After the storm - how to move on with NTLM

cyberstoph.org/posts/2021/09/after-the-storm-how-to-move-on-with-ntlm

September 24, 2021

I remember that, about 15 years ago, we already flagged the absence of SMB signing as a vulnerability in reports. Though at that time, we circled more around the theoretical risk of someone tampering SMB traffic due to the lack of integrity protection. None of us really had an idea how to make use of that vulnerability. The later obviously changed.

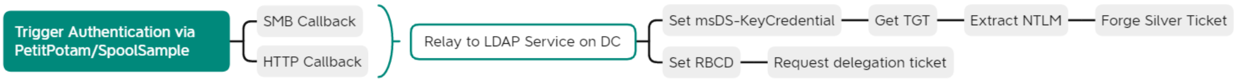
July 2021 came and brought with it a storm of various Print Spooler RCEs (PrintNightmare), one of the simplest LPE's Windows has seen (HiveNightmare,seriousSAM) and a little hippo (PetitPotam) who anchored the "NTLM problem" in the center of our attention. Now as the storm has passed, we're confronted with a familiar question: "how to get rid of NTLM?". And the unhappy truth is, we won't - at least not in the near future.

NTLM as a technology is backed in so deep in the Windows operating system that the only realistic move away from NTLM would be during the transition to a new operating system version (like Windows 11) cause that's how IT departments can handle these kinds of changes. However, it seems that this train has already passed. Windows 11 will bring us a centered start menu and rounded corners but probably nothing new with regards to NTLM. How to move on then?

Short-term strategies

The attack-vector at hand relies on the combination of coercing an NTLM authentication through either SpoolSample/Printerbug or PetitPotam and then forwarding that to a useful target, like the LDAP service of the DC. Why Microsoft tries to center this issue around Active Directory Certificate Services (ESC8), though this is just one possible avenue of attack is beyond my imagination. That information misguides customers into thinking that they are not at risk if they don't have an ACDS web interface running, which is dangerously wrong.

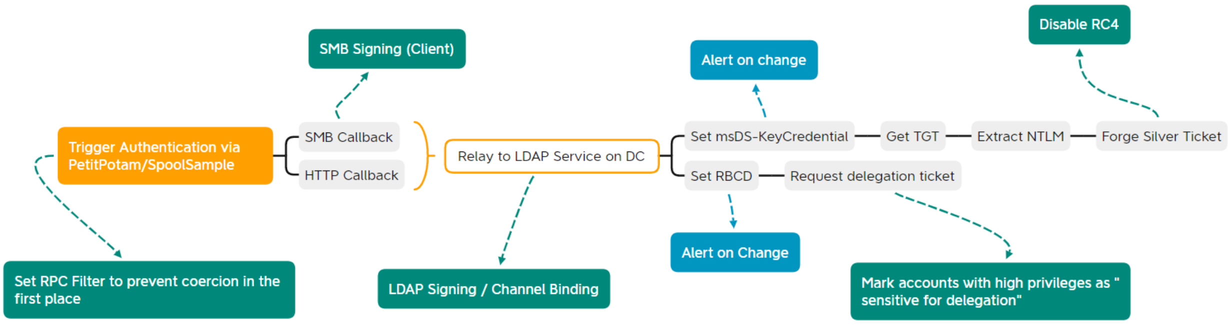
Anyway, the graphic below outlines two possible ways how this attack can play out from start to end.


Trigger Authentication via PetitPotam/SpoolSample
SMB Callback
HTTP Callback
Relay to LDAP Service on DC
Set msDS-KeyCredential
Get TGT
Extract NTLM
Forge Silver Ticket
Set RBCD
Request delegation ticket

The path above uses the `msDS-KeyCredential` attribute to add a certificat to the account which in turn allows to request a TGT via PKINIT. The path below sets the `msDS-AllowedToActOnBehalfOfOtherIdentity` attribute, which configures resource-based constrained delegation. This gives an account of your choosing the right to delegate against the victim, so you can use S4U2Self/S4U2Proxy to get a service ticket in the name of a domain admin and on we go.

If the last two sentences do not make any sense to you, then I recommend this article on RBCD by Will Schroeder and this article on Shadow Credentials by Elad Shamir.

As you can see, regardless of the attack path, there are always a couple of different steps necessary. If you take a closer look at each step of the attack, you'll notice that you can put specific mitigations in place for every step. In fact, defenders have various possibilities to break this attack throughout the way. Let's visualice that and have a look at the graphic again.


SMB Signing (Client)
Alert on change
Disable RC4
Trigger Authentication via PetitPotam/SpoolSample
SMB Callback
HTTP Callback
Relay to LDAP Service on DC
Set msDS-KeyCredential
Get TGT
Extract NTLM
Forge Silver Ticket
Set RBCD
Request delegation ticket
Alert on Change
Set RPC Filter to prevent coercion in the first place
LDAP Signing / Channel Binding
Mark accounts with high privileges as " sensitive for delegation"

The most effective way to break these attacks in the current situation is to prevent coercion of authentication, so you stop the attack at step one. Effective in this case means with regards to the current situation. To quote my history teacher: "there's a difference between the trigger and the source of a problem".

If you remove PetitPotam/SpoolSample from the attack, you just kill the trigger but not the source. The source is the weakness of NTLM against relay attacks and if another PetitPotam is discovered next months, we'll start all over again.

Therefore, the most sustainable measure would be to remove NTLM from the game, but let's discuss this in the next section. Focusing at the first path in the graphic above, you could also break the attack at the last step through disabling RC4 for Kerberos. Since you can only create RC4 silver tickets with an NTLM hash, this would thwart the whole attack path and is also relatively (sigh…) easy to implement if you focus on computer accounts and ignore user-based service accounts for the moment.

The point I want to make here is: if you draw these graphics for all attack paths you can identify in your environment, you can find the choke points that are the easiest for you to implement in short time. This will likely not result in the best solution, but in a solution that works in your indiviudal environment.

Long-term strategies

As mentioned earlier, removing NTLM from the equation entirely is something only Microsoft can initiate, and it seems to me that they have chosen another path. That leaves us with two options to prevent NTLM relay sustainably: 1. enforce signing on all relevant protocols (at least SMB, LDAP, HTTP) 2. disable NTLM for network authentication

Going for signing will probably be easier to achieve if you live in a relatively homogeneous Microsoft environment, but leaves the risk of a future, yet to be discovered, NTLM based attack to break your neck again. Disabling NTLM is better from a long-term perspective but could result in any number of applications breaking. You might also notice that some very common Microsoft applications rely heavily on NTLM by default (talking about you, Exchange!).

My recommendation would be to set this up as an internal project or whatever the best formal definition in your environment is to get a hand on manpower and do two things:

first, build awareness that NTLM needs to die (probably find better wording). This is our stretch goal. It won't happen any time soon but it's good to know where we're heading. Make sure that any new application works without NTLM e.g., by disabling NTLM via GPO on all new servers you bring to production.

Second, enforce signing step-by-step. Start with LDAP, since this is the most relevant vector for the attacker. If you can remove LDAP from the equation, you remove the attacks that are possible by design like RBCD (see previous section), which should tremendously reduce the attack surface. This buys you time to implement SMB signing and you'll need that time since SMB signing almost certainly breaks stuff.

Regardeless of what you do though, the most important thing is to start now. It's time to get rid of that evil called NTLM who is whipping our assess since Hernan Ochoa released WCE 10 years ago. Yes - TEN YEARS AGO! The storm is over…let's rebuild :-)