FTP 安全扩展

本备忘录状态

为文档为 Internet community 指明了一种 Internet 标准跟踪协议。它需要进一步进行讨论和建议以得到改进。请参考最新版的“Internet 正式协议标准”(STD1)来获得本协议的标准化程度和状态。发布本备忘录不受限制。

版权声明

概要

本文档定义了 FTP 规范 STD 9, RFC 959，“文件传输协议(FTP）”（1985 年 10 月）的扩展。这个扩展为控制和数据信道提供了强大的认证、完整性、和保密性。介绍了新的可选指令、回应、和文件传输编码方式。

本规范中将介绍下列新的可选指令：

AUTH (认证/安全机制)
ADAT (认证/安全数据)
PROT (数据信道保护等级)
PBSZ (保护缓冲区大小)
CCC (清除指令信道)
MIC (完整性保护指令)
CONF (保密性保护指令) 和
ENC (私密性保护指令)

为了被保护的回应也介绍了一种新的回应类型（6yz）。

以上指令没有要求必须被实现，但它们之间存在相互依存性。这些依存性可以从指令中得到证明。

注意：本规范兼容 STD 9, RFC 959。

1. 介绍

目前，文件传输协议在STD 9, RFC 959中定义，在Internet上的一些地方用用户名和密码通过明文传输来实现服务器端对客户端的认证（通过USER和PASS指令）。除了像“匿名”FTP文档这样的服务以外，这意味着客户正在冒着自己被局域网或广域网上的其他人监听，从而密码被窃取的危险。如果不能或不愿意承认内部网的安全隐患，那么潜在的攻击者将通过FTP服务器来窃取被暴露的文件密码。

除了用一种安全的方式鉴别用户的问题，还有鉴别服务器，保护敏感的数据和/或校验数据完整性的问题。一个攻击者仅仅通过监视网络就可以访问重要和敏感数据，或通过积极的手段来删除或修改正在传输的文件使其完整性遭到破坏。一个活跃的攻击者还可以对他选择的站点伪造文件传输，并可

以调用服务器上的其他指令。FTP目前还不能对指令，回应，或传输的数据的真实性提供任何加密和认证。注意，这些安全服务甚至对匿名文件访问也是有价值的。

当前安全地发送文件的实际应用通常使用下面的方法：

1.用人工分发的密钥提前加密，通过FTP传输文件。

2.用人工分发的密钥加密文件，通过电子邮件发送经过编码的文件。

3.通过PEM消息，或

4.通过使用Kerberos增强过的rcp指令。

没有一种方法是事实上的标准，也没有一种是真正交互的。需要使用FTP进行安全地传输文件的方式，它应该支持FTP协议保持一致的方式，并利用现有的安全机制和技术的优势。如果这些安全方面的服务能被引入FTP协议中，并且与协议中原有的其他服务和睦共处的话，那么将FTP协议扩展就成为必然的事情了。

尽管FTP控制连接遵从TELNET协议，并且TELNET已经定义了一个认证和加密选项，[TELNET-SEC]，[RFC-1123]，明确地禁止通过控制连接来流通TELNET选项（除了Synch 和 IP）。

另外，TELNET的认证和加密选项没有提供单独的完整性保护（没有机密性），也没有对数据信道的保护。

2.FTP安全概览

从最高的高度，FTP 安全扩展寻求一种理论机制，能提供鉴别和/或批准连接，和完整性和/或机密性来保护指令，回应和数据传输。

就 FTP 安全来说，认证是确定客户身份和/或服务器身份的安全方式，通常使用加密技术。基本的 FTP 协议没有认证的概念。

授权是确定一个用户登录的过程。基本的授权过程包括 USER，PASS，和 ACCT 指令。就 FTP 安全扩展来说，用一种安全机制建立的认证可以用来决定是否授权给用户。

没有安全扩展，客户端的认证这样的说法就会不会出现。FTP 认证是使用一个明文形式的密码作为 PASS 指令的参数，在网络上传递来完成的。密码的持有者假定是被授权传输的 USER 指令中的用户名，但是客户端的身份从来不能进行安全的确认。

FTP 安全的交互过程的开始，是客户端使用 AUTH 指令告诉服务器它想用什么安全机制。服务器会接受机制，或拒绝机制，或服务器没有实现安全扩展所以完全拒绝了这个指令。客户端会再试其他的安全机制，直到一个请求被服务器接受。

这样才可以允许进行最基本的连接协商。（如果想得到更多更复杂的协商，还要再执行一个安全机制。）服务器端的回应会指明客户端是否必须对它所提出的安全认证机制作进一步的解释。如果不需要，通常就意味着这种机制将会对口令（由PASS命令指定）进行与以往不同的解释，比如信令或一次性口令系统。

如果服务器需要额外的安全信息，那么客户端和服务器会进入到一个安全的数据交换状态。客户端会发送一个包含安全数据的第一块的 ADAT 指令。服务器的回应会指出是否数据交换已完成，是否有错误，或是否需要更多的数据。服务器的回应可随意地包含需要客户端解释的安全数据。如果需要更多的数据，客户端会发送另一个包含数据下一块的 ADAT 指令，并等待服务器的回应。交换可以根据需要持续很长时间。一旦交换完成，客户端和服务器就确定了安全联系。安全联系将包括认证（客户端，服务器，或相互的）和用于完整性和/或机密性的关键信息，使用中依赖的机制。

“security data”这个术语并非是臆造出来的。安全数据交换的目的是建立一个安全联系，就象上面所说的那样在客户端可能不会包括任何的认证过程。例如，一个Diffie-Hellman交换创建了一个密钥，但是没有认证发生。如果一个FTP服务器有一个RAS密钥对而客户端没有的话,那客户端可以确认服务器端的身份,而服务器却不能确认客户端的身份。

一旦双方建立了安全联系，作为此关联一部分的认证，将代替传统的用户名/口令机制去审核并准许用户连接到服务器。为标识用户将在服务器上的身份，需要用USER命令向服务器提供一个用户名。

为了防止攻击者插入或删除控制流中的命令，如果安全联系支持完整性，那么服务器和客户端必须在控制流中使用完整性保护，除非它先传输了一个 CCC 指令取消了这次请求。

完整性保护使用 MIC 和 ENC 指令，和 63z 回应码来完成。CCC 指令必须在完整性保护下传送。如果双方没有建立安全关联，或建立的安全关联中不支持数据完整性保护，或者已成功使用了 CCC 命令，那么指令和回应的传送很可能就不是完整的（就是说，传送的是明文或只是进行了加密的数据）。

一旦客户端和服务器端用 PBSZ 命令协商了一个合适的缓冲区，用来装入数据信道中的保护数据，安全机制也会被用来保护数本文档没有指定什么政策。特别是在协议实现方面，客户端和服务器端可以根据双方建立的安全关联有选择地限制那些可以被实现的活动。例如，服务器端会要求客户端通过安全机制而不是口令方式进行认证，会要求客户端从信令中提供一个一次性口令，会要求至少要对命令信道进行完整性保护，或或要求某个指定的文件加密后再传输。为确保被下载文件的有效性，如果没有数据完整性保护机制的话，与匿名服务器连接的客户端会被拒绝对文件传输的请求。

除了下一节将要谈到的依赖性外，不需要特别的功能设置。这意味着认证功能，数据完整性保护功能或保密保护功能没有要求必须实现，尽管不包括这些功能的安全机制是没有多大用处的。以下的安全机制的实现都是可以接受的，例如只有数据完整性保护，单向认证及加密功能，或因政策和技术方面考虑而要求的任何的功能子集。当然有时基于某种策略的需要，某些实际应用可能需要更强大的安全防护以至于超过了现在所能提供的防护水平。

3.新 FTP 指令

以下的指令是可选的，但它们中的某些有相互依赖性。它们是 FTP 的访问控制指令的扩展。

文档中所述的回应码只是推荐的，不是必须的。目的是回应码描述存在的成功和失败模式的完整范围，但服务器端可对返回给客户端的回应码加以限制。例如，服务器可以执行一个特殊的安全机制，但使用它时却有某些策略限制。此时服务器端会返回回应码 534，或者当它不想泄露它不允许但支持的机制时会返回回应码 504。如果服务器端某些情况下使用的回应码不是文档中推荐的回应码的话，它应尽量保证只有此回应码的最后一个数字与推荐的回应码不同。任何情况下，服务器端返回的回应码必须是文档中规定的，对应它所收到命令的所有回应中的一个，并且此回应码的必须与此情况下推荐的回应码相同。

认证/安全机制（AUTH）

此参数是鉴定可支持的机制 Telnet 字串。该字串不区分大小写。除了以“X-”开头的为本地使用而保留的值之外，该值必须已在 IANA 注册。

如果服务器端不承认 AUTH 命令，它必须返回回应码 500 。对任何未被承认的指令也使用 500 回应码。如果服务器端承认 AUTH 命令，但没有实现安全扩展，它应返回回应码 502。

如果服务器端不理解指定的安全机制，它应返回回应码 504。

如果服务器端不愿接受指定的安全机制，它应返回回应码 534。

如果服务器端不能接受指定的安全机制，例如当请求的资源不可用时它应返回回应码 431。

如果服务器端愿意接受指定的安全机制，但它还需要安全数据，它必须返回回应码 334。

如果服务器端愿意接受指定的安全机制，并且不需要任何的安全数据，它必须返回回应码 234。

如果服务器端返回回应码 334，它可能会包括下一节所述的安全数据。

有些服务器允许 AUTH 命令被再次使用，以此建立新的认证。如果 AUTH 命令被接受，则由于此前使用的安全指令而引起的 FTP 服务器任何状态的改变都将被复位。这种情况下，服务器也必须要求用户身份的重新核准（参看文中第 4 节有关于“核准”的解释）（就是说重新发出部分或全部的 AUTH，PASS，和 ACCT 指令）。

认证/安全数据（ADAT）

此参数是用 base 64 编码表示的安全数据的 Telnet 字串（参看第 9 节“base 64 编码”）。如果表示成功的回应码返回给客户端，且服务器端希望将安全数据再传回客户端，那么服务器端应会使用“ADAT=base 64data”这样格式的字串作为回应的正文部分。

以上情况下的安全数据是由 AUTH 指令指定的安全机制的具体体现。ADAT 指令及相关的回应允许客户端和服务器端使用任意的安全协议。为使双方都能明白哪些可选安全协议对双方是可用的，双方的安全数据交换中必

须包含足够多的信息。例如如果客户端不支持数据加密，服务器必须能知道，因此它不会发送加密的指令信道回应。强烈建议在安全机制中加入命令信道序列化功能，以确保命令不被删除，重新排序或重发。
ADAT 命令必须在成功的 AUTH 指令令之前发出，并且在安全数据交换完成之后（成功或不成功）不能被使用，除非在 AUTH 指令之前复位了安全状态。

如果服务器没有收到 AUTH 指令，或者安全数据交换已完成，而服务器的安全状态还没有用 AUTH 指令复位时，它应返回回应码 503 。

如果服务器不能解码由 base 64 编码的参数，它应返回回应码 501。

如果服务器拒绝安全数据（例如校验失败），它应返回回应码 535 。

如果服务器接受了安全数据，并且还需要额外的数据，则它应返回回应码 335。

如果服务器接受了安全数据，但不再需要任何额外数据的话（也就是安全数据交换已成功完成），它必须返回回应码 235。

如果服务器返回的回应码是 235 或 335 时，那么在它回应的正文部分会包含如上所述的安全数据。

如果 ADAT 命令返回错误，安全数据交换将失败，此时客户端必须复位其内部的安全状态。如果客户端变得与服务器端不同步时（例如，当服务器返回一个由 AUTH 命令产生的回应码 234 时，但客户端此时还有更多的数据要传送），客户端必须使服务器端的安全状态复位。

保护缓冲区大小（PBSZ）

此参数是一个十进制整数，它表示在文件传输期间，能够发送或接收的编码数据块的最大字节数。此数值不能超出一个 32 位无符号整数所能表示的范围。

此命令允许客户端和服务器端为连接而协商一块最大的保护缓冲区。没有缺省的大小，在客户端必须在发出第一个 PORT 指令之前发出 PBSZ 指令。

PBSZ 命令必须被成功的安全数据交换保护。

否则，服务器端必须返回回应码 200。如果客户端所提出的缓冲区尺寸对服务器来说过大，它必须在回应的正文中用“PBSZ=number”形式的字符串指明一个小一些的缓冲区尺寸。此时，客户端和服务器端必须使用那个较小的数值作为缓冲区尺寸值。

数据信道保护等级（PROT）

此参数是描述数据信道防护等级的单一 Telnet 字串码。

此命令指示服务器，客户端与服务器端之间将使用哪种类型的数据信道保护。如下代码被指定：

C - Clear
S - Safe
E - Confidential

P - Private

如果其他等级没有被指定，则默认等级是 Clear。Clear 保护级是指数据信道将传送未加工的文件数据，未经安全处理。Safe 保护级是指数据将使用数据完整性保护。Confidential 保护级是指数据将使用保密性保护。Private 保护级是指数据将同时使用数据完整性保密性加以保护。

某一安全机制可以不必提供以上所有的数据信道保护等级。当然它也可以在某一防护等级提供比要求中更多的数据保护机制（比如，某安全机制提供保密性保护机制，但由于 API 或其他方面的考虑，又在编码过程中加入了数据完整性保护机制）。

PROT 指令必须在客户与服务器双方协商保护缓冲区大小后才能发出。

如果服务器不理解 PROT 指令中指定的保护等级，它应返回回应码 504。

如果当前安全机制不支持指定的保护等级，服务器返回回应码 536。

如果服务器还未与客户端完成关于保护缓冲区大小的协商，它应返回回应码 503。

如果以前客户端从未发出过 PBSZ 指令，则客户端发出的 PROT 指令将被服务器拒绝，同时返回回应码 503。

如果服务器不愿接受指定的保护等级，它应返回回应码 534。

如果服务器不能够接受指定的防护等级，例如如果某一所需资源不可用，它应返回回应码 431。

否则，服务器必须返回回应码 200 以表明它接受了客户端指定的保护等级。

清除指令信道（CCC）

该指令没有参数。

某些环境下使用某一安全机制认证及核准客户端和服务器端，但不对随后的指令进行任何完整性检查，这种做法是可取的。

某些环境下使用某一安全机制认证及核准客户端和服务器端，但不对随后的指令进行任何完整性检查，这种做法是可取的。它将会被使用在如下的环境中，在此环境中的 IP 层安全将不被忽视，以此确保主机身份被完全确认，并且 TCP 协议数据流不被篡改，但是在此环境中用户身份认证还是要求的。

如果在任何连接中指令未受保护，那么攻击者就可以插入一个指令到控制流，并且服务器没办法知道哪一个是非法的。为了防止这样的攻击，一旦安全数据交换成功完成，如果安全机制支持数据完整性保护，那么必须使用完整性保护功能（通过 MIC 或 ENC 指令和回应码 631，或 632 来实现），直到使用 CCC 命令使命令信道失去对消息的完整性保护功能。CCC 命令本身必须在完整性保护之下使用。

一旦 CCC 命令成功完成，如果一个指令没有被保护，那么该指令的回应也不会被保护。因为客户端发出 CCC 命令后将不再支持数据完整性保护，由此显然此举是为支持客户端与服务器端的互操作性的（interoperability）。此命令必须使用在安全数据交换之前使用。

如果该指令没有在完整性机制保护之下使用，服务器端必须返回回应码 533。

如果服务器不愿关闭完整性保护功能，它应返回回应码 534。

否则，服务器必须返回回应码 200，以表明指令信道上现在将传送不受保护的指令及回应。

完整性保护指令（MIC）和
机密性保护指令（CONF）和
私密性保护指令（ENC）

MIC 指令的参数是一个 Telnet 字串，该字串是由经过 base64 编码的“安全”信息构成，而此安全信息则由安全机制中专门进行信息完整性保护的过程所产生。 CONF 指令的参数是一个 Telnet 字串，该字串是由经过 base64 编码的“保密”信息构成，而此安全信息则由安全机制中专门进行保密性保护的过程所产生。 ENC 指令的参数是一个 Telnet 字串，该字串是由经过 base64 编码的“私密”信息构成，而此安全信息则由安全机制中专门进行信息完整性保护和保密性保护的过程所产生。

服务器端将解码和/或校验这些经过编码的信息。

这些指令必须在成功进行安全数据交换之前使用。

服务器可能会要求在成功进行数据交换后使用的第一个指令是 CCC，并且没有执行任何保护指令。在这样的情况下，服务器端应返回回应码 502。

如果服务器不能解码由 base64 方式编码的参数，它应返回回应码 501。

如果服务器还未完成与客户端的安全数据交换，它应返回回应码 503。

如果服务器在某一支持数据完整性保护的安全机制下完成了与客户端的安全数据交换，但由于策略或某些机制实现方面的限制，它还需要收到一个 CCC 指令，它应返回回应码 503。

如果服务器因不支持当前的安全机制而拒绝该指令的话，它应返回回应码 537。

如果服务器拒绝接受该指令（例如校验和错误），它应返回回应码 535。

如果服务器不愿接受该指令（例如，由某些策略决定的私密性，或者是在收到 CCC 指令之前先收到了一个 CONF 指令），它应返回回应码 533。

否则，此指令将被作为 FTP 指令所解释。无需包含行结束码，但如果包含了，则它必须是 Telnet 行结束码，而非本地的行结束码。

某些情况下或是任何可能情况下，服务器会要求所有指令被保护。这样除了 MIC,CONF,和 ENC 指令外，它应对其它指令返回回应码 533。

4.登录核准

安全数据交换会在其它事情中，以一种安全方式确立客户端对于服务器端的唯一性身份，此身份将被作为用户登录核准的唯一输入。

为回应 FTP 客户端的登录指令（AUTH ,PASS,ACCT），服务器可以选择改变某些 RFC 959 中所指定的命令及回应顺序。还有一些可用的新回应。

如果服务器愿意准许由 USER 指令指定的基于安全数据交换确立的具有唯一性的用户登录，它应返回回应码 232。

如果安全机制要求一个质问/回应口令，它应对 USER 命令回应 336。回应的正文部分应包括质问信息。在此情况下，客户端在提示用户输入口令之前必须将质问信息显示给用户。这一过程对于复杂的或某些提供对话框或其它输入方式的图形用户界面的客户程序来说特别合适。用户有时可能需要用回应 336 中正文部分的质问信息来构造有效口令。因此这些客户程序应避免在将用户名发给服务器端之前就提示用户输入口令。

5.新 FTP 回应码

新的回应码分为两类。第一类是新的 FTP 安全扩展指令所必需的。第二类是用来指明被保护的回应的新回应码。

5.1 新的独有的回应码

232 用户登录，使用安全数据交换鉴别

234 安全数据交换完成

235 [ADAT=base64data]

此回应指明安全数据交换成功完成。方括号不包含在此回应内容中，它只指明回应中的安全数据是可选的。

334 [ADAT=base64data]

此回应指明客户端所请求的安全机制可用，并且包含了客户端发出下一个指令所需要的安全数据。方括号不在此回应内容中，它只指明回应中的安全数据是可选的。

335 [ADAT=base64data]

此回应指明客户传来的安全数据被接受，并且还需要额外的安全数据来完成数据交换。方括号不在此回应 内容中，它只指明回应中的安全数据是可选的。

336 Username okay, need password. Challenge is "...."

用户名通过，需要口令。质问是“……”

安全机制所选取的质询内容所代表的含义对系统的使用者来说应该是有实际意义的（sensible）。

431 Need some unavailable resource to process security.

某些相关资源不可用。

533 Command protection level denied for policy reasons.

由于策略方面的原因，命令保护等级被拒绝。

534 Request denied for policy reasons.

由于策略方面的原因，请求被拒绝。

535 Failed security check (hash, sequence, etc).
安全检查失败，(哈希排序，顺序，及其他)
536 Requested PROT level not supported by mechanism.
安全机制不支持请求的保护等级。
537 Command protection level not supported by security mechanism.
安全机制不支持指令保护等级。

5.2 被保护回应

引入一种新的回应类型：

6yz Protected reply

有三种此种类型的回应码。第一个回应码 631 指明一个受到数据完整性保护的回应。第二个回应码 632 指明一个保密性及数据完整性保护的回应。第三个回应码 633 指明一个保密性保护的回应。

回应 631 的正文部分是一个 Telnet 字串，该字串是由经过 base64 编码的“安全”信息构成，而此安全信息则由安全机制中专门进行信息完整性保护的过程所产生。 回应 632 的正文部分是一个 Telnet 字串，该字串是由经过 base64 编码的“私密”信息构成，而此安全信息则由安全机制中专门进行保密性保护和信息完整性保护的过程所产生。回应 633 的正文部分是一个 Telnet 字串，该字串是由经过 base64 编码的“保密”信息构成，而此安全信息则由安全机制中专门进行保密性保护的过程所产生。

客户端将解码及校验这些经过编码的回应。如何处理解码失败或回应的校验是实现细节问题。回应不必包括行结束码，但如果包含了，则它必须是 Telne 行结束码，而非本地的行结束码。

保护回应只可以在安全数据交换成功之后发出。

回应 63z 也许是多行的回应。在这样的情况下，回应的纯文本部分必须被分成一定数量的段，每段都要受到保护，然后还要采用 base 64 方式编码，目的是为了能整齐地分隔多行的回应。在纯文本的回应和经过编码的和经过编码的回应中的行间不需要任何联系。Telnet 行结束码必须出现在经过编码的回应的纯文本部分，在文本最后对如上的行结束码的使用是可选的。

多行回应的格式必须比[RFC959]的续文中所描述的还要严格。特别是在最后一行之前的每一行必须由一个回应码紧跟一个连字符，然后接一个回应的 base 64 编码段。

例如，如果纯文本回应是

123-First line
Second line
234 A line beginning with numbers
123 The last line

那么最终的受保护回应可以是下列的任一形式， （因为页面的限制，第一个例子有一行被打断了。）

631 base64(protect("123-First line\r\nSecond line\r\n 234 A line
631-base64(protect("123-First line\r\n"))
631-base64(protect("Second line\r\n"))
631-base64(protect(" 234 A line beginning with numbers\r\n"))
631 base64(protect("123 The last line"))

631-base64(protect("123-First line\r\nSecond line\r\n 234 A line b"))
631 base64(protect("eginning with numbers\r\n123 The last line\r\n"))

6.数据信道封装

当客户端和服务器端的数据传输（方向任意）受到保护时，必须对数据实施某些转变和封装，以便接收者能顺利的将传输的文件解码。

当有关被传送文件的表示类型，文件结构，传输模式被转变后，发送者必须对她们执行所有的保护措施。通过数据通道传送的数据，基于保护的目的，将被作为字节流对待。

当以认证方式传送文件时，只有此文件的个别数据块被进行认证处理，而不是整个文件。因而可能会发生如下情况，即使用将非法数据块混入数据流的攻击方式时，认证却认为数据合法，从而导致接收者无法检测出受到的文件是不可靠的。为防范此类攻击，使用的安全机制在某些实现细节上应包含防止此类攻击的机制。附录 I 的一些 GSS-API 机制和附录 II 的 Kerberos 机制可以用来做这件事。

发送者必须取得输入的字节流，并将其拆分成许多的数据块，而这样的数据块经过安全机制的某一具体过程编码后，其尺寸不能大于客户端和服务器端用 PBSZ 协商的缓冲区尺寸。每一个数据块必须经过编码，并且将编码后的数据块与其长度值一同传送，此长度值由一四字节的无符号整数表示，这是最重要的几个字节。

当到达被传文件的尾部时，发送者必须将一字节值均为零的数据块编码，然后在关闭连接之前将这个最后的数据块发送给接收者。

接收者将读出如上的四字节长度值，再读取由此长度值指定的字节数的一块数据，然后解码并使用安全机制的某一具体过程对其进行校验。必须重复以上过程直到它收到了一个编码字节值全为零的数据块。这表明已到了编码字节流的尾部。

任何有关表示类型、文件结构和传输模式的转变将由接收者在如上操作过程所生成的字节流上进行。

当使用块传输模式时，发送者的（明文）缓冲区大小独立于块的大小。

如果当前保护等级与服务器为特殊的文件传输的安全需要而指定的保护等级不一致，服务器将对 STOR，STOU，RETR，LIST，NLST，或 APPE 指令返回回应码 534。

如果在数据传输阶段，在服务器端的任一数据保护服务功能失败的话，服务器将对数据传输指令（无论是 STOR，STOU，RETR，LIST，NLST，还是 APPE）返回回应码 535。

7．可能的策略方面的考虑

当没有关于客户端和服务器端策略限制时，这是少数执行时应实行的建议。

—— 一旦进行了安全数据交换，服务器应该要求所有的指令都被保护（完整性的和/或保密性的），并且对所有回应也进行保护。回应使用的保护等级应与相关指令的保护等级相同。这些包括回应指明的 MIC，CONF，和 ENC 指令执行失败的回应。特别地，要求 AUTH 和 ADAT 指令被保护是没意义的；要求 PROT 和 PBSZ 指令被保护是有意义且有用处的。特别，还有在期望某功能的安全机制实现过程中，基于互操作性的考虑 CCC 命令被定义，但不推荐使用它。

—— 在任何可能的情况下，客户端都应对 PASS 指令加密。如果服务器知道使用了加密功能，那它拒绝接受没有加密的 PASS 指令就是合理的。

—— 尽管没有安全指令被要求在必须实现，推荐在考虑网络站点的策略因素（例如输出控制）及其所能支持的机制的前提下，尽量提供所有能被实现的指令。

8.公布的规范

这些节在 RFC 959 的第 5.3 和 5.4 节后已有示范，描述的是相同的信息，除了标准的 FTP 指令和回应。

8.1 FTP 安全指令和参数

```
AUTH <SP> <mechanism-name> <CRLF>
ADAT <SP> <base64data> <CRLF>
PROT <SP> <prot-code> <CRLF>
PBSZ <SP> <decimal-integer> <CRLF>
MIC <SP> <base64data> <CRLF>
CONF <SP> <base64data> <CRLF>
ENC <SP> <base64data> <CRLF>

<mechanism-name> ::= <string>
<base64data> ::= <string>      ；必须符合第9节描述的格式
<prot-code> ::= C | S | E | P
<decimal-integer> ::= any decimal integer from 1 to (2^32)-1
```

8.2 指令-回应序列

```
安全联系设置
   AUTH
      234
      334
      502, 504, 534, 431
```

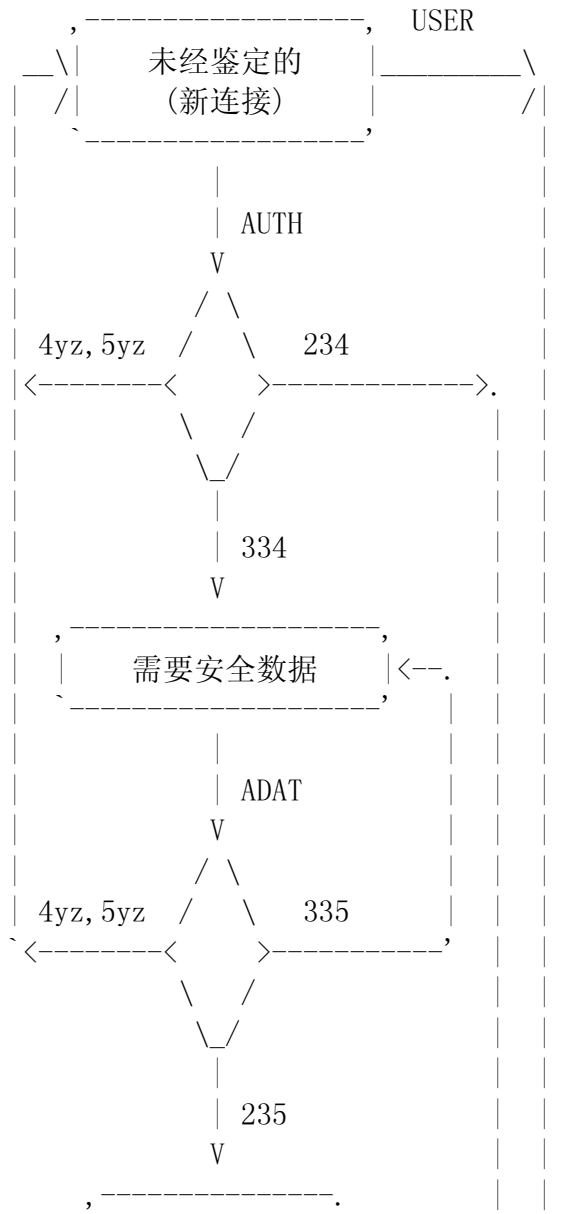
500, 501, 421
ADAT
235
335
503, 501, 535
500, 501, 421
数据保护协商顺序
PBSZ
200
503
500, 501, 421, 530
PROT
200
504, 536, 503, 534, 431
500, 501, 421, 530
指令信道保护顺序
MIC
535, 533
500, 501, 421
CONF
535, 533
500, 501, 421
ENC
535, 533
500, 501, 421
安全性增强的登录顺序（只有新的回应码被列出）
USER
232
336
数据信道顺序(只有新的回应码被列出)
STOR
534, 535
STOU
534, 535
RETR
534, 535
LIST
534, 535
NLST
534, 535
APPE

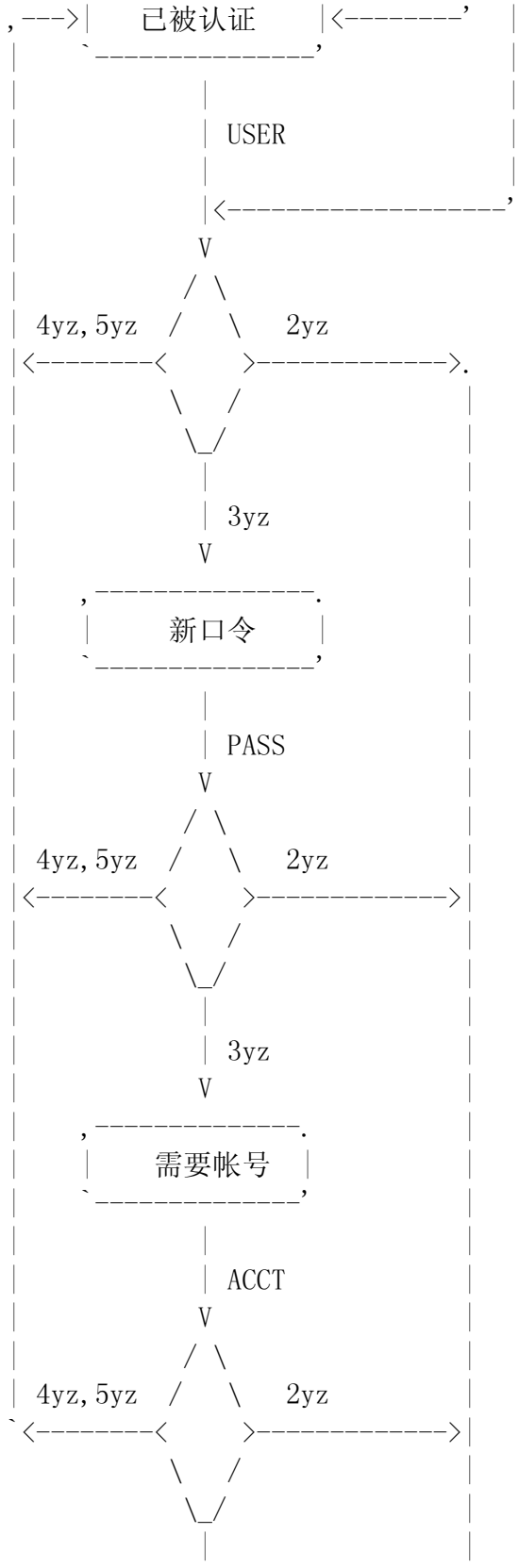
534, 535

除了这些回应码外，任何安全指令可以返回回应码 500，501，502，533，或 421。一旦安全数据交换成功完成，任何 ftp 命令都可以返回一个封装在回应 631，632，或 633 中的回应码。

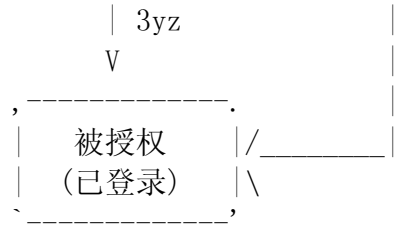
9. 状态图

本节演示了在一个增强了安全机制的 FTP 服务器上进行认证和核准的流程。长方形块显示了客户端必须发出指令那一时刻的状态，菱形块显示了服务器必须给出指令回应那一时刻的状态。





当客户端和服务器端之间完成认证后，如果完整性保护机制可用，则指令必须受到此机制的保护，CCC指令可以用来取消这种限制。



10. base 64编码

base 64 编码类似于[RFC-1421]中 4.3.2.4 节所描述的 Printable 编码，除了必须不含有行间中断。该编码定义如下。

由进行常规保护的安全机制所产生的位串从左到右经某种方式编码后就可变成所有因特网上的站点都可以表示的 ASCII 字符，尽管不必使用相同的位模式（例如，尽管字符“E”在基于 ASCII 的系统中用十六进制数 45 表示，而在基于 EBCDIC 的系统中用十六进制数 C5 表示，但这两种表示法所代表的意义在本地系统中是一样的）。

使用国际字母表 IA5 的一个包含 64 个字符的子集，使用 6 位二进制数就可以将这 64 个可打印字符完全表示（以上的字符子集在 IA5 和 ASCII 中是完全相同的）。字符“=”用于在编码后输出的可打印字符序列的尾部进行填充。

编码时以包含 24 位二进制的位串为一组作为编码程序的输入数据，经过编码后输出 4 个字符。具体做法是，将 64 个可打印字符按“[A--Z][a--z][0--9]+ /”的顺序排列并按排列的自然顺序建立它们的索引，索引值从 0--63。将一个输入的 24 位位串从左到右开始每六位一组，共分成 4 组，每一组的值作为索引值在 64 个如上排列的字符序列中去查找，与索引值相同的那个字符将作为编码程序的输出字符。选择以上 64 个字符作为编码的输出字符是因为它们具有普遍可表示性，并且 Telnet 协议中有特殊意义的字符（例如“<CR>“,<LF>“”，IAC）被排除在 64 字符子集之外。

如果在消息尾部不能凑够 24 位，此时需经特殊处理。完全的编码数量总是在消息的尾部。当即将输入到编码程序中的位串不够 24 位时，以位“0”（从右侧）去填充，从而构成 6 位一组的整数。不需要代表其实际输入数据的那个输出字符则由字符“=”来代替。常规编码方式下每一个 24 位位串编码后输出的 4 个字符都是 64 字符子集中的字符。只有以下几种情况例外。(1)如果最终输入的位串位数是 24 的整数倍，那么编码输出的字符数是 4 的整数倍，且无字符“=”填充。（2）如果在需编码数据尾部只剩下一个 8 位串可供输入，那么在经过 16 个“0”位填充后经编码输出的 4 个字符中最后两个将是“=”。（3）如果在需编码的数据尾部只剩下一个 16 位串可供输入，那么在经过 8 个“0”位填充后经编码输出的 4 个字符中最后一个将是“=”。

执行者必须记住，对于 ADAT, MIC, CONF,ENC 指令和 63z 回应的 base 64 编码其长度是任意的。因此在处理之前应确保整行都被读取。对控制通道进行一些连续读取应该是必要的。只因为一个经 base 64 编码后的命令行太

长，服务器就将其拒绝，这一做法是不合适的。（假设发来的上下文可以用别的方式很好地解码）。

读取经 base 64 编码的命令及回应时一定要多加小心。

11. 安全方面的考虑

整个文档都与文件传输协议安全问题的有关。

由于安全的环境不能在两个服务器之间使用它们的工具来建立，因此与第三方的传输安全是不能通过这些扩展部分来保证的。（在服务器之间不存在用于传送 ADAT 信令的控制连接）。以后将会对这些问题进行更深的研究。

12. 感谢

（略——译者注）

13. 参考书目

[TELNET-SEC]

Borman, D., “Telnet认证和加密选项”，Work in Progress.

[RFC-1123]

Braden, R., “Internet主机必要条件——应用程序和支持”， STD 3, RFC 1123, 1989年10月。

[RFC-1421]

Linn, J., ″Internet电子邮件的加密增强：第一部分：消息加密和认证程序″,RFC 1421, 1993年2月。

14.作者地址

（略——译者注）

附录一：GSSAPI 规范

为最大限度的利用新的安全机制，新的安全机制以 GSSAPI 机制提供的方式而不是 FTP 安全机制实现是很好的选择。由于只需修改很少的代码或根本不需修改任何的代码，这将使得现有的 FTP 系统更容易地支持新机制。此外这些机制对于其他的协议如构建于 GSSAPI 之上的 IMAP 协议也是可用的，而机制的设计者不用为此作额外的说明或实现方面的工作。

使用 GSSAPI 接口实现的所有安全机制都应称作 GSSAPI 机制（因为 AUTH 指令）。如果服务器支持一个使用 GSSAPI 的安全机制，它必须返回回应码 334 以表明它所希望收到的下一个指令是 ADAT。

客户端必须从调用 GSS_Init_Sec_Context 函数开始认证过程，然后传递 0（开始时）及一个 targ_name 给 input_context_handle 函数开始认证过程。这里的 targ_name 与由 GSS_Import_Name 而来的 output_name 是等同的，

GSS_Input_Name 函数是与其输入是基于主机服务的 input_name_type 函数和参数为“ftp@hostname"的 input_name_string 函数一同被调用的。而函数 input_name_string 参数中的 hostname 表示服务器端所属主机的以小写字母给出的全名。（如果 input_name_string 执行失败，则客户端应以“host@hostname”作为函数的输入参数）由 GSS_Init_Sec_Context 产生的输出信令必须经 base 64 编码,然后作为 ADAT 指令的参数发送给服务器端。如果 GSS_Init_Sec_Context 返回 GSS_S_CONTINUE_NEEDED，那么客户端必须期待着 ADAT 指令的回应中同时返回一个信令。此信令将用作再次调用 GSS_Init_Sec_Context 的参数。这种情况如果 GSS_Init_Sec_Context 没有返回 output_token 那么服务器如上的 ADAT 指令返回的回应码就是 235。如果 GSS_Init_Sec_Context 返回 GSS_S_COMPLETE,客户端将不再期望服务器返回任何信令客户端必须认为服务器已被认证。

服务器端必须将客户端用 ADAT 传来的参数进行 base 64 解码，并将 resultant_token 作为输入信令传给 GSS_Accept_Sec_Context 函数，设置 acceptor_cred_handle 为 NULL（为了“使用缺省的信任信息）并将 0 传给 input_Context_handle （最初时）。如果 GSS_Accept_Sec_Context 返回一个输出信令，它必须经过 base 64 编码返回给客户，并在回应正文中包含“ADAT=base64 编码串“如果 GSS_Accept_Sec_Context 返回 GSS_S_COMPLETE,服务器对 ADAT 指令的回应码必须是 235。并且服务器必须认为客户端被认证。如果 GSS_Accept_Sec_Context 函数返回 GSS_S_CONTINUE_NEEDED,则对 ADAT 的回应码是 335。否则回应码是 535，回应正文部分描述的是错误信息。

对于 GSS_Init_Sec_Context 和 GSS_Accept_Sec_context 的 chan_binding 输入应使用客户端和服务器端各自的因特网地址作为发起者和接收者的地址，它们的地址类型都应是 GSS_C_AF_INET。没有别的应用程序数据被具体要求。

由于 GSSAPI 支持建立在安全环境下的访问，服务器对客户端认证时可以不需要真正确立其某个身份。

有关 MIC 指令，631 回应及安全文件传输的函数过程是

GSS_Wrap 是给发送者使用, 并且 conf_flag == FALSE

GSS_Unwrap 是给接收者使用

有关 ENC 指令，632 回应秘密性文件传输的函数过程是

GSS_Wrap 是给发送者使用，并且 conf_flag == TRUE
GSS_Unwrap 是给发送者使用。

CONF 指令及回应 633 不被支持。

客户端和服务器端都应检查 conf_avail 的值，以确定对方是否支持数据的保密保护服务。

当安全状态需要被复位时（当服务器收到了 AUTH 指令或者 REIN 指令）需要调用 GSS_Delete_Sec_Context 函数完成此功能。

附录二：Kerberos version 4 规范

使用 Kerberos V4 协议实现的安全机制应称为 KERBEROS V4 机制（因为 AUTH 指令）。如果服务器支持 KERBEROS_V4 机制，它必须返回回应码 334 以表明它期望收到的下一个命令是 ADAT。客户端必须通过调用 krb_mk_req(3)函数，并使用“ftp”的首要名称信息作为其参数来获得对于 Kerberos 主“ftp.hostname@realm”的准入信息（ticket,通常是数百个字节序列）。这里的参数“ftp"是和服务器所在主机的正规名称（由 krb_get_phost(3)返回）的第一部分及服务器的域名（由 krb_realmofhost(3)返回）和一个任意的校验和是等同的。此准入信息必须经过 base 64 编码然后作为一个 ADAT 指令的参数传递。

如果首要名“ftp”没有在 Kerberos 数据库中注册，那么客户端必须求助于主名“rcmd”（与 realm 相同的实例）然后服务器必须只能接受这些主名中的一个，而不能两者均接受。一般的如果在服务器的 srvtab 中有一个关于“ftp”的键，那么必须只能使用主名“ftp”，否则只能使用首要名“rcmd”。

服务器必须 base 64 解码由客户端经 ADAT 命令传来的参数，然后将结果作为参数传给 krb_rd_req(3)函数，服务器必须将从认证者传来的校验和加 1，而后将其转换为网络字节序（首先是最重要的字节）再用 krb_mk_safe(3)将其作标记，并进行 base 64 编码。如上过程成功的话，服务器返回回应码 235，并在其正文包含“ADAT=base 64 编码串“，如果失败了，服务器返回回应码 535。

收到了服务器端的回应码 235 后，客户端必须将其回应正文部分的 base 64 编码串进行解码，并将其网络字节序转换过来，将结果传给函数 krb_rd_safe(3) ，如果返回的校验和部分等于它以前发送的校验和的数值加 1，客户端必须认为服务器端已通过其认证。

有关 MIC 命令 631 回应及安全文件传输的过程函数是

krb_mk_safe(3) 用于发送者。
krb_rd_safe(3) 用于接收者。

有关 ENC 命令 632 回应及秘密文件传输的过程函数是

krb_mk_priv(3) 用于发送者。
krb_rd_priv(3) 用于接收者。

没有对于 CONF 命令和 633 回应的支持。

注意，对于如何进行关于数据完整性验证，保密保护等功能实现方式的协商，KerV4 的规范没有相关的条文约定，还有使用 ADAT 命令进行交换的安全数据，也没有对其进行传送的支持，而无论对方是否提供保密保护服务。

为了不超过 PBSZ 命令所规定的缓冲区尺寸，实现者必须切记，由使用 krb_mk_safe(3)和 krb_mk_priv(3) 函数时引起的透明文本缓冲区增长量应分别在 31 个字节和 26 个字节内。

完整的版权声明

（略——译者注）

原文：RFC 2228 《FTP Security Extensions》
译者：comehope 2002年5月
博客：http://www.comehope.com