Group Policy Objects

GPOs - Background

- Group policy objects (GPOs) are essentially collections of settings that are applied to groupings of computers (***and users!***)
  - By default, group policy is updated in the background every 90 minutes, with a randomized offset of 0-30 minutes
  - Settings are stored as files in SYSVOL that all domain users can read
- What (interesting) things can GPOs set?
  - Local admin passwords
  - Local group membership
  - User rights assignment (i.e. SeLoadDriverPrivilege)
  - LAPS settings
  - Registry entries
  - Scheduled tasks, logon/logoff scripts, and tons more!

GPO Settings

- After settings are defined in a GPO, the GPO is linked to:
  - A site
  - A domain object itself (i.e. the 'Default-Domain-Policy')
  - An organizational unit (OU) - this is the most common application
- These links can easily be enumerated through the **gpLink** attribute of OU/site/domain objects in AD

```
PS C:\Users\dfm.a\Desktop> Get-DomainOU -LDAPFilter "(gplink=*)" | Select -Last
1


usncreated             : 58277
name                   : Workstations
gplink                 : [LDAP://cn={47543975-8606-4B80-A86C-FCA31369F434},cn=po
                         licies,cn=system,DC=testlab,DC=local;0]
whenchanged            : 4/10/2017 10:40:13 PM
objectclass            : {top, organizationalUnit}
usnchanged             : 58287
dscorepropagationdata  : {4/10/2017 10:39:25 PM, 1/1/1601 12:00:00 AM}
distinguishedname      : OU=Workstations,DC=testlab,DC=local
ou                     : Workstations
```

OU GPO Inheritance

- When a machine enumerates OU GPOs that it may need to apply, it starts with the “lowest-level” OU
  - i.e. for “CN=WINDOWS1,OU=Child,OU=Parent, …”, “OU=Child” is applied before “OU=Parent”
- OUs can block inheritance of GPOs applied to higher level OUs by setting **gpOptions=1**
- BUT higher level GPOs can be set to “enforced”, which overrides any lower-level OU attempts to block it
  - PowerView’s **Get-DomainGPO -ComputerIdentity** handles all this logic for you :)

GPO -> Computer Correlation

- If you have a *particular* GPO and you want to know what systems it applies to:
  - **Get-DomainOU -GPLink '<GUID>' | % {Get-DomainComputer -SearchBase $_.distinguishedname -Properties dnshostname}**

```
PS C:\Users\dfm.a\Desktop> Get-DomainOU -GPLink 'D61EC832-B979-4BC6-B1
B7-ACF2147EF76D' | % {Get-DomainComputer -SearchBase $_.distinguishedn
ame -Properties dnshostname}

dnshostname
-----------
WINDOWS2.testlab.local
```

Restricted Groups

- There are two ways that GPOs can set local group memberships: **Restricted Groups** and **Group Policy Preferences**
- The information for Restricted Groups (GPO\Computer Configuration\Windows Settings\Security Settings\Restricted Groups) is stored at as an .ini file in **GPO\MACHINE\Microsoft\Windows NT\SecEdit\GptTmpl.inf**
  - We want the *S-1-5-32-544__members ('Administrators') and the name/SID of any domain group with a 'GROUP__memberof = *S-1-5-32-544' set (meaning that group is a member of local administrators)
  - Can modify the local group SID (i.e. can substitute "Remote Desktop Users"/S-1-5-32-555)

Restricted Groups

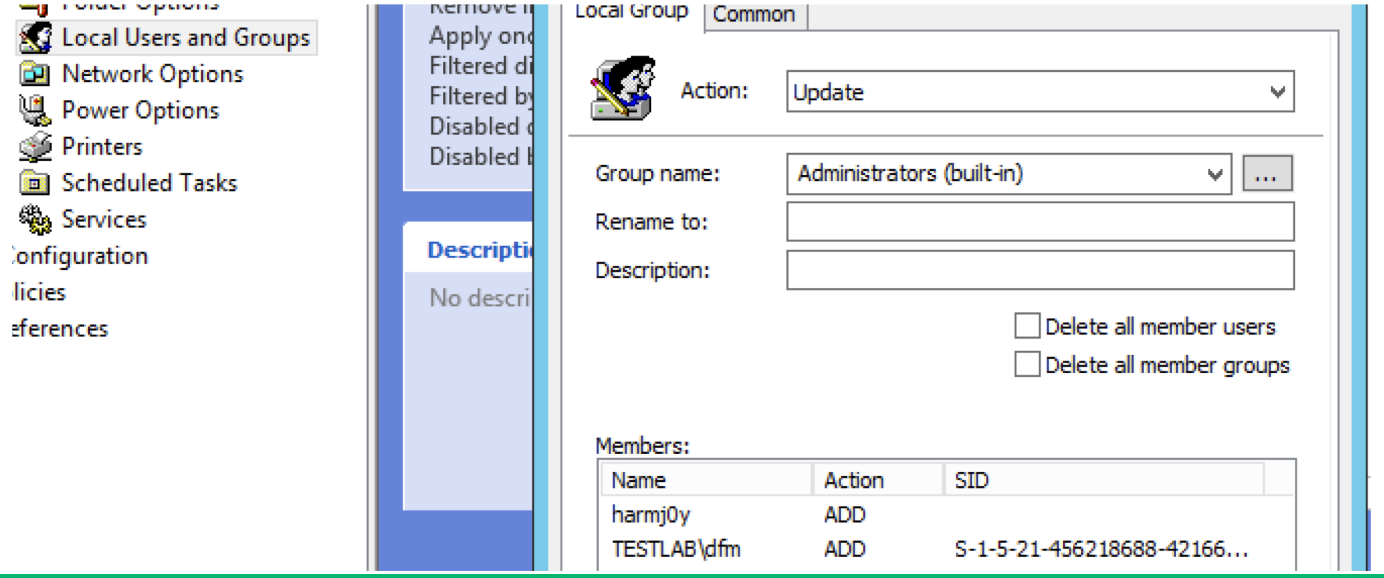
- Here's how local groups can be nested, which determined what relationships we cared about in the previous slide using Restricted Groups:

	Local Group	Domain Group
Using of "Members"	• Local Users • Domain Users • Domain Groups	Not applicable
Using "Member Of"	Not Applicable (*)	• Local Groups

http://social.technet.microsoft.com/wiki/contents/articles/20402.active-directory-group-policy-restricted-groups.aspx

Group Policy Preferences

- Settings are stored as an .XML in GPO\MACHINE\Preferences\Groups\Groups.xml
  - Allows for really granular applications of settings through environmental keying (by hostname, WMI info, etc.)



GPO Local Group Correlation

- **For mass enumeration:**
  - Enumerate all GPO objects
  - Parse any Restricted Groups (GptTmpl.inf) files found, as well as any Group Policy Preferences (Groups.xml), extracting out any information that modifies local group membership
  - For any GPO that modifies local groups, search for any OU, site, and/or domain object where the gPlink field matches the GPO GUID
  - Enumerate all computers that are a part of the OU/site/domain
- **For specific user/group enumeration:**
  - Enumerate all groups the user/group is a nested part of
  - Filter the raw GPO mapping by the SIDs for the user/group and any group the target is a part of

Sidenote: Code Execution With GPOs

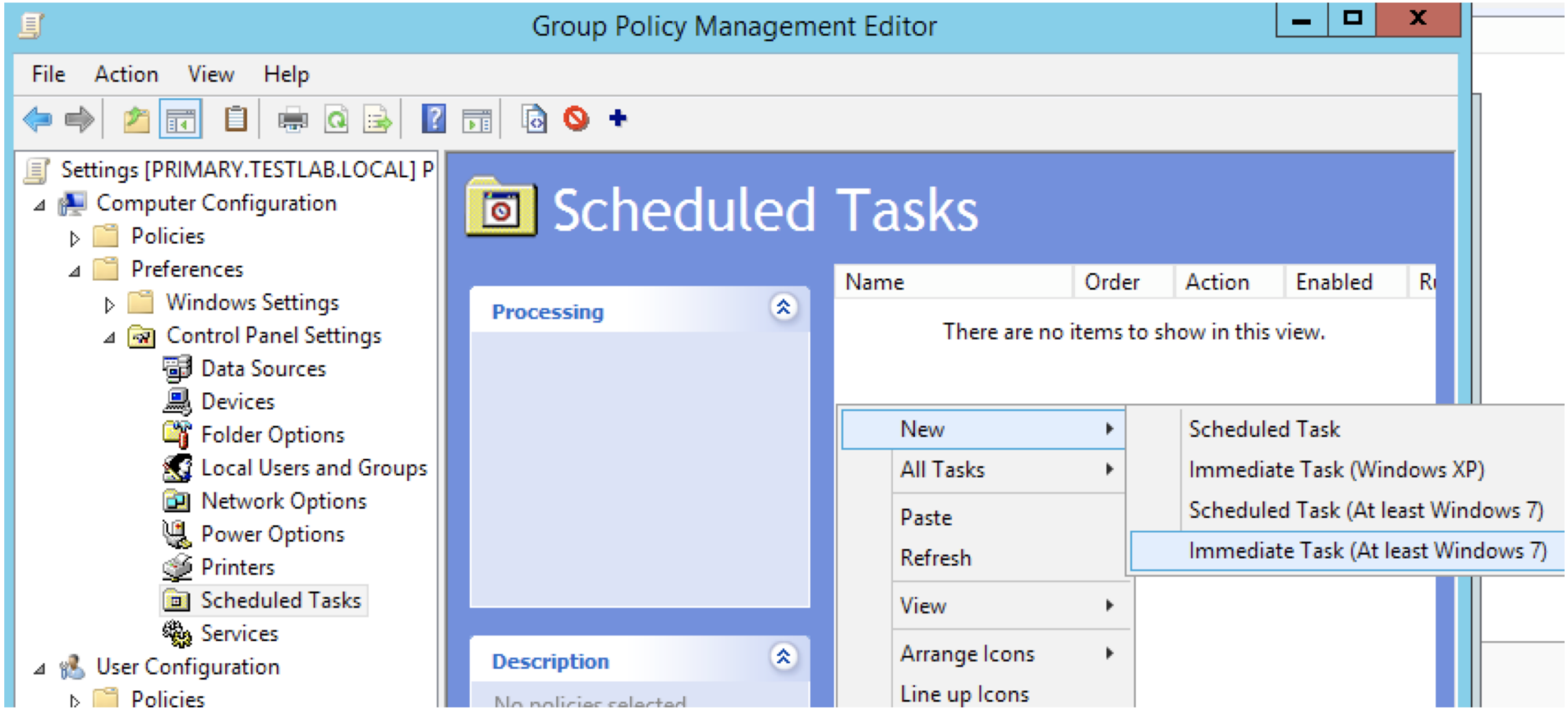
- ACLs come later, but what we care about with GPOs are the edit rights to the **gpcfilesyspath** property
  - These rights are cloned onto the GPO folder in SYSVOL
  - Remember that GPOs can apply to both *users* and *computers*

- There a large number of different ways GPOs can be used to compromise users/machines they're applied to

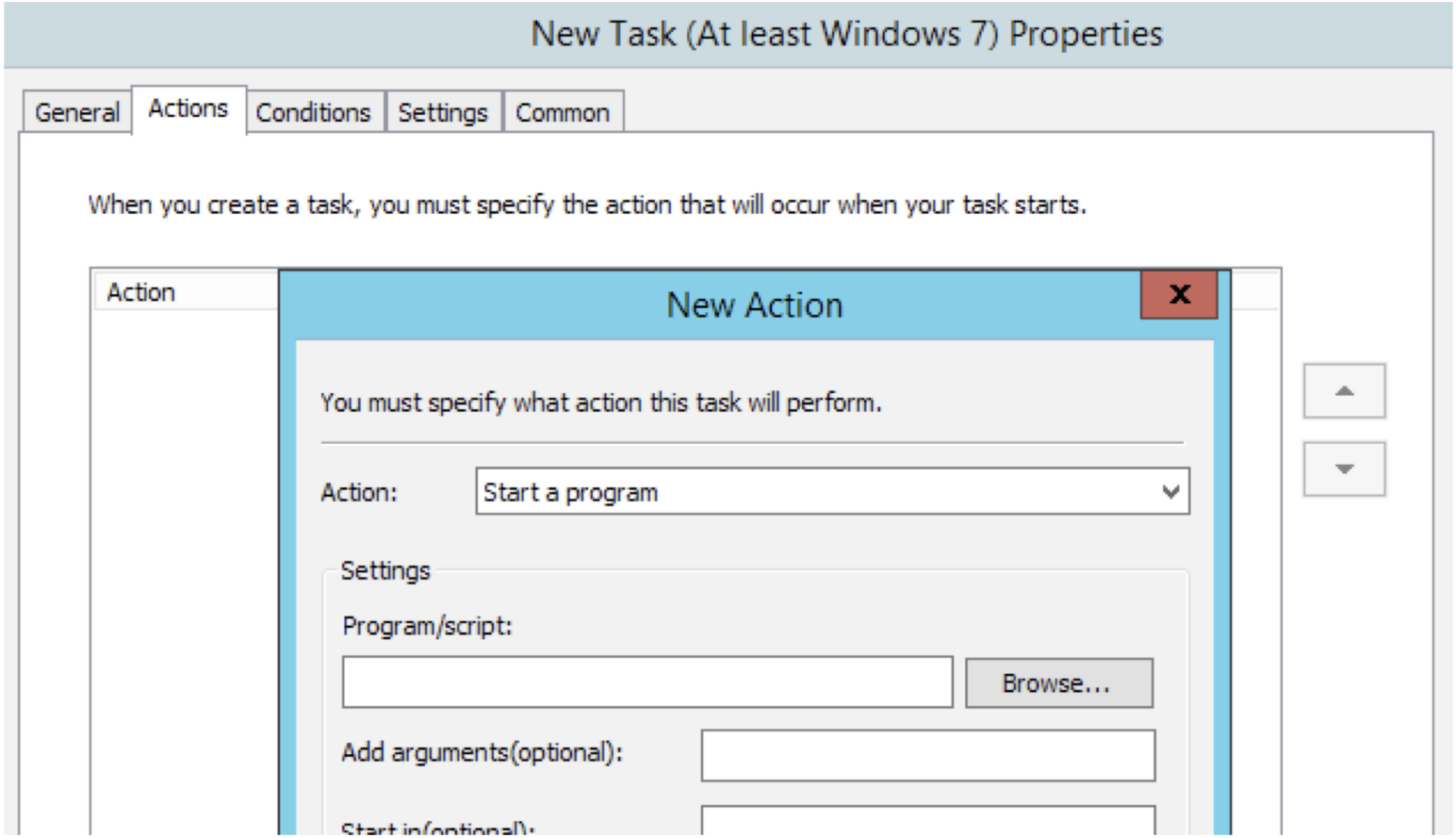
Code Execution With GPOs

- There are a number of ways GPOs can be used to gain code execution on a system or user the GPO is applied to:
  - Add local admin with Restricted Groups/GPP
  - Add registry autoruns
  - Software Installation -> push out .MSI packages
  - Scripts -> push scripts to startup/shutdown folder
  - Shortcuts -> malicious LNK file
  - Scheduled tasks -> New Immediate Scheduled Task, New Scheduled Task
- Our preference is an “Immediate” scheduled task, which runs and then deletes itself immediate after

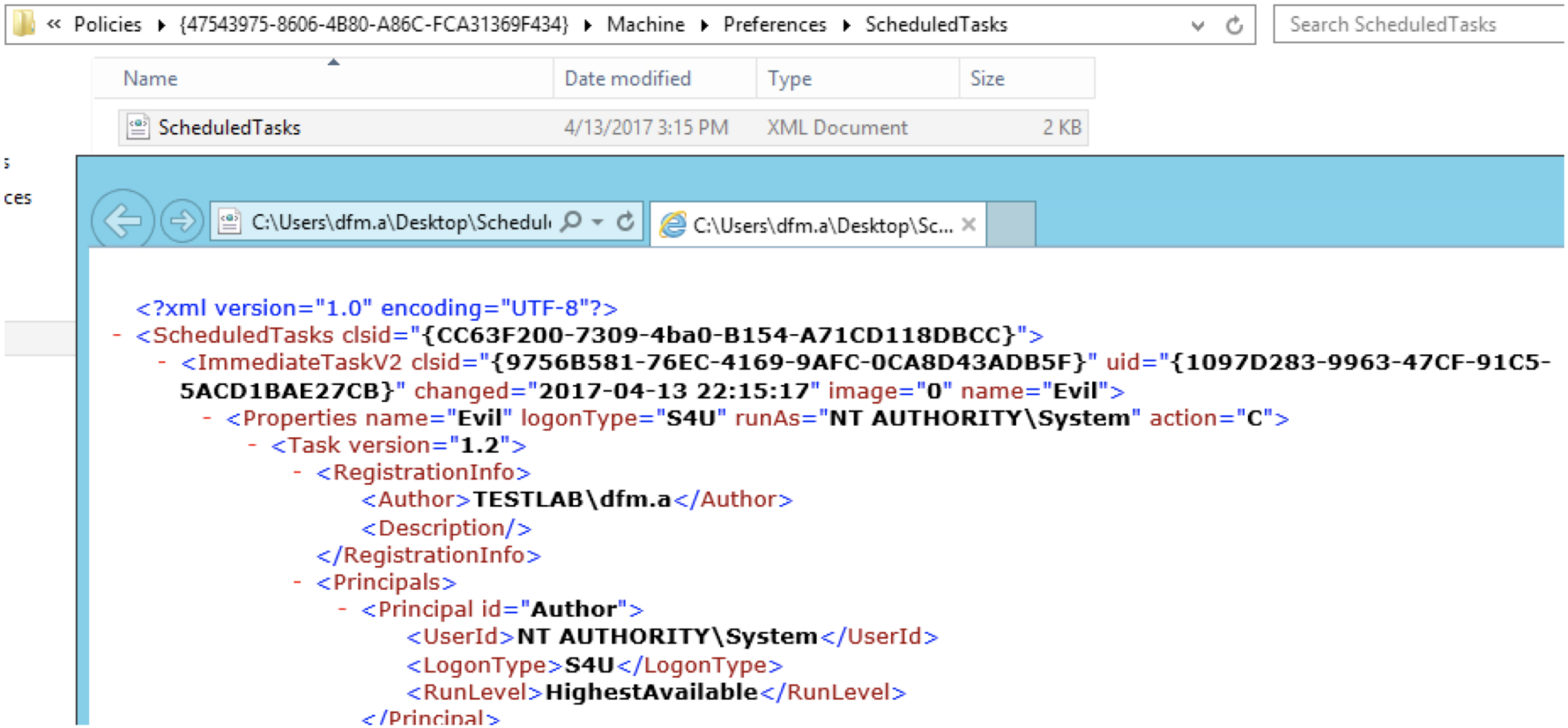
Code Execution With GPOs



Code Execution With GPOs



Code Execution With GPOs



This would be a good time to attempt Lab: GPOs