# AD Enterprise Population

# AD Enterprise Population

## Requirements

- An already configured server with AD DS (Active Directory Domain Services) is needed to run this tools, if you haven't done so you Building AD Lab section.

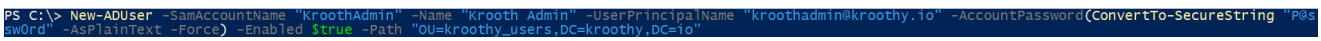
### New Admin User

- We need to have a user with Domain Admin and Schema Admin, while it is indeed possible to use the Administrator account, we don't want to do that for reasons we'll talk about in our Active Directory Best Practices blog. So then, yes you guessed it we're creating an admin account and adding it to Domain Admins and Schema Admins groups.

  > ⚠ **Caution**
  > Replace the names and the specific paths, according to your setup. All names should be placed inside a quotes.

- Adding new user:

```
New-ADUser -SamAccountName <User Name> -Name <Full Name> -UserPrinciapal "<User Name>@doamin.domainextension" -AccountPassword(ConvertTo-SecureString <Your_Password> -AsPlainText -Force) -Enabled $true -Path "OU=<your_path>,DC=domain,DC=domainextension"
```




> 🖉 **Note**
>
> The passwords used in this setup are not of best practice and are only used for Lab purpose. Production passwords should be way more secure than this.

- Add the New user to Domain Admins and Schema Admins security group:

```
$user = <User_to_be_added_to_group>
$groups = @(<groups_we_want_to_add_user_to>)

foreach($group in $groups) {
```

```
                $groupObject = Get-ADGroup -Identity $group
                Add-ADPrincipalGroupMembership -Identity $user -MemberOf
$groupObject
        }
```

imgs/49045218190e3ee7544c2ed91e474ea9_MD5.jpeg

```
PS C:\> $user = "KroothAdmin"
PS C:\> $groups = @("Domain Admins", "Schema Admins")
PS C:\> foreach ($group in $groups) {
>> $groupObject = Get-ADGroup -Identity $group
>> Add-ADPrincipalGroupMembership -Identity $user -MemberOf $groupObject
>> }
```

### Install Git

- We need to install git for windows, we can do that by going to here
- Once we download the setup, we can easily install git, it's pretty intutive.
- Finally restart our server.

## AD Enterprise Population Process

- For this we'll be mainly using the tool Badblood by @davidprowe, and some other PowerShell scripts as well.

## Badblood

- Badblood is a tools is a security tool for Active Directory, it used to populate an AD, so professionals can simulate/learn/demo security related concepts regarding AD.
- Each Badblood script execution generates different results, with different users, groups, computers and permissions.
- Now, let's get to the fun part AD Enterprise Population > Steps

> ⚠ **Warning**
> Under no circumstance should you run badblood in a production system, this is meant only for Lab purposes.

### Steps

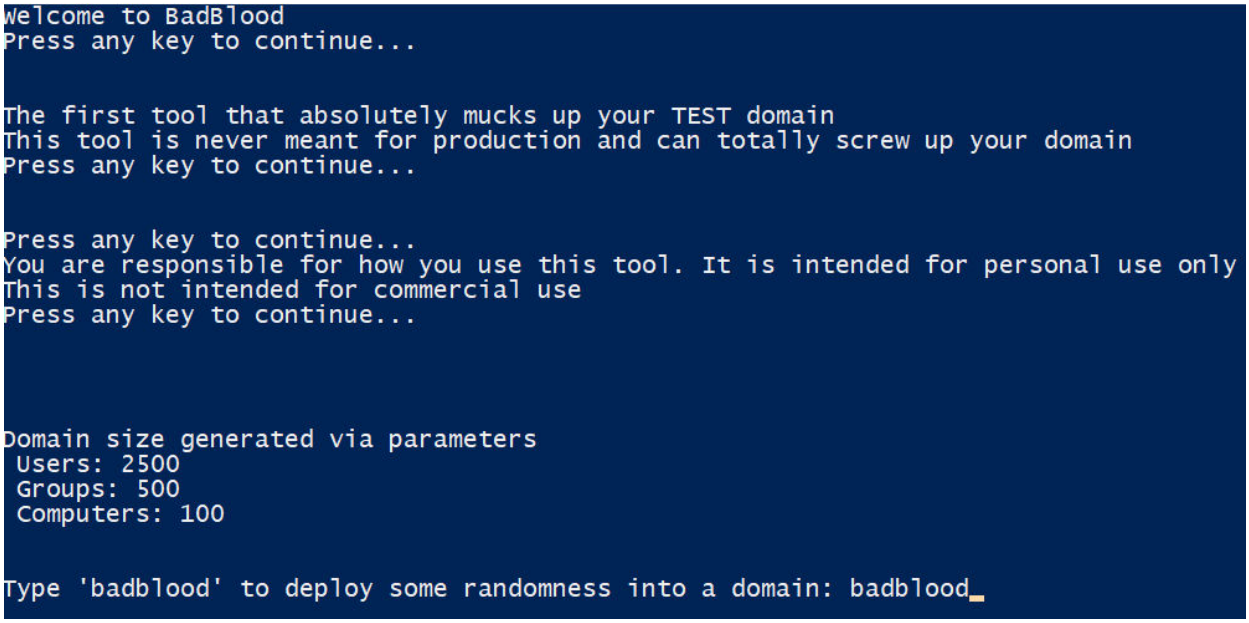
- Installing Badblood (Cloning the repo)

```
git clone https://github.com/davidprowe/badblood.git
```

- After cloning the repo, we should be able to run badblood:

```
cd .\badblood\
.\Invoke-BadBlood.ps1
```

- We are then prompted several times, to press any keys, and also we're promoted to type 'badblood' to add some randomness to the generated Users, Groups and Computers.
  imgs/2a5bc352d041f5137be746733f537fd5_MD5.jpeg




- This will take quite sometime and then we'll then see success in populating our AD, for better population, we can run badblood several times, as badblood generates d/t data each time around, to do that we follow the same step above.

# {{References}}


- https://www.secframe.com/badblood/
- https://github.com/davidprowe/BadBlood?tab=readme-ov-file
- https://gitforwindows.org/