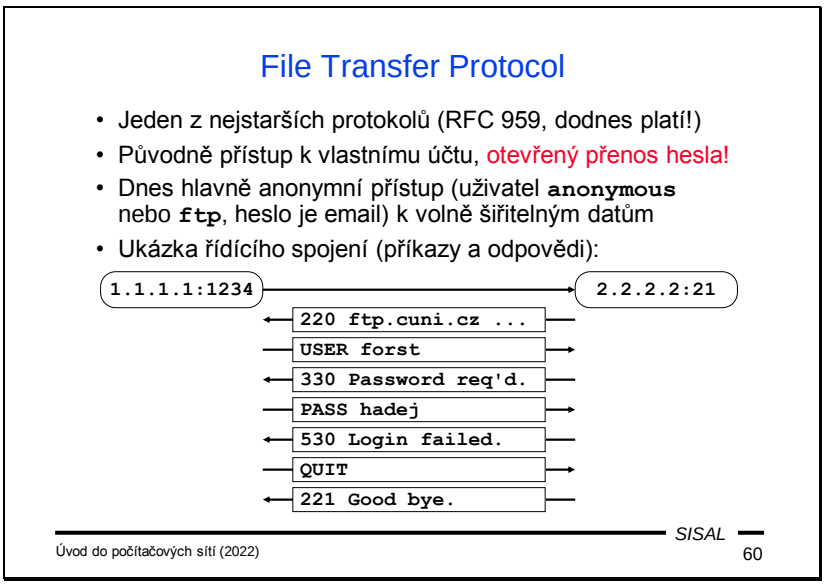
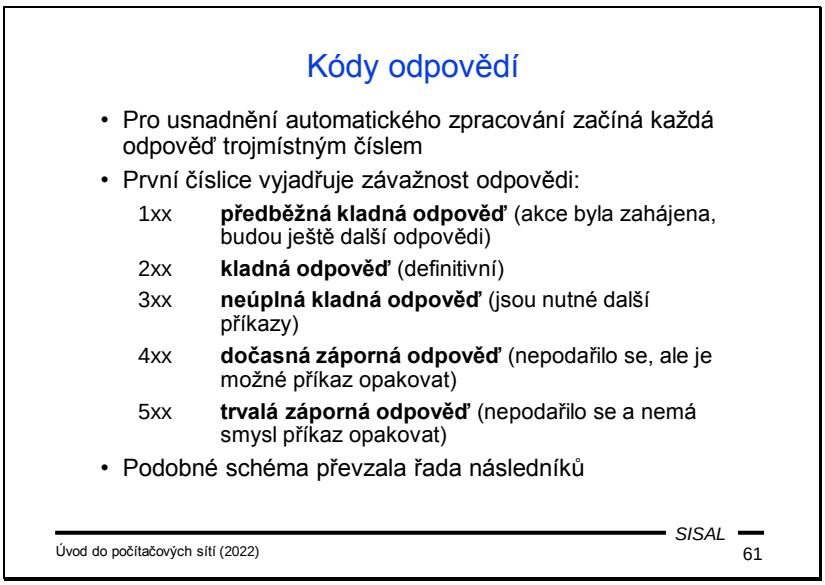
Snímek 60



File Transfer Protocol je jedním z nejstarších dodnes používaných protokolů. Jeho původním smyslem bylo umožnit vzdálený přístup k uživatelskému účtu za účelem přenosu souborů z nebo na vzdálený počítač. Pro přihlášení tímto způsobem je nutné uživatele autentikovat a to se ve FTP dělá přenosem otevřeného hesla, což v současné době představuje pochopitelné bezpečnostní riziko. Záhy však byl tento přístup doplněn stejně (a možná a více) důležitou, zato méně riskantní variantou, kdy se používá anonymní uživatel a místo hesla by se měla posílat emailová adresa uživatele (jen ze statistických důvodů). Takto přihlášený uživatel má pak přístup k volně dostupným dokumentům, programům apod. Ve své době znamenalo FTP stejně důležitý zdroj, jako dnes WWW. A některé servery dodnes používají FTP archivy, aniž to vlastně uživatel tuší – webové prohlížeče zobrazují odkaz na zdroj dostupný pomocí FTP stejným způsobem jako odkaz na HTTP, a když na ně uživatel klikne, navážou FTP spojení a výsledek dají uživateli stejným způsobem k dispozici. Běžný uživatel přitom vůbec nemusí zjistit, že celý přenos se odehrál jiným protokolem.

FTP je textový protokol; klient naváže tzv. *řídící spojení* na server na portu 21 a po něm posílá řádky s příkazy, zatímco server stejným kanálem posílá řádky s odpověďmi.

Snímek 61



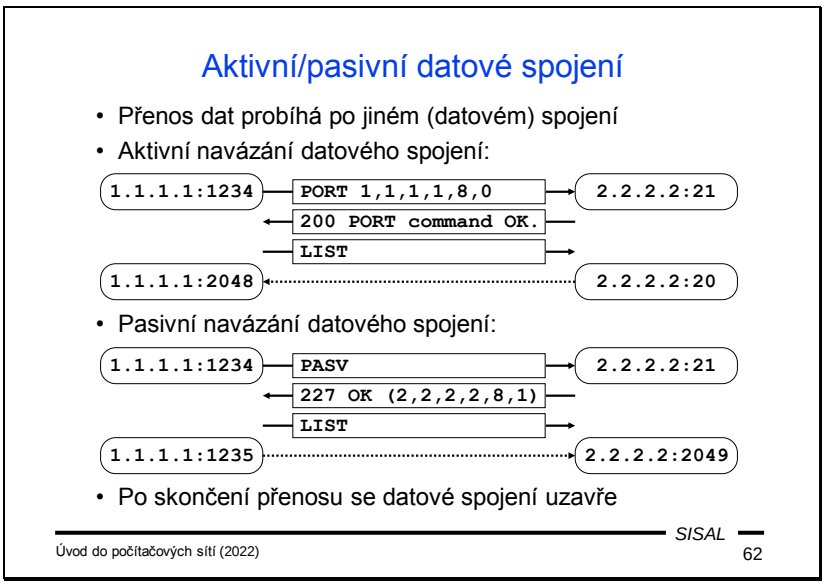
FTP byl jedním z prvních protokolů, který začal identifikovat odpovědi trojmístným kódem, kde první číslice vyjadřuje charakter odpovědi. Tento způsob umožňuje snadné zpracování odpovědí, aniž je třeba analyzovat text odpovědi, který navíc může být lokalizován do jiného jazyka.

Typy odpovědí:

- Předběžná kladná odpověď. Odpověď znamená, že server přijal požadavek a začal ho řešit. Od klienta není v dané chvíli očekávána žádná další akce, míč je na straně serveru.
- Definitivní kladná odpověď. Server dokončil úspěšně požadovanou operaci.
- Neúplná kladná odpověď. Odpověď znamená, že server přijal požadavek, ale pro jeho vyřešení potřebuje od klienta ještě dodatečné informace. Jaké informace to jsou, musí klient vědět z popisu příslušné operace v protokolu. Např. při operaci přihlášení uživatele musí klient na odpověď 3xx zareagovat posláním příkazu obsahujícího heslo. Jinými slovy – všechno je v pořádku, ale čeká se na aktivitu klienta.
- Dočasná záporná odpověď. Operace se nepovedla, nicméně chyba není fatální. Obvyklým důvodem je momentální přetížení serveru. Klient může požadavek zopakovat po nějaké době v identickém znění a je šance, že požadavek uspěje.
- Trvalá záporná odpověď. Tato chyba je fatální a ani opakované zaslání shodného požadavku nebude úspěšné.

Stejný mechanizmus později převzala i řada jiných protokolů, některé se shodným významem prefixu (např. SMTP), jiné s mírnými odlišnostmi (např. HTTP, určitě už jste někdy dostali od serveru odpověď 404 – stránka nenalezena).

Snímek 62



Druhým aspektem FTP, který má přesah i do současných protokolů, je používání dodatečných datových kanálů.

Kromě řídícího spojení, po kterém se posílají jen požadavky klienta a odpovědi serveru, se totiž ve FTP otevírají zvláštní *datová spojení*, pomocí nichž se přenáší veškerý datový obsah. Na každý nový přenos (download souboru, upload souboru nebo download výpisu obsahu adresáře) se musí otevřít nové TCP spojení, které se po dokončení přenosu zase zavře. Jenže s otevíráním dodatečných datových kanálů přicházejí problémy – obě strany se musejí dohodnout, **kdo** bude datový kanál otevírat a na **jakou adresu a port** se má spojení navázat.

FTP rozlišuje dva způsoby otevírání datového kanálu nazývané podle toho, jakou roli při nich hraje server:

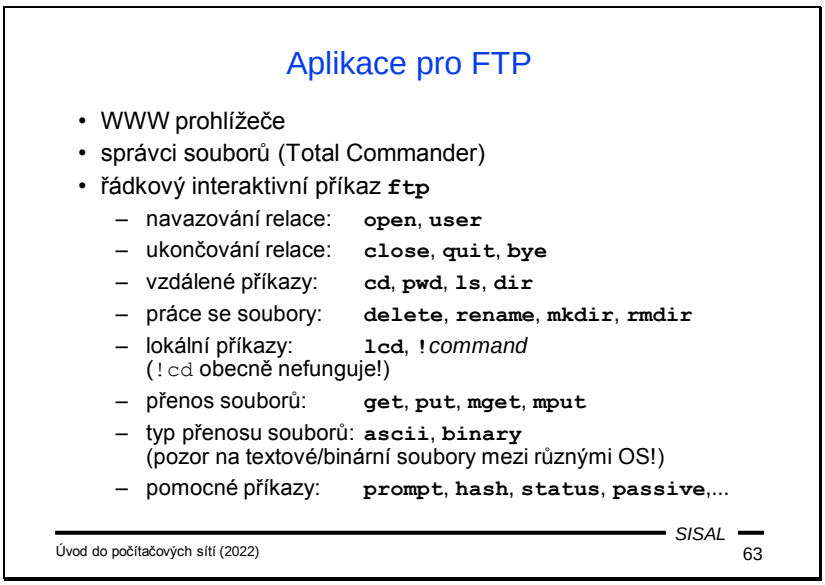
- **Aktivní** datové spojení navazuje server. Server teoreticky nepotřebuje žádné další informace – podle původní koncepce totiž navazuje spojení na stejnou adresu a port, kterou používá klient. Na své straně spojení použije svou adresu, ovšem použít port 21 nemůže (to by všechny parametry nového spojení byly shodné se spojením řídícím a to není možné), takže proto byl vyhrazen ještě jeden port 20 nazvaný *ftp-data*, který server může použít jako zdrojový. Alternativou je samozřejmě použití generického portu, podobně jako to dělá běžný klient při navazování spojení. Řešení s použitím adresy a portu klienta jako cílové adresy pro datové spojení není ale úplně praktické, a proto dnes většina implementací zahajuje aktivní datové spojení příkazem PORT (případně EPRT pro IPv6), kterým serveru dává vědět jinou adresu a/nebo port. Syntaxe příkazu je možná maličko zarážející, ale je to sice hůře čitelný, ale efektivní zápis – je to prostě šest bajtů adresy socketu (4 bajty IP adresy a 2 bajty portu).

- Vedle aktivního způsobu navázání datového spojení existuje ještě i způsob **pasivní**. V tomto případě bude spojení navazovat klient a bude tedy potřebovat od serveru adresu a port. O ně si požádá příkazem PASV (resp. EPSV pro IPv6) a server reaguje odpovědí, která obsahuje v závorce požadovanou adresu socketu.

Návrh řešení navazování datových kanálů odpovídá stavu, v jakém se problematika bezpečnosti sítě nacházela v sedmdesátých letech minulého století. Všechny adresy byly veřejné a každá byla odkudkoliv přístupná. Ale v okamžiku, kdy začneme do sítě zavádět ochranná opatření a používat privátní adresy, se situace komplikuje. V případě aktivního spojení server nejspíše nebude mít dovoleno otevřít spojení směrem ke klientovi, i kdyby ten měl veřejnou adresu. Klient ale spíše bude mít adresu privátní, na kterou se server spojit už vůbec nedokáže. Řešením je zapojení vyšší logiky do překladu adres, kdy router provádějící NAT musí modifikovat dokonce i **obsah** zpráv mezi klientem a serverem a měnit v nich patřičným způsobem posílané adresy. U pasivního spojení je situace o něco lepší, server posílá klientovi svoji veřejnou adresu, takže jediným omezením může být, pokud router na perimetru naší sítě umožňuje otevírání spojení pouze na **definovaný seznam** portů.

Problematika otevírání dodatečných datových kanálů není specifikem pouze FTP. Např. protokol SIP, kterým se dnes nejčastěji realizuje přenos hlasu po TCP/IP síti, řeší úplně stejné problémy.

Jen připomínám, že směr, kterým se otevírá datové spojení, nijak nesouvisí se směrem, kterým posléze potečou kanálem data.



Pro přístup k protokolu FTP lze v současnosti použít:

- Webové prohlížeče. K downloadu nebo výpisu adresáře stačí zadat patřičné URL, pro upload je často nutné nainstalovat nějaké rozšíření.
- Správci souborů. Řada aplikací pro správu souborů umí s FTP pracovat. Např. Total Commander umí v panelu zobrazit místo obsahu lokálního disku obsah adresáře na FTP serveru a se soubory provádět běžné operace. Pro studium FTP je zajímavé, že v dodatečném okně je možné sledovat FTP příkazy, kterými jsou jednotlivé operace realizovány.
- Poslední záchranou je nejstarší aplikace, interaktivní program **ftp**, který přijímá řádkové příkazy. Většina z nich má logický význam, jen je třeba při přenosech dávat pozor na přenos souborů mezi operačními systémy, které používají jiné konce řádek u textových souborů (např. Windows vs. UNIX) – pokud nezvolíme správný režim přenosu (**ascii** pro textové a **binary** pro binární), budou soubory na cílovém počítači nečitelné resp. poškozené.

Elektronická pošta

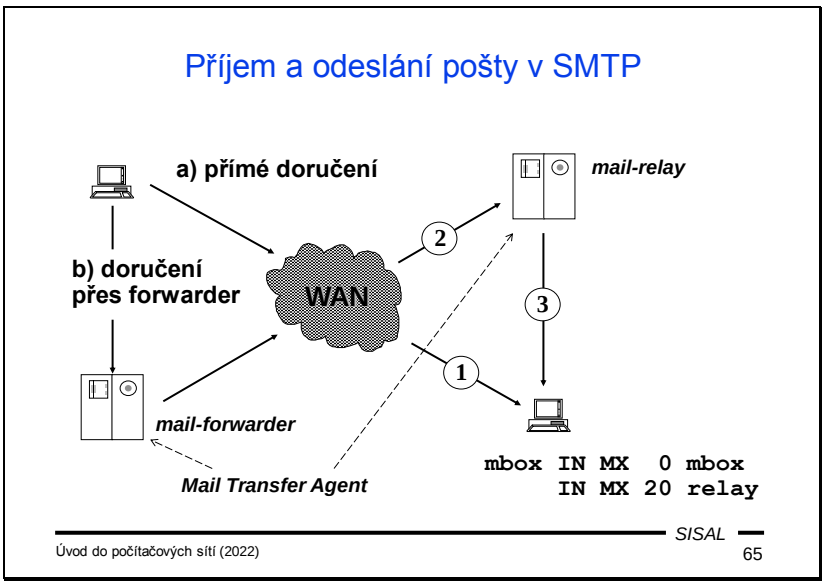
- Obecná služba, existuje i mimo Internet
  - off-line předávání zpráv příp. souborů
  - off-line použití informačních služeb
  - diskusní kluby (mailing-listy, konference)
  - komunikace mimo Internet
- Na Internetu funguje na základě RFC 821, 2821 a 5321 (protokol SMTP resp. ESMTP) a RFC 822, 2822 a 5322 (formát zpráv) na portu 25
- E-mailová adresa v Internetu (typicky):
  login`@`*počítač* nebo *alias*`@`*doména*
  např.:
  `forst@ms.ms.mff.cuni.cz, Libor.Forst@cuni.cz`

Elektronická pošta je další z velmi starých služeb, pocházející z doby před „moderním Internetem“. Už v počítačové prehistorii sloužila jako jeden z mála komunikačních mostů mezi počítačovými sítěmi nejrůznějšího charakteru a tomu také odpovídala variabilita formátů poštovních adres. V současné době se na internetu ustálila podoba složená za dvou částí oddělených zavináčem. V základním tvaru je před zavináčem název poštovní schránky a za ním název serveru, kde schránka leží. Tento tvar je ale jednak příliš závislý na aktuálním stavu (změna názvu serveru by znamenala změnu adresy u všech uživatelů) a jednak představuje riziko, protože případný útočník z adresy přečte jak jméno serveru, tak jméno účtu a může se ho pokusit napadnout. Z obou příčin se tedy častěji používá namísto jména účtu nějaký alias a namísto konkrétního jména serveru jméno domény nebo služby.

Původní koncept měl sloužit k přenosu krátkých zpráv (do 64 kB), buďto jako osobní korespondence, korespondence uvnitř určité skupiny lidí (mailing-listu), ale také jako off-line metoda používání různých služeb (např. vyhledávání dokumentů ve FTP archivech). Postupem času se ale začala používat i přenosu souborů, což ale přináší určité komplikace.

Na internetu se pro přenos pošty používá protokol SMTP, textový protokol na TCP portu 25 s principem posílání zpráv a odpovědí, podobně jako je tomu u FTP.

Snímek 65



Pojďme se nyní podívat, co se děje, když uživatel vydá povel k odeslání dopisu.

Poštovní program zkontroluje část adresy za zavináčem a zjistí, jaký server bere poštu pro danou doménu. Teoreticky je možné, že v cestě mezi klientským počítačem a cílovým serverem nestojí žádné překážky a program tedy může navázat SMTP spojení na cílový server a pokusit se dopis předat. Nebývá to ovšem zvykem, dopisu se do cesty většinou staví několik uzlů, které si dopis postupně předávají.

Na straně odesilatele bývá důvodem většinou to, že poštovní program má zjednodušenou konfiguraci a namísto složitých pravidel se v něm pouze nastaví, že mail předává pomocí SMTP nějakému serveru v lokální síti (tzv. *mail-forwarderu*) určenému pro další zpracování pošty. Tomuto kroku se říká *mail-submission*. Takových forwarderů může být případně i více, např. kvůli několika stupňům překladu adres apod.

Každý uzel, který přijímá a dále doručuje poštu, se nazývá Mail Transfer Agent (MTA). Jednotlivé MTA si dopis předávají pomocí SMTP protokolu, kde odesílající MTA dočasně funguje jako klient a přijímající MTA jako server. Pokud server není cílový, uloží dopis do fronty a podle lokálních pravidel se periodicky pokouší dopis doručit dalšímu MTA. Poznamenejme, že počet MTA po cestě nijak nesouvisí s počtem sítí resp. routerů.

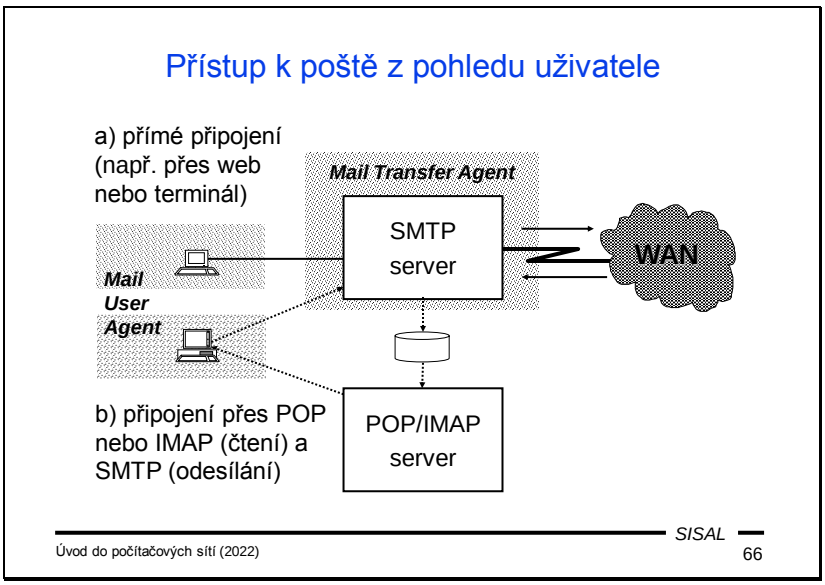
Pokud je cílový mailbox na serveru, na který je volný přístup z internetu, poslední MTA se pokusí mail doručit na tento server. V případě, že doručení není možné, mail zůstává ve frontě na posledním MTA (obecně „někde v internetu“). Pokud tomu chce správce poštovního serveru zabránit, může v DNS nastavit pro svůj server MX záznam. MX záznamy obsahují jméno *mail exchangeru*, který je oprávněn dočasně

nebo trvale přijímat poštu pro daný stroj nebo doménu, a jeho prioritu (čím nižší číslo, tím vyšší priorita). V takovém případě doručující MTA zkouší postupně jednotlivé exchangery, podle priorit, dokud neuspěje.

Celou cestu přišedšího mailu internetem může příjemce sledovat pomocí hlaviček Received v doručeném dopisu.

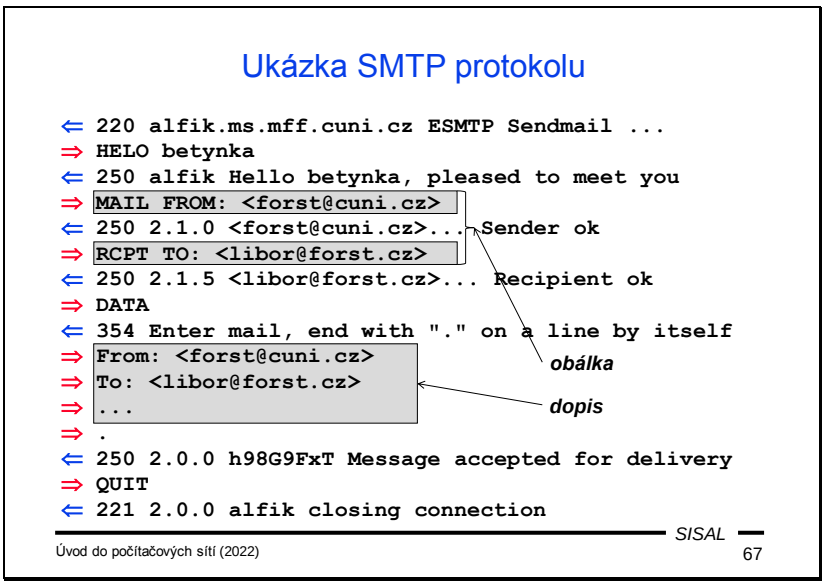
Vzhledem k tomu, jak probíhá doručování dopisu, je asi zřejmé, proč tento protokol není vhodný pro předávání **velkých** souborů. Každý MTA uzel na cestě musí dopis uložit někam do svojí fronty jen proto, aby ho vzápětí z fronty vyjmul a poslal dál.

Snímek 66



Z hlediska uživatele je poštovní systém přístupný prostřednictvím nějakého „poštovního programu“ nazývaného Mail User Agent (MUA). V principu existují dvě možnosti, jak je MUA do systému připojen:

- První možností je přímé připojení. Uživatel se ze svého počítače připojí na MTA, kde má svůj mailbox. Způsob připojení může být různý, např. vzdálené přihlášení se k systému, anebo webová aplikace běžící na serveru s poštou, a tomu také odpovídají možnosti autentikace uživatele. Vlastní aplikace má v tomto případě přímý přístup k dopisům v mailboxu uživatele a zároveň k lokálním službám MTA, takže odesílané zprávy zařazuje přímo do fronty MTA.
- Alternativou je připojení k mailboxu pomocí speciálního poštovního protokolu POP nebo IMAP. Klient tohoto protokolu se připojí na server běžící na MTA, autentikuje se v tomto protokolu a na základě uživatelských požadavků zadává serveru povely pro zacházení s mailboxem. Tyto povely ovšem zahrnují pouze práci s **doručenými** dopisy. Pro odesílání musí klient použít normální mail submission pomocí SMTP, a to nikoliv nutně na stejný server. Proto také v konfiguraci aplikace nalezneme oddělené sekce pro POP/IMAP a SMTP. Například v situaci, kdy uživatel cestuje a připojuje se k vlastnímu serveru „zvenku“, nemusí být jeho server ochoten od něj přijmout dopis k odeslání, protože SMTP nemá obecně povinnou autentikaci, takže pro server nemusí být uživatel dostatečně věrohodný. Takový případ lze řešit např. tím, že aplikace bude dočasně používat k odesílání dopisů lokální server v místě připojení.



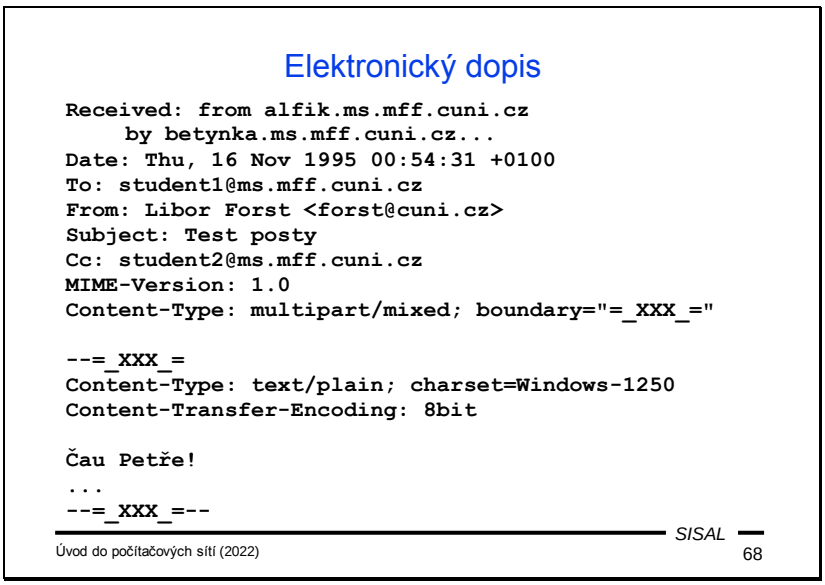
Protokol SMTP se podobá FTP, klient posílá jednotlivé příkazy jako textové řádky a server podobně odpovídá, a to i včetně trojmístného kódu se shodným významem první číslice.

Nový dopis začíná klient příkazem MAIL FROM, který má jako parametr (ve špičatých závorkách) adresu odesilatele. Po tomto příkazu následuje jeden nebo více příkazů RCPT TO s adresou jednoho příjemce. Každého příjemce zvlášť server potvrzuje – pokud se rozhodne některého neakceptovat, odpoví místo zprávy 250 zprávou 450 (dočasná chyba) nebo 550 (trvalá chyba). Pokud ale odpoví 250, **přebírá tím zodpovědnost** za doručení dopisu, anebo odeslání zprávy o neúspěšném doručení, tzv. Delivery Status Notification (DSN). Po příkazu s posledním adresátem následuje příkaz DATA, na který server odpoví zprávou 354, načež klient začne posílat text dopisu, tedy zhruba to, co posléze příjemce vidí ve svém mailboxu. Protokol je textový, protože tak byly původně zprávy zamýšleny, a končí řádkou, která obsahuje pouze samotnou tečku. Klienti SMTP protokolu musejí proto dávat pozor na řádky textu dopisu, které začínají tečkou (anebo bajty binárního kódu, které shodou okolností tvoří posloupnost CR LF tečka!), ty serveru posílají s předsunutou jednou tečkou navíc, kterou server při ukládání dopisu zase odmazává. Na ukončovací řádku s tečkou server reaguje standardní odpovědí (250, 450 nebo 550) a klient může skončit příkazem QUIT, anebo pokračovat dalším dopisem.

Poznámka 1: Všimněte si vztahu mezi adresami, které klient sdělil serveru jako adresu odesilatele a příjemců, a těmi, které pak příjemce vidí. **Žádný neexistuje!** Adresám použitým v protokolu se říká *obálka* a jejich vztah k tělu dopisu je stejný jako vztah adresy napsané na skutečné poštovní obálce a adres napsaných na

dopisu vloženém do obálky. Je to zkrátka pouze text, který odesilatel může libovolně zfalšovat, nicméně se jím, bohužel, následně řídí váš poštovní program…

Poznámka 2: DSN (Delivery Status Notification) dopisy má za povinnost generovat ten MTA, který přijal dopis (odpověděl 250 na konkrétního adresáta i na poslání těla dopisu), ale jemuž se nepodařilo dopis doručit nebo předat dalšímu MTA. Protože odesilatelem je v tomto případě sám poštovní systém, v příkazu MAIL FROM se **nevyplňuje** žádná adresa. Takovéto dopisy mívají na některých uzlech usnadněný průchod (např. se méně důsledně kontroluje, zda obsahují viry nebo spam), a právě proto tento způsob odesílání zneužívají některé spamovací automaty. Bohužel to některé správce vede naopak k tomu, že doručování DSN jednoduše **zakážou**. To je ovšem nehorázný zásah do práv jejich uživatelů, protože celý mailový systém je postaven na principu *best-effort*, jehož klíčovým prvkem je, že o případném neúspěchu doručení bude odesilatel vyrozuměn.



Dopis se skládá ze dvou částí oddělených prázdnou řádkou.

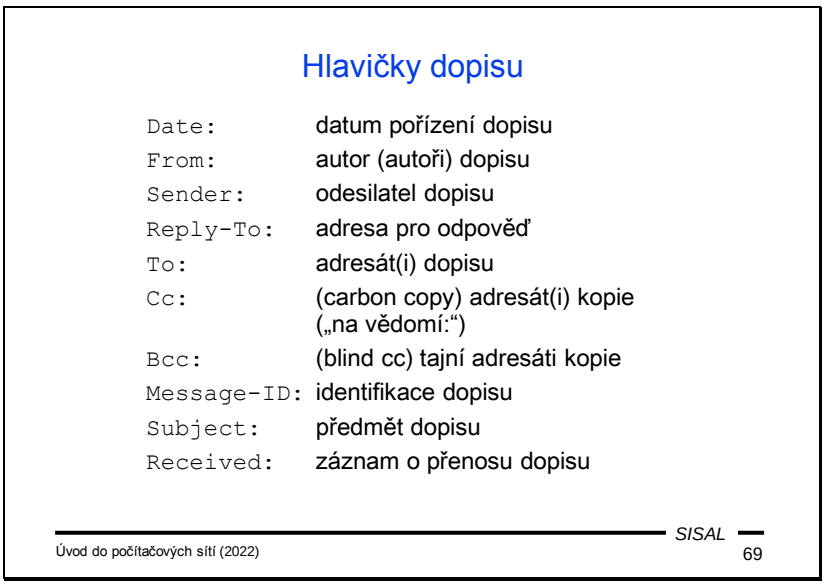
První část je tzv. *záhlaví*, které obsahuje *hlavičky*, řádky s řídícími informacemi, s relativně pevným formátem a se znaky výhradně z tabulky ASCII (tedy pouze sedmibitové znaky 0x00 až 0x7f, žádné znaky národních abeced). Hlavičky jsou určeny jak pro informaci pro koncového uživatele, tak pro práci mailovacích programů, které obvykle uživateli zobrazují jen vybranou část hlaviček, nicméně mívají operaci nebo nastavení, které umožní uživateli vidět záhlaví celé.

Druhou část tvoří text dopisu. V původním konceptu bylo i zde možné používat pouze ASCII znaky a psát jen prostý text.

Postupem času se ale prosadila možnost používat v textu dopisu znaky národních abeced. Existuje rozšíření protokolu nazývané ESMTP, v jehož rámci se mohou klient a server domluvit, že oba jsou schopni akceptovat i osmibitové znaky – klient pak může použít tzv. **8-bit** kódování obsahu při přenosu.

Druhou zásadní novinkou byla možnost definovat **strukturu a sémantiku** těla dopisu pomocí rozšíření MIME. Díky tomu bylo možné začít pracovat např. s přílohami obsahujícími soubory.

Snímek 69

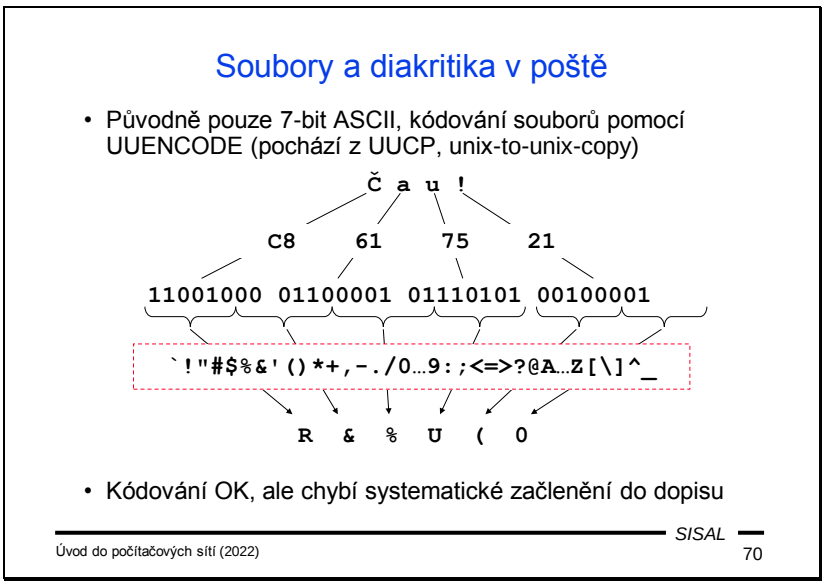


Nejdůležitější hlavičky:

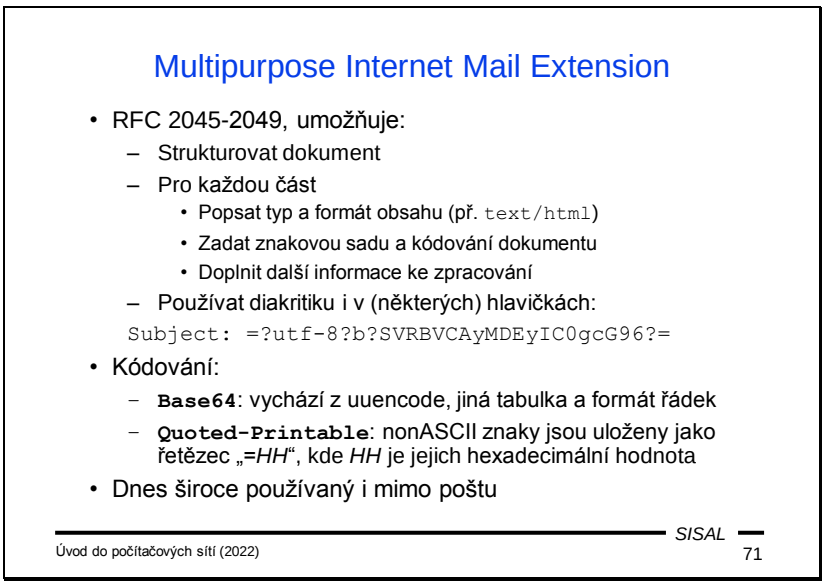
- Date – obsahuje datum vzniku dopisu v pevném (americkém) formátu.
- From – obsahuje adresu autora (nebo autorů) dopisu; buďto prostou e-mailovou adresu, anebo adresu doplněnou o komentář (obvykle jméno a příjmení), a to nejčastěji ve formátu „komentář <adresa>“ anebo „(komentář) adresa“. Hlavička byla opravdu zamýšlena jako seznam autorů, neboli pokud dopis tvoří více lidí, měli by zde být uvedeni všichni. Mailovací aplikace pro to ale obvykle nemají nijak propracovanou podporu, takže uživatelé to ignorují a píšou obvykle do From jen jediného autora. Naopak, uvedení více „autorů“ bývá často příznakem spamu – spamovací automaty totiž vycházejí z pochybného předpokladu, že pokud ve From použijí „známou adresu“, mail snáze projde kontrolou, a proto mívají ve From dlouhý seznam adres.
- Sender – je naopak hlavička, kde smí být jen jedna adresa, pokud dopis posílá osoba, která není ve From (např. sekretářka za svého šéfa), anebo pro vyznačení, kdo ze seznamu ve From dopis skutečně odeslal.
- Reply-To – obsahuje adresu, kterou by příjemce měl použít při odesílání odpovědi (např. adresa mailing listu, ze kterého dopis přišel).
- To – obsahuje seznam adresátů dopisu.
- Cc – obsahuje seznam adresátů, kterým je poslána kopie dopisu. Logicky by to měli být lidé, kteří nejsou očekáváni jako účastníci dialogu a jsou jen „v kopii“. Pozor ale na to, že běžný uživatel nekouká při příjmu dopisu na to, zda byl uveden jako To nebo Cc. Není proto dobré psát kamarádovi dopis s oslovením „Nazdar, kámo“ a v kopii ho poslat panu děkanovi. Praktický problém je i v tom, že řada poštovních aplikací při vytváření odpovědi původní To a Cc adresáty pomíchá. Název je odvozen od pojmenování kopírovacího papíru, kterým se při psaní na stroji pořizovaly kopie.

- Bcc – („blind carbon copy“) je hlavička, která se nevyskytuje ve skutečném dopisu, zobrazuje ji pouze poštovní program a adresáta přidá do SMTP obálky, ale nikoliv do textu. Ostatní příjemci tedy nevědí o tom, že daný člověk byl rovněž mezi adresáty.
- Message-ID – je technická hlavička, kterou vyrábí odesílací poštovní program a program příjemce by ji měl použít při přípravě odpovědi. Umožňuje logické sdružování odpovědí do diskusních „vláken“.
- Subject – obsahuje stručné shrnutí obsahu dopisu (jako se v obchodních dopisech používá „Věc“).
- Received – je technická hlavička, kterou do dopisu doplňuje obvykle každý uzel, který dopis přeposílal. Obsahuje název MTA, čas doručení, lokální identifikátor a případně další údaje, které mohou usnadnit stopování dopisu.

Snímek 70



Jak už bylo řečeno, v původním návrhu bylo možné používat v celém dopisu pouze ASCII znaky. Jenomže uživatelé potřebovali občas přenést i soubory, které měly binární charakter. Podobný problém už ale v té době měli uživatelé UNIXové sítě UUCP (UNIX-to-UNIX copy), kteří také potřebovali posílat soubory sedmibitovým kanálem a kteří vymysleli kódování UUENCODE, jež jim to umožňovalo. V tomto kódování se vezmou vždy tři bajty původního souboru, těchto 24 bitů se rozdělí do čtyř skupin po šesti bitech a tyto šestibitové kódy se převedou na čtyři tisknutelné znaky pomocí pevné tabulky. Zakódovaný soubor má tedy oproti originálu o 33 % větší velikost. UUENCODE vzniklo v době, kdy nebylo běžné na počítači rozlišovat malá a velké písmena, takže v tabulce bylo jen 26 (velkých) písmen, 10 číslic a zbylých 28 míst musely zaujmout téměř všechny ostatní znaky včetně takových, které mají v různých prostředích speciální významy (středník, apostrof, uvozovka, hvězdička, hranatá závorka, zavináč aj.). Nicméně kódování bylo na světě, zakódovaný soubor se ještě obložil řádkami begin a end a bylo možné ho vložit do textu dopisu. Problémem tohoto řešení byl hlavně fakt, že složení dopisu, tj. zda, kde a jaké vložené soubory obsahuje, se dalo zjistit pouze analýzou celého textu dopisu.



Problém strukturování dokumentu vyřešilo rozšíření MIME (Multipurpose Internet Mail Extension), které se později navzdory svému jménu začalo používat i v řadě jiných protokolů (např. HTTP).

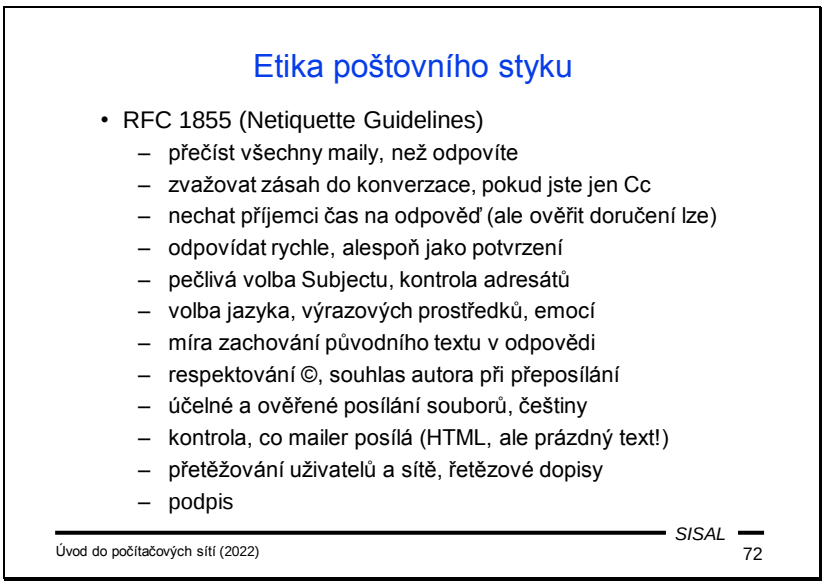
Dokument v MIME formátu obsahuje hlavičku a tělo (podobně jako mail samotný), přičemž hlavička mj. definuje:
- typ dokumentu; MIME typy se skládají z hlavního typu a podtypu (např. text/html, image/jpeg, audio/mp3, application/pdf)
- použitou znakovou sadu
- způsob kódování
- původní název souboru
- předpokládaný způsob zpracování (zobrazit jako přílohu nebo vložit do textu)…

Jedním z klíčových hlavních typů dokumentu je ovšem **multipart**, což znamená, že tělo dokumentu je dále strukturované. Odesílající aplikace vygeneruje náhodný řetězec, který se použije jako oddělovač jednotlivých částí strukturovaného dokumentu a každá z těchto částí je opět standardní MIME dokument. Pokud dnes v běžném poštovním programu připojíte k dopisu dva soubory, bude dopis poslán jako MIME multipart dokument, kde první část bude text dopisu a další části budou přiložené soubory.

MIME definuje dvě základní metody kódování:
- Metoda nazvaná **Base64** není ve své podstatě nic jiného než staré známé UUENCODE. MIME pouze nahradilo původní kódovací tabulku modernější (dnešní počítače nemají problém s malými a velkými písmeny, čímž dostáváme místo 26 celých 52 znaků a připočteme-li číslice, stačí doplnit dva nealfanumerické znaky, plus a lomítko) a trochu změnilo formát řádek.

- Metoda **Quoted-Printable** se používá hlavně pro textové soubory, u nichž většina textu je tvořena ASCII znaky. Každý non-ASCII znak se převede na sekvenci „=HH“, kde HH je hexadecimální hodnota znaku (např. rovnítko samotné je třeba zakódovat jako „=3D“). Výhodou oproti Base64 je to, že text zůstává alespoň trochu čitelný. Z hlediska efektivity kódování je důležitý poměr ASCII a non-ASCII znaků – pokud bychom takto kódovali text obsahující samé non-ASCII znaky, dostaneme se na objem kódu ve výši 300 % originálu (oproti pevným 133 % u Base64), naproti tomu zakódováním čistého ASCII textu se dostaneme jen zlehka nad 100 %.

Kromě výhod pro přenos dokumentů přineslo MIME také možnost vkládat non-ASCII znaky do (některých!) hlaviček (např. Subject). Kódování vypadá tak, že začíná znaky „=?“ následovanými identifikátorem použité znakové sady, otazníkem, identifikátorem použitého kódování („b“ nebo „q“), otazníkem, zakódovaným textem a závěrečnou sekvencí „?=“.

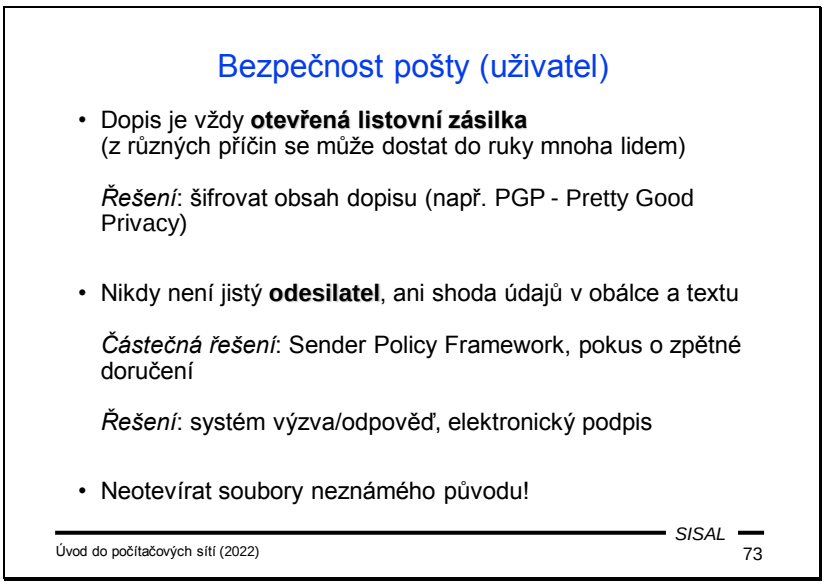


Styl, jakým by se měla používat elektronická pošta, podléhá určitým pravidlům. Některá z nich se podobají obecným pravidlům z neelektronické komunikace, některá jsou zcela nová. Mají povahu doporučení, ale většina z nich by se skutečně měla za běžných okolností dodržovat. Specifičnost elektronické komunikace vedla dokonce k tomu, že nejdůležitější pravidla byla sepsána formou RFC.

Jak by se tedy uživatel měl chovat:

- Když si otevřete svůj mailbox s příchozími dopisy, měli byate nejprve prohlédnout všechny došlé dopisy a teprve pak na ně začít odpovídat. Často se totiž ukáže, že diskuse v nějakém vláknu už pokročila tak, že vaše odpověď na první dopis v řadě by byla zbytečná nebo matoucí.
- RFC říká, že pokud jste pouze Cc adresát, měli byste pečlivě zvážit, když chcete do diskuse aktivně zasáhnout. Problém se současnými poštovními programy ale je, že po několikáté odpovědi už obvykle nepoznáte, zda jste původně byli Cc nebo To. Měli byste to ale většinou poznat z povahy dialogu.
- Pokud někomu pošlete dotaz či požadavek, měli byste mu nechat čas na odpověď. Pošta je off-line služba a vy nevíte, v jaké situaci se příjemce nachází, jak brzo se ke čtení pošty dostane. Samozřejmě ale můžete např. druhý den poslat slušně formulovaný dopis s prosbou alespoň o potvrzení, zda dopis došel, protože v tomto ohledu nikdy nemáte absolutní jistotu. Pochopitelně je rovněž možné v urgentním případě kontaktovat příjemce jinou cestou a upozornit ho na váš dopis.
- Komplementárním pravidlem ale je, že byste měli odpovídat rychle. Neboli po přečtení dopisu se rozhodnout, zda jste schopni se věnovat odpovědi v nějakém rozumném čase (podle povahy dopisu obvykle několik hodin, nejdéle do druhého dne), a v negativním případě alespoň potvrdit příjem dopisu s odhadem, kdy se mu budete moci věnovat.

- Pracujte se Subjectem. Zkuste v něm pár slovy vyjádřit, o co v dopisu jde, aby příjemce věděl, jak je důležitý a čeho se týká. Subject „Dotaz“ je totéž jako dopis bez Subjectu. Pokud se během odpovědí téma dopisu změní, pozměňte Subject.
- Kontrolujte si pečlivě seznam adresátů, zda dopis posíláte všem, komu jste chtěli, a zda ho naopak neposíláte někomu, komu jít nemá.
- Volte pečlivě jazyk a styl dopisu podle povahy obsahu a seznamu adresátů (myslete i na Cc adresáty).
- Při odesílání odpovědi zvažte, zda je nutné přikládat původní dopis (v případě, že vaše odpověď bude „Ano“, tak to nejspíše nutné je) a zda je ho nutné přikládat celý. Zvláště pokud je původní dopis dlouhý a vy reagujete jen na několik vět z něj, zvažte okopírování pouze relevantních vět do vaší odpovědi.
- Respektuje práva jiných účastníků diskuse. Formálně nejste oprávněni přeposlat text nebo obrázek, jehož nejste autorem, třetí osobě bez souhlasu autora. Samozřejmě mezi třemi kamarády takovou věc nebudete řešit, ale pokud jde o oficiálnější komunikaci, je třeba to dodržovat.
- Pokud se chystáte poslat větší soubor více adresátům, zvažte, zda ho opravdu potřebuje drtivá většina z nich. V opačném případě soubor uložte na nějakém vhodném místě (váš web, servery typu Úschovna, DropBox apod.) a pošlete pouze odkaz.
- I když váš dopis neobsahuje velký soubor, ale má hodně adresátů, pečlivě zvažte, zda ho opravdu chcete poslat všem. Samostatnou kapitolou jsou pochopitelně různé řetězové dopisy.
- Posuzujte obsah svého dopisu a tomu přizpůsobte formát. Odstrašující případ je, když někdo vytvoří dokument (např. ve Wordu) obsahující dvě věty a ten přiloží jako soubor k dopisu, místo aby ty dvě věty napsal přímo do dopisu. Současné poštovní aplikace k problému přistupují podobně a i dopisy, u nichž to absolutně není nutné často posílají jako HTML dokument. To může některým příjemcům působit potíže, zejména pokud mailer pošle dopis s MIME dokumentem typu multipart/alternative, což znamená, že dopis obsahuje dvě záměnné varianty, jednou je text/html a druhou je text/plain, který je ale prázdný!
- Podepište se.



Uživatel e-mailu by měl mít na paměti, že obsah dopisu není během přepravy nijak chráněn a díky technickým chybám nebo nějakému útoku se případně může dostat do ruky i lidem, kterým nebyl určen. Nemluvě o možnosti vlastní chyby odesilatele při psaní adresy příjemce. Tomu by odesilatel měl přizpůsobit obsah dopisu. A pokud je obsah dopisu důvěrný, měl by použít některý šifrovací systém.

Stejně tak je jasné, že díky otevřenému protokolu může do obálky i do dopisu kdokoliv napsat cokoliv. Příjemce nemá nikdy jistotu, kdo je skutečným odesilatelem, a rozhodně by neměl nikdy slepě věřit tomu, co je napsáno v dopisu. Jak už jsme si říkali, pokud nejde o nic závažného, příjemce asi dokáže podle stylu textu odhadnout pravost, ale je třeba být obezřetný. Pokud mi kamarád pošle soubor, o kterém jsme se den před tím bavili u oběda, asi budu věřit, že je od něj. Pokud ale pošle samotný soubor, aniž by ho avizoval a aniž bych z textu poznal, že je pravý, soubor rozhodně **nebudu otevírat**, dokud mu například nezavolám.

Poštovní servery přijímající poštu pro koncové uživatele se brání dopisům s nejistým odesilatelem (např. tak, že se pokusí uvedenému odesilateli poslat DSN zprávu), ale taková ochrana není zcela spolehlivá. Některé automatické systémy používají známou metodu challenge/response, spolehlivou ochranu ovšem představuje pouze elektronický podpis.

Snímek 74

Bezpečnost pošty (klient, server)

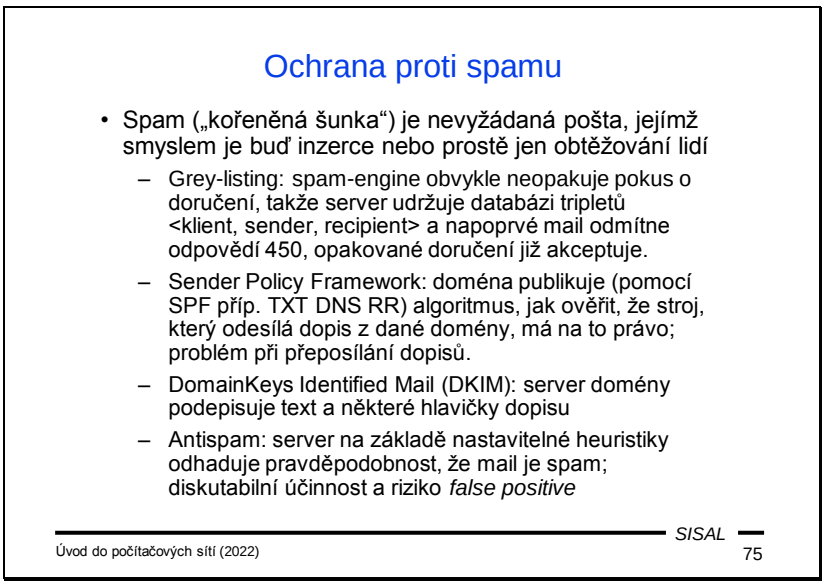
- Běžný server by měl posílat maily lokálních klientů/uživatelů komukoliv, ostatní maily pouze lokálním uživatelům; jinak se jedná o tzv. *open-relay* a hrozí riziko zneužití pro rozesílání hromadných mailů a díky tomu zablokování komunikace od jiných serverů.
- Ze stejného důvodu může při prvotním vložení mailu do systému (*mail submission*) server (někdy označovaný jako MSA) požadovat, aby se klient autentikoval pomocí ESMTP příkazu AUTH (je to součást SASL profilu pro SMTP).
- Klient může pomocí ESMTP příkazu STARTTLS požádat o zahájení SSL/TLS spojení (např. mezi pobočkami firmy, jinak je šifrování spíše problém uživatele).

Správně nakonfigurovaný poštovní server by měl rozlišovat dopisy poslané „jeho“ uživateli, které by měl bez omezení doručovat, a dopisy přicházející „zvenku“, z nichž by měl přijmout pouze takové, které jsou určeny nějakému „jeho“ uživateli. Pokud server dovolí komukoliv, aby se připojil a poslal dopis kamkoliv, říká se mu *open-relay server* a hrozí riziko, že bude zneužíván pro rozesílání hromadných mailů. Spamovací automaty se totiž snaží za sebou „zametat stopy“ tím, že dopis přeposílají přes několik open-relay serverů, takže dohledání skutečného původce je komplikované. A naopak existují organizace, které servery ve světě prověřují, vyhledávají ty, jež jsou open-relay, a sestavují jejich seznam, který pak mohou použít správci poštovních serverů pro blokování příjmu dopisů od kohokoliv z open-relay serverů.

Problém nastává v okamžiku, kdy se nějaký „vlastní“ uživatel serveru potřebuje připojit na svůj poštovní server „zvenku“, jelikož je zrovna mimo lokální síť. Z hlediska serveru je to cizí klient a odeslání dopisu bude odmítnuto. A uživatel nemá jak serveru prokázat svou identitu, neboť SMTP protokol sám o sobě autentikaci nepoužívá. Existuje ale rozšíření, pomocí nějž může server (obvykle na jiném portu, 587) autentikaci vynutit.

Vzhledem k tomu, že přenos dopisu mezi odesilatelem a příjemcem není záležitostí jednoho spojení, ale dopis se po cestě několikrát ukládá, je otázka zabezpečení obsahu dopisu plně záležitostí koncových uživatelů. Proto mezi běžnými MTA není spojení nijak šifrováno. Existují ale výjimky – např. pokud máme dva firemní poštovní servery propojené veřejnou sítí, dává smysl spojení mezi nimi šifrovat, aby všechny firemní dopisy nemusely být šifrované svými odesilateli. Rozšíření ESMTP obsahuje příkaz STARTTLS, kterým klient požádá server, aby zahájili šifrování pomocí mezivrstvy Transport Layer Security.

Snímek 75



Nevyžádaná pošta se v poslední době stala velmi nepříjemnou součástí elektronické poštovní komunikace. Ochrana proti ní je obtížná, protože jakékoliv omezení se částečně dotkne i reálných dopisů poslaných skutečnými uživateli.

Metoda zvaná **grey-listing** využívá vlastností protokolu a zkušenosti, že spamovací automaty nekontrolují úspěšnost doručení (v obrovském množství adres příjemců je jistě řada špatných adres, ale pro celkový efekt je to nepodstatná část, takže nedává smysl neúspěšné odeslání detekovat a opakovat). Server proto na první pokus o doručení dopisu nějakému příjemci, od nějakého odesilatele a z nějaké IP adresy odpoví (na RCPT TO příkaz) odpovědí 450. Pokud to byl řádný dopis, klient se po určité době (typicky 15 min) pokusí o nové doručení, server zjistí, že došlo k opakovanému pokusu, a na nový požadavek již normálně odpoví 250 a dopis doručí. Zároveň si k dané trojici poznamená čas, kdy došlo k poslednímu pokusu o doručení, a po nějakou dobu (např. 5 týdnů) všechny další dopisy propouští bez zdržení, přičemž každý další pokus posouvá tuto časovou známku. Výsledkem je, že pokud si spolu dva partneři vymění alespoň jeden dopis měsíčně, jejich komunikace probíhá bez zdržení, s výjimkou 15minutového zdržení prvního dopisu. Pokud naopak po prvním pokusu nedojde do nějaké doby (např. 1 hod) k opakovanému pokusu, daná trojice se dostane na určitou dobu na „černou listinu“. Právě kombinace myšlenky černé a bílé listiny dala metodě název.

Dalším pokusem o řešení byl tzv. Sender Policy Framework. Základní myšlenka je prostá: doména definuje, jaké servery používá pro odesílání pošty, a od nikoho jiného by nikdo neměl převzít dopis, jehož odesilatelem je někdo z dané domény. Byla vytvořena relativně složitá syntaktická pravidla, jak definovat správné odesílací MTA, a ta se nejprve publikovala pomocí DNS záznamu TXT, později dokonce nově vytvořeného záznamu SPF. Jenže celý princip má háček v tom, že jakmile dopis

dorazí někam, kde má uživatel nastavené forwardování došlé pošty na jiné místo, ověření selže, protože dopis s původním odesilatelem najednou rozesílá úplně jiný stroj! Byly sice vytvořeny opravné mechanizmy, ale celý systém se ukázal jako nepružný, a to dokonce tak, že DNS typ SPF byl zrušen.

Vznikla ale úspěšnější náhrada, tzv. DomainKeys Identified Mail. Základní myšlenka je stejná, rozdíl je v tom, že odesílací stroj dopis **podepíše**. Přesněji řečeno, podepíše text dopisu a vybrané hlavičky. Přijímající server podpis zkontroluje a podle výsledku dopis buďto doručí do cílového mailboxu nebo odmítne. Oproti SPF má toto řešení výhodu v tom, že forwardování podpis nijak neovlivní. Všechny odesílací MTA pro danou doménu je možné vybavit vlastním klíčem, takže je možné odesílání distribuovat na více strojů a seznam flexibilně měnit.

Běžným prostředkem ochrany jsou rovněž různé **antispamy**. Jsou to algoritmy, které se pokoušejí odhadovat pravděpodobnost, že dopis je spam. Používají řadu atributů, kterým lokální správce může dát vlastní váhy. Atributy se týkají jak formy (např. Subject samými velkými písmeny), tak obsahu (např. frekvence výskytu některých slov). Podle výsledného skóre je možné dopis buďto úplně zahodit, uložit do karantény nebo jen označkovat a doručit. Bohužel mají tyto algoritmy diskutabilní účinnost, protože spamovacím automatům se čím dál lepe daří napodobovat skutečné dopisy, a naopak přísným nastavením pravidel se zvyšuje procento neoprávněně špatně klasifikovaných dopisů (tzv. *false positive* výsledek).

Snímek 76

Souhrn 4

- Jaká jsou nejdůležitější bezpečnostní rizika FTP?
- Co je podstatou problémů při otevírání dodatečných datových kanálů? na kazdy prenos se otevira nove spojeni
- Vysvětlete pojmy MTA a MX a jejich roli v přenosu pošty.
- Co je „obálka“ a jak souvisí s obsahem hlaviček dopisu?
- Jak lze používat non-ASCII znaky v hlavičkách a těle dopisu?
- Jak se liší UUENCODE, Base64 a Quoted-Printable?
- V čem spočívá problém open-relay serverů?
- Vysvětlete podstatu grey-listingu a DKIM.