Konfigurácia procesu D2Connector pre SmartWeb

V tejto kapitole je na príkladoch vysvetlené, ako nakofigurova 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 (prinajmenšom *kernel*), ku ktorej sa môžeme pripoji. Spôsoby pripojenia sú rozdelené do troch kategórií, priom možnosti z jednotlivých kategórií možno ubovone kombinova.

Pripojenie je možné nadviaza dvomi základnými spôsobmi:

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

Z hadiska zabezpeenia pred odpúvaním je možné nadviaza:

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

Z hadiska 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 ohadu na to, i 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 príruke systému D2000. Okrem toho akceptuje nasledovné parametre príkazového riadku:

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

D2Connector nadväzuje spojenie vždy len jedným spôsobom z ôsmych možných kombinácií. Tzn. bu sa aktívne pripája, alebo poúva, ale nie obidvoje naraz. Rovnako komunikuje bu nezabezpeeným alebo zabezpeeným spôsobom, ale nikdy neumožuje obidva spôsoby súasne. Bu je pripojený stále k jednému *Kernel*-u alebo sa prepína na aktuálny „hot". V prípade, že sa ku D2000 aplikácií pripája viac rôznych klientských aplikácií, ktoré vyžadujú rôzne spôsoby pripojenia, je potrebné naštartova pre každý spôsob samostatnú inštanciu *D2Connector*a.

Základný spôsob pripojenia

Ide o nezabezpeené spojenie, ktoré iniciuje JAPI.

D2Connector môžeme naštartova bez parametrov a bude poúva na pripojenie na porte 3120

```
> d2connector.exe
```

alebo zmeni poúvajúci port napríklad na 3121:

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

Nadviazanie reverzného spojenia

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

Klientská aplikácia je umiestnená na poítai `portal.dmz.customer.com` a poúva na porte 3125 na všetkých svojich sieových rozhraniach. *D2Connector* spustíme v režime pripájania sa:

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

Nadviazanie zabezpeeného spojenia

Ide o spojenie medzi *D2Connector*-om a *JConnector*-om zabezpeené 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 úastníkov v roli „TLS servera" a druhý „TLS klienta", priom tieto roly nie sú závislé na tom, kto inicioval TCP spojenie. „TLS server" sa preukazuje certifikátom, ku ktorému vlastní aj súkromný kú. „TLS klient" overuje platnos certifikátu a pravos kúov[5]. Pre JAPI je „TLS server" vždy *D2Connector* a „TLS klient" vždy *JConnector*.

Predpokladom pre vytvorenie zabezpeeného spojenia je, že máme RSA kúový pár a k nemu X.509 certifikát. Certifikát je uložený v súbore vo formáte `*.crt` a musí ma k jeho kópii prístup aj *D2Connector* aj *JConnector*. Privátny kú je uložený v nešifrovanej forme v súbore vo formáte `*.pem` a prístup k nemu musí ma iba *JConnector*. Vzhadom k tomu, že *JConnector* považuje *D2Connector* za dôveryhodný iba na základe toho, že sa preukázal rovnakým certifikátom, ako oakával, použitý certifikát môže by „self-signed" a nie je vôbec potrebné získa certifikát od nejakej certifikanej autority.

> **POZOR:** Z bezpenostného hadiska je **vemi** dôležité, aby bol súbor s certifikátom uložený tak, aby ho nikto bez príslušného oprávnenia nemohol zmeni. (Na ítanie môže by verejný.) Takisto je **vemi** dôležité, aby súbor so súkromným kúom mohol preíta iba *D2Connector* a nikto ho nemohol zmeni. V prípade porušenia týchto podmienok hrozí kompromitácia dôveryhodnosti certifikátu a otvára sa možnos odpoúvania zabezpeenej komunikácie.

D2Connector je potrebné spusti s parametrami `--CONNECTOR_TLS_CERT` a `--CONNECTOR_TLS_PK`, ktoré odkazujú na súbory s certifikátom a súkromným kúom. V príklade je certifikát uložený v súbore `certificate.crt` a súkromný kú v súbore `private.pem`. Poda potreby sa môže použi aj parameter `--DCC` alebo `--CONNECTOR_LISTEN_PORT`.

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

Vytvorenie certifikátu pre úely zabezpeeného spojenia

Pre vytvorenie „self-signed" certifikátu je možné použi napríklad aplikáciu OpenSSL z príkazového riadku. Najskôr musíme vytvori kúový pár pre RSA šifru. V príklade generujeme 2048 bitový kúový pár do súboru `private.pem`.

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

Ku kúovému páru vytvoríme „Certificate Signing Request" žiados 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 žiados vygenerovaným súkromným kúom, ím z neho vytvoríme „selfsigned" certifikát, ktorý bude v súbore `certificate.crt`. Certifikát bude ma platnos 365 dní poínajúc aktuálnym asom.

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