Intercepter-NG
absolute capture

Group Policy Hijacking.

Intro

One more time (how much longer? lol) SMB and Domain Network brings us lots of fun. A new attack vector has been found! This time it is called Group Policy Hijacking. Though it's not my own discovery I am in fact the first person who released a complete exploit for this vector to public as a part of Intercepter-NG. And as always being a bit lazy I will not give a detailed description of the problem since it was provided earlier within two good blog posts that I really suggest you reading:

[1]. https://labs.mwrinfosecurity.com/blog/how-to-own-any-windows-network-with-group-policy-hijacking-attacks/

[2]. https://blog.coresecurity.com/2015/05/18/ms15-011-microsoft-windows-group-policy-real-exploitation-via-a-smb-mitm-attack/

The problem

Anyways I have to say something about it at least in a few words. There is a period of time consisting of 90 minutes + a random number between 0 and 30 during which a member of the domain network checks if there are any updates available for the Group Policy. It is done by means of SMB via network path \\DC\SYSVOL\domain.name\Policies\UUID\gpt.ini.

The content of gpt.ini:

```
[General]

Version=12345
```

This number is a relative version of current policies. If it is equal to the previous value from the last update the system understands that there is nothing to update and resets countdown to the next 90+ minutes. Otherwise the Group Policies have to be updated and the client asks DC for the so called active CSE (client side extensions). CSE consist of different logon scripts, scheduler tasks and so on. The major part of CSE is in fact a regular file with preferences inside. Thus being the man-in-the-middle we are able to replace some of the existing CSE and get some profit: command execution, new scheduler task... But! It would have been too easy. Actually all CSE are disabled by default, so we have to find another way.

In case the version number in gpt.ini has changed the client also asks for another file GptTmpl.inf. This template allows us to modify the client's registry remotely! You may think of putting the payload into autorun registry keys pointing to some network path, but you'll have to wait for the computer to reboot. Authors of [1] and [2] used the AppInit_DLL registry key as a Proof-of-Concept to demonstrate the exploitation of Group Policy Hijacking. Should I remind you that this key has been disabled for many years now and it is completely useless in the wild? As it was said before this was just a PoC and I had to find another way to get remote shell without the need of rebooting...

The magic

Enough time had been spent on looking for a magical registry key that can bring the desired result. At last I found it:

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
NT\currentversion\image file execution options]

With it you can register a personal debugger for a single .exe file, i.e. you can tell the system to open calc.exe with c:\path\debugger.exe so when you run calc.exe it actually calls:

c:\path\debugger.exe calc.exe

It looks very promising, but only at a first glance. You can put the payload as a custom debugger for IE or another application that the target may run during the work - this is obviously a much faster way to execute the payload. But a problem appears if the target user has restricted permissions on the system. You still can change his registry with Group Policy Hijacking and you still can execute your payload, but you have to clean up after your previous steps and remove the record of the custom debugger because after the attack stops the network path with the payload is no longer available and target application (IE, calc...) does not work anymore. Of course you need administrative access to clean HKEY_LOCAL_MACHINE. I'll skip the details of ongoing search to find a way to avoid these nuances. So here is the final solution.

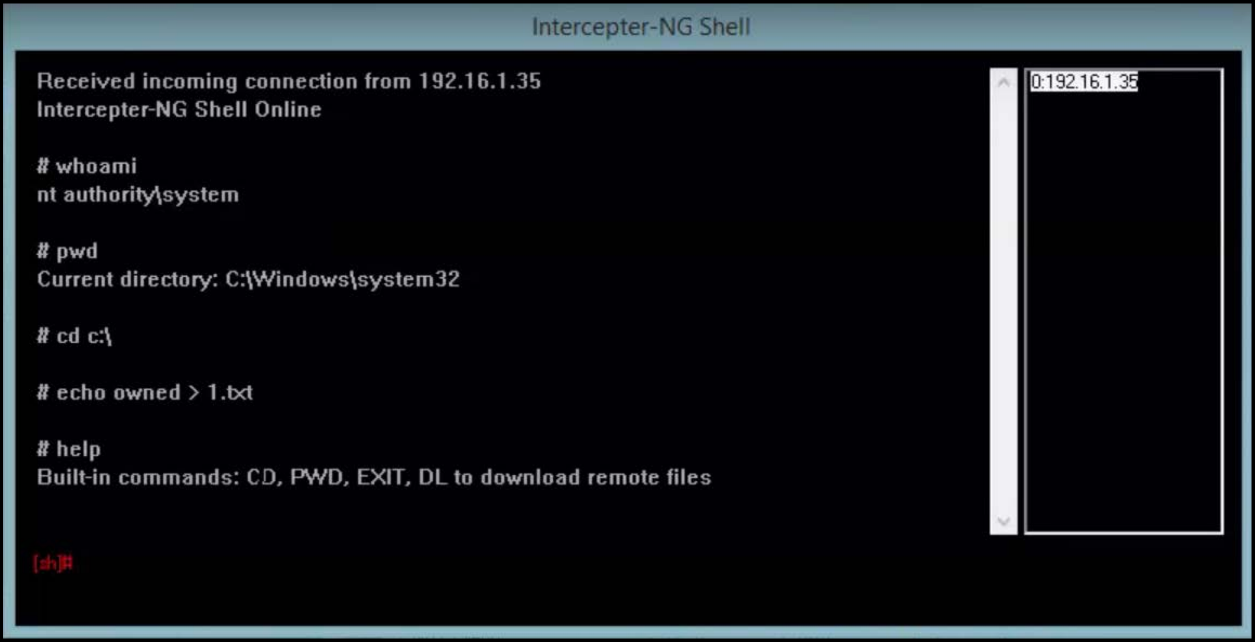
When the system applies new group policies it creates a new process taskhost.exe with SYSTEM privileges, even on a restricted user. It is clear now that we should set up our custom 'debugger' for taskhost.exe and this way we will get an RCE with SYSTEM privileges immediately, without rebooting or waiting for user's actions!

Conclusion

It's unbelievable but most of existing Domain Networks are completely defenseless against the Group Policy Hijacking technique. Using ARP poisoning and waiting for 90+rand()%30 minutes guarantees execution of the payload. Intercepter-NG uses its custom connect back shell that cleans everything on the target machine. It also supports multi-shells so you can run an attack against a lot of domain members and switch between them. Before using make sure your shared resources are available from the outside. Check **Network access: Let Everyone permissions apply to anonymous users** and other sharing options. It has been tested on patched Windows 7\8.1 and servers 2008R2\2012R2. Microsoft released a patch for MS15-011, but the defense mechanism should be configured manually. The bulletin also contains an interesting phrase:

«Users whose accounts are configured to have fewer user rights on the system could be less affected than users who operate with administrative user rights».

As it has been shown earlier everyone is vulnerable equally. Demonstration video: https://youtu.be/wVLD2iT6ADo?t=196



```
Received incoming connection from 192.16.1.35
Intercepter-NG Shell Online

# whoami
nt authority\system

# pwd
Current directory: C:\Windows\system32

# cd c:\

# echo owned > 1.txt

# help
Built-in commands: CD, PWD, EXIT, DL to download remote files


[sh]#
```

Contacts

Site: http://sniff.su

Twitter: @IntercepterNG

Email: intercepter.mail@gmail.com

Blog: http://intercepter-ng.blogspot.ru

Forum: http://intercepterng.boards.net

Mirror: intercepter-ng.github.io