THM-Protocols and Servers 2(协议和服务器2)-学习

本文相关的TryHackMe实验房间链接： https://tryhackme.com/room/protocolsandservers2

通过学习相关知识点：了解针对密码和明文流量的攻击；探索通过 SSH 和 SSL/TLS 缓解攻击的选项。

H2 介绍

在前一篇文章中，简单介绍了协议和服务器的相关知识点，涵盖了许多协议：

- Telnet
- HTTP
- FTP
- SMTP
- POP3
- IMAP

实现这些协议的服务器会受到不同类型的攻击，在此仅举几例：

1. 嗅探攻击（网络数据包捕获）
2. 中间人 (MITM：Man-in-the-Middle) 攻击
3. 密码攻击（身份验证攻击）
4. 漏洞攻击

从安全的角度来看，我们总是需要考虑我们的目标是保护什么； 通常我们会考虑安全三元组：机密性、完整性和可用性 (CIA)。

机密性（Confidentiality）是指保持通信内容对预期方可访问，防止未授权的用户访问数据，简单来说就是“不能看”；完整性（Integrity）是指确保发送的任何数据在到达目的地时都是准确、一致和完整的，防止未授权的修改数据，也就是“不能改”；最后，可用性（availability ）是指能够在我们需要时访问服务，保证经过授权的客户能及时准确的不间断的访问数据，也就是“一直用”。
不同的组织会对这三个方面给予不同的重视。例如，机密性将是情报机构的最高优先事项；网上银行将最重视交易的完整性；而对于任何通过投放广告赚钱的平台来说，可用性都是最重要的。

知道我们正在保护机密性、完整性和可用性 (CIA)，攻击者则旨在导致泄露、变更和破坏 (DAD)，下图很好地反映了这一点。

这些攻击（DAD）将直接影响系统的安全。例如，网络数据包捕获违反了安全的机密性并会导致信息泄露，成功的密码攻击也可能导致泄露；另一方面，中间人 (MITM) 攻击会破坏系统的完整性，因为它可以更改通信数据。我们将在本文中主要关注三种攻击：网络数据包捕获、中间人攻击、密码攻击。防范这些攻击是协议设计和服务器实现不可或缺的一部分。

漏洞同样危及系统安全，但漏洞的范围更广，被利用的漏洞对目标系统有不同的影响。例如，利用拒绝服务 (DoS) 漏洞可能会影响系统的可用性，而利用远程代码执行 (RCE) 漏洞可能会导致更严重的损害。需要注意的是，漏洞本身会产生风险，但是只有当漏洞被利用时才会发生损害，在本文中我们不会过多讨论关于漏洞的知识点。

本文将重点讨论如何升级或替换协议以防止数据的泄露和更改，即保护传输数据的机密性和完整性。 此外，我们还简单引入了如何使用Hydra来破解弱密码。

建议在继续处理以下知识点时启动TryHackMe相关房间中的AttackBox和目标虚拟机，你可以通过 Telnet 或 Netcat 连接到不同协议所相关的服务，以获得更好的练习和学习体验。

H2 嗅探攻击

嗅探攻击是指使用网络数据包捕获工具收集有关目标的信息。当协议以明文形式进行通信时，正在交换的数据就可以被第三方捕获以进行分析；也就是说：如果数据在传输过程中未加密，则进行 简单的网络数据包捕获操作 就可以获取到目标相关信息，例如目标的私人消息内容以及登录凭据（用户名和密码）。

只要用户具有适当的权限（Linux 上的 root 权限或者MS Windows 上的管理员权限），就可以利用以太网 (802.3) 网卡进行嗅探攻击。 有许多程序或者工具都可用于捕获网络数据包，我们考虑以下几个：

1. Tcpdump 一个免费的开源命令行界面 (CLI) 程序，已移植到许多操作系统上。
2. Wireshark 一个免费的开源图形用户界面 (GUI) 程序，可用于多种操作系统，包括 Linux、macOS 和 MS Windows。
3. Tshark 是 Wireshark 的 CLI 替代品。

虽然有几种专门的工具可用于捕获密码甚至完整的消息，但是，类似的功能仍然可以通过使用Tcpdump 和 Wireshark 进行一些额外的操作来实现。

假设一个用户使用 POP3 检查他的电子邮件，我们就可以使用 Tcpdump 尝试捕获这个用户的用户名和密码。在下面的终端输出中，我们使用了命令 `sudo tcpdump port 110 -A`，在解释这个命令之前，我们应该知道实现这个攻击需要访问网络流量：例如，通过窃听工具或带有端口镜像的交换机访问网络流量，或者，如果我们发起成功的中间人 (MITM) 攻击，我们也可以访问正在交换的网络流量。

我们需要在命令开头添加 `sudo`，因为数据包捕获操作需要 root 权限；我们知道 POP3 使用端口 110，所以我们使用端口 `110` 过滤数据包（这样可以限制捕获的数据包数量并显示与 POP3 服务器交换的数据包）；最后，我们想以 ASCII 格式显示捕获的数据包的内容，所以我们需要添加 `-A`。

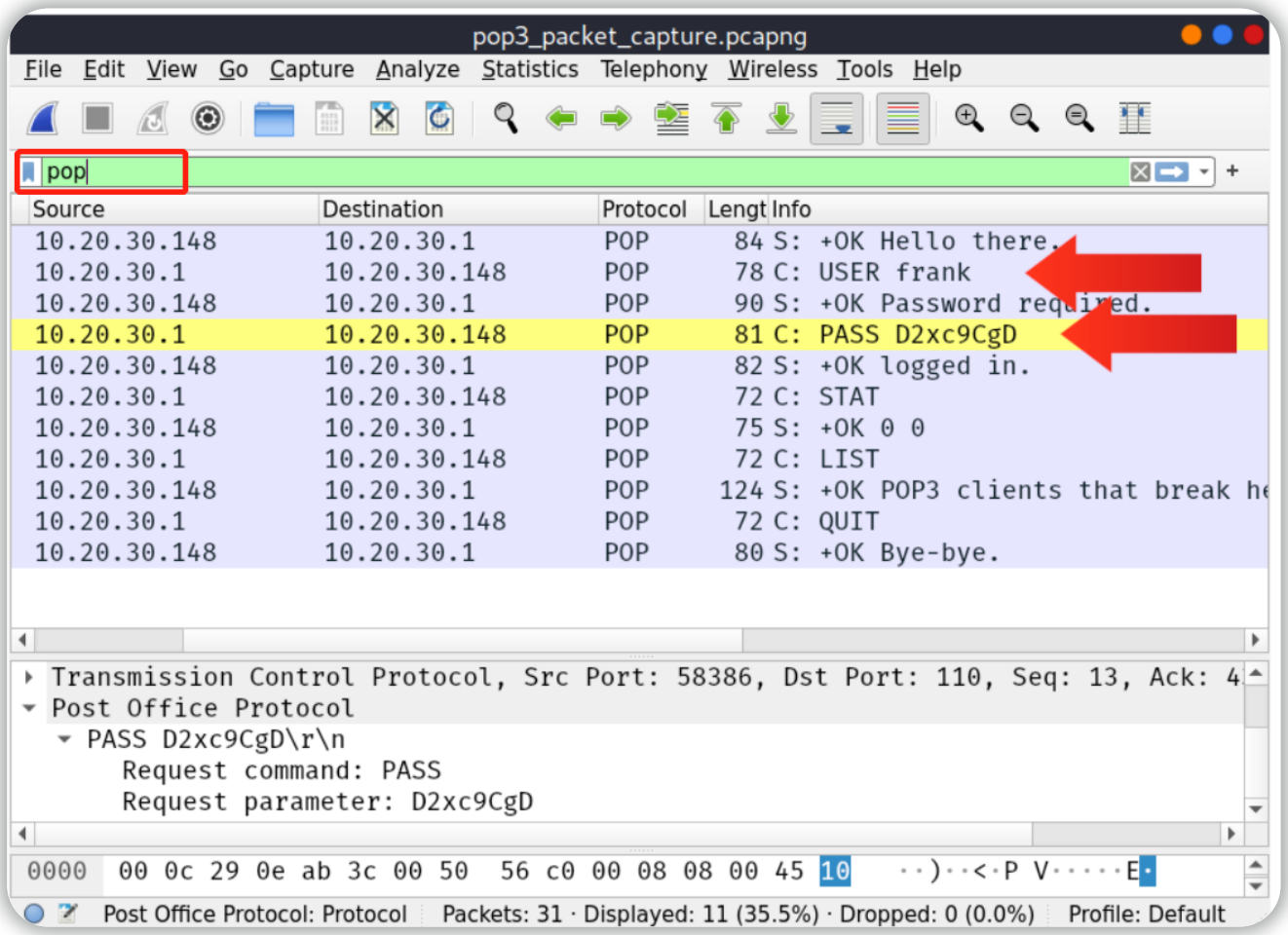
```
pentester@TryHackMe$ sudo tcpdump port 110 -A
[...]
09:05:15.132861 IP 10.20.30.1.58386 > 10.20.30.148.pop3: Flags [P.], seq 1:13, ack
19, win 502, options [nop,nop,TS val 423360697 ecr 3958275530], length 12
E..@.V@.@.g.
...
......n......".............
.;....}.USER frank

09:05:15.133465 IP 10.20.30.148.pop3 > 10.20.30.1.58386: Flags [.], ack 13, win 510,
options [nop,nop,TS val 3958280553 ecr 423360697], length 0
E..4..@.@.O~
...
....n...."..........?P.....
...i.;..
09:05:15.133610 IP 10.20.30.148.pop3 > 10.20.30.1.58386: Flags [P.], seq 19:43, ack
13, win 510, options [nop,nop,TS val 3958280553 ecr 423360697], length 24
E..L..@.@.Oe
...
....n...."..........<-.....
...i.;..+OK Password required.

09:05:15.133660 IP 10.20.30.1.58386 > 10.20.30.148.pop3: Flags [.], ack 43, win 502,
options [nop,nop,TS val 423360698 ecr 3958280553], length 0
E..4.W@.@.g.
...
......n......".....??.....
.;.....i
09:05:22.852695 IP 10.20.30.1.58386 > 10.20.30.148.pop3: Flags [P.], seq 13:28, ack
43, win 502, options [nop,nop,TS val 423368417 ecr 3958280553], length 15
E..C.X@.@.g.
...
......n......".....6......
.<.....iPASS D2xc9CgD
[...]
```

在上面的终端输出中，我们删除了不重要的数据包，以帮助你更好地关注重要的数据包。 用户名和密码都是在各自的数据包中发送的：第一个数据包明确显示了用户名信息“ `USER frank` ”，而最后一个数据包则显示了密码信息“ `PASS D2xc9CgD` ”。

我们也可以使用 Wireshark 来达到同样的效果。 在下面的 Wireshark 窗口中，可以看到我们在过滤字段中输入了 `pop` ，现在我们已经过滤出了我们感兴趣的流量数据，我们可以看到用户名信息和密码信息被成功捕获。



简而言之，任何使用明文通信的协议都容易受到嗅探攻击，这种攻击成功的唯一要求是能够访问两个通信系统之间的系统。

面对嗅探攻击，有效的缓解措施是在任何网络协议之上再添加一个加密层。在实际应用中，传输层安全协议 (TLS-Transport Layer Security) 已被添加到 HTTP协议、FTP协议、SMTP协议、POP3协议、IMAP协议和许多其他协议中；对于远程访问服务，可以使用安全的替代方案 Secure Shell (SSH)协议来取代 Telnet协议。

答题

回答以下问题

您需要在命令中添加什么才能 `sudo tcpdump` 仅捕获 Telnet 流量？

port 23 正确答案

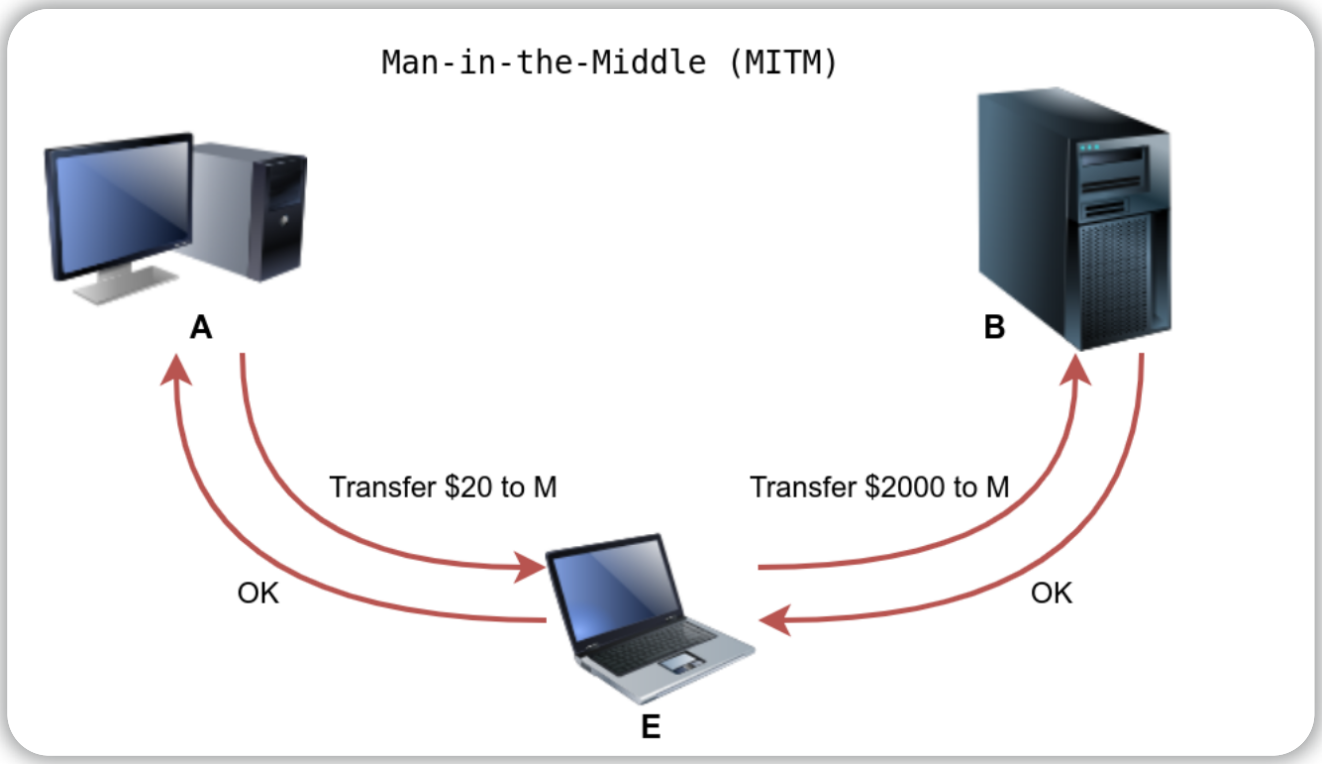
您可以与 Wireshark 一起使用以仅显示 IMAP 流量的最简单的显示过滤是什么？

imap 正确答案

H2 中间人(MITM：Man-in-the-Middle)攻击

当受害者 (A) 认为他们正在与合法目的地 (B) 进行通信但实际上却是在不知不觉中与攻击者 (E) 发生通信时，就会发生中间人 (MITM) 攻击。

在下图中，我们可以看到 A 请求向 M 转账 20 美元； 但是，E 更改了此条消息并用新值替换了原始值；最终 B 收到修改后的消息并采取了相应的行动。

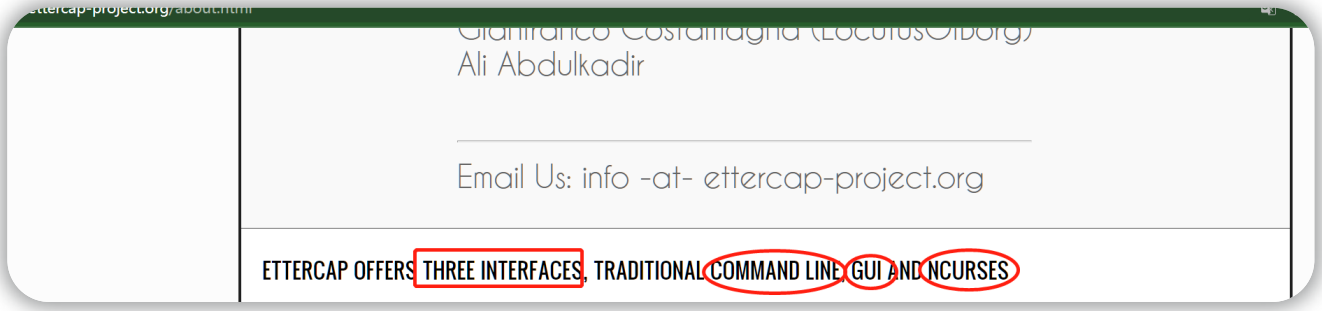


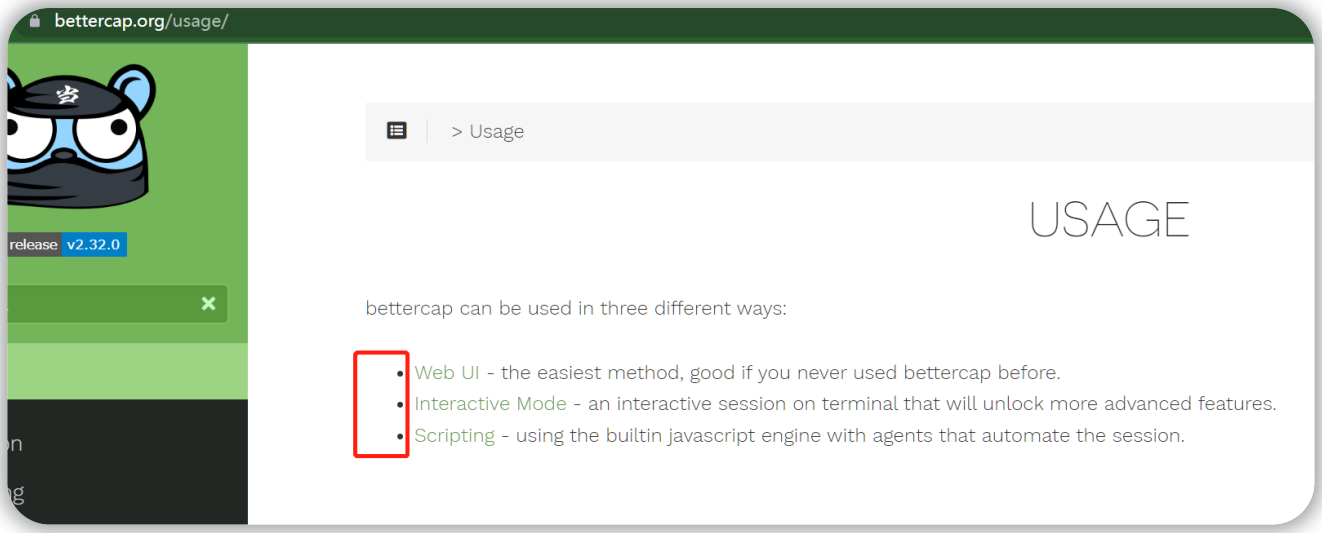
如果双方不确认每条消息的真实性和完整性，则这种攻击的实现就会相对简单，在某些情况下，正在使用的所选协议可能并不提供安全身份验证或完整性检查；此外，一些协议具有固有的不安全性，使它们更容易受到中间人攻击影响。

每当你通过 HTTP 浏览时，都容易受到 MITM 攻击，可怕的是你可能无法察觉到这种攻击；许多工具可以帮助你执行此类攻击，例如 Ettercap 和 Bettercap 。

MITM 还可以影响其他明文协议，例如 FTP、SMTP 和 POP3等。缓解这种攻击需要使用密码学，具体的解决方案在于使用适当的身份验证机制以及交换消息时进行加密或签名操作，在公钥基础设施 (PKI) 和受信任的根证书的帮助下，传输层安全协议 (TLS-Transport Layer Security) 可以有效防止 MITM 攻击。

答题





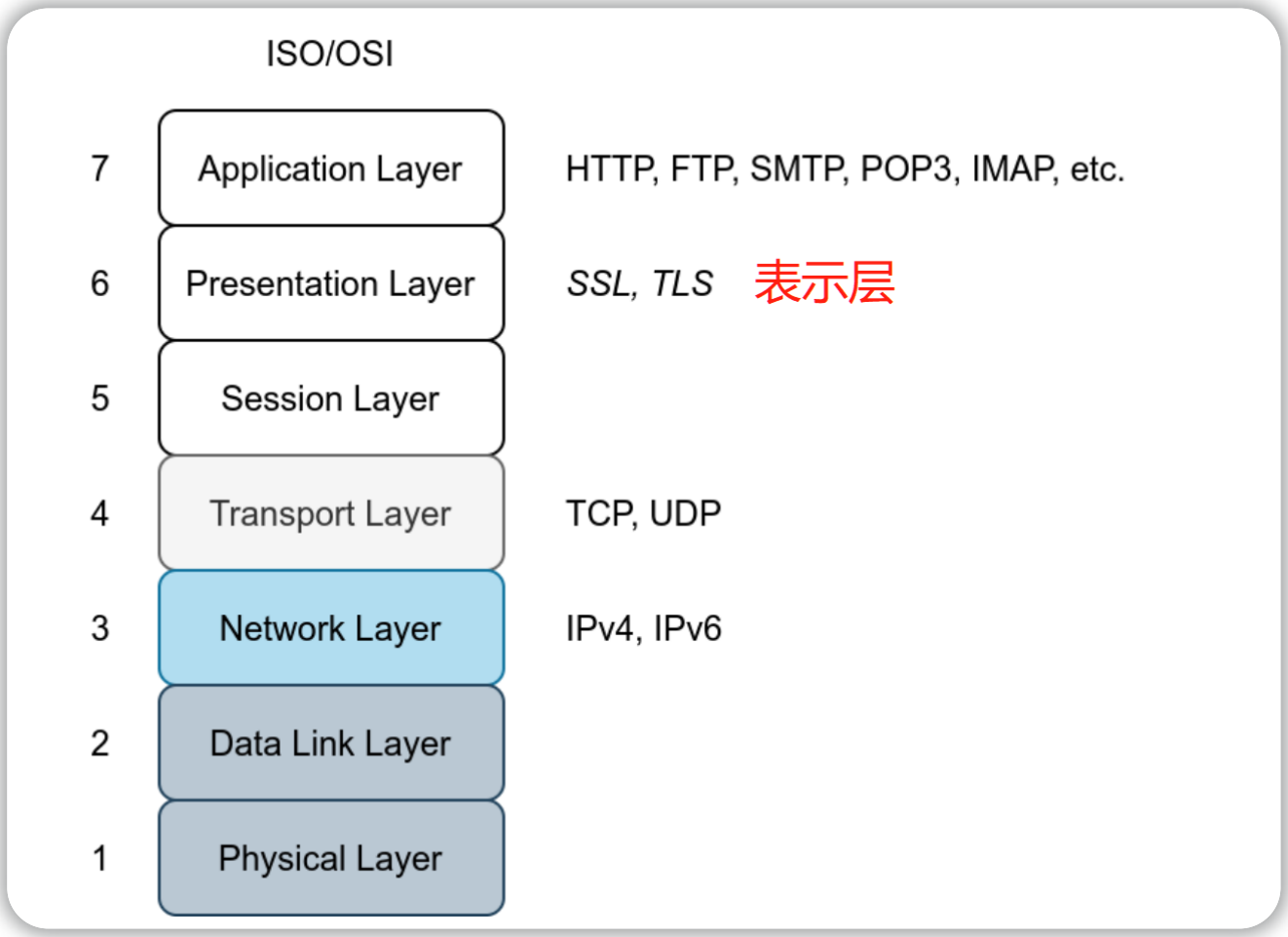
> https://www.ettercap-project.org/about.html
>
> https://www.bettercap.org/usage/

H2 传输层安全协议 (TLS)

在本小节，我们将了解一种标准解决方案来保护正在交换的数据包的机密性和完整性，以下方法可以有效防止密码嗅探攻击和中间人（MITM） 攻击。

SSL（安全套接层-Secure Sockets Layer）始于万维网出现新的应用程序，例如在线购物、需要发送支付信息等。Netscape 在 1994 年引入 SSL，并在 1996 年发布 SSL 3.0，出于更多安全性需求，在1999 年又引入了 TLS（传输层安全-Transport Layer Security）协议。在我们解释 TLS 和 SSL 提供什么功能之前，让我们看看它们是如何适应计算机网络模型的。

到目前为止，我们介绍的通用协议都是以明文形式发送数据，这使得任何有权访问网络的人都可以捕获、保存和分析正在交换的消息。下图显示了 ISO/OSI 网络模型，到目前为止，我们所讨论的协议都在应用层。在ISO/OSI 模型中，我们可以通过表示层来为我们的协议添加加密，这样处理之后，数据将以加密格式（密文）而不是其原始形式呈现。



由于 SSL 和 TLS 之间的密切关系，因此可以使用一个来代替另一个；但是，TLS 比 SSL 更安全，TLS实际上已经取代了 SSL。 我们本可以放弃 SSL，只写 TLS 而不是 SSL/TLS，但我们将继续提及这两者以避免任何歧义，因为 SSL 一词仍在广泛使用。 在现实场景下，我们可以期望所有现代服务器都在使用 TLS协议。

通过使用 SSL/TLS 加密可以升级现有的明文协议，比如：我们可以使用 TLS 来升级 HTTP、FTP、SMTP、POP3 和 IMAP 等等明文协议。 下表列出了我们在通过 SSL/TLS 进行加密升级前后所涵盖的协议及其默认端口。 该列表并不详尽，此列表的目的是帮助我们更好地理解明文协议的加密升级过程。

Protocol	Default Port	Secured Protocol	Default Port with TLS
HTTP	80	HTTPS	443
FTP	21	FTPS	990
SMTP	25	SMTPS	465
POP3	110	POP3S	995
IMAP	143	IMAPS	993

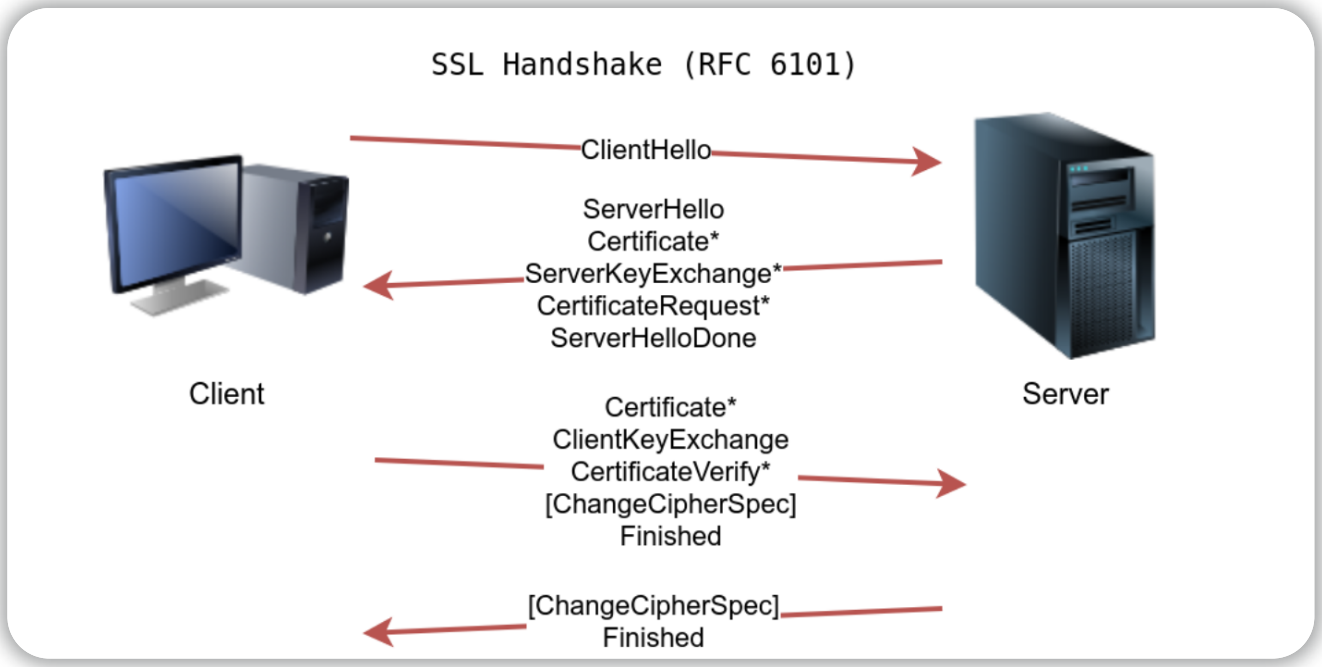
考虑HTTP协议进行加密升级的情况，要通过 HTTP 协议检索网页，Web 浏览器至少需要执行以下两个步骤：

1. 与远程 Web 服务器建立 TCP 连接
2. 向 Web 服务器发送 HTTP 请求，例如 GET 和 POST 请求。

HTTPS协议需要额外的步骤来加密流量，新步骤发生在建立 TCP 连接之后和发送 HTTP 请求之前。 这个额外的步骤可以从前面展示的图像中的 ISO/OSI 模型中推断出来。 HTTPS 至少需要以下三个步骤：

1. 建立 TCP 连接
2. 建立 SSL/TLS 连接
3. 向网络服务器发送 HTTP 请求

要建立 SSL/TLS 连接，客户端需要与服务器执行正确的握手。 握手是基于 RFC 6101的，SSL的连接建立如下图所示。



与服务器建立TCP连接后，客户端还需要建立SSL/TLS连接，如上图所示。这些术语可能看起来很复杂，但我们可以将上面四个步骤简化为：

1. 客户端向服务器发送 ClientHello 以指示其功能，例如支持的算法。
2. 服务器以 ServerHello 响应，指示所选的连接参数。如果需要服务器身份验证，服务器则会提供其证书，证书是用于识别自己的数字文件，通常由第三方进行数字签名；此外，它可能会在其 ServerKeyExchange 消息中发送生成主密钥所需的附加信息，然后再发送 ServerHelloDone 消息以指示协商已完成。
3. 客户端以 ClientKeyExchange 响应，其中会包含生成主密钥所需的附加信息；此外，它会切换到加密方式并通过ChangeCipherSpec消息通知服务器。
4. 服务器也切换到加密方式，并通过ChangeCipherSpec消息通知客户端。

如果这听起来仍然很复杂，请不要担心，我们只需要了解它的要点即可：客户端能够与具有公共证书的服务器就密钥达成一致，此密钥是安全生成的，因此监控通道的第三方将无法发现它；客户端和服务器之间的进一步通信将使用生成的密钥进行加密。

因此，一旦建立了 SSL/TLS 握手，任何监控通信通道的他人都无法访问 HTTP 请求和发生交换的数据。

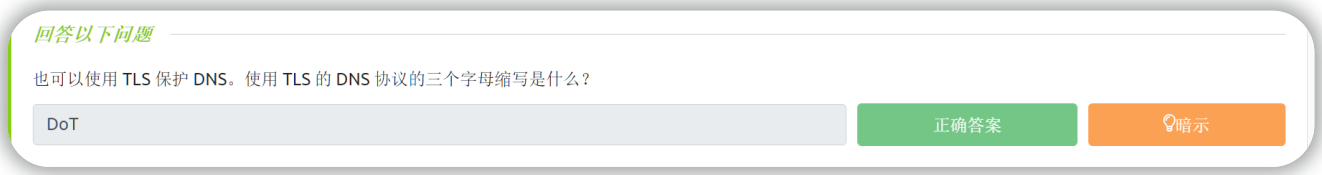
最后一点，为了使 SSL/TLS 有效，尤其是在通过 HTTPS 浏览网页时，我们需要依赖于由我们系统所信任的证书颁发机构签署的公共证书。换句话说，当我们通过 HTTPS 浏览 TryHackMe 时，我们的浏览器期望 TryHackMe Web 服务器提供来自受信任证书颁发机构的签名证书，如下例所示；这样，我们的浏览器就可以确保它正在与正确的服务器进行通信，而尽量避免了中间人（MITM）攻击的发生。

在上图中，我们可以看到以下信息：

1. 证书发给谁？ 指的是将要使用此证书的公司的名称。
2. 谁颁发的证书？ 指的是颁发此证书的证书颁发机构。
3. 证书的有效期，避免使用已过期的证书。

我们不必为我们访问的每个站点手动检查证书，我们的web浏览器会为我们做好这件事。 依赖于服务器所提供的证书，web浏览器将能够确保我们正在与正确的服务器进行通信并能尽量保护我们的通信安全。

答题



H2 安全外壳协议 (SSH)

安全外壳 (SSH) 的目的是为远程系统管理提供一种更加安全的方式；换句话说，它允许你通过网络安全地连接到另一个系统并在远程系统上执行命令。SSH中的“S”代表secure，SSH的特点可以简单概括为：

1. 可以确认远程服务器的身份；
2. 交换的消息是加密的，只能由预期的收件人解密；
3. 双方都可以检测到消息中的任何修改。

以上三点是由密码学保证的，在更专业的术语中，它们是安全三元组中的机密性和完整性的一部分，可以通过正确使用不同的加密算法来实现。

要使用 SSH，你首先需要一个 SSH 服务器和一个 SSH 客户端；SSH 服务器默认监听 22 端口，SSH 客户端可以使用以下方式进行身份验证：

- 用户名和密码
- 私钥和公钥（需要SSH 服务器被配置为识别相应的公钥，然后才能识别到公钥）

在2018 年之后的Linux、macOS 和 MS Windows 版本上，你可以使用以下命令连接到 SSH 服务器 `ssh username@MACHINE_IP` ；该命令将尝试使用登录名 `username` 连接到 IP 地址 `MACHINE_IP` 所对应的服务器。如果 SSH 服务器正在侦听默认端口，它会要求你提供 `username` 所对应的密码（ `password` ）。

一旦成功通过身份验证，用户将可以访问目标服务器的终端，下面的终端输出是使用 SSH 访问 Debian Linux服务器的示例。

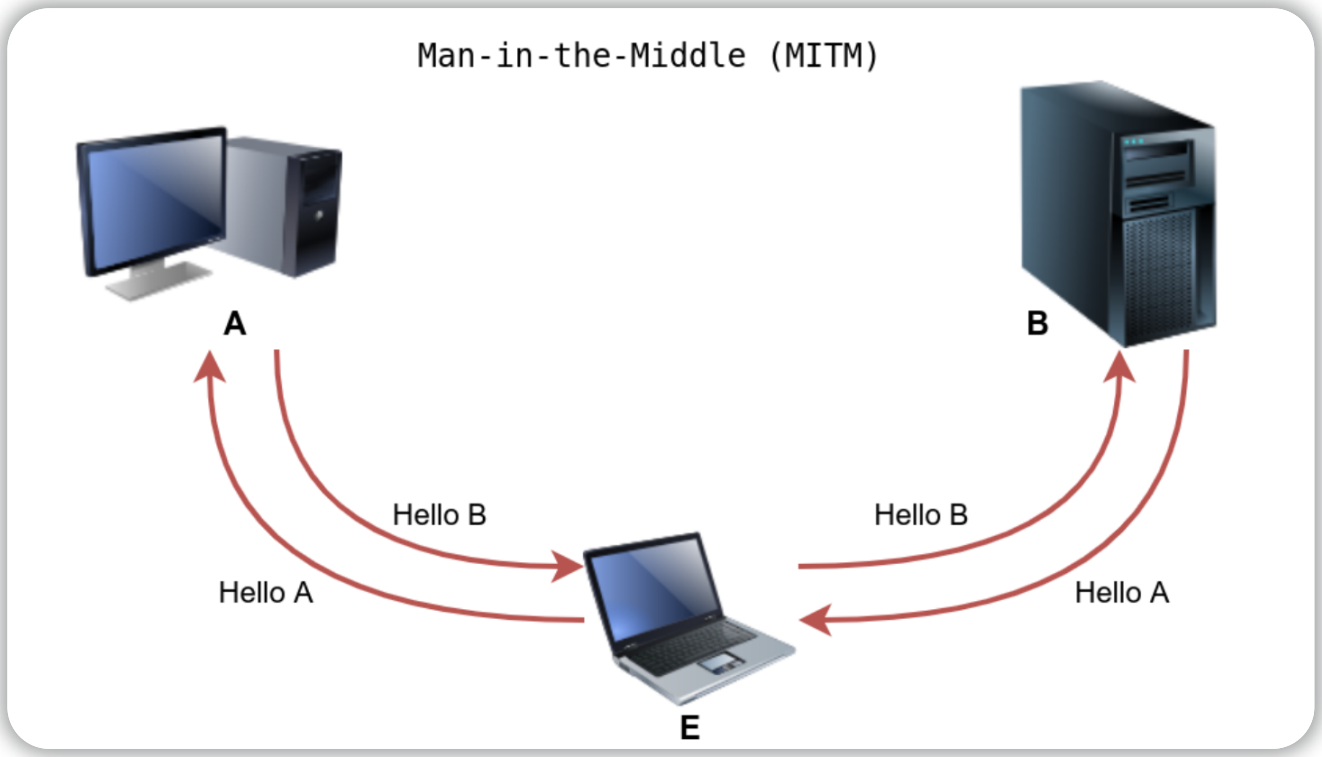
```
user@TryHackMe$ ssh mark@MACHINE_IP
mark@MACHINE_IP's password: XBtc49AB

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Sep 20 13:53:17 2021
mark@debian8:~$
```

在上面的示例中，我们先输入命令 `ssh mark@MACHINE_IP` ，一旦我们输入了正确的密码，我们就可以访问远程系统的终端。SSH协议用于远程管理比较安全可靠，因为我们的用户名和密码都是通过加密发送的；此外，我们在远程系统上执行的所有命令也都将通过加密通道进行发送。

请注意，如果这是我们第一次连接到这个系统，我们需要确认 SSH 服务器公钥的指纹，以避免中间人 (MITM) 攻击。如前所述，MITM 发生在恶意方 E 位于 A 和 B 之间，并伪装成B与 A 进行通信，伪装成A与 B 进行通信，这样 A 和 B 就会认为他们正在直接与彼此进行通信。在使用SSH的情况下，我们通常没有第三方来检查公钥是否有效，因此我们需要手动执行相关操作。



我们可以使用基于 SSH 协议的 SCP（安全复制协议）来通过 SSH连接 进行文件传输操作。

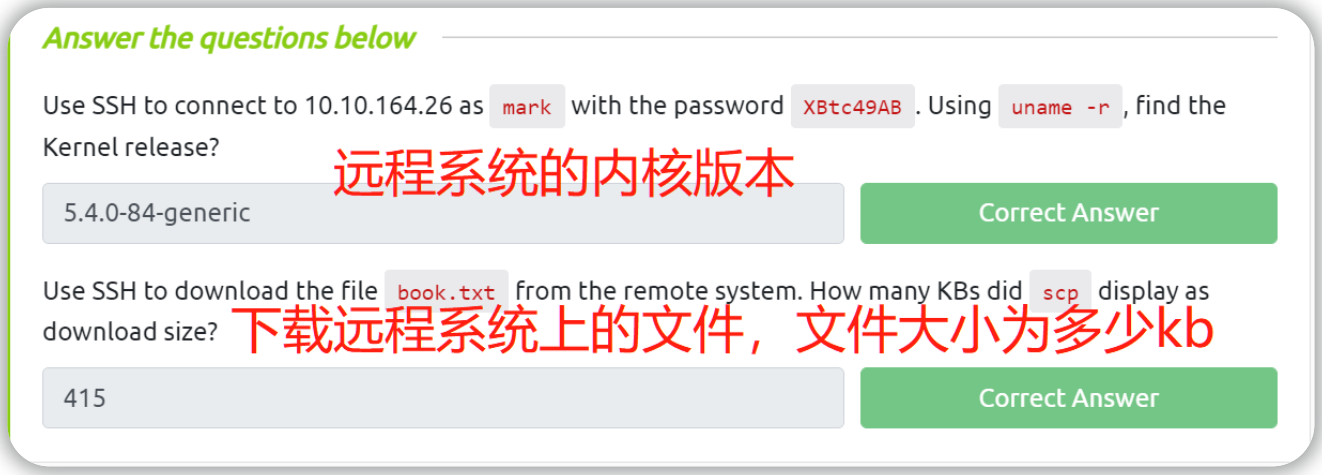
SCP的语法示例是 `scp mark@MACHINE_IP:/home/mark/archive.tar.gz ~` ，该命令将复制远程系统上 位于 `/home/mark` 目录下的 `archive.tar.gz` 文件到本地机的 `~` 目录下， `~` 代表本地机当前登录用户的用户目录(如：home/user1或者/root)。

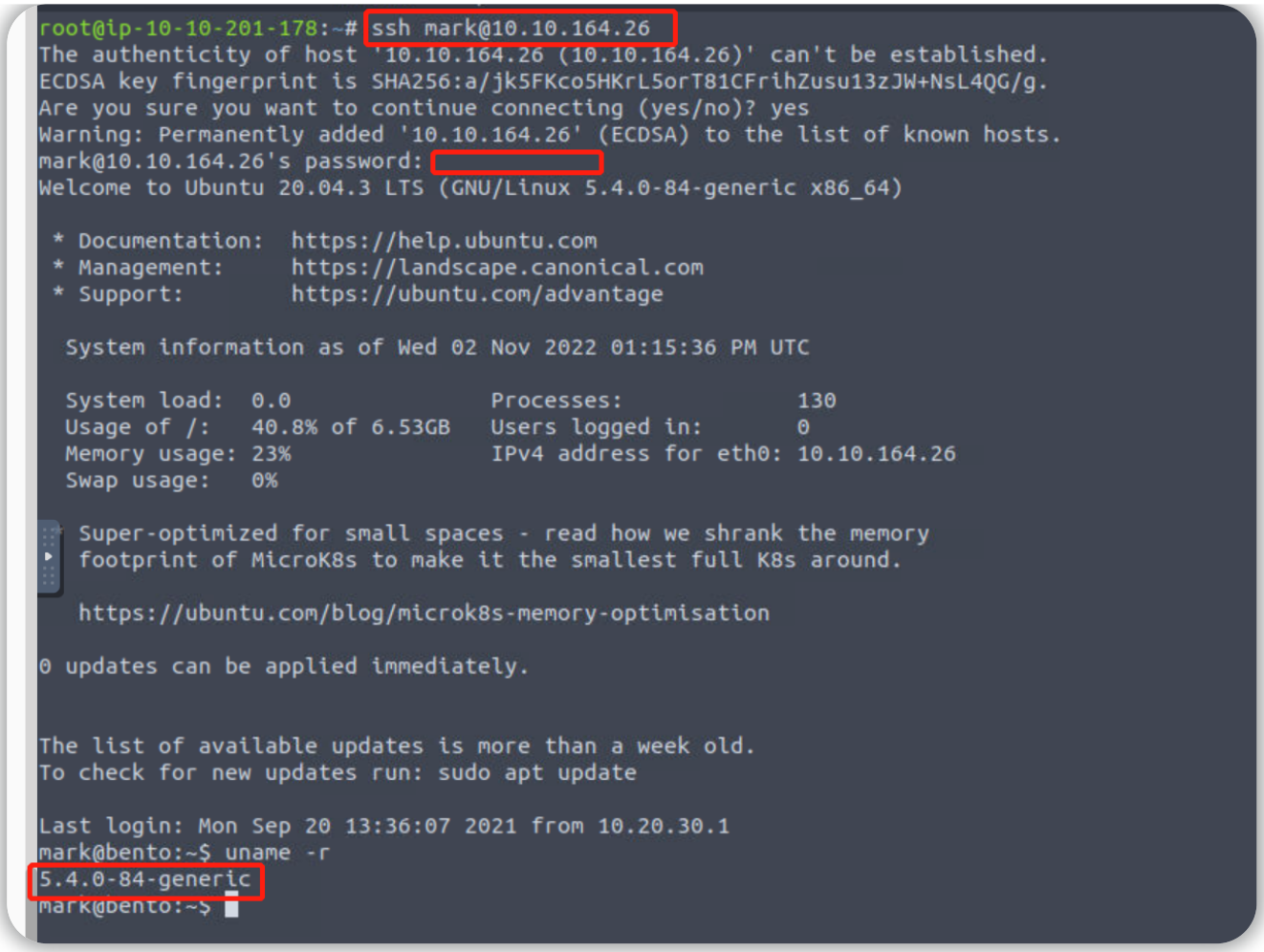
SCP的另一个示例语法是 `scp backup.tar.bz2 mark@MACHINE_IP:/home/mark/` ，此命令会将文件 `backup.tar.bz2` 从本地系统复制到远程系统上的 `/home/mark/` 目录下。

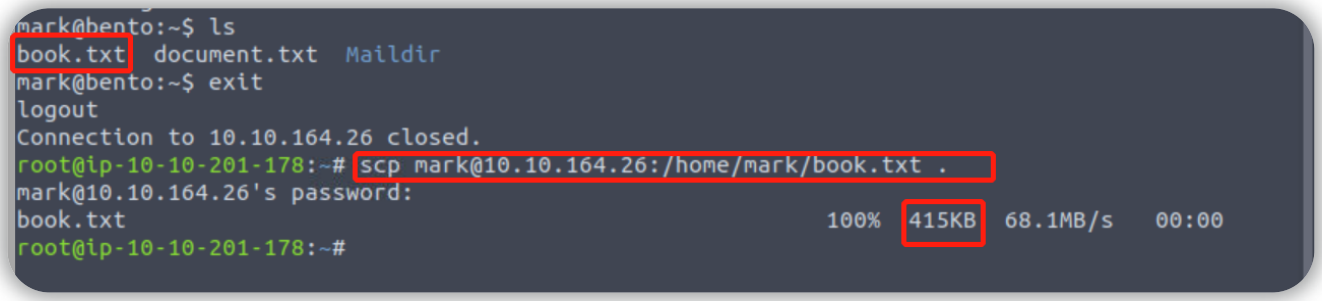
```
user@TryHackMe$ scp document.txt mark@MACHINE_IP:/home/mark
mark@MACHINE_IP's password:
document.txt                                  100% 1997KB  70.4MB/s   00:00
```

FTP可以通过使用SSL/TLS 进行安全加密，加密之后的协议为FTPS（端口990）；值得一提的是，FTP也可以使用SSH协议来保护，即SFTP协议，默认情况下，此服务会侦听端口22，SSH也是默认侦听端口22。

答题







H2 密码攻击

我们讨论了网络数据包捕获和MITM（中间人）攻击，以及如何使用 TLS 和 SSH 缓解这些攻击。我们将在本小节介绍第三种攻击：密码攻击。

许多协议在使用时，会要求你进行身份验证。身份验证是为了证明你自己是谁，当我们使用诸如 POP3 之类的协议时，我们不应该在验证我们的身份之前就获得对邮箱的访问权限。下面是一个POP3协议相关示例，在此示例中，我们使用用户名 frank，然后提供正确的用户密码，成功通过了 服务器对我们进行的身份验证。由此可知：密码是通过身份验证的一种重要方式。

```
pentester@TryHackMe$ telnet MACHINE_IP 110
Trying MACHINE_IP ...
```

```
Connected to MACHINE_IP.
Escape character is '^]'.
+OK MACHINE_IP Mail Server POP3 Wed, 15 Sep 2021 11:05:34 +0300
USER frank
+OK frank
PASS D2xc9CgD
+OK 1 messages (179) octets
STAT
+OK 1 179
LIST
+OK 1 messages (179) octets
1 179
.
RETR 1
+OK
From: Mail Server
To: Frank
subject: Sending email with Telnet
Hello Frank,
I am just writing to say hi!
.
QUIT
+OK MACHINE_IP closing connection
Connection closed by foreign host.
```

身份验证可以通过以下方式之一或以下任意两者的组合来实现：

1. 你所知道的信息，例如密码和 PIN 码。
2. 你所拥有的东西，例如 SIM 卡、RFID 卡和 USB dongle （USB加密狗）。
3. 你本身的独特特征，例如指纹和虹膜。

本小节将集中于对密码进行攻击，即目标所知道的验证信息。如果你使用 Telnet、SSH、POP3 和 IMAP 等协议访问相关服务器并尝试进行通信，总是需要密码才能获得访问权限。根据 2013 年 Adobe 漏洞泄露的 1.5 亿用户名和密码，排名前十的密码是：

- 123456
- 123456789
- password
- adobe123
- 12345678
- qwerty
- 1234567
- 111111
- photoshop

- 123123

我们可以看到只有两个密码与 Adobe 及其产品相关，其余的都是比较通用的弱密码；这种情况可能在过去十年中又发生了新的变化，但是，123456、1234567、12345678 和 123456789 仍然是许多用户的常见密码选择；而且很多人可能还没有意识到 qwerty 的不安全性，它仍然被许多人用作密码。

针对密码的攻击通常通过以下方式进行：

1. 密码猜测：猜测密码需要对目标有一些了解，例如用户所养宠物的名字和用户的出生年份。
2. 密码字典攻击：这种方法扩展了密码猜测的范围，并会尝试将所有有效词汇包含在密码字典或密码列表中。
3. 暴力破解：这种攻击是最费时费力的攻击，攻击者会尝试所有可能的字符组合，组合的种类将快速增长（随着字符数量呈指数增长）。

我们可以关注密码字典攻击，随着时间的推移，黑客们已经编制了一个又一个密码列表，其中包含很多因数据泄露而暴露的密码。一个例子是RockYou数据泄露事件中暴露的密码列表，你可以在 AttackBox 上找到该密码列表 `/usr/share/wordlists/rockyou.txt` 。密码列表的选择还应该取决于你对不同目标的了解程度，例如，法语用户可能会使用法语单词作为密码而不是英语单词。

如果我们想要一种自动化的方式来尝试使用常用密码或密码列表中的条目以便进行密码攻击，我们可以使用 Hydra 工具。Hydra 支持多种协议，包括FTP协议、POP3协议、IMAP协议、SMTP协议、SSH协议以及所有与 HTTP 相关的方法。

Hydra的一般语法是 `hydra -l username -P wordlist.txt server service` ，每个参数的含义是：

- `-l username` ： `-l` 应该在 `username` 之前，表示目标的登录（login）名。
- `-P wordlist.txt` ： `-P` 在 `wordlist.txt` 文件之前，该文件是一个文本文件，其中包含你要尝试进行匹配的密码列表。
- `server` 是目标服务器的主机名或 IP 地址。
- `service` 表示你试图发起密码字典攻击的对应服务。

以下为一些具体示例：

- `hydra -l mark -P /usr/share/wordlists/rockyou.txt MACHINE_IP ftp` 将mark用作用户名，在 FTP 服务器上迭代匹配密码字典所提供的密码；
- `hydra -l mark -P /usr/share/wordlists/rockyou.txt ftp://MACHINE_IP` 与上一个示例相同， `MACHINE_IP ftp` = `ftp://MACHINE_IP` ；
- `hydra -l frank -P /usr/share/wordlists/rockyou.txt MACHINE_IP ssh` 将用 `frank` 作用户名，因为它尝试使用不同的密码通过 SSH 登录。

你还可以添加一些额外的可选参数：

- `-s PORT` 为相关的服务指定非默认端口；
- `-V` 或者 `-vV` 显示详细信息，让 Hydra 显示正在进行尝试的用户名和密码组合。这种详细程度非常便于查看进度，尤其是在你对语法没有信心的情况下；
- `-t n` 其中 n 代表与目标的并行连接数，如 `-t 16` 表示将创建 16 个线程用于连接到目标；
- `-d` 用于调试，以获取有关正在发生的事情的更多详细信息。调试输出可以为你省去很多麻烦，例如，如果 Hydra 尝试连接到关闭的端口并超时，如果使用了 `-d` 则会立即显示这一信息。

成功破解密码后，你可以使用 `CTRL-C` 结束密码攻击过程。在实际情况下，密码攻击通常需要较长的时间，如果你希望 Hydra 能够实时更新密码攻击的进度，选择使用详细信息选项或者调试选项会非常有用。

总之，我们可以使用工具（例如 使用Hydra结合合适的密码列表）有效地执行针对登录系统的密码攻击。对于密码攻击的缓解方法可能很复杂，并且也取决于目标系统的具体情况，一些可以缓解密码攻击的方法如下：

- 密码策略：对用户设置的密码实施最低复杂性限制（如：要求至少12位、带英文字母、带符号等）。
- 帐户锁定：在一定次数的身份验证失败后自动锁定对应的帐户。
- 限制身份验证尝试：延迟对登录尝试的响应，对于知道密码的人来说，几秒钟的延迟是可以容忍的，但它们会严重阻碍自动化密码攻击工具的运行。
- 使用CAPTCHA（图片验证码）：此处需要提出一个机器难以解决的问题，以图形用户界面 (GUI)的方式在登陆页面显示一个验证码（注意：CAPTCHA 代表全自动区分计算机和人类的图灵测试）
- 要求使用公共证书进行身份验证：这种方法适用于 SSH等服务。
- 双重身份验证：要求用户提供可通过其他方式获得的验证码，例如通过电子邮件、手机app或 SMS(短信息服务-Short Message Service)等方式获得的验证码。
- 还有许多更复杂的验证方法，可能需要一些关于用户的既定信息，例如基于 IP 的地理定位验证。

使用上述方法的组合是缓解密码攻击的绝佳方法。

答题

```
root@ip-10-10-201-178:~# hydra -l lazie -P /usr/share/wordlists/rockyou.txt 10.10.164.26 imap
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organ
izations, or for illegal purposes.

                                                    注意使用小写imap
Hydra (http://www.thc.org/thc-hydra) starting at 2022-11-02 13:27:53
[INFO] several providers have implemented cracking protection, check with a small wordlist fi
rst - and stay legal!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~8
96525 tries per task
[DATA] attacking imap://10.10.164.26:143/

[143][imap] host: 10.10.164.26   login: lazie   password: butterfly
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2022-11-02 13:28:24
```

H2 小结

本文涉及的三种常见的攻击是：

1. 嗅探攻击
2. 中间人攻击
3. 密码攻击

对于上述每一项，我们关注的是攻击细节和缓解攻击的一些方法。

我们最好记住常用协议的默认端口号，为方便起见，此处将常见的协议按字母顺序排列在下表中：

Protocol	TCP Port	Application(s)	Data Security
FTP	21	File Transfer	Cleartext 明文
FTPS	990	File Transfer	Encrypted 加密
HTTP	80	Worldwide Web WWW	Cleartext
HTTPS	443	Worldwide Web	Encrypted
IMAP	143	Email (MDA)	Cleartext
IMAPS	993	Email (MDA)	Encrypted
POP3	110	Email (MDA)	Cleartext
POP3S	995	Email (MDA)	Encrypted
SFTP	22	File Transfer	Encrypted
SSH	22	Remote Access and File Transfer 远程控制和文件传输	Encrypted
SMTP	25	Email (MTA)	Cleartext
SMTPS	465	Email (MTA)	Encrypted
Telnet	23	Remote Access 远程控制	Cleartext

尽管有很多安全性的协议存在，Hydra 仍然是一个非常有效的工具，你可以从终端启动它来尝试破解不同的密码。我们在下表中总结了它的主要选项：

选项	解释
`-l username`	提供登录名
`-P WordList.txt`	指定要使用的密码列表
`server service`	设置要攻击的服务器地址和服务
`-s PORT`	在非默认服务端口号的情况下使用
`-V` 或者 `-vV`	详细输出 显示正在尝试的用户名和密码组合
`-d`	如果详细输出没有帮助，则显示调试输出