FTP en encryptie

Johan Swenker

De letters FTP staan voor File Transfer Protocol.
File transfer doen we nog heel veel. Maar tegenwoordig doen we dat meestal over HTTP, of beter nog: over HTTPS.

Eigenlijk is dit verhaal daar een beetje mee begonnen.

Ondersteuning door Chrome gestopt
In **security.nl** stond onlangs een verhaal dat de Google-browser Chrome stopt met het ondersteunen van FTP.

https://www.security.nl/posting/674447/Google+wijst+Chrome-gebruikers+op+einde+van+FTP-ondersteuning

En toen vond ik het nodig om eens na te gaan wat ik nog aan FTP-software op mijn (Linux-)computer heb staan.

Het begin
FTP is een protocol uit de beginjaren van het internet. *De internet RFC[1] 959* uit oktober 1985 begint met een historisch overzicht, over file transfer vanaf 1971:
https://tools.ietf.org/html/rfc959

Encryptie was in die begintijd nog niet gebruikelijk. Alle data, en met name ook het wachtwoord, werd zonder enige vorm van versleuteling overgedragen van client naar server.

Waarom nu stoppen?
Dat is nu dan ook de reden om te stoppen met (het ondersteunen van) FTP. Maar lees even door, zo eenvoudig is het niet. Een van mijn taken op mijn werk is het beoordelen van firewallverzoeken.

Bij niet-versleutelde protocollen, zoals telnet, FTP of http, is mijn standaard reactie: niet toegestaan.
Maar is dat wel terecht? Zeker als je dat vergelijkt met SMTP (e-mail) dat in principe ook niet versleuteld is.

SSL/TLS
Bij e-mail is mijn verweer dat SMTP overschakelt naar SSL/TLS zodra dat mogelijk is.

Dan maar eens onderzoeken: misschien kan FTP ook overschakelen naar TLS? TLS is een doorontwikkeling van SSL.
De eerste versie van TLS zou je zelfs kunnen beschouwen als SSL 3.1!
Het grootste verschil is, dat TLS in een RFC-gestandaardiseerd is, terwijl SSL een de-factostandaard van Netscape is.

Op onderzoek
Op mijn laptop gebruik ik Linux, meer in het bijzonder Ubuntu. Bij Ubuntu, en ook bij alle andere op Debian gebaseerde distributies, kun je met de commando's 'apt' en 'apt-cache' opvragen wat voor softwarepakketten er in de distributie zitten.

Het commando 'apt-cache search ftp ssl' laat alle pakketten zien die FTP en SSL in de beschrijving hebben staan. Dat zijn er te veel; ik ga alleen iets vertellen over ftp-ssl, ftpd-ssl, pure-ftpd en filezilla.

FileZilla is een grafische FTP-client die ook voor Windows beschikbaar is.

De beide pakketten met de letter 'D' in de naam zijn FTP-servers, leuk voor hobbyisten en systeembeheerders.

FileZilla Server
Als je onder Windows wilt spelen met een FTP-server die SSL/TLS ondersteunt, kijk dan eens bij FileZilla Server:
https://filezilla-project.org

Ftp-ssl
Als je niet bang bent voor de commandoregel, dan is **ftp-ssl** bruikbaar, en dan is ook het installeren eenvoudig:
sudo apt install ftp-ssl

Ftp-ssl werkt als elke andere FTP-client die op de commandoregel werkt. Bij het opstarten kun je SSL/TLS-parameters opgeven, daarna niet meer. Twee bijzondere opties zijn: het eisen van TLS, of juist het verbieden daarvan.

Het eisen van TLS werkte op mijn thuisnetwerkje als volgt:

```
$ ftp-ssl -z secure nano
Connected to nano.swenker.org.
220 swenker.xs4all.nl FTP server (Version 6.4/OpenBSD/
    Linux-FTPd-0.17) ready.
Name (nano:johan):
500 'AUTH TLS': command not understood.
500 'AUTH SSL': command not understood.
SSL not available
Login failed.
ftp>
```

Perfect toch? Als de server geen TLS spreekt en ook geen SSL, dan zorgt ' z secure' ervoor dat het inloggen faalt, zelfs voordat het wachtwoord gevraagd is. Het wachtwoord zal dus nooit onversleuteld verstuurd worden.

TLS-certificaat
Een website die via https benaderd wordt, heeft een certificaat nodig. Elke server die het netwerkverkeer versleutelt met TLS heeft een certificaat nodig.

Voor sommige FTP-experimenten heb ik een certificaat van OpenVPN misbruikt. De beschrijving van **pure-ftpd** vertelt hoe je met OpenSSL een self-signed certificaat maakt[2].

Ftpd-ssl
Over ftpd-ssl valt weinig spannends te vertellen. Nadat ik in */etc/ftpd-ssl/ftpd.pem* een geldig certificaat had geplaatst, werkte het gewoon. Overigens kent ook ftpd-ssl de optie *-z secure* om TLS af te dwingen.

1. De definitie van het hele internet is in standaarden vast gelegd. Die standaarden heten RFC's. Deze naam stamt uit de tijd dat het nog echt een 'request for comment' was.
   Je kunt gaan zoeken naar RFC's die op 1 april uitgegeven zijn. Dan vind je onder andere RFC 1149 en 2322, of zelfs een hele lijst op **https://www.cs.hmc.edu/~awooster/joke_rfcs.html**
2. `openssl req -x509 -nodes -days 7300 -newkey rsa:2048 -sha256 -keyout /etc/ssl/private/pure-ftpd.pem -out /etc/ssl/private/pure-ftpd.pem`

Pure-ftpd
Een standaard geïnstalleerde pure-ftpd heeft TLS gedisabled. Dat is met *ftp-ssl -z secure* makkelijk aan te tonen. Nadat ik pure-ftpd opnieuw opgestart had, maar nu met de optie *-Y 3* werkte de versleutelde verbinding:

```
$ ftp-ssl -z secure localhost
Connected to localhost.
220------ Welcome to Pure-FTPd [privsep] [TLS] ------
220-You are user number 1 of 50 allowed.
220-Local time is now 22:03. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of
    inactivity.
Name (localhost:johan):
234 AUTH TLS OK.
[SSL Cipher ECDHE-RSA-AES128-GCM-SHA256]
200 PBSZ=0
200 Data protection level set to "private"
[Encrypted data transfer.]
331 User johan OK. Password required
Password:
230 OK. Current directory is /home/johan
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Controle versleuteling
Maar is het verkeer echt versleuteld?

Dat is aan te tonen door al het netwerkverkeer af te vangen en te analyseren. Met Wireshark, wat zowel onder Linux als onder Windows beschikbaar is, kun je al het netwerkverkeer afvangen, analyseren en in een bestand zetten.

Onder Linux gebruik ik vaak het programma **strings** om te kijken of er leesbare tekst in een bestand staat:

```
$ strings pure-ftpd.pcap
b?:,
b?:,
220------ Welcome to Pure-FTPd [privsep] [TLS] ------
220-You are user number 1 of 50 allowed.
220-Local time is now 22:11. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of
    inactivity.
AUTH TLS
b?:-
b?:-
234 AUTH TLS OK.
b?:-
DOWNGRD
      Groningen1
201016192357Z
401011192357Z0.1
      Groningen1
y7}p
$_Epo
U_6_
CXAq
CQ'-
)PO}
```

De server geeft een welkomstboodschap. Daarna zien we het woord Groningen, dat onderdeel is van het TLS certificaat, en de rest is onsamenhangende binaire bagger. Met name het wachtwoord is niet als klare tekst terug te vinden.

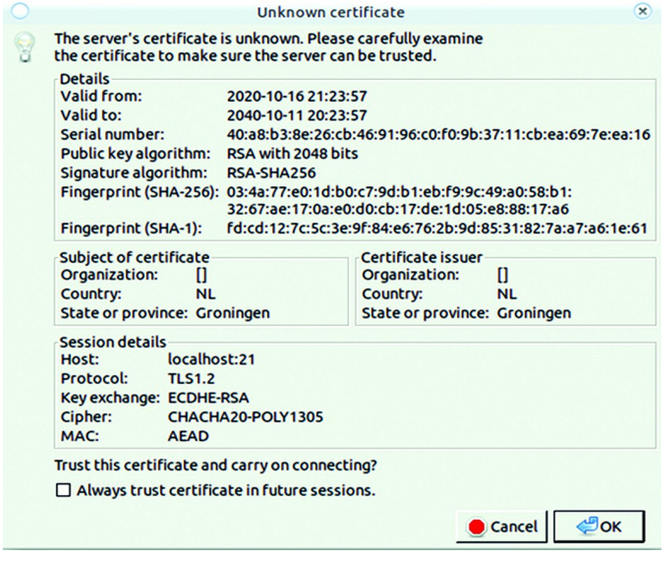
Nog een controle
Ik wil eigenlijk zeker weten dat pure-ftpd TLS afdwingt, dus dat al het verkeer daadwerkelijk versleuteld is.

Dat is uit te proberen. Met de optie *-z nossl* zal ftp-ssl geen TLS meer gebruiken. De reactie van pure-ftpd is dat inloggen niet lukt:

```
$ ftp-ssl -z nossl localhost
Connected to localhost.
220------ Welcome to Pure-FTPd [privsep] [TLS] ------
220-You are user number 2 of 50 allowed.
220-Local time is now 22:03. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of
    inactivity.
Name (localhost:johan):
421-Sorry, cleartext sessions and weak ciphers are
    not accepted on this server.
421 Please reconnect using TLS security mechanisms.
Login failed.
No control connection for command: Success
ftp>
```

Dat 'Success' is natuurlijk niet geheel terecht.

FileZilla
Zoals in onderstaand screenshot te zien is, was FileZilla niet zo heel blij met mijn self-signed certificaat. Dat is uiteraard op te lossen door het certificaat te laten onderteken door een van de vele bekende Certificaat Authoriteiten; bijvoorbeeld bij: https://letsencrypt.org



Alternatieven
Onder Windows: Een review door André Reinink van het reeds eerder genoemde FileZilla, staat in SoftwareBus 2019-3:
https://www.compusers.nl/sites/default/files/swb-jaargangen/2019/2019-3/SwB20193_Review_FileZilla.pdf
WinSCP is van harte aanbevolen voor alle Windows-gebruikers. Het werkt prima samen met putty en pageant. Een review door Hans Vosman van WinSCP staat in SoftwareBus 2018-5:
https://www.compusers.nl/sites/default/files/swb-jaargangen/2018/2018-5/SwB20185_Review_WinSCP.pdf
Onder Linux: **scp** en **sftp**, beide onderdeel van OpenSSH, is de standaard manier om onder Linux versleutelde file transfer te doen.

Toekomst van FTP
Of ik zelf FTP nog gebruik? Nee, vrijwel niet. Op mijn thuisnetwerk gebruik ik meestal *scp* om bestanden heen en weer te kopiëren.
Er was een tijd dat ik een website met FTP moest updaten. Die website wordt nu door iemand anders beheerd, dus dat updateprobleem ben ik kwijt.
Op mijn thuisnetwerk heb ik een IoT-apparaat dat ik alleen met FTP kan aanspreken. Maar het meeste FTP-gebruik dat ik doe is naar de test-directory van een download-server van internet provider XS4ALL:
`wget -O /dev/null ftp://dl.xs4all.nl/pub/test/10MiB.bin`
en dat laat zien dat ik echt maar een 20 Mbps internetverbinding heb.