Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

Interested in learning more?
Check out the list of upcoming events offering "Securing Windows and PowerShell Automation (Security 505)" at http://www.giac.org/registration/gcwn

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.
The Effectiveness of Tools in Detecting the ‘Maleficent Seven’ Privileges in the Windows Environment

GIAC (GCWN) Gold Certification

Author: Tobias Mccurry, tobiasmccurry@gmail.com
Advisor: Sally Vandeven
Accepted: November 2017

Abstract

Windows privileges add to the complexity of Windows user permissions. Each additional user added to a group could lead to a domain compromise if not evaluated. Privileges can override permission causing a gap of perceived effective permission. Currently, system administrators rely on tools such as Security Explorer, Permissions Analyzer for Active Directory, or Gold Finger help with this problem. An analysis of these three tools that are supposed to help with permissions is needed to provide administrators a window into these complex effective permissions. The results of this research discovered a gap in identifying users with privileges with the current tools available. This gap was filled by the author by using powershell.
Introduction

Windows administration can become complex as a company evolves and expands. Nesting can occur within organizational units and groups. Windows also allows for Active Directory objects (organizational units, groups, and users) to be assigned special privileges. Several programs offer help with management of permissions and privileges. This paper will explore how Security Explorer, Permissions Analyzer for Active Directory, and Gold Finger can help with managing permissions and if these programs can find any special Windows privileges. Access to seven specific privileges, covered later, could allow a user to escalate privileges to administrator level. To understand how Windows approaches managing permissions, this paper will provide some background on how permissions or privileges work.

1.1. Permissions/Privileges

Permissions are granted to users to accomplish tasks such as reading files or start a program. “Permissions are also called Access Control Lists” (Fossen, Securing Windows with PowerShell and the Critical Security Controls, 2016). A privilege is an addition to permissions. A conflict between privileges and permissions leads to privileges taking precedence (Microsoft, 2017). Over time, the ideas of Discretionary Access, Mandatory Access, Role-based Access, and Rule-based Access Control were developed (NIST, 2017). Each one of the ideas became incorporated into modern operating systems, frameworks, and later ingrained into standards like PCI and NIST. Understanding each idea in detail will help explain how Windows permissions and privileges were structured.

1.1.1. Discretionary Access Control (DAC)

Every object has an owner; the owner can give access to others, known as Discretionary Access Control. This level of control is like letting someone temporarily live with you and giving them a key. As the owner, you still maintain ownership but have given them permission to use the item. The owner then allows or disallows that permission (DEFENSE, 1983). File and folder access uses this same permission model.
The Operating System (OS) will check to see if the user has been granted permission by the owner.

1.1.2. Mandatory Access Control (MAC)

Each object requires certain permission to be accessed; the user needs that permission to access the object known as Mandatory Access Control. Permission to access the system is defined by an administrator. When a user needs to access it, the Operating System will check to see if the user has the same permission as the object (VINCENT, Kuhn, Xie, & Hwang, 2011). An example of this is similar to a lock (operating system) and key (authenticated user). A lock will unlock with the correct key; only the correct key will be allowed to unlock the lock. In Windows, access to administrative objects is limited, and the User Account Control (UAC) is the way to enforce MAC of processes (Microsoft, 2017).

1.1.3. Role Based Access Control (RBAC)

Role-Based Access Control is based on the idea that as groups of people need to access an object, users are added to that group. Roles become naturally defined as teams are developed to handle tasks or projects. For example, for a role such as developers, an Administrator would create a Developers group, assign the desired permissions to that group and then add user accounts to the group. Role-Based access control allows for flexibility; organizations then have the flexibility to add and remove people to and from roles, worrying only about the permission of the role itself (Hu, Ferraiolo, & Kuhn, 2006).

1.1.4. Rule-Based Access Control

System administrators can define parameters that have to be met for access to be granted also known as rule-based access. The rules are defined by the administrator. An example of this is “allowed to access something during a certain time of the day” or “remote connection from specific IPs” (Gentry, 2012). Some other examples are based on
ports, User-Agent, or data submitted. These types of rules are similar to firewall rules. A certain set of criteria needs to be met for the access to be granted.

1.2. **Windows Approach**

Windows has evolved over the years to accommodate these aforementioned frameworks and ideas. Windows has become a mixture of DAC, MAC, rule-based, and RBAC permissions. Windows split the management of these permissions into organizational units, groups, and users. Managing permissions this way allows for the flexibility to manage groups of systems using different permissions and privileges. Windows Active Directory can be organized using organizational units, groups and users.

1.2.1. **Organization Units (OU)**

In Active Directory, Organizational Units are one of the most general containers in which objects can be placed. Aside from the domain, an OU is the top of the hierarchical structure. An OU can contain other OUs, groups, or users. OUs are the biggest unit to which Group Policy can be applied (Microsoft, 2017).

1.2.2. **Groups**

Groups can exist in the domain, OU, and in other groups. Groups can be either a security or distribution group. Distribution groups only serve the purpose of sending a group of people the same email. Security groups can be used to manage users and computers with the added benefit of being able to use group policies in a particular group (Microsoft, 2000).

1.2.3. **Users**

Users can be a part of an OU or a group. Users are accounts that can log in to the domain. Users can be local or domain. Local users are only allowed to authenticate on the system they are on. Sometimes service accounts are also created as Domain User accounts to perform actions or run a service. Users will typically authenticate with username and password (Microsoft, 2000).
1.2.4. Special privileges

Windows has 26 total special privileges that can be granted in addition to permissions. These privileges are defined in “Winnt.h” or “Windows.h” and embedded in almost every Window’s executable (Microsoft, 2017). Programs can also request to use these privileges on behalf of the user. Users and programs request these privileges to temporarily perform actions. Privileges also add to the complexity of user permission management (Microsoft, 2017).

2. The Maleficent Seven

SANS “Securing Windows and PowerShell Automation” highlights seven of the 26 windows special privileges. The author, Jason Fossen, determined that these privileges could easily lead to a domain compromise (Fossen, Securing Windows with PowerShell and the Critical Security Controls, 2016). Research completed by Foxglove security has also determined nine similar privileges could be used to escalate privileges (Foxglove security, 2017). Microsoft provides little details on each of the privileges on their respective Microsoft Developer Network (MSDN) page, but this research will provide as much details about the privileges as possible. Process Explorer demonstrates all the privileges the anti-virus agent requested in figure 2.
Figure 2: AgentSvc Privileges

Each one of these privileges is needed to ensure the application can perform as intended. The sections below will go into detail about the privileges identified as the maleficent seven.

2.1. **SeImpersonatePrivilege**

Impersonation is one of the most used privileges in Windows. It allows for a user, program, or thread to impersonate a client or specified account. Programs, threads, or services use this privilege to perform tasks that require different permissions. A “user
sends requests and commands over the network to the server; the service uses the Security Access Table (SAT) representing the user to act on behalf of the user” (Fossen, Securing Windows with PowerShell and the Critical Security Controls, 2016). Tools like Incognito use this privilege to impersonate users.

### 2.2. **SeDebugPrivilege**

Debugging is another one of the most commonly requested privileges. Some developers will ask for this privilege for regular programming, not understanding that they do not need it. Users by default can debug their own application. This privilege gives the ability to attach to any program and read. Tools such as Mimikatz (Metcalf, 2016), or reflective DLL injection (Graham, 2016) use this privilege. By default, only the local Administrators group has the Debug Programs privilege (Fossen, Securing Windows with PowerShell and the Critical Security Controls, 2016).

### 2.3. **SeTcbPrivilege**

The SeTcbPrivilege privilege allows the user or process to act as part of the operating system. If a user or process has this privilege, it can request any other privilege and can even create a token that does not have an identity. Access token can be described as “an object that describes the security context of a process or thread. The information in a token includes the identity and privileges of the user account associated with the process or thread” (Microsoft, 2017). A user with this privilege is like a user with a golden ticket (Warren, 2017). The user can create tickets that can mimic other users.

### 2.4. **SeCreateTokenPrivilege**

As a user logs into a computer, Windows builds the Security Access Token (SAT). As the user requests resources, the SAT is checked. With access to the SeCreateTokenPrivilege the ability to create a SAT for Administrator’s groups is easily granted. Foxglove security demonstrated the ability to enable the local administrator
group on a token by iterating over the groups and elevating it from user to admin (Foxglove security, 2017).

2.5. **SeLoadDriverPrivilege**

Users will be able to load and unload device drivers at will with the SeLoadDriverPrivilege. SeLoadDriverPrivilege could be used to load a malicious driver into the operating system. Device drivers run at kernel level allowing them to circumvent any operating restrictions. Under some condition, this privilege with some registry tricks would allow for an unsigned driver to be loaded (Foxglove security, 2017).

2.6. **SeRestorePrivilege**

The SeRestorePrivilege to restore files can be assigned to other groups or users instead of Backup Operators group. With this privilege assigned, a user can bypass any permission to restore a file or directory. The SeRestorePrivilege is the equivalent of giving a user “Traverse folder / Execute file and Write” permissions on all files and directories. Bypassing “NTFS permissions and replace any file” allows for backdooring of files (Fossen, Securing Windows with PowerShell and the Critical Security Controls, 2016).

2.7. **SeTakeOwnershipPrivilege**

A user or process with SeOwnershipPrivilege can take ownership of any Active directory object. Objects in the domain such as files, folders, registry keys, services, processes, and threads can be changed. Even if the owner is denied access, they can still change permissions.

2.8. **Privilege conclusion**

Each one of the privileges listed above could allow a user to escalate privileges. The combined research from the SANS SEC505: Securing Windows and PowerShell
Automation (Fossen, Securing Windows with PowerShell, the Critical Security Controls, 2016) and Foxglove security (Foxglove security, 2017) go into more advanced detail about each of the privileges and how malware could use them. An example of the difference of privileges assigned to a user and an elevated UAC level with that same user is demonstrated below in Figure 3.

![Figure 3: Normal command prompt](image1)

Figure 3: Normal command prompt

Figure 4 demonstrates the privileges available on an elevated command prompt.

![Figure 4: Elevated Command Prompt](image2)

Figure 4: Elevated Command Prompt
The state column as depicted in the graphic above does not mean the privilege cannot be used by the application. The enabled or disabled state indicates if the current process is using that privilege. The changing of the state of privileges was demonstrated earlier with the SeDebugPrivilege with Mimikatz enabling the privilege. The Appendix demonstrates which privileges an attacker can start with and through a UAC escalation, gain access to more privileges.

3. Comparison of the three programs

The testing environment was set up in the following fashion to have an accurate test between Gold Finger, Security Explorer, and Permission Analyzer. A Windows 2012 Domain Controller Server that was on the latest patch level (August 2017) was used as the control. The server had various OUs, Groups, and Users added to ensure a large domain was simulated. The PowerShell scripts that were used to create the OUs, Groups, Users, and privileges have been included in the Appendix. The special Windows privileges were assigned to semi-random groups and users. The virtual machine was copied and the test program was installed on each one.

3.1. Grading System

Each program was graded with seven yes/no questions. The questions should provide readers the ability to determine if the tool will help them or not. The grading, defined below, must include at least “The Maleficent Seven” privileges previously outlined. Additional installation and usage pictures are provided in the Appendix for reference. The list of questions is shown below in Table 1.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is a full Demo available?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Did it detect all the OUs?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Did it detect all the groups in the OUs?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Did it detect all the Users in the group?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Does it present the data in an understandable manner?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Does it detect the windows special privileges?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Does it highlight any privileges?</td>
<td>Yes/No</td>
</tr>
</tbody>
</table>

Table 1: Grading Table

3.2. Gold Finger 6.0

Gold Finger by Paramount Defense is a graphical user program; its main goal is to help with auditing in Windows. Refer to the section on installation in the Appendix for more details on installation and usage. The program is limited on what reports it generates until a trial license or full license is entered. The program can generate 218 different reports that can help system administrators identify problem areas. A complete list of all the auditing questions is provided in the Appendix. According to the website, the program will “Accurately and instantly identify, audit and document all privileged users and effective privileged access in Active Directory” (Paramount Defenses Inc, 2017).

Goldfinger offers multiple different report options that can be exported in CSV or PDF format. The report provides details such as date, time, scope, and findings. In Figure 5 below all security groups in the domain are listed.
The following is a list of all security groups in the domain sans:

### 1. 198

- **Scope:** Domain local
- **Empty:** No
- **Managed By:**
- **Notes:**
- **EmailAddress:**
- **SAM Account Name:** sans>198
- **Security Identifier:** S-1-5-21-88529767-1952201999-4127942948-1221
- **Created On:** 2017-08-24 13:47 hrs Eastern Daylight Time
- **Last Modified On:** 2017-09-11 20:12 hrs Eastern Daylight Time
- **Distinguished Name:** CN=198,OU=Kree Sentries,DC=sans,DC=lab
- **Description:**

---

**Figure 5: Security Groups Report**

**Results**

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is a full Demo available?</td>
<td>Yes (7 days)</td>
</tr>
<tr>
<td>Did it detect all the OUs?</td>
<td>Yes</td>
</tr>
<tr>
<td>Did it detect all the groups in the OUs?</td>
<td>Yes</td>
</tr>
<tr>
<td>Did it detect all the Users in the group?</td>
<td>Yes</td>
</tr>
<tr>
<td>Does it present the data in an understandable manner?</td>
<td>Yes</td>
</tr>
<tr>
<td>Does it detect the windows special privilege?</td>
<td>No</td>
</tr>
<tr>
<td>Does it highlight any privileges?</td>
<td>No</td>
</tr>
</tbody>
</table>

**Table 2: Grading Table for Gold Finger**

Gold Finger provides very detailed reporting on permissions in different ways. The program is comprised of eight tools and each tool has various reports it can run with the ability to export in CSV or PDF. Gold Finger is an excellent tool to identify assigned
Windows permissions, however it does not specifically identify any of “The Maleficent Seven”. Table 2 highlights that Gold Finger helps determine user permissions but missed their privileges.

### 3.3. Security Explorer 9.7.0.477

Security Explorer by Quest is a graphical program that centers around querying and reporting on permissions. The Appendix contains installation instructions for Security Explorer. The program requires Microsoft Access database engine to be installed. Security Explorer has functions that looked at NTFS, share, registry, printer, service, SharePoint, SQL server, and exchange security as well as group and user management. The browse function allows the user to interact and make selections of items such as computers, networks, or scope in the window.

The group and user management reporting functions are very detailed providing name, parent group name, SID, disabled, locked, Domain, Computer, last login, comment, and login count. Security Explorer did eventually create a report, but there were no users on the report. The report in Figure 6 below was generated after selecting some groups.
Figure 6: Security Explorer Report

Security explorer was not able to detect all the OUs or Users in the OUs. The reporting feature provides details such as SID, if the account is disabled, locked, and expired. Table 3 below summaries how Security Explorer faired against the grading system.

Results

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is a full Demo available?</td>
<td>Yes (30 days)</td>
</tr>
<tr>
<td>Did it detect all the OUs?</td>
<td>No</td>
</tr>
<tr>
<td>Did it detect all the groups in the OUs?</td>
<td>Yes</td>
</tr>
<tr>
<td>Did it detect all the Users in the group?</td>
<td>No</td>
</tr>
<tr>
<td>Does it present the data in an understandable manner?</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Security Explorer was more interactive than Gold Finger. Security Explorer had only three functions that focused on user and group permissions. Security Explorer does provide tools to help with different areas of system administration. The reporting did provide some details but not enough to help manage users. Again, this tool does not provide any insight into Windows privileges.

### 3.4. **Permission Analyzer 1.0.0.68**

Permission Analyzer by SolarWinds is a graphical program that is more focused on showing the permissions for server file shares. Installation took only two steps making it the quickest installation out of the three programs. After installation, the program requires authentication to the domain controller to continue. Table 4 highlights the how Permission Analyzer faired against the grading system.

#### Results

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Is a full Demo available?</strong></td>
<td>Yes, full program</td>
</tr>
<tr>
<td><strong>Did it detect all the OUs?</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Did it detect all the groups in the OUs?</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Did it detect all the Users in the group?</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Does it present the data in an understandable manner?</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Does it detect the windows special permissions?</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Does it highlight any privileges?</strong></td>
<td>No</td>
</tr>
</tbody>
</table>

Table 3: Grading Table for Security Explorer
Table 4: Grading Table for Permission Analyzer

Permission Analyzer only has one function, to display folder permissions, but provides links to other programs that can be downloaded. Based on that information it will display the results as follows: append data, change permissions, create directories, create files, delete, delete subdirectories and files, execute file, full control, list directory, modify, read, read and execute, read attributes, read data, read extended attributes, read permissions, synchronize, take ownership, transverse, write, write attributes, write data, and write extended attributes (SolarWinds Worldwide, LLC, 2017). This information does not provide any insight into actual user permissions.

4. Conclusion

4.1. Most Effective/Successful Program

None of the three programs checked “The Maleficent seven privileges” described in this research. Permission Analyzer was the least effective at detecting permissions. Security Explorer expands permission validation by being able to check permissions in various ways. Security Explorer is a useful tool in an administrator’s toolbox. Gold Finger was the only program that scored the highest on the grading system. Gold Finger provided the most detail through its various reporting functions. Although Gold Finger provides an adequate level of detail, it does not handle or examine privileges. All programs were deficient in discovering and listing Windows special privileges. An alternative solution is needed to fill the gap, of finding the maleficent 7 privileges, missed by the tools above.

4.2. Alternative solution

Gold Finger provided the most detailed information about permissions but not of the privileges in Windows. This research confirms there is a gap in security and understanding of privileges over permissions. Windows privileges still need to be
addressed, as this is currently a hidden problem that at least these three tools did not focus on. An alternative solution was developed using PowerShell.

Two PowerShell functions were created to help identify highly privileged users. The work was based on Tony Pombo’s UserRights.ps1 script (MCP, 2017). The “Find-BadPrivileges” function will list all the accounts on the local or remote system with any of the privileges outlined in the paper. The “Find-BadPrivilegesDomain” function relies on the Active Directory commandlet to be installed. Using the commandlet, a list of computers will be extracted from Active Directory. Using that list, each computer will then be queried for its privileged users. If the Active Directory commandlet is not installed visit: https://blogs.technet.microsoft.com/ashleymcgкле/2016/02/26/install-the-active-directory-PowerShell-module-on-windows-10/

The “Find-BadPrivilege” script was ran on the local system to discover the highly privileged accounts as demonstrated in Figure 7 below.
The “Find-BadPrivilegeDomain” script was run on the domain to discover highly privileged accounts as demonstrated in the figure below.
The script lists the privilege and then all the accounts with that privilege on that host. This tool provides administrators with a complete domain picture of user accounts with bad privileges. The script fills the gap missed by the other tools. The script is provided in the appendix and on github at: https://github.com/lordsaibat/finding_windows_privileges

With this script, system administrators will be able to close the gap created by combining privileges and permissions. Find-BadPrivilegeDomain gives domain administrators insight into which users have had one of “The Seven Maleficent” privileges added to their account. With this extra tool in the toolbox, defenders now can lock down their Active Directory environments further.
References


http://csrc.nist.gov/groups/SNS/rbac/documents/design_implementation/Managing_Role_Permission_Relationships.htm


Indiana University. (2017, May 16). In Windows, what are local users and domain users? Retrieved from Indiana University: https://kb.iu.edu/d/anbn


5. **Appendix A**
5.1. Attacker gaining access to more privileges

Several sessions to the same host under different accounts was gained.

Starting with AAriel, each accounts’ privileges where enumerated.

Figure 59: Different levels of meterpreter sessions
Aariel is part of the Domain Admins group as demonstrated in the figure below, but due to the process starting in a lower UAC level certain privileges were not granted.

Examining the reverse meterpreter executed under SANS/Aariel demonstrates it does not have the SeImpersonate or SeDebug privileges.
The next session that was created was under the FCastle user.
Fastle is a standard domain user, but also part of the “Backup Operators”, which should have the SeRestorePrivileges.

The reverse meterpreter session for FCastle again does not have the correct privileges.
The next account was the Domain Administrator account showing several privileges.
The Domain Administrator account was part of the local Administrators group and the Domain Admins group.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User name</td>
<td>Administrator</td>
</tr>
<tr>
<td>Full Name</td>
<td></td>
</tr>
<tr>
<td>Comment</td>
<td>Built-in account for administering the computer/domain</td>
</tr>
<tr>
<td>User’s comment</td>
<td></td>
</tr>
<tr>
<td>Country/region code</td>
<td>000 (System Default)</td>
</tr>
<tr>
<td>Account active</td>
<td>Yes</td>
</tr>
<tr>
<td>Account expires</td>
<td>Never</td>
</tr>
<tr>
<td>Password last set</td>
<td>10/8/2017 7:16:21 PM</td>
</tr>
<tr>
<td>Password expires</td>
<td>11/19/2017 7:16:21 PM</td>
</tr>
<tr>
<td>Password changeable</td>
<td>10/9/2017 7:16:21 PM</td>
</tr>
<tr>
<td>Password required</td>
<td>Yes</td>
</tr>
<tr>
<td>User may change password</td>
<td>Yes</td>
</tr>
<tr>
<td>Workstations allowed</td>
<td>All</td>
</tr>
<tr>
<td>Logon script</td>
<td></td>
</tr>
<tr>
<td>User profile</td>
<td></td>
</tr>
<tr>
<td>Home directory</td>
<td></td>
</tr>
<tr>
<td>Last logon</td>
<td>10/8/2017 7:32:05 PM</td>
</tr>
<tr>
<td>Logon hours allowed</td>
<td>All</td>
</tr>
<tr>
<td>Local Group Memberships</td>
<td>*Administrators</td>
</tr>
<tr>
<td>Global Group memberships</td>
<td>*Domain Users</td>
</tr>
<tr>
<td></td>
<td>*Domain Admins</td>
</tr>
<tr>
<td></td>
<td>*Enterprise Admins</td>
</tr>
<tr>
<td></td>
<td>*Group Policy Creator</td>
</tr>
<tr>
<td></td>
<td>*Schema Admins</td>
</tr>
</tbody>
</table>

The command completed successfully.

Figure 67: Domain Admin group memberships
The reverse meterpreter executed under conhost.exe for the Domain Administrator user contained the necessary SeDebug and SeImpersonate privileges. (Which will help to elevate to “system” level below).

The meterpreter session with local Administrator account was enumerated to determine its privileges.

Figure 68: Meterpreter session running under the Domain Admin account

Figure 69: Privileges under a Local Admin account
Confirming this user is part of the local Administrators group.

Figure 70: Local Administrator account
This reverse meterpreter for the local administrator account did not have SeDebug or SeImpersonate privileges again.

Figure 71: Local Admin account meterpreter privileges
System level privileges were attempted on each session, only the session with the SeDebugPrivilege was elevated.

Figure 72: Gaining System level

The system level access was gained through Named Pipe Impersonation (In Memory/Admin)

Figure 73: getsystem help

The privileges command was re run on the session with the system access now.
To gain those privileges in Windows 10, the user had to run an elevated shell. This was simulated by using the “exploit/windows/local/ask” in Metasploit to have the user “Aariel” run a new session.
Figure 75: Elevating UAC

The new session has new privileges due to being in an elevated shell.

Figure 76: Privileges under UAC elevated shell

Examining the new session created has the additional privileges.
The loaded incognito module demonstrates additional tokens.

Compared to an unelevated session with the same user.
Figure 79: Original Arial session

Using the elevated session, it was able to elevate to system level again.

Figure 80: System level gain via in memory

Confirming that the session with the system level now has all the additional privileges.
Figure 81: Additional privileges on System level access
5.2. Gold Finger Auditing and Reporting functions

Security Audit Tool

1. List of all domain user accounts
2. List of all enabled domain user accounts
3. List of all disabled domain user accounts
4. List of all locked domain user accounts
5. List of all unlocked domain user accounts
6. List of all administrative domain user accounts
7. List of all domain user accounts that have logged on in the last few days
8. List of all domain user accounts that have not logged on in the last few days
9. List of all domain user accounts that have never logged on
10. List of all domain user accounts that have logged on at least once
11. List of all domain user accounts that failed a logon attempt in the last few days
12. List of all domain user accounts created in the last few days
13. List of all domain user accounts changed in the last few days
14. List of all domain user accounts deleted in the last few days
15. List of all domain user accounts that have an expiration date
16. List of all domain user accounts that do not have an expiration date
17. List of all domain user accounts that expired in the last few days
18. List of all domain user accounts that will expire in the next few days
19. List of all domain user accounts that require passwords to logon
20. List of all domain user accounts that do not require passwords to logon
21. List of all domain user accounts whose passwords never expire
22. List of all domain user accounts whose password must be changed at next logon
23. List of all domain user accounts whose passwords have changed in the last few days
24. List of all domain user accounts whose passwords have not changed in the last few days
25. List of all domain user accounts whose passwords are stored using reversible encryption
26. List of all domain user accounts that require Smart cards for login
27. List of all domain user accounts that are marked sensitive and cannot be delegated
28. List of all domain user accounts that are not marked sensitive and cannot be delegated
29. List of all domain user accounts that can logon to any workstation
30. List of all domain user accounts that can only logon to specific workstations
31. List of all domain user accounts for which specific logon hours have not been specified
32. List of all domain user accounts for which specific logon hours have been specified
33. List of all domain user accounts that can logon anytime
34. List of all domain user accounts for which a logon script is specified
35. List of all domain user accounts for which no logon script is specified
36. List of all domain user accounts for which no description is specified
37. List of all security groups
38. List of all built-in security groups
39. List of all domain local security groups
40. List of all global security groups
41. List of all universal security groups
42. List of all administrative security groups
43. List of all security groups created in the last few days
44. List of all security groups changed in the last few days
45. List of all security groups deleted in the last few days
46. List of all security groups that cannot be deleted
47. List of all security groups that have members
48. List of all security groups for which no manager is designated
49. List of all security groups for which no description is specified
50. List of all domain computer accounts
51. List of all enabled domain computer accounts
52. List of all disabled domain computer accounts
53. List of all domain computer accounts that have authenticated in the last few days
54. List of all domain computer accounts that have not authenticated in the last few days
55. List of all domain computers accounts that have never authenticated
56. List of all domain computer accounts that have authenticated at least once
57. List of all domain computer accounts created in the last few days
58. List of all domain computer accounts changed in the last few days
59. List of all domain computer accounts deleted in the last few days
60. List of all domain controllers
61. List of all domain computer accounts that are trusted for delegation
62. List of all domain computer accounts that are trusted for unconstrained delegation
63. List of all domain computer accounts for which no description is specified
64. List of all domain computer accounts for which no location is specified
65. List of all domain computer accounts for which no manager is designated
66. List of all organization units
67. List of all organization units created in the last days
68. List of all organization units changed in the last few days
69. List of all organization units deleted in the last few days
70. List of all organization units for which no manager is designated
71. List of all organizational units for which no description is specified
72. List of all containers
73. List of all containers created in the last few days
74. List of all containers changed in the last few days
75. List of all containers deleted in the last few days
76. List of all containers for which no description is specified
77. List of all group policy objects
78. List of all enabled group policy objects
79. List of all disabled group policy objects
80. List of all group policy objects whose user settings are disabled
81. List of all group policy objects whose computer settings are disabled
82. List of all group policy objects whose user and computer settings are disabled
83. List of all group policy objects created in the last few days
84. List of all group policy objects changed in the last few days
85. List of all group policy objects deleted in the last few days
86. List of all printers published in Active Directory
87. List of all printers published in Active Directory in the last few days
88. List of all printers published in Active Directory which changed in the last few days
89. List of all printers published in Active Directory that were deleted in the last few days
90. List of all contacts
91. List of all contacts created in the last few days
92. List of all contacts changed in the last few days
93. List of all contacts deleted in the last few days
94. List of all service connection points
95. List of all service connection points created in the last few days
96. List of all service connection points changed in the last few days
97. List of all service connection points deleted in the last few days
98. List of all service connection for which no keywords are specified
99. List of all service connection for which no DNS service name is specified
100. List of all objects

Group Membership Reporting Tool
1. View the direct membership of an Active Directory security group
2. View the complete nested group membership of an Active Directory security group
3. View the complete list of all Active Directory security groups to which a user belongs

Token Viewer & Token-Size Calculator
1. View the complete list of all SIDs contained in a domain user account’s access token, when accessing resources in a specific domain
2. Compute and list, for multiple domain user accounts, the size of their tokens, when accessing resources in a specific domain

ACL Viewer
1. View the ACL of an Active Directory object
2. View the SACL (System ACL) of an Active Directory object

ACL Exporter
1. Export ACLs of all objects in an Active Directory tree
2. Export SACLs (System ACLs) of all objects in an Active Directory tree

Permissions Analyzer
1. Who has what permissions on an Active Directory object?
2. Who has what permissions in an Active Directory tree?

Effective Permissions Calculator
1. Who has what effective permissions on an Active Directory object?

Effective Access Calculator
2. Who has what effect access (i.e. who can perform what administrative tasks) on an Active Directory object?

Administrative Access / Delegation Audit Tool
3. Who can create user accounts?
4. Who can delete user accounts?
5. Who can reset user account passwords?
6. Who can disable/enable user accounts?
7. Who can unlock locked user accounts?
8. Who can change the expiration date of user accounts?
9. Who can disable/enable smart card requirement for interactive logon by user accounts?
10. Who can force users to change their user account passwords at next logon?
11. Who can prevent users from changing their user account passwords?
12. Who can change the logon name of user accounts?
13. Who can change the Pre-Windows 2000 logon name of user accounts?
14. Who can change the logon hours of user account?
15. Who can change the logon of user accounts?
16. Who can change the profile path for user accounts?
17. Who can change the logon script for user accounts?
18. Who can change alternate security identities associated with user accounts?
19. Who can change whether or not user accounts are sensitive and cannot be delegated?
20. Who can change whether or not DES encryption types should be used for user accounts?
21. Who can change whether or not Kerberos pre-authentication is required for user accounts?
22. Who can change the first name of user accounts?
23. Who can change the last name of user accounts?
24. Who can change the display name of user accounts?
25. Who can change the description of user accounts?
26. Who can change the office location of user accounts?
27. Who can change the organizational title of user accounts?
28. Who can change the organizational department of user accounts?
29. Who can change the organizational manager for user accounts?
30. Who can change the picture associated with user accounts?
31. Who can change the security permissions protecting user accounts?
32. Who can create computer accounts?
33. Who can delete computer accounts?
34. Who can reset computer accounts?
35. Who can disable/enable computer accounts?
36. Who can change the expiration date of computer accounts?
37. Who can change the computer name (Pre-Windows 2000) of computer accounts?
38. Who can change the DNS name of computer accounts?
39. Who can change the machine role of computer accounts?
40. Who can change the description of computer accounts?
41. Who can change the location of computer accounts?
42. Who can change the Service Principal Names (SPNs) of computer accounts?
43. Who can change alternate security identities associated with computer accounts?
44. Who can change the designated manager of computer accounts?
45. Who can change the picture associated with computer accounts?
46. Who can change the security permissions protecting computer accounts?
47. Who can create security groups?
48. Who can delete security groups?
49. Who can change security group memberships?
50. Who can add/remove oneself to/from the membership of security groups?
51. Who can change security group scopes?
52. Who can change security group types?
53. Who can change the group name (Pre-Windows 2000) of security groups?
54. Who can change the description of security groups?
55. Who can change the email-address of security groups?
56. Who can change notes annotated for security groups?
57. Who can change the designated manager of security groups?
58. Who can change the security permissions protecting security groups?
59. Who can change the maximum password age for domain user account?
60. Who can change the minimum password age for domain user accounts?
61. Who can change the lockout duration for domain user account?
62. Who can change the lockout threshold for domain user accounts?
63. Who can change the lockout observation window for domain user accounts?
64. Who can create organization units (OUs)?
65. Who can delete organization units (OUs)?
66. Who can disable group policies linked to organizational units?
67. Who can change the list of group policies linked to organization units?
68. Who can change the precedence of group policies linked to organizational units?
69. Who can generate resultant set of policy (logging-mode) for users/computers in an OU?
70. Who can generate resultant set of policy (planning-mode) for users/computers in an OU?
71. Who can change the description of organizational units?
72. Who can change the street address of organization units?
73. Who can change the city of organizational units?
74. Who can change the state/province of organization units?
75. Who can change the zip/postal-code of organization units?
76. Who can change the country of organization units?
77. Who can change the designated manager of organizational units?
78. Who can change the security permissions protecting organizational units?
79. Who can create containers?
80. Who can delete containers?
81. Who can change the description of containers?
82. Who can change the security permissions protecting containers?
83. Who can create service connection points?
84. Who can delete service connection points?
85. Who can change the keywords of service connection points?
86. Who can change the description of service connections points?
87. Who can change the binding information of service connection points?
88. Who can change the service DNS name of service connection points?
89. Who can change the service DNS name type of service connection points?
90. Who can change the vendor of service connection points?
91. Who can change the version number of service connection points?
92. Who can change the hi version number of service connection points?
93. Who can change the low version of service connection points?
94. Who can change the class-name of service connection points?
95. Who can change the security permissions protecting service connection points?
96. Who can create group policy containers?
97. Who can delete group policy containers?
98. Who can change the security permissions protecting group policy containers?
99. Who can create contacts?
100. Who can delete contacts?
101. Who can change the security permissions protecting contacts?
102. Who can create (publish) printers?
103. Who can delete published printers?
104. Who can change the description of published printers?
105. Who can change the share name of published printers?
106. Who can change the designated manager of published printers?
107. Who can change the security permissions protecting published printers?
## Upcoming Training

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS Prague August 2019</td>
<td>Prague, Czech Republic</td>
<td>Aug 12, 2019 - Aug 17, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Network Security 2019</td>
<td>Las Vegas, NV</td>
<td>Sep 09, 2019 - Sep 16, 2019</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Online</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
<tr>
<td>SANS SelfStudy</td>
<td>Books &amp; MP3s Only</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>