DNSSEC with PowerShell und dnscmd (on Windows 2012)

Windows 2012 can [DNSSEC sign zones using the GUI Management tools](). That works fine for a handfull of zones. But how about automation, signing hundreds of zones? Windows 2012 could be used as a "hidden master" DNSSEC signing engine that provides DNSSEC signed zones to Unix based front-end servers, while automating the key rollovers and continuous re-singing of the zones to refresh the record signatures. In that scenario, automation can be important.

dnscmd

One automation option is the `dnscmd` tool. However, `dnscmd` is deprecated and might be removed from future versions of Windows. The more modern way is to use PowerShell (see below).

```
Usage: DnsCmd <ServerName> /ZoneAddSKD <ZoneName> /Alg <KeyAlg> [/Length
<KeyLength>]
                      [/KSP <KeyStorageProviderName>] [/Flags <KeyFlags>]
[/StoreKeysInAD]
                      [/DoNotStoreKeysInAD] [/InitialRolloverOffset
<Seconds>]
                      [/DNSKEYSignatureValidityPeriod <Seconds>]
[/DSSignatureValidityPeriod <Seconds>]
                      [/StandardSignatureValidityPeriod <Seconds>]
[/RolloverPeriod <Seconds>]

  <KeyStorageProviderName> -- "Microsoft Software Key Storage Provider" or
                     other KSP installed on this system
  <KeyAlg>        -- the key algorithm mnemonic string. Currently only
                     "RSASHA1", "NSEC3RSASHA1", "RSASHA256", "RSASHA512",
                     "ECDSAP256SHA256" and "ECDSAP384SHA384" are supported.
  <KeyLength>     -- length of the key in bits. For RSA algorithms the value
                     can range from 1024 bits in 64 bit increments up to 4096
                     bits. Key size is ignored for other algorithms.
  <KeyFlags>      -- bits to be set to 1 in DNSKEY flags field. If
                     is "KSK",  the Secure Entry Point bit will be set to 1
                     to indicate that this key is a Key Signing Key. If no
                     /Flags is given, the key is considered to be a Zone
                     Signing Key.
  /StoreKeysInAD  -- if specified, keys will be stored on the zone object in
                     Active Directory and will replicate to other DNS servers
  /DoNotStoreKeysInAD -- if specified, keys will be stored in a self-signed
                     certificate in the local machine certificate store and
                     will not replicate to other DNS servers

In future versions of Windows, Microsoft might remove dnscmd.exe
```

The `dnscmd` subcommand `ZoneAddSKD` creates a new DNSSEC key and assigns that key to a zone. Without a key assigned, `dnscmd` will refuse to sign the zone. Be careful not to assign more than the needed keys to a zone. Every call to `ZoneAddSKD` will add a new key to the zone. Usually we want to have one Key Signing Key (KSK), and one Zone Signing Key (ZSK):

creating the KSK

```
PS C:\Users\Administrator> dnscmd server1 /ZoneAddSkd dnscmd.example.com /Alg
RSASHA256 /Length 2048 /Flags KSK

Command returned the following Signing Key Descriptor:

SKD GUID {EDADBF50-A733-4C67-A95E-8145C77BE0A1}
        key storage provider                = Microsoft Software Key Storage
Provider
        store keys in AD                    = 0
        KSK flag                            = 1
        signing algorithm                   = RSASHA256
        key size                            = 2048
        initial rollover offset             = 0
        DNSKEY signature validity period    = 604800
        DS signature validity period        = 604800
        standard signature validity period  = 864000
        rollover period                     = 65232000
        next rollover action                = Normal

Command completed successfully
```

creating the ZSK

```
PS C:\Users\Administrator> dnscmd server1 /ZoneAddSkd dnscmd.example.com /Alg
RSASHA256 /Length 1024

Command returned the following Signing Key Descriptor:

SKD GUID {47D410EE-35A4-483C-9B3B-1FB763F5AAB5}
        key storage provider                = Microsoft Software Key Storage
Provider
        store keys in AD                    = 0
        KSK flag                            = 0
        signing algorithm                   = RSASHA256
        key size                            = 1024
        initial rollover offset             = 0
        DNSKEY signature validity period    = 604800
        DS signature validity period        = 604800
        standard signature validity period  = 864000
        rollover period                     = 7776000
        next rollover action                = Normal

Command completed successfully.
```

To be precise, the commands above do not directly create the keys, instead they create a "Signing Key Descriptor (SKD)", which is a kind of policy that describes how DNSSEC keys are generated for this zone. The Windows 2012 DNS server then creates new keys based on the descriptors whenever needed (for example during a key rollover).

Now we can use `dnscmd` subcommand "ZoneSign" to sign the zone. The parameters for "ZoneSign" are:

```
Usage: DnsCmd  /ZoneSign <ZoneName>
```

```
  Generates keys and adds DNSSEC records to the zone using the zone signing
  parameters and Signing Key Descriptors (SKDs) currently set on this zone.

In future versions of Windows, Microsoft might remove dnscmd.exe.
```

The only required parameter is the name of the zone to sign. This zone must already have DNSSEC keys attached:

```
PS C:\Users\Administrator> dnscmd server1 /ZoneSign dnscmd.example.com
Command completed successfully.
```

PowerShell

With PowerShell, we use the command `Add-DnsServerSigningKey` to create and add the DNSSEC Signing Key Descriptors to a DNS zone:

```
SYNTAX: Add-DnsServerSigningKey [-ZoneName] <string>
             [[-Type] <string> {KeySigningKey | ZoneSigningKey}]
             [[-CryptoAlgorithm] <string> {RsaSha1 | RsaSha256 | RsaSha512 |
RsaSha1NSec3 | ECDsaP256Sha256 | ECDsaP384Sha384}]
             [[-KeyLength] <uint32>]
             [-ComputerName <string>]
             [-InitialRolloverOffset <timespan>]
             [-DnsKeySignatureValidityPeriod <timespan>]
             [-DSSignatureValidityPeriod <timespan>]
             [-ZoneSignatureValidityPeriod<timespan>]
             [-RolloverPeriod <timespan>]
             [-ActiveKey <string>]
             [-StandbyKey <string>]
             [-NextKey <string>]
             [-KeyStorageProvider <string>]
             [-StoreKeysInAD <bool>]
             [-PassThru]
             [-CimSession <CimSession[]>]
             [-ThrottleLimit <int>]
             [-AsJob]
             [-WhatIf]
             [-Confirm]
             [<CommonParameters>]
```

Again, we creating one for the KSK, and one for the ZSK:

creating the KSK

```
PS C:\Users\Administrator> Add-DnsServerSigningKey powershell.example.com -
Type KeySigningKey -CryptoAlgorithm RsaSha256 -KeyLength 2048
```

creating the ZSK

```
PS C:\Users\Administrator> Add-DnsServerSigningKey powershell.example.com -
Type ZoneSigningKey -CryptoAlgorithm RsaSha256 -KeyLength 1024
```

The last step is to sign the zone, where we use `Invoke-DnsServerZoneSign`.

```
SYNTAX: Invoke-DnsServerZoneSign [-ZoneName] <string>
                    [-SignWithDefault]
                    [-DoResign]
                    [-ComputerName <string>]
                    [-Force]
                    [-PassThru]
                    [-CimSession <CimSession[]>]
                    [-ThrottleLimit <int>]
                    [-AsJob]
                    [-WhatIf]
                    [-Confirm]
                    [<CommonParameters>]
PS C:\Users\Administrator> Invoke-DnsServerZoneSign powershell.example.com

Confirm
This will initiate online signing of the zone powershell.example.com on
server SERVER1.
Do you want to continue?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): y
```