Relay Attacks via Cobalt Strike Beacons

pkb1s.github.io/Relay-attacks-via-Cobalt-Strike-beacons/

February 1, 2020

6 minute read

Introduction

Back in 2018, Will Shroeder, Lee Christensen and Matt Nelson shared their awesome research around Active Directory trusts at DerbyCon. During the last part of their presentation they showed how we can abuse the Print Spooler service in order to force a computer to authenticate against another computer. Lee also released a tool that allows us to do this easily called SpoolSample. If you are not familiar with this attack I highly recommend reading the following blog posts:

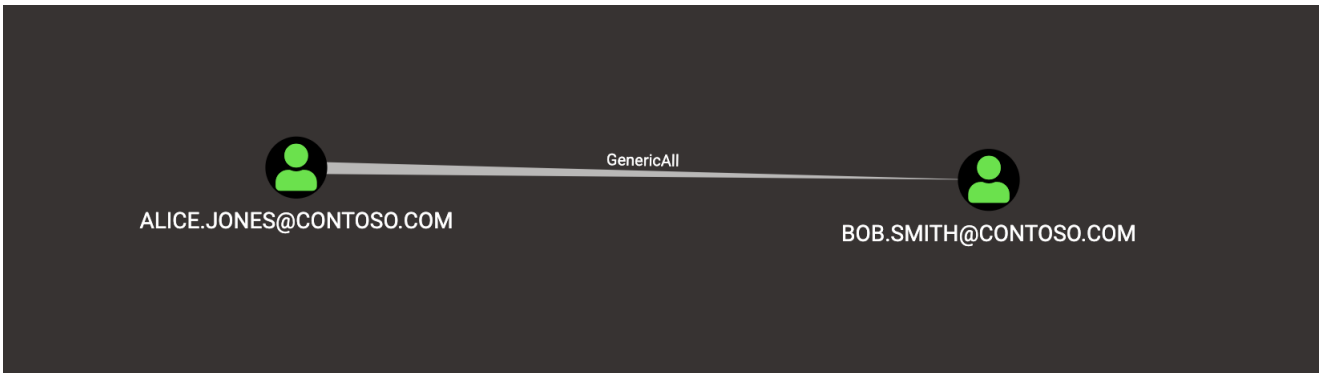
Most of the abuses I have seen so far are using the SpoolSample tool along with compromising a server with Unconstrained Delegation enabled. This allows the attacker to force a computer authenticate back to the attacker using Kerberos and since Unconstrained Delegetation is enabled on the compromised server, the victim also sends their TGT within the TGS. However, there is another way to compromise computers.

If we run the SpoolSample tool with IP addresses as arguments instead of domain names, the target computer will initiate an authentication attempt using Net-NTLM and it is known for many years that Net-NTLM authentication is vulnerable to relay attacks. This means that we can use SpoolSample to make a computer object authenticate back to a computer we control and relay this authentication to another host.

The scenario where this is useful is the following:

OK, lets put this attack aside for now and we will come back to it later.

Another scenario that we find very often when reviewing AD environments is when a user object has rights such as `GenericAll` or `GenericWrite` on another user object similar to the following:



If we have compromised Alice then we can do the following:

- Use a targeted Kerberoasting attack against Bob by setting an SPN and requesting a TGS
- Force a password change for Bob

However, both of these attacks have limitations. For targeted kerberoasting the user must be configured with a weak password in order to crack it. As for changing Bob's password it might be something that you don't want to do during a red team operation to avoid disruption or raising suspicion.

So I started looking at the different attributes a user has and another option is to modify one of the following use attributes:

- homeDirectory - Specifies the home directory of the account and it can be a UNC path.
- profilePath - Specifies a path to the user's profile and it can also be a UNC path.

By modifying any of these attributes, we can point them to a UNC path of a computer under your control and perform an SMB relay attack.

Something you might be wondering so far is this - You have been telling me these ways of exploiting AD objects based on ACL misconfigurations and SMB relay attacks but you haven't told me how to perform a relay attack if all I have is a Cobalt Strike beacon.

Keep reading and your question will be answered ;)

Relay Attacks

So far I have mentioned relay attacks, and specifically SMB relays. This kind of attack has been known for many years. If you want to learn more about SMB relay you can read the following posts:

- https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-under-5-minutes.html
- https://www.sans.org/blog/smb-relay-demystified-and-ntlmv2-pwnage-with-python/

The rest of this post is based on the reader's basic understanding of relay attacks so make sure you have read the above posts.

OK, I understand how SMB relay attacks work. Now what?

When I performed relay attacks in the past I was always doing an internal pentest and I had physical access to the target network. But, why not use these powerful attacks while on a red team operation and you have to do everything through a Cobalt Strike beacon?

It looks like there is a way to do this! Actually this is possible for a few years now thanks to the very cool divertTCPconn and hwfwbypass projects.

DivertTCPconn is based on hwfwbypass and both are written in C++. It is using the amazingly complicated WinDivert project written by Basil:

> https://github.com/basil00/Divert

How does it work?

According to it's description, WinDivert is a kernel driver that allows for user-mode packet interception and modification. The user needs to specify a filter and any packets that match this filter will be intercepted and can be modified.

```
The WinDivert.sys driver is installed below the Windows network stack.  The
following actions occur:

(1) A new packet enters the network stack and is intercepted by WinDivert.sys
(2a) If the packet matches the PROGRAM-defined filter, it is diverted.  The
    PROGRAM can then read the packet using a call to WinDivertRecv().
(2b) If the packet does not match the filter, the packet continues as normal.
(3) PROGRAM either drops, modifies, or re-injects the packet.  PROGRAM can
    re-inject the (modified) using a call to WinDivertSend().
```

The most important thing that WinDivert allows us to do is to intercept traffic going to an open Windows port and redirect it to another port by modifying the TCP source and destination ports of each packet, recalculating the TCP checksums and reinjecting the packets into the network stack.

How does this help us?

On Windows, port 445 is always running by default. I won't go into detail about the process using port 445 because this is already analysed in the following post, so please go ahead and read it:

> https://diablohorn.com/2018/08/25/remote-ntlm-relaying-through-meterpreter-on-windows-port-445/

As mentioned in the above post, it also contains another interesting idea. Using WinDivert to perform an SMB relay attack via Metasploit. You can upload a few DLLs and a driver file to the target host along with the divertTCPconn.exe and execute them. I found this attack to be awesome, but what I didn't like was that you had to upload multiple DLLs on the target host.

So my goal was to do the same attack by dropping the minimum amount of files on disk and also executing the attack through Cobalt Strike.

SMB Relay through Cobalt Strike

First of all, I wanted to make use of Cobalt Strike's `execute-assembly` function so I decided to write my code using the .NET framework. My initial thought would be to re-write divertTCPconn in C# and then everything would work. It turns out that this was very complicated. Fortunately, I found the following NuGet package by TechnikEmpire:

> https://github.com/TechnikEmpire/WinDivertSharp

Using WinDivertSharp, I was able to write a tool called SharpRelay to communicate with the WinDivert driver and perform any packet modification I wanted. The only requirement for this attack to work is to have a beacon with local administrator privileges or with the ability to load drivers. The attack using SharpRelay works as follows:

- Upload the ***signed*** WinDivert driver into any folder on the compromised host
- Run SharpRelay to modify the destination port of the incoming packets on port 445 and redirect them to another port (e.g. 8445)
- From our beacon run the Cobalt Strike's `rportfwd` command to forward port 8445 of the compromised host to our teamserver's port 445.
- Start a socks server to forward the relayed traffic back to the victim network
- Run Impacket's ntlmrelayx with proxychains to do the SMB relay
- When a victim tries to access port 445 of the compromised host the NTLM authentication will be forwarded to our teamserver and relayed to another machine

The code of SharpRelay can be found here:

> https://github.com/pkb1s/SharpRelay

Also, a big part of the code I used for the packet interception was taken from this project by TechnikEmpire:

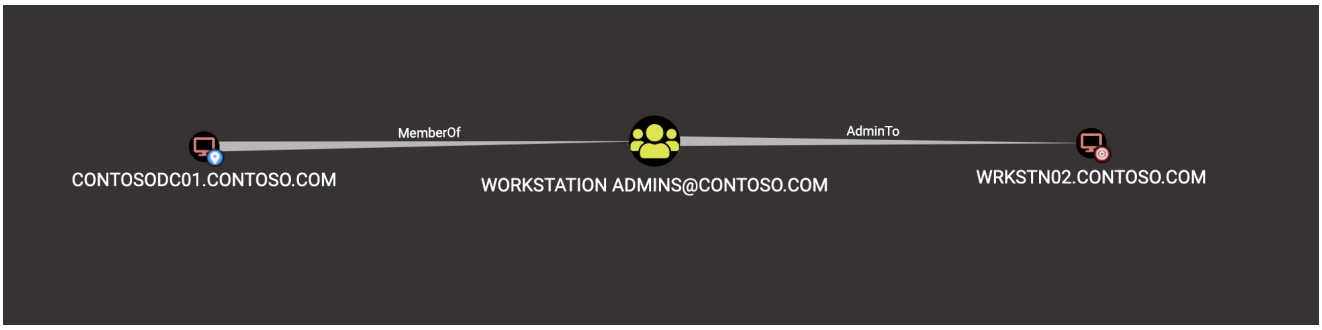
> https://github.com/TechnikEmpire/CitadelCore

Show me a video or it didn't happen

To demonstrate the attacks I described in the beginning of the post, I made the following videos.

SpoolSample to SMB Relay

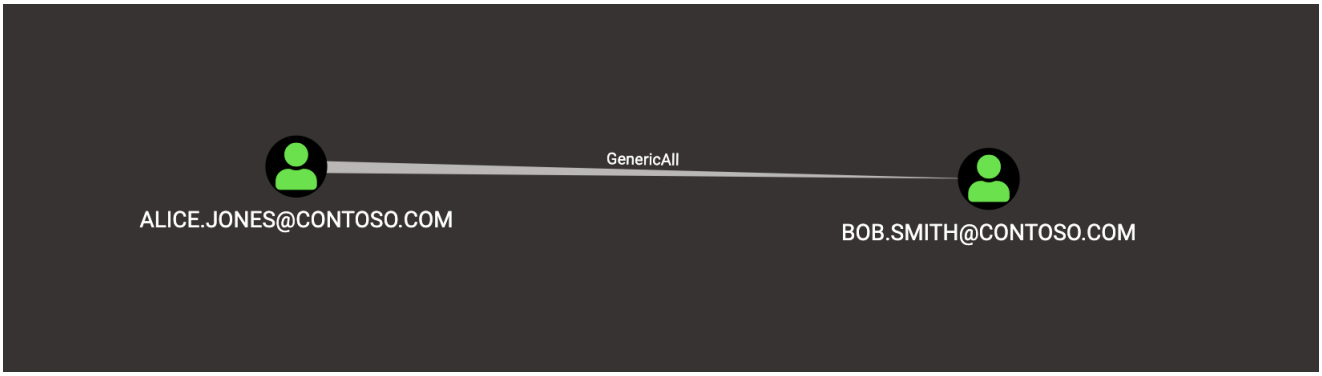
As mentioned earlier we have the following scenario:



The following video demonstrates how we can use the SpoolSample tool to compromise a computer object via an SMB relay attack:

Abusing weak ACLs on a User Object

As shown earlier, the scenario we are going to abuse it the following:



Having a local administrator beacon running as Alice, we will modify Bob's `homeDirectory` attribute and point it to the workstation where we have our beacon running (10.1.1.20). Next time Bob logs in to his workstation he will try to authenticate against the compromised host and we will perform our SMB relay attack: