Zabezpečení komunikace, ACL

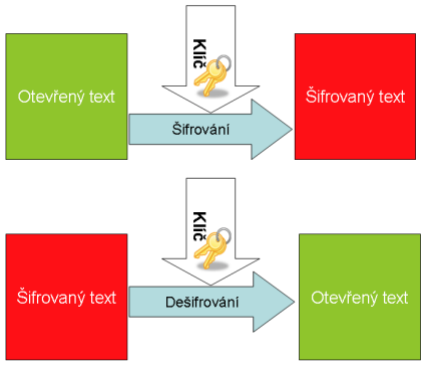
Joseph Soukup – P4

Kryptografie

Taky známo jako šifrování je nauka o metodách utajování smyslu zpráv převodem do podoby, která je čitelná jen se speciální znalostí
Někdy je pojem obecněji používán pro vědu o čemkoli spojeném se šiframi jako alternativa k pojmu kryptologie
Kryptologie zahrnuje kryptografii a kryptoanalýzu neboli luštění zašifrovaných zpráv.
Šifra nebo šifrování se označuje kryptografický algoritmus, který převádí čitelnou zprávu neboli prostý text na její nečitelnou podobu neboli šifrový text.
Opačný postup se nazývá dešifrování
Utajovaným prvkem není algoritmus, ale parametr šifrování (klíč)

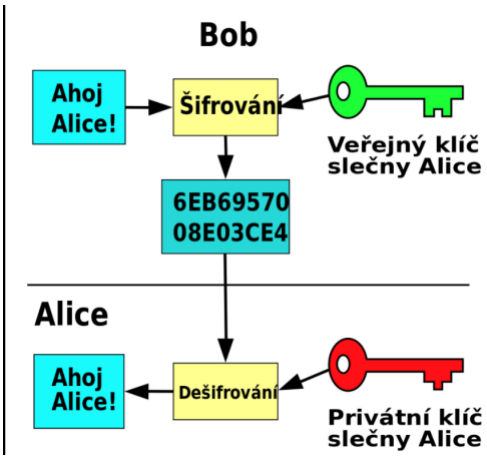
Symetrická

- Nazývá se také konvenční
- Šifrovací algoritmus, který používá k šifrování i dešifrování jediný klíč (liší se od algoritmů s veřejným klíčem, které mají dvojice klíčů – tajný a veřejný)
- Výhodou je nízká výpočetní náročnost (s veřejným klíčem mohou být i 100 000x pomalejší)
- Nevýhodou je nutnost sdílení tajného klíče => odesílatel a příjemce se musí předem dohodnout na tajném klíči
- **Použití**:
    - Často používané společně asymetrickými šifry
    - Otevřený text se zašifruje symetrickou šifrou s náhodně vygenerovaným klíčem
    - Tento symetrický klíč se zašifruje veřejným klíčem asymetrické šifry => dešifrovat data může pouze majitel tajného klíče dané asymetrické šifry
- **Rozdělení**:
    - Symetrické šifry se dělí na dva typy:
        - Proudové šifry zpracovávají otevřený text po jednotlivých bitech
        - Blokové šifry rozdělí otevřený text na bloky stejné velikosti a doplní vhodným způsobem poslední blok na stejnou velikost (většina 64 bitů, AES - 123 bitů)
    - Blokové:
        - AES
        - Blowfish
        - DES
    - Proudové:
        - FISH
        - RC4
- **Generování klíčů**:
    - Pro šifrování relace jsou použity symetrické šifrovací klíče (generování = pseudonáhodné klíčové generátory)
    - Díky nedostatečné přítomnosti náhody při generování => prolomení šifry (nutnost zdroje vysoké entropie = míra neuspořádanosti)

- 

Asymetrická

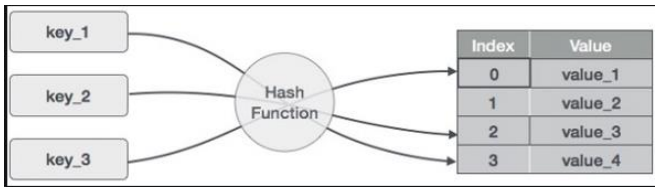
- Pro šifrování a dešifrování používá odlišné klíče
- Důvěrnost (nemožnost čtení nepovolané osobě)
- Autenticitu (identifikace autora zprávy pomocí veřejného klíče = elektronický podpis)
- Nepopiratelnost zprávy (zprávu mohl vytvořit pouze vlastník privátního klíče)
- **Základní principy**:
    - Šifrovací klíč sestává z dvou částí:
        - Jedna část pro šifrování zpráv (příjemce nemusí znát):
        - Druhá pro dešifrování (odesílatel jí zpravidla nezná)
    - Odesílat s příjemcem nemusí sdílet žádné tajemství (žádná výměna klíčů = hlavní výhoda)
    - Veřejný a soukromý klíč:
        - Šifrovací klíč je veřejný, majitel klíče ho volně uveřejní (kdokoliv může šifrovat jemu určené zprávy)
        - Dešifrovací klíč je privátní, majitel jej drží v tajnosti a pomocí něj může tyto zprávy dešifrovat (existují metody, kdy oba jsou soukromý)
        - Šifrovací klíč **e** a dešifrovací klíč **d** = matematicky svázány (nezbytná podmínka je, že se nedají vypočítat, pokud jeden zná druhý)
    - **Šifrování:**
        - C = f(m, e)
    - **Dešifrování:**
        - M = g(c, d)
- **Mechanismy:**
    - Jednocestné funkce = operace, lze provést pouze v jednom směru: ze vstupu jednoduše vypočítat výstup, z výstupu ale velmi obtížně získat vstup
    - Příklad je násobení:
        - 100x5 (mega easy)
        - Rozluštit z jakých násobitelů se skládá 500 (už docela na hovno)
        - Na tomto principu = RSA
    - Další podobné problémy jsou výpočet diskrétního logaritmu, či problém batohu

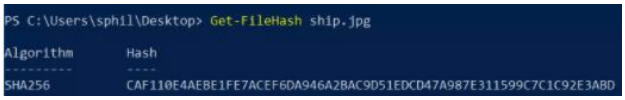
- 

- Příklad:
    - Elektronický podpis
    - Uživatel, který chce poslat zprávu, tak spočítá její digitální podpis a ten pošle spolu se zprávou příjemci
    - Digitální podpis může být spočítán jedině pomocí odesílatelova privátního klíče, ale pro ověření stačí pouze veřejný klíč

Hashování

- Matematická funkce (algoritmus) pro převod vstupních dat do (relativně) malého čísla
- Výstup hashovací funkce se označuje **výtah, miniatura, otisk, fingerprint** či **hash**
- Hashovací funkce se používají k rychlejšímu prohledávání tabulky, porovnání dat, při hledání podobných DNA sekvencí v bioinformace
- V podobě kryptografické hashovací funkce je používána pro vytváření a ověřování elektronického podpisu, zajištění integrity dat, ochranu uložených hesel
- **Vlastnosti:**
    - Jakékoliv množství vstupní dat poskytuje stejně dlouhý výstup (otisk)
    - Malou změnou vstupních dat dosáhneme velké změny na výstupu
    - Z hashe je prakticky nemožné rekonstruovat původní text zprávy
    - V praxi je vysoce nepravděpodobné, že dvěma různými zprávám odpovídá stejný hash = pomocí hashe lze v praxi identifikovat právě jednu zprávu
- **Popis:**
    - Formálně jde o funkci *h,* která převádí vstupní posloupnost bitů (či bytů) na posloupnost pevné délky *n* bitů
    - Z definice plyne existence kolizí, to znamená dvojic vstupních dat (x, y), x != y, takových, že h(x) = h(y).
    - Dvojice různých vstupních dat může mít stejný otisk
    - Kolize jsou nežádoucí, ale v principu jim nelze vyhnout, protože počet možných různých vstupních zpráv je větší než počet možných různých otisků
- **Použití:**
    - Hashovací tabulka
        - Datová struktura **hashovací struktura** používá hashovací funkci na transformaci klíče na index, podle kterého se do tabulky přistupuje

- 

  - Otisk, signatura
    - Kontrolní otisk souboru jako metoda detekce chyb při přenosu nebo ukládání
    - V tomto případě jde o náhodné a neúmyslné chyby a příkladem metody je cyklický redundantní součet (CRC)
- 

- SHA je rodina pěti algoritmů SHA-1, SHA-224, SHA-256, SHA-384, a SHA512
- SHA-1 Vytvoří obraz zprávy dlouhý 160 bitů
- Ostatní vytvoří délku podle čísla
- SHA se používá u různých protokolů a aplikací včetně TLS a SSL

Certifikáty

- Vydávám certifikační autoritou je v asymetrický kryptografii digitálně podepsaný veřejný šifrovací klíč
- Uchovává se ve formátu X.509 (obsahuje informace o majiteli veřejného klíče a vydavateli certifikátu)
- Certifikáty jsou používány pro identifikaci protistrany při vytváření zabezpečeného spojení (HTTPS, VPN)
- Na základě principu přenosu důvěry je možné důvěřovat neznámým certifikátům, které jsou podepsány důvěryhodnou certifikační autoritou
- Třídy certifikátů:
  - Class1 – jednotlivec – email
  - Class2 – organizace – prokázání identity
  - Class3 – určena pro servery a digitální podpisy – nezávislé potvrzení identity certifikační autoritou
  - Class4 – Online obchodní transakce mezi společnostmi
  - Class5 – soukromé subjekty nebo vládní bezpečnost
- Vytvoření certifikátu
  - Vytváří certifikační autorita
  - Nejprve ověří údaje o majiteli předloženého veřejného šifrovacího klíče, doplní identifikační údaje a následně veřejný klíč elektronicky podepíše
- Ověření certifikátu
  - Platný elektronický podpis zaručuje, že s certifikátem nebylo manipulováno
  - Postup ověření:
    - Elektronické ověření pomocí software:
      - Platnost certifikátu
      - Kontrola elektronického podpisu
    - Kontrola prováděná člověkem:
      - Položky certifikátu s identifikací majitele a certifikační autority
      - Postup přenosu důvěry
  - Platnost:
    - Většinou jeden rok (kvůli z neužitelností)
  - Revokační systém:
    - Zneplatnění certifikátu před vypršení platnosti
  - Kvalita:
    - Důvěryhodnost certifikační autority
    - Kvalita použitých kryptografických algoritmů

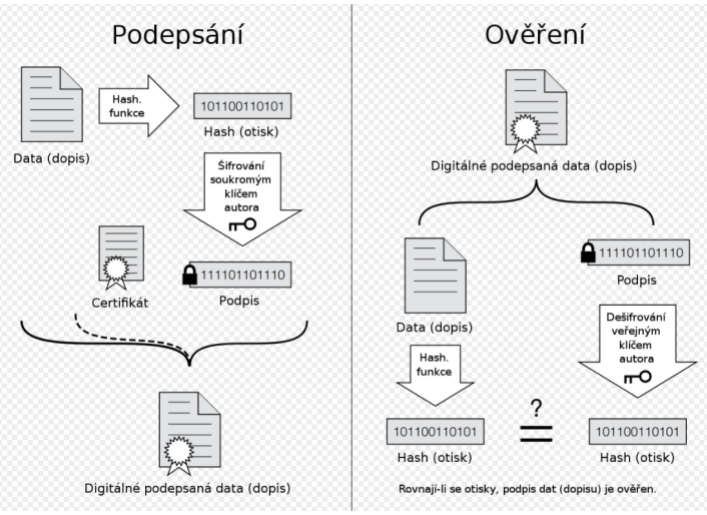
        - Transparentnost softwarového procesu ověření certifikátu
        - Chování uživatele, který certifikát ověřuje
- Self-signed certifikát
    - Při vytvoření páru šifrovacích klíčů je možné veřejný klíč obratem podepsat odpovídajícím privátním klíčem

Certifikační autorita

- Subjekt, který vydává digitální certifikáty (elektronicky podepsané veřejné šifrovací klíče), čímž usnadňuje využívání PKI (Public Key Infrastructure) tak, že svojí autoritou potvrzuje pravdivost údajů, které jsou ve volně dostupném veřejném klíči uvedeny
- Důvěry v certifikační autoritu:
    - Hodnota digitálního certifikátu je úměrná míře důvěry, kterou máme k údajům v něm uvedených
- Přenos důvěry:
    - V počítači jsou šifrovací klíče uloženy v úložišti certifikátů nebo v klíčence.
    - Při ověřování autentičnosti veřejného klíče můžeme využít toho, že klíč je digitálně podepsán důvěryhodnou certifikační autoritou
    - Pokud je digitální podpis certifikátu platný a důvěřujeme certifikační autoritě, která klíč podepsala, přeneseme důvěru a věříme v důvěryhodnost neznámého veřejného klíče.

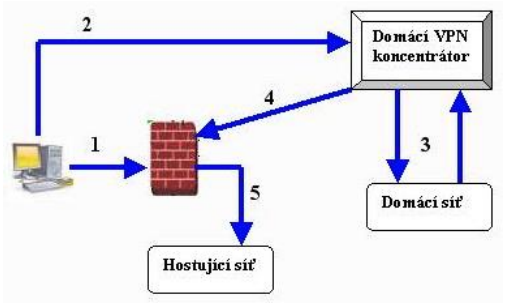
Elektronický podpis

- Označení specifických dat, které v počítači nahrazují klasický vlastnoruční podpis, respektive ověřený podpis
- Připojen k datové zprávě nebo je s ní logicky spojen => ověření totožnosti podepsané osoby ve vztahu k datové zprávě
- V anonymním světě internetu ověřit totožnost
- Vytvořen pro konkrétní data a je možné pomocí počítače ověřit, zda jsou data v té podobě, ve které byla podepsána.



VPN (Virtual Private Network)

- Prostředek k propojení několika počítačů prostřednictvím nedůvěryhodné počítačové sítě (např. veřejný internet)
- Lze tak snadno dosáhnout stavu, kdy spojené počítače budou mezi sebou moci komunikovat, jako kdyby byly propojeny v rámci jediné uzavřené privátní sítě.

- Při navazování spojení je totožnost obou stran ověřována pomocí digitálních certifikátů, dojde k autentizaci, veškerá komunikace je šifrovaná => bezpečné spojení
- Využití:
    - Mezi počítači, co jsou připojeny k internetu => připojení firemních notebooků připojených kdekoliv k Internetu do firemního intranetu.
    - K propojení se ve firemní síti nejprve zprovozní VPN server, zajistí připojení k internetu, ke kterému se pak připojí VPN klienti z jakéhokoliv místa, které je taky k internetu připojeno
    - VPN server plní funkci síťové brány, která zprostředkovává připojení, zajišťuje zabezpečení a šifrování veškeré komunikace
- Použití VPN při autentizaci v síti:
    - Uživatel se připojil na VPN koncentrátor a je tedy tím, za koho se vydává
    - Celou síť chrání firewall (pouze některé VPN brány)
    - Aby se uživatel někam dostal musí mít navázané spojení do domovské sítě
    - 1) uživatel se snaží připojit na síť, ale brání ho firewall
    - 2) Musí navázat spojení na VPN koncentrátor
    - 3) Ověří důvěryhodnost
    - 4) Pokud je uživatel známý, odešle firewallu žádost o povolení komunikace
    - 5) Poté nastane spojení
    - 

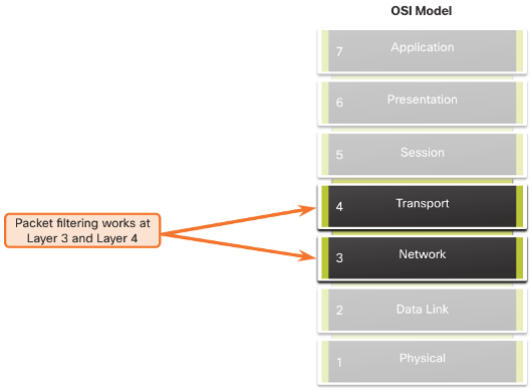
Příklad implementace (SSL) - Secure Sockets Layer

- Je protokol, respektive vrstva vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP), která poskytuje zabezpečení komunikace šifrováním a autentizaci stran
- **Využití**:
    - Bezpečná komunikace mezi webovými servery (HTTPS), což je zabezpečená verze protokolu http
    - Po vytvoření SSL spojení je komunikace mezi serverem a klientem šifrovaná => zabezpečená
    - **Obvyklá využití**:
        - On-line obchody, které přijímají objednávky a údaje o platebních kartách
        - www portály a projekty s administrací pro zabezpečení hesel a dat
        - komunikace s obchodním partnerem
        - zabezpečení přístupu k poště mimo firemní síť
- **Princip:**
    - SSL funguje na principu asymetrické šifry, kdy každá z komunikujících stran má dvojici šifrovaných klíčů (veřejný a soukromý)
    - Veřejný klíč je možno zveřejnit a pokud tímto klíčem kdokoliv zašifruje nějakou zprávu, je zajištěno, že ji bude moci rozšifrovat jen majitel použitého veřejného klíče svým soukromým klíčem
    - SSL spojení (SSL handshake):
        - Klient pošle požadavek na SSL spojení
        - Server pošle klientovi zpět certifikát serveru (obsahuje veřejný klíč serveru)

        - Podle certifikátu si klient server ověří
        - Na základě obdržených informací vygeneruje klient základ šifrovacího klíče, kterým se bude šifrovat komunikace. Ten zašifruje veřejným klíčem serveru a pošle mu ho
        - Server použije svůj soukromý klíč k rozšifrování základu šifrovacího klíče -> z tohoto základu jak klient, tak server vygenerují hlavní šifrovací klíč
        - Klient a server si potvrdí šifrování tímto klíčem
        - Zabezpečené spojení
        - Aplikace od teď dál komunikují přes šifrované spojení (POST se do té doby nepošle)
    - Během první fáze ustanovení bezpečného spojení si klient a server dohodnou kryptografické algoritmy, které budou použity:
        - Pro výměnu klíčů: RSA, Diffie-Hellman
        - Symetrická šifra: RC2, RC4 nebo AES
        - Jednocestné hashovací funkce: MD5 nebo SHA

ACL

- Řada příkazů, které se používají na filtrování packetů na základě informací v hlavičce packetu
- Když je ACL nasazen na interface, tak router vyhodnocuje všechny packety, které interfacem prochází, zda mohou být přeposlány
    - Používá sekvenční seznam povolení (permit) a odmítnutí (deny) = access control entries (ACEs)
- Když packety prochází interfacem nakonfigurovaným s AC, tak router porovnává informace s ACE v sekvenčním pořadí, aby mohl rozhodnout, zda se packet rovná nějakému ACE
    - => Packet filtering
- Několik routerem prováděné úkoly potřebuje ALC:
    - Omezení síťového provozu => zvětšení výkonu sítě
    - Kontrola řízení toku
    - Základní úroveň bezpečnosti
    - Filtrování provozu na základě druhu provozu
    - Povoluje nebo zakazuje hostům přístup
- **Packet filtering**:
    - Kontrolování přístupu sítě analyzováním příchozích a odchozích packetů a buď je přeposílá anebo zahazuje na základě určitých požadavků
    - Vyskytuje se na 3 nebo 4 vrstvě
    - Cisco podporují 2 druhy:
        - Standart ACL
        - Extended ACL



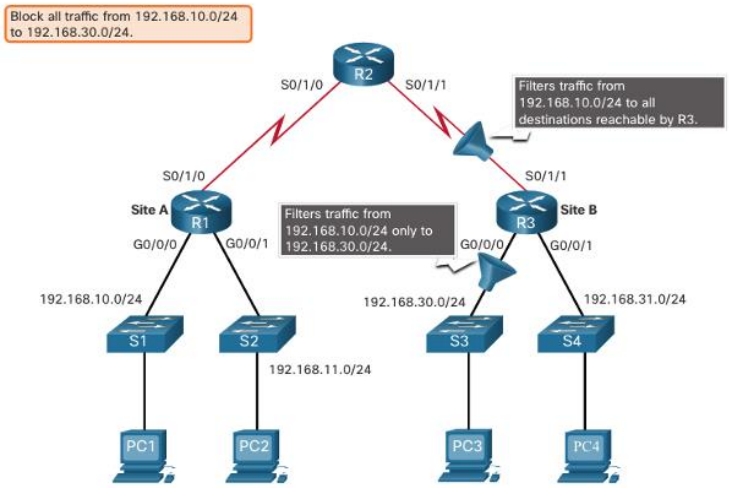
    - ACL se může konfigurovat na příchozí (inbound) nebo odchozí (outbound) traffic
        - Inbound ACL filtruje inbound packety, než jsou poslány na outbound interface

        - Outbound ACL filtruje packety po poslání bez ohledu na inbound interface

    - Proces (inbound standart IPv4 ACL)
        - Vytáhne si z hlavičky IPv4 adresu
        - Začne porovnávat IPv4 adresy s ACE
        - Když nastane shoda, tak router buď zakáže (deny) nebo povolí (permit)
        - Pokud se ničemu nerovná, tak je automaticky zahozen, protože je deny ACE, kterej je automaticky přiřazen všem ACL

Standartní

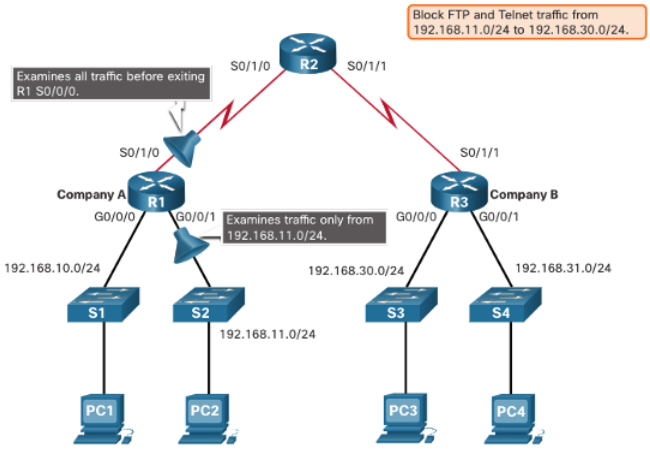
- Povolují nebo zakazují packety na základě zdrojové IPv4 adresy



    - Administrátor chce předejít moc velkého provozu ze sítě 192.168.10.0/24 do sítě 192.168.30.0/24
    - Dva interface řešení:
        - R3 S0/1/1 interface (inbound)
            - Zakázat provoz ze sítě .10
            - Zároveň by filtrovalo provoz do sítě .31 (což nechceme)
        - R3 G0/0 interface (outbound)
            - Neovlivní jiné sítě
            - .10 bude moc stále komunikovat s .31

Rozšířený

- Povolují nebo zakazují packety na základě zdrojové a cílové IPv4 adresy, druhu protokolu, zdrojového a cílového TCP a UDP portu apod.
- Měli by být co nejblíž ke zdroji

- 

  - Zakázat telnet a FTP provoz do 192.168.30.0/24 sítě ze sítě 192.168.11.0/24, přičemž chtějí povolit veškerý jiný provoz
    - Administrátor nekontroluje R3
    - Řešení je nastavit ACL na R1, který specifikuje jak cílovou, tak zdrojovou adresu
    - Dvě možné interface:
      - R1 S0/1/0 interface (outbound)
        - Bude zpracovávat veškeré packety z R1
      - R1 G0/0/1 interface (inbound)
        - Pouze packety z .11 jsou ACL zpracovány na R1
        - Filtr je limitován na odcházející packety z .11 => nejlepší řešení

Jmenný ACL

- Upřednostňovaná metoda při konfiguraci ACL
- Speciální, standartní a rozšířený ACL mohou být pojmenovány na základě jejího účelu
- Např. jméno pro rozšířený ACL FTP-FILTER je lepší než mít ACL se jménem ACL 100
- **Ip access-list** je příkaz na vytvoření jmenného ACL ,:
  - ```
    R1(config)# ip access-list extended FTP-FILTER
    R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp
    R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp-data
    ```

Zabezpečení přístupu k managování switche a routeru

- Enable secret:
  - Heslo na privilegovanou úroveň
  - **enable password** – ne tolik šifrované
  - Další možnost jsou externí AAA servery (jména a hesla jsou uloženy jinde než na switchy) => lepší díky centralizované administraci
  - Service password-encryption (ve startup configu nebudou jako text)
- Systémové bannery:
  - Když se někdo chce dostat na switch, tak dostane námi definované hlášení (banner)
- HTTPS
  - Pokud se rozhodneme používat webové rozhraní => tak používat HTTPS
  - http není šifrované
  - **ip http secure server**
  - limitování zdrojových adres co mohou přistupovat přes HTTPS
    ```

    - ACL které bude povolovat pouze námi definované zdrojové adresy => aplikujeme na HTTPS interface s příkazem **ip http access-class**
- Konzolová bezpečnost:
    - Pokud na switch mají přístup více lidí tak by měla být zabezpečená konzole
    - Měla by se používat stejná autentifikace v konzoli jako ve virtuálním terminálu (vty)
- Zabezpečení vty:
    - Vždycky zabezpečit
    - ACL pro limitování zdrojovým adres (Telnet nebo SSH)
    -
```
Switch1(config)# access-list 10 permit 192.168.1.10
Switch1(config)# access-list 10 permit 192.168.2.10
Switch1(config)# line vty 0 4
Switch1(config-line)# access-class 10 in
```
- SSH místo telnet
    - Telnet není moc šifrovaný
    - SSH je silně šifrováno
- Bezpečný SNMP přístup
    - Abychom předešli neautorizovaným změnám v konfigurace měli bychom vypnou read-write SNMP přístup
    - **Snmp server community** string **RW**
- Nepoužívané switch porty:
    - Shutdown na všechny nepoužívané porty
    - Každý port pomocí **switchport mode access** (odděleno)
    - nebo **switchport host**
    -
```
Switch1(config)# interface fastethernet 0/1
Switch1(config-if)# switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled
```

Vytvoření ACL a nasazení ACL na porty směrovače (filtrace provozu dovnitř a ven – porovnání)

- Musí se nejdřív pořádně naplánovat
- Doporučení:
    - Napsat si vše do textového editoru
    - Přidat do editoru příkazy
    - Přidat si poznámky (remarky)
    - Kopírovat do přístroje
    - Testovat zda funguje
- Číslovanou ACL:
    -
```
Router(config)# access-list access-list-number {deny | permit | remark text} source [source-wildcard]
```

Parameter	Description
access-list-number	Number range is 1 to 99 or 1300 to 1999
deny	Denies access if the condition is matched
permit	Permits access if the condition is matched
remark *text*	(Optional) text entry for documentation purposes
source	Identifies the source network or host address to filter
source-wildcard	(Optional) 32-bit wildcard mask that is applied to the source
log	(Optional) Generates and sends an informational message when the ACE is matched

- Jmennou standartní IPv4 ACL:

```
Router(config)# ip access-list standard access-list-name
```

```
R1(config)# ip access-list standard NO-ACCESS
R1(config-std-nacl)# ?
Standard Access List configuration commands:
  <1-2147483647>  Sequence Number
  default         Set a command to its defaults
  deny            Specify packets to reject
  exit            Exit from access-list configuration mode
  no              Negate a command or set its defaults
  permit          Specify packets to forward
  remark          Access list entry comment
R1(config-std-nacl)#
```

- Na interface:

```
Router(config-if) # ip access-group {access-list-number | access-list-name} {in | out}
```