Magdeburger Journal zur Sicherheitsforschung

Gegründet 2011 | ISSN: 2192-4260
Herausgegeben von Stefan Schumacher
Erschienen im Magdeburger Institut für Sicherheitsforschung
http://www.sicherheitsforschung-magdeburg.de/publikationen/journal.html

This article appears in the special edition »In Depth Security – Proceedings of the DeepSec Conferences«.
Edited by Stefan Schumacher and René Pfeiffer

BadGPO

Using Group Policy Objects for Persistence and Lateral Movement

Immanuel Willi and Yves Kraft

Group Policy is a feature which provides centralized management and configuration functions for the Microsoft operating system, application and user settings. Group Policy is simply the easiest way to reach out and configure computer and user settings on networks based on Active Directory Domain Services (AD DS). Such policies are widely used in enterprise environments to control settings of clients and servers: registry settings, security options, scripts, folders, software installation and maintenance, just to name a few. Settings are contained in so-called Group Policy Objects (GPOs) and can be misused in a sneaky way to distribute malware and gain persistence in an automated manner in a post exploitation scenario of an already compromised domain. In a proof of concept, inspired by Phineas Fishers' article about pwning HackingTeam, we will show how persistence and lateral movement in a compromised company network can be achieved, and demonstrate some PowershellEmpire Framework modules which we created. PowershellEmpire is basically a post-exploitation framework that utilises the widely-deployed PowerShell tool for all your system-smashing needs. There are already functionalities built-in regarding GPOs. We tried to further evolve the miss-use of GPOs in additional scenarios. Furthermore, we will discuss some countermeasures including detection and prevention mechanisms.

Citation: Willi, I. & Kraft, Y. (2017). BadGPO: Using Group Policy Objects for Persistence and Lateral Movement. *Magdeburger Journal zur Sicherheitsforschung*, *13*, 772–778. Retrieved June 28, 2017, from http://www.sicherheitsforschung-magdeburg.de/uploads/journal/MJS_052_Willi_GPO.pdf

1 Introduction

After an Italian company which sells spyware was compromised, a detailed report was published last year on how the hacker had carried out the attack.[1] The same hacker had already successfully attacked a German-British company which had also produced negative headlines by selling Trojans and other hacking tools. The report was published by the attacker himself and provides in-depth information about the techniques used in such a complex attack. The conclusion is especially interesting, as the attack is a classic Advanced Persistent Threat (APT) attack. The adversary is usually very careful, with the aim of remaining undiscovered for a longer period of time to compromise systems in the target network and gather large amounts of data.

While looking into the report, two paragraphs have caught our attention:[2]

- Remote Management [Line 565]
  5) GPO
  If all those protocols are disabled or blocked by the firewall, once you're Domain Admin, you can use GPO to give users a login script, install an msi, execute a scheduled task [13], or, like we'll see with the computer of M**** R***** (one of H***** T****'s sysadmins), use GPO to enable WMI and open the firewall.
- Persistence [Line 726]
  To hack companies, persistence isn't needed since companies never sleep. I always use Duqu 2 style »persistence», executing in RAM on a couple high-uptime servers.

In the first section, the idea of misusing Group Policy Objects as an offensive attack instrument is interesting. The approach of abusing GPOs to deliver malware or deploy illegitimate configurations to target systems is useful, since when spreading malware via GPOs (similar to using PowerShell), legitimate »on-board« tools of the Windows domain administration are employed. Using GPOs, malware is smuggled past firewalls IDS/IPS (intrusion detection/prevention systems) and all domain-joined systems can be reached. A further advantage is that during an attack a target system does not need to be online, as malicious GPO payloads are delivered as soon as the target system logs back into the domain. Even if the attack happened weeks ago the attack may still work. Additionally, GPOs oftentimes grow over time and are chaotically stored and linked in the Microsoft Active Directory. Furthermore, naming conventions are neglected and processes to remove obsolete GPOs are missing, which all plays into the hands of an attacker who misuses GPOs.

In the second section mentioned above, the approach of using a backdoor in the RAM of a high-availability server to create a persistent connection into a compromised network is interesting. After restarting an infected system, a backdoor would get lost instantly. However, this not will happen with multiple high-availability servers, because not all of them will get rebooted at the same time.

There is a wide range of established tools for system administrators to manage their companies IT. Similarly, there are tools for »black-hat« attackers or penetration testers facilitating post-exploitation tasks in »Advanced Persistent Threat (APT)« scenarios. Such frameworks offer a large variety of options, such as functions for managing compromised systems, evaluating more targets, implementing backdoors, and escalating privileges or spying on users.

With »PowerShell Empire«, a powerful framework was started about a year ago, which meets all of the above requirements and is still in development. One of the biggest advantages of the module-based »PowerShell Empire« framework is the small forensic footprint left on target systems. Executing PowerShell commands is not recognized as a malicious activity by antivirus or endpoint protection software, because PowerShell is a legitimate on-board tool on Windows-based systems.

Additionally, the communication to the »Command and Control« instance is encrypted and most of the modules are developed in a way to run completely on memory on the target system and not on the hard disk. Since the malicious code does not touch the hard disk, antivirus and endpoint protection systems will have an even harder time to detect an attack. If data needs to be written on the hard disk developers of Empire modules are advised to mark the module as not »opsec-safe«.

To combine the ideas mentioned in the respective two paragraphs, we have developed various PowerShell Empire modules. These modules equip penetration testers in ethical hacking projects or red team engagements with tools in post-exploitation scenarios to expand access to peripheral systems and provide persistence in a convenient and fully automated way via GPOs. The only prerequisite is that access rights as a domain administrator must have been obtained.

Table 1 shows the developed modules.

To achieve different objectives, the described modules can be combined in different ways, as shown in Fig. 1

1.1 Scenario 1: Backdoor in memory

An attacker locally starts a PowerShell Empire »Listener« and a corresponding »Launcher« on the domain controller, which connects back to the »Listener«. Since outbound ports are not blocked in most firewalls, the connection can be established. Using the module »**getGPO**«, an attacker can read all existing GPOs, write down their UUIDs and add content with malicious settings (see Fig. 2). The module »**setGpRegistryValue**« creates a »run« or »run once« registry key on the linked target machines.

1 https://arstechnica.com/security/2016/04/how-hacking-team-got-hacked-phineas-phisher

2 http://pastebin.com/raw/0SNSvyjJ

getGPO	Read GPOs from Domain Controller. It is possible to read out all GPOs, or only specific ones (by name or GUID).
setGpRegistryValue	This module is intended to set a »run« or »run once« registry value using GPOs. It creates a new (or modifies an existing) GPO on the Domain Controller. Options for linking and enabling GPOs can be provided if required.
newGpFirewallRule	This module is intended to set a Windows firewall rule using GPOs. All configuration options of a Windows firewall rule can be provided.
newGpSetServiceStatus	Starts or stops a Windows service by setting the startup mode for the given service.
invokeGPUpdate	The deployed GPO will change the settings on a client after up to 90 minutes. The module invokeGPUpdate enforces an immediate update of the GPOs.

Table 1: Developed Modules

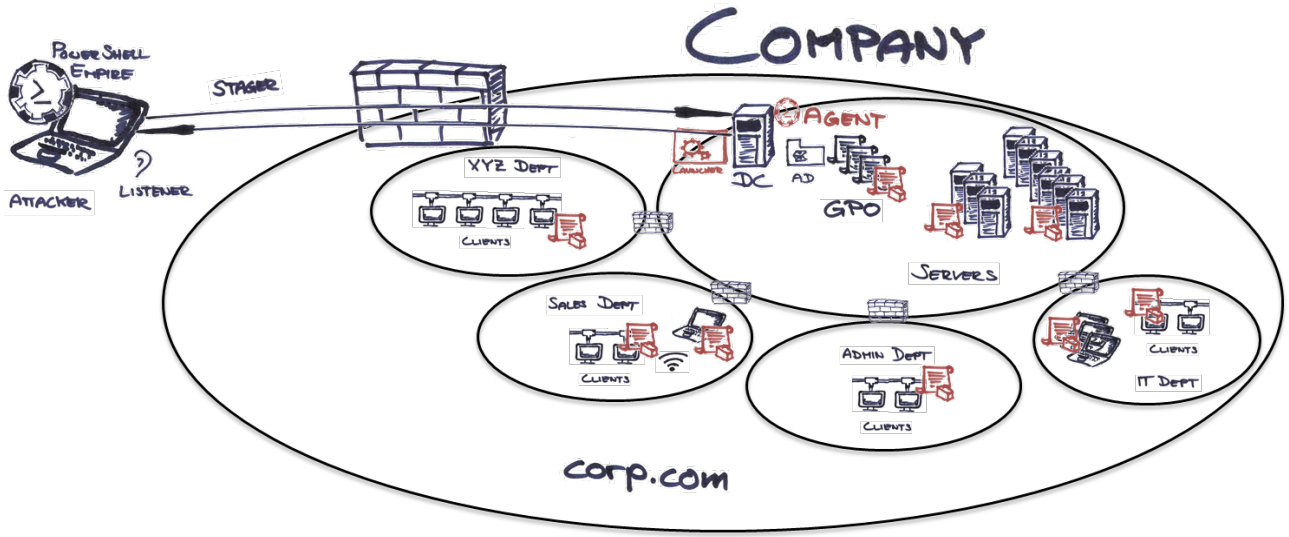

Figure 1: Use of manipulated Group Policy Objects (GPOs)

The PowerShell code to be executed again contains a »Launcher«, which initiates an outbound connection to the attacker.

1.2 Scenario 2: Manipulate the Windows firewall and start a service

In another scenario, a connection to the domain controller is created. The goal of the attack in this scenario is to manipulate the Windows firewall rules and start the Windows Management Instrumentation (WMI) service, which executes remotely sent commands in a hardened environment. First, the »**getGPO**« module is used to read the existing Group Policies. After finding a suitable GPO, the module »**newGpFirewallRule**« is employed to extend the GPO with a malicious firewall rule (see Fig. 3). This rule will allow any incoming TCP connections on the target system. The module »**newGpSetServiceStatus**« starts the WMI service on the target machine. To bypass the default waiting period of 90 minutes until the manipulated GPO is applied, the module »**invokeGPUpdate**« is executed, which enables immediate remote access to the WMI service.

Having remote WMI access on multiple target machines, actions such as local file searches can be carried out. Using this technique, file searches are much more scalable, considerably faster and unobtrusive compared to manual file searches in remote desktop protocol (RDP) connections.

With the inactive firewall on all domain-joined systems of the fictitious company Corp.com and the started WMI service, all recipients of the manipulated GPO can be reached and receive commands remotely sent to WMI.

To remotely search for a local file (e. g. proof.txt), the following command can be used on the »c:\« drive via WMI:

Get-Content <list of IPs> | ForEach-Object {Get-wmiobject CIM_DataFile -filter »Drive='c:' AND Filename ='proof' AND extension='txt' -Impersonate 3 -computername $_ | Select PSComputername, Name -Unique}

The result is shown in Fig. 4

2 Countermeasures against the manipulation of GPOs

Generally, a compromised domain administrator account is a serious and time-consuming security issue to resolve. As an immediate action, suspicious accounts with administrative privileges can be deactivated or passwords of administrative accounts can be reset. However, ensuring that unauthorized third parties cannot gain access to internal systems anymore requires in-depth technical expert know-how.

The following preventative measures may be implemented:

- Review of the GPOs on a regular basis
- Clearly defined naming conventions for GPOs
- Change of the monitoring and intrusion detection infrastructure to cover
  - Logging of the creation of GPOs
  - Logging of GPO changes
- Limiting the privileges of administrative users
- Limiting access rights for the execution of administrative tools

During our research, we have identified the calculation of a hash value of the »Group Policy« folder on the domain controller as a further measure. An exact match of a previously saved hash value eliminates illegitimate manipulations of GPOs. Similar measures are used on host-based intrusion detection systems.

2.1 General countermeasures against Advanced Persistent Threat (APT) attacks

Instead of reacting to a compromised system landscape, it is advisable to implement preventive measures to make an APT attacker's life harder. A classic APT scenario can be broken down into the four stages preparation, infection, deployment and persistence. These stages form a circular process which can be repeated on other systems in the target network after gaining an initial foothold. Fig. 5 shows the circular process of the APT lifecycle

Unfortunately, there is no pre-built »all-in-one« solution to prevent APT attacks. Instead, it is advisable to take measures for the different stages of an APT attack:

- (Web application) firewalls
- Segmentation of networks
- Network access control (NAC)
- Proactively install updates for software and operating systems
- Multifactor authentication mechanisms for systems exposed to the internet or highly confidential/sensitive systems
- Security incident event monitoring
- Penetration tests carried out on a regular basis
- IT security awareness trainings
- Restrict access rights based on the »least privilege« principle
- Implementation of endpoint control software
- Application whitelisting
- Develop security alliances with other companies of the same industry

The principle of »Defense in Depth« refers to a defense with multiple layers of security controls to slow down or even prevent an attack. The time saved can then be used to detect on-going attacks. The more se-

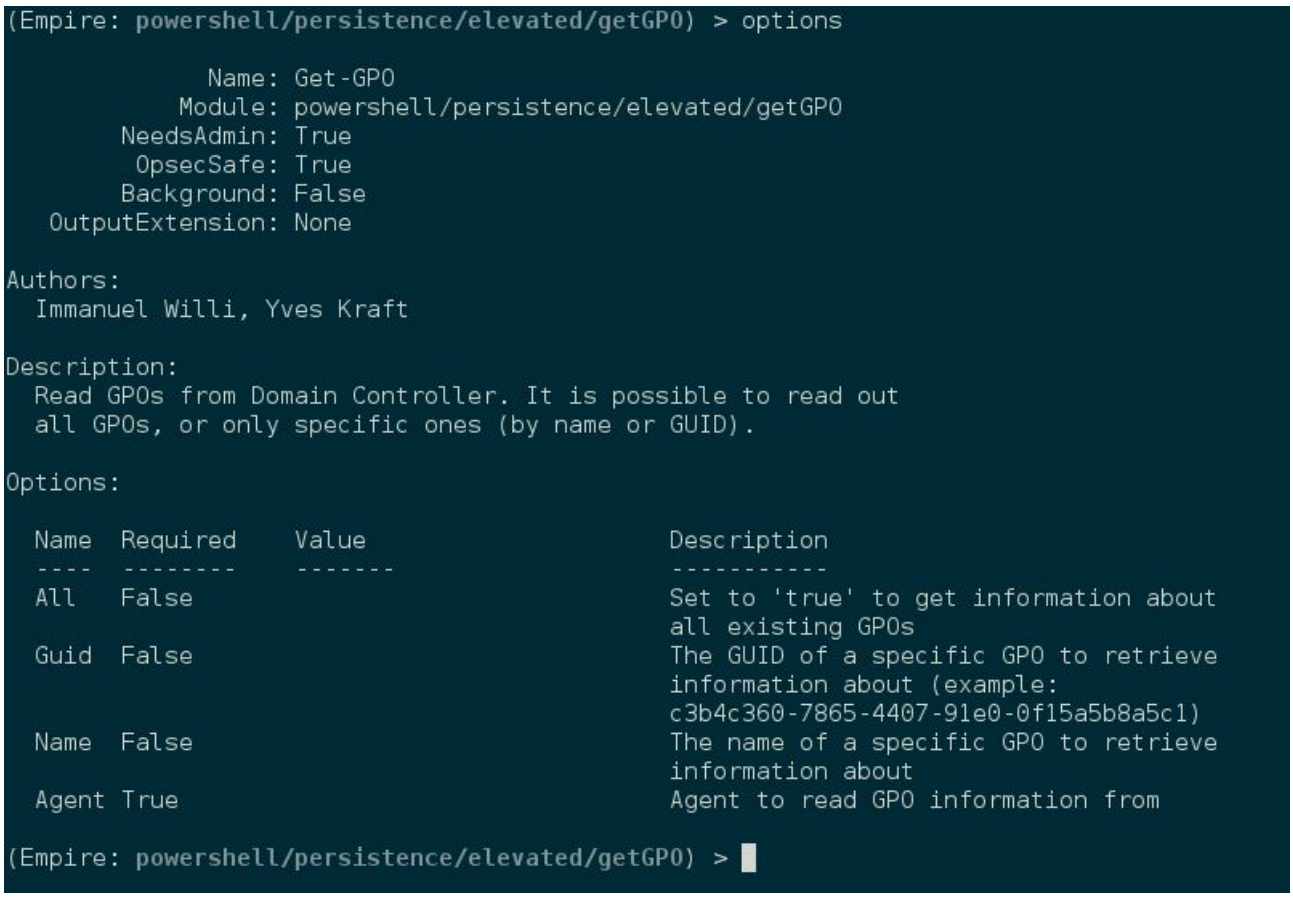

Figure 2: PowerShell Empire Module getGPO

curity measures are implemented, the more resource- and cost-intensive a successful attack will be. Each one of the four stages of the APT lifecycle should contain multiple lines of defense. In the end, the more tightly-knit the network of the defense is, the more excusable is one less effective or ineffective measure.

3 About the Authors

Since 2011 Yves Kraft works as a Senior Security Consultant and Penetration Tester for the Swiss company Oneconsult and is Branch Manager of the subsidiary Office in Bern. His area of expertise are penetration tests of Windows and Linux, system hardening, ethical hacking in IoT environments and software defined radio. As a former system and network engineer, Yves Kraft managed numerous servers, applications and networks. He worked for a large Swiss university, the government and the financial industry. He studied computer science at the Bern University of Applied Sciences (BFH) with a focus on IT security. Yves Kraft is an Offensive Security Certified Professional (OSCP), a certified OSSTMM Professional Security Tester (OPST), OSSTMM Professional Security Analyst (OPSA), OSSTMM Professional Security Expert (OPSE), OSSTMM trainer and ISO 27001 Lead Auditor and regularly speaks at local and international security conferences.

Before entering the IT security industry, Immanuel Willi worked for several years as a system administrator and head of IT services in the academic field. After completing his extra-occupational Bachelor's degree in computer science, he joined Oneconsult AG in 2013. Since then, he has been working as a penetration tester and senior security consultant, and is thus familiar with the perspective of both an attacker and a defender. Immanuel Willi holds a number of certifications such as ISO / IEC 27001 Lead Auditor, Offensive Security Certified Professional (OSCP) and Information Systems Security Professional (CISSP).

```
             Name: NewGpo_Firewall_Rule
           Module: powershell/persistence/elevated/2017/newGpFirewallRule
       NeedsAdmin: True
        OpsecSafe: False
MinLanguageVersion: 2
       Background: False
   OutputExtension: None

Authors:
  Yves Kraft, Immanuel Willi

Description:
  This module is intended to set a Windows Firewall Rule using
  Group Policy Objects (GPO). It creates a new (or modifies an
  existing) GPO on the Domain Controller. Options for linking
  and enabling GPOs can be provided if required. Requirements:
  This module needs Domain Admin privileges, and needs to be
  run against a Domain Controller!The deployed GPO will change
  the Firewalls settings on a client after up to 90 Minutes,
  or immediatley when executing the invokeGPUpdate module.

Options:

  Name           Required    Value                  Description
  ----           --------    -------                -----------
  GpoName        True                               Either the module creates a new GPO with
                                                    the given name, or extends an existing
                                                    GPO (i.e. "Default Domain Policy"). The
                                                    default value is a random string.
  RemoteAddress  False       Any                    Specifies that network packets with
                                                    matching IP addresses match this rule.
                                                    This parameter value is the second end
                                                    point of an IPsec rule and specifies the
                                                    computers that are subject to the
                                                    requirements of this rule. This
                                                    parameter value is an IPv4 or IPv6
                                                    address, hostname, subnet, range, or
                                                    Any.
  RuleName       True        New_FWRule             Specifies that only matching firewall
                                                    rules of the indicated name are created.
                                                    This parameter acts just like a file
                                                    name, in that only one rule with a given
                                                    name may exist in a policy store at a
                                                    time. During group policy processing and
                                                    policy merge, rules that have the same
                                                    name but come from multiple stores being
```

Figure 3: PowerShell Empire Module new_GPO_Firewall_Rule

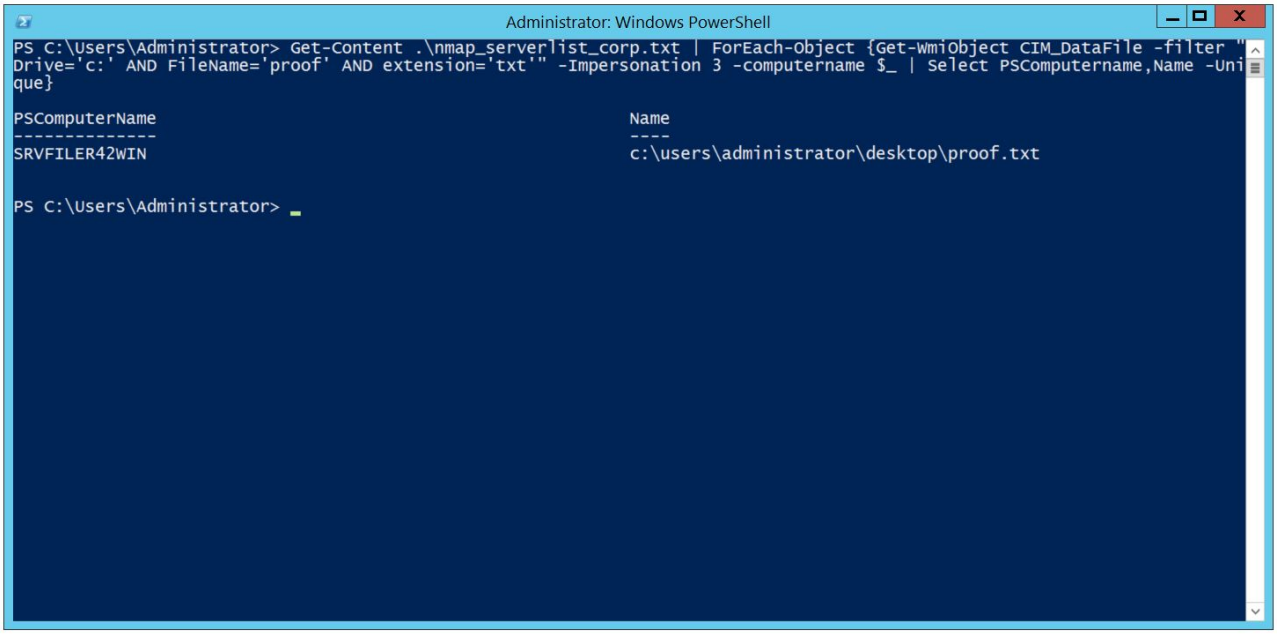

Figure 4: Local file search using remote WMI

Figure 5: APT lifecycle