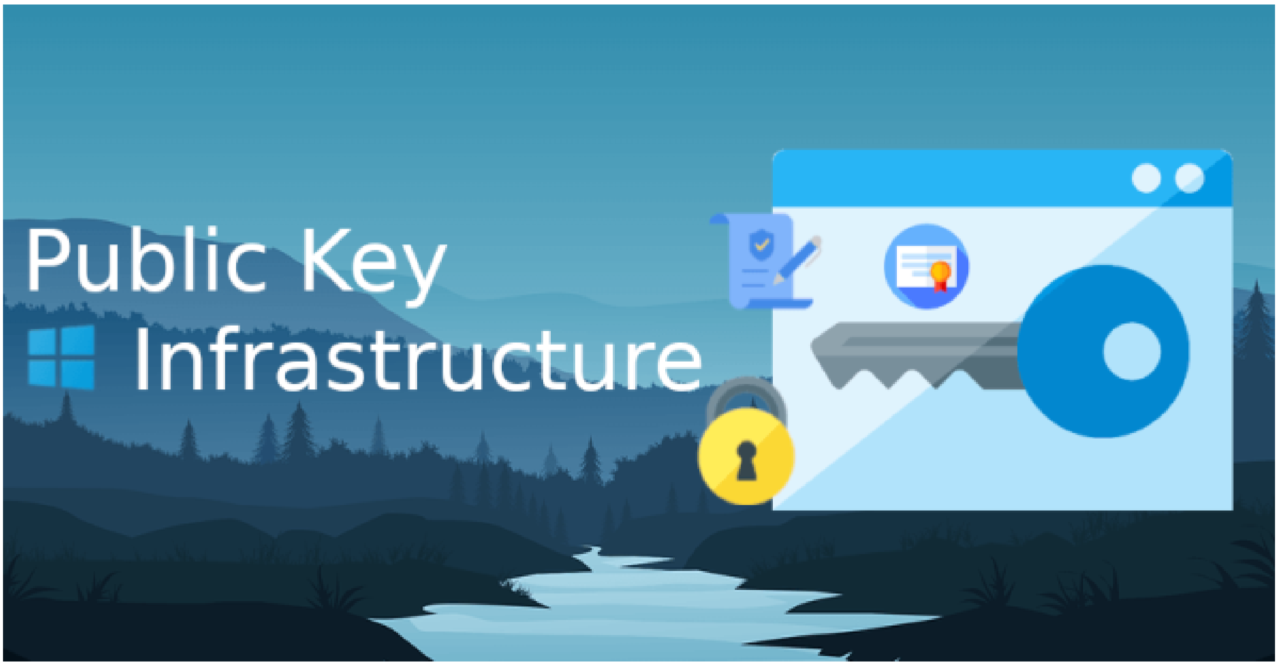
Part 5 – PKI Best Practices: Creating Unique Object Identifiers (OIDs)

michaelwaterman.nl/2025/05/10/part-5-pki-best-practices-creating-unique-object-identifiers-oids

Michael Waterman

May 10, 2025



Introduction: Why OIDs matter in PKI

When building or managing a Public Key Infrastructure (PKI), precision and uniqueness are not optional, they're very essential. Don't be one of many, be your unique self! One key element that reflects this is the Object Identifier (OID). OIDs are globally unique values used to identify everything from certificate policies and application purposes to custom certificate extensions and cryptographic algorithms.

In the context of PKI, you might use an OID to:

- Define certificate policies (e.g., "This certificate may be used for code signing")
- Identify extended key usages (e.g., Smart Card Logon, Client Authentication)
- Tag custom extensions or internal use-cases (e.g., VPN client identification)

Every time you create a new certificate template, define a custom policy, or build a trust model between systems, you're likely using (or should be using) an OID. And most importantly: it has to be unique.

There are two main strategies for obtaining private OIDs:

1. *Requesting a Private Enterprise Number (PEN) from IANA*, and using it to build your OID hierarchy.
2. Automatically generating a forest-unique OID based on your Active Directory GUID, using Microsoft's assigned OID namespace.

In this post, we'll break down both options: how they work, when to use which, and how to generate and manage them using PowerShell.

Option 1: Requesting a Private Enterprise Number (PEN) from IANA

A Private Enterprise Number (PEN) is a unique identifier assigned to your organization by IANA. Once obtained, it gives you control over your own OID namespace under:

```
1.3.6.1.4.1.<your PEN>
```

Understanding the Structure `1.3.6.1.4.1`

The OID structure assigned by IANA for private enterprises follows a globally recognized format:

- `1` — ISO (International Organization for Standardization)
- `3` — Identified Organization
- `6` — U.S. Department of Defense
- `1` — Internet (IETF-assigned OIDs)
- `4` — Private
- `1` — Enterprise identifiers (i.e., PENs)

Any PEN you receive from IANA is appended to this path. For example, if your organization receives PEN 12345, your OID root becomes:

```
1.3.6.1.4.1.12345
```

From here, you can create sub-OIDs such as:

```
1.3.6.1.4.1.12345.1.1# Internal certificate policy
1.3.6.1.4.1.12345.2.5# Application-specific extension
```

As you can see, these policies should have unique numbers but don't have their origin in technology, they are written policies, how to handle certificates, but also on managing the infrastructure.

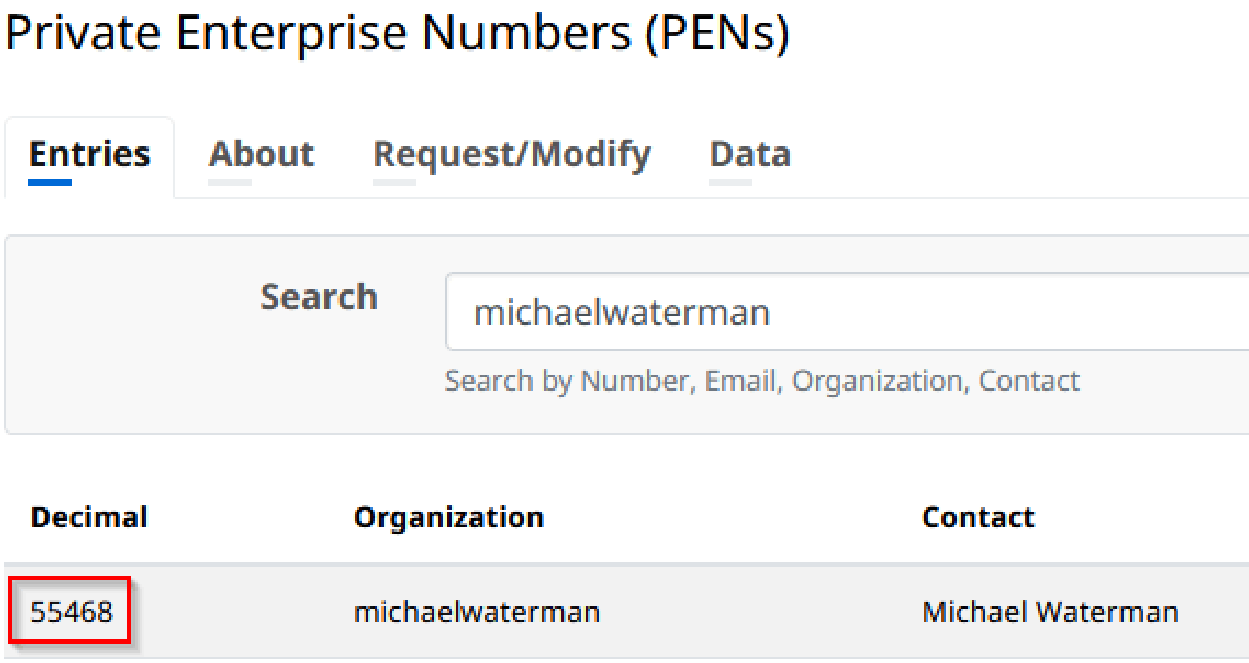
Advantages:

- Globally unique and recognized
- Clean and short OID structure
- Full control over sub-OID structure

How to request it:

1. Go to IANA PEN request page
2. Submit your organization name and contact information
3. Approval typically takes 1–2 business days

Once approved, your PEN and organization name appear publicly on the IANA enterprise number registry. Suppose you want to lookup my unique PEN number, go to the following url and type in my name “michaelwaterman”.



Private Enterprise Numbers (PENs)

Search: michaelwaterman

Search by Number, Email, Organization, Contact

Decimal	Organization	Contact
55468	michaelwaterman	Michael Waterman

As you can see in the picture above, my unique number is *55468*.

This is fundamentally different from Microsoft’s method of using the forest GUID , which leverages Microsoft’s own PEN (`311`) and embeds your forest identity as a long byte sequence. How Microsoft solves this without requiring an IANA request is up next, by leveraging your Active Directory forest’s unique GUID.

Option 2: Forest-GUID based OIDs via Microsoft’s namespace

If you don’t want to go through the process of requesting a PEN, Microsoft provides an alternative: using your Active Directory forest’s GUID to generate a private and unique OID within their enterprise-assigned space. The structure for this is:

```
1.3.6.1.4.1.311.21.8.<a.b.c.d.e...>.1.xxx
```

Breakdown of the prefix `1.3.6.1.4.1.311.21.8`

Each part of this OID prefix has a specific meaning:

- `1.3.6.1.4.1` — The IANA-assigned OID arc for private enterprises (see the previous chapter for an in-depth explenation)

- `311` — Microsoft’s Private Enterprise Number (PEN)
- `21` — Designates Microsoft’s certificate-related subtree
- `8` — Reserved for forest-specific private OIDs (one per AD forest)

The `<a.b.c.d.e...>` portion that follows is derived from the forest’s globally unique identifier (GUID), converted to a decimal byte sequence. This ensures uniqueness without needing external registration.

Why This Works

Your AD forest has a globally unique GUID. By converting this into a decimal byte sequence, you get an OID that’s just as unique as an IANA-assigned one — without submitting any form.

PowerShell Example: Generate OID from Forest GUID

```
# Query Active Directory to automatically get the forest root object's GUID
$searchBase="CN=Partitions,"+(Get-ADRootDSE).configurationNamingContext
$forestRootNC=(Get-ADRootDSE).rootDomainNamingContext
$forestObject=Get-ADObject-SearchBase $searchBase-LDAPFilter "
(nCName=$forestRootNC)"-Properties objectGUID

# Convert the GUID to an OID-friendly decimal format
$guid=$forestObject.objectGUID
$bytes=$guid.ToByteArray()
$oidComponent=($bytes|ForEach-Object{[int]$_ })-join'.'
$customOID="1.3.6.1.4.1.311.21.8.$oidComponent.1.402"
Write-Output"Generated forest-based OID: $customOID"
```

This results in a long but fully unique OID, like:

```
1.3.6.1.4.1.311.21.8.244.70.31.158.19.45.241.64.207.144.177.179.242.233.250.172.1.
402
```

Advantages:

- No external request required
- Automatically unique per forest
- Fully compatible with Microsoft PKI services

Considerations:

- The resulting OID is longer and less readable
- You should keep documentation of what each `.1.xxx` sub-OID means

You can now define your own policies by extending this base, for example:

```
...<GUID bytes>.1.401# Smartcard policy
...<GUID bytes>.1.402# VPN policy
```

Both approaches are valid — the best one depends on your organization's needs. In the next section, we'll compare them side by side.

Comparison table: IANA PEN vs Forest-GUID OID

When deciding between an IANA-registered PEN and Microsoft's forest-based GUID OIDs, it's important to understand the implications for structure, uniqueness, and management. The table below compares the two approaches to help you choose the one that fits your PKI and operational needs best:

Conclusion

Choosing the right strategy for private OID management depends on your operational context and goals. Both the IANA-assigned PEN and the forest-GUID-based Microsoft method provide unique advantages, and the best choice comes down to your environment: having your own unique OIDs is essential, don't use the same OID everyone copies from the Internet.

- If you need a clean, auditable, and globally recognized namespace, especially for public-facing or standards-aligned deployments, an IANA-assigned PEN is the right choice.
- If you're operating within an internal AD environment and need fast, automated, and conflict-free OID generation, Microsoft's forest-GUID-based approach offers simplicity and scalability.

In either case, clarity and documentation are key. Know what each OID represents, keep track of your custom extensions and policies, and structure your hierarchy deliberately.

With the PowerShell tools and insights from this guide, you're now ready to create and manage OIDs confidently within your PKI ecosystem.

Next up we'll dive into the CAPolicy.inf file where we will be using the information from this blog, stay tuned!

As always, if you have any feedback, please let me know! Until next time.