的博客

对 HTTPS 的中间人攻击

2018-02-23

本文翻译自 2009 年 2月 03 日发表于 IEEE 的文章 《Man-in-the-middle attack to the HTTPS protocol》，原文链接为

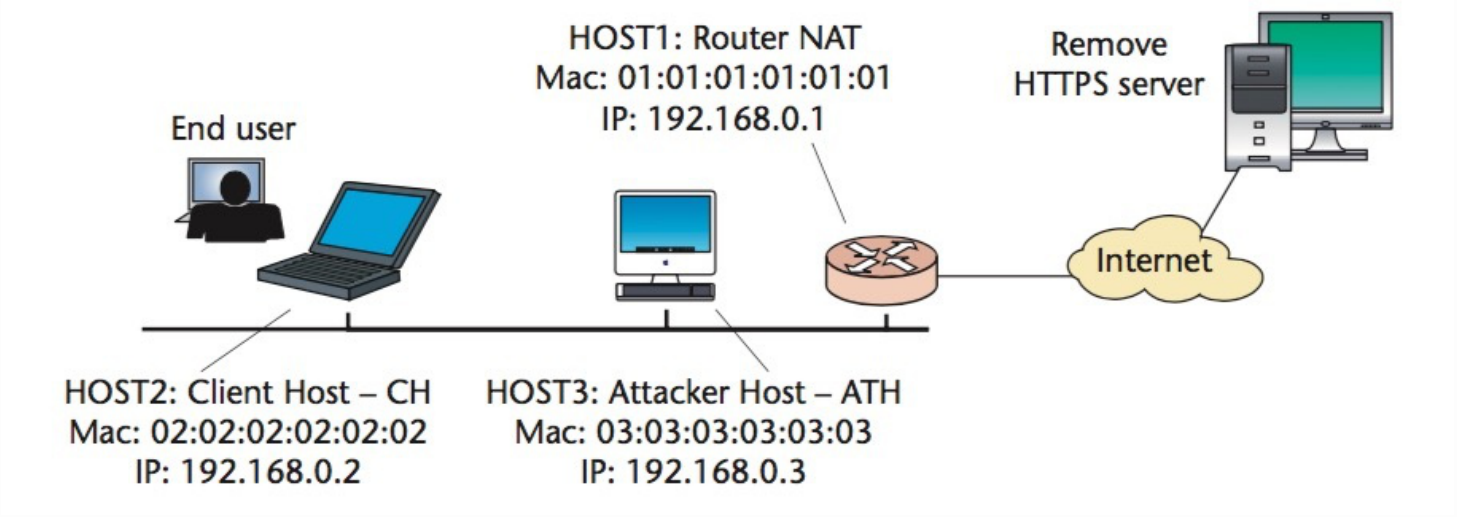
对于网上银行、电子商业和电子采购等需要处理敏感信息的交易，基于 Web 的应用通过 HTTPS 协议来保证其中的隐私和安全。用户相信这种协议可以避免他们个人信息、金融信息或私密信息等被他人截获。

在 1994 年，网景公司为了敏感信息的通讯引入了安全套接层协议（Secure Socket Layer，即 SSL）。随后在 1999 年，互联网工程任务小组（IETF）采纳了它，将其完善为安全传输层协议（Transport Layer Security，即 TLS），并通过它将 HTTP 协议加密为 HTTPS 协议。一个 HTTPS 的链接表明浏览器会在 443 端口通过 HTTP 协议下载页面，并在 HTTP 和 TCP 协议之间引入附加的 TLS 加密/认证层。因此，大部分人认为基于 HTTPS 的信息交换是安全的。在这篇文章中，我们将指出这种想法的危险之处 — 攻击者可以成功地拦截信息从而使信息安全失效。

基本概念

中间人攻击（**man-in-the-middle**，即 **MITM**）利用的是 HTTPS 服务器会向浏览器发送带有公钥的证书这一点。如果这个证书是不值得信任的话，那么整个通讯过程就会变得容易被攻击。中间人攻击会用被修改的证书替换 HTTPS 服务器的认证证书。如果当浏览器发出警告提醒时，用户不进行二次确认的话，那么攻击就完成了。这种情况经常发生，尤其是当用户访问局域网网站遇到自行签名的证书的时候。

本篇文章展示的攻击过程所用网络如下图1所示：用户（Client Host）想通过 HTTPS 和服务器（Remove HTTPS server）进行加密通讯，攻击者（Attacker Host）在网络中扮演网关的角色，它通过拦截源发出的流量，然后再发送给目标，从而获取了篡改信息的能力，而且没有其它人会意识到这一点。



攻击者通过以下步骤实施攻击：

- 在用户和局域网路由器间扮演网关的角色。
- 将用户建立连接的请求不做任何修改地发送给服务器。
- 拦截服务器返回的请求。
- 创建一个错误的自签名证书来替换原本的证书。
- 将错误的证书发送给用户。
- 当用户收到了证书之后，就会和攻击者间建立一个加密的通讯，同时攻击者也会和服务器建立另一个加密的通讯。

完成了这些步骤之后，用户和服务器显然都会以为和对方建立起了安全的通讯。然而实际上，攻击者因为拥有关键的 key，所以获得了破译通信内容的能力。a

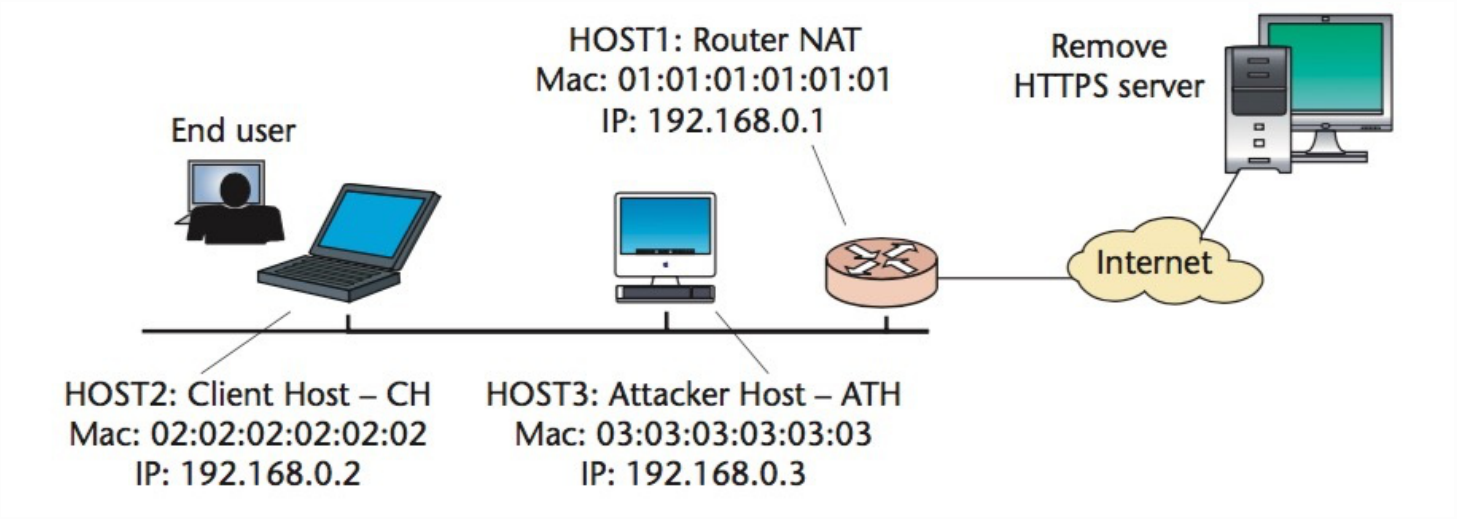
根据网络的配置情况，上述的攻击方式有不同的变形。例如攻击者可以和用户在同一个网路，也可以和服务器在同一个网络，或者在因特网上任何其它的地方。本篇文章假设的是最常见的情况 — 用户通过连接一台路由器来访问因特网。

中间人攻击通过恶意地修改地址解析协议（Address Resolution Protocol，即 ARP）和域名系统（Domain Name System，即 DNS）实现。特别地，利用 ARP 投毒技术来嗅探并获取直接使用 IP 在局域网上转发的流量。具体来说就是主机依靠 MAC 地址传递 IP 数据报，使用 ARP 协议来查找匹配的 IP 和 MAC 地址。鉴于 ARP 请求是广播的形式，所以攻击者可以读取它们并了解其他局域网内主机的 MAC 地址。然后，攻击者通过冒充其他主机的 MAC 地址来实施ARP欺骗。最后通过破坏 IP 和 MAC 地址之间的匹配关系，攻击者可以从截获来自或发送给特定 IP 地址的流量。

实际例子

假设大学里的一个学生，想从实验室的 Web 安全服务器（例如 https://example.unibo.it）上获取信息。这个学生需要登录到 HTTPS 服务器中来查看他的大学活动记录。

回到图1，让我们假设连接到局域网的主机使用私有 IP 地址 - 例如 192.168.0.0/24 ，并且路由器实现了网络地址转换（NAT）。 这代表了今天一个相当典型的配置。 图 1 显示了相关主机的IP和MAC地址（为了方便，下方再次展示了图 1 的内容）。



攻击者从对这个局域网进行 ARP 投毒开始。接着他从 ARP 相关的请求中得知了 HOST2（即用户）和 HOST1（即路由器）的 IP 地址。在攻击过程中，攻击者周期性地向 HOST1 发送 ARP 回包，将自己扮演为 HOST 2，即向 HOST1 声明 IP 为 192.168.0.2 的设备对应的 MAC 地址为 03:03:03:03:03:03；类似地，也向 HOST2 发送 ARP 回包，将自己扮演为 HOST1，即向 HOST2 声明 IP 为 192.168.0.1 的设备对应的 MAC 地址为 03:03:03:03:03:03。在攻击前，ARP 缓存如下图 （a）部分所示；随后，它会类似于下图（b）部分。这里进行 ARP 投毒的目的是将 HOST1（即用户）和 HOST2（即路由器）间的流量均导向 HOST3（即攻击者）。

```
[user@HOST1] $ arp -a
host2.victim.org (192.168.0.2) at 02:02:02:02:02:02 [ether]
   on eth0

[user@HOST2] $ arp -a
host1.victim.org (192.168.0.1) at 01:01:01:01:01:01 [ether]
   on eth0
(a)

[user@HOST1] $ arp -a
host2.victim.org (192.168.0.2) at 03:03:03:03:03:03 [ether]
   on eth0

[user@HOST2] $ arp -a
host1.victim.org (192.168.0.1) at 03:03:03:03:03:03 [ether]
   on eth0
(b)
```

现在，攻击者必须执行 DNS 投毒以将所有 DNS 请求重定向到攻击者的计算机。目标是让所有从用户到 Web 服务器的传输都通过攻击者。 攻击者使用名为 dnsspoof 的工具来回复由用户指向其他 IP 地址发出的DNS 请求，而不是它们应该传送至的点。 在我们的例子中，当用户发送一个 DNS 请求 example.unibo.it 的 IP 地址时，由于 ARP 有毒，请求被发送至攻击者。 接着攻击者实施了 DNS 投

毒，并向用户发送了一个DNS 回包 — 将 exmaple.unibo.it 转换为自己的 IP 地址 192.168.0.3。 此时，HOST2 和 HOST1 之间的所有传输以及 HOST2 的 DNS 请求都可以被攻击者拦截和修改。

为了让用户相信所有的事情都是正常的，攻击者必须通过对 IP 路由表进行简单的修改，从而在用户和服务器之间转发流量。 例如，攻击者可以使用 frag-router 启用 IP 转发（http://packages.qa.debian.org/f/fragrouter.html）。

一旦攻击者截获了用户和服务器之间的传输，攻击者就可以启动一个伪造的 HTTPS 会话，在这个会话中攻击者可以窃听和解密它的流量。 这需要攻击者通过 webmitm 工具来以服务器的名义生成伪造的证书。 第一次运行 webmitm 需要攻击者回答几个问题（参见图3）。 在实践中，攻击者应该提供与服务器相关的答案，因为用户的浏览器可能会显示该答案。

```
root@5[knoppix]# webmitm -d
Generating RSA private key, 1024 bit long modules
. . . . . + + + + + +
. . . . . + + + + + +
E is 65537 (0x10001)
You are about to be asked to enter information that will be
incorporated
Country Name (2 letter code) [AU]:IT
State or Province Name (full name) [Some-State]: Italy
Locality Name (eg, city) []: Cesena
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
University of Bologna
Organizational Unit Name (eg, YOUR name) []: Marco Ramilli
Email Address []:marco.ramilli@unibo.it
Please enter the following 'extra' attributes
To be sent with your certificate request
A challenge password []: IEEESecurityPrivacy
An optional company name []:
Signature ok
Subject=/C=IT/ST=Italy/L=Cesena/0=University of Bologna/
OU=Security/CN=Marco
Ramilli/emailAddress=marco.ramilli@unibo.it
Getting Private key
Webmitm: certificate generated
```

如果在连接至 https://example.unibo.it 时，用户接受了伪造的证书，那么攻击者就可以拦截并解密所有的流量应为他（而不是服务器）拥有与证书中公钥相配的私钥。为了嗅探网络中的流量，我们使用Wireshark这款流行的跨平台的协议分析软件。

当用户连接到 HTTPS 站点时，它必须接受新的证书。 大多数浏览器在用户收到未由受信任的权威机构

签名的证书时会提醒用户。 颁发“自签名”的 HTTPS 证书很常见，最终用户通常熟悉此警报并快速接受证书。 这种攻击通过利用这种坏习惯而发挥作用 — 事实上，一位经验丰富的用户如果仔细阅读证书的话，将很有可能发现异常情况。 尽管如此，我们认为对缺乏官方签名的证书进行详细检查的行为并不常见。 因此，攻击者很有可能在其控制下发起 HTTPS 会话。 在这种情况下，攻击者可以捕获安全信道上的通信。 当它决定终止捕获阶段时，攻击者可以保存捕获的网络流量并使用 ssldump 工具对文件进行解密，如下图 4 å所示。

```
root@6[knoppix]# ssldump -r temp -k webmitm.crt -d > out
root@6[knoppix]# ls
Desktop out temp tmp webmitm.crt
```

在明文文件中，攻击者可能会发送诸如用户的用户名和密码等保密信息。 然而这通常是很长的一系列字符，其中很难识别包含所寻求信息的文本位置。 攻击者可以通过检查原始网页的输入语法来解决这个问题（如下图 5 所示）。

```
&ndsp;<input type="text" name="userid" maxlength="64" size="14"
value="">
</span>
<td align="left" class="unique">
<span class="uniqueError">
 
</span>
</td>
</tr>
<tr class="unique">
<td align="right" class="unique">
<span class="uniqueLabel">
 Password
</span>
</td>
<td align="left" class="unique">
<span class="uniqueinput">
 <input type="password" name="password" maxlength="10"
size="14" value="">
</span>
.. .. ..
```

最后，通过搜索被拦截的流量得到的解密文件，可以获得所需的用户名和密码，如下图 6 所示。

```
Connection: Keep-Alive
Cache-Contrlo: no-cache
Cookie: JSESSIONID=62ED8B17175165BE92D43F2E61DB0A10
Family=studente&
userid=marco.ramilli&password=IEEESP----------------------------
20 20 19.5022 (12.4627) C>S application_data
21 21 19.5022 (0.00000) C>S application_data
22 22 21.4693 (1.9671 ) S>C application_data
19 17 21.4973 (1.9794 ) S>C application_data
-------------------------------------------------------------------
```

我们已经证明，可以通过利用通用局域网的一些属性以及缺乏经验的用户的典型行为，从而攻击通过 HTTPS 保护的基于 Web 的连接。上述攻击的实施并非很容易，但对于有经验的用户来说，这当然也不难。 此外，攻击者可以轻易伪造证书，这突显了网站应该认识到自签名证书潜在的危害。因为用户很快习惯了浏览器发出的警告; 所以他们会忽视这些危险！ 尽管强大的加密技术是提供数据保护的强大工具，但其提供的安全性却是整个链条中中最薄弱的一环。

#网络 #翻译

上一篇 下一篇