些不会使用或无法使用加密代理软件的人用，结合《如何使用代理服务器》和AutoProxy，那么全民代理的时代说不定会到来哦。

· 如何使用代理服务器上网

　什么是代理服务器

　　代理服务器（Proxy Server）是网上提供转接功能的服务器，在一般情况下，我们使用网络浏览器直接去连接其他Internet站点取得网络信息时，是直接联系到目的站点服务器，然后由目的站点服务器把信息传送回来。代理服务器是介于客户端和Web服务器之间的另一台服务器，有了它之后，浏览器不是直接到Web服务器去取回网页而是向代理服务器发出请求，信号会先送到代理服务器，由代理服务器来取回浏览器所需要的信息并传送给你的浏览器。

　　比如你想访问的目的网站是A，由于某种原因你不能访问到网站A或者你不想直接访问网站A（这样通过代理服务器网站A，对网站A而已可以隐藏你自己的身份，也就是不知道是谁访问的网站，而认为是代理服务器访问的），此时你就可以使用代理服务器，在实际访问网站的时候，你在浏览器的地址栏内和你以前一样输入你要访问的网站，浏览器会自动先访问代理服务器，然后代理服务器会自动给你转接到你的目标网站。

　　代理服务器的作用

　　提高访问速度：通常代理服务器都设置一个较大的缓冲区，当有外界的信息通过时，同时也将其保存到缓冲区中，当其他用户再访问相同的信息时，则直接由缓冲区中取出信息，传给用户，以提高访问速度。

　　隐藏真实身份：上网者也可以通过代理服务器隐藏自己的真实地址信息，还可隐藏自己的IP，防止被黑客攻击。

　　突破限制：有时候网络供应商会对上网用户的端口，目的网站，协议，游戏，即时通讯软件等的限制，使用代理服务器都可以突破这些限制。

　　代理服务器主要类型

　　HTTP代理：最简单的一种代理形式，能够代理客户机的HTTP访问，上网浏览网页使用的都是HTTP协议，通常的HTTP代理端口为80、3128或8080端口。

　　SOCKS代理：SOCKS代理与HTTP等其他类型的代理不同，它只是简单地传递数据包，而并不关心是何种应用协议，既可以是HTTP协议，也可以是FTP协议，或者其他任何协议，所以SOCKS代理服务器比其他类型的代理服务器速度要快得多。SOCKS代理又分为SOCKS4和 SOCKS5，二者不同的是SOCKS4代理只支持TCP协议（即传输控制协议），而SOCKS5代理则既支持TCP协议又支持UDP协议（即用户数据包协议），还支持各种身份验证机制、服务器端远程域名解析（解决DNS污染就靠这个了）等。SOCK4能做到的SOCKS5都可得到，但SOCKS5能够做到的SOCKS则不一定能做到。目前SOCKS5是最常用的一种SOCKS代理。

　　如何使用HTTP代理服务器

　　HTTP代理服务器的设置方法，对于IE和FireFox设置略有不同。设置前需要先找一些可用的免费代理服务器地址。

　　IE的设置是这样的，打开IE浏览器，选择菜单栏的“工具/Internet选项...”。

　　这时候分两种情况，对于ADSL拨号用户来说，选择一个网络连接后，点“设置”，如下图所示，选中代理服务器，填入地址和端口号。

IE代理服务器设置

　　对于局域网用户来说，需要点“局域网设置”，如下图所示，选中代理服务器，填入地址和端口号。

IE代理服务器设置

　　FireFox的设置和IE类似，打开FireFox浏览器，选择菜单栏的“工具/选项...”。

　　这时选择“高级/网络”，点设置，就出现下面的界面，就可以进行代理服务器的设置了，选中“手动配置代理”，然后填写代理服务器的地址和端口。

FireFox代理服务器设置

　　如何使用SOCKS代理服务器

　　这里就到重点内容了，SOCKS代理是目前功能最为全面，使用最为稳定的代理服务器，我目前上网就只用SSH搭建SOCKS代理服务器上网，访问网络没有任何限制。下面我就着重讲一下如何使用SOCKS代理服务器。

　　用SSH搭建SOCKS代理上网，建议使用Firefox浏览器，因为Firefox支持SOCKS代理远程域名解析，而IE只能通过类似SocksCap这样的第三方软件实现，不是很方便。

　　配置Firefox浏览器

　　在Firefox设置SOCKS远程域名解析，主要是为了防止DNS污染，具体设置方法是，在Firefox地址栏中，输入 about:config ，按确认，修改里面的一项数值，改成 network.proxy.socks\_remote\_dns=true 就可以了。

Firefox远程域名解析设置

　　然后，打开FireFox浏览器，选择菜单栏的“工具/选项...”。选择“高级/网络”，点设置，就出现下面的界面，就可以进行代理服务器的设置了，选中“手动配置代理”，然后在SOCKS主机上，填写代理服务器的地址127.0.0.1，端口1080，SOCKS类型选择“SOCKS V5”，这时Firefox就配置结束。

Firefox的SOCKS5代理设置

　　设置SSH

　　配置好了Firefox，就该配置SSH了，安全外壳协议（Secure Shell Protocol / SSH）是一种在不安全网络上提供安全远程登录及其它安全网络服务的协议。常用的SSH工具有开源软件PuTTY，支持SSH远程登录的主机可以实现socks5代理服务器的功能，不过在PuTTY中没有配置文件，需要手动设置才能实现，且无法保存，而PuTTY完整版自带的pLink可以实现命令行方式调用PuTTY实现SSH的加密通道。

　　具体的方法是，去PuTTY官方网站下载pLink这个文件，pLink的调用参数是：plink -C -v -N -pw 密码 -D 本地端口 远程用户@IP或域名:远程希望打开的端口。

　　新建一个文件，写入以下内容，另存为pLink.bat批处理文件，并放在Putty的安装目录内。

　　@plink -N Username@sshServer -pw Password -D 127.0.0.1:1080

　　请将Username sshServer Password三处改为用户自己登陆SSH服务器的用户名、服务器地址和密码。这个SSH帐号可以通过多种方法获得，例如用户购买了某些国外主机空间或VPS就会有SSH帐号，或者在淘宝网也有SSH帐号出售，我自用的SSH帐号是用每年100美元购买虚拟主机时赠送的，通常SSH帐号的价格大约是每年几十元人民币左右，也有少量国外网站提供免费的SSH帐号。

　　执行这个批处理文件，保持其窗口开启，一旦关闭窗口代理便失效。然后打开已经配置好127.0.0.1:1080的Socks5代理的Firefox浏览器，就可以使用SOCKS代理服务器上网了。

　　其他设置技巧

为了方便代理服务器的快速切换，我推荐两个FireFox代理服务器扩展，一个是QuickProxy，可以实现一键切换代理功能，QuickProxy安装后在状态栏有一个按钮，点击后可以启用、关闭Firefox浏览器的默认代理设置，可以快速在代理和非代理之间切换，很方便。界面如下图所示。另一个是AutoProxy，通过一份无法访问的域名列表目录，实现自动在代理和非代理之间切换，该域名列表目录由志愿者手动维护。AutoProxy还可实现自定义您自己的代理规则；订阅规则列表；自定义代理服务器等功能。使用AutoProxy的时候需要注意，点“代理服务器-编辑代理服务器”，将ssh -D的端口由7070修改为1080，点“代理服务器-选择代理服务器”，将ssh -D设置为默认。

* 1. Tinyproxy

tinyproxy使用主要注意事项：

apt-get update

apt-get install tinyproxy

# 编辑tinyproxy，取消Allow 10.0.0.0/8行首注释

/etc/init.d/tinyproxy start

# 设置虚拟机联网方式为NAT和端口转发，默认tinyproxy监听8888端口

# 主机浏览器设置代理指向tinyproxy的服务地址

# 虚拟机里开启wireshark抓包

# 主机访问https站点

# 结束抓包，分析抓包结果

wireshark分析HTTP代理流量技巧：

· http.request.method eq CONNECT 查看所有HTTPS代理请求

· http.request.method eq GET 查看所有HTTP GET代理请求

· 使用wireshark解密HTTPS流量的方法 方法2

· 使用wireshark提取pcap包中的SSL证书

1、wireshark首选项中确认TCP协议的Allow subdissector to reassemble TCP streams 选项处于启用状态

2、通过显示筛选过滤规则（例如：tcp.port == 443），找到SSL会话

3、通过packet list里的info列找到Certificate

在packet details面板里依次展开Handshake Protocol: Certificate --> Certificates，如果有多个证书，会看到多个默认折叠起来的Certificate

右键选中Certificate，在右键菜单里使用Export Selected Packet Bytes功能 即可导出DER格式的SSL证书

1. 使用openssl命令行工具解析DER证书 openssl x509 -in xxx.der -inform der -text -noout

1.2. buprsuite

铁器 · Burp Suite：Burp Suite 是用于攻击web 应用程序的集成平台。它包含了许多工具，并为这些工具设计了许多接口，以促进加快攻击应用程序的过程。所有的工具都共享一个能处理并显示HTTP 消息，持久性，认证，代理，日志，警报的一个强大的可扩展的框架。

1. Web应用漏洞攻防

1. 重点推荐WebGoat实验

基于WebGoat 7.0.1 Release

Authentication Flows ( 6.脆弱的访问控制 )

Forgot Password

Multi Level Login 2

Cross-Site Scripting

Stored XSS Attacks ( 4.跨站点脚本(XSS) )

Cross Site Request Forgery (CSRF) ( 11.跨站点请求伪造 )

Injection Flows ( 3. 注⼊缺陷 )

Command injection

LAB: SQL Injection

Malicious Execution

Malicious File Execution ( 13. ⽂件上传漏洞 )

Parameter Tampering ( 1. 未验证的用户输⼊ )

Bypass HTML Field Restrictions

Exploit Hidden Fields

Session Management Flows ( 7.脆弱认证和会话管理 )

Session Fixation

2. 修改WebGoat 7.0.1的默认监听IP地址

编辑.extract/webapps/WebGoat/WEB-INF/web.xml，将其中的所有localhost替换成你想要让WebGoat内置Web服务器监听的IP地址。修改后的相关行配置类似如下：

http://192.168.56.101:8080/{context-path}/servlet/{classname}

http://192.168.56.101:8080/{contextpath}/graph

http://192.168.56.101:8080/{contextpath}/saveCustomer.mvc

3. 工具推荐

burpsuite (Kali内置），免费版即可满足课程实验和大部分实战渗透测试之手工测试需求

1. sql注入，缓冲区溢出
2. select \* from users where id =1 and/or 1
3. select \* from users where id =1 and 0
4. Select \* from users where id=2 order by 4
5. Select \* from users where id=2 union select 1,2,@@version（user()） order by 1(注意字段个数，结果显示位置可不同)
6. 技巧：cheat sheet、sqlmap
7. Samy蠕虫
8. Beef
9. Console.log(document.cookie)
10. Date +%s | md5sum
11. dom操作
12. Security content automation protocol

自动化扫描工具（验证漏洞存在性）

描述性语言.

1. 分页

\*select \*from users limit 10;

1. single page application
2. 消除风险付出成本
3. php文件包含
4. 防火墙

1. 课内实验

1.1. 实验环境准备

虚拟机使用NAT方式联网，安装proftpd

使用stand alone模式启动proftpd

配置proftpd支持匿名访问（从FTP上下载配置文件）

1.2. 开始实验

虚拟机使用Host-only联网

启动proftpd和apache

实验各种防火墙规则

1.3. iptables实验清单

1.3.1. 重置iptables规则为允许所有

#!/bin/bash

IPT="/sbin/iptables"

# Flush old rules, old custom tables

$IPT --flush

$IPT --delete-chain

# Set default policies for all three default chains

$IPT -P INPUT ACCEPT

$IPT -P FORWARD ACCEPT

$IPT -P OUTPUT ACCEPT

1.3.2. 禁止指定IP访问

#!/bin/bash

# iptables script generated 2011-10-13

# http://www.mista.nu/iptables

IPT="/sbin/iptables"

# Flush old rules, old custom tables

$IPT --flush

$IPT --delete-chain

# Set default policies for all three default chains

$IPT -P INPUT DROP

$IPT -P FORWARD DROP

$IPT -P OUTPUT ACCEPT

# Enable free use of loopback interfaces

$IPT -A INPUT -i lo -j ACCEPT

$IPT -A OUTPUT -o lo -j ACCEPT

# All TCP sessions should begin with SYN

$IPT -A INPUT -p tcp ! --syn -m state --state NEW -s 0.0.0.0/0 -j DROP

# Accept inbound TCP packets

$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Block a specific IP address

$IPT -A INPUT -p IP -s 192.168.56.1 -j DROP

1.3.3. 禁止TCP / UDP指定端口访问

#!/bin/bash

# iptables script generated 2011-10-13

# http://www.mista.nu/iptables

IPT="/sbin/iptables"

# Flush old rules, old custom tables

$IPT --flush

$IPT --delete-chain

# Set default policies for all three default chains

$IPT -P INPUT DROP

$IPT -P FORWARD DROP

$IPT -P OUTPUT ACCEPT

# Enable free use of loopback interfaces

$IPT -A INPUT -i lo -j ACCEPT

$IPT -A OUTPUT -o lo -j ACCEPT

# All TCP sessions should begin with SYN

$IPT -A INPUT -p tcp ! --syn -m state --state NEW -s 0.0.0.0/0 -j DROP

# Accept inbound TCP packets

$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPT -A INPUT -p tcp --dport 22 -m state --state NEW -s 0.0.0.0/0 -j DROP

# 通过在源和目的主机(防火墙所在主机)上抓包，对比DROP和REJECT指令在防火墙响应行为上的差异

#$IPT -A INPUT -p tcp --dport 22 -m state --state NEW -s 0.0.0.0/0 -j REJECT

1.3.4. 禁止ICMP ping

#!/bin/bash

# iptables script generated 2011-10-13

# http://www.mista.nu/iptables

IPT="/sbin/iptables"

# Flush old rules, old custom tables

$IPT --flush

$IPT --delete-chain

# Set default policies for all three default chains

$IPT -P INPUT DROP

$IPT -P FORWARD DROP

$IPT -P OUTPUT ACCEPT

# Enable free use of loopback interfaces

$IPT -A INPUT -i lo -j ACCEPT

$IPT -A OUTPUT -o lo -j ACCEPT

# All TCP sessions should begin with SYN

$IPT -A INPUT -p tcp ! --syn -m state --state NEW -s 0.0.0.0/0 -j DROP

# Accept inbound TCP packets

$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# DROP inbound ICMP messages

$IPT -A INPUT -p ICMP --icmp-type 8 -s 0.0.0.0/0 -j DROP

1.3.5. 允许proftpd的远程访问

#!/bin/bash

# iptables script generated 2011-10-13

# http://www.mista.nu/iptables

IPT="/sbin/iptables"

# Flush old rules, old custom tables

$IPT --flush

$IPT --delete-chain

# Set default policies for all three default chains

$IPT -P INPUT DROP

$IPT -P FORWARD DROP

$IPT -P OUTPUT ACCEPT

# Enable free use of loopback interfaces

$IPT -A INPUT -i lo -j ACCEPT

$IPT -A OUTPUT -o lo -j ACCEPT

# All TCP sessions should begin with SYN

$IPT -A INPUT -p tcp ! --syn -m state --state NEW -s 0.0.0.0/0 -j DROP

# Accept inbound TCP packets

$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPT -A INPUT -p tcp --dport 22 -m state --state NEW -s 0.0.0.0/0 -j ACCEPT

# Allow FTP connection and transfer

$IPT -A INPUT -p tcp --dport 21 -m state --state NEW -s 0.0.0.0/0 -j ACCEPT

$IPT -A INPUT -p tcp --sport 49152:65534 --dport 49152:65534 -m state --state NEW -j ACCEPT

附：proftpd的一个示例配置文件

#

# /etc/proftpd/proftpd.conf -- This is a basic ProFTPD configuration file.

# To really apply changes reload proftpd after modifications.

#

# Includes DSO modules

Include /etc/proftpd/modules.conf

# Set off to disable IPv6 support which is annoying on IPv4 only boxes.

UseIPv6 on

# If set on you can experience a longer connection delay in many cases.

IdentLookups off

ServerName "Debian"

ServerType standalone

DeferWelcome off

MultilineRFC2228 on

DefaultServer on

ShowSymlinks on

TimeoutNoTransfer 600

TimeoutStalled 600

TimeoutIdle 1200

DisplayLogin welcome.msg

DisplayChdir .message true

ListOptions "-l"

DenyFilter \\*.\*/

# Use this to jail all users in their homes

# DefaultRoot ~

# Users require a valid shell listed in /etc/shells to login.

# Use this directive to release that constrain.

# RequireValidShell off

# Port 21 is the standard FTP port.

Port 21

# In some cases you have to specify passive ports range to by-pass

# firewall limitations. Ephemeral ports can be used for that, but

# feel free to use a more narrow range.

PassivePorts 49152 65534

# If your host was NATted, this option is useful in order to

# allow passive tranfers to work. You have to use your public

# address and opening the passive ports used on your firewall as well.

# MasqueradeAddress 1.2.3.4

# This is useful for masquerading address with dynamic IPs:

# refresh any configured MasqueradeAddress directives every 8 hours

<IfModule mod\_dynmasq.c>

# DynMasqRefresh 28800

</IfModule>

# To prevent DoS attacks, set the maximum number of child processes

# to 30. If you need to allow more than 30 concurrent connections

# at once, simply increase this value. Note that this ONLY works

# in standalone mode, in inetd mode you should use an inetd server

# that allows you to limit maximum number of processes per service

# (such as xinetd)

MaxInstances 30

# Set the user and group that the server normally runs at.

User proftpd

Group nogroup

# Umask 022 is a good standard umask to prevent new files and dirs

# (second parm) from being group and world writable.

Umask 022 022

# Normally, we want files to be overwriteable.

AllowOverwrite on

# Uncomment this if you are using NIS or LDAP via NSS to retrieve passwords:

# PersistentPasswd off

# This is required to use both PAM-based authentication and local passwords

# AuthOrder mod\_auth\_pam.c\* mod\_auth\_unix.c

# Be warned: use of this directive impacts CPU average load!

# Uncomment this if you like to see progress and transfer rate with ftpwho

# in downloads. That is not needed for uploads rates.

#

# UseSendFile off

TransferLog /var/log/proftpd/xferlog

SystemLog /var/log/proftpd/proftpd.log

<IfModule mod\_quotatab.c>

QuotaEngine off

</IfModule>

<IfModule mod\_ratio.c>

Ratios off

</IfModule>

# Delay engine reduces impact of the so-called Timing Attack described in

# http://security.lss.hr/index.php?page=details&ID=LSS-2004-10-02

# It is on by default.

<IfModule mod\_delay.c>

DelayEngine on

</IfModule>

<IfModule mod\_ctrls.c>

ControlsEngine off

ControlsMaxClients 2

ControlsLog /var/log/proftpd/controls.log

ControlsInterval 5

ControlsSocket /var/run/proftpd/proftpd.sock

</IfModule>

<IfModule mod\_ctrls\_admin.c>

AdminControlsEngine off

</IfModule>

#

# Alternative authentication frameworks

#

#Include /etc/proftpd/ldap.conf

#Include /etc/proftpd/sql.conf

#

# This is used for FTPS connections

#

#Include /etc/proftpd/tls.conf

# A basic anonymous configuration, no upload directories.

<Anonymous ~ftp>

User ftp

Group nogroup

# # We want clients to be able to login with "anonymous" as well as "ftp"

UserAlias anonymous ftp

# # Cosmetic changes, all files belongs to ftp user

DirFakeUser on ftp

DirFakeGroup on ftp

#

RequireValidShell off

#

# Limit the maximum number of anonymous logins

MaxClients 10

#

# We want 'welcome.msg' displayed at login, and '.message' displayed

# in each newly chdired directory.

DisplayLogin welcome.msg

DisplayChdir .message

# Limit WRITE everywhere in the anonymous chroot

<Directory \*>

<Limit WRITE>

DenyAll

</Limit>

</Directory>

# Uncomment this if you're brave.

# <Directory incoming>

# # Umask 022 is a good standard umask to prevent new files and dirs

# # (second parm) from being group and world writable.

# Umask 022 022

# <Limit READ WRITE>

# DenyAll

# </Limit>

# <Limit STOR>

# AllowAll

# </Limit>

# </Directory>

</Anonymous>

1.3.6. 匹配应用层指定字符串: cnssisbad

#!/bin/bash

# iptables script generated 2011-10-13

# http://www.mista.nu/iptables

IPT="/sbin/iptables"

# Flush old rules, old custom tables

$IPT --flush

$IPT --delete-chain

# Set default policies for all three default chains

$IPT -P INPUT DROP

$IPT -P FORWARD DROP

$IPT -P OUTPUT ACCEPT

# Enable free use of loopback interfaces

$IPT -A INPUT -i lo -j ACCEPT

$IPT -A OUTPUT -o lo -j ACCEPT

# All TCP sessions should begin with SYN

$IPT -A INPUT -p tcp ! --syn -m state --state NEW -s 0.0.0.0/0 -j DROP

# Block 'Bad String'

$IPT -A INPUT -p tcp -m string --algo bm --string 'cnssisbad' -j REJECT --reject-with tcp-reset

# Accept inbound TCP packets

$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPT -A INPUT -p tcp --dport 22 -m state --state NEW -s 0.0.0.0/0 -j ACCEPT

$IPT -A INPUT -p tcp --dport 80 -m state --state NEW -s 0.0.0.0/0 -j ACCEPT

# Allow FTP connection and transfer

$IPT -A INPUT -p tcp --dport 21 -m state --state NEW -s 0.0.0.0/0 -j ACCEPT

$IPT -A INPUT -p tcp --sport 49152:65534 --dport 49152:65534 -m state --state NEW -j ACCEPT

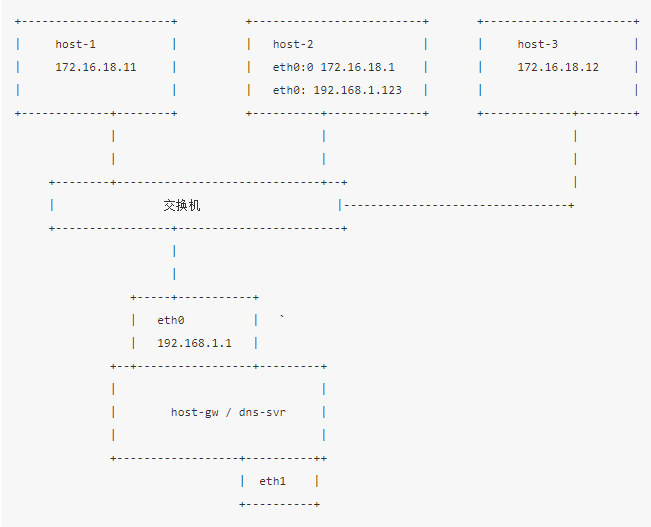
1.3.7. 记录指定模式的数据报文信息到syslog

探索：自行修改上述实验脚本中的 -j 参数为 LOG ，验证 iptables 在完成特定packet的特征匹配之后可以记录相关信息到系统日志。

2. 课外实验

2.1. 场景描述

局域网拓扑如下：



上图的补充文字说明如下：

· host-gw指的是该局域网的网关，已经配置为NAT方式，局域网内的主机host-2可以正常无障碍访问互联网；

· dns-svr指的是该局域网中的DNS解析服务器，可以正常提供域名解析服务；

交换机没有设置VLAN，所有端口正常工作；

· host-2上配置了iptables规则；

· host-1上配置了默认网关指向IP地址：172.16.18.1，域名解析服务器配置为IP：192.168.1.1

· host-3上配置了默认网关指向IP地址：172.16.18.1，域名解析服务器配置为IP：192.168.1.1

2.2. host-2上的iptables配置脚本如下：

#!/bin/bash

IPT="/sbin/iptables"

$IPT --flush

$IPT --delete-chain

$IPT -P INPUT DROP

$IPT -P FORWARD DROP

$IPT -P OUTPUT ACCEPT

$IPT -N forward\_demo

$IPT -N icmp\_demo

$IPT -A INPUT -i lo -j ACCEPT

$IPT -A OUTPUT -o lo -j ACCEPT

$IPT -A INPUT -p tcp ! --syn -m state --state NEW -s 0.0.0.0/0 -j DROP

$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPT -A INPUT -p icmp -j icmp\_demo

$IPT -A icmp\_demo -p icmp -i eth0 -j ACCEPT

$IPT -A icmp\_demo -j RETURN

$IPT -A FORWARD -j forward\_demo

$IPT -A forward\_demo -j LOG --log-prefix FORWARD\_DEMO

$IPT -A forward\_demo -p tcp --dport 80 -m string --algo bm --string 'baidu' -j DROP

$IPT -A forward\_demo -p tcp -s 172.16.18.11 -j ACCEPT

$IPT -A forward\_demo -p tcp -d 172.16.18.11 -j ACCEPT

$IPT -A forward\_demo -p udp -s 172.16.18.11 --dport 53 -j ACCEPT

$IPT -A forward\_demo -p udp -s 172.16.18.1 --dport 53 -j ACCEPT

$IPT -A forward\_demo -p udp -s 192.168.1.1 --sport 53 -j ACCEPT

$IPT -A forward\_demo -p tcp -s 172.16.18.1 -j ACCEPT

$IPT -A forward\_demo -s 172.16.18.1 -j RETURN

$IPT -t nat -A POSTROUTING -s 172.16.18.1/24 -o eth0 -j MASQUERADE

2.3. 实验任务要求

请对上述脚本逐行添加代码注释

host-1可以ping通ip: 172.16.18.1吗？

host-1可以ping通ip: 192.168.1.1吗？

host-1可以ping通域名: www.baidu.com吗？

host-1可以访问： http://www.baidu.com 吗？

host-1可以访问：http://61.135.169.121 吗？

host-3可以ping通ip: 172.16.18.1吗？

host-3可以ping通ip: 192.168.1.1吗？

host-3可以访问互联网吗？

1. 入侵检测

1. Snort

1.1. 安装

# 禁止在apt安装时弹出交互式配置界面

export DEBIAN\_FRONTEND=noninteractive

apt install snort

1.2. 实验一：配置snort为嗅探模式

# 显示IP/TCP/UDP/ICMP头

snort –v

# 显示应用层数据

snort -vd

# 显示数据链路层报文头

snort -vde

# -b 参数表示报文存储格式为 tcpdump 格式文件

# -q 静默操作，不显示版本欢迎信息和初始化信息

snort -q -v -b -i eth1 "port not 22"

# 使用 CTRL-C 退出嗅探模式

# 嗅探到的数据包会保存在 /var/log/snort/snort.log.<epoch timestamp>

# 其中<epoch timestamp>为抓包开始时间的UNIX Epoch Time格式串

# 可以通过命令 date -d @<epoch timestamp> 转换时间为人类可读格式

# exampel: date -d @1511870195 转换时间为人类可读格式

# 上述命令用tshark等价实现如下：

tshark -i eth1 -f "port not 22" -w 1\_tshark.pcap

1.3. 实验二：配置并启用snort内置规则

# /etc/snort/snort.conf 中的 HOME\_NET 和 EXTERNAL\_NET 需要正确定义

# 例如，学习实验目的，可以将上述两个变量值均设置为 any

snort -q -A console -b -i eth1 -c /etc/snort/snort.conf -l /var/log/snort/

1.4. 实验三：自定义snort规则

# 新建自定义 snort 规则文件

cat << EOF > /etc/snort/rules/cnss.rules

alert tcp $EXTERNAL\_NET any -> $HTTP\_SERVERS 80 (msg:"Access Violation has been detected on /etc/passwd ";flags: A+; content:"/etc/passwd"; nocase;sid:1000001; rev:1;)

alert tcp $EXTERNAL\_NET any -> $HTTP\_SERVERS 80 (msg:"Possible too many connections toward my http server"; threshold:type threshold, track by\_src, count 100, seconds 2; classtype:attempted-dos; sid:1000002; rev:1;)

EOF

# 添加配置代码到 /etc/snort/snort.conf

include $RULE\_PATH/cnss.rules

snort -q -A fast -b -i eth1 -c /etc/snort/snort.conf -l /var/log/snort/

1.5. 实验四：和防火墙联动

本实验需要用到的脚本代码 Guardian-1.7.tar.gz ，请下载后解压缩：

# 解压缩 Guardian-1.7.tar.gz

tar zxf guardian.tar.gz

# 安装 Guardian 的依赖 lib

apt install libperl4-corelibs-perl

本次实验需要的一个最小化的实验环境是2台主机，假设网络拓扑信息如下：

VM-1 IP: 192.168.56.101 eth1 Host-only 运行 nmap 扫描 VM-1

VM-2 IP: 192.168.56.102 eth1 Host-only 运行 snort 和 guardian.pl

开始实验之前，VM-1 和 VM-2 可以互相双向访问。

在VM-2上先后开启 snort 和 guardian.pl

# 开启 snort

snort -q -A fast -b -i eth1 -c /etc/snort/snort.conf -l /var/log/snort/

# 假设 guardian.tar.gz 解压缩后文件均在 /root/guardian 下

cd /root/guardian

编辑 guardian.conf 并保存，确认以下2个参数的配置符合主机的实际环境参数。

HostIpAddr 192.168.56.102

Interface eth1

# 启动 guardian.pl

perl guardian.pl -c guardian.conf

在VM-1上用 nmap 暴力扫描 VM-2：

nmap 192.168.56.102 -A -T4 -n -vv

guardian.conf 中默认的来源IP被屏蔽时间是 60 秒（屏蔽期间如果黑名单上的来源IP再次触发snort报警消息，则屏蔽时间会继续累加60秒）

root@KaliRolling:~/guardian# iptables -L -n

Chain INPUT (policy ACCEPT)

target prot opt source destination

REJECT tcp -- 192.168.56.101 0.0.0.0/0 reject-with tcp-reset

DROP all -- 192.168.56.101 0.0.0.0/0

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

# 1分钟后，guardian.pl 会删除刚才添加的2条 iptables 规则

root@KaliRolling:~/guardian# iptables -L -n

Chain INPUT (policy ACCEPT)

target prot opt source destination

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

2. 实验思考题

IDS与防火墙的联动防御方式相比IPS方式防御存在哪些缺陷？是否存在相比较而言的优势？

使用 Suricata 代替 Snort ， 重复本实验。

配置 Suricata 为 IPS 模式，重复 实验四 。

1. 应用程序安全加固

fail2ban实验

第十二章 计算机取证

## 1. 使用bro来完成取证分析

### 1.1. 安装bro

# apt-get install bro bro-aux

### 1.2. 实验环境基本信息

root@KaliRolling:/usr/share/bro/site# lsb\_release -a

No LSB modules are available.

Distributor ID: Kali

Description: Kali GNU/Linux Rolling

Release: kali-rolling

Codename: kali-rolling

root@KaliRolling:/usr/share/bro/site# uname -a

Linux KaliRolling 4.6.0-kali1-amd64 #1 SMP Debian 4.6.4-1kali1 (2016-07-21) x86\_64 GNU/Linux

root@KaliRolling:/usr/share/bro/site# bro -v

bro version 2.4.1

### 1.3. 编辑bro配置文件

编辑 /etc/bro/site/local.bro，在文件尾部追加两行新配置代码

@load frameworks/files/extract-all-files

@load mytuning.bro

* 在/etc/bro/site/目录下创建新文件mytuning.bro，[内容为](https://www.bro.org/documentation/faq.html" \l "why-isn-t-bro-producing-the-logs-i-expect-a-note-about-checksums" \t "https://sec.cuc.edu.cn/huangwei/textbook/ns/chap0x12/_blank)：

redef ignore\_checksums = T;

### 1.4. 使用bro自动化分析pcap文件

bro -r attack-trace.pcap /etc/bro/site/local.bro

出现警告信息WARNING: No Site::local\_nets have been defined. It's usually a good idea to define your local networks.对于本次入侵取证实验来说没有影响。

如果要解决上述警告信息，也很简单，同样是编辑mytuning.bro，增加一行变量定义即可

redef Site::local\_nets = { 192.150.11.0/24 };

注意添加和不添加上述一行变量定义除了bro运行过程中是否会产生警告信息的差异，增加这行关于本地网络IP地址范围的定义对于本次实验来说会新增2个日志文件，会报告在当前流量（数据包文件）中发现了本地网络IP和该IP关联的已知服务信息。

在attack-trace.pcap文件的当前目录下会生成一些.log文件和一个extract\_files目录，在该目录下我们会发现有一个文件。

# file extract-1240198114.648099-FTP\_DATA-FHUsSu3rWdP07eRE4l

extract-1240198114.648099-FTP\_DATA-FHUsSu3rWdP07eRE4l: PE32 executable (GUI) Intel 80386, for MS Windows

将该文件上传到[virustotal](https://virustotal.com/" \t "https://sec.cuc.edu.cn/huangwei/textbook/ns/chap0x12/_blank)我们会发现匹配了一个[历史扫描报告](https://virustotal.com/en/file/b14ccb3786af7553f7c251623499a7fe67974dde69d3dffd65733871cddf6b6d/analysis/" \t "https://sec.cuc.edu.cn/huangwei/textbook/ns/chap0x12/_blank)，该报告表明这是一个已知的后门程序！

至此，基于这个发现就可以进行逆向倒推，寻找入侵线索了。

通过阅读/usr/share/bro/base/files/extract/main.bro的源代码

if ( ! args?$extract\_filename )

args$extract\_filename = cat("extract-", f$last\_active, "-", f$source,

"-", f$id);

f$info$extracted = args$extract\_filename;

我们了解到该文件名的最右一个-右侧对应的字符串FHUsSu3rWdP07eRE4l是files.log中的文件唯一标识。

通过查看files.log，发现该文件提取自网络会话标识（bro根据IP五元组计算出的一个会话唯一性散列值）为CP0WpmULcjBpkDTQf的FTP会话。

该CP0WpmULcjBpkDTQf会话标识在conn.log中可以找到对应的IP五元组信息。

通过conn.log的会话标识匹配，我们发现该PE文件来自于IPv4地址为：98.114.205.102的主机。

### 1.5. Bro的一些其他技巧

* ftp.log 中默认不会显示捕获的FTP登录口令，我们可以通过在 /etc/bro/site/mytuning.bro 中增加以下变量重定义来实现：

redef FTP::default\_capture\_password = T;

* SMB协议识别

Bro从2.5Beta版开始增加了对SMB协议的识别支持，根据SMB协议识别插件作者在2016年9月的一次[公开技术分享PPT](https://www.bro.org/brocon2016/slides/hall_smb.pdf" \t "https://sec.cuc.edu.cn/huangwei/textbook/ns/chap0x12/_blank)的描述我们可以了解到：自2014年起开始出行并快速流行起来的勒索软件威胁，人们重新开始审视企业内部的文件共享机制，SMB作为一种常见的局域网（文件、打印机和串口等）共享技术通信协议规范，Bro2.5Beta之前并没有提供SMB协议的开箱即用识别和还原能力。

本文写作之时，如果想体验Bro的SMB协议识别功能需要使用Bro官网git主干分支上的代码在本地自行编译。好在官网的源代码编译指南写的很详细清楚，我在完成上述实验笔记的环境中对照官网的编译指南成功的编译了Bro2.5Beta。

root@kali:~/Downloads# ls -l 2.4.1

total 240

-rw-r--r-- 1 root root 189103 Dec 21 09:04 attack-trace.pcap

-rw-r--r-- 1 root root 1194 Dec 21 10:05 conn.log

drwxr-xr-x 2 root root 4096 Dec 21 10:05 extract\_files

-rw-r--r-- 1 root root 821 Dec 21 10:05 files.log

-rw-r--r-- 1 root root 849 Dec 21 10:05 ftp.log

-rw-r--r-- 1 root root 206 Dec 21 10:05 known\_hosts.log

-rw-r--r-- 1 root root 275 Dec 21 10:05 known\_services.log

-rw-r--r-- 1 root root 16505 Dec 21 10:05 loaded\_scripts.log

-rw-r--r-- 1 root root 253 Dec 21 10:05 packet\_filter.log

-rw-r--r-- 1 root root 565 Dec 21 10:05 pe.log

root@kali:~/Downloads# ls -l 2.5

total 260

-rw-r--r-- 1 root root 189103 Dec 20 21:15 attack-trace.pcap

-rw-r--r-- 1 root root 278 Dec 21 09:08 capture\_loss.log

-rw-r--r-- 1 root root 1197 Dec 21 09:08 conn.log

-rw-r--r-- 1 root root 470 Dec 21 09:08 dpd.log

drwxr-xr-x 2 root root 4096 Dec 21 09:07 extract\_files

-rw-r--r-- 1 root root 821 Dec 21 09:08 files.log

-rw-r--r-- 1 root root 849 Dec 21 09:08 ftp.log

-rw-r--r-- 1 root root 206 Dec 21 09:08 known\_hosts.log

-rw-r--r-- 1 root root 276 Dec 21 09:08 known\_services.log

-rw-r--r-- 1 root root 23552 Dec 21 09:08 loaded\_scripts.log

-rw-r--r-- 1 root root 397 Dec 21 09:08 ntlm.log

-rw-r--r-- 1 root root 253 Dec 21 09:08 packet\_filter.log

-rw-r--r-- 1 root root 565 Dec 21 09:08 pe.log

-rw-r--r-- 1 root root 705 Dec 21 09:08 stats.log

使用Bro 2.4.1和2.5Beta使用相同的检测规则（2.5Beta启用了“@load protocols/smb）的情况下生成的日志文件如上所示。

对比 known\_services.log 可以看到2.5Beta可以很好的识别出SMB协议流量了。

root@kali:~/Downloads# cat 2.4.1/known\_services.log#separator \x09#set\_separator ,#empty\_field (empty)#unset\_field -#path known\_services#open 2016-12-21-10-05-56#fields ts host port\_num port\_proto service#types time addr port enum set[string]

1240198118.462473 192.150.11.111 445 tcp (empty)#close 2016-12-21-10-05-56

root@kali:~/Downloads# cat 2.5/known\_services.log#separator \x09#set\_separator ,#empty\_field (empty)#unset\_field -#path known\_services#open 2016-12-21-09-08-22#fields ts host port\_num port\_proto service#types time addr port enum set[string]

1240198108.976883 192.150.11.111 445 tcp SMB,NTLM#close 2016-12-21-09-08-22

* 使用正确的分隔符进行过滤显示的重要性

# 从头开始查看日志文件，显示前1行

head -n1 conn.log

# Bro的日志文件默认使用的分隔符显示为ASCII码\x09，通过以下命令可以查看该ASCII码对应的“可打印字符”echo -n -e '\x09' | hexdump -c

# 使用awk打印给定日志文件的第N列数据

awk -F '\t' '{print $12}' conn.log

* 查看Bro的超长行日志时的横向滚动技巧

less -S conn.log

* 使用bro-cut（在bro-aux软件包中）更“优雅”的查看日志中关注的数据列

# 查看conn.log中所有可用的“列名”

grep ^#fields conn.log | tr '\t' '\n'

# 按照“列名”输出conn.log中我们关注的一些“列”

bro-cut ts id.orig\_h id.orig\_p id.resp\_h id\_resp\_p proto < conn.log

# 将UNIX时间戳格式转换成人类可读的时间（但该方法对于大日志文件处理性能非常低）

bro-cut -d < conn.log