FreeRadius pro IPLOG

Obsah

1 Úvod 3
1.1 Účel 3
1.2 Rozsah 3
2 Topologie a názvosloví 3
2.1 Podporované autentizační metody 4
2.1.1 EAP-TLS (Transport Layer Security) 4
2.1.2 EAP-TTLS (Tunneled Transport Layer Security) 4
2.1.3 PEAP (Protected EAP) 4
2.2 Podmínky instalace FreeRadius v zařízení IPLOG-GAMA 5
3 Instalace FreeRadius v IPLOG GAMA 5
3.1 Základní příkazy z terminálu 6
3.2 Konfigurační soubory freeradius serveru 7
3.3 Webové rozhraní RADIUS v IPLOG-GAMA 8
4 Certifikáty 8
4.1 Generování certifikátu z OS Windows 11 8
4.1.1 Instalace OpenSSL 9
4.1.2 Příprava souborů pro generování certifikátů 10
4.1.3 Skripty 10
4.1.4 Generování CA 10
4.1.5 Generování certifikátu pro radius server 10
4.1.6 Generování certifikátu pro klienta (supplicanta) 10
4.2 Upload serverových certifikátů do IPLOG-GAMA 11
4.3 Upload klientského certifikátů do zařízení 11
5 Konfigurace switche 12
5.1 SIMULand.v4 13
5.1.1 System (Globální povolení 802.1x) 13
5.1.2 Pae (Port mód) 13
5.1.3 Radius (Konfigurace autorizačního serveru) 13
6 Statusy a logy 14
7 Revize 15
8 Přehled autentizačních metod (FreeRADIUS) 16

1 Úvod

1.1 Účel

Tento dokument slouží k popisu instalace a základní konfigurace balíčku FreeRadius Serveru pro IPLOG.

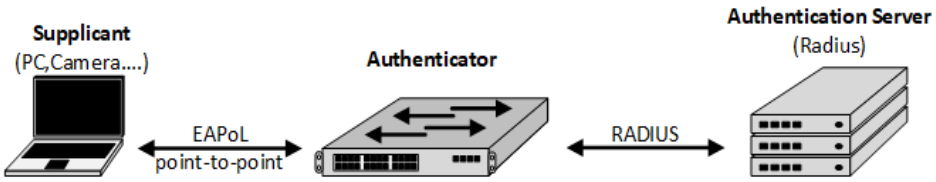
1.2 Rozsah

Tento dokument popisuje:

- Topologie a názvosloví
- Instalace FreeRadius v IPLOG-GAMA
- Základní popis
- Generování certifikátů + upload
- Konfigurace switche
- Ověření funkčnosti

2 Topologie a názvosloví

Standard IEEE 802.1X definuje následující tři požadované komponenty:



- **Supplicant:** Zařízení nebo uživatel na jednom konci point-to-point segmentu LAN, který se pokouší autentizovat prostřednictvím Authenticatoru umístěného na druhém konci spojení.
- **Authenticator:** Zařízení na jednom konci point-to-point segmentu LAN, které zajišťuje ověření Supplicanta připojeného na druhém konci tohoto spojení, a zprostředkovává komunikaci s autentizačním serverem.
- **Authentication Server:** Systém, který poskytuje autentizační služby pro Authenticator. Na základě údajů poskytnutých Supplicantem rozhoduje, zda má Supplicant oprávnění k přístupu ke zdrojům sítě, ve které je Authenticator umístěn.

K ověření Supplicanta lze využít heslo, což představuje méně bezpečný způsob autentizace, nebo *PKI (Public Key Infrastructure)* – infrastrukturu veřejných klíčů, která je založena na asymetrické kryptografii a pracuje s následujícími pojmy:

- **Certifikační autorita (CA – Certificate Authority):** Důvěryhodná entita, která vydává digitální certifikáty a ověřuje identitu subjektů, jako jsou domény, organizace nebo jednotlivci.
- **Klient:** Subjekt (např. uživatel, zařízení, aplikace), který využívá certifikát k autentizaci nebo ke šifrované komunikaci.
- **Radius server:** RADIUS server provádí autentizaci tím, že ověřuje identitu uživatele nebo zařízení, které se pokouší připojit k síti – například pomocí uživatelského jména a hesla, digitálního certifikátu nebo jednorázového hesla.
- **Soukromý klíč (Private Key):** Důvěrná část asymetrického klíčového páru, používaná k digitálnímu podepisování nebo dešifrování dat. Musí být bezpečně uložen a chráněn před neoprávněným přístupem.

Veřejný klíč (Public Key): Volně dostupná část klíčového páru, která slouží k ověření digitálního podpisu nebo k šifrování dat.

2.1 Podporované autentizační metody

Balíček FreeRadius Serveru pro IPLOG podporuje EAP-TLS, EAP-TTLS a PEAP metody v rámci EAP (Extensible Authentication Protocol), které se používají k zajištění bezpečnosti při připojování klientských zařízení do síťových prostředí. Každý z těchto protokolů má svoje specifika, pokud jde o způsob ověřování a zabezpečení komunikace. Zde jsou hlavní rozdíly mezi těmito protokoly:

🕮 Přehled všech podporovaných metod je uveden na konci tohoto dokumentu.

2.1.1 EAP-TLS (Transport Layer Security)

- **Ověření certifikáty:** EAP-TLS používá certifikáty k oboustrannému ověření mezi klientem a serverem. To znamená, že jak klient (zařízení), tak autentizační server musí mít platné certifikáty.
- **Nejvyšší úroveň bezpečnosti:** Tento protokol je považován za nejbezpečnější, protože používá certifikáty a silné šifrování pro ověření jak na straně klienta, tak serveru.
- **Nastavení a správa:** Správa certifikátů může být složitější, protože každé zařízení i server musí mít platný certifikát a správně nakonfigurovanou infrastrukturu PKI (Public Key Infrastructure).
- **Výhoda:** Velmi silná bezpečnostní architektura, která nevyžaduje žádné heslo pro autentizaci, což snižuje riziko útoků jako phishing nebo zachycení hesla.
- **Nevýhoda:** Vyžaduje složitější nastavení a správu certifikátů.

2.1.2 EAP-TTLS (Tunneled Transport Layer Security)

- **Ověření certifikátem serveru:** EAP-TTLS používá certifikát serveru pro ověření serveru, ale na rozdíl od EAP-TLS nevyžaduje, aby klient (zařízení) měl certifikát. Místo toho klient používá tradiční autentizační metody, jako je jméno a heslo nebo jiné formy ověřování.
- **Tunneled Authentication:** Vytváří bezpečný "tunel" mezi klientem a serverem pomocí TLS, což znamená, že uživatelská jména a hesla jsou chráněna před odposlechem. Po navázání bezpečného kanálu se uživatelská jména a hesla mohou přenášet v bezpečí.
- **Výhoda:** Snazší správa, protože klient nemusí mít certifikát, a je to stále dostatečně bezpečné pro většinu aplikací.
- **Nevýhoda:** Menší úroveň bezpečnosti než EAP-TLS, protože uživatelská jména a hesla mohou být teoreticky slabým článkem.

2.1.3 PEAP (Protected EAP)

- **Podobné EAP-TTLS:** PEAP je podobný EAP-TTLS v tom, že používá zabezpečený tunel pomocí certifikátu serveru a umožňuje klientovi používat tradiční metody ověřování, jako je jméno a heslo.
- **Další úroveň šifrování:** PEAP vytváří šifrovaný tunel mezi klientem a serverem, který chrání proces ověření. Stejně jako u EAP-TTLS se autentizace provádí uvnitř tohoto tunelu.

- **Další detaily:** PEAP je často implementován s EAP-MSCHAPv2 (Microsoft Challenge Handshake Authentication Protocol v2), což je metoda, která umožňuje ověření pomocí uživatelského jména a hesla.
- **Výhoda:** Snadná implementace, protože většina klientských zařízení již podporuje PEAP a nevyžaduje správu certifikátů na straně klienta.
- **Nevýhoda:** Závisí na kvalitě a síle použitých metod ověřování (například jméno + heslo), což může být slabší než použití certifikátů.

2.2 Podmínky instalace FreeRadius v zařízení IPLOG-GAMA

Pro instalaci balíčku je potřeba mít IPLOG připojený k internetu nebo mít stažené a nakopírované poslední verze balíčků:

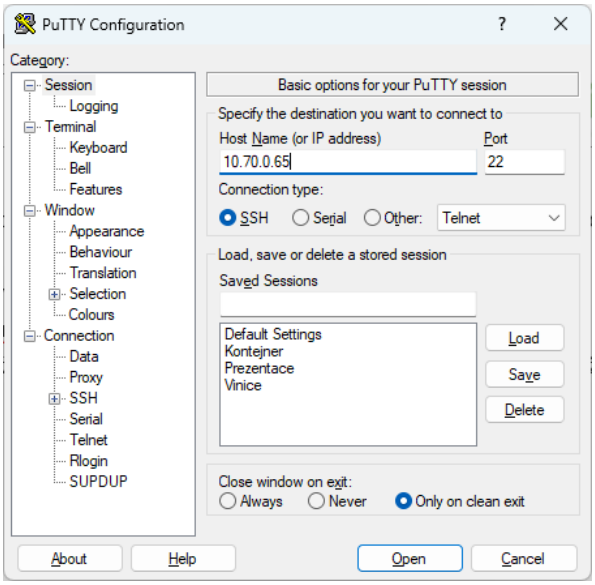
- freeradius-server,
- metel-webui-freeradius-server

a verze FW v IPLOG 12.0.16006 a novější.

🕮 Instalace balíčku v offline režimu: https://wiki.iplog.eu/wiki/Offline_OPKG

3 Instalace FreeRadius v IPLOG GAMA

Spusťte aplikaci ***Putty*** a přihlaste se do IPLOG jednotky jako root uživatel. Výchozí uživatel je: ***„root“,*** bez hesla.



Zadejte příkaz ***„opkg update“***

```
root@iplog:~# opkg update
Downloading http://www.iplog.eu/opkg/base/Packages.gz.
Updated source 'base'.
Downloading http://www.iplog.eu/opkg/firmware/Packages.gz.
Updated source 'firmware'.
```

🕮 Příkazem „***opkg list***“ můžete zobrazit seznam všech dostupných balíčků.
Zadejte příkaz „***opkg install freeradius-server***“, který nainstaluje vše potřebné a spustí balíček pro FreeRadius server. Instalace může trvat až 30 minut. Při instalaci se vytváří certifikáty, které lze použít pro testovací účely.

```
root@iplog:~# opkg install freeradius-server
Installing freeradius-server (2.2.10-16006.087112456) on root.
Downloading http://www.iplog.eu/opkg/base/freeradius-
server_2.2.10-16006.087112456_all.ipk.
Installing metel-webui-freeradius-server (0.0.3-16001.086153748)
on root.
Downloading http://www.iplog.eu/opkg/base/metel-webui-freeradius-
server_0.0.3-16001.086153748_all.ipk.
Configuring metel-webui-freeradius-server.
Configuring freeradius-server.
…
…
…
Starting radiusd: OK
root@iplog:~#
```

Po instalaci se automaticky spustí služba.

🕮 Doporučujeme nainstalovat i ntpd balíček pro synchronizaci času.

```
root@iplog:~# opkg install metel-ntpd-server-internet
Installing metel-ntpd-server-internet (0.0.1-14780) on root.
Downloading http://www.iplog.eu/opkg/base/metel-ntpd-server-
internet_0.0.1-14780_all.ipk.
Installing ntpd (4.2.8p9-12326) on root.
Downloading http://www.iplog.eu/opkg/base/ntpd_4.2.8p9-
12326_all.ipk.
Stopping ntpd: FAIL
Configuring ntpd.
Starting ntpd: OK
Configuring metel-ntpd-server-internet.
Restarting ntpd:
Stopping ntpd: OK
Starting ntpd: OK
```

3.1 Základní příkazy z terminálu

- Pozastavení služby radius serveru: „**/etc/init.d/S49radiusd stop**“

```
root@iplog:~# /etc/init.d/S49radiusd stop
Stopping radiusd: OK
```

- Zapnutí službu radius serveru „**/etc/init.d/S49radiusd start**“

```
root@iplog:~# /etc/init.d/S49radiusd start
Starting radiusd: OK
```

- Restart služby radius serveru „**/etc/init.d/S49radiusd restart**“

```
root@iplog:~# /etc/init.d/S49radiusd restart
Stopping radiusd: OK
Starting radiusd: OK
```

- Zapnutí radius serveru v debug režimu, kdy se vypisují všechny logy přímo do terminálu
  „radiusd -X“

```
Module: Instantiating module "detail" from file /etc/raddb/modules/detail
  detail {
        detailfile = "/var/log/radius/radacct/%{%{Packet-Src-IP-Address}:-%{Packet-Src-IPv6-
Address}}/detail-%Y%m%d"
        header = "%t"
        detailperm = 384
        dirperm = 493
        locking = no
        log_packet_header = no
        escape_filenames = no
  }
 Module: Linked to module rlm_attr_filter
 Module: Instantiating module "attr_filter.accounting_response" from file
/etc/raddb/modules/attr_filter
  attr_filter attr_filter.accounting_response {
        attrsfile = "/etc/raddb/attrs.accounting_response"
        key = "%{User-Name}"
        relaxed = no
  }
reading pairlist file /etc/raddb/attrs.accounting_response
 Module: Checking session {...} for more modules to load
 Module: Linked to module rlm_radutmp
 Module: Instantiating module "radutmp" from file /etc/raddb/modules/radutmp
  radutmp {
        filename = "/var/log/radius/radutmp"
        username = "%{User-Name}"
        case_sensitive = yes
        check_with_nas = yes
        perm = 384
        callerid = yes
  }
 Module: Checking post-proxy {...} for more modules to load
 Module: Checking post-auth {...} for more modules to load
 Module: Instantiating module "attr_filter.access_reject" from file
/etc/raddb/modules/attr_filter
  attr_filter attr_filter.access_reject {
        attrsfile = "/etc/raddb/attrs.access_reject"
        key = "%{User-Name}"
        relaxed = no
  }
reading pairlist file /etc/raddb/attrs.access_reject
} # modules} # server
server inner-tunnel { # from file /etc/raddb/sites-enabled/inner-tunnel
```

3.2 Konfigurační soubory freeradius serveru

Základní konfigurace freeradius serveru se skládá z několika souborů, které je možné editovat a tím konfigurovat vlastnosti aplikace freeradius server. Soubory jsou uloženy v „***etc/raddb/***“.
radius.conf - Soubor obsahuje konfiguraci autentizačního serveru. Když se server spustí, přečte tento soubor a uloží jej do mezipaměti. Data se analyzují za účelem nastavení hodnot proměnných nebo k určení další konfigurace, například modulů.
client.conf - Soubor obsahuje definice klientů (authenticator) RADIUS serveru. Obvykle se jedná o NAS, přístupové body, switche, uživatele atd. Každý klient má „***short name***“, kterým se odlišuje od ostatních klientů. Sdílený „**secret“** slouží k šifrování a podepisování paketů mezi klientem a FreeRADIUS. Script při instalaci freeradius serveru automaticky přidá klienta pro testovací účely.

```
### METEL defaults
client 0.0.0.0/0 {
        secret          = metel
        shortname       = metel_radius_client
}
```

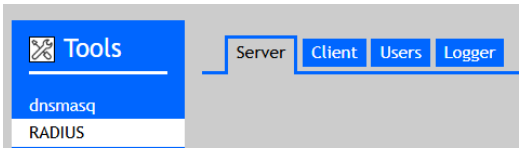
eap.conf - Soubor ve FreeRADIUS serveru slouží ke konfiguraci **EAP (Extensible Authentication Protocol)** – tedy protokolu, který se často používá pro autentizaci uživatelů v sítích. Nastavení typu EAP – např.: EAP-TLS (na základě klientských certifikátů), EAP-TTLS, PEAP (často kombinovaný s MSCHAPv2), a další. Cesta k certifikátům a klíčům – pro EAP-TLS nebo jiné typy využívající šifrování, musí se specifikovat: CA certifikát, serverový certifikát, privátní klíč. Parametry TLS spojení – např.: podporované šifry, požadavek na ověření klienta, OCSP/CRL nastavení.
users - Soubor users ve FreeRADIUS serveru slouží k **lokálnímu uchovávání uživatelských účtů a jejich autentizačních/autorizačních parametrů**. Je to v podstatě jednoduchý způsob, jak definovat uživatele přímo v konfiguračním souboru, bez potřeby databáze nebo externí autentizační služby (např. LDAP). Suppplicant se ověřuje pouze na základě jména hesla. Po instalaci je automaticky vytvořen uživatel “metel” s heslem “metel”.

```
### METEL defaults
metel          Cleartext-Password := "metel"
               Reply-Message = "Hello, %{User-Name}"
```

Aplikace obsahuje více možností a hlubší konfigurace viz. Textový popis u každého souboru. Více informací je možné také dohledat na https://freeradius.org/documentation/

3.3 Webové rozhraní RADIUS v IPLOG-GAMA

Po instalaci balíčku FreeRadius IPLOG automaticky přidá do webového rozhraní v menu **Tools** položku **RADIUS**. Toto je pouze minimální základ konfigurace RADIUS serveru nutný ke spuštění.



Server – Upload CA a serverového certifikátu pro freeReadis server.
Client – Sdílený ***„secret“*** slouží k šifrování a podepisování paketů mezi klientem (switchem) a FreeRADIUS (IPLOG). Toto heslo lze změnit pouze pro defaultně vytvořeného klienta „client 0.0.0.0/0“ viz. ***/etc/raddb/client.conf.***
Users – Vytváření, mazání uživatelů (supplicantů), kteří se autentizují pomocí jména a hesla.
Logger – Výpis logu od FreeRadius serveru.

4 Certifikáty

Po instalaci FreeRadius se vytvoří výchozí certifikáty, které jsou umístěny ve složce ***/etc/raddb/certs*.** Tyto certifikáty lze použít pro testovací účely. Podrobnější popis je v souboru ***/etc/raddb/certs/README*.**

4.1 Generování certifikátu z OS Windows 11

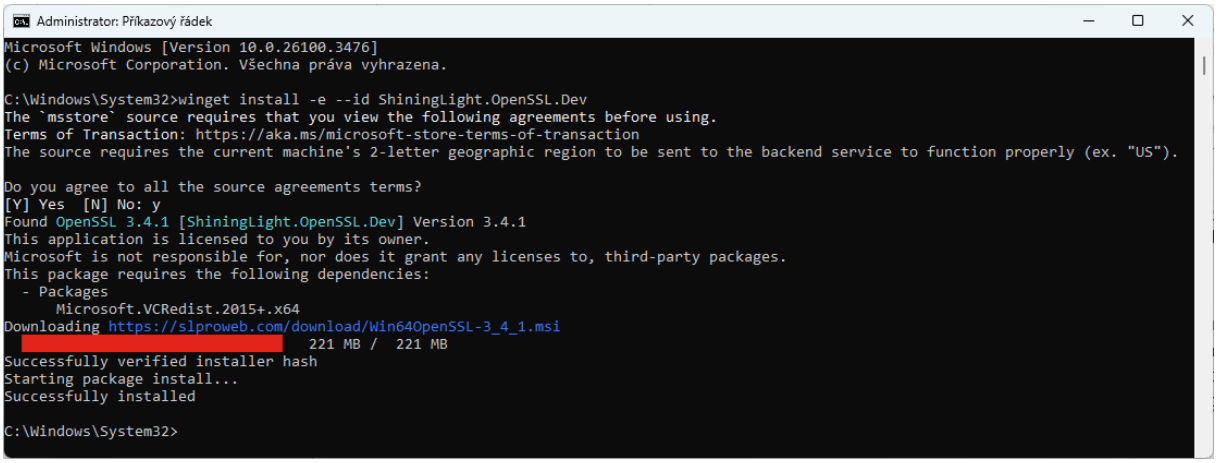
Ze stránek www.mete.eu si stáhnětě testovací generátor certifikátu.
Stručný návod slouží výhradně k vývojovému nebo testovacímu generování certifikátů pomocí nástroje ***OpenSSL*** v prostředí Windows. Vygenerované certifikáty obsahují nejmenší možnou sadu nastavení, která dovoluje jejich úspěšné nasazení pro systém RADIUS -> authenticator/klient -> supplicant. Pro ostré nasazení certifikátů je

doporučeno prozkoumat konfigurační soubory ve složce ***/etc/raddb/certs/*.cnf (ca.cnf, server.cnf, client.cnf)*** ve vašem IPLOG zařízení.

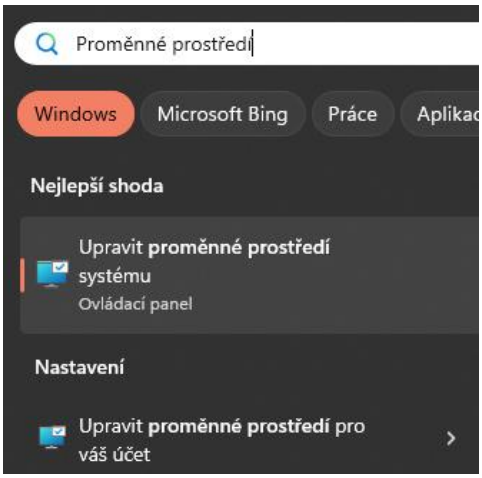
4.1.1 Instalace OpenSSL

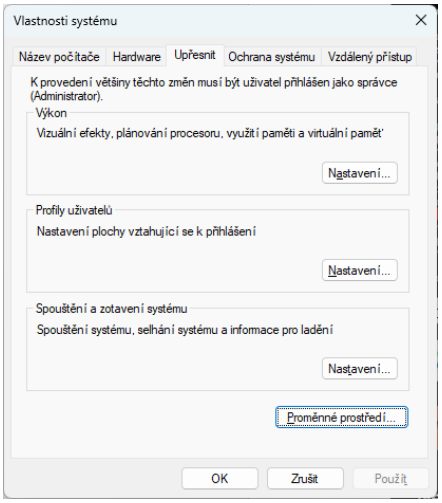
Instalace pomocí winget (doporučeno)

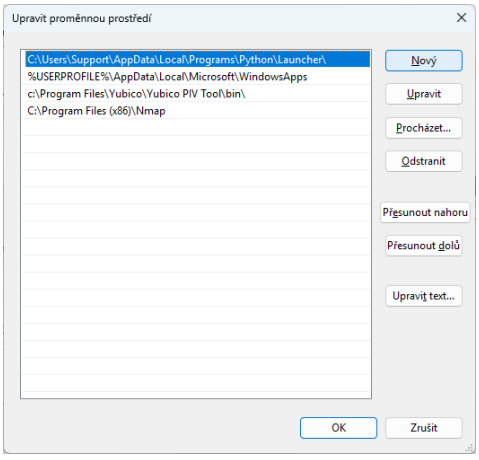
1. Spusťte příkazový řádek jako správce a zadejte příkaz ***„winget install -e --id ShiningLight.OpenSSL.Dev“***

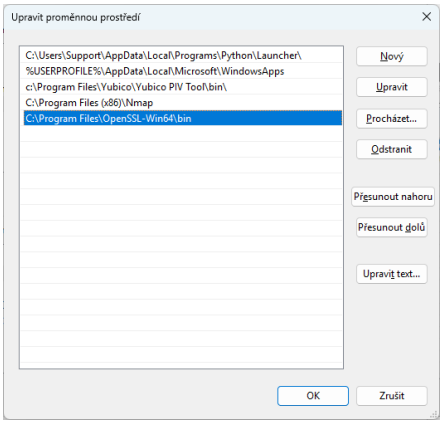


2. Přidejte složku ***"C:\Program Files\OpenSSL-Win64\bin"*** do PATH (proměnné prostředí).









Ruční instalace
1. Otevřete stránku: ***https://slproweb.com/products/Win32OpenSSL.html***
2. Stáhněte a nainstalujte **"Win64 OpenSSL vX.X.X"**
3. Během instalace povolte přidání OpenSSL do systémové proměnné PATH (nebo ji přidejte ručně)

4.1.2 Příprava souborů pro generování certifikátů

Ve stejné složce mějte všechny 3 BAT skripty potřebné ke generování certifikátů (init_ca.bat, generate_server.bat, generate_client.bat). Dále musí být ve stejné složce 4 soubory pro správné rozšířené parametry certifikátu ***(xpca.ext, xpserver.ext, xpclient.ext, openssl.cnf).***

4.1.3 Skripty

Skripty spusťte přímo ve složce tomto pořadí:
init_ca.bat
generate_server.bat
generate_client.bat
OpenSSL se při spuštění těchto skriptů dotáže pomocí interaktivních promptů na potřebné údaje k vystavení certifikátu (CN, země, organizace, společnost atd.).

🕮 ***Common Name*** – doporučujeme zadávat podle názvu/IP/mac… adresy zařízení, tak aby z certifikátu bylo zřetelné, pro jakého klienta byl certifikát vygenerován.

4.1.4 Generování CA

Pro vygenerování certifikační autority spusťte ***"init_ca.bat".***
ca.key - privátní klíč certifikační autority
ca.crt - certifikát certifikační autority (pro RADIUS server i klientské zařízení)
ca.srl - soubor se sériovým číslem (vygenerovaný automaticky)
ca.csr - žádost o certifikát (volitelné, lze smazat)

🕮 Pokud při generování CA zvolíte, že chcete použít soukromé heslo klíče, tak při generování dalších certifikátu (server, klient) budete vyzvání k zadání tohoto hesla.

4.1.5 Generování certifikátu pro radius server

Spusťte "**generate_server.bat**". V pracovní složce by se měly objevit tyto soubory.
server.key - privátní klíč serveru
server.pem - serverový certifikát podepsaný CA (pro RADIUS server)
server.csr - žádost o certifikát (volitelné, lze smazat)

🕮 Pokud při generování certifikátu zvolíte, že chcete použít soukromé heslo klíče, tak při importu certifikátu do zařízení musíte toto heslo zadat.

4.1.6 Generování certifikátu pro klienta (supplicanta)

Spusťte ***"generate_client.bat".*** V pracovní složce by se měly objevit tyto soubory.
client.key - privátní klíč klienta
client.crt - klientský certifikát podepsaný CA (pro klientské zařízení)
client.csr - žádost o certifikát (volitelné, lze smazat)

🕮 Pokud při generování zvolíte, že chcete použít soukromé heslo klíče, tak při importu certifikátu do zařízení musíte toto heslo zadat.

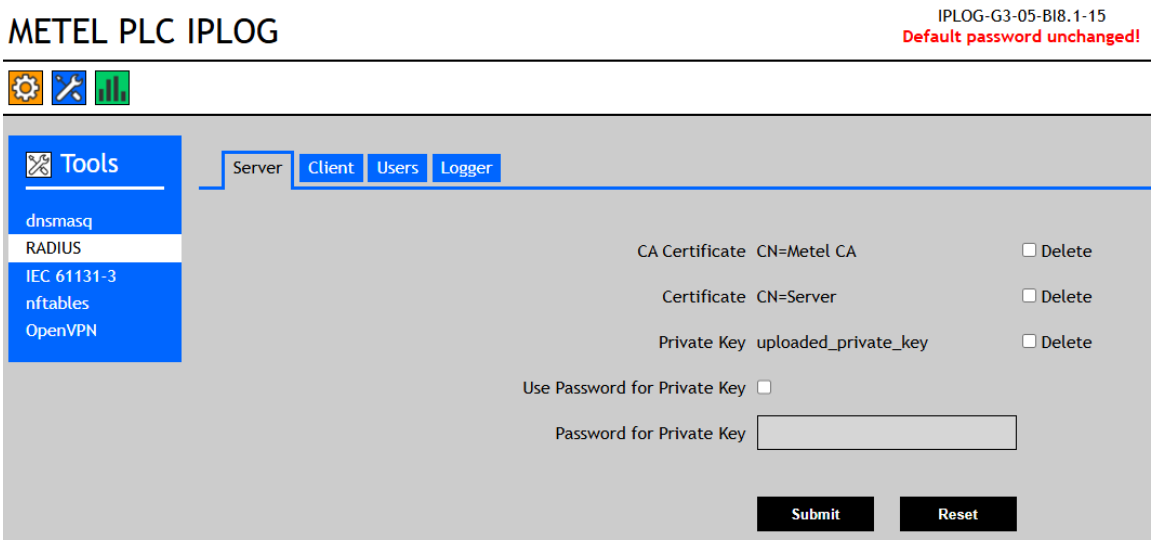
🕮 Vygenerované certifikáty je nyní nutné nahrát do Radius serveru a supplicanta. Pro správné fungování režimu **EAP-TLS** se ujistěte, že RADIUS server i zařízení, které se připojuje (supplicant), mají správně nastavený a vzájemně synchronizovaný čas. V případě potřeby lze certifikáty převést do jiných formátů pomocí OpenSSL.

🕮 Pokud potřebujete vygenerovat další certifikát pro klienta, uložte si stávající certifikát a spusťte znova ***generate_client.bat.***

4.2 Upload serverových certifikátů do IPLOG-GAMA

Otevřete webové rozhraní, zadejte IP adresu IPLOGu a přihlaste se jako root uživatel.
V záložce Tools/RADIUS/Server nahrajte následující soubory:
CA Certificate: ca.pem
Certificate: server.pem
Private Key: server.key

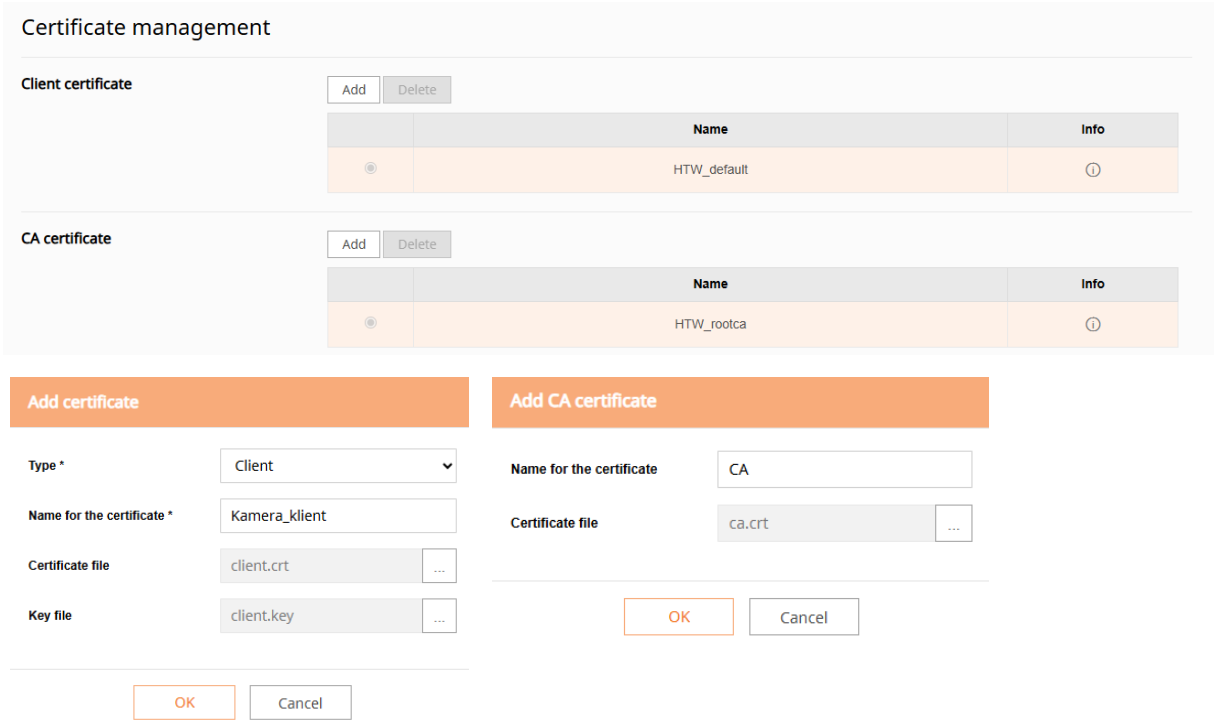
🕮 Pokud je soukromý klíč (private key) chráněný heslem, povolte možnost ***"Use Password for Private Key"*** a zadejte heslo.



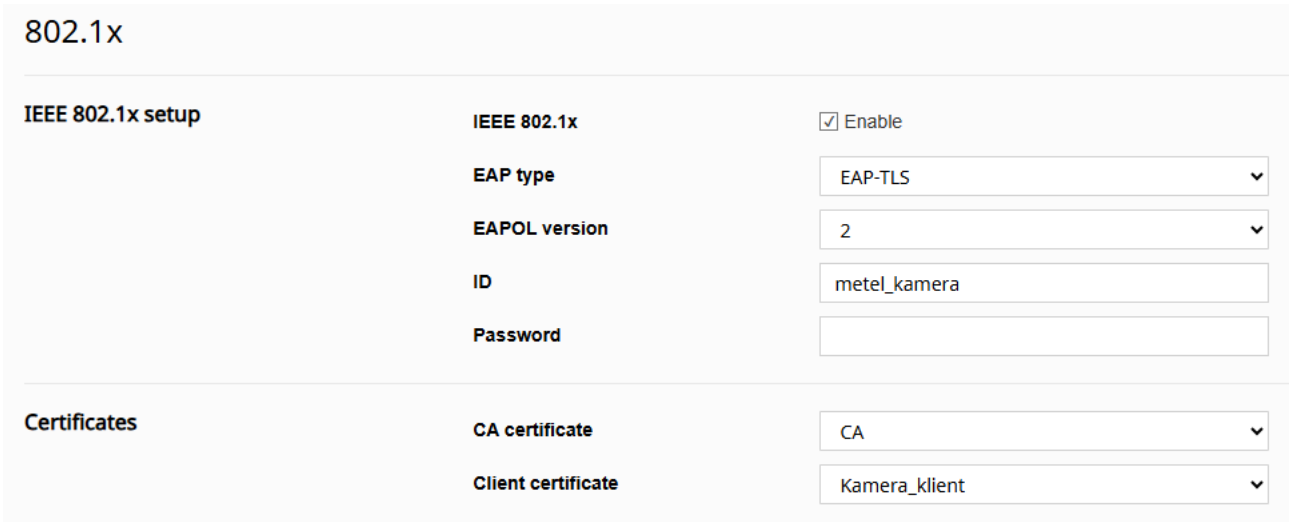
4.3 Upload klientského certifikátů do zařízení

Otevřete webové rozhraní klienta a nahrajte požadované soubory s certifikátem. Postup se může lišit podle konkrétního výrobce zařízení – každý může mít odlišný způsob nahrávání certifikátu a podporovat různé formáty (např. .pfx, .pem, .cer apod.).
CA Certificate: ca.pem
Certificate: client.pem
Private Key: client.key

Příklad upload certifikátů do kamery



Ukázka nastavení a zapnutí 802.1X na kameře



🕮 V případě, že by byl soukromý klíč (private key) chráněný heslem, zadejte toto heslo do pole ***"Password".***

5 Konfigurace switche

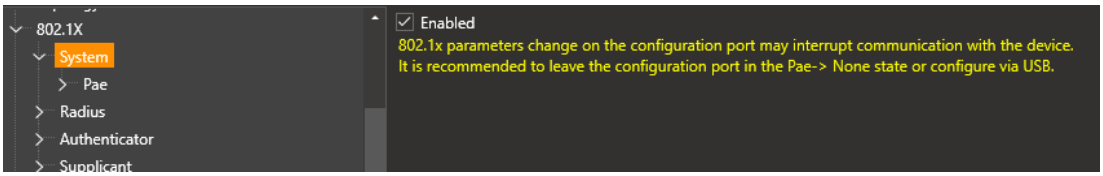
Tato sekce popisuje základní konfiguraci IEEE 802.1x ve switchi.

5.1 SIMULand.v4

- ***System*** – Globální povolení 802.1x autorizace. Port mód (None, Authenticator, Supplicant).
- ***Radius*** – Konfigurace navázání spojení s autorizačním serverem.
- ***Authenticator*** – Konfigurace chování switche jako Authenticator.
- ***Supplicant*** – Konfigurace chování switche jako Supplicant.
- Pro náš případ bude switch v režimu Authenticator.

5.1.1 System (Globální povolení 802.1x)

Checkbox ***"Enabled"*** v sekci ***802.1x -> System*** globálně povoluje nebo zakazuje 802.1x pro všechny porty.
Zaškrtnuto - Aktivují se všechny konfigurační parametry nabídky 802.1x (System, Radius, Pea, Authenticator, Supplicant) a nastavené parametry budou použity k autentizaci připojených supplicantů (klientů) do LAN.
Nezaškrtnuto - 802.1x je zakázáno na všech portech.



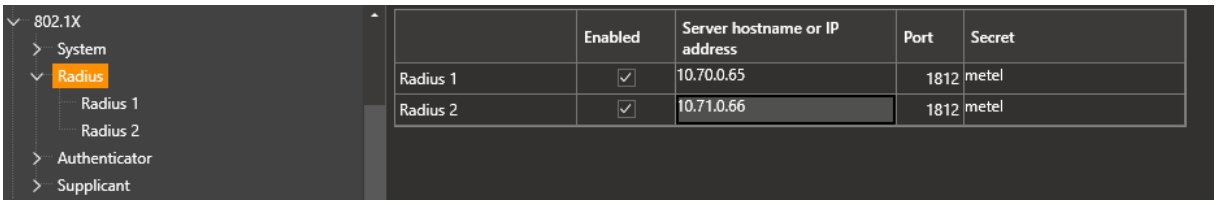
5.1.2 Pae (Port mód)

Pro náš případ na všech portech, kde bude koncové zařízení ověřováno přes Radius server zvolíme režim Authenticator.

5.1.3 Radius (Konfigurace autorizačního serveru)

Autorizační (RADIUS) server ověřuje a kontroluje identitu připojených supplicantů a upozorňuje switch, zda může být supplicant oprávněn pro přístup k síti LAN nebo službám switche. V případě, že je port nakonfigurován jako **Authenticator**, pro úspěšné ověření supplicanta vyžaduje:
Enabled - Zaškrtnuto - Povolení použití nastavených parametrů serveru.
Nezaškrtnuto - Ignoruje nastavené parametry serveru.
Port - Specifikuje UDP/TCP port k autorizaci (1812 je default).
Secret - Určuje autorizační a šifrovací klíč používaný mezi switchem a službou RADIUS spuštěnou na autorizačním serveru Radius. Switch (authenticator) i server musí být nakonfigurovány tak, aby používaly stejné sdílené heslo.

Server Hostname or IP Address – IP adresa nebo název serveru.

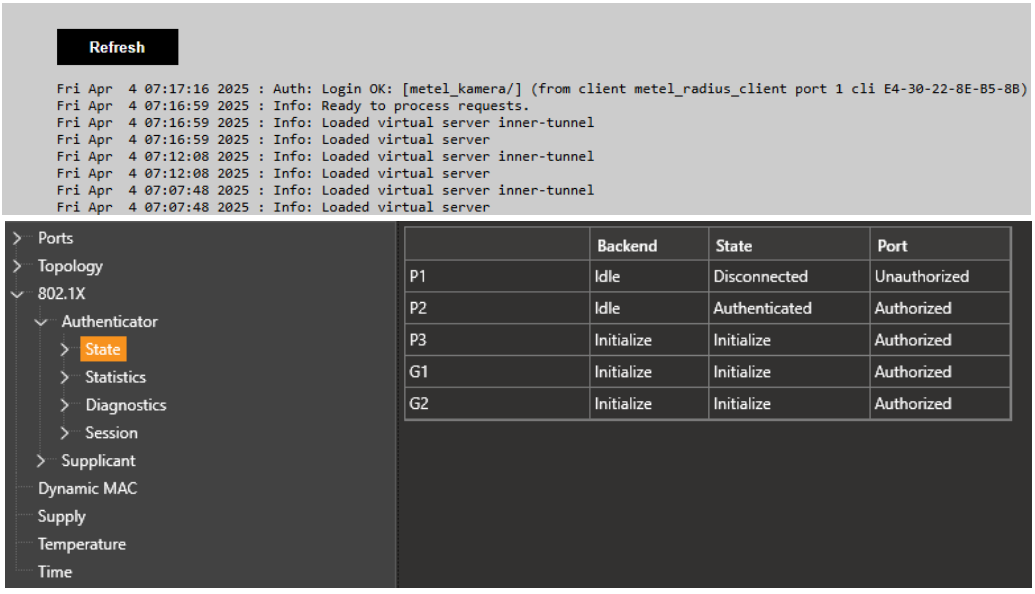


Radius server sekvence
Konfigurace Radius serveru z prvního řádku (Radius 1) se uplatňuje jako výchozí pro komunikaci s Radius serverem (hlavní server). Pokaždé, když se spustí nový proces autentizace, switch se pokouší zaslat autorizační data právě na tento server. Pokud první server třikrát v řadě neodpoví (prodleva závisí na konfiguraci Auhentificator a Server Timeout), switch se pokusí navázat komunikaci s druhým serverem (Radius 2). Druhý server je obvykle použit jako záložní a doporučuje se jeho lokální připojení v síti LAN.
🕮 Detailní informace ke konfiguraci 802.1X ve switchi jsou k dispozici v aplikační poznámce 802.1X.
🕮 Pokud jsou v systému použity VLAN, tak port, kde je připojen Radius server **musí** být členem managementové VLAN switche. Koncová zařízení mohou být členem jiné VLAN, než je management switche.

6 Statusy a logy

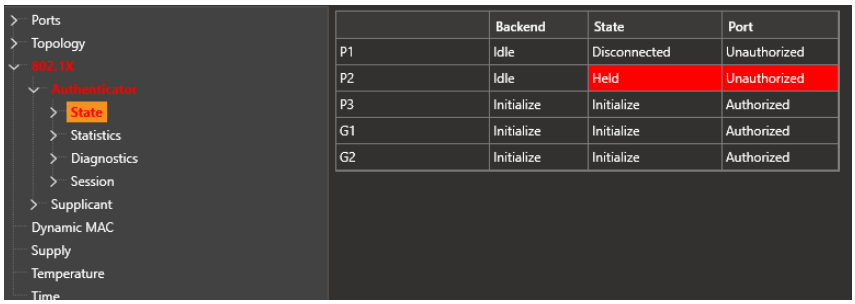
Pří úspěšném ověření bude v logu radius serveru zobrazeno Login OK.
🕮 Pro podrobný výpis logu, je potřeba zapnout Radius server v debug režimu viz. #Základní příkazy z terminálu, logy se poté vypisují pouze v konzoli.

Výpis logů ve webovém rozhranní



Ve statusu switche je vidět, že port je autorizovaný.

```
Fri Apr  4 07:22:27 2025 : Auth: Login incorrect (unable to get local issuer certificate): [metel_kamera/] (from client metel_radius_client
port 1 cli E4-30-22-8E-B5-8B)
Fri Apr  4 07:22:27 2025 : Error: SSL: SSL_read failed in a system call (-1), TLS session fails.
Fri Apr  4 07:22:27 2025 : Error: rlm_eap: SSL error error:14089086:SSL routines:ssl3_get_client_certificate:certificate verify failed
Fri Apr  4 07:22:27 2025 : Error:     TLS_accept: error in error
Fri Apr  4 07:22:27 2025 : Error: TLS Alert write:fatal:unknown CA
Fri Apr  4 07:22:27 2025 : Error: --> verify error:num=20:unable to get local issuer certificate
Fri Apr  4 07:21:22 2025 : Auth: Login incorrect (unable to get local issuer certificate): [metel_kamera/] (from client metel_radius_client
port 1 cli E4-30-22-8E-B5-8B)
```



Ukázka neúspěšné autentizace (port P2 je zablokován)

7 Revize

1.0 – Výchozí revize
1.1 – Doplněna poznámka o VLAN v 5.1.3
1.2 – Balíček freeradius server obsahuje “metel-config-freeradius-server”

8 Přehled autentizačních metod (FreeRADIUS)

Metoda	Šifrování přihlašovacích údajů	Potřeba certifikátu	Zabezpečení	Kompatibilita s AD	Použití	Poznámky
PAP	❌ Prostý text	❌ Ne	❌ Nízké	✅ Ano	VPN, TTLS	Jednoduché, ale nezabezpečené
CHAP	✅ Hash challenge	❌ Ne	⚠️ Střední	❌ Ne	Zastaralé systémy	Lepší než PAP, ale slabé
MS-CHAP	✅ Hash challenge	❌ Ne	⚠️ Nízké	⚠️ Omezeně	VPN, AD	Zastaralý, slabý hash
MS-CHAPv2	✅ Vylepšený hash challenge	❌ Ne	⚠️ Střední	✅ Ano	VPN, Wi-Fi (PEAP)	Často používaný s PEAP
EAP-MSCHAPv2	✅ Tunelované + hash	✅ Server	✅ Dobré	✅ Ano	Wi-Fi, VPN	Běžné s PEAP, podporuje AD
EAP-MD5	✅ Hash	❌ Ne	❌ Nízké	❌ Ne	Testovací	Nevhodné pro produkci
EAP-GTC	✅ Tunelovaný text	✅ Server (PEAP/TTLS)	⚠️ Střední	⚠️ Omezeně	Wi-Fi (TTLS/PEAP)	Nepodporuje MS-AD
EAP-GPSK	✅ Symetrické klíče	❌ Ne	✅ Dobré	❌ Ne	IoT, specifické	Málo podporováno
EAP-TLS	✅ Certifikáty	✅ Klient + Server	✅ ✅ Vysoké	⚠️ Ne přímo	Wi-Fi Enterprise	Nejbezpečnější, ale náročné na správu certifikátů
bare EAP-TLS	✅ Certifikáty	✅ Klient + Server	✅ ✅ Vysoké	⚠️ Ne přímo	Vývoj, testování	„Čistý“ EAP-TLS bez wrapperu
EAP-TTLS	✅ TLS tunel + libovolná metoda uvnitř	✅ Server	✅ Dobré	✅ Ano (s MSCHAPv2)	Wi-Fi	Flexibilní, cert jen na serveru
EAP-SIM	✅ SIM autentizace (UMTS/GSM)	❌ Ne	✅ Dobré	❌ Ne	Mobilní sítě	Používá se hlavně v mobilních operátorech