The only command you will ever need to understand and fix your Group Policies (GPO)

evotec.xyz/the-only-command-you-will-ever-need-to-understand-and-fix-your-group-policies-gpo/

January 24, 2021

I've been working on cleaning up **Group Policies** for a couple of months. While it may seem trivial, things get complicated when you're tasked with managing 5000 **GPOs** created over 15 years by multiple teams without any best practices in mind. While working on **GPOZaurr** (my new **PowerShell** module), I've noticed that the more code I wrote to manage those **GPOs**, the more I knew passing this knowledge to admins who will be executing this on a **weekly/monthly basis** is going to be a challenge. That's why I've decided to follow a similar approach as my other Active Directory testing module called **Testimo**. I've created a single command that analyses **Group Policies** using different methods and shows views from different angles to deliver the full picture. On top of that, it provides a solution (or it tries to) so that it's fairly easy to fix – as long as you agree with what it proposes.

Please be careful when using this on production

I've done a lot of research and put a lot of effort into making sure this **PowerShell** module works as expected. However, I do make mistakes. Contrary to my usual work, this module is not read-only. To almost every read command, there is also a set or remove command. It can change things, delete them, or modify them. If you don't understand what will happen, don't do it. Review source code, run read commands first to understand the output, what it's showing. If you have doubts – don't use it or create an issue on **GitHub** to clarify. All cmdlets that have the ability to write/delete contain **Whatlf**/**LimitProcessing** count parameters. Use them before implementing any changes!

Please keep in mind I've tested GPOZaurr only on English based Active Directory. I have no clue how it will behave on non-English systems. As I've not worked with other languages for a while, I don't remember if object types are still reported in English by PowerShell or reported in language equivalent. Be careful.

Useful Links
Please make sure to visit **GitHub** to review sources or report issues. If you're going to use it, I recommend doing it via **PowerShellGallery** as that version is minimized and optimized. Reviewing sources is easier on the GitHub version as it has more comments and is divided into sections.

The code is published on GitHub

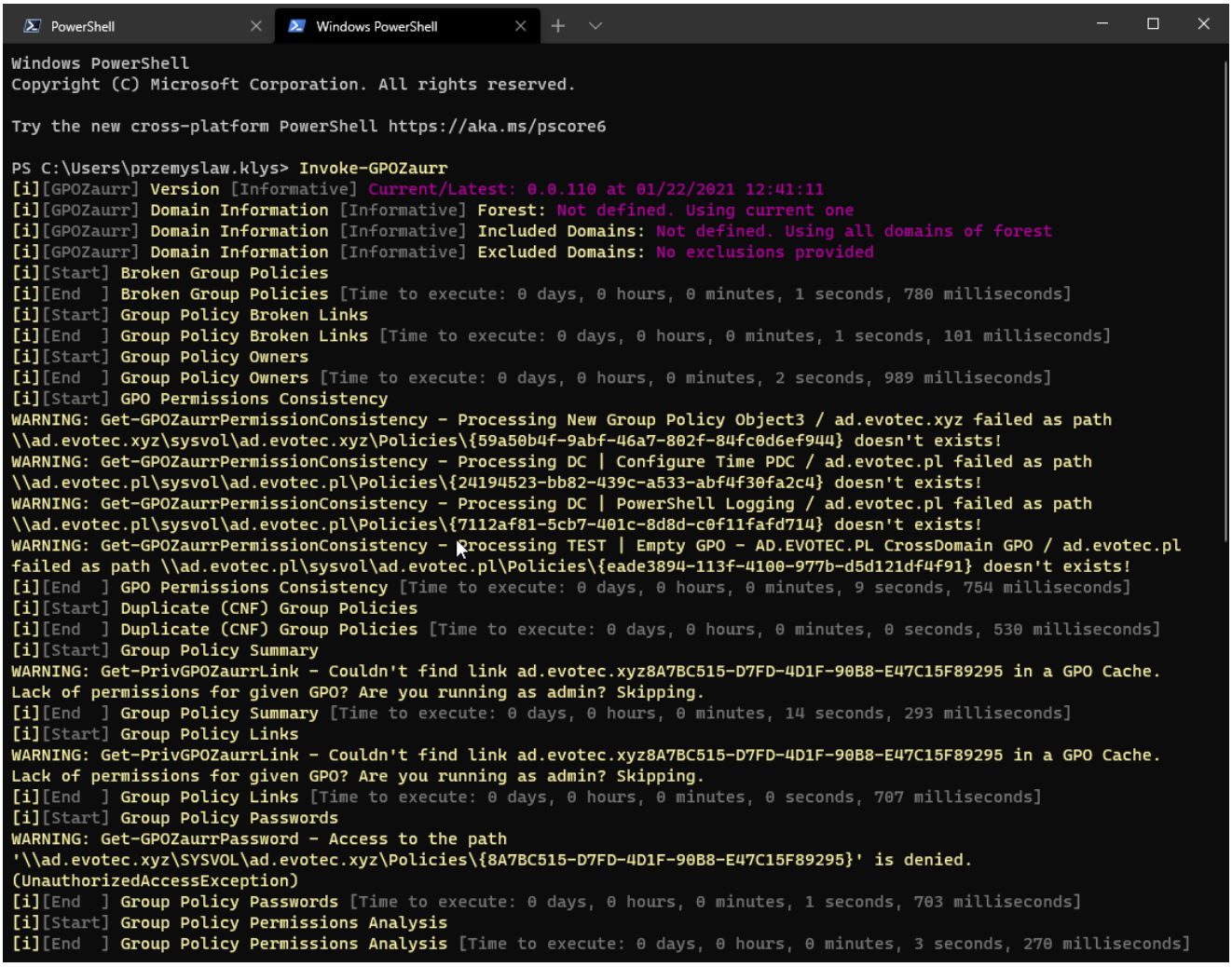
Issues should be reported on GitHub

Code is published as a module on PowerShellGallery

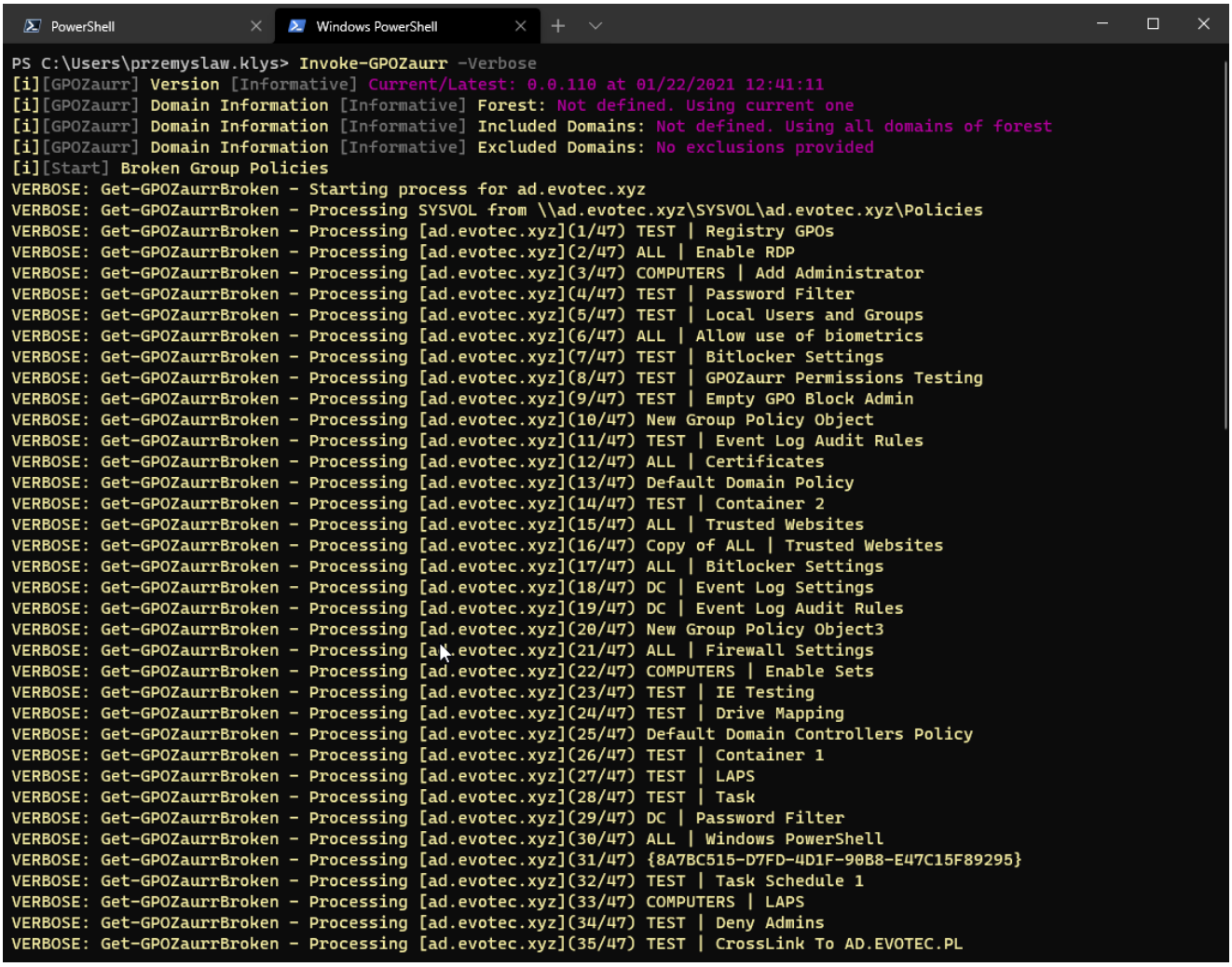
The module is signed with a certificate, like any new modules that I create or update.

```
Install-Module GPOZaurr -Force
```

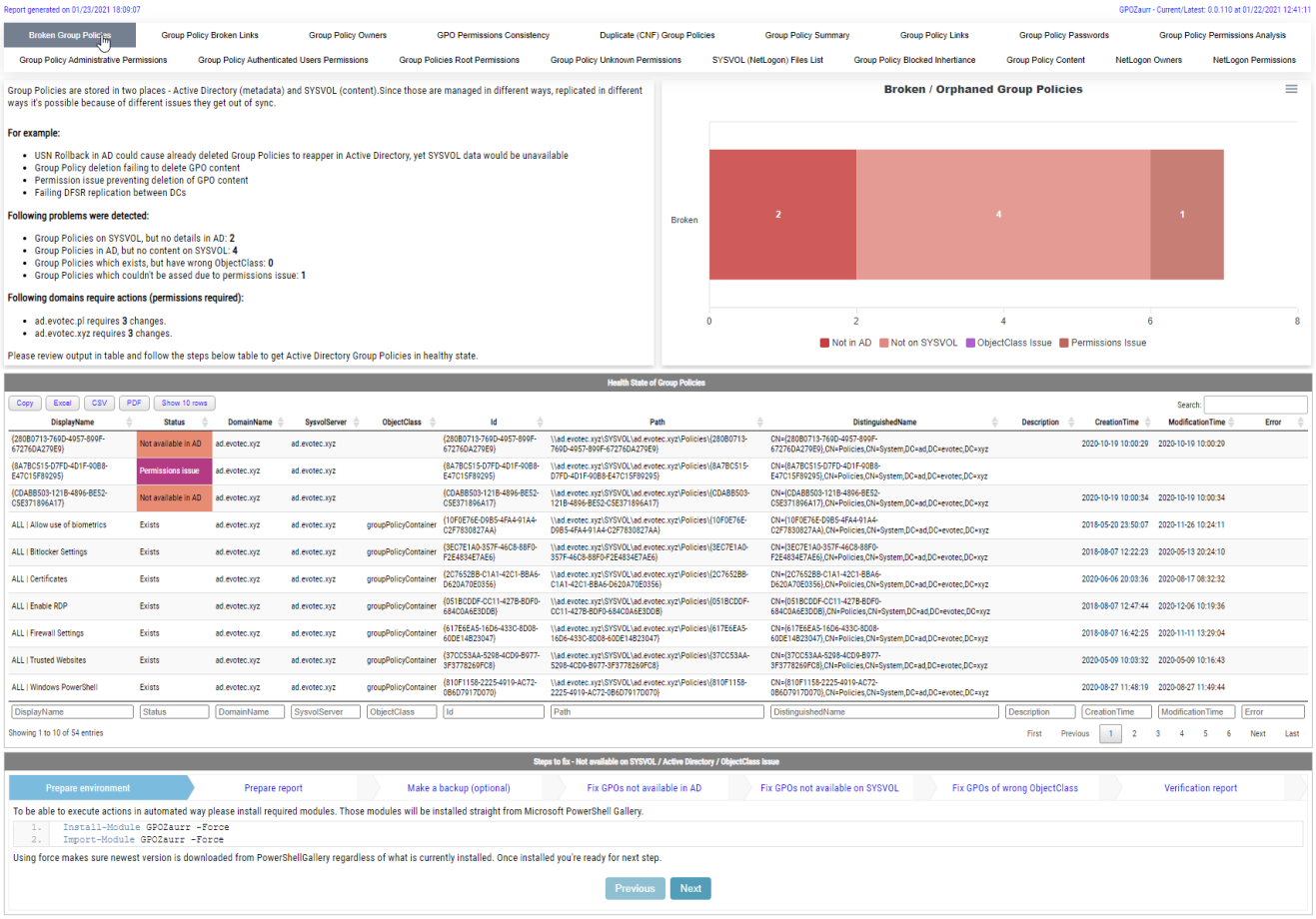
Invoke-GPOZaurr - One command that makes a difference
As mentioned before, **Invoke-GPOZaurr** follows a similar pattern to what **Invoke-Testimo** does. When run without any parameters, it will go thru all available reports one by one to deliver a full-scope scan. Keep in mind that running this cmdlet without any parameters is fine for small domains, but it will take hours to complete for larger domains. For the domain of 5000 GPOs, some reports can take even 2 hours to complete.



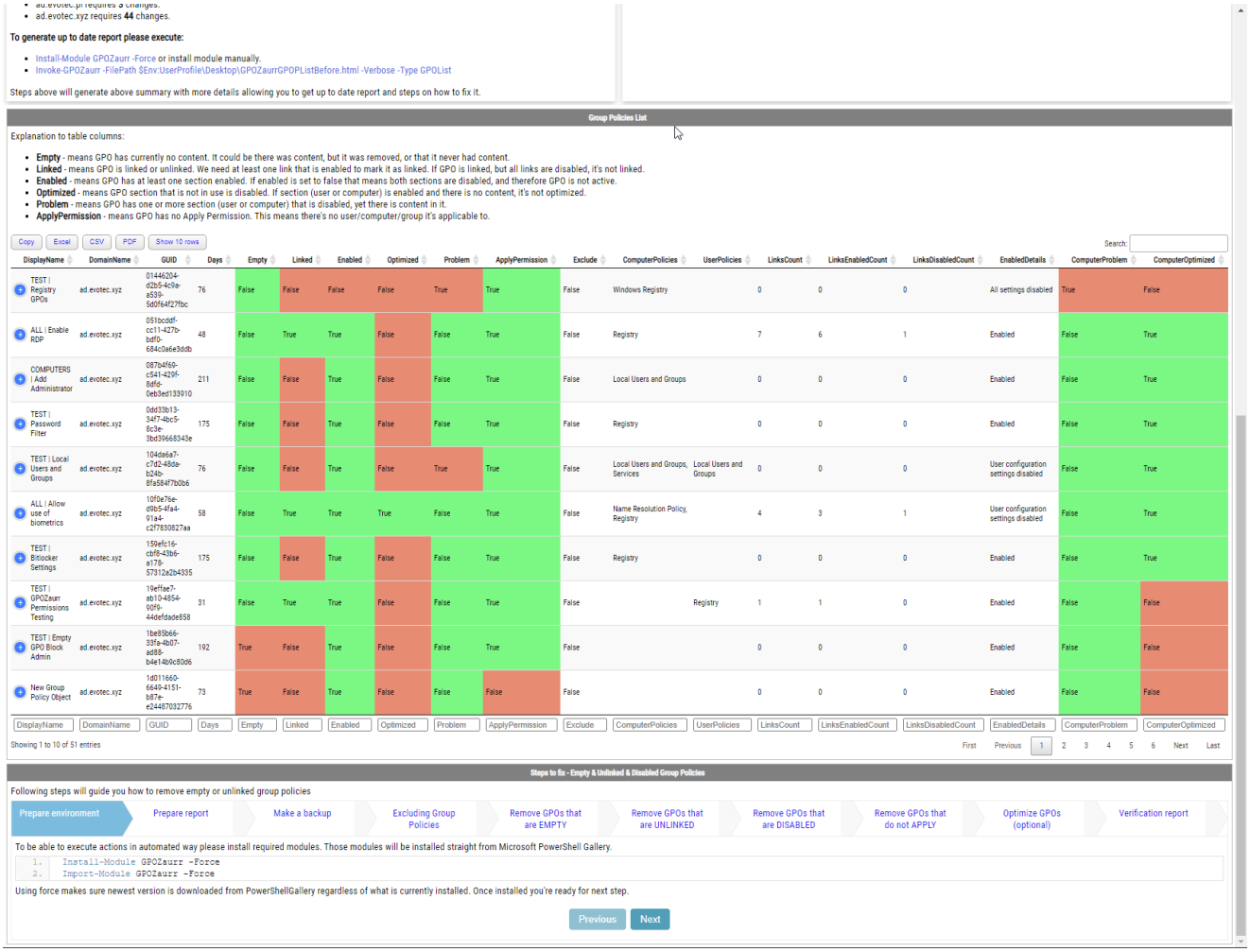
When run, it will display a short information about what it is currently doing and which report is being generated. If you have a large domain and things take time, you may want to use **Invoke-GPOZaurr** with **Verbose** parameter to get additional information.



Once the cmdlet is complete HTML report will open up automatically.



As you can see on the screenshot above, multiple reports were created, each on a different tab. The design of the report is mostly the same. There is information about what the report detected and why it did so on the report's top left. It also gives you a summary of your whole forest and where the issues are found. In the top right corner, I've added a small chart that visualizes the current status. Some charts will show only problems. Some will show multiple statues – all depending on the type of report getting generated. There is usually one, but sometimes more tables with displayed information depending on the problem in the second section. Tables are color-coded to visualize better what is bad or to distinguish multiple problems within the same report. Tables also allow you to export data to Excel, CSV or PDF.



Finally, the last section contains the solution to the problem described. It usually provides step by step instructions on fixing the problem if you choose to fix it. Most of the time, solutions are automated to the point where a single line of code can fix an issue. For example, **delete all empty GPOs**, **delete all unlinked GPOs**, and so on. One command, zero effort.

Invoke-GPOZaurr - Available reports

Currently, **Invoke-GPOZaurr** has few built-in reports. Some of them are more advanced, some of them are for review only. Here's the full list for today. Not everything is 100% finished. Some will require some updates soon as I get more time and feedback. Feel free to report issues/improve those reports with more information.

- **GPOBroken** – this report can detect GPOs that are broken. By broken GPOs, I mean those which exist in AD but have no SYSVOL content or vice versa – have SYSVOL content, but there's no AD metadata. Additionally, it's able to detect GPO objects that are no longer GroupPolicy object. – Then, it provides an easy way to fix it using given step-by-step instructions.
- **GPOBrokenLink** – this report can detect links that have no matching GPO. For example, if a GPO is deleted, sometimes links to that GPO are not properly removed. This command can detect that and propose a solution.

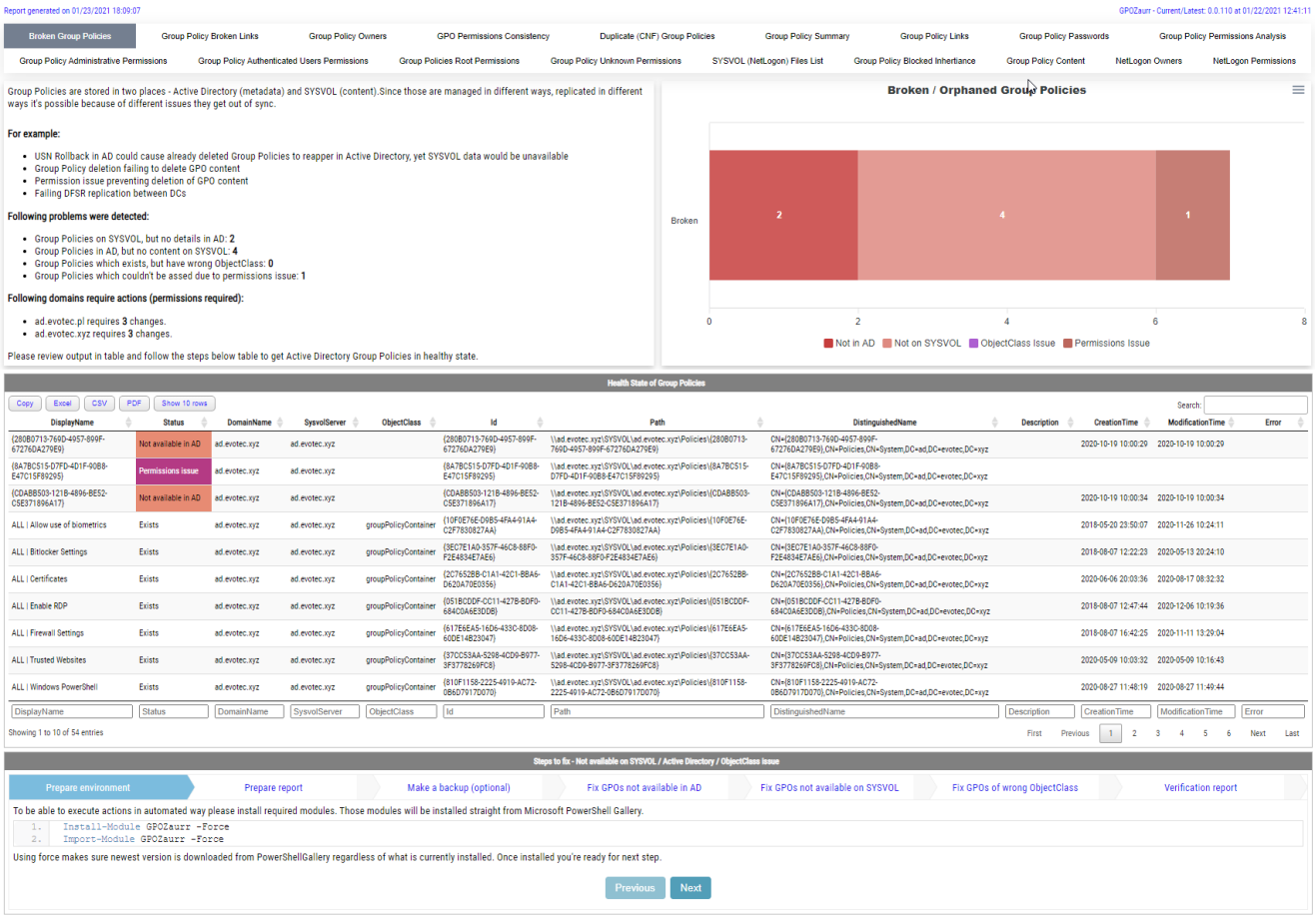
- **GPOOwners** – this report focuses on GPO Owners. By design, if Domain Admin creates GPO, the owner of GPO is the domain admins group. This report detects GPOs that are not owned by Domain Admins (in both SYSVOL and AD) and provides a way to fix them.
- **GPOConsistency** – this report detects inconsistent permissions between Active Directory and SYSVOL, verifying that files/folders inside each GPO match permissions as required. It then provides you an option to fix it.
- **GPODuplicates** – this report detects GPOs that are CNF, otherwise known as duplicate AD Objects, and provides a way to remove them.
- **GPOList** – this report summarizes all group policies focusing on detecting Empty, Unlinked, Disabled, No Apply Permissions GPOs. It also can detect GPOs that are not optimized or have potential problems (disabled section, but still settings in it)
- **GPOLinks** – this report summarizes links showing where the GPO is linked, whether it's linked to any site, cross-domain, or the status of links.
- **GPOPassword** – this report should detect passwords stored in GPOs.
- **GPOPermissions** – this report provides full permissions overview for all GPOs. It detects GPOs missing read permissions for Authenticated Users, GPOs that miss Domain Admins, Enterprise Admins, or SYSTEM permissions. It also detects GPOs that have Unknown permissions available. Finally, it allows you to fix permissions for all those GPOs easily. It's basically a one-stop for all permission needs.
- **GPOPermissionsAdministrative** – this report focuses only on detecting missing Domain Admins, Enterprise Admins permissions and allows you to fix those in no time.
- **GPOPermissionsRead** – similar to an administrative report, but this one focuses on Authenticated Users missing their permissions.
- **GPOPermissionsRoot** – this report shows all permissions assigned to the root of the group policy container. It allows you to verify who can manage all GPOs quickly.
- **GPOPermissionsUnknown** – this report focuses on detecting unknown permissions (deleted users) and allows you to remove them painlessly.
- **GPOFiles** – this report lists all files in the SYSVOL folder (including hidden ones) and tries to make a decent guess whether the file placement based on extension/type makes sense or requires additional verification. This was written to find potential malware or legacy files that can be safely deleted.
- **GPOBlockedInheritance** – this report checks for all Organizational Units with blocked inheritance and verifies the number of users or computers affected.
- **GPOAnalysis** – this report reads all content of group policies and puts them into 70+ categories. It can show things like GPOs that do Drive Mapping, Bitlocker, Laps, Printers, etc. It's handy to find dead settings, dead hosts, or settings that no longer make sense.
- **NetLogonOwners** – this report focuses on detecting NetLogon Owners and a way to fix it to default, secure values.
- **NetLogonPermissions** – this report provides an overview and assessment of all permissions on the NetLogon share.

- **SysVolLegacyFiles** – this report detects SYSVOL Legacy Files (.adm) files

Invoke-GPOZaurr - Report GPOBroken

Group Policies are stored in two places – Active Directory (metadata) and SYSVOL (content). Since those are managed in different ways, replicated in different ways, it's possible because of different issues, and they get out of sync.

```
Invoke-GPOZaurr -Type GPOBroken
```

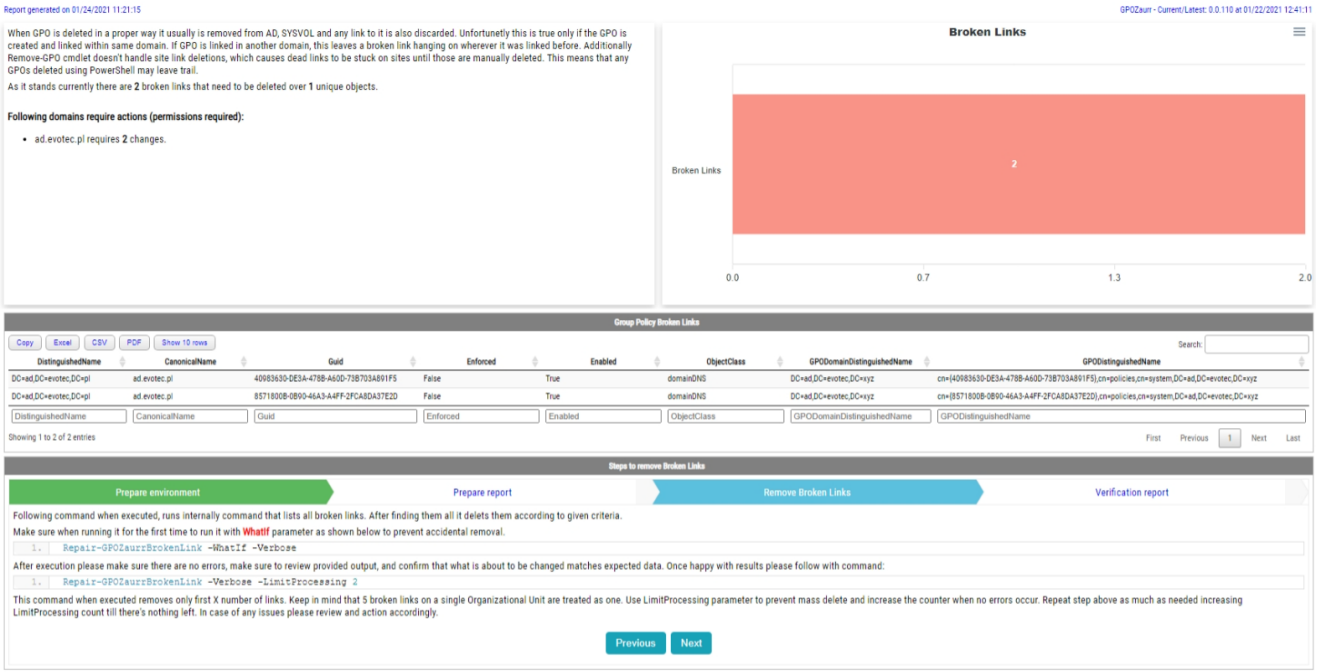


With just a few simple steps, you can have that fixed in a couple of minutes. Keep in mind that you need to have healthy replication of group policies for this to work and not report false positives. If you have unhealthy replication and wrong, DC will get asked about those issues you could potentially remove legitimate content.

Invoke-GPOZaurr - Report GPOBrokenLink

When GPO is deleted correctly, it usually is removed from AD, SYSVOL, and any link to it is also discarded. Unfortunately, this is true only if the GPO is created and linked within the same domain. If GPO is linked in another domain, this leaves a broken link hanging on before it was linked. Additionally, the Remove-GPO cmdlet doesn't handle site link deletions, which causes dead links to be stuck on sites until those are manually deleted. This means that any GPOs deleted using PowerShell may leave a trail.

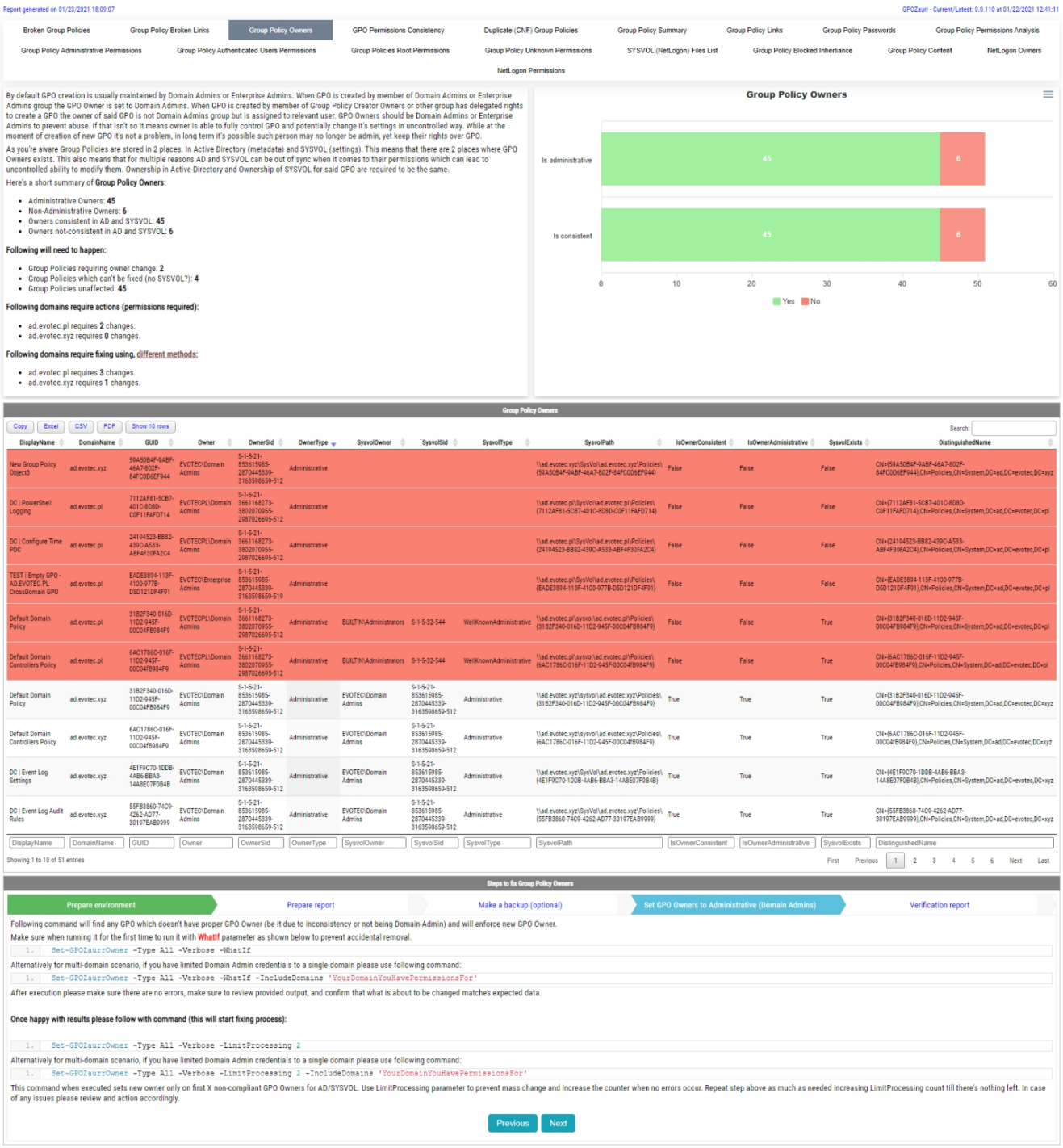
```
Invoke-GPOZaurr -Type GPOBrokenLink
```



Invoke-GPOZaurr - Report GPOOwners

By default, GPO creation is usually maintained by Domain Admins or Enterprise Admins. When GPO is created by Domain Admins or Enterprise Admins group members, the GPO Owner is set to Domain Admins. When GPO is created by a member of Group Policy Creator Owners or other group has delegated rights to create a GPO, the owner of said GPO is not Domain Admins group but is assigned to the relevant user. GPO Owners should be Domain Admins or Enterprise Admins to prevent abuse. If that isn't so, it means the owner can fully control GPO and potentially change its settings in an uncontrolled way. While at the moment of creation of new GPO, it's not a problem, in the long term, it's possible such a person may no longer be admin, yet keep their rights over GPO. As your aware, Group Policies are stored in 2 places. In Active Directory (metadata) and SYSVOL (settings). This means that there are 2 places where GPO Owners exists. This also means that for multiple reasons, AD and SYSVOL can be out of sync when it comes to their permissions, which can lead to uncontrolled ability to modify them. Ownership in Active Directory and Ownership of SYSVOL for said GPO is required to be the same.

```
Invoke-GPOZaurr -Type GPOOwners
```

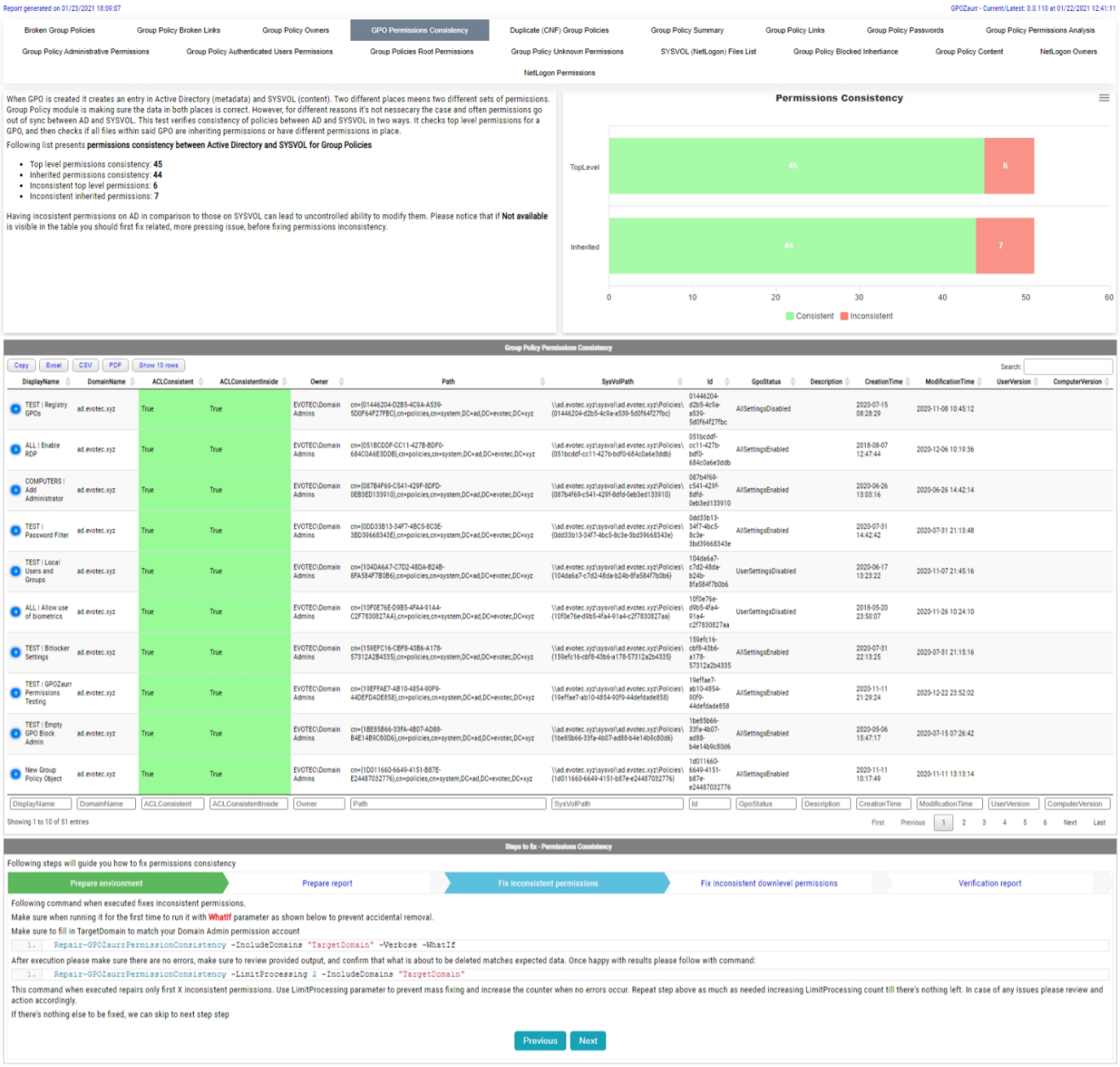


This report is fairly complete with detection and automated fix.

Invoke-GPOZaurr - Report GPOConsistency

When GPO is created, it creates an entry in Active Directory (metadata) and SYSVOL (content). Two different places mean two different sets of permissions. The group Policy module is making sure the data in both places is correct. However, it's not necessarily the case for different reasons, and often permissions go out of sync between AD and SYSVOL. This test verifies the consistency of policies between AD and SYSVOL in two ways. It checks top-level permissions for a GPO and then checks if all files within said GPO is inheriting permissions or have different permissions in place.

```
Invoke-GPOZaurr -Type GPOConsistency
```

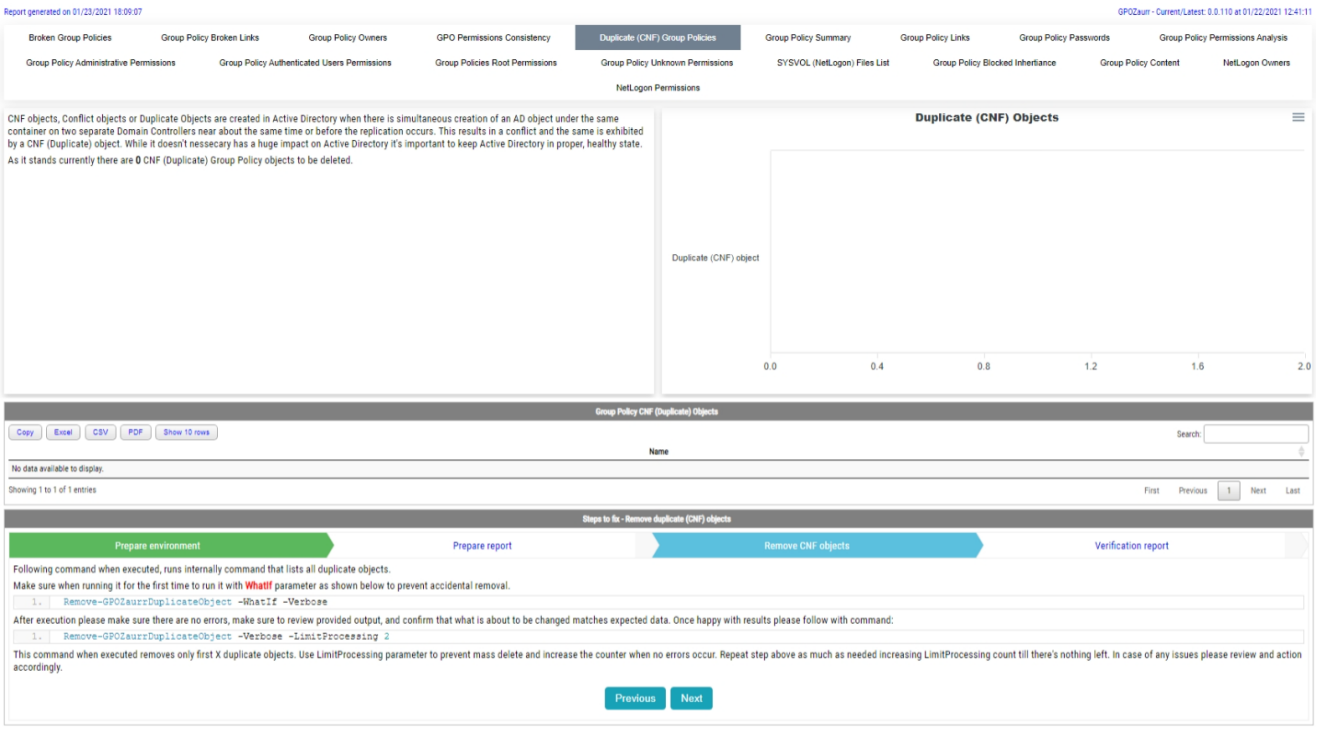


This report is fairly complete and with an automated fix.

Invoke-GPOZaurr - Report GPODuplicates

CNF objects, Conflict objects, or Duplicate Objects are created in Active Directory when there is simultaneous creation of an AD object under the same container on two separate Domain Controllers near about the same time or before the replication occurs. This results in a conflict and a CNF (Duplicate) object exhibits the same. While it doesn't necessarily have a huge impact on Active Directory, it's important to keep Active Directory in a proper, healthy state.

```
Invoke-GPOZaurr -Type GPODuplicates
```



This report is fairly complete and with an automated fix. Be advised above screenshot doesn't show any detected problems because it's pretty hard to generate duplicated objects on-demand, so my test environment doesn't have any. But it does detect those.

Invoke-GPOZaurr - Report GPOList

Over time Administrators add more and more group policies as business requirements change. Due to neglect or thinking it may serve its purpose, later on, many Group Policies often have no value at all. Either the Group Policy is not linked to anything and stays unlinked forever, or GPO is linked, but the link (links) are disabled, or GPO is totally disabled. Then there are Group Policies that are targetting certain groups or persons, and that group is removed, leaving Group Policy doing nothing. Additionally, sometimes new GPO is created without any settings, or the settings are removed over time, but GPO stays in place.

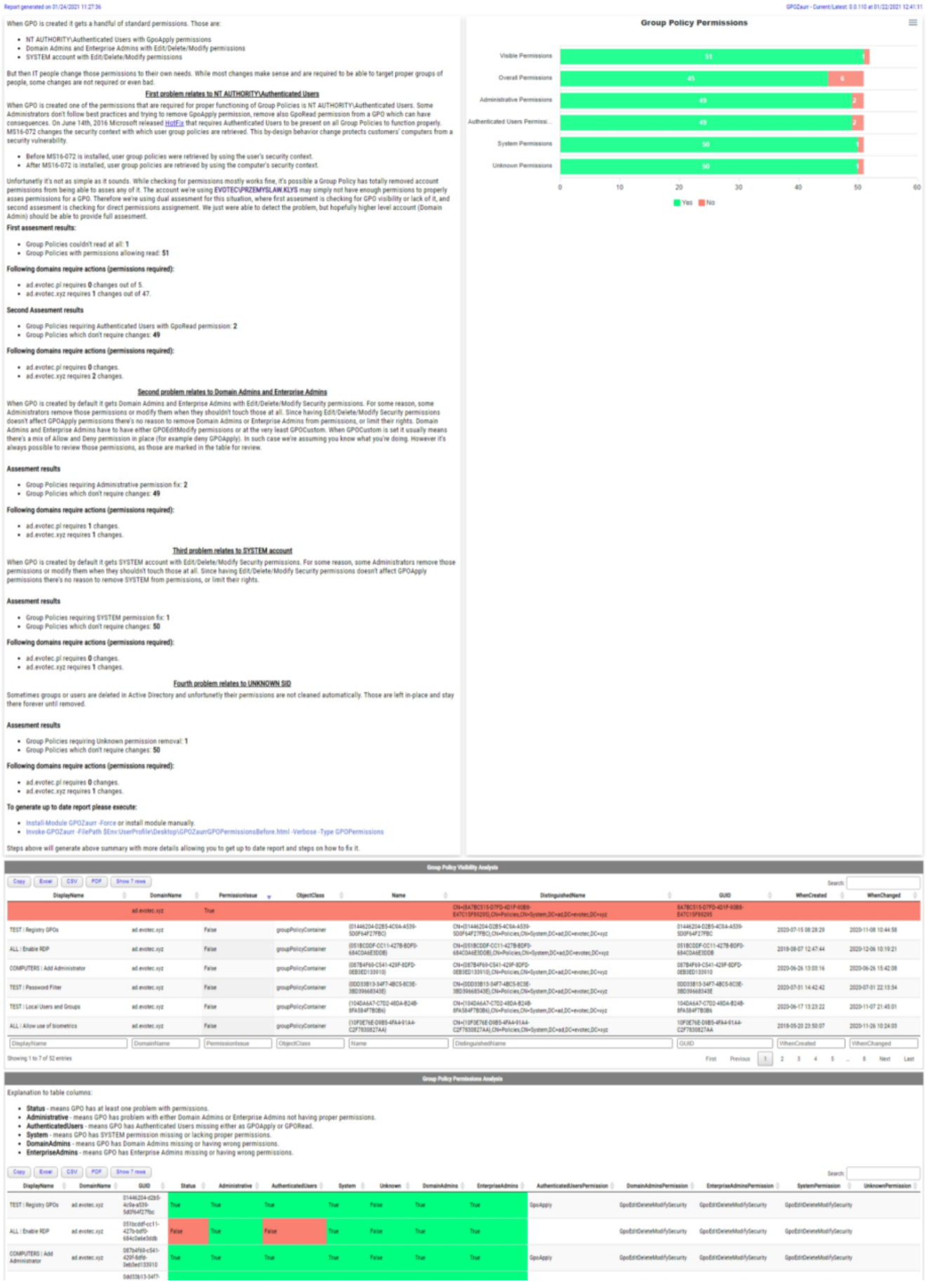
```
Invoke-GPOZaurr -Type GPOList
```

This report is fairly complete and provides automated fixes for most issues detected.

Invoke-GPOZaurr - Report GPOPermissions

The following report contains a full overview of all permissions around Group Policies. It detects 4 different problems (lack of authenticated users, wrong permissions for Domain Admins and Enterprise Admins, Unknown permissions, and lack of proper permission for SYSTEM account). It also contains all permissions, so it's easy to review all permissions from a single place. For each problem, automation is developed, so it's fairly easy to fix any issues as long as you agree with what's proposed.

```
Invoke-GPOZaurr -Type GPOPermissions
```



Report generated on 01/24/2021 11:27:36

GPOZaurr - Current/Latest: 0.0.110 at 01/22/2021 12:41:11

When GPO is created it gets a handful of standard permissions. Those are:

- NT AUTHORITY\Authenticated Users with GpoApply permissions
- Domain Admins and Enterprise Admins with Edit/Delete/Modify permissions
- SYSTEM account with Edit/Delete/Modify permissions

But then IT people change those permissions to their own needs. While most changes make sense and are required to be able to target proper groups of people, some changes are not required or even bad.

First problem relates to NT AUTHORITY\Authenticated Users

When GPO is created one of the permissions that are required for proper functioning of Group Policies is NT AUTHORITY\Authenticated Users. Some Administrators don't follow best practices and trying to remove GpoApply permission, remove also GpoRead permission from a GPO which can have consequences. On June 14th, 2016 Microsoft released HotFix that requires Authenticated Users to be present on all Group Policies to function properly. MS16-072 changes the security context with which user group policies are retrieved. This by-design behavior change protects customers' computers from a security vulnerability.

- Before MS16-072 is installed, user group policies were retrieved by using the user's security context.
- After MS16-072 is installed, user group policies are retrieved by using the computer's security context.

Unfortunetly it's not as simple as it sounds. While checking for permissions mostly works fine, it's possible a Group Policy has totally removed account permissions from being able to asses any of it. The account we're using **EVOTEC\PRZEMYSLAW.KLYS** may simply not have enough permisions to properly asses permissions for a GPO. Therefore we're using dual assesment for this situation, where first assesment is checking for GPO visibility or lack of it, and second assesment is checking for direct permissions assignement. We just were able to detect the problem, but hopefully higher level account (Domain Admin) should be able to provide full assesment.

First assesment results:

- Group Policies couldn't read at all: **1**
- Group Policies with permissions allowing read: **51**

Following domains require actions (permissions required):

- ad.evotec.pl requires **0** changes out of 5.
- ad.evotec.xyz requires **1** changes out of 47.

Second Assesment results

- Group Policies requiring Authenticated Users with GpoRead permission: **2**
- Group Policies which don't require changes: **49**

Following domains require actions (permissions required):

- ad.evotec.pl requires **0** changes.
- ad.evotec.xyz requires **2** changes.

Second problem relates to Domain Admins and Enterprise Admins

When GPO is created by default it gets Domain Admins and Enterprise Admins with Edit/Delete/Modify Security permissions. For some reason, some Administrators remove those permissions or modify them when they shouldn't touch those at all. Since having Edit/Delete/Modify Security permissions doesn't affect GPOApply permissions there's no reason to remove Domain Admins or Enterprise Admins from permissions, or limit their rights. Domain Admins and Enterprise Admins have to have either GPOEditModify permissions or at the very least GPOCustom. When GPOCustom is set it usually means there's a mix of Allow and Deny permission in place (for example deny GPOApply). In such case we're assuming you know what you're doing. However it's always possible to review those permissions, as those are marked in the table for review.

Assesment results

- Group Policies requiring Administrative permission fix: **2**
- Group Policies which don't require changes: **49**

Following domains require actions (permissions required):

- ad.evotec.pl requires **1** changes.
- ad.evotec.xyz requires **1** changes.

Third problem relates to SYSTEM account

When GPO is created by default it gets SYSTEM account with Edit/Delete/Modify Security permissions. For some reason, some Administrators remove those permissions or modify them when they shouldn't touch those at all. Since having Edit/Delete/Modify Security permissions doesn't affect GPOApply permissions there's no reason to remove SYSTEM from permissions, or limit their rights.

Assesment results

- Group Policies requiring SYSTEM permission fix: **1**
- Group Policies which don't require changes: **50**

Following domains require actions (permissions required):

- ad.evotec.pl requires **0** changes.
- ad.evotec.xyz requires **1** changes.

Fourth problem relates to UNKNOWN SID

Sometimes groups or users are deleted in Active Directory and unfortunetly their permissions are not cleaned automatically. Those are left in-place and stay there forever until removed.

Assesment results

- Group Policies requiring Unknown permission removal: **1**
- Group Policies which don't require changes: **50**

Following domains require actions (permissions required):

- ad.evotec.pl requires **0** changes.
- ad.evotec.xyz requires **1** changes.

To generate up to date report please execute:

- Install-Module GPOZaurr -Force or install module manually.
- Invoke-GPOZaurr -FilePath $Env:UserProfile\Desktop\GPOZaurrGPOPermissionsBefore.html -Verbose -Type GPOPermissions

Steps above will generate above summary with more details allowing you to get up to date report and steps on how to fix it.

Group Policy Visibility Analysis

Copy | Excel | CSV | PDF | Show 7 rows | Search:

DisplayName	DomainName	PermissionIssue	ObjectClass	Name	DistinguishedName	GUID	WhenCreated	WhenChanged
	ad.evotec.xyz	True			CN={8A7BC515-07FD-4D1F-92B8-B47C15F89295},CN=Policies,CN=System,DC=ad,DC=evotec,DC=xyz	{8A7BC515-07FD-4D1F-92B8-B47C15F89295}		
TEST | Registry GPOs	ad.evotec.xyz	False	groupPolicyContainer	{01446204-02B5-4C9A-A539-5D0F64F27FBC}	CN={01446204-02B5-4C9A-A539-5D0F64F27FBC},CN=Policies,CN=System,DC=ad,DC=evotec,DC=xyz	{01446204-02B5-4C9A-A539-5D0F64F27FBC}	2020-07-15 08:28:29	2020-11-08 10:44:58
ALL | Enable RDP	ad.evotec.xyz	False	groupPolicyContainer	{051BCDDF-CC11-427B-BDFD-684C0A6E3DDB}	CN={051BCDDF-CC11-427B-BDFD-684C0A6E3DDB},CN=Policies,CN=System,DC=ad,DC=evotec,DC=xyz	{051BCDDF-CC11-427B-BDFD-684C0A6E3DDB}	2018-08-07 12:47:44	2020-12-06 10:19:21
COMPUTERS | Add Administrator	ad.evotec.xyz	False	groupPolicyContainer	{0E7B4F69-C541-429F-8DFD-0EB3ED133910}	CN={0E7B4F69-C541-429F-8DFD-0EB3ED133910},CN=Policies,CN=System,DC=ad,DC=evotec,DC=xyz	{0E7B4F69-C541-429F-8DFD-0EB3ED133910}	2020-06-26 13:03:16	2020-06-26 15:42:08
TEST | Password Filter	ad.evotec.xyz	False	groupPolicyContainer	{0DD33B13-34F7-4BC5-8C3E-3BD39668343E}	CN={0DD33B13-34F7-4BC5-8C3E-3BD39668343E},CN=Policies,CN=System,DC=ad,DC=evotec,DC=xyz	{0DD33B13-34F7-4BC5-8C3E-3BD39668343E}	2020-07-01 14:42:42	2020-07-01 22:13:34
TEST | Local Users and Groups	ad.evotec.xyz	False	groupPolicyContainer	{104DA6A7-C7D2-48DA-B24B-8FA584F7B0B6}	CN={104DA6A7-C7D2-48DA-B24B-8FA584F7B0B6},CN=Policies,CN=System,DC=ad,DC=evotec,DC=xyz	{104DA6A7-C7D2-48DA-B24B-8FA584F7B0B6}	2020-06-17 13:23:22	2020-11-07 21:45:01
ALL | Allow use of biometrics	ad.evotec.xyz	False	groupPolicyContainer	{10F0E76E-D0B5-4FA4-91A4-C2F7830827AA}	CN={10F0E76E-D0B5-4FA4-91A4-C2F7830827AA},CN=Policies,CN=System,DC=ad,DC=evotec,DC=xyz	{10F0E76E-D0B5-4FA4-91A4-C2F7830827AA}	2018-05-20 23:50:07	2020-11-26 10:24:03

Showing 1 to 7 of 52 entries

First Previous 1 2 3 4 5 … 8 Next Last

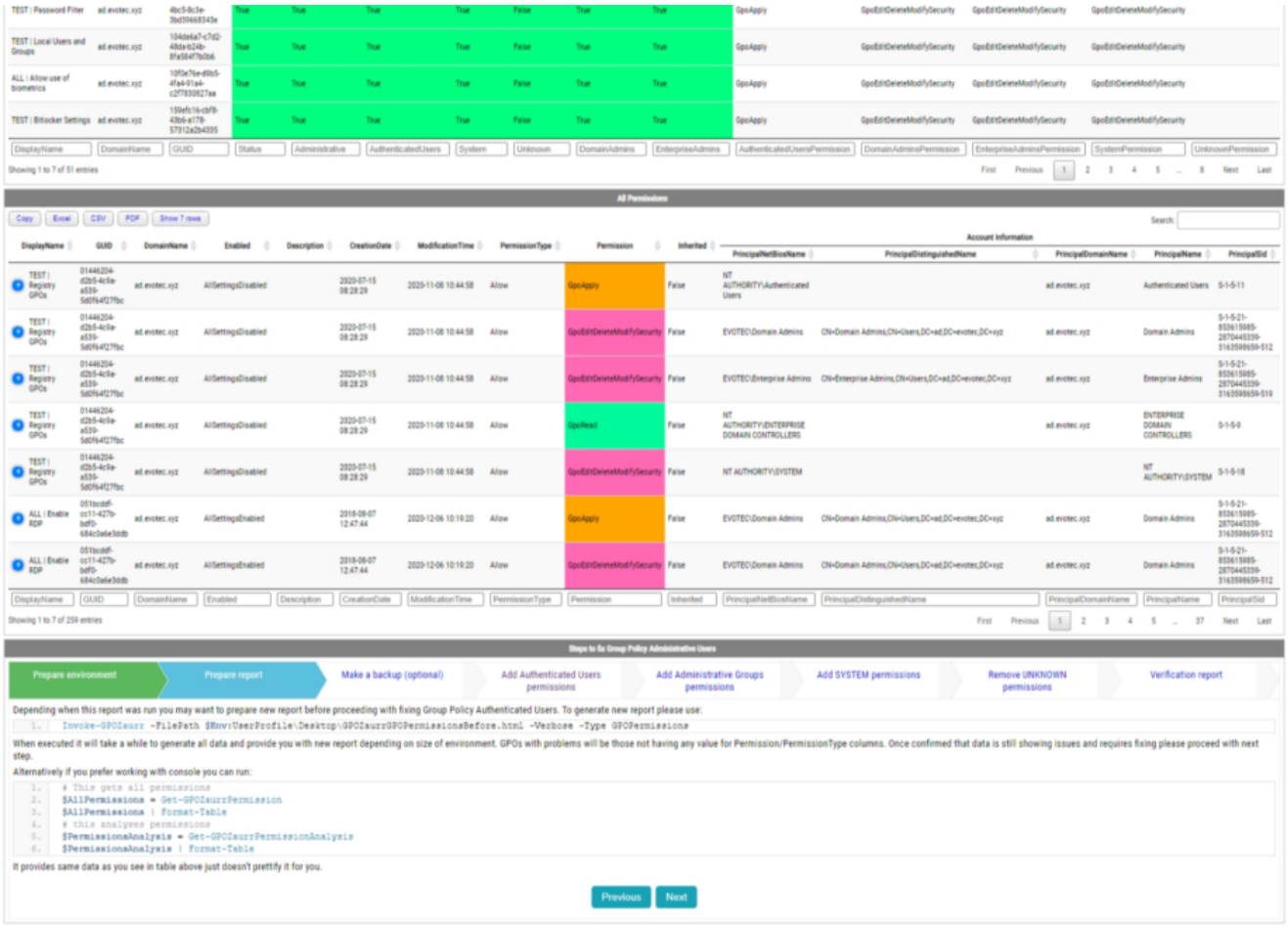
Group Policy Permissions Analysis

Explanation to table columns:

- **Status** - means GPO has at least one problem with permissions.
- **Administrative** - means GPO has problem with either Domain Admins or Enterprise Admins not having proper permissions.
- **AuthenticatedUsers** - means GPO has Authenticated Users missing either as GPOApply or GPORead.
- **System** - means GPO has SYSTEM permission missing or lacking proper permissions.
- **DomainAdmins** - means GPO has Domain Admins missing or having wrong permissions.
- **EnterpriseAdmins** - means GPO has Enterprise Admins missing or having wrong permissions.

Copy | Excel | CSV | PDF | Show 7 rows | Search:

DisplayName	DomainName	GUID	Status	Administrative	AuthenticatedUsers	System	Unknown	DomainAdmins	EnterpriseAdmins	AuthenticatedUsersPermission	DomainAdminsPermission	EnterpriseAdminsPermission	SystemPermission	UnknownPermission
TEST | Registry GPOs	ad.evotec.xyz	01446204-02b5-4c9a-a539-5d0f64f27fbc	True	True	True	True	False	True	True	GpoApply	GpoEditDeleteModifySecurity	GpoEditDeleteModifySecurity	GpoEditDeleteModifySecurity	
ALL | Enable RDP	ad.evotec.xyz	051bcddf-cc11-427b-bdf0-684c0a6e3ddb	False	True	False	True	False	True	True		GpoEditDeleteModifySecurity	GpoEditDeleteModifySecurity	GpoEditDeleteModifySecurity	
COMPUTERS | Add Administrator	ad.evotec.xyz	0E7b4f69-c541-429f-8dfd-0eb3ed133910	True	True	True	True	False	True	True	GpoApply	GpoEditDeleteModifySecurity	GpoEditDeleteModifySecurity	GpoEditDeleteModifySecurity	
		0dd33b13-34f7-												



This report is interactive, meaning clicking on a GPO in one table limits permissions shown in another table. **GPOPermissions** type is kind of ultimate way for you to deal with permissions. I've made one report that covers what 3 different reports were covering before.

```
Invoke-GPOZaurr -Type
GPOPermissionsRead,GPOPermissionsAdministrative,GPOPermissionsUnknown
```

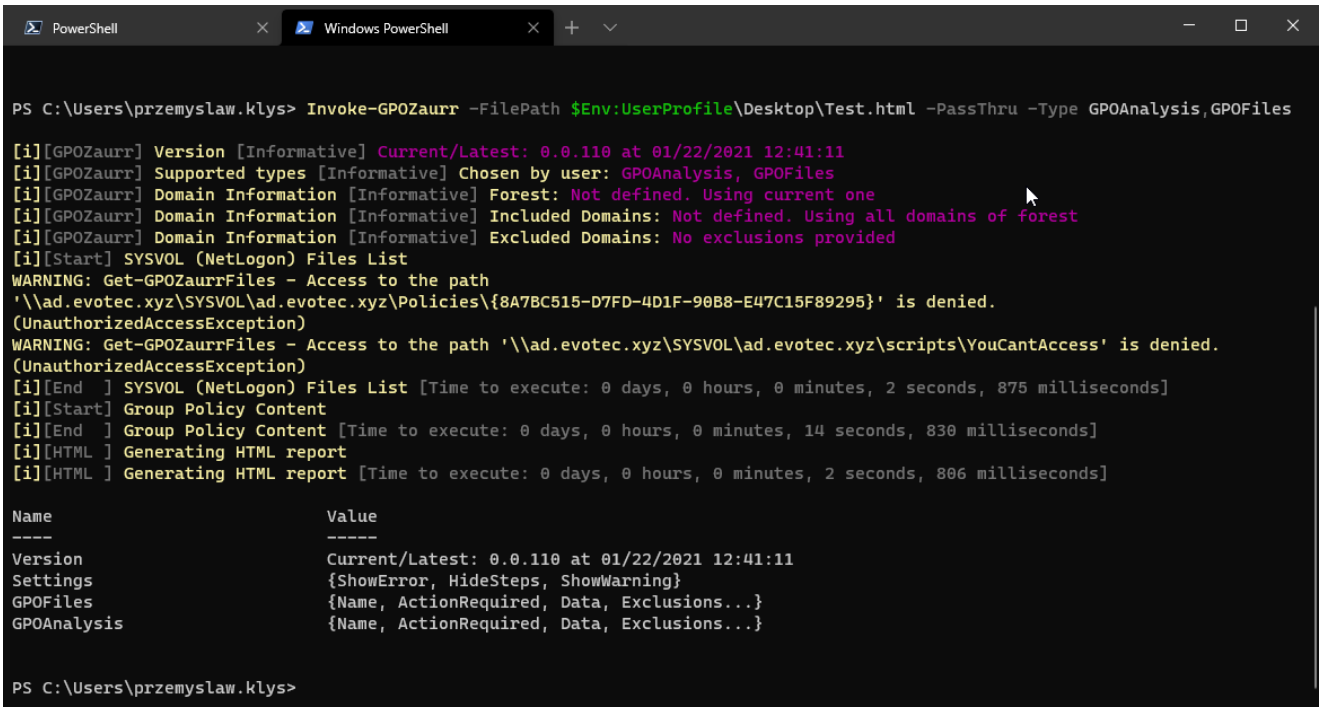
So while you can use the cmdlet above with each type separately – it's easier to use one.

Invoke-GPOZaurr - advanced usage

Invoke-GPOZaurr is basically a wrapper of around 20 or so different GPO cmdlets that I have developed over a period of six months. I was worried that with so many cmdlets being available in my module and my laziness in the documentation, I thought **Invoke-GPOZaurr's** three-step approach (Describe Problem, Provide Data, Offer Solution) was an experiment that I believe will help me manage my GPOs efficiently for years to come. Not everything is completed, but at the current state, it's good enough for release. It allows you to understand where you stand without spending days, weeks, or months of analysis depending on how big your Active Directory is. Of course, this one little command has few more options that allow for different customization options.

```
Invoke-GPOZaurr [[-Type] <string[]>] [[-ExcludeGroupPolicies] <scriptblock>] [-
FilePath <string>] [-PassThru] [-HideHTML] [-HideSteps] [-ShowError] [-ShowWarning]
[-Forest <string>] [-ExcludeDomains <string[]>] [-IncludeDomains <string[]>]
[<CommonParameters>]
```

Using a **Type** parameter, you can ask for one or multiple types. Providing **FilePath** parameter, you can tell **GPOZaurr** where to save created **HTML** file. PassThru, on the other hand, is useful to have **HTML** generated and get the output of the reports back to you for future analysis.



It's also possible to hide steps to fix a given problem. This can be useful if you're doing an overview for your Client/Management and don't want to show how to fix it.

```
Invoke-GPOZaurr -FilePath $Env:UserProfile\Desktop\Test.html -Type GPOBroken -
HideSteps
```

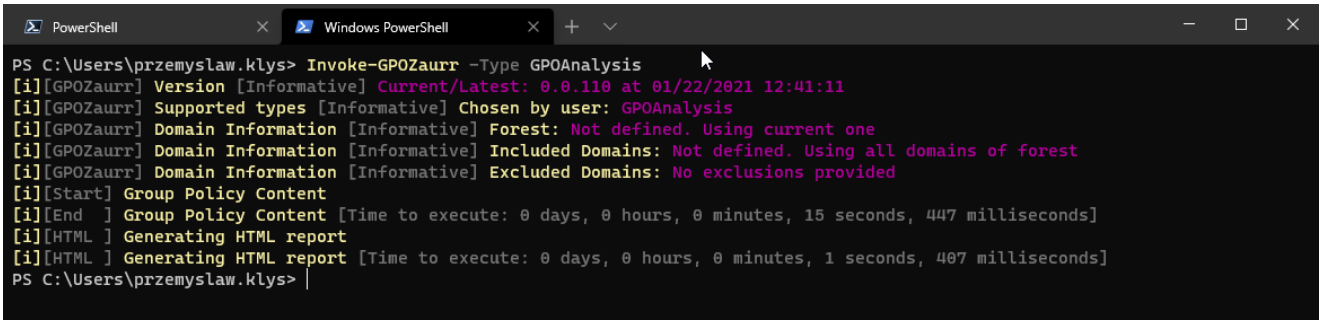
Using **HideHTML** parameter prevents auto-opening of HTML. It's useful for automation purposes.

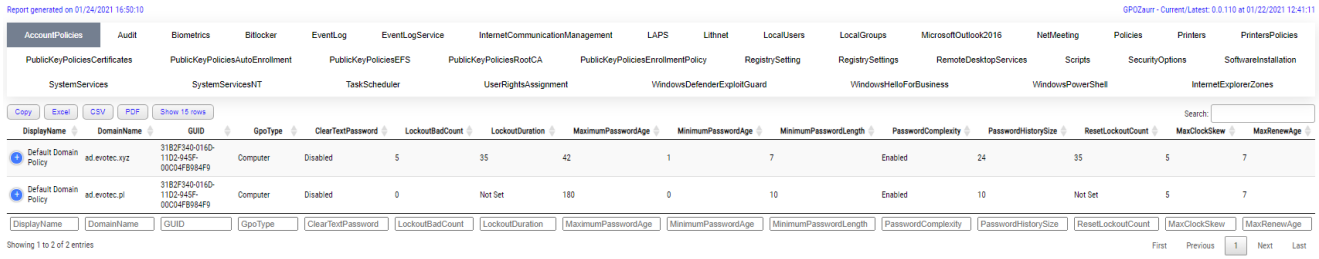
```
Invoke-GPOZaurr -FilePath $Env:UserProfile\Desktop\Test.html -Type GPOBroken -
HideSteps -HideHTML
```

Invoke-GPOZaurr - Type GPOAnalysis

GPO Analysis report is one of the coolest ones I've made. It's able to provide a lot of smaller reports that show the content of group policies. Each report is a separate tab. Using **GPO GUI**, you would normally show you similar output, but this one does it globally. If you've ever tried to find all GPOs that map drives, find ones that have script execution – it's the way to go.

```
Invoke-GPOZaurr -Type GPOAnalysis
```





Report generated on 01/24/2021 16:50:10

GPOZaurr - Current/Latest: 0.0.110 at 01/22/2021 12:41:11

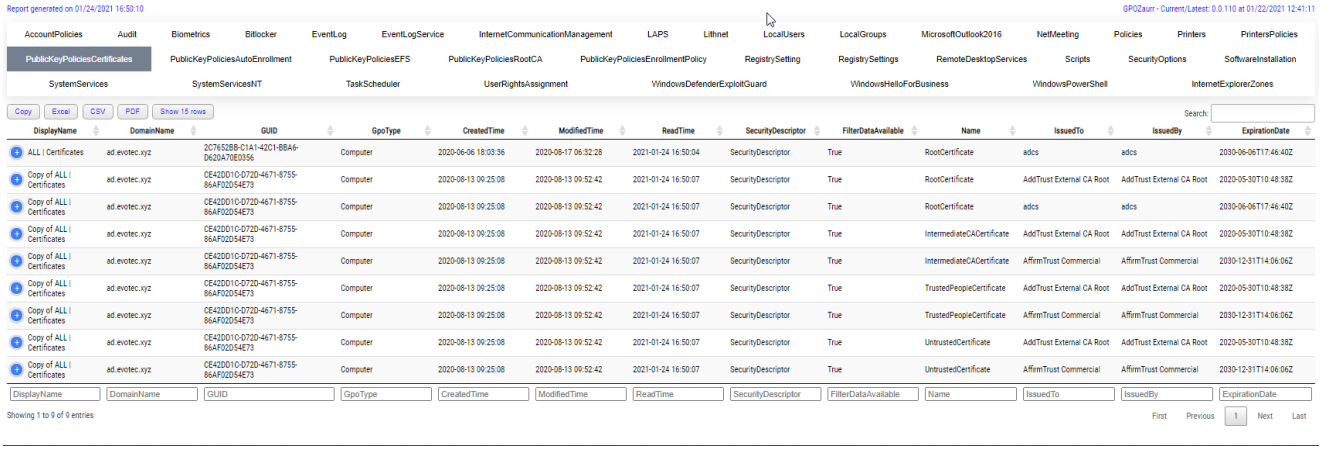
AccountPolicies | Audit | Biometrics | Bitlocker | EventLog | EventLogService | InternetCommunicationManagement | LAPS | Lithnet | LocalUsers | LocalGroups | MicrosoftOutlook2016 | NetMeeting | Policies | Printers | PrintersPolicies | PublicKeyPoliciesCertificates | PublicKeyPoliciesAutoEnrollment | PublicKeyPoliciesEFS | PublicKeyPoliciesRootCA | PublicKeyPoliciesEnrollmentPolicy | RegistrySetting | RegistrySettings | RemoteDesktopServices | Scripts | SecurityOptions | SoftwareInstallation | SystemServices | SystemServicesNT | TaskScheduler | UserRightsAssignment | WindowsDefenderExploitGuard | WindowsHelloForBusiness | WindowsPowerShell | InternetExplorerZones

Copy | Excel | CSV | PDF | Show 15 rows | Search:

DisplayName	DomainName	GUID	GpoType	ClearTextPassword	LockoutBadCount	LockoutDuration	MaximumPasswordAge	MinimumPasswordAge	MinimumPasswordLength	PasswordComplexity	PasswordHistorySize	ResetLockoutCount	MaxClockSkew	MaxRenewAge
Default Domain Policy	ad.evotec.xyz	31B2F340-016D-11D2-945F-00C04FB984F9	Computer	Disabled	5	35	42	1	7	Enabled	24	35	5	7
Default Domain Policy	ad.evotec.pl	31B2F340-016D-11D2-945F-00C04FB984F9	Computer	Disabled	0	Not Set	180	0	10	Enabled	10	Not Set	5	7

Showing 1 to 2 of 2 entries

First | Previous | 1 | Next | Last



Report generated on 01/24/2021 16:50:10

GPOZaurr - Current/Latest: 0.0.110 at 01/22/2021 12:41:11

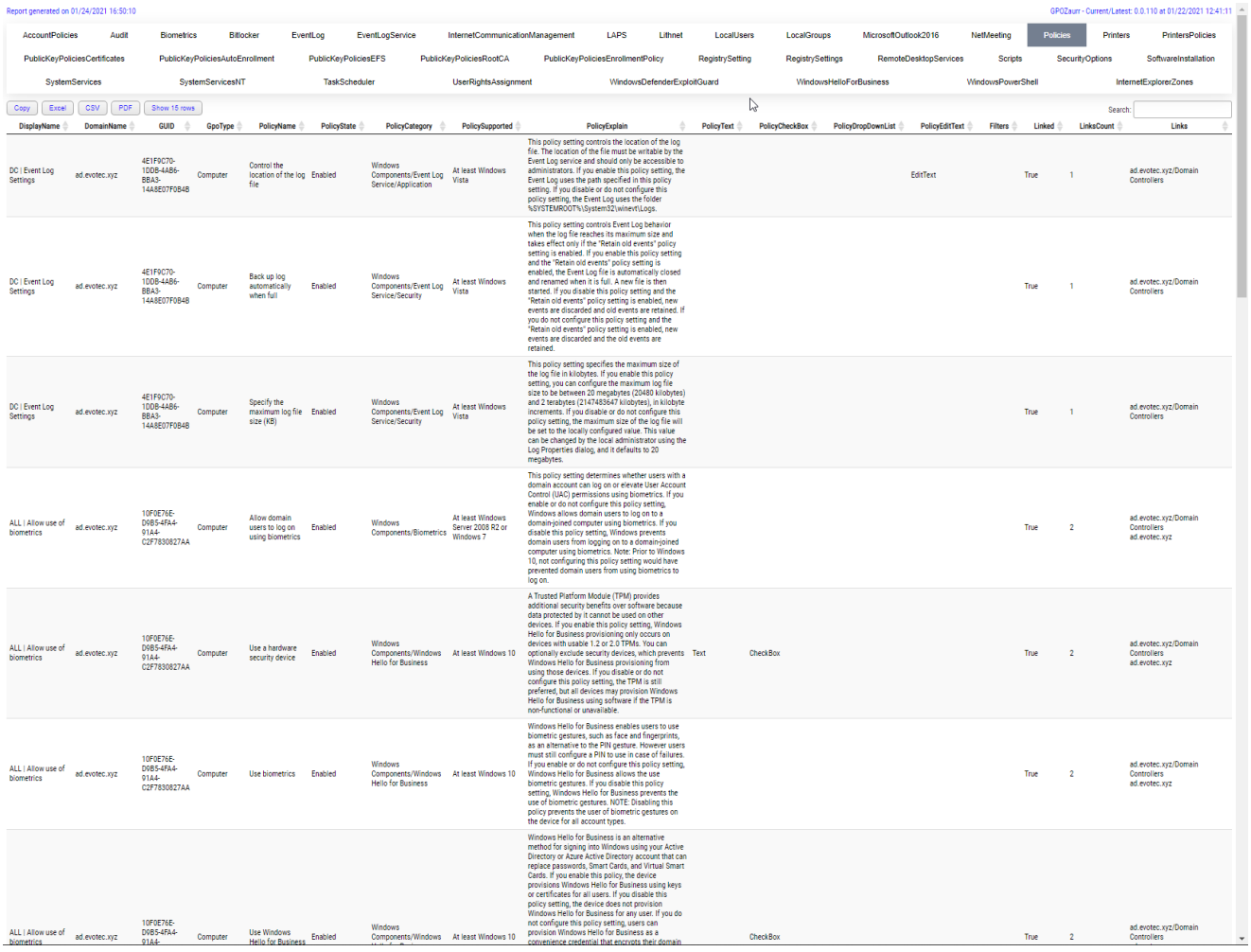
AccountPolicies | Audit | Biometrics | Bitlocker | EventLog | EventLogService | InternetCommunicationManagement | LAPS | Lithnet | LocalUsers | LocalGroups | MicrosoftOutlook2016 | NetMeeting | Policies | Printers | PrintersPolicies | PublicKeyPoliciesCertificates | PublicKeyPoliciesAutoEnrollment | PublicKeyPoliciesEFS | PublicKeyPoliciesRootCA | PublicKeyPoliciesEnrollmentPolicy | RegistrySetting | RegistrySettings | RemoteDesktopServices | Scripts | SecurityOptions | SoftwareInstallation | SystemServices | SystemServicesNT | TaskScheduler | UserRightsAssignment | WindowsDefenderExploitGuard | WindowsHelloForBusiness | WindowsPowerShell | InternetExplorerZones

Copy | Excel | CSV | PDF | Show 15 rows | Search:

DisplayName	DomainName	GUID	GpoType	CreatedTime	ModifiedTime	ReadTime	SecurityDescriptor	FilterDataAvailable	Name	IssuedTo	IssuedBy	ExpirationDate
ALL | Certificates	ad.evotec.xyz	207652BB-C1A1-42C1-BBA6-D620A70E0356	Computer	2020-06-06 18:03:36	2020-08-17 06:32:28	2021-01-24 16:50:04	SecurityDescriptor	True	RootCertificate	adcs	adcs	2030-06-06T17:46:40Z
Copy of ALL | Certificates	ad.evotec.xyz	CE42DD1C-D72D-4671-8755-86AF02D54E73	Computer	2020-08-13 09:25:08	2020-08-13 09:52:42	2021-01-24 16:50:07	SecurityDescriptor	True	RootCertificate	AddTrust External CA Root	AddTrust External CA Root	2020-05-30T10:48:38Z
Copy of ALL | Certificates	ad.evotec.xyz	CE42DD1C-D72D-4671-8755-86AF02D54E73	Computer	2020-08-13 09:25:08	2020-08-13 09:52:42	2021-01-24 16:50:07	SecurityDescriptor	True	RootCertificate	adcs	adcs	2030-06-06T17:46:40Z
Copy of ALL | Certificates	ad.evotec.xyz	CE42DD1C-D72D-4671-8755-86AF02D54E73	Computer	2020-08-13 09:25:08	2020-08-13 09:52:42	2021-01-24 16:50:07	SecurityDescriptor	True	IntermediateCACertificate	AddTrust External CA Root	AddTrust External CA Root	2020-05-30T10:48:38Z
Copy of ALL | Certificates	ad.evotec.xyz	CE42DD1C-D72D-4671-8755-86AF02D54E73	Computer	2020-08-13 09:25:08	2020-08-13 09:52:42	2021-01-24 16:50:07	SecurityDescriptor	True	IntermediateCACertificate	AffirmTrust Commercial	AffirmTrust Commercial	2030-12-31T14:06:06Z
Copy of ALL | Certificates	ad.evotec.xyz	CE42DD1C-D72D-4671-8755-86AF02D54E73	Computer	2020-08-13 09:25:08	2020-08-13 09:52:42	2021-01-24 16:50:07	SecurityDescriptor	True	TrustedPeopleCertificate	AddTrust External CA Root	AddTrust External CA Root	2020-05-30T10:48:38Z
Copy of ALL | Certificates	ad.evotec.xyz	CE42DD1C-D72D-4671-8755-86AF02D54E73	Computer	2020-08-13 09:25:08	2020-08-13 09:52:42	2021-01-24 16:50:07	SecurityDescriptor	True	TrustedPeopleCertificate	AffirmTrust Commercial	AffirmTrust Commercial	2030-12-31T14:06:06Z
Copy of ALL | Certificates	ad.evotec.xyz	CE42DD1C-D72D-4671-8755-86AF02D54E73	Computer	2020-08-13 09:25:08	2020-08-13 09:52:42	2021-01-24 16:50:07	SecurityDescriptor	True	UntrustedCertificate	AddTrust External CA Root	AddTrust External CA Root	2020-05-30T10:48:38Z
Copy of ALL | Certificates	ad.evotec.xyz	CE42DD1C-D72D-4671-8755-86AF02D54E73	Computer	2020-08-13 09:25:08	2020-08-13 09:52:42	2021-01-24 16:50:07	SecurityDescriptor	True	UntrustedCertificate	AffirmTrust Commercial	AffirmTrust Commercial	2030-12-31T14:06:06Z

Showing 1 to 9 of 9 entries

First | Previous | 1 | Next | Last



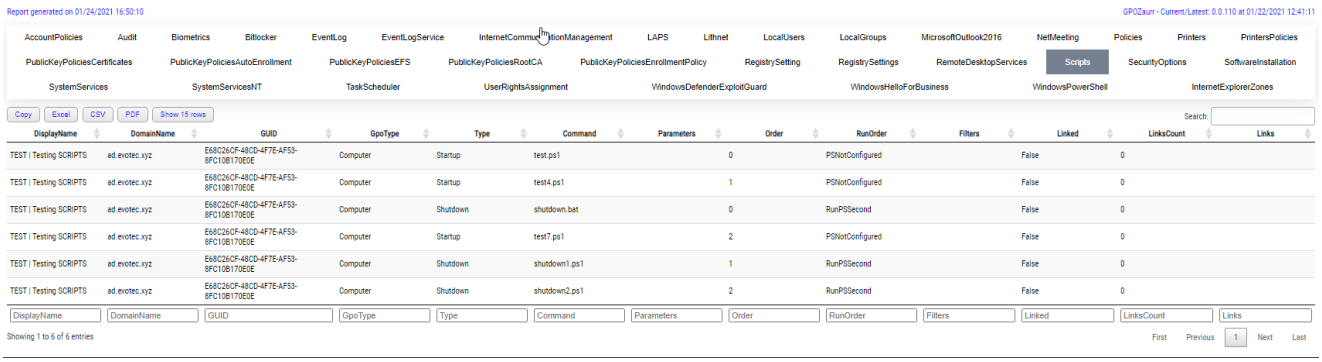
Report generated on 01/24/2021 16:50:10

GPOZaurr - Current/Latest: 0.0.110 at 01/22/2021 12:41:11

AccountPolicies | Audit | Biometrics | Bitlocker | EventLog | EventLogService | InternetCommunicationManagement | LAPS | Lithnet | LocalUsers | LocalGroups | MicrosoftOutlook2016 | NetMeeting | Policies | Printers | PrintersPolicies | PublicKeyPoliciesCertificates | PublicKeyPoliciesAutoEnrollment | PublicKeyPoliciesEFS | PublicKeyPoliciesRootCA | PublicKeyPoliciesEnrollmentPolicy | RegistrySetting | RegistrySettings | RemoteDesktopServices | Scripts | SecurityOptions | SoftwareInstallation | SystemServices | SystemServicesNT | TaskScheduler | UserRightsAssignment | WindowsDefenderExploitGuard | WindowsHelloForBusiness | WindowsPowerShell | InternetExplorerZones

Copy | Excel | CSV | PDF | Show 15 rows | Search:

DisplayName	DomainName	GUID	GpoType	PolicyName	PolicyState	PolicyCategory	PolicySupported	PolicyExplain	PolicyText	PolicyCheckBox	PolicyDropDownList	PolicyEditText	Filters	Linked	LinksCount	Links
DC | Event Log Settings	ad.evotec.xyz	4E1F9C70-1DDB-4AB6-BBA3-14A8E07F0B4B	Computer	Control the location of the log file	Enabled	Windows Components/Event Log Service/Application	At least Windows Vista	This policy setting controls the location of the log file. The location of the file must be writable by the Event Log service and should only be accessible to administrators. If you enable this policy setting, the Event Log uses the path specified in this policy setting. If you disable or do not configure this policy setting, the Event Log uses the folder %SYSTEMROOT%\System32\winevt\Logs.				EditText		True	1	ad.evotec.xyz/Domain Controllers
DC | Event Log Settings	ad.evotec.xyz	4E1F9C70-1DDB-4AB6-BBA3-14A8E07F0B4B	Computer	Back up log automatically when full	Enabled	Windows Components/Event Log Service/Security	At least Windows Vista	This policy setting controls Event Log behavior when the log file reaches its maximum size and takes effect only if the "Retain old events" policy setting is enabled. If you enable this policy setting and the "Retain old events" policy setting is enabled, the Event Log file is automatically closed and renamed when it is full. A new file is then started. If you disable this policy setting and the "Retain old events" policy setting is enabled, new events are discarded and old events are retained. If you do not configure this policy setting and the "Retain old events" policy setting is enabled, new events are discarded and the old events are retained.						True	1	ad.evotec.xyz/Domain Controllers
DC | Event Log Settings	ad.evotec.xyz	4E1F9C70-1DDB-4AB6-BBA3-14A8E07F0B4B	Computer	Specify the maximum log file size (KB)	Enabled	Windows Components/Event Log Service/Security	At least Windows Vista	This policy setting specifies the maximum size of the log file in kilobytes. If you enable this policy setting, you can configure the maximum log file size to be between 20 megabytes (20480 kilobytes) and 2 terabytes (2147483647 kilobytes), in kilobyte increments. If you disable or do not configure this policy setting, the maximum size of the log file will be set to the locally configured value. This value can be changed by the local administrator using the Log Properties dialog, and it defaults to 20 megabytes.						True	1	ad.evotec.xyz/Domain Controllers
ALL | Allow use of biometrics	ad.evotec.xyz	10F0E76E-D9B5-4FA4-91A4-C2F7830827AA	Computer	Allow domain users to log on using biometrics	Enabled	Windows Components/Biometrics	At least Windows Server 2008 R2 or Windows 7	This policy setting determines whether users with a domain account can log on or elevate User Account Control (UAC) permissions using biometrics. If you enable or do not configure this policy setting, Windows allows domain users to log on to a domain-joined computer using biometrics. If you disable this policy setting, Windows prevents domain users from logging on to a domain-joined computer using biometrics. Note: Prior to Windows 10, not configuring this policy setting would have prevented domain users from using biometrics to log on.						True	2	ad.evotec.xyz/Domain Controllers ad.evotec.xyz
ALL | Allow use of biometrics	ad.evotec.xyz	10F0E76E-D9B5-4FA4-91A4-C2F7830827AA	Computer	Use a hardware security device	Enabled	Windows Components/Windows Hello for Business	At least Windows 10	A Trusted Platform Module (TPM) provides additional security benefits over software because data protected by it cannot be used on other devices. If you enable this policy setting, Windows Hello for Business provisioning only occurs on devices with usable 1.2 or 2.0 TPMs. You can optionally exclude security devices, which prevents Windows Hello for Business provisioning from using those devices. If you disable or do not configure this policy setting, the TPM is still preferred, but all devices may provision Windows Hello for Business using software if the TPM is non-functional or unavailable.	Text	CheckBox				True	2	ad.evotec.xyz/Domain Controllers ad.evotec.xyz
ALL | Allow use of biometrics	ad.evotec.xyz	10F0E76E-D9B5-4FA4-91A4-C2F7830827AA	Computer	Use biometrics	Enabled	Windows Components/Windows Hello for Business	At least Windows 10	Windows Hello for Business enables users to use biometric gestures, such as face and fingerprints, as an alternative to the PIN gesture. However users must still configure a PIN to use in case of failures. If you enable or do not configure this policy setting, Windows Hello for Business allows the use biometric gestures. If you disable this policy setting, Windows Hello for Business prevents the use of biometric gestures. NOTE: Disabling this policy prevents the user of biometric gestures on the device for all account types.						True	2	ad.evotec.xyz/Domain Controllers ad.evotec.xyz
ALL | Allow use of biometrics	ad.evotec.xyz	10F0E76E-D9B5-4FA4-	Computer	Use Windows Hello for Business	Enabled	Windows Components/Windows	At least Windows 10	Windows Hello for Business is an alternative method for signing into Windows using your Active Directory or Azure Active Directory account that can replace passwords, Smart Cards, and Virtual Smart Cards. If you enable this policy, the device provisions Windows Hello for Business using keys or certificates for all users. If you disable this policy setting, the device does not provision Windows Hello for Business for any user. If you do not configure this policy setting, users can provision Windows Hello for Business as a convenience credential that encrypts their domain		CheckBox				True	2	ad.evotec.xyz/Domain Controllers



The idea for every report is that each setting is stored per each line. This sometimes means that if the setting has a potential of 50 options, the report will generate 50 columns. I've not found an easy way to make it readable without custom creating and every report. While I do that for some of the reports, some are totally autogenerated. If you feel something is not covered or require a better report, open up an issue, and we can see what can be done.

Invoke-GPOZaurr - Automating GPOZaurr to Email

Since I want to keep my group policies healthy at all times, I've developed small automation. This automation deals with one report and sends an email to a ticketing system if there is a problem or sends an update to the AD team that everything is great. This automation uses PSWriteHTML (which is also used to generate **HTML** anyway). I've developed the module where the description on each report is available to use outside of **GPOZaurr** (that's where the **PassThru** parameter is useful).

```
Import-Module GPOZaurr -Force

$PasswordSecureString = 'passwordSecureString'

$Types = @(
    @{
        Name    = 'GPOOwners'
        Path    = "$PSScriptRoot\Reports\GPOOwners_$(Get-Date -f yyyy-MM-
dd_HHmmss).html"
        Subject = '[AD Compliance] Group Policy Owners Issue'
        Ticket  = '[Ticket#2001000](https://linkToChangeRequest)'
        Attach  = $true
    }
    @{
        Name    = 'GPODuplicates'
        Path    = "$PSScriptRoot\Reports\GPODuplicates_$(Get-Date -f yyyy-MM-
dd_HHmmss).html"
        Subject = '[AD Compliance] Group Policy Duplicate (Conflicting) Objects
Detected'
        Ticket  = '[Ticket#2001000](https://linkToChangeRequest)'
        Attach  = $true
    }
    @{
        Name    = 'NetLogonOwners'
        Path    = "$PSScriptRoot\Reports\NetLogonPermissions_$(Get-Date -f yyyy-MM-
dd_HHmmss).html"
        Subject = '[AD Compliance] NetLogon Owners Issue'
        Ticket  = '[Ticket#2001000](https://linkToChangeRequest)'
        Attach  = $true
    }
    @{
        Name    = 'GPOConsistency'
        Path    = "$PSScriptRoot\Reports\GPOConsistency_$(Get-Date -f yyyy-MM-
dd_HHmmss).html"
        Subject = '[AD Compliance] Group Policy Consistency'
        Ticket  = '[Ticket#2001000](https://linkToChangeRequest)'
        Attach  = $true
    }
    # Too big
    @{
        Name    = 'GPOPermissions'
        Path    = "$PSScriptRoot\Reports\GPOPermissions_$(Get-Date -f yyyy-MM-
dd_HHmmss).html"
        Subject = '[AD Compliance] Group Policy Permissions Analysis'
        Ticket  = '[Ticket#2001000](https://linkToChangeRequest)'
        Attach  = $false
    }
    @{
        Name    = 'GPOList'
        Path    = "$PSScriptRoot\Reports\GPOList_$(Get-Date -f yyyy-MM-
dd_HHmmss).html"
        Subject = '[AD Compliance] Group Policy Empty & Unlinked & Disabled Cleanup'
        Ticket  = '[Ticket#2001000](https://linkToChangeRequest)'
        Attach  = $false
    }
```

```
    @{
        Name    = 'GPOBroken';
        Path    = "$PSScriptRoot\Reports\GPOOrphans_$(Get-Date -f yyyy-MM-
dd_HHmmss).html";
        Subject = '[AD Compliance] Group Policy Orphaned/Broken Cleanup'
        Ticket  = '[Ticket#2001000](https://linkToChangeRequest)'
        Attach  = $true
    }
    @{
        Name    = 'GPOBrokenLink'
        Path    = "$PSScriptRoot\Reports\GPOBrokenLink_$(Get-Date -f yyyy-MM-
dd_HHmmss).html"
        Subject = '[AD Compliance] Group Policy Broken Links'
        Ticket  = '[Ticket#2001000](https://linkToChangeRequest)'
        Attach  = $true
    }
)

foreach ($Type in $Types) {
    $EmailHeaderBadReport = EmailHeader {
        EmailFrom -Address 'EmailFrom@evotec.pl'
        EmailTo -Addresses "przemyslawklys@evotec.pl", 'otherguy@evotec.pl'
        EmailServer -Server 'smtpServer' -SSL -Port 25 -UserName 'login' -Password
$PasswordSecureString -PasswordAsSecure
        EmailOptions -Priority High -DeliveryNotifications Never
        EmailSubject -Subject $Type.Subject
        if ($Type.Attach -eq $true) {
            EmailAttachment -FilePath $Type.Path
        }
    }
    $EmailHeaderGoodReport = EmailHeader {
        EmailFrom -Address 'EmailFrom@evotec.pl'
        EmailTo -Addresses "przemyslawklys@test.pl", 'otherguy@evotec.pl'
        EmailServer -Server 'smtpServer' -SSL -Port 25 -UserName 'login' -Password
$PasswordSecureString -PasswordAsSecure
        EmailOptions -Priority Low -DeliveryNotifications Never
        EmailSubject -Subject $Type.Subject
        if ($Type.Attach -eq $true) {
            EmailAttachment -FilePath $Type.Path
        }
    }
    $ReportOutput = Invoke-GPOZaurr -FilePath $Type.Path -Type $Type.Name -PassThru -
HideHTML -Verbose
    foreach ($Report in $ReportOutput.Keys | Where-Object { $_ -notin 'Version',
'Settings' }) {
        if ($ReportOutput[$Report]['ActionRequired'] -eq $true) {
            Email {
                $EmailHeaderBadReport
                EmailBody {
                    EmailText -Text 'Hello Team,' -LineBreak
                    EmailText -Text "I've found disprepency in our domain that needs
to be fixed and I need your help. " -LineBreak

                    $ReportOutput[$Report]['Summary']
```

```
                    EmailText -LineBreak
                    EmailText -TextBlock {
                        "This automation was approved by CAB in $($Type.Ticket). "
                        "The goal is to keep Group Policies Healthy at all times! "
                        "In case of issues please contact Przemyslaw Klys "
                    } -LineBreak
                    EmailText -Text 'With regards,'
                    EmailText -Text 'Automated Monitoring'

                } -FontSize 10pt
            }
        } else {
            Email {
                $EmailHeaderGoodReport
                EmailBody {
                    EmailText -Text 'Hello Team,' -LineBreak
                    EmailText -Text "I've run the report and everything is looking
great. Nothing to do here, but just wanted to say - great job! " -LineBreak

                    $ReportOutput[$Report]['Summary']

                    EmailText -LineBreak
                    EmailText -TextBlock {
                        "This automation was approved by CAB in $($Type.Ticket). "
                        "The goal is to keep Group Policies Healthy at all times! "
                        "In case of issues please contact Przemyslaw Klys "
                    } -LineBreak
                    EmailText -Text 'With regards,'
                    EmailText -Text 'Automated Monitoring'
                } -FontSize 10pt
            }
        }
    }
}
```

Keep in mind that some of those reports can get really large. For example, the permissions report for 4000 GPOs is about 30MB in size. On the other hand, some other reports are much smaller. This is why there's an option to choose whether to attach a report or not.

Summary

GPOZaurr is a huge module. It contains a lot of reports, and just a handful of those are shown here. It's almost **20000** lines of code. It can deal with all sorts of **GPO**/**SYSVOL**/**NETLOGON** problems you may have. Feel free to explore. On GitHub, the full source code is available (and somewhat readable – one function per file) and about 40 different examples. Not everything may be easy to understand, but I plan to release more blog posts on different ways to deal with issues. What's important to know is that this module will work just fine with just user credentials. Of course, if you've removed authenticated users from a GPO, some reports will skip it, others will mark it as unavailable, but it does work. Of course, fixing issues will require Domain Admin, but that you can do manually – not even running GPOZaurr as Domain Admin.

The code is published on GitHub

Issues should be reported on GitHub

Code is published as a module on PowerShellGallery

The module is signed with a certificate, like any new modules that I create or update.

```
Install-Module GPOZaurr -Force
```

GO Ahead! Have fun! Make sure to report any issues, or if you feel like something would require covering more ground, let me know. My goal is to have **GPOZaurr** as the only way to deal with **Group Policies**.