

# Konfigurácia procesu D2Connector pre SmartWeb

V tejto kapitole je na príkladoch vysvetlené, ako nakonfigurovať spojenie na strane D2000 aplikácie pre potreby SmartWeb platformy. Na komunikáciu medzi SmartWeb aplikáciou a D2000 sa využíva JAPI knižnica, ktorá komunikuje s procesom D2Connector na strane D2000. Predpokladom tejto kapitoly je aktívna D2000 aplikácia (príjajmešom *kernel*), ku ktorej sa môžeme pripojiť. Spôsoby pripojenia sú rozdelené do troch kategórií, pričom možnosti z jednotlivých kategórií možno ubovone kombinovať.

Pripojenie je možné nadviazať dvomi základnými spôsobmi:

- Spojenie aktívne nadväzuje SmartWeb server (základný spôsob pripojenia). Je to štandardný postup, jednoduchší na použitie a ladenie a je aplikovateľný všade tam, kde to bezpečnostná politika v miestnej sieti dovoľuje.
- Spojenie aktívne nadväzuje *D2Connector* (reverzné pripojenie). Tento postup sa používa v prípade, že sa SmartWeb server nachádza v tzv. „demilitarizovanej zóne“ (DMZ) – segmente počítačovej siete, do ktorej je možné nadviazať spojenie aj z lokálneho intranetu aj z vonkajšieho internetu, ale z DMZ von nie je možné nadviazať žiadne spojenie. (Podporované od verzie 10.1.39)

Z hľadiska zabezpečenia pred odpoúvaním je možné nadviazať:

- Nezabezpečené spojenie. Hoci je JAPI protokol binárny, všetky texty sa prenášajú v itatenej podobe. Nezabezpečené spojenie je vhodné počas ladenia alebo v prípade, že komunikácia medzi *D2Connector*-om a *JConnector*-om prebieha v bezpečnej sieti.
- Spojenie zabezpečené protokolom TLS v1.2. *JAPI* a *D2Connector* komunikujú šifrovaným protokolom, pričom *D2Connector* preukazuje svoju identitu certifikátom a privátnym kúom, ktorý *JConnector* porovná s certifikátom, ktorý vlastní on. (Podporované od verzie 10.1.39)

Z hľadiska pripojenia k „hot“ *kernel*-u v redundantnej skupine:

- *D2Connector* sa vždy pripája k „hot“ serveru.
- *D2Connector* je stále pripojený k tomu istému *Kernel* u bez ohľadu na to, či je „hot“ alebo „stand-by server“.

## Parametre pre spustenie *D2Connector*-a

*D2Connector* je proces systému D2000 a je distribuovaný ako konzolová aplikácia (*d2connector.exe*). Akceptuje štandardné parametre procesov D2000 pre spustenie z príkazového riadku, ktoré sú popísané v Online referennej príručke systému D2000. Okrem toho akceptuje nasledovné parametre príkazového riadku:

- `--CONNECTOR_LISTEN_PORT=<port>` - nastaví číslo TCP portu, na ktorom *D2Connector* počúva na prichádzajúce spojenie od JAPI. Ak nie je uvedené inak, počúva na porte 3120. (Parameter je ignorovaný, ak sa použije v kombinácii s `--DCC`)
- `--DCC=<hostname:port>` - prepne *D2Connector* z režimu počúvania do režimu aktívneho pripájania sa na uvedenú adresu (DNS alebo IP) a port. Pokým sa mu nepodarí nadviazať spojenie, pokúša sa o to každých 30 sekúnd. Po ukončení spojenia sa opäť zane pokúša o nadviazanie spojenia.
- `--CONNECTOR_TLS_CERT=<path.crt>` - zapne TLS zabezpečenie a nastaví cestu k súboru s certifikátom vo formáte `.crt`.
- `--CONNECTOR_TLS_PK=<path.pem>` - zapne TLS zabezpečenie a nastaví cestu k súboru s privátnym kúom k certifikátu vo formáte `.pem`. Obidva TLS parametre je potrebné použiť spolu.

*D2Connector* nadväzuje spojenie vždy len jedným spôsobom z ôsmich možných kombinácií. Tzn. bu sa aktívne pripája, alebo počúva, ale nie obidvoje naraz. Rovnako komunikuje bu nezabezpečeným alebo zabezpečeným spôsobom, ale nikdy neumožňuje obidva spôsoby súčasne. Bu je pripojený stále k jednému *Kernel*-u alebo sa prepína na aktuálny „hot“. V prípade, že sa ku D2000 aplikácii pripája viac rôznych klientských aplikácií, ktoré vyžadujú rôzne spôsoby pripojenia, je potrebné naštartovať pre každý spôsob samostatnú inštanciu *D2Connectora*.

## Základný spôsob pripojenia

Ide o nezabezpečené spojenie, ktoré iniciuje JAPI.

*D2Connector* môžeme naštartovať bez parametrov a bude počúvať na pripojenie na porte 3120

```
> d2connector.exe
```

alebo zmení počúvajúci port napríklad na 3121:

```
> d2connector.exe --CONNECTOR_LISTEN_PORT=3121
```

## Nadviazanie reverzného spojenia

Ide o nezabezpečené spojenie medzi *D2Connector*-om a SmartWeb aplikáciou nachádzajúcou sa v DMZ, z ktorej nedokáže iniciovať TCP spojenie. Môže však počúvať na prichádzajúce TCP spojenie, ktoré bude iniciovať *D2Connector*.

Klientská aplikácia je umiestnená na počítači [portal.dmz.customer.com](http://portal.dmz.customer.com) a počúva na porte 3125 na všetkých svojich sieťových rozhraniach. *D2Connector* spustíme v režime pripájania sa:

```
> d2connector.exe --DCC=portal.dmz.customer.com:3125
```

## Nadviazanie zabezpečeného spojenia

Ide o spojenie medzi *D2Connector*-om a *JConnector*-om zabezpečené protokolom TLS v1.2. Postup je podobný pre štandardné aj reverzné spojenie, príklad preto zaha obidve možnosti.

Pre nadviazanie TLS spojenia je potrebné, aby bol jeden z účastníkov v roli „TLS servera“ a druhý „TLS klienta“, pričom tieto roly nie sú závislé na tom, kto inicioval TCP spojenie. „TLS server“ sa preukazuje certifikátom, ku ktorému vlastní aj súkromný kľúč. „TLS klient“ overuje platnosť certifikátu a pravosť kľúčov<sup>5</sup>. Pre JAPI je „TLS server“ vždy *D2Connector* a „TLS klient“ vždy *JConnector*.

Predpokladom pre vytvorenie zabezpečeného spojenia je, že máme RSA kľúčový pár a k nemu X.509 certifikát. Certifikát je uložený v súbore vo formáte \*.  
crt a musí mať k jeho kópii prístup aj *D2Connector* aj *JConnector*. Prívätý kľúč je uložený v nešifrovanej forme v súbore vo formáte \*.pem a prístup k nemu musí mať iba *JConnector*. Vzhľadom k tomu, že *JConnector* považuje *D2Connector* za dôveryhodný iba na základe toho, že sa preukázal rovnakým certifikátom, ako ošakoval, použitý certifikát môže byť „self-signed“ a nie je vôbec potrebné získať certifikát od nejakej certifikanej autority.



**POZOR:** Z bezpečnostného hľadiska je **veľmi** dôležité, aby bol súbor s certifikátom uložený tak, aby ho nikto bez príslušného oprávnenia nemohol zmeniť. (Na íťanie môže byť verejný.) Takisto je **veľmi** dôležité, aby súbor so súkromným kľúčom mohol prejsť iba *D2Connector* a nikto ho nemohol zmeniť. V prípade porušenia týchto podmienok hrozí kompromitácia dôveryhodnosti certifikátu a otvára sa možnosť odpočúvania zabezpečenej komunikácie.

*D2Connector* je potrebné spustiť s parametrami --CONNECTOR\_TLS\_CERT a --CONNECTOR\_TLS\_PK, ktoré odkazujú na súbory s certifikátom a súkromným kľúčom. V príklade je certifikát uložený v súbore certificate.crt a súkromný kľúč v súbore private.pem. Podľa potreby sa môže použiť aj parameter --DCC alebo --CONNECTOR\_LISTEN\_PORT.

```
> d2connector.exe --CONNECTOR_TLS_CERT=certificate.crt --CONNECTOR_TLS_PK=private.pem
```

## Vytvorenie certifikátu pre účely zabezpečeného spojenia

Pre vytvorenie „self-signed“ certifikátu je možné použiť napríklad aplikáciu OpenSSL z príkazového riadku. Najskôr musíme vytvoriť kľúčový pár pre RSA šifru. V príklade generujeme 2048 bitový kľúčový pár do súboru private.pem.

```
> openssl.exe genrsa -out private.pem 2048
```

Ku kľúčovému páru vytvoríme „Certificate Signing Request“ žiadosť o vystavenie certifikátu, ktorá bude uložená v súbore request.csr. OpenSSL sa opýta na viaceré údaje, ktoré zapíše do žiadosti a ktoré budú vo výslednom certifikáte uvedené. JAPI však tieto údaje neskúma a nekontroluje.

```
> openssl.exe req -new -key private.pem -out request.csr
```

Následne podpíšeme žiadosť vygenerovaným súkromným kľúčom, čím z neho vytvoríme „selfsigned“ certifikát, ktorý bude v súbore certificate.crt. Certifikát bude mať platnosť 365 dní počnúc aktuálnym dňom.

```
> openssl.exe x509 -req -days 365 -in request.csr -signkey private.pem -out certificate.crt
```