#SHAREorg

SHARE
Technology · Connections · Results

Securing Your FTP Transmissions
Session #11576

Colin van der Ross
Software Diversified Services

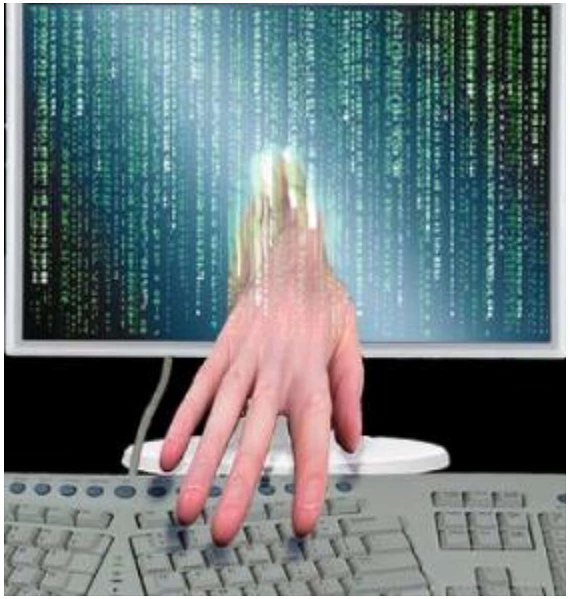
February 5th 2013

SHARE in San Francisco 2013

Agenda

- FTP and Compliance
- FTP Basics
- Securing Your FTP Transmissions
- Summary

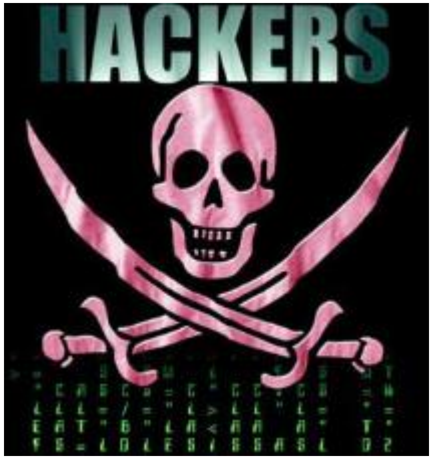
Data Compromises Are On The Increase



- 855 incidents, 174 million compromised records in 2012
- 98% stemmed from external Agents (+6%)
- 4% implicated Internal Employees (-13%)
- < 1 % committed by Business partners (<>)
- 58% of all data theft tied to activist groups

Source: Verizon 2012 Data Breach Investigation Report

How Did these Breaches Occur ?



- 81% used some form of Hacking (+31%)
- 69% incorporated Malware (+20%)
- 10% involved physical attacks (-19%)
- 7% employed Social Tactics (-4%)
- 5% resulted from privilege misuse (-12%)

Source: Verizon 2012 Data Breach Investigation Report

What Commonalties Exist ?

- 79% of victims were targets of opportunity (-4%)
- 96% of attacks were not highly difficult (+4%)
- 94% of all data compromised involved servers (+18%)
- 85% of breaches took weeks or more to discover (+6%)
- 92% of incidents were discovered by a third party (+6%)
- 97% of breaches were avoidable through simple or intermediate controls (+1%)
- 96% of victims subject to PCI DSS had not achieved compliance (+7%)

Source: Verizon 2012 Data Breach Investigation Report

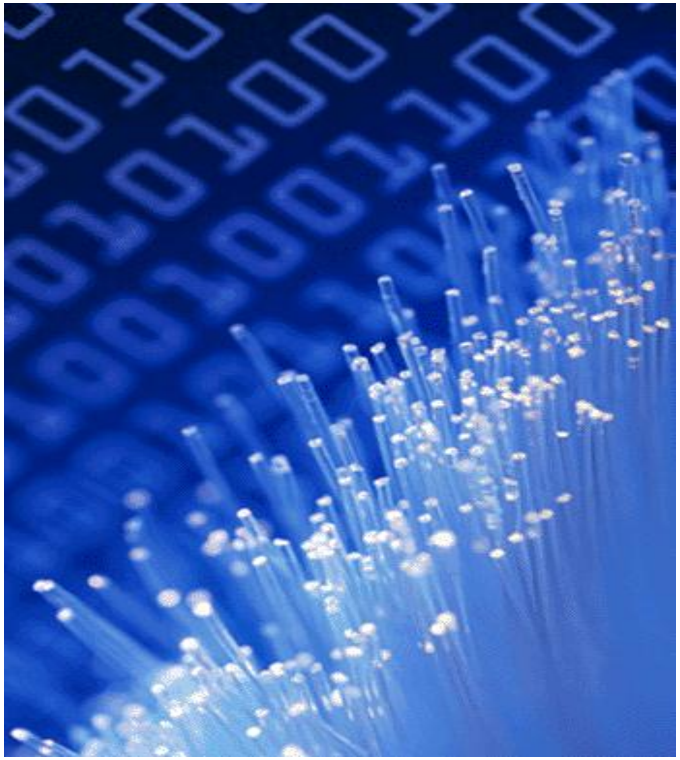
High Penalties

- Hard Costs
- Loss of Trade Secrets
- Litigation and Liability
  - FISMA (Federal Information Security Management Act)
  - Sarbanes-Oxley Act
  - PCI
  - Data Protection Act (UK)
  - Company Policies and Practices
- Company Reputation
- Identity Theft

Truth is ….

- The FTP technology is robust and has endured, but ...
- …It fails to meet all requirements of the modern business enterprise
  - FTP needs a little help

FTP is….



- Primary tool for moving data
- Unifies data transfer across today's business enterprise
- More data in more forms and locations than ever, and the need to share it *only* among authorized users
  - Mainframes and Servers
  - Desktop computers
  - Laptop computers
  - Wireless networks
  - Handheld devices

FTP Now

- Integrated part of daily operations
- Data transfer is automated and runs unattended

- Data moves throughout the *global* enterprise
- Timing is critical for data movement across the enterprise
- Operations staff must be sure that the job has been done correctly

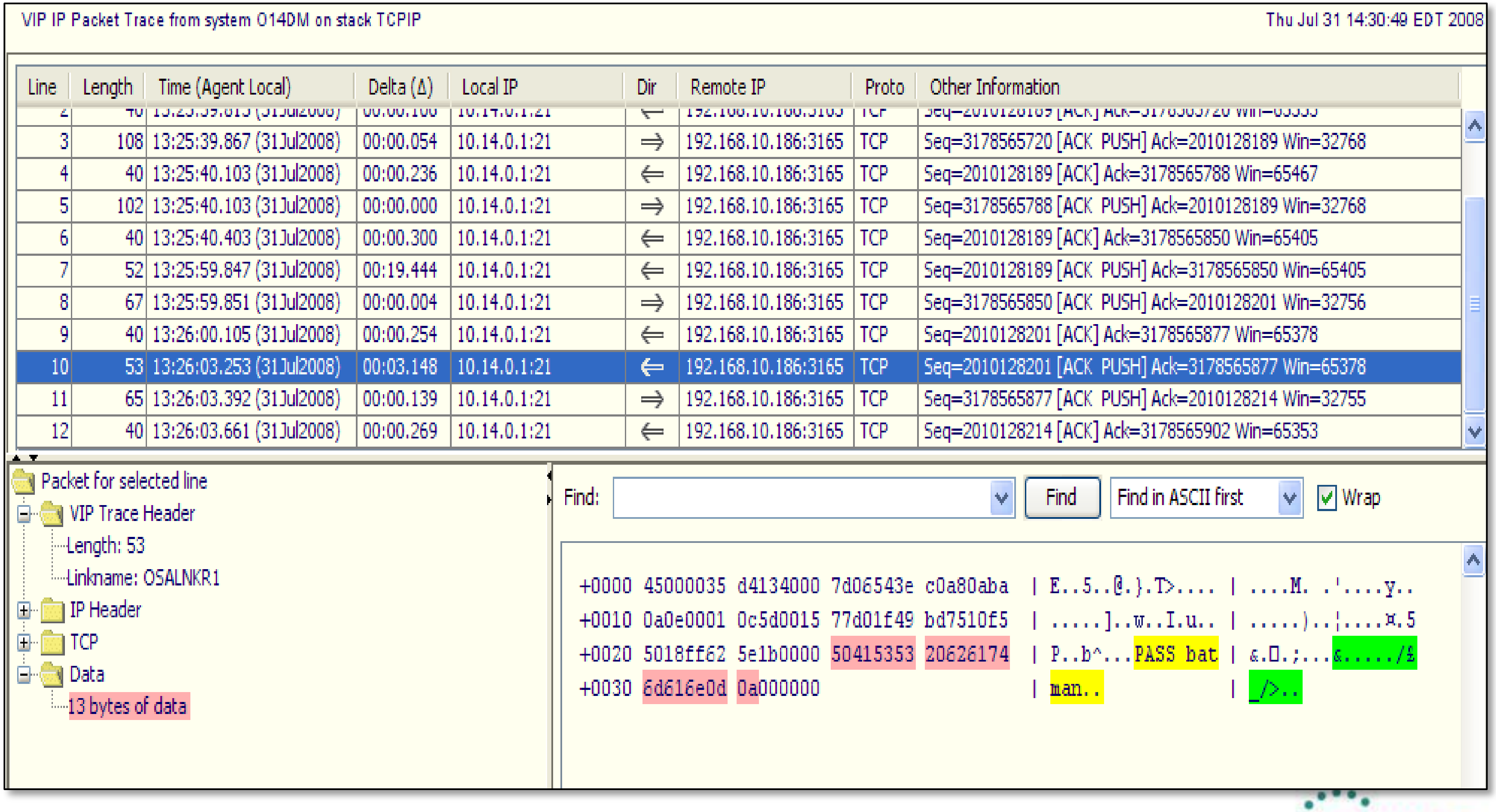
So Why Is FTP So Vulnerable ?

- Security is one of the major opponents of FTP
- FTP over the Internet has security implications
- User name and Passwords can be transmitted in the clear
- Anyone along the FTP path can “sniff” the User Name and password
- Passwords can be used to gain access to systems

Why is FTP so Vulnerable ?

- Anyone with ***Read*** access, also has ***“Transfer Out”*** access
- Read Clear Text Exposure
  - Password interception
  - Eavesdropping
- Hijacking
  - “Man in the middle”
  - Connection “hijack”
  - Spyware
- Wireless Connectivity
  - Can open portal behind firewall

FTP Packet Trace Example



VIP IP Packet Trace from system O14DM on stack TCPIP

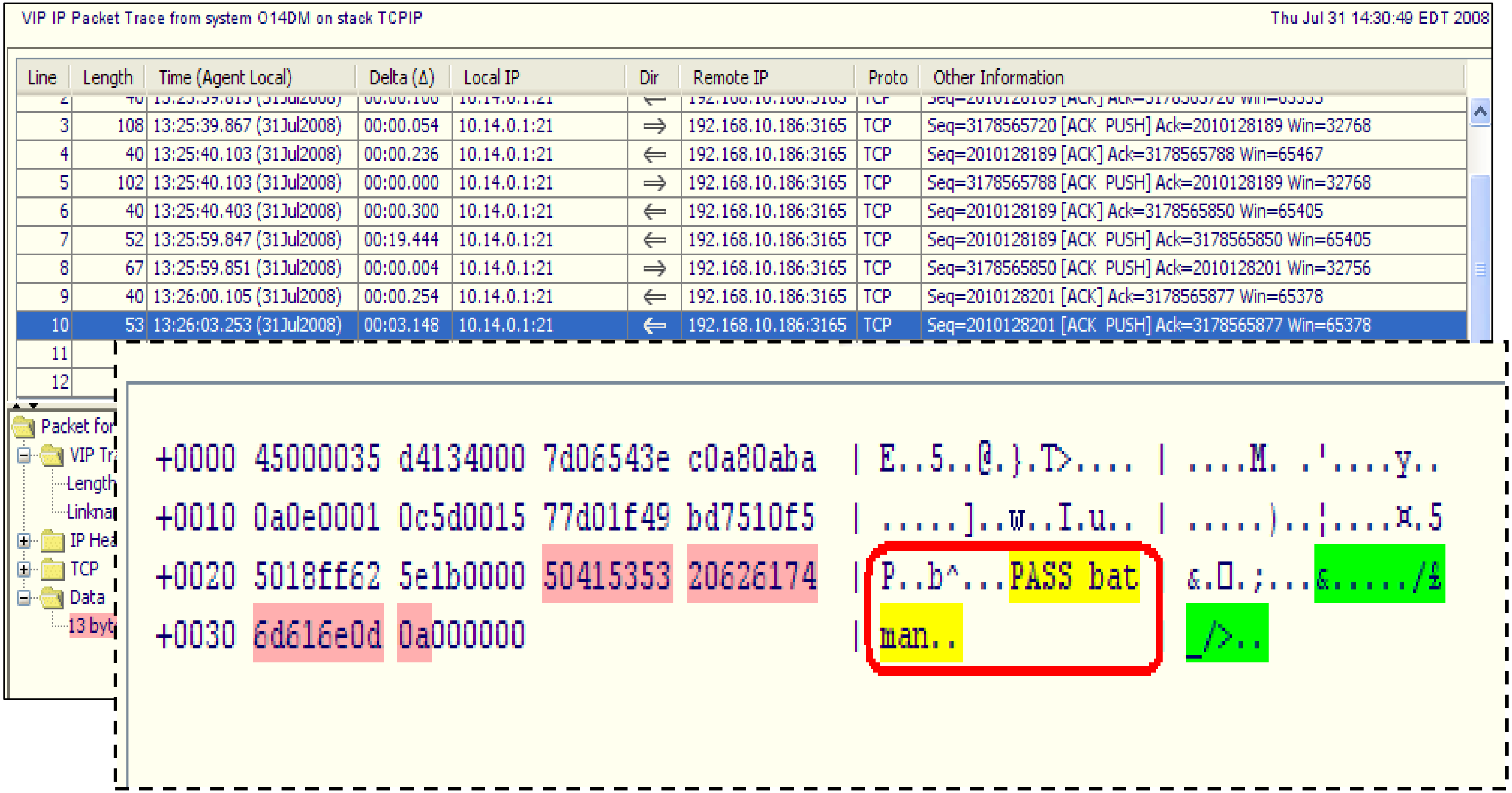
Thu Jul 31 14:30:49 EDT 2008

Line	Length	Time (Agent Local)	Delta (Δ)	Local IP	Dir	Remote IP	Proto	Other Information
2	40	13:25:39.813 (31Jul2008)	00:00.106	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128189 [ACK] Ack=3178565720 Win=65535
3	108	13:25:39.867 (31Jul2008)	00:00.054	10.14.0.1:21	⇒	192.168.10.186:3165	TCP	Seq=3178565720 [ACK PUSH] Ack=2010128189 Win=32768
4	40	13:25:40.103 (31Jul2008)	00:00.236	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128189 [ACK] Ack=3178565788 Win=65467
5	102	13:25:40.103 (31Jul2008)	00:00.000	10.14.0.1:21	⇒	192.168.10.186:3165	TCP	Seq=3178565788 [ACK PUSH] Ack=2010128189 Win=32768
6	40	13:25:40.403 (31Jul2008)	00:00.300	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128189 [ACK] Ack=3178565850 Win=65405
7	52	13:25:59.847 (31Jul2008)	00:19.444	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128189 [ACK PUSH] Ack=3178565850 Win=65405
8	67	13:25:59.851 (31Jul2008)	00:00.004	10.14.0.1:21	⇒	192.168.10.186:3165	TCP	Seq=3178565850 [ACK PUSH] Ack=2010128201 Win=32756
9	40	13:26:00.105 (31Jul2008)	00:00.254	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128201 [ACK] Ack=3178565877 Win=65378
10	53	13:26:03.253 (31Jul2008)	00:03.148	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128201 [ACK PUSH] Ack=3178565877 Win=65378
11	65	13:26:03.392 (31Jul2008)	00:00.139	10.14.0.1:21	⇒	192.168.10.186:3165	TCP	Seq=3178565877 [ACK PUSH] Ack=2010128214 Win=32755
12	40	13:26:03.661 (31Jul2008)	00:00.269	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128214 [ACK] Ack=3178565902 Win=65353

Packet for selected line
- VIP Trace Header
  - Length: 53
  - Linkname: OSALNKR1
- IP Header
- TCP
- Data
  - 13 bytes of data

```
+0000 45000035 d4134000 7d06543e c0a80aba  | E..5..@.}.T>.... | ....M. .'....y..
+0010 0a0e0001 0c5d0015 77d01f49 bd7510f5  | .....]..w..I.u.. | .....)..¦....¤.5
+0020 5018ff62 5e1b0000 50415353 20626174  | P..b^...PASS bat | &.□.;...&...../£
+0030 6d616e0d 0a000000                    | man..            | _/>..
```

Passwords are in the CLEAR



VIP IP Packet Trace from system O14DM on stack TCPIP — Thu Jul 31 14:30:49 EDT 2008

Line	Length	Time (Agent Local)	Delta (Δ)	Local IP	Dir	Remote IP	Proto	Other Information
2	40	13:25:39.813 (31Jul2008)	00:00.100	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128189 [ACK] Ack=3178565720 Win=65535
3	108	13:25:39.867 (31Jul2008)	00:00.054	10.14.0.1:21	⇒	192.168.10.186:3165	TCP	Seq=3178565720 [ACK PUSH] Ack=2010128189 Win=32768
4	40	13:25:40.103 (31Jul2008)	00:00.236	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128189 [ACK] Ack=3178565788 Win=65467
5	102	13:25:40.103 (31Jul2008)	00:00.000	10.14.0.1:21	⇒	192.168.10.186:3165	TCP	Seq=3178565788 [ACK PUSH] Ack=2010128189 Win=32768
6	40	13:25:40.403 (31Jul2008)	00:00.300	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128189 [ACK] Ack=3178565850 Win=65405
7	52	13:25:59.847 (31Jul2008)	00:19.444	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128189 [ACK PUSH] Ack=3178565850 Win=65405
8	67	13:25:59.851 (31Jul2008)	00:00.004	10.14.0.1:21	⇒	192.168.10.186:3165	TCP	Seq=3178565850 [ACK PUSH] Ack=2010128201 Win=32756
9	40	13:26:00.105 (31Jul2008)	00:00.254	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128201 [ACK] Ack=3178565877 Win=65378
10	53	13:26:03.253 (31Jul2008)	00:03.148	10.14.0.1:21	⇐	192.168.10.186:3165	TCP	Seq=2010128201 [ACK PUSH] Ack=3178565877 Win=65378
11								
12								

```
+0000 45000035 d4134000 7d06543e c0a80aba | E..5..@.}.T>.... | ....M. .'....y..
+0010 0a0e0001 0c5d0015 77d01f49 bd7510f5 | .....]..w..I.u.. | .....)..¦....¤.5
+0020 5018ff62 5e1b0000 50415353 20626174 | P..b^...PASS bat   &.□.;...&...../£
+0030 6d616e0d 0a000000                   | man..              _/>..
```

Review of FTP Concepts

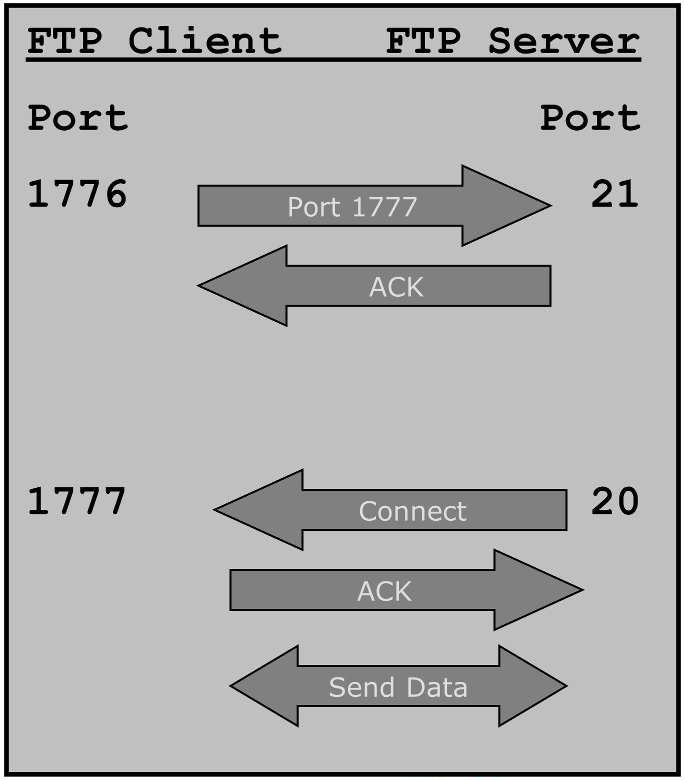
- FTP Ports
  - Control port
  - Data transfer port
- FTP Types
  - *Active* versus *Passive*
  - *Proxy*
- FTP Modes for data
- FTP Exits
- FTP SMF Records
  - Record of activity

FTP Well-Known Ports

- Control Connection
  - Well-known port 21
  - Long-running connection
  - Transmits instructions
- Data Connection
  - Well-known port 20 for *active* FTP
  - Ephemeral port for *passive* FTP
  - One for each file transferred

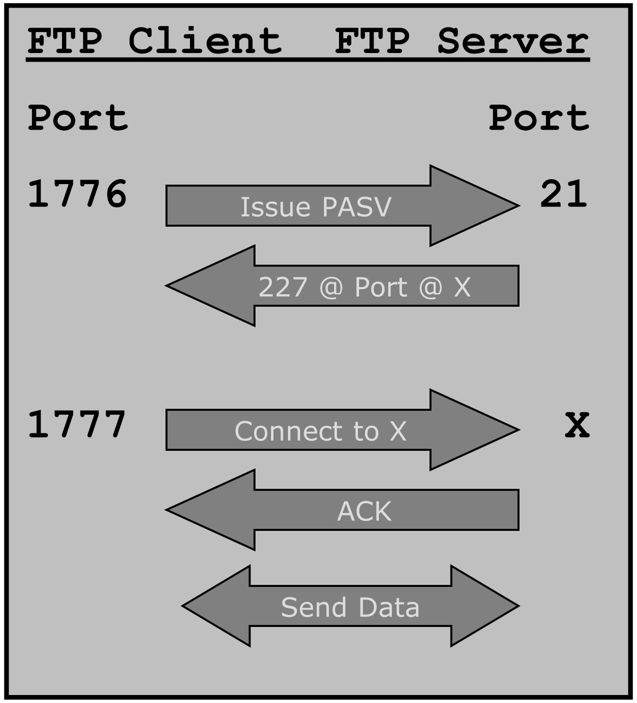
FTP Types – Active, Passive, Proxy

- Active FTP
  - Client connects to server port 21
  - Client starts a listening port and sends command to FTP server
  - Server initiates data connection to the client
  - Server connects to client's data port from its local port 20



FTP Types – Active, Passive, Proxy

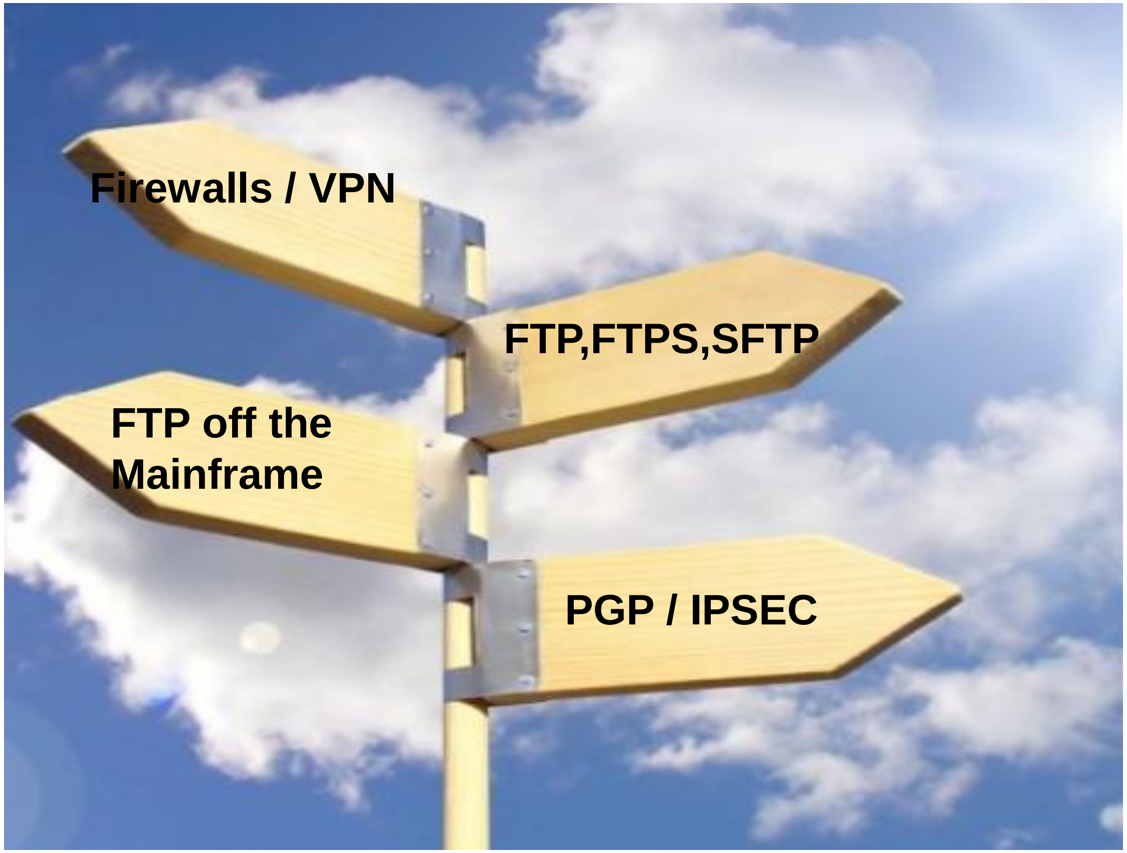
- Passive FTP, *Firewall Friendly*
  - Client initiates data connection with server
  - Client first contacts server port 21
  - Client issues *PASV* command
  - Server opens new, random, data port;
    informs client
    (data port greater than 1024)
  - Client connects to new data port



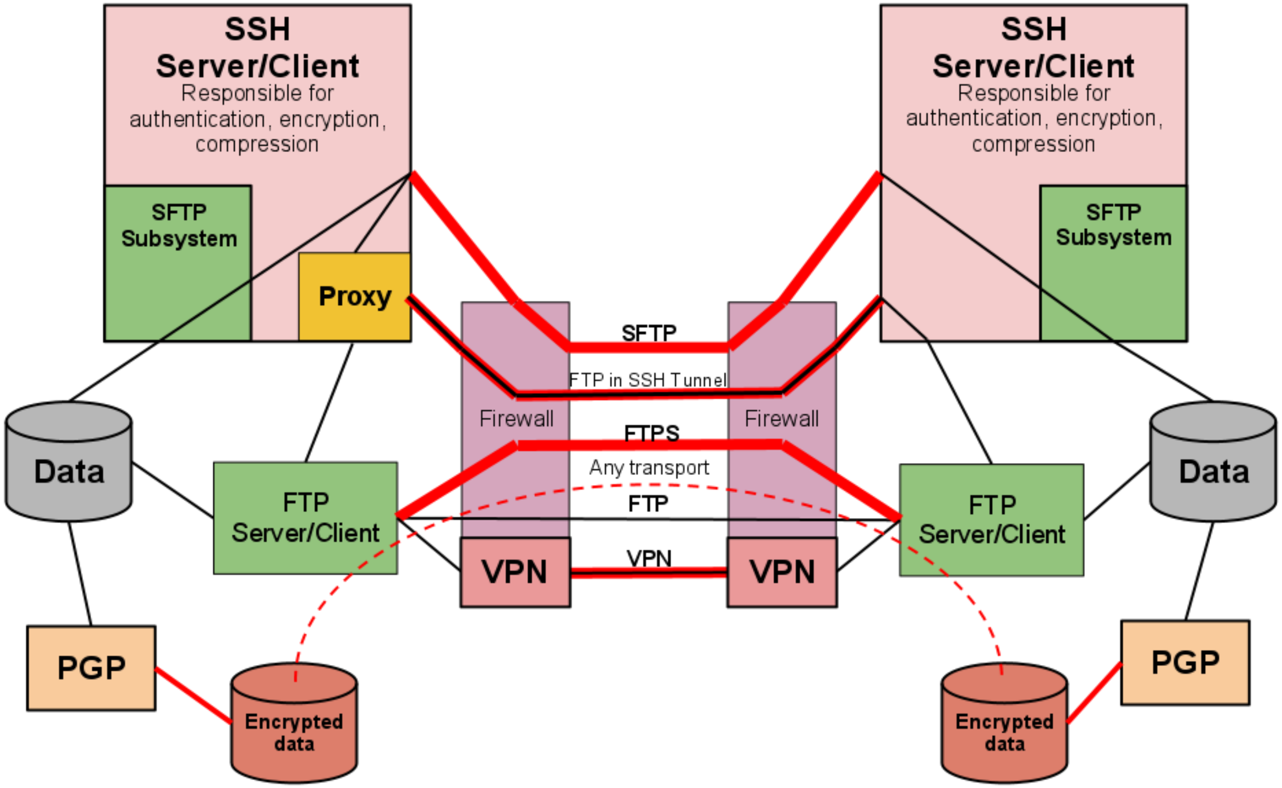
FTP Types – Active, Passive, Proxy

- Proxy FTP
  - PROXY command allows an FTP subcommand to be issued on a secondary control connection
  - FTP client connects simultaneously to two FTP servers
  - Client can then initiate a data connection between the two servers
  - Files are transferred between the servers on ephemeral ports

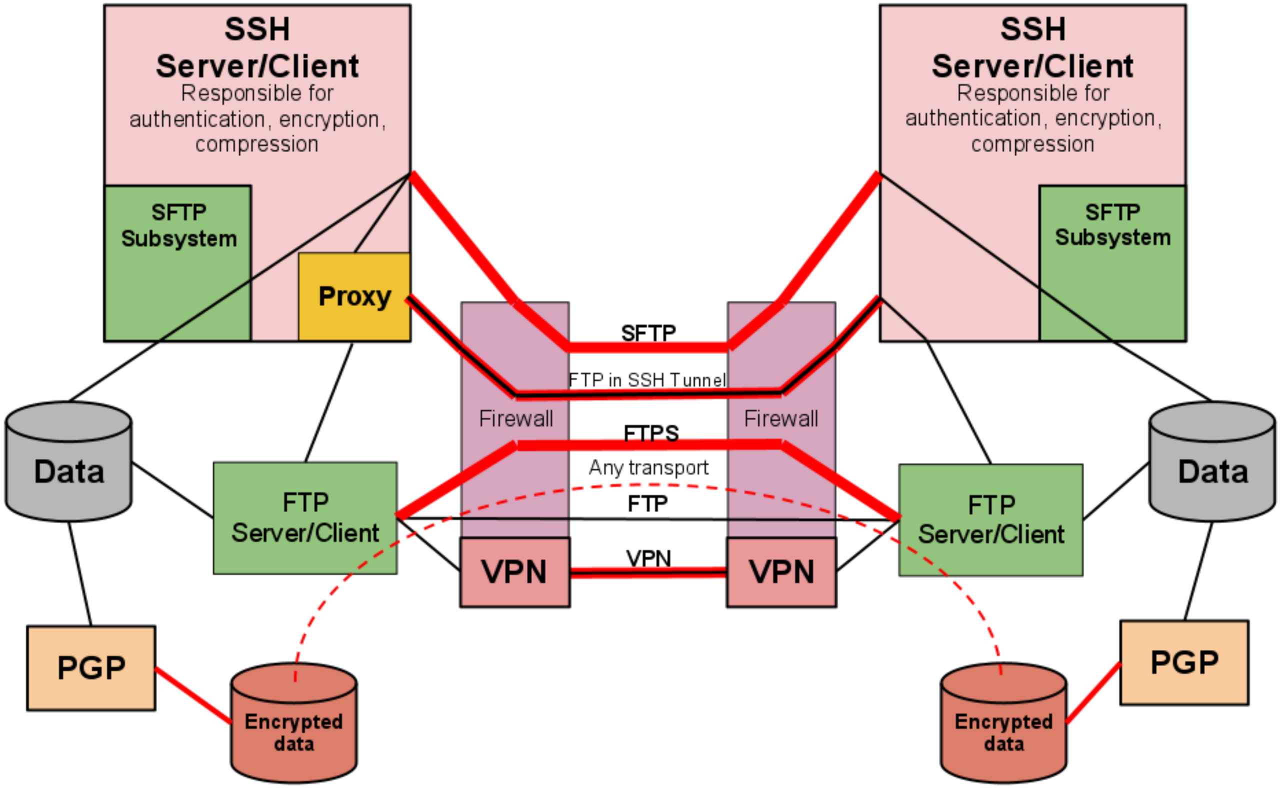
Securing your FTP Transmissions



Options to Secure your FTP Data

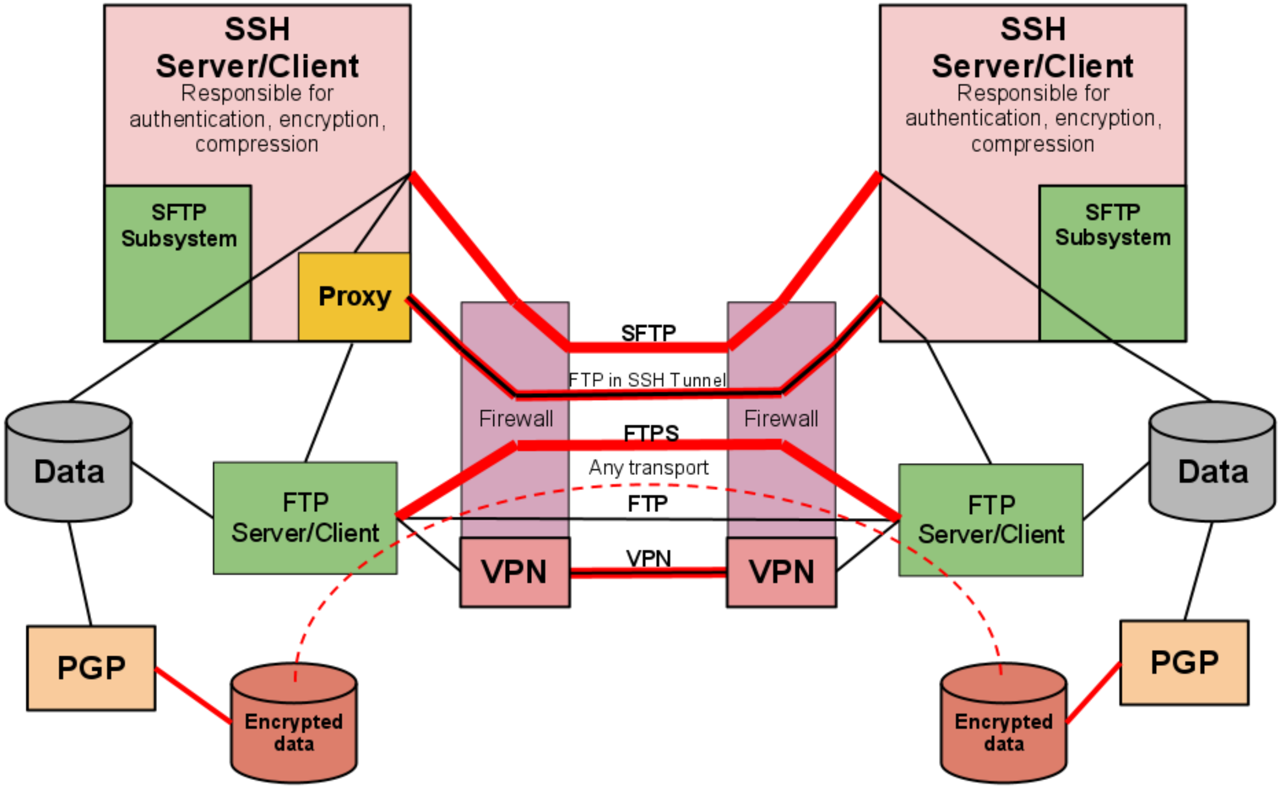


Options to Secure your FTP Data



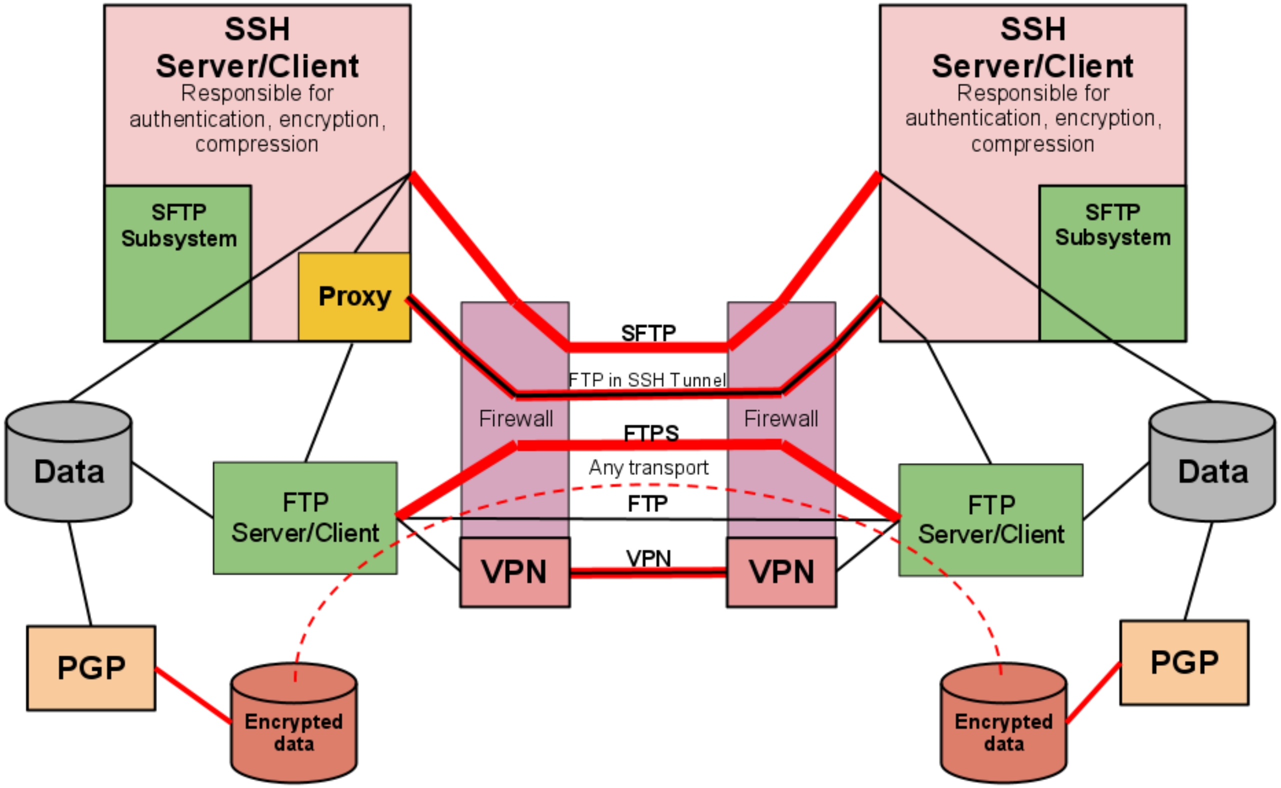
- What are some alternatives

Options to Secure your FTP Data



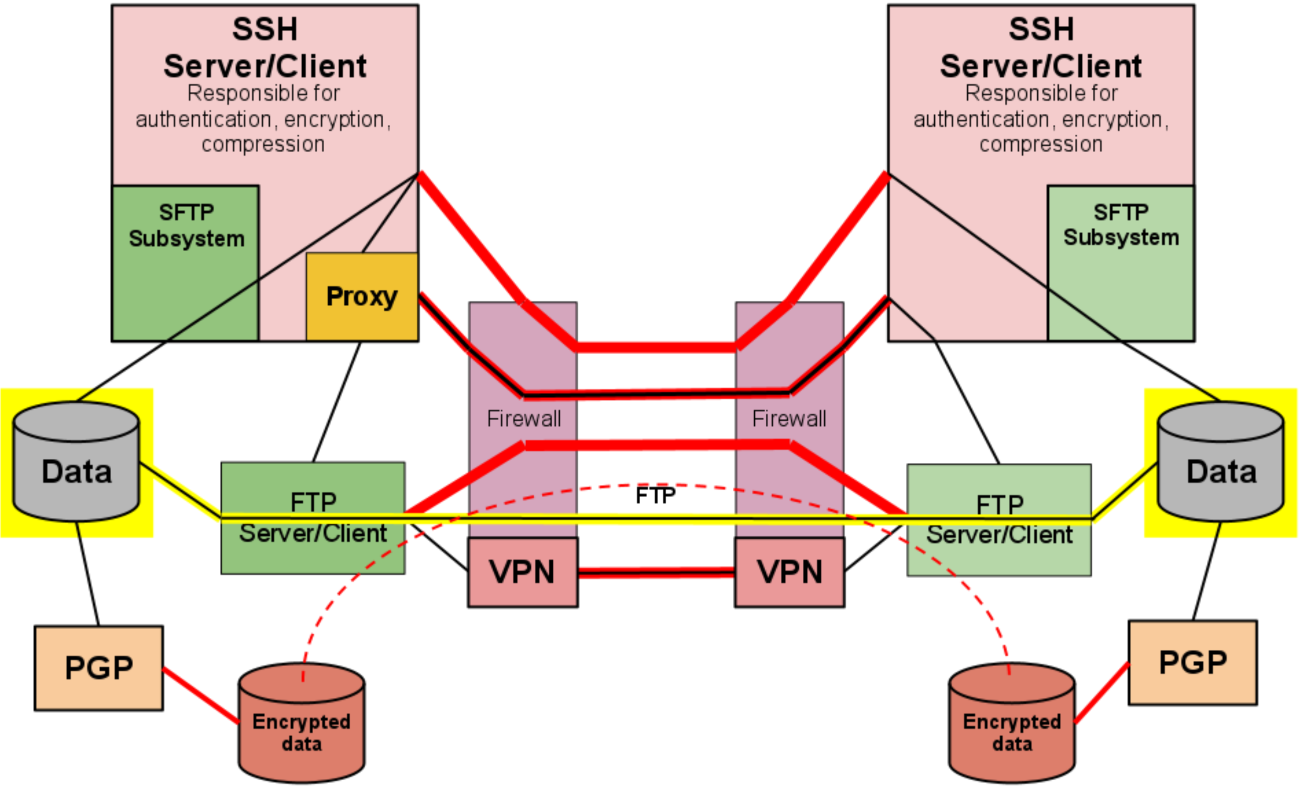
- What are some alternatives
- Why or why not use the methods and tools

Options to Secure your FTP Data



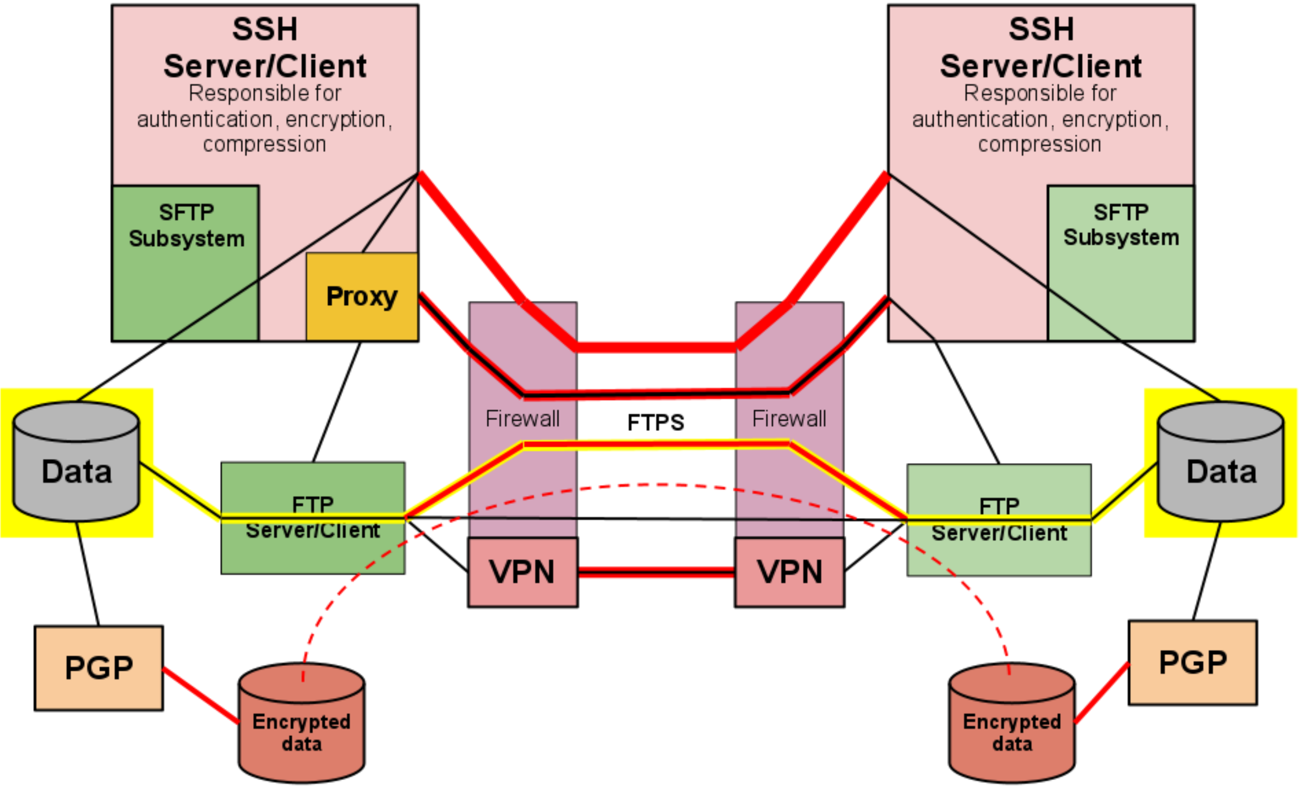
- What are some alternatives
- Why or why not use the methods and tools
- When is a good time to use the solution

FTP (File Transfer Protocol)



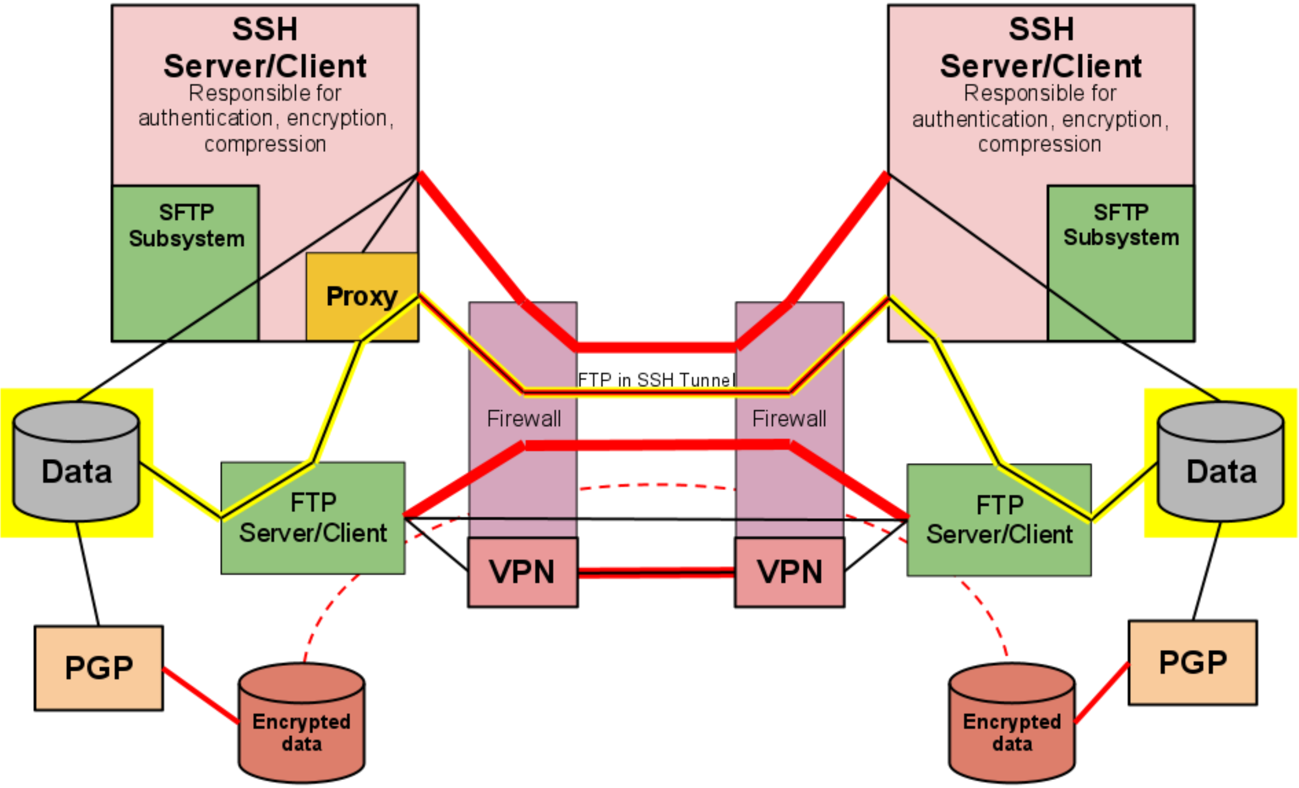
- FTP

FTPS (FTP over SSL)



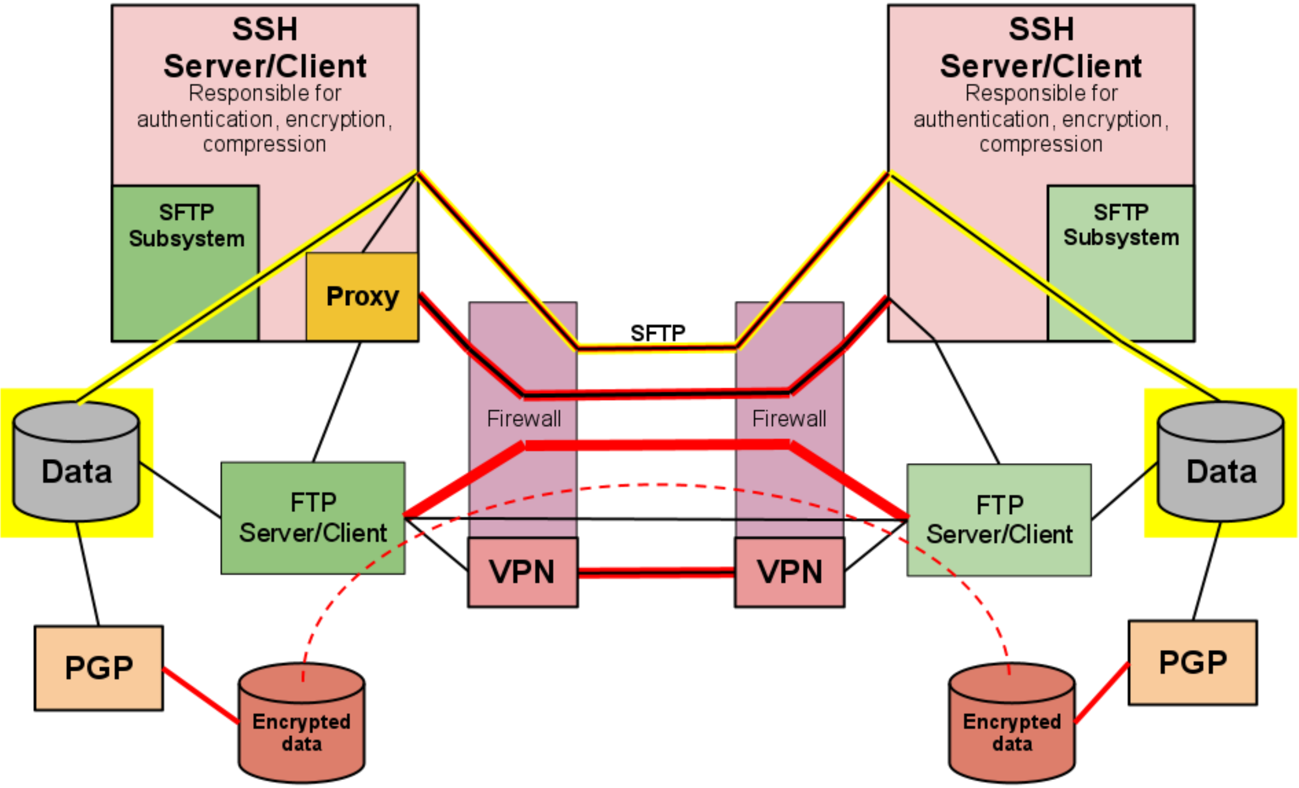
- FTP
- FTPS

FTP over SSH Tunnel



- FTP
- FTPS
- FTP over SSH Tunnel

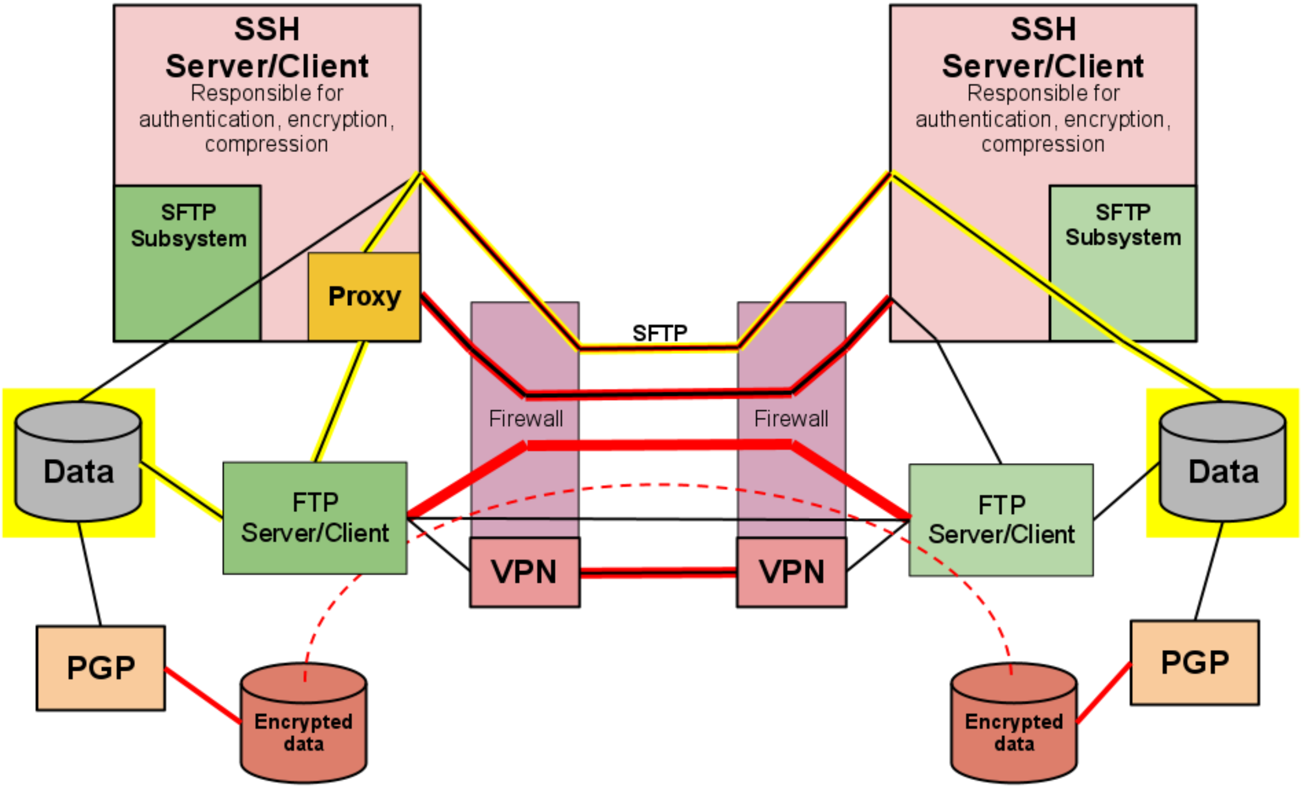
SFTP (SSH Secure FTP)



- FTP
- FTPS
- FTP over SSH Tunnel

- SFTP

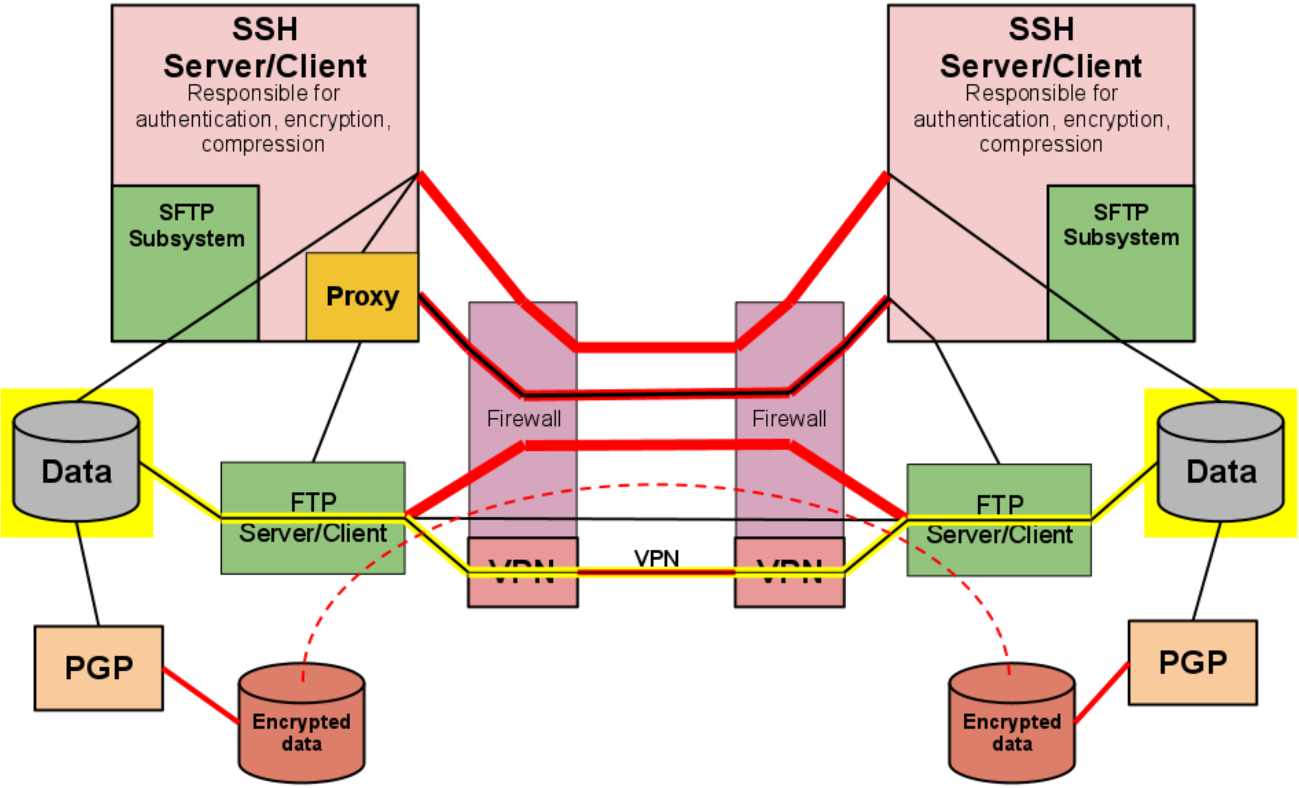
FTP/SFTP Hybrid



- FTP
- FTPS
- FTP over SSH Tunnel

- SFTP
- FTP to SFTP

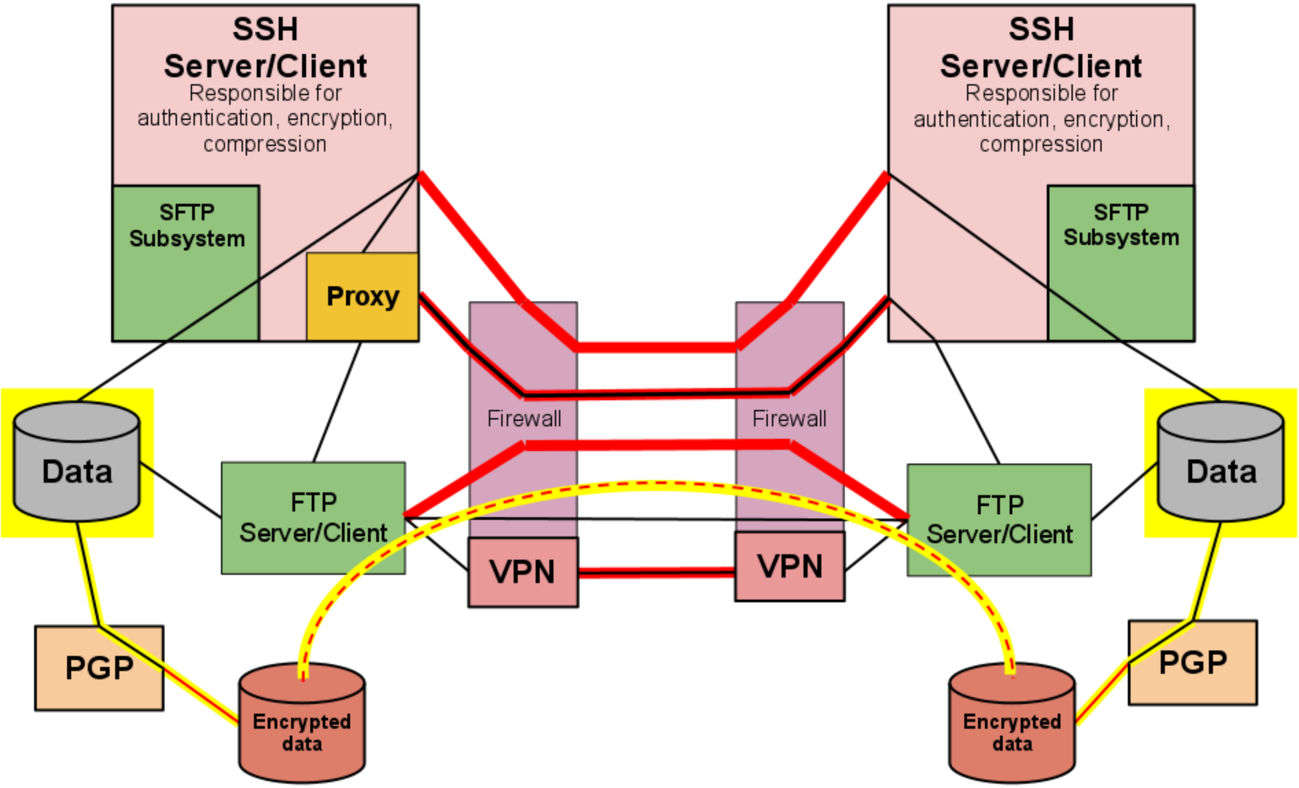
VPN (Virtual Private Network)



- FTP
- FTPS
- FTP over SSH Tunnel

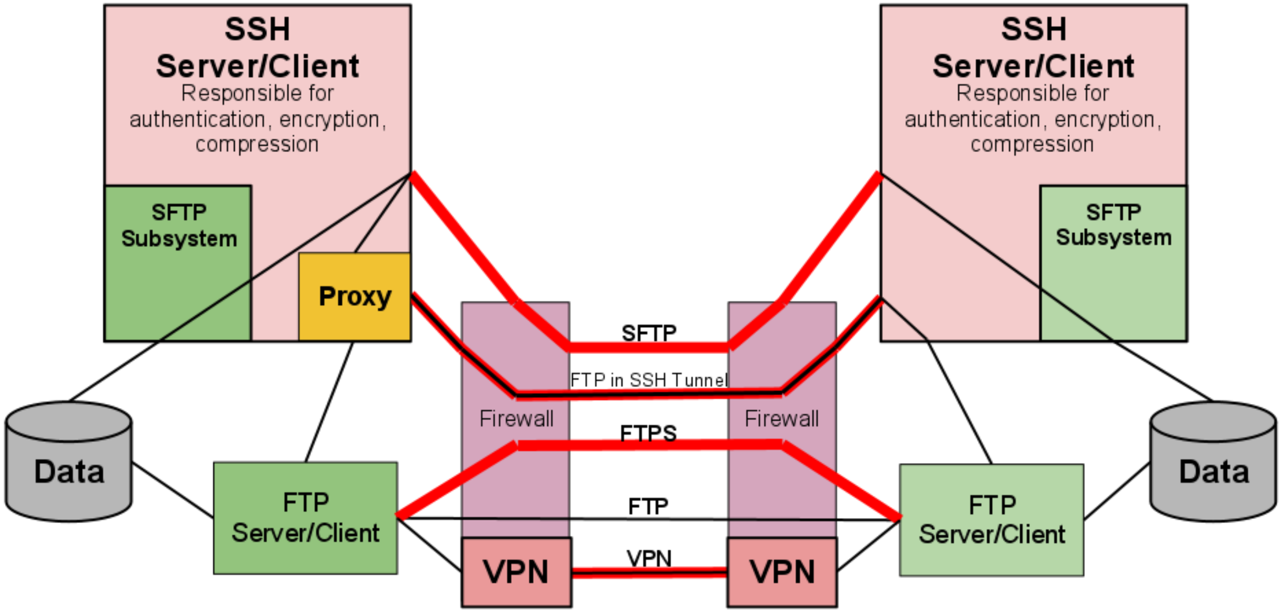
- SFTP
- FTP to SFTP
- VPN

PGP (Data at rest)



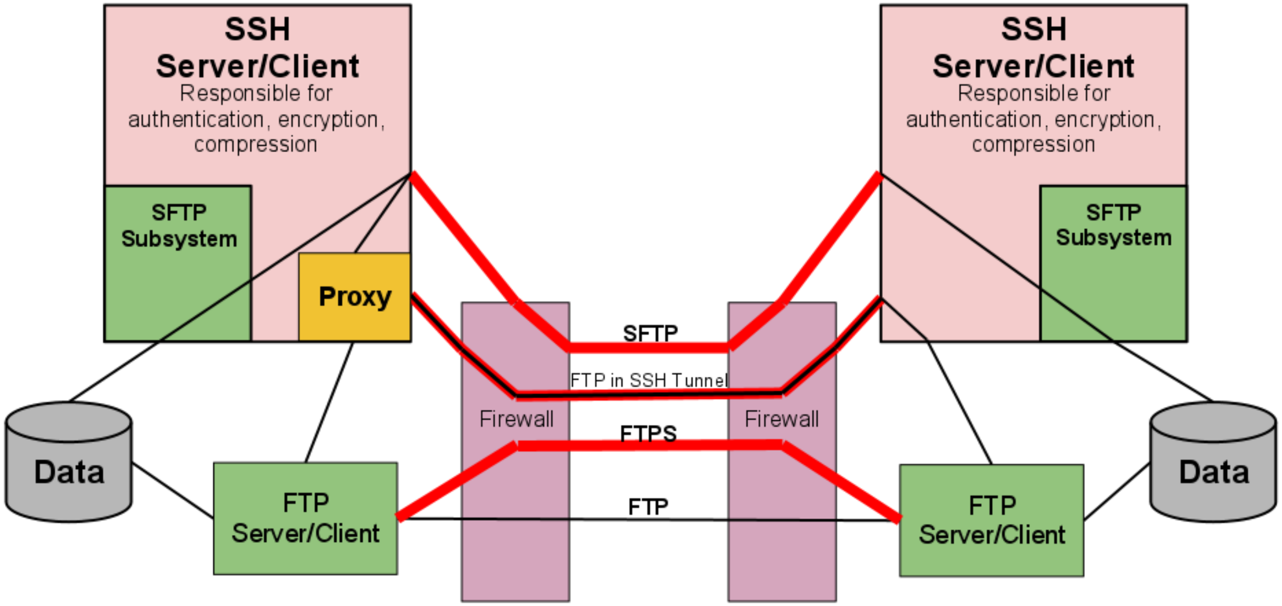
- FTP
- FTPS
- FTP over SSH Tunnel

- SFTP
- FTP to SFTP
- VPN
- PGP



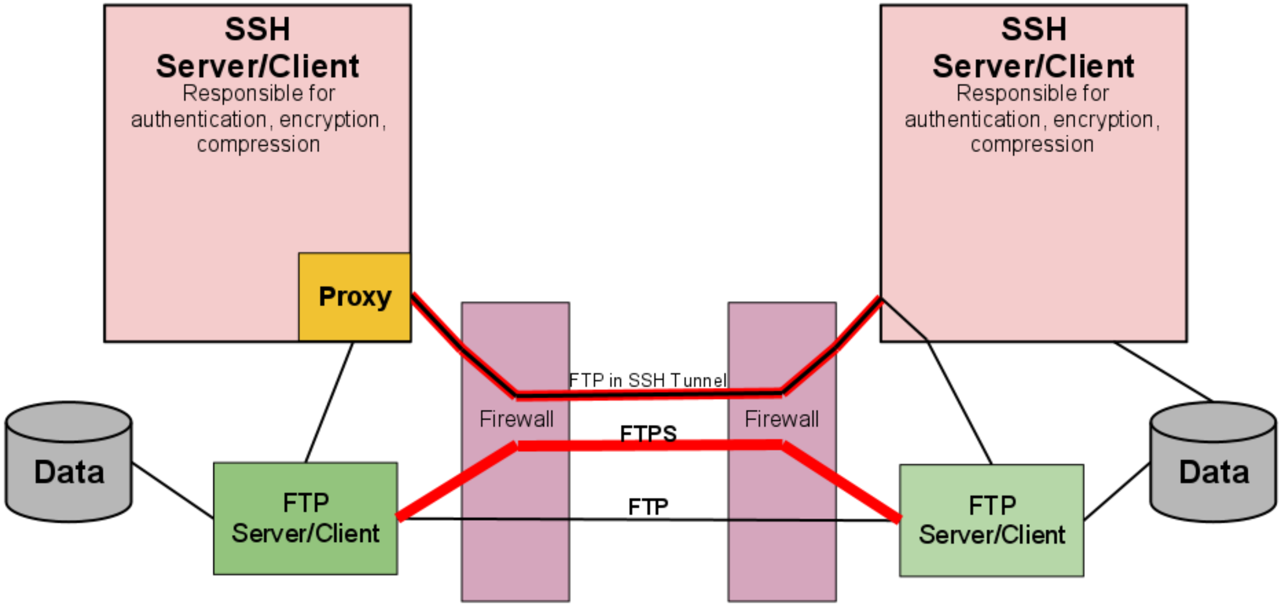
- FTP
- FTPS
- FTP over SSH Tunnel

- SFTP
- FTP to SFTP
- VPN

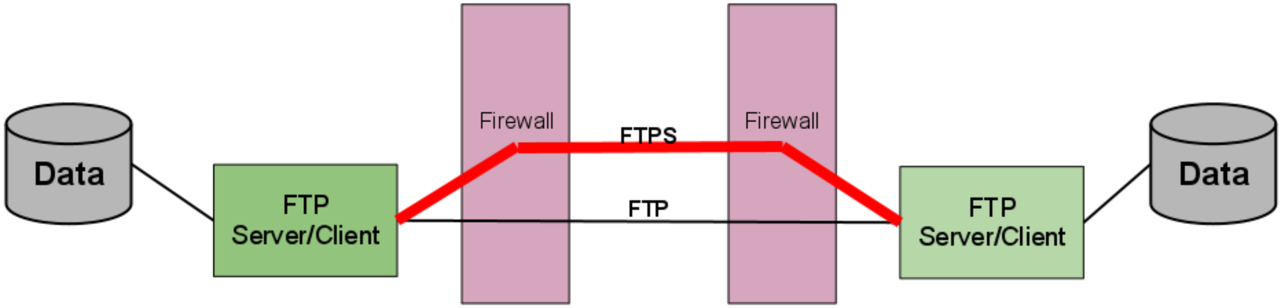


- FTP
- FTPS
- FTP over SSH Tunnel

- SFTP
- FTP to SFTP

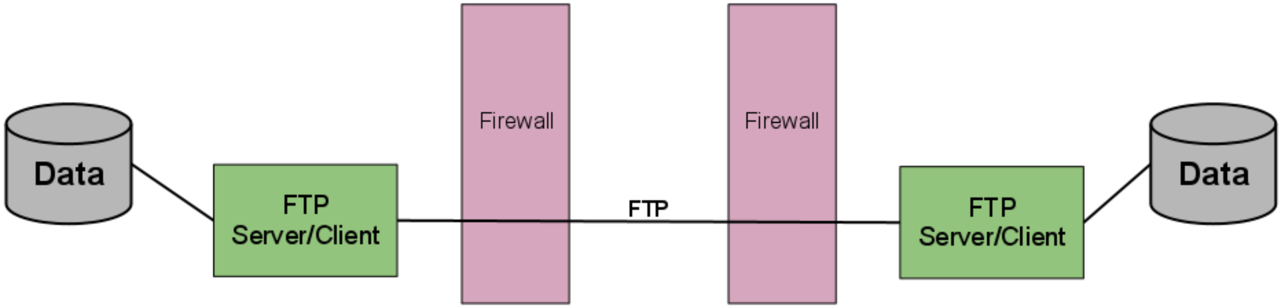


- FTP
- FTPS
- FTP over SSH Tunnel


Data
FTP
Server/Client
Firewall
FTPS
FTP
Firewall
FTP
Server/Client
Data

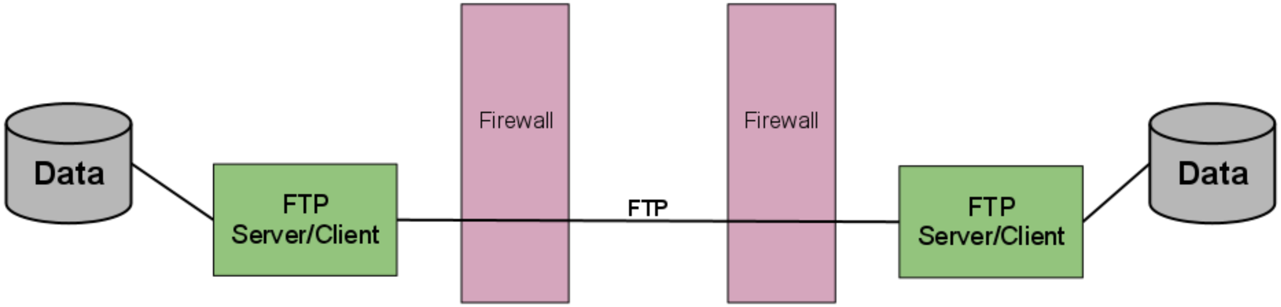
- FTP
- FTPS

FTP



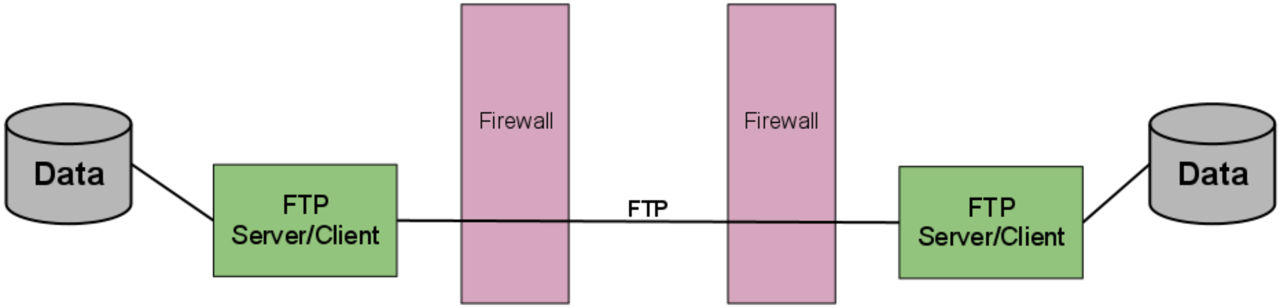
- Pros
    - Ubiquitous

FTP



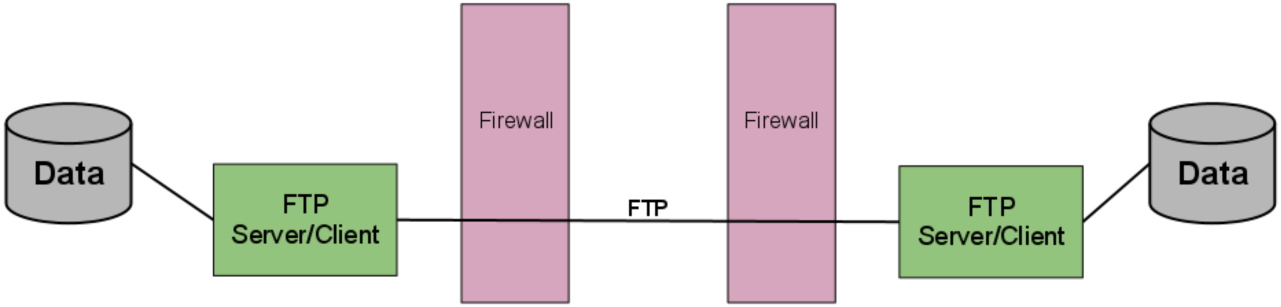
- Pros
  - Ubiquitous
  - Common knowledge

FTP



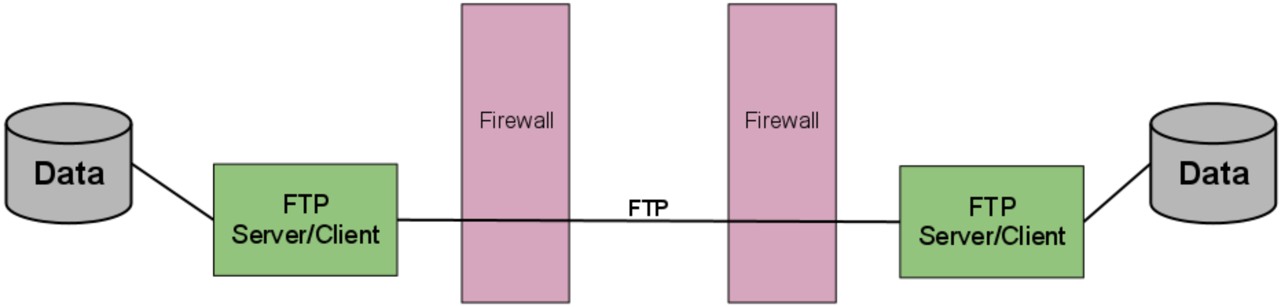
- Pros
  - Ubiquitous
  - Common knowledge
  - Included in base OS

FTP



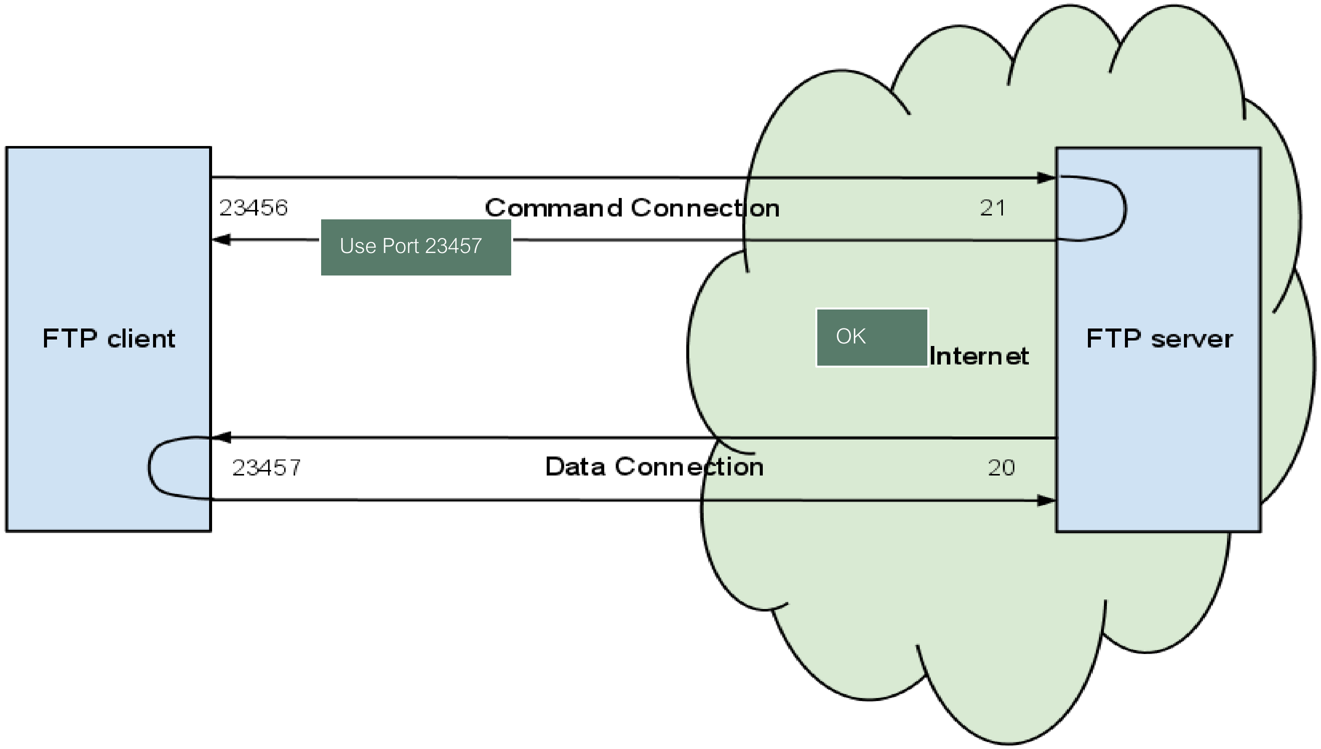
- Pros
  - Ubiquitous
  - Common knowledge
  - Included in base OS
- Cons
  - Very little security

FTP



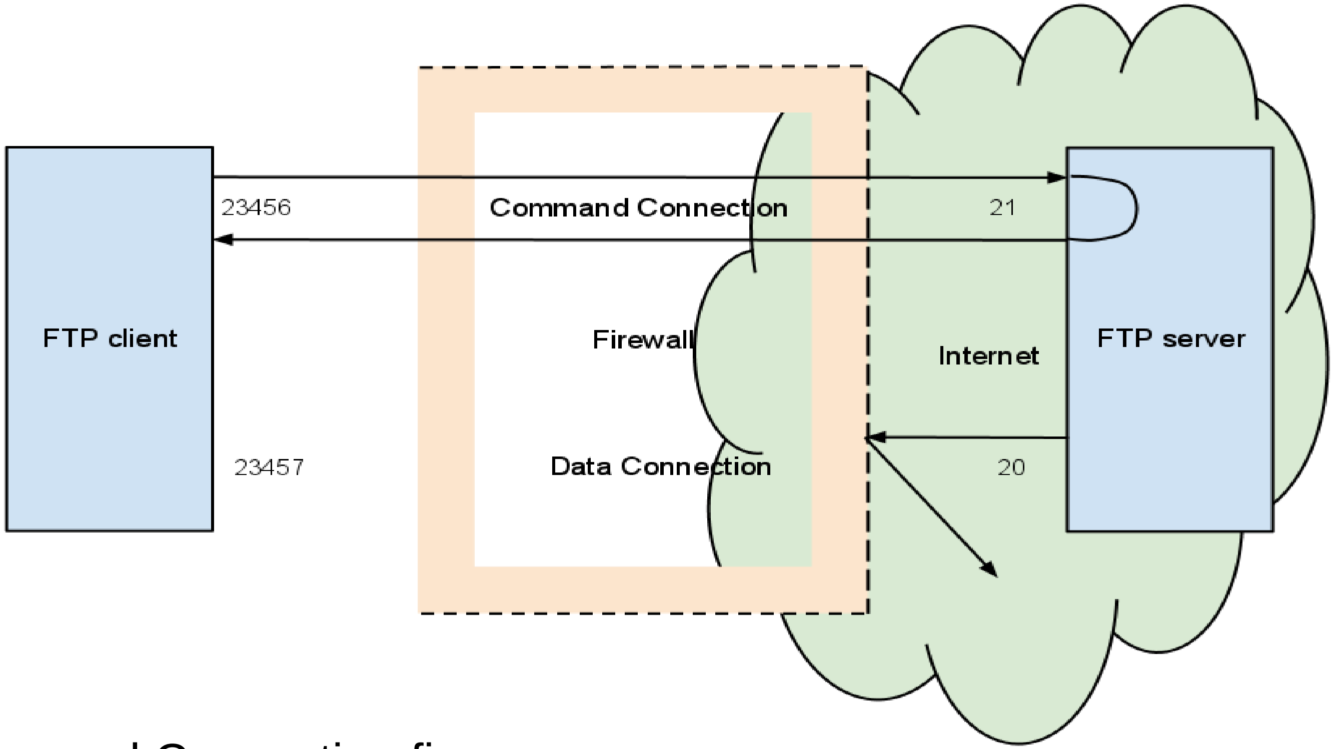
- Pros
  - Ubiquitous
  - Common knowledge
  - Included in base OS
- Cons
  - Very little security
  - Not firewall friendly

Active Firewall



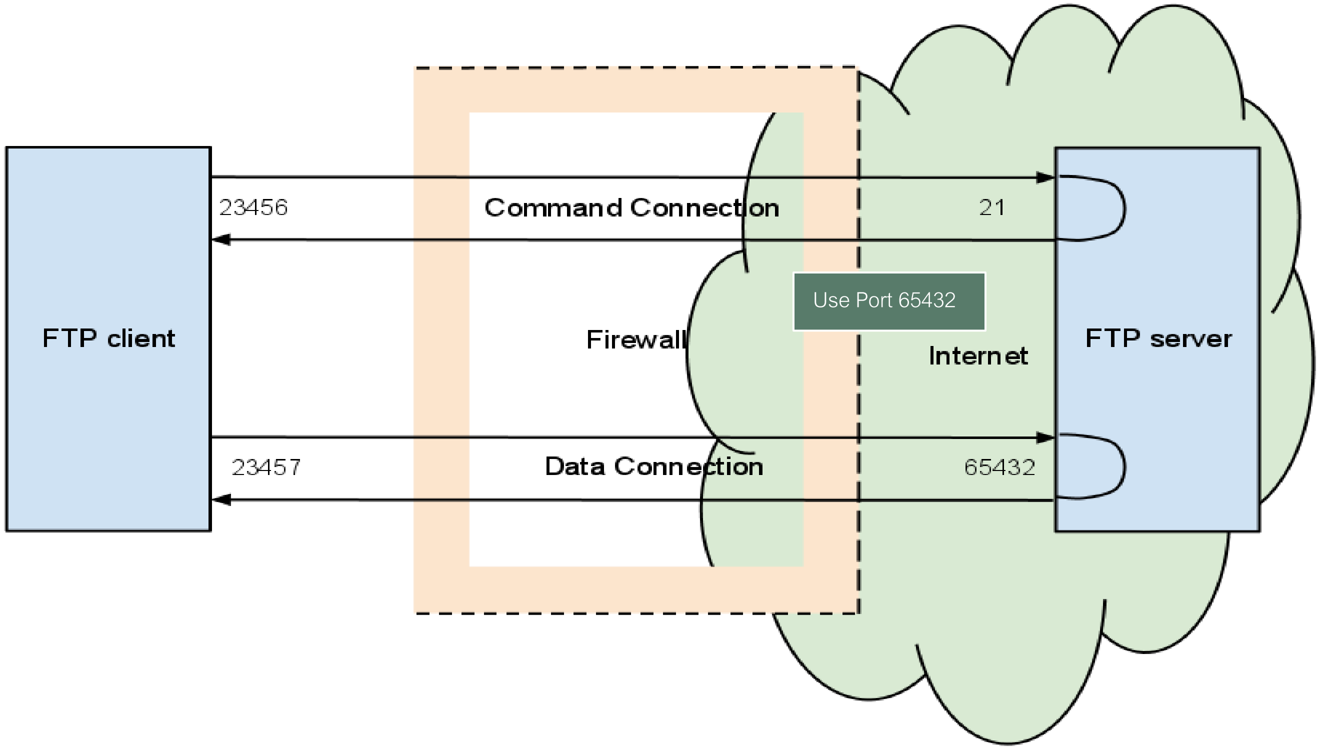
- Client requests a Connection to Server
- Server opens a Data Connection to transfer data to partners
- FTP protocol was designed around a trusting relationship

Active FTP with Firewall



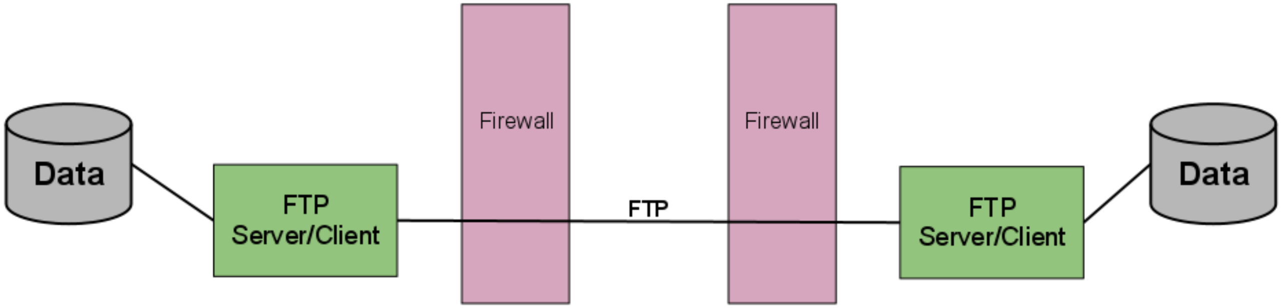
- Command Connection fine
- Firewall Blocks the new connection coming in from outside
- Passive Mode introduced

Passive FTP



- FTP Server configured to support Passive mode
- Server starts a listener on an ephemeral port
- Passes port number to client on Command Connection
- Ephemeral port numbers always not acceptable
- Sniff the Port number on Command Connection

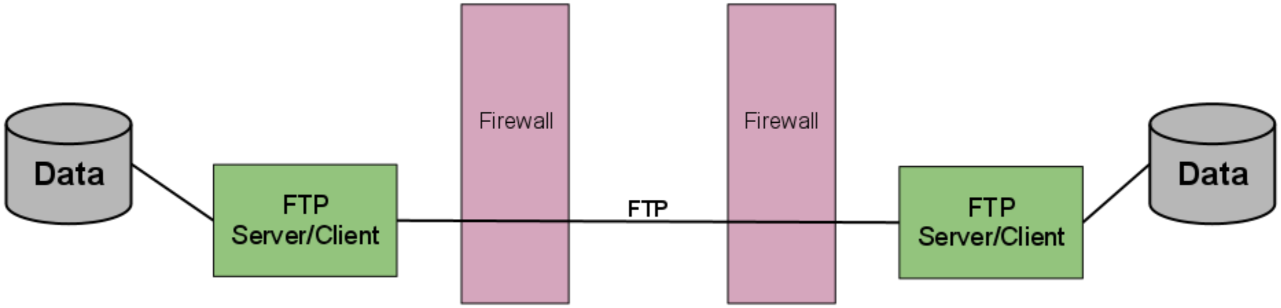
FTP



- Pros
  - Ubiquitous
  - Common knowledge
  - Included in base OS

- Cons
  - Very little security
  - Not firewall friendly
  - No native compression

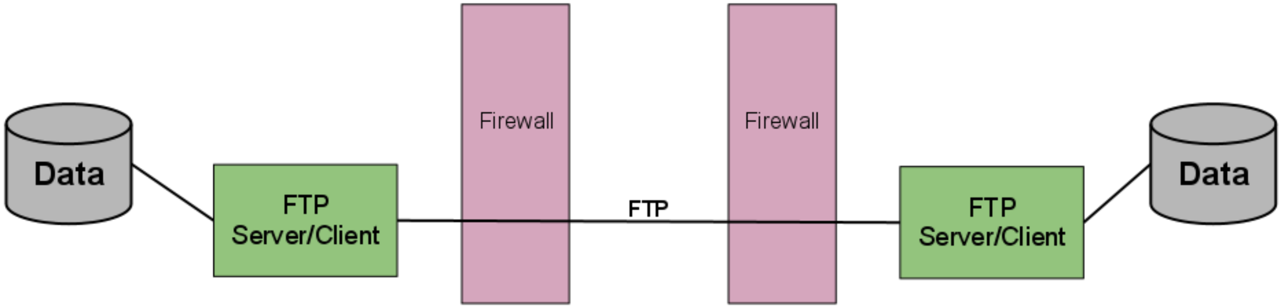
FTP



- Pros
  - Ubiquitous
  - Common knowledge
  - Included in base OS

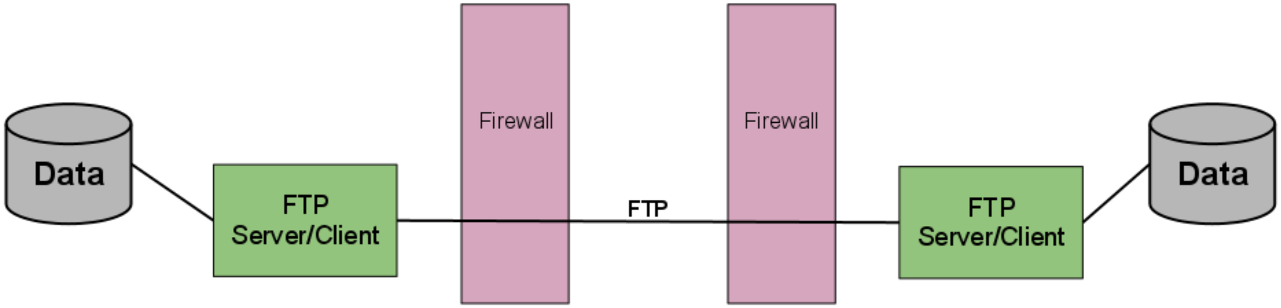
- Cons
  - Very little security
  - Not firewall friendly
  - No native compression
  - Lacks integrity validation

FTP



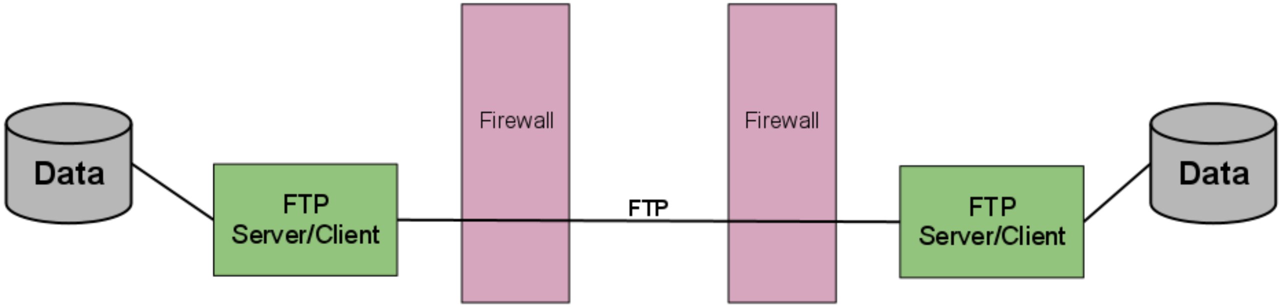
- Common uses
  - Public information

FTP



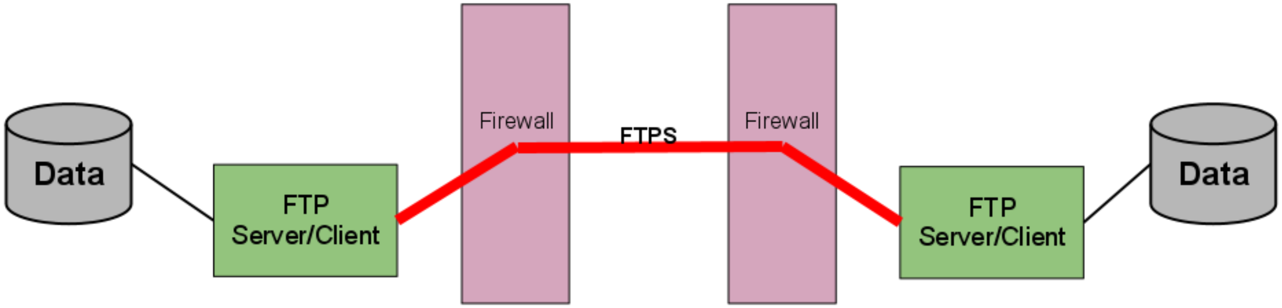
- Common uses
  - Public information
  - Intranet transfers (careful, not everyone on the intranet is safe)

FTP



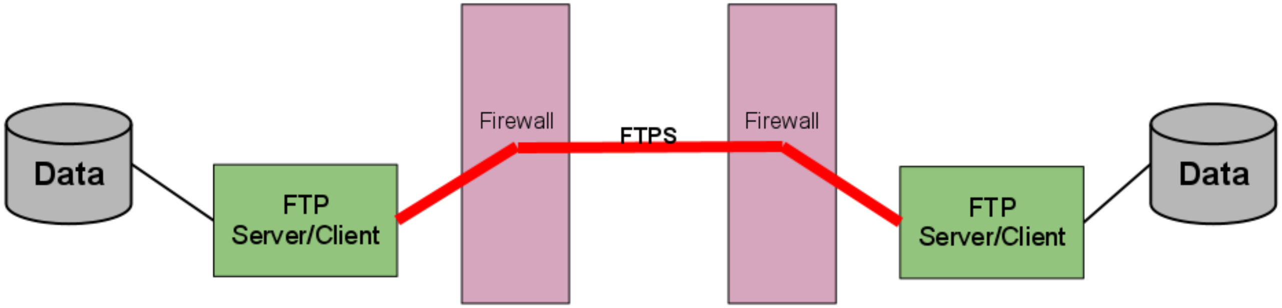
- Common uses
  - Public information
  - Intranet transfers (careful, not everyone on the intranet is safe)
  - Far too many things that should really use something better

FTP over SSL (FTPS)



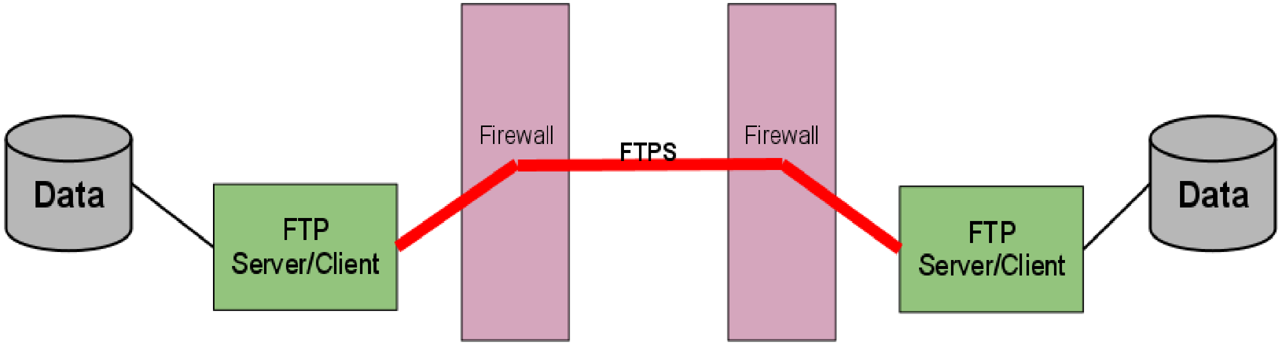
- Pros
  - Same FTP familiarity

FTP over SSL (FTPS)



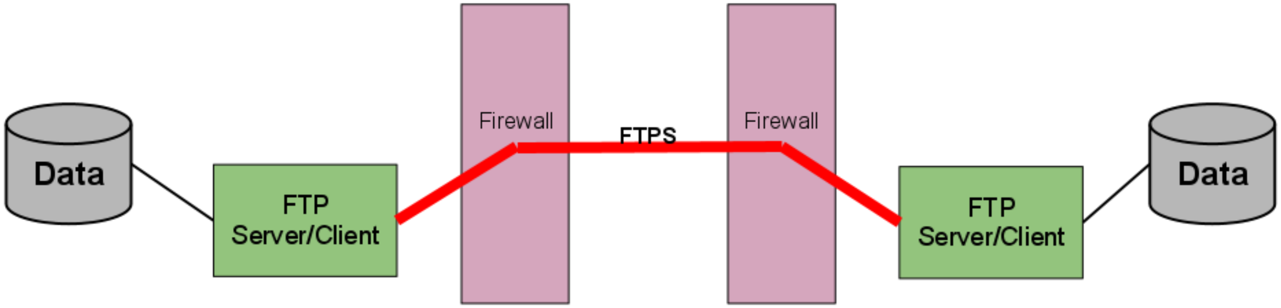
- Pros
  - Same FTP familiarity
  - Included in base z/OS

FTP over SSL (FTPS)



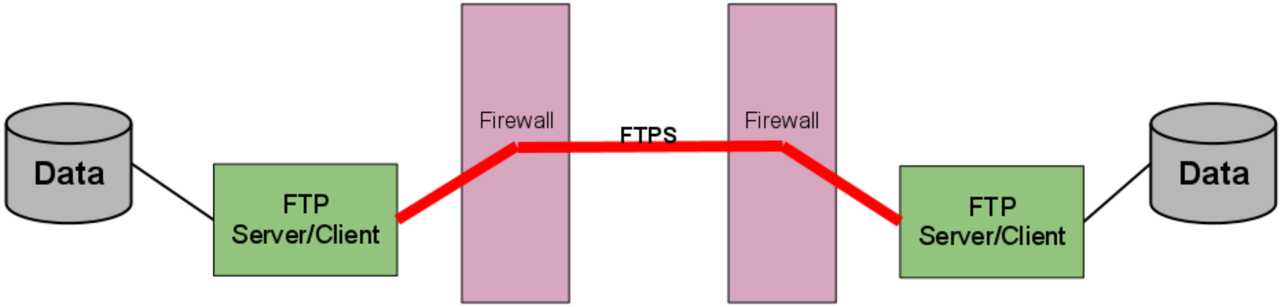
- Pros
  - Same FTP familiarity
  - Included in base z/OS
  - Supports X.509 certificates (trusted authority) and keberos

FTP over SSL (FTPS)



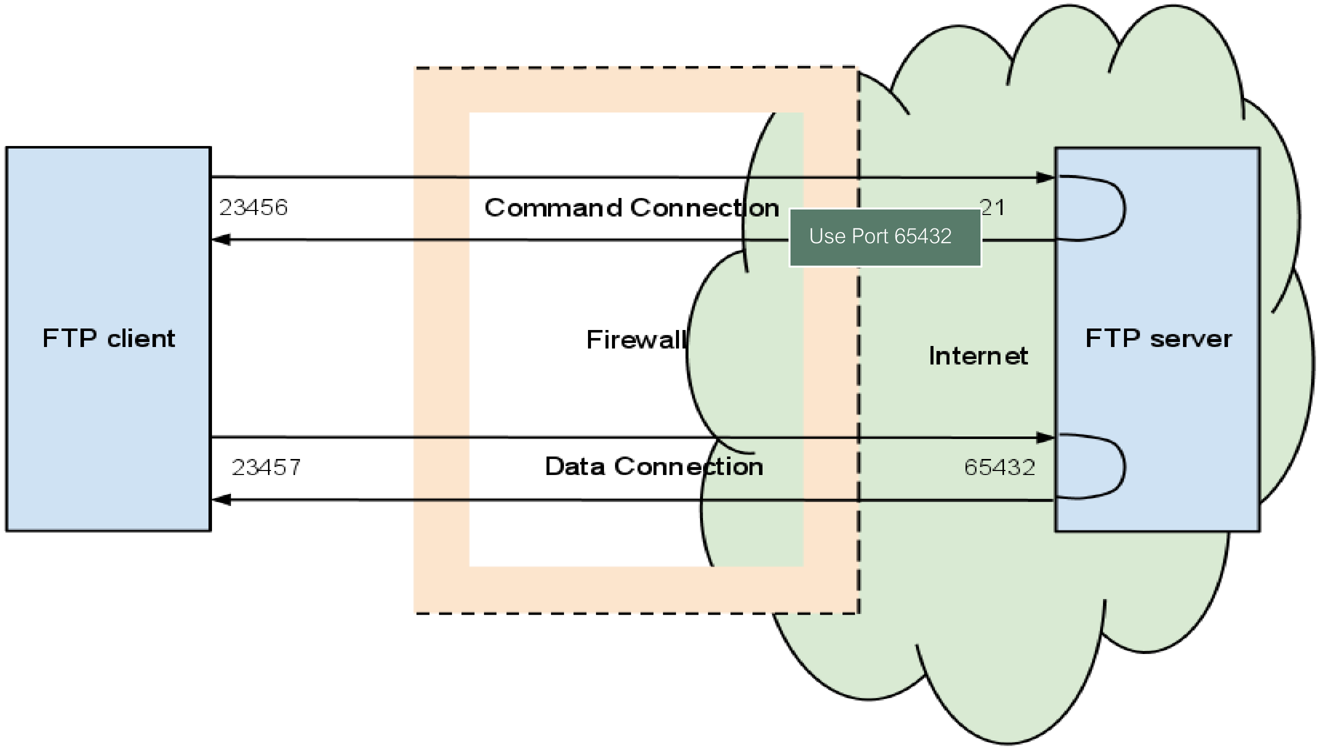
- Pros
  - Same FTP familiarity
  - Included in base z/OS
  - Supports X.509 certificates (trusted authority) and keberos
  - RACF keyrings supported

FTP over SSL (FTPS)

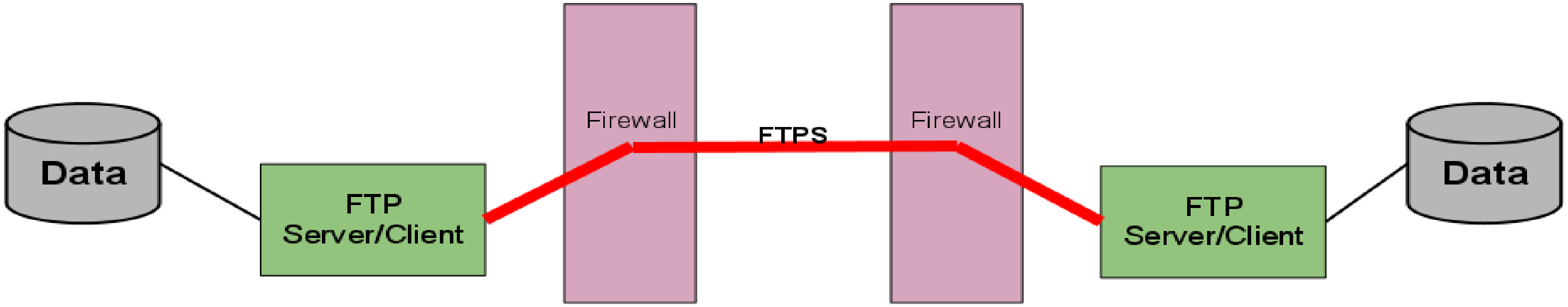


- Pros
  - Same FTP familiarity
  - Included in base z/OS
  - Supports X.509 certificates (trusted authority) and keberos
  - RACF keyrings supported
- Cons
  - Not firewall friendly (even worse than straight FTP)

Passive FTP


FTP client
23456
Command Connection
21
Use Port 65432
Firewall
Internet
FTP server
23457
Data Connection
65432

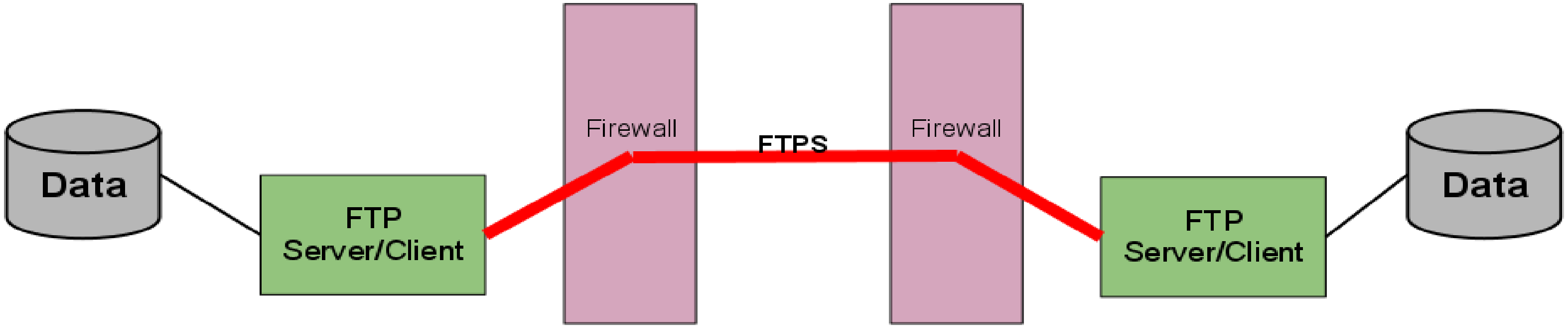
FTP over SSL (FTPS)



- Pros
  - Same FTP familiarity
  - Included in base z/OS
  - Supports X.509 certificates (trusted authority) and keberos
  - RACF keyrings supported

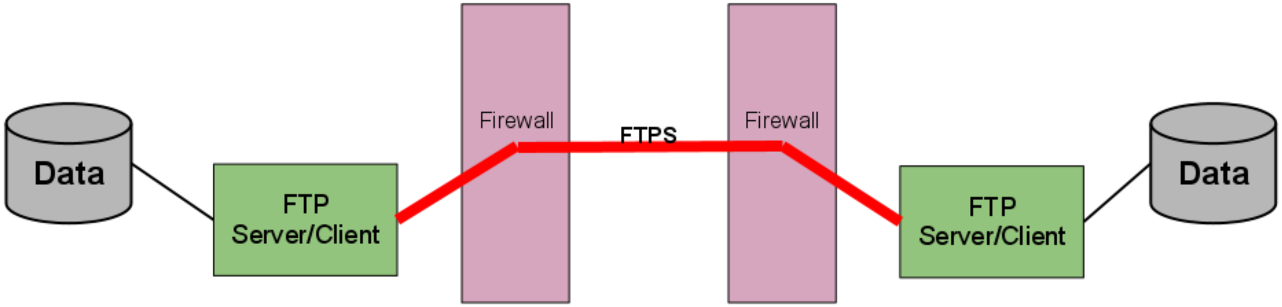
- Cons
  - Not firewall friendly (even worse than straight FTP)
  - Can’t assume it’s available on the other end

FTP over SSL (FTPS)



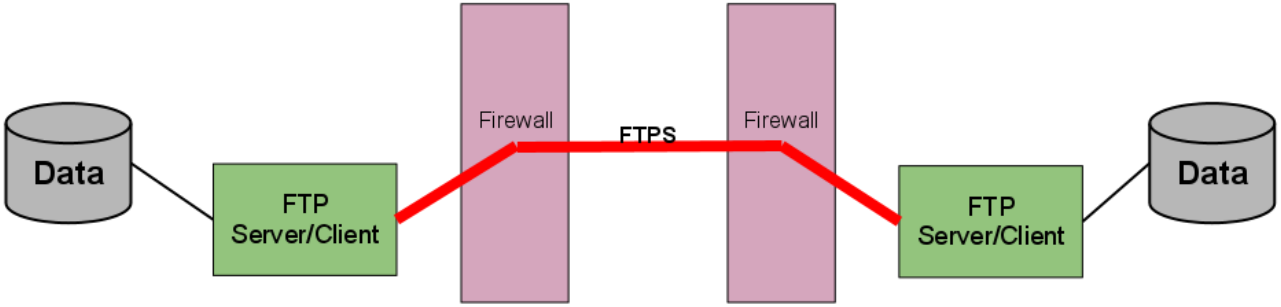
- Common Uses
    - z/OS to z/OS

FTP over SSL (FTPS)



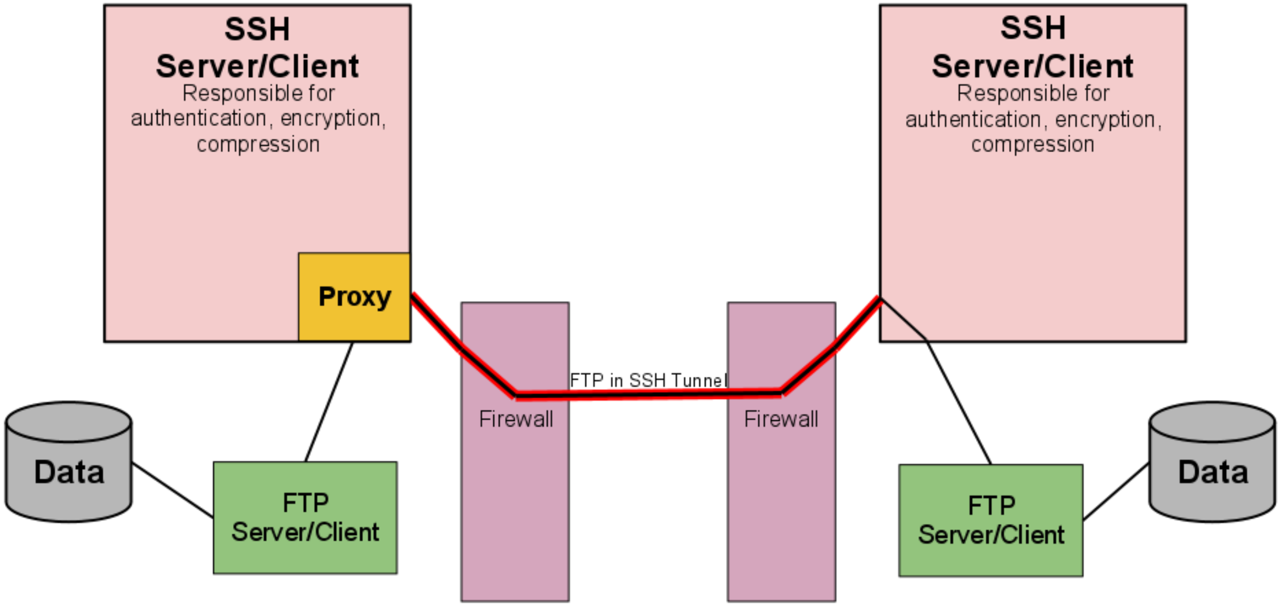
- Common Uses
  - z/OS to z/OS
  - z/OS to i/Series

FTP over SSL (FTPS)



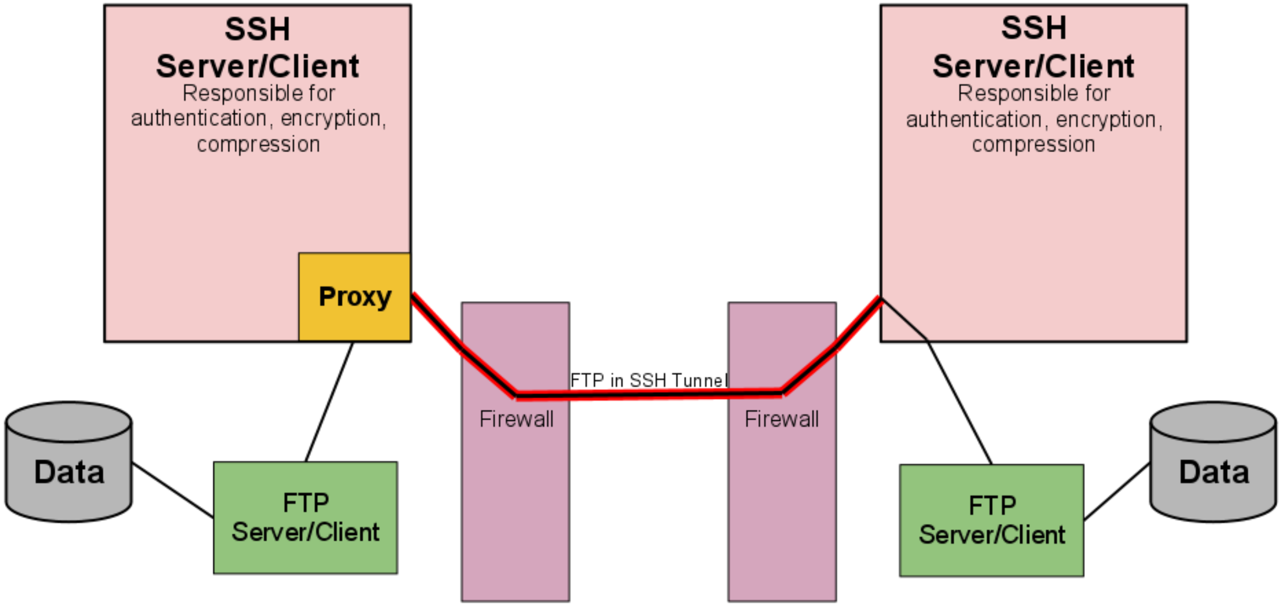
- Common Uses
  - z/OS to z/OS
  - z/OS to i/Series
  - Servers and clients available on platforms

FTP over SSH Tunnel


SSH
Server/Client
Responsible for authentication, encryption, compression
Proxy
Data
FTP
Server/Client
Firewall
FTP in SSH Tunnel
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
FTP
Server/Client
Data

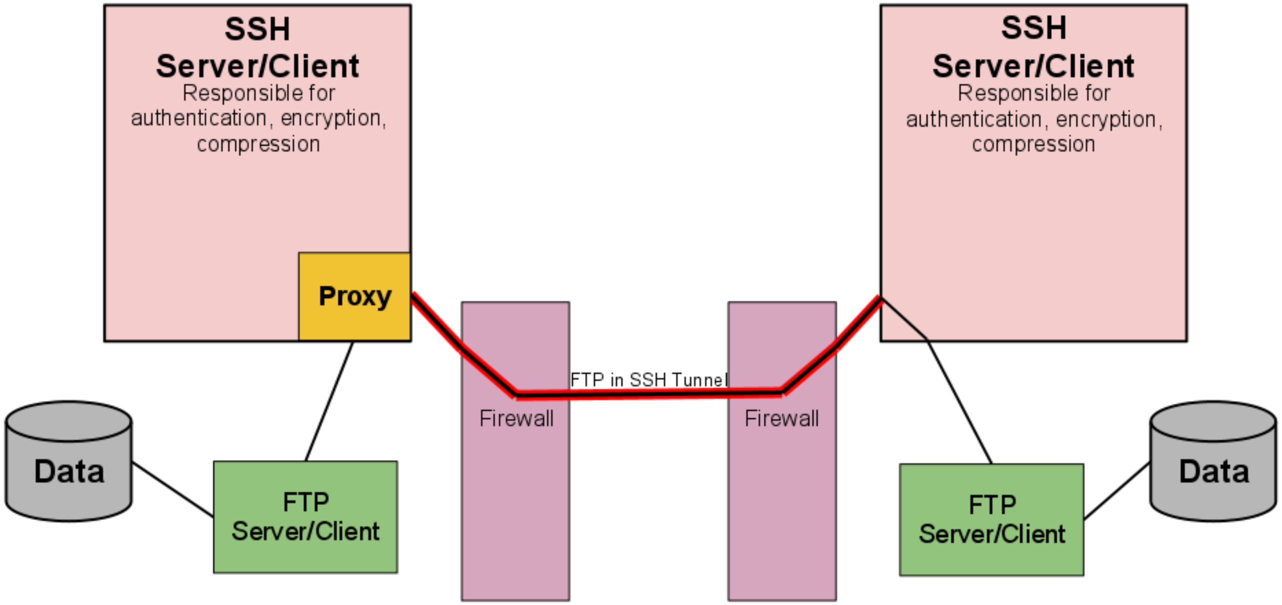
- Pros
  - Same FTP familiarity

FTP over SSH Tunnel



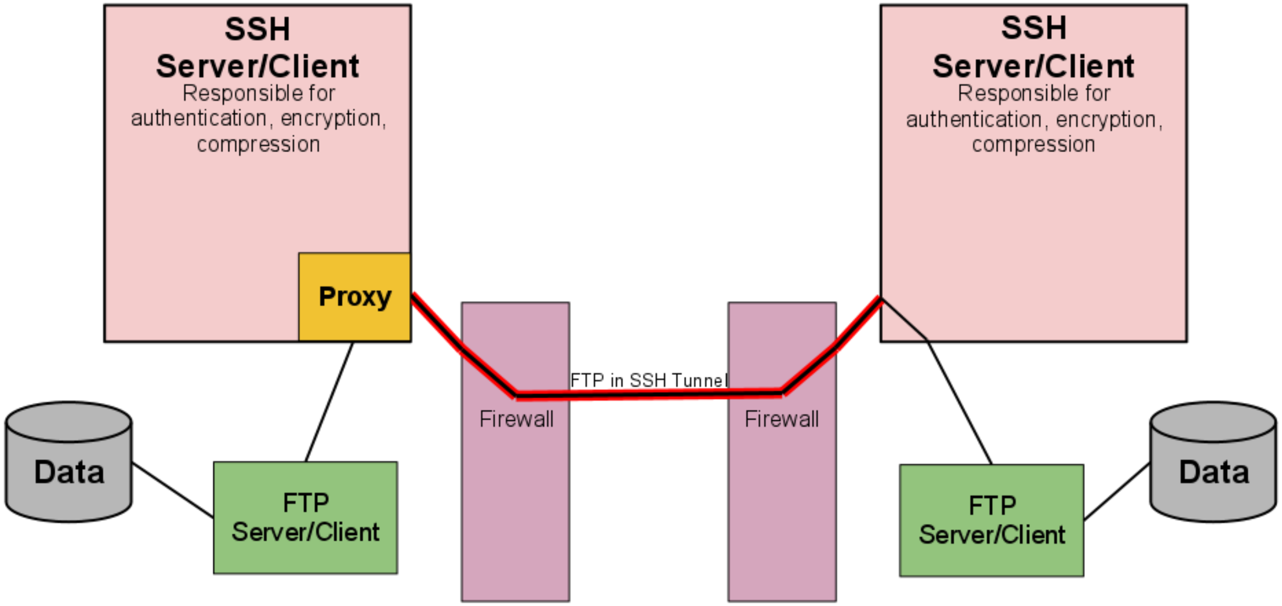
- Pros
  - Same FTP familiarity
  - Firewall friendly – Port 22

FTP over SSH Tunnel


SSH
Server/Client
Responsible for authentication, encryption, compression
Proxy
FTP in SSH Tunnel
Firewall
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
Data
FTP
Server/Client
FTP
Server/Client
Data

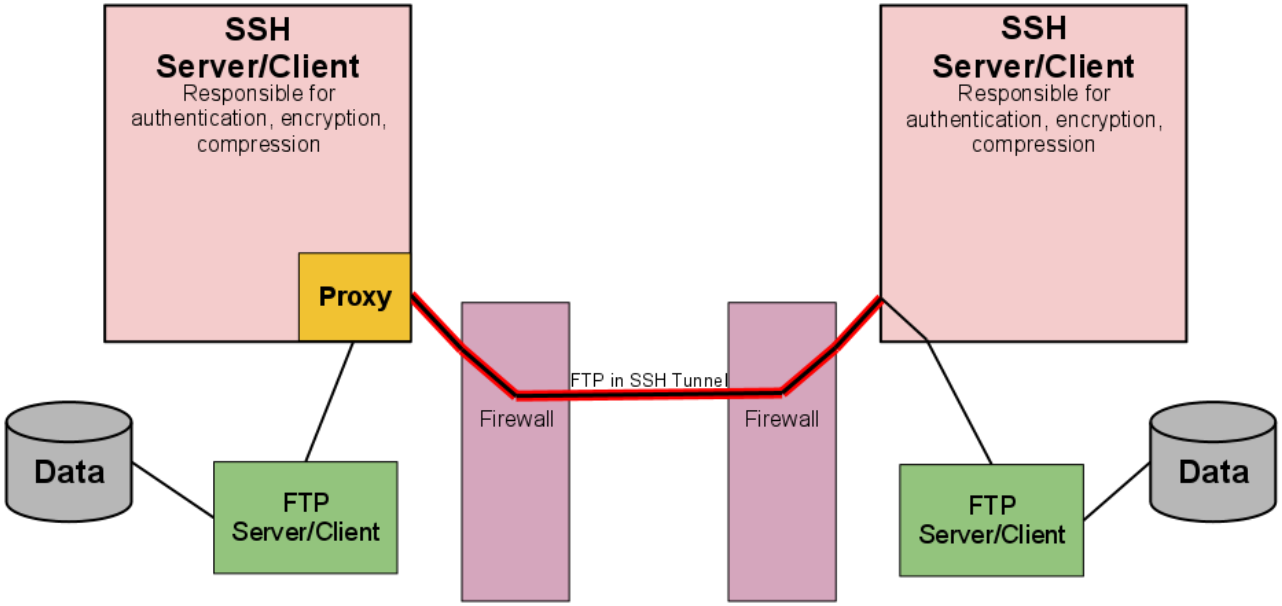
- Pros
  - Same FTP familiarity
  - Firewall friendly
  - Compression of data – IFL’s

FTP over SSH Tunnel



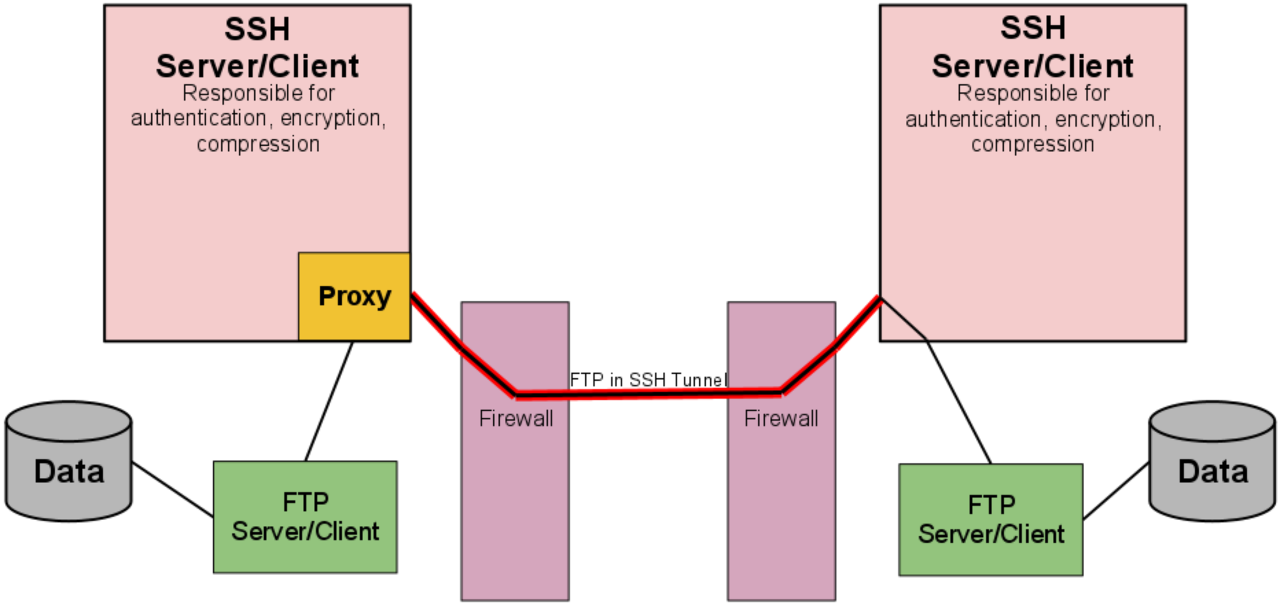
- Pros
  - Same FTP familiarity
  - Firewall friendly
  - Compression of data
  - Good checksums of data, at least for the internet piece

FTP over SSH Tunnel



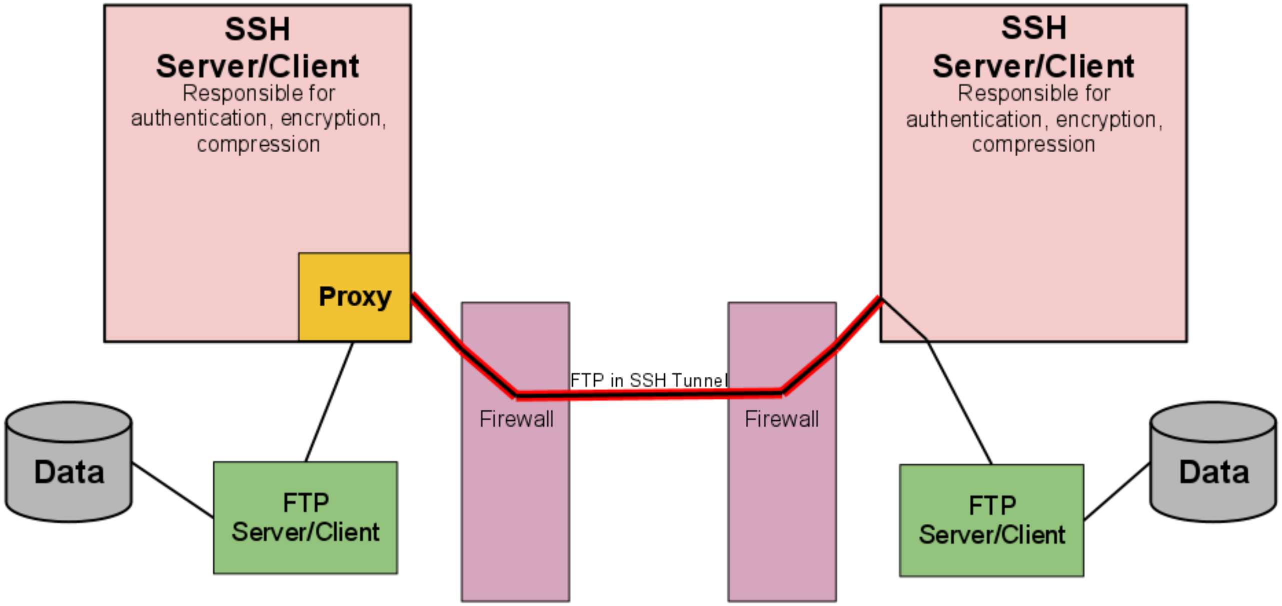
- Pros
  - Same FTP familiarity
  - Firewall friendly
  - Compression of data
  - Good checksums of data, at least for the internet piece
- Cons
  - More parts need to be choreographed

FTP over SSH Tunnel



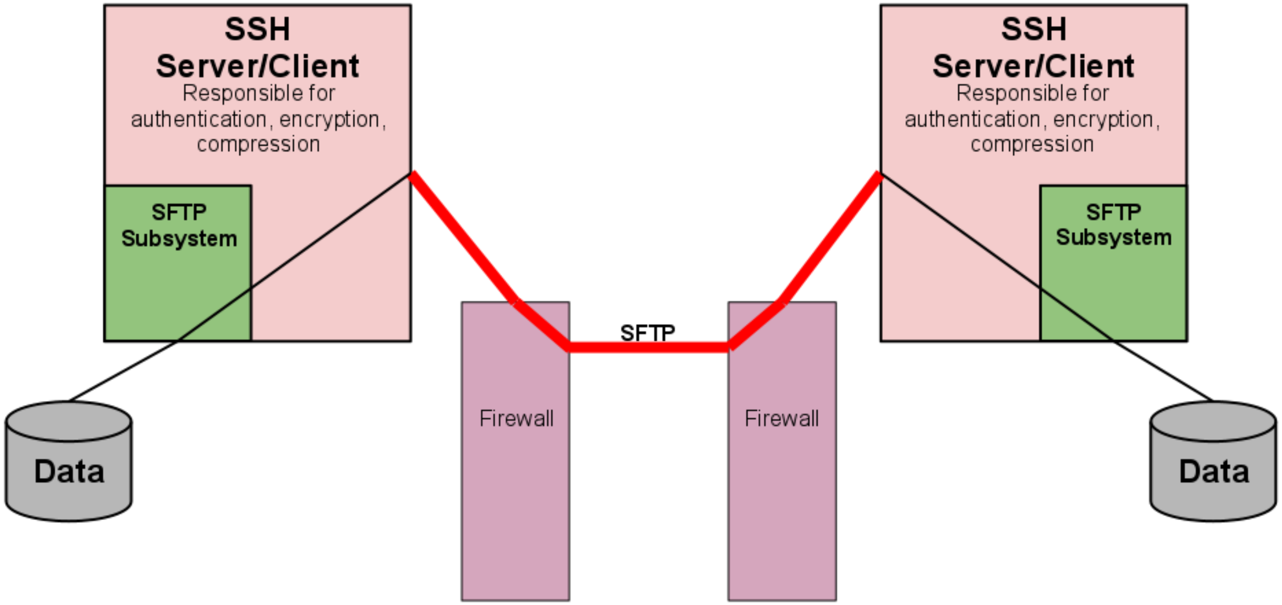
- Pros
  - Same FTP familiarity
  - Firewall friendly
  - Compression of data
  - Good checksums of data, at least for the internet piece
- Cons
  - More parts need to be choreographed
  - Requires SSH and FTP on both ends

FTP over SSH Tunnel



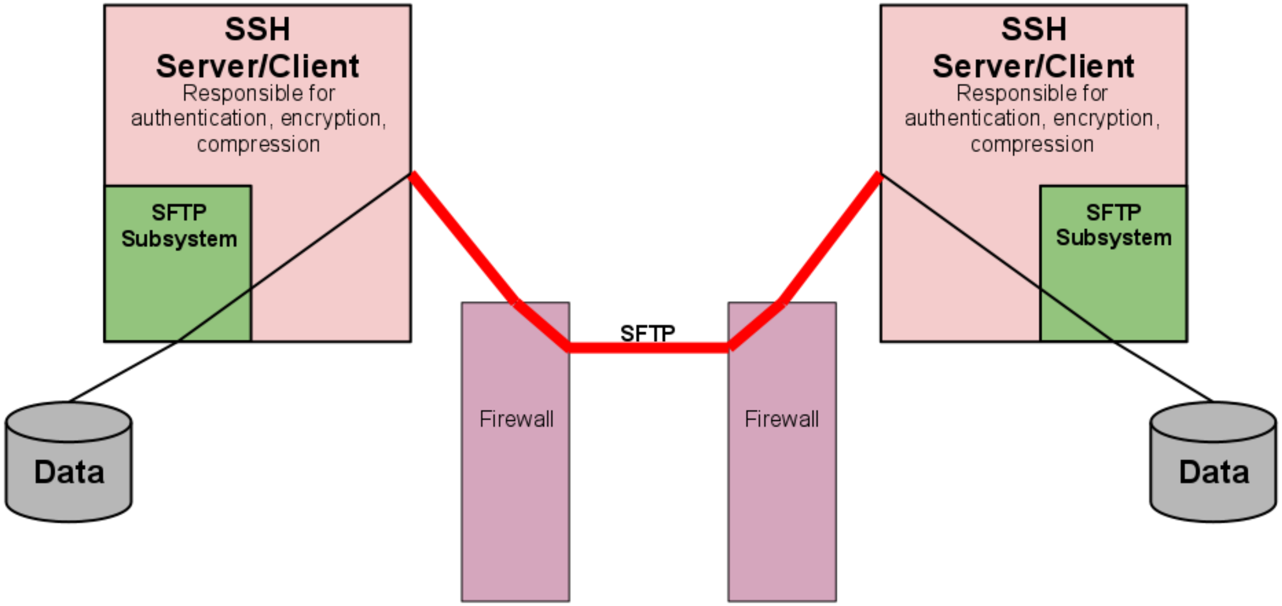
- Common uses
  - Sites that have a significant reliance on FTP already in place that need to implement SSH encryption for transit
  - Little modification to batch jobs

Secure FTP (SFTP)


SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data
Firewall
SFTP
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data

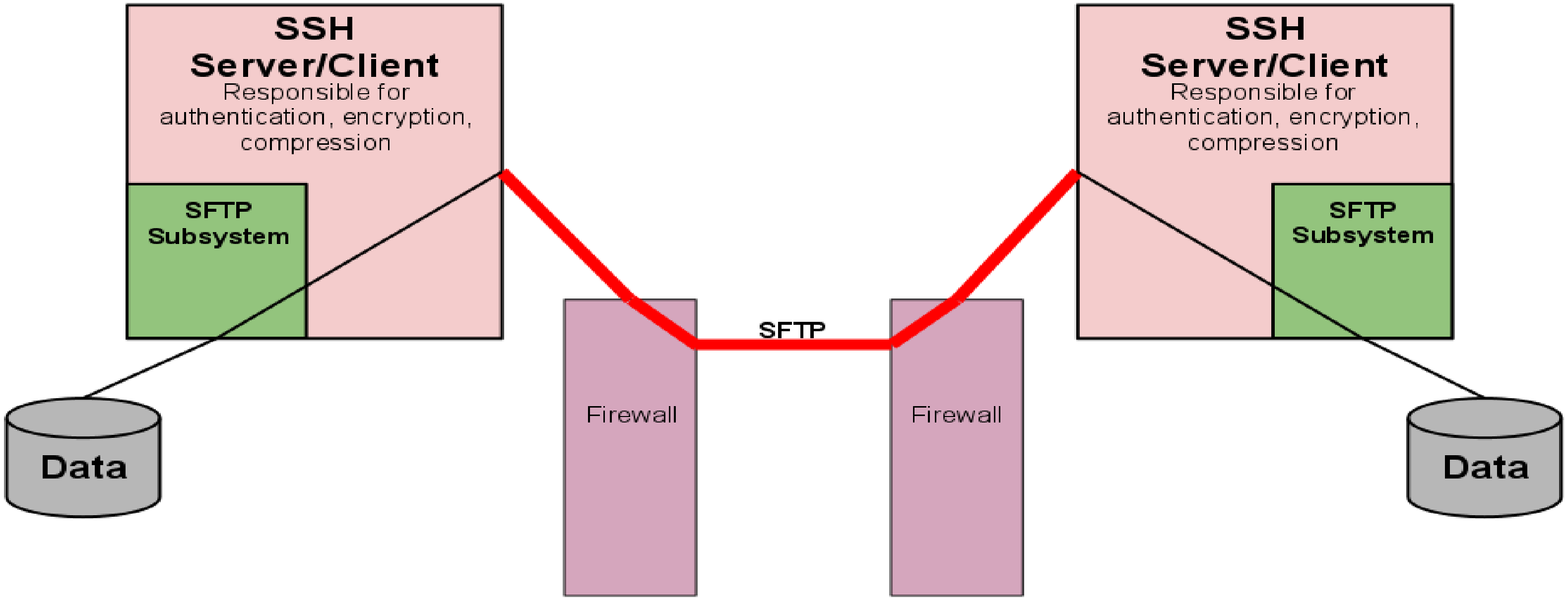
- Pros
  - Point to point encryption
  - SFTP is not encryption engine
  - SSH component responsible for Encryption (TCP traffic)
  - SFTP is the “addon” makes file transfers convenient to run through SSH tunnel

Secure FTP (SFTP)


SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data
Firewall
SFTP
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data

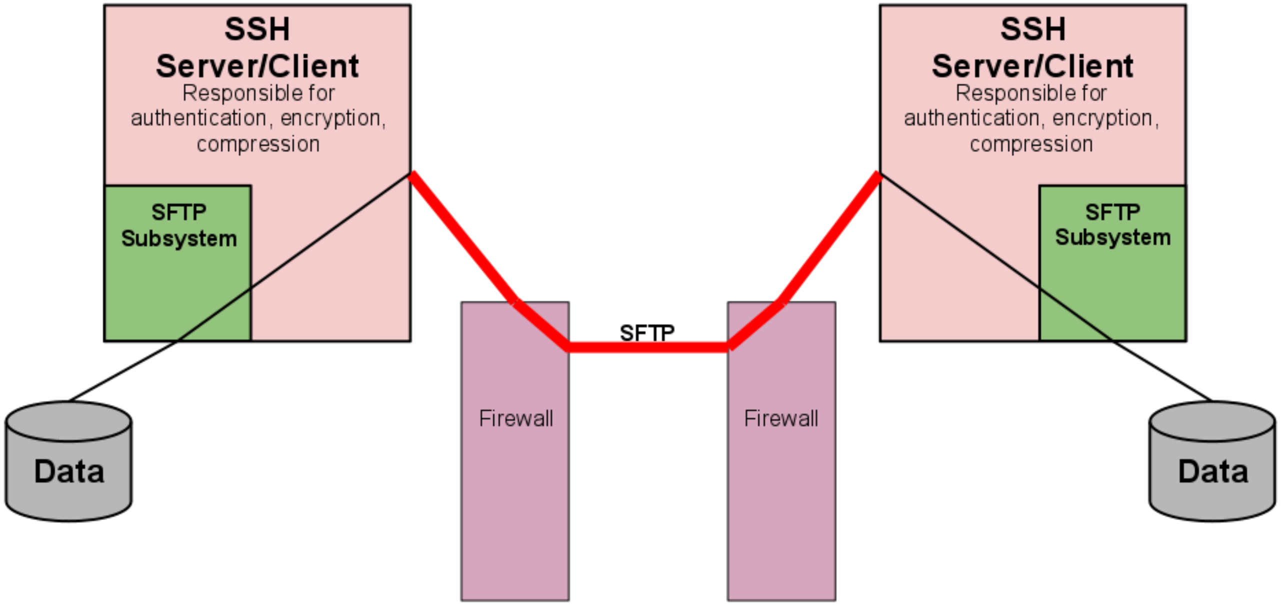
- Pros
  - Point to point encryption
  - Compression and Integrity built-in

Secure FTP (SFTP)


SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data
Firewall
SFTP
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data

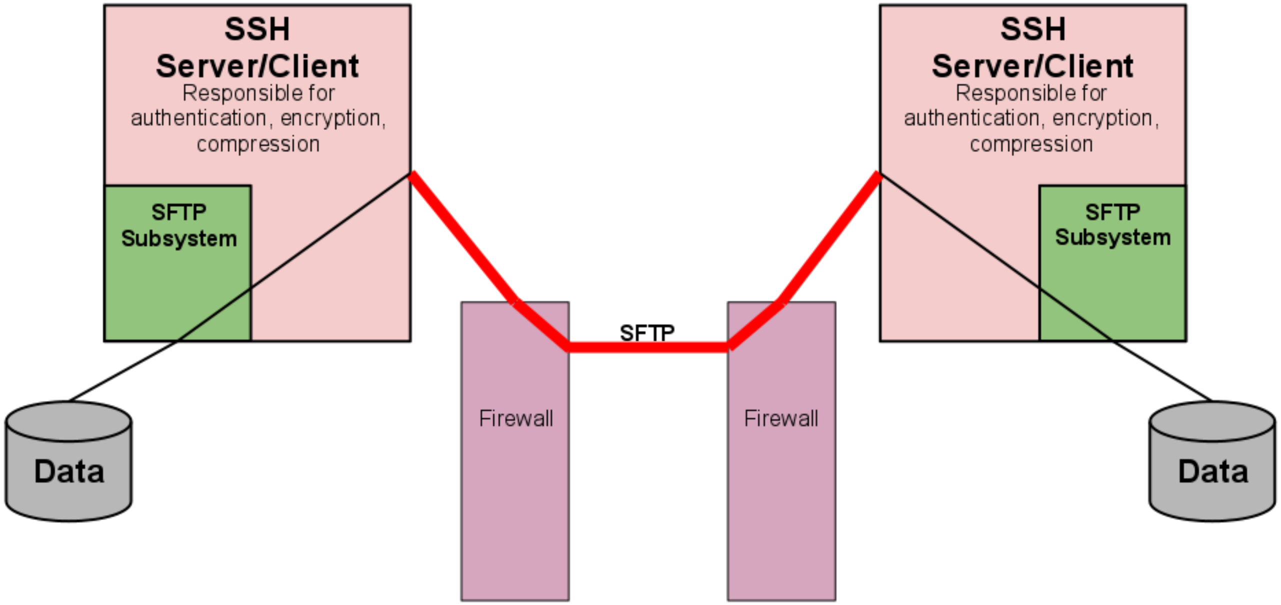
- Pros
  - Point to point encryption
  - Compression and Integrity built-in
  - Already ready to go on Unix/Linux servers - Semi Ubiquitous

Secure FTP (SFTP)


SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data
Firewall
SFTP
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data

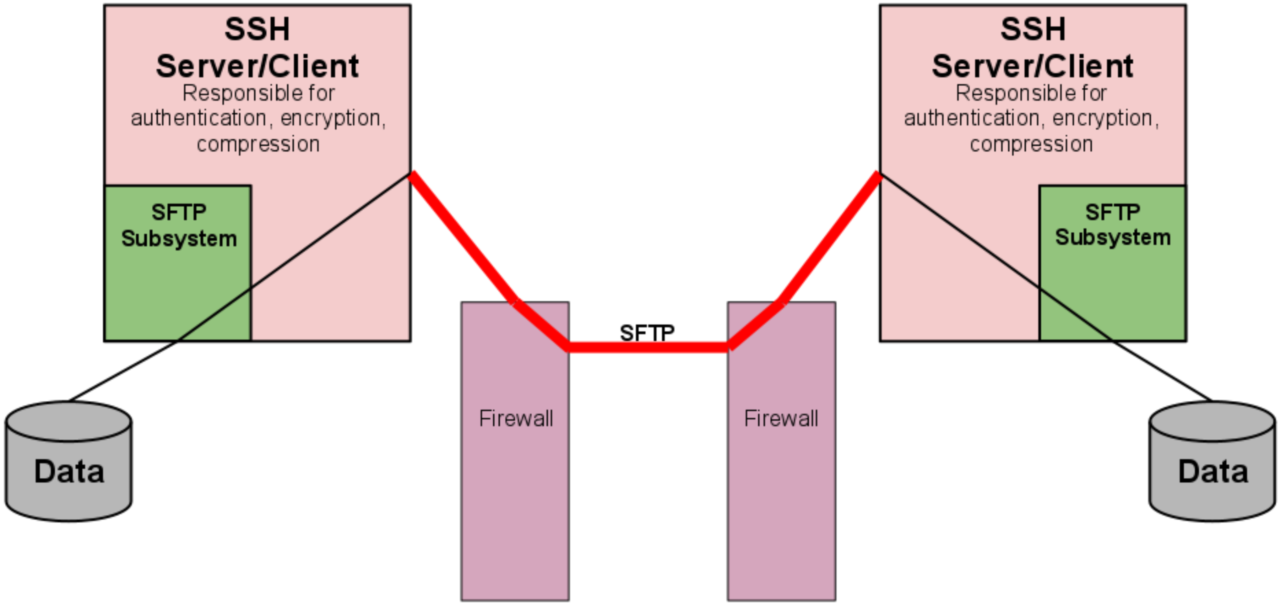
- Pros
  - Point to point encryption
  - Compression and Integrity built-in
  - Already ready to go on Unix/Linux servers
- Cons
  - Not part of base on z/OS or Windows – Base SFTP has some limitations (EBCDIC to ASCII translation – MVS Datasets- JES Queue

Secure FTP (SFTP)


SSH Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data
Firewall
SFTP
Firewall
SSH Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data

- Pros
  - Point to point encryption
  - Compression and Integrity built-in
  - Already ready to go on Unix/Linux servers
- Cons
  - Not part of base on z/OS or windows
  - May not be as familiar to users

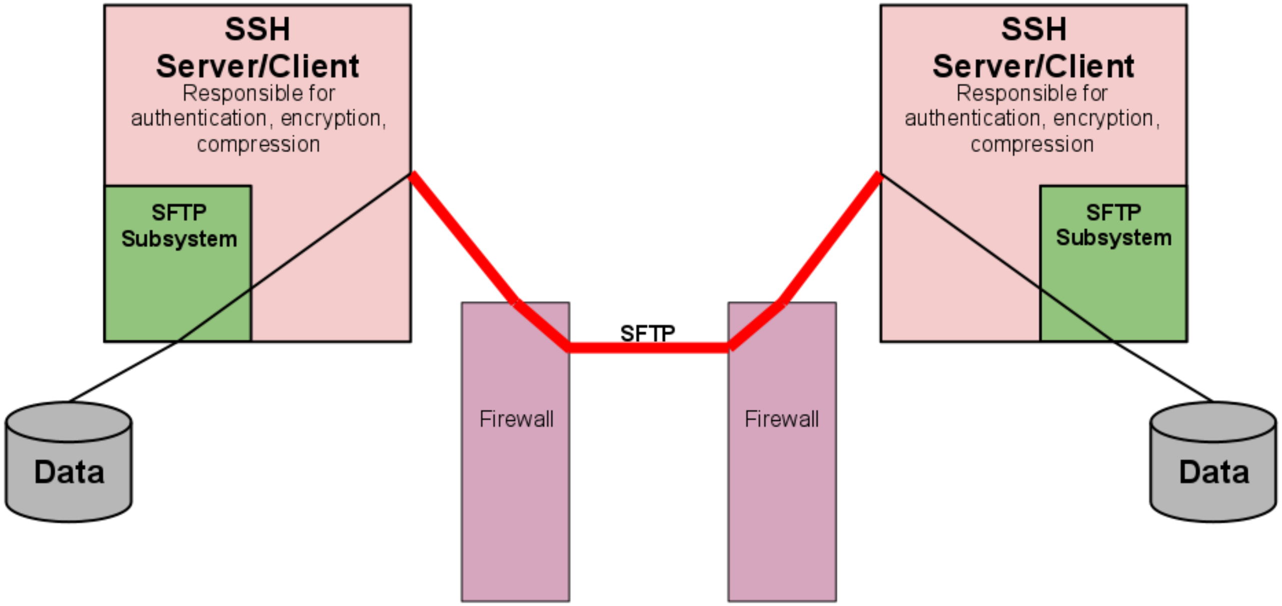
Secure FTP (SFTP)


SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Firewall
SFTP
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data
Data

- Pros
  - Point to point encryption
  - Compression and Integrity built-in
  - Already ready to go on Unix/Linux servers

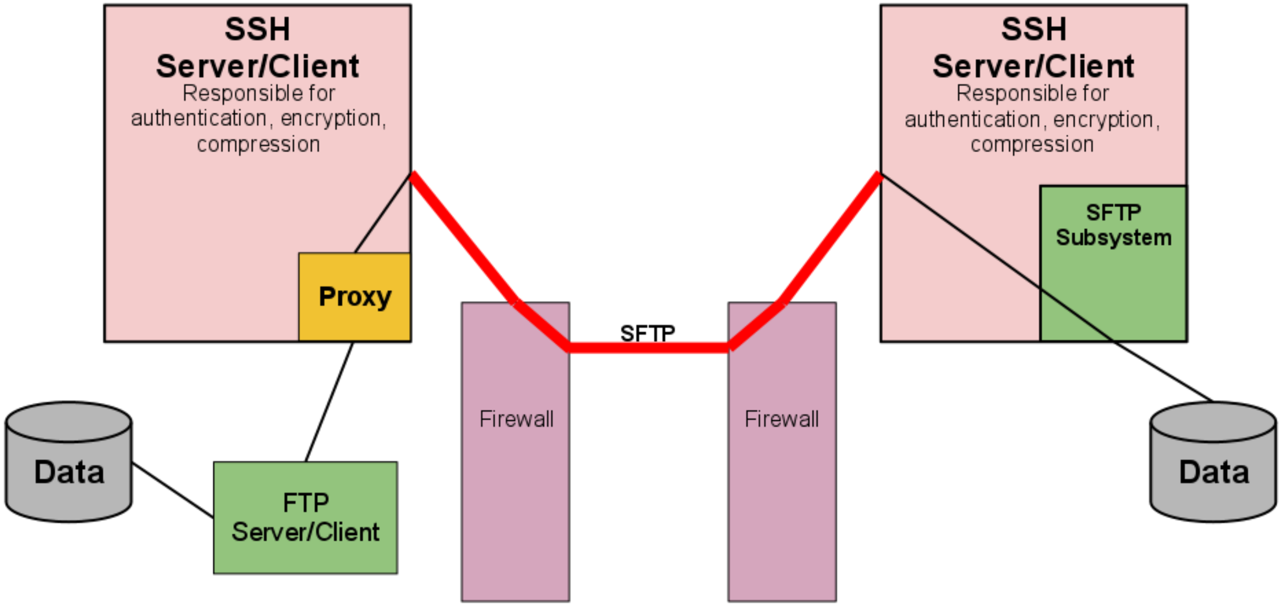
- Cons
  - Not part of base on z/OS or windows
  - May not be as familiar to users
  - Only protects data in transit

Secure FTP (SFTP)


SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data
Firewall
SFTP
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP Subsystem
Data

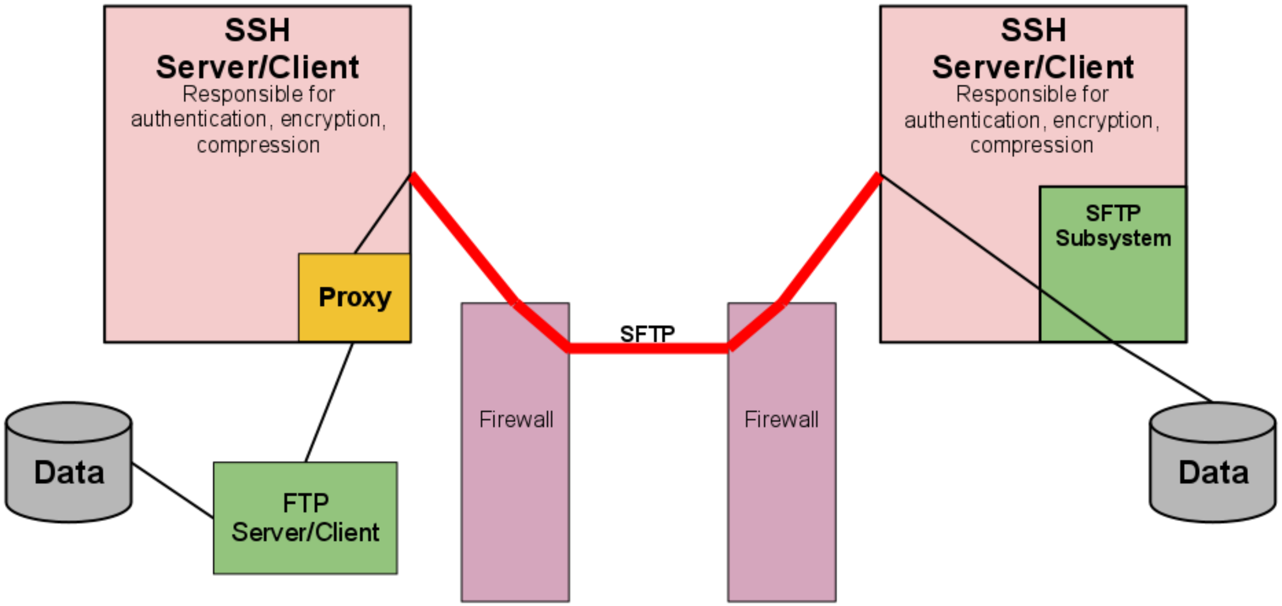
- Common uses
  - Easy access for distribution to Unix/Linux farms

FTP to SFTP Conversion (Vendor Solution)



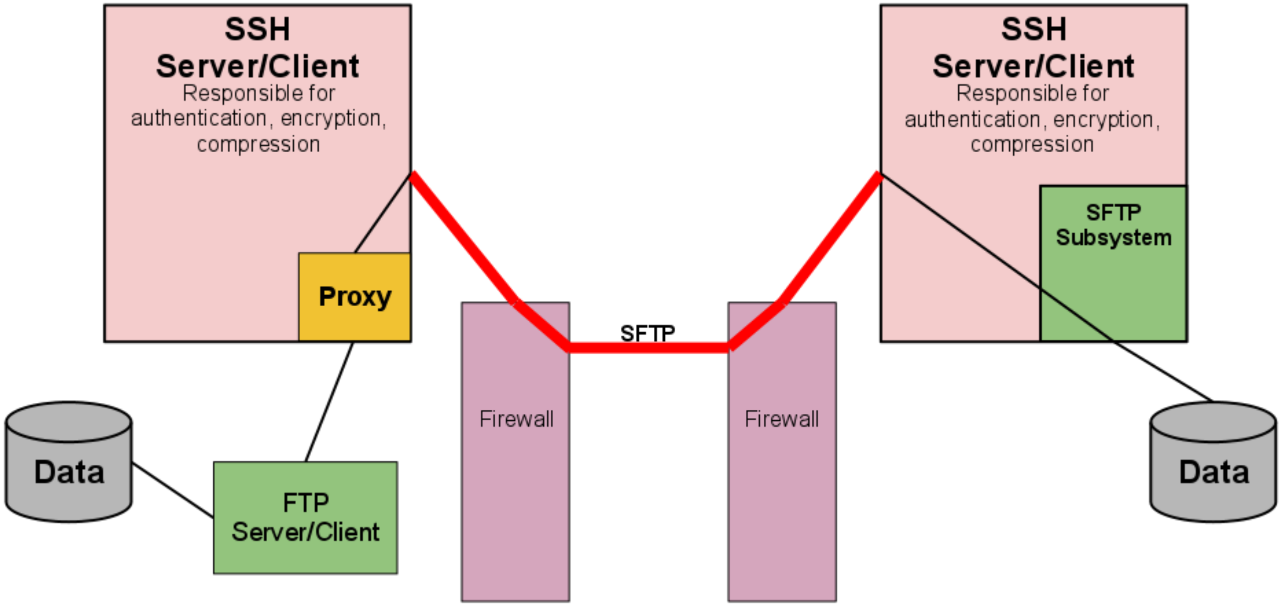
- Pros
  - Satisfies SFTP requirement – Commands routed to Proxy and converted)

FTP to SFTP Conversion (Vendor Solution)


SSH
Server/Client
Responsible for authentication, encryption, compression
Proxy
Data
FTP
Server/Client
Firewall
SFTP
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP
Subsystem
Data

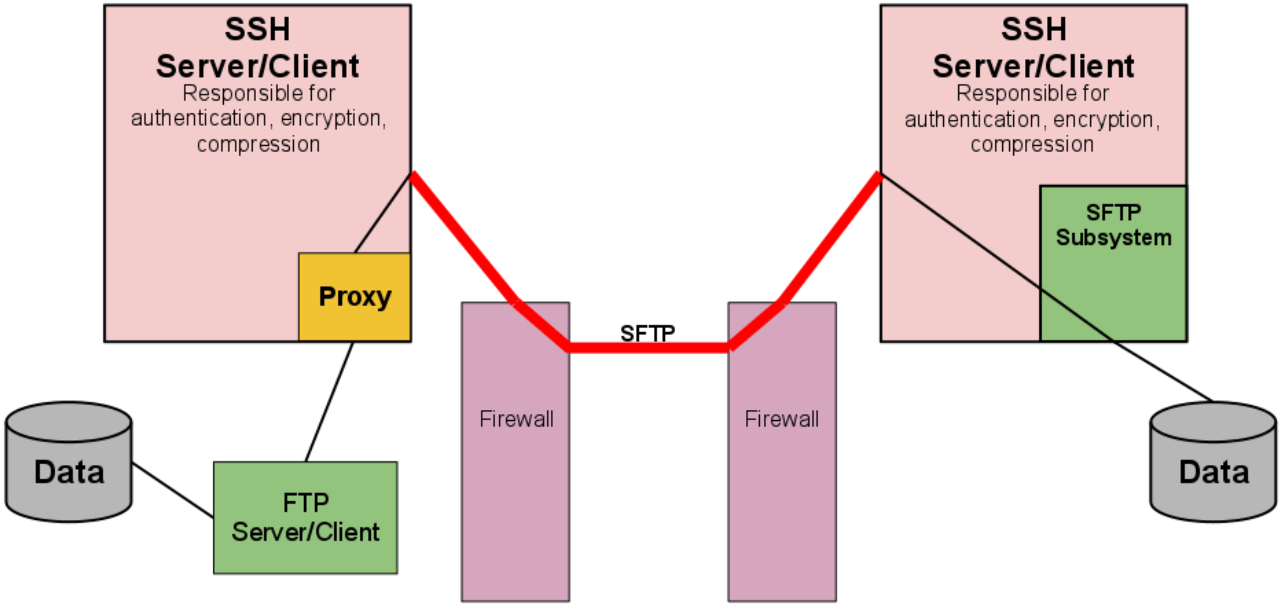
- Pros
  - Satisfies SFTP requirement
  - Can still use the FTP client on the z/OS side – Configure FTP jobsteps to use Proxy

FTP to SFTP Conversion (Vendor Solution)


SSH
Server/Client
Responsible for authentication, encryption, compression
Proxy
Data
FTP
Server/Client
Firewall
SFTP
Firewall
SSH
Server/Client
Responsible for authentication, encryption, compression
SFTP
Subsystem
Data

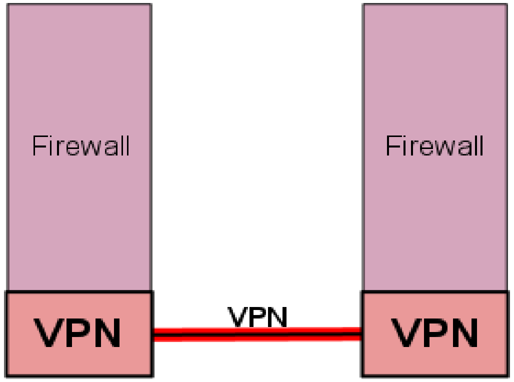
- Pros
  - Satisfies SFTP requirement
  - Can still use the FTP client on the z/OS side
- Cons
  - FTP / SFTP not a perfect match of functions

FTP to SFTP Conversion (Vendor Solution)



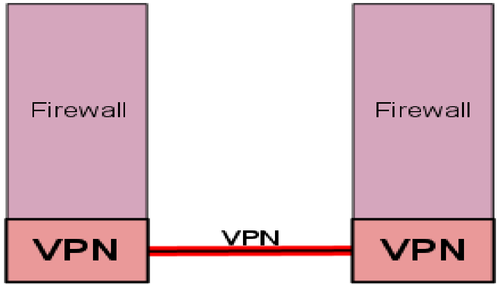
- Common uses
  - Leveraging FTP already in place, but transitioning it to your SFTP knowledgeable partners

VPN



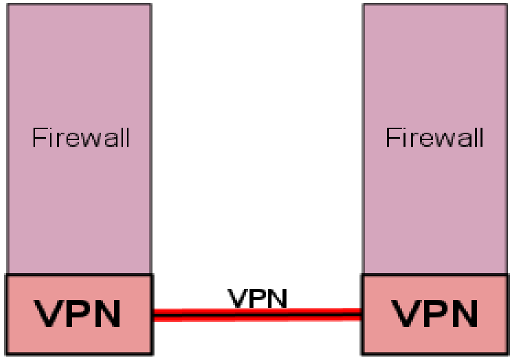
- Pros
  - Network to Network encryption (everything covered)

VPN



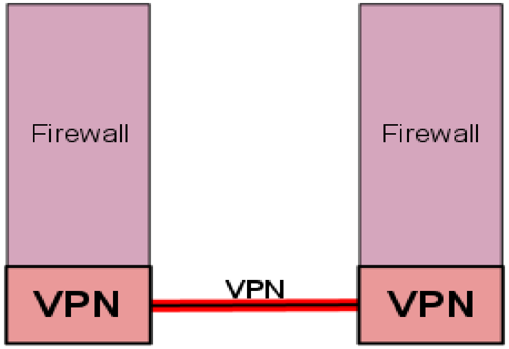
- Pros
  - Network to Network encryption (everything covered)
  - Some integrity built-in

VPN



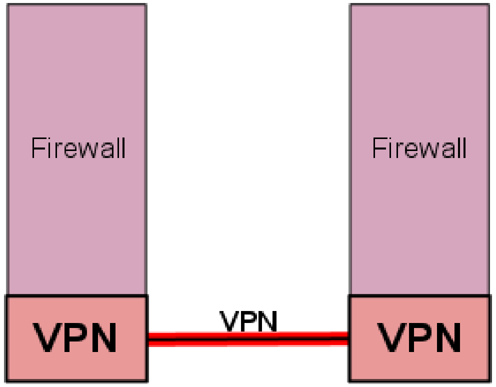
- Pros
  - Network to Network encryption (everything covered)
  - Some integrity built-in
  - Compression might be included

VPN



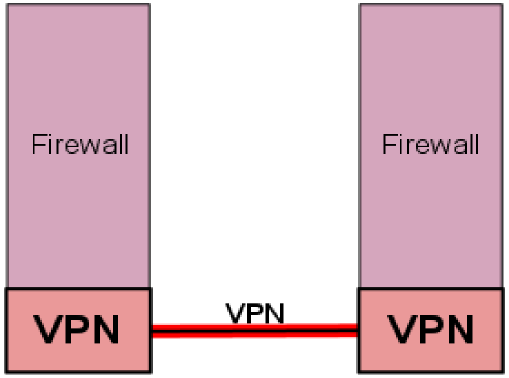
- Pros
  - Network to Network encryption (everything covered)
  - Some integrity built-in
  - Compression might be included
  - Transparent to the applications

VPN



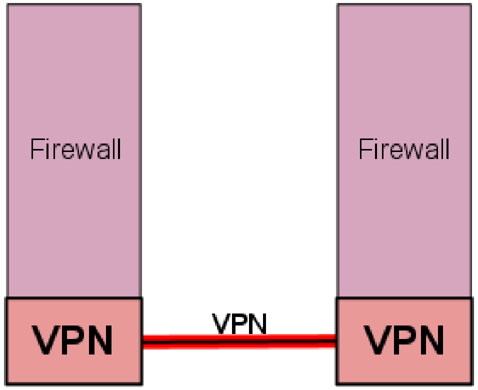
- Pros
  - Network to Network encryption (everything covered)
  - Some integrity built-in
  - Compression might be included
  - Transparent to the applications
- Cons
  - More complex to set up

VPN



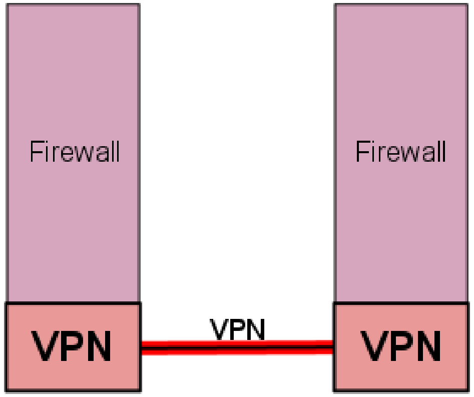
- Pros
  - Network to Network encryption (everything covered)
  - Some integrity built-in
  - Compression might be included
  - Transparent to the applications
- Cons
  - More complex to set up
  - Intranet traffic is unprotected

VPN



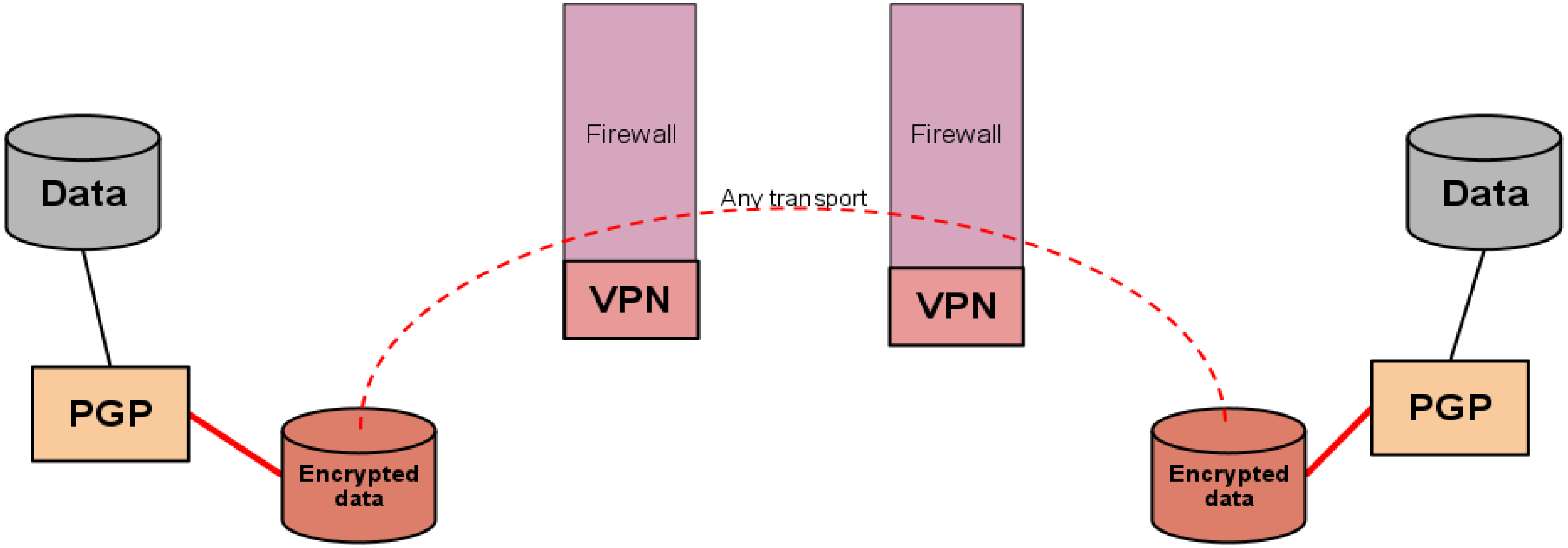
- Pros
  - Network to Network encryption (everything covered)
  - Some integrity built-in
  - Compression might be included
  - Transparent to the applications
- Cons
  - More complex to set up
  - Intranet traffic is unprotected
  - Usually managed by another group

VPN



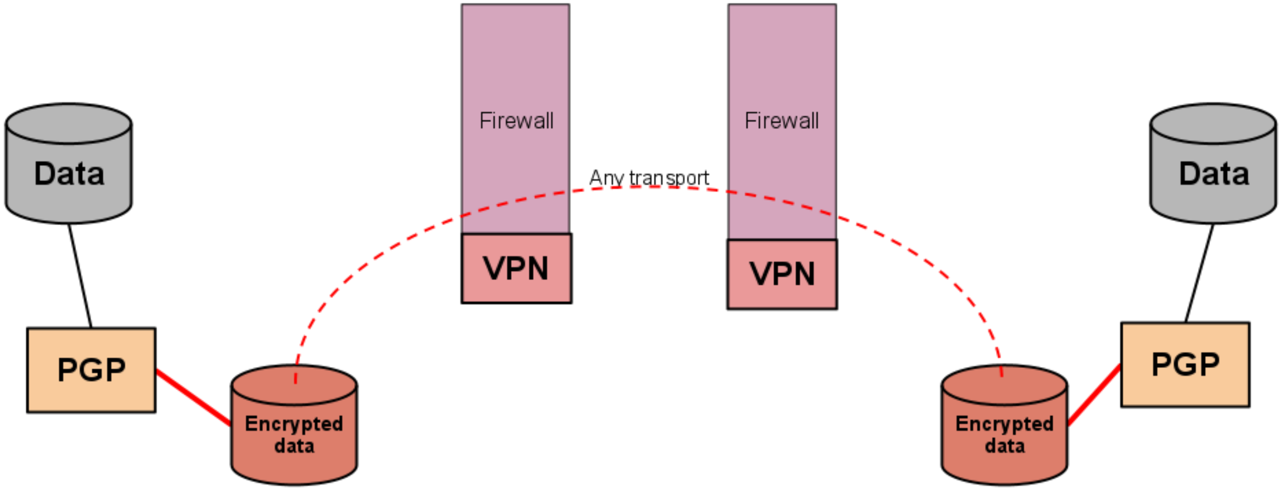
- Common uses
  - Trusted partner networks

PGP (Data at Rest)



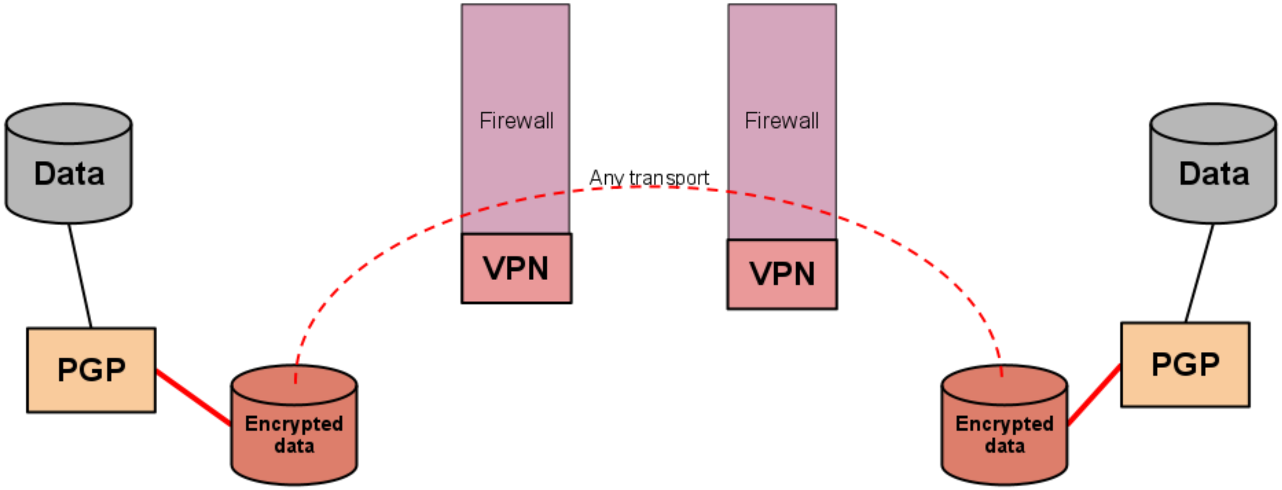
- Pros
  - Full control of sensitive data

PGP (Data at Rest)



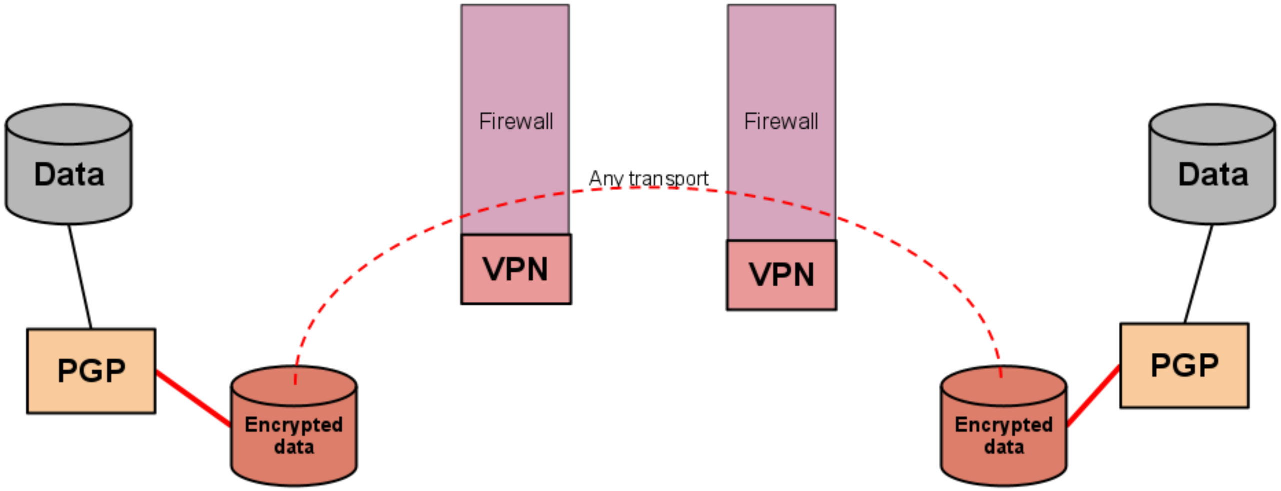
- Pros
  - Full control of sensitive data
  - Transport is not important

PGP (Data at Rest)



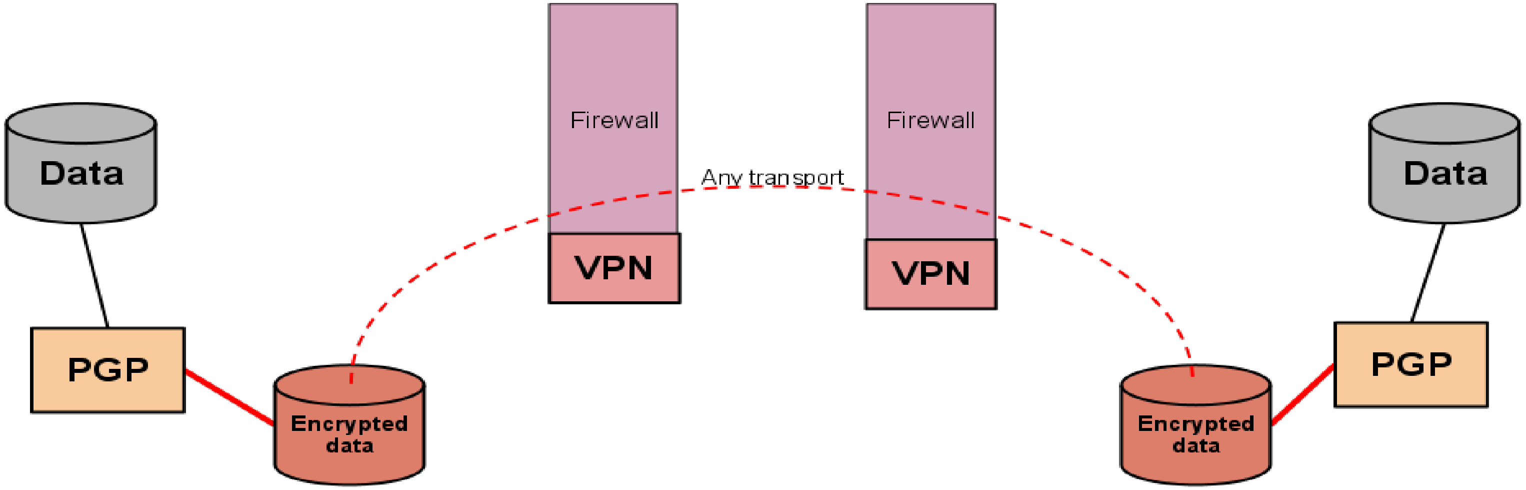
- Pros
  - Full control of sensitive data
  - Transport is not important
  - Compression and Integrity

PGP (Data at Rest)



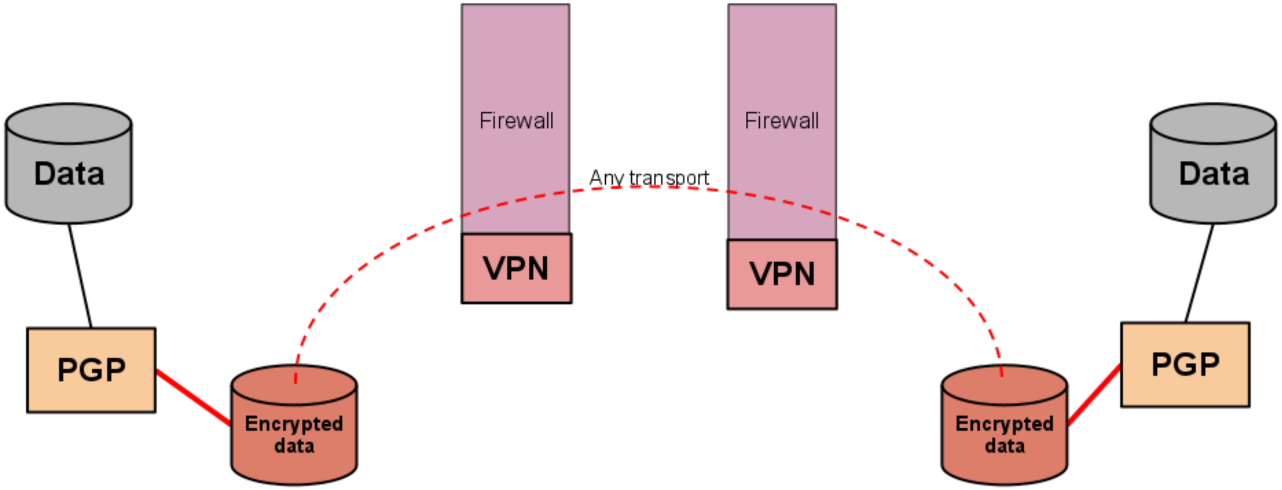
- Pros
  - Full control of sensitive data
  - Transport is not important
  - Compression and Integrity
  - Not just for transfers

PGP (Data at Rest)



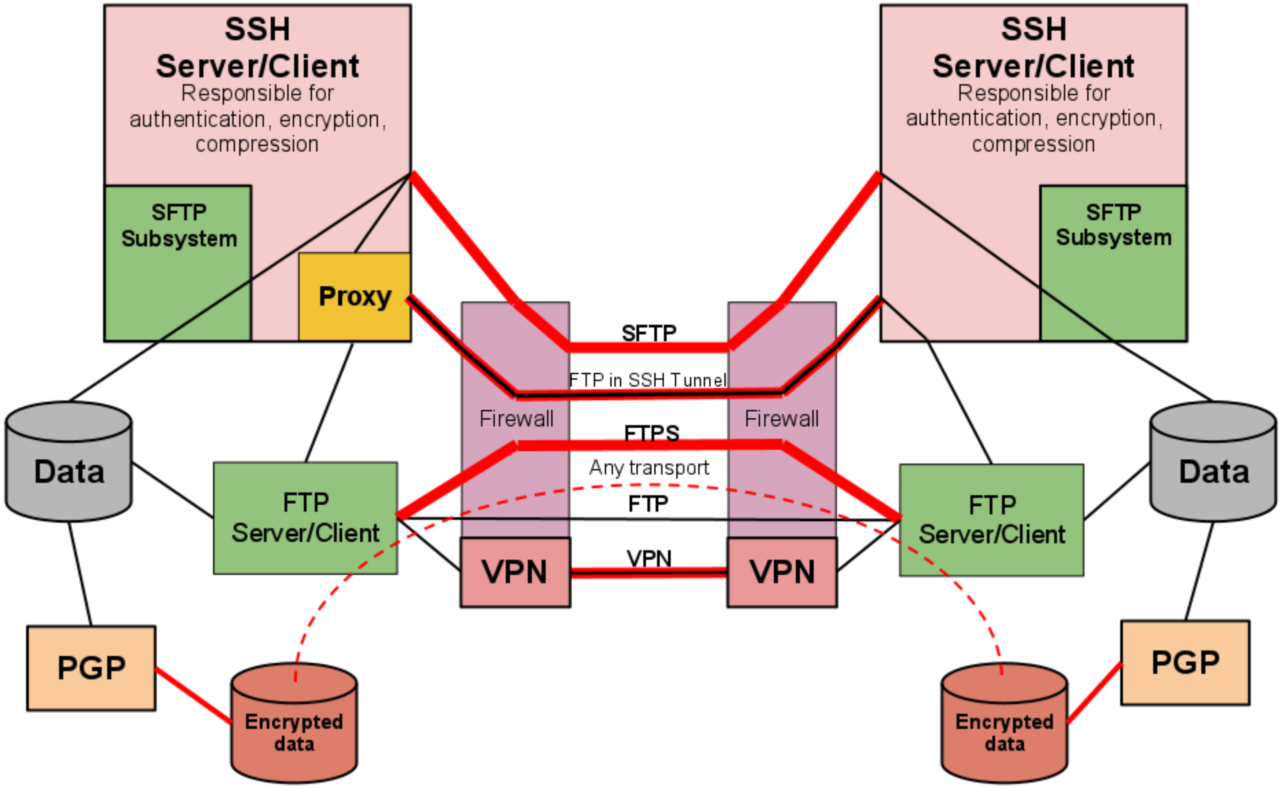
- Pros
  - Full control of sensitive data
  - Transport is not important
  - Compression and Integrity
  - Not just for transfers
- Cons
  - Requires staging of data

PGP (Data at Rest)



- Common uses
  - Sensitive data that needs protection at destination as well as in transit

FTP – All The Options



- Common uses
  - Mixed requirements – unfortunately, one size rarely fits all properly

Summary

- What is that you wish to accomplish ?
- Evaluate each solution and determine which solution/s is best for your company
- Implement one / or more solutions
- Regular Audits to make sure your compliant

Thank You