ALSID ACADEMY

Episode 8

DCSHADOW EXPLAINED: A TECHNICAL DEEP DIVE INTO THE NEW AD ATTACK

On January 24, 2018, security researchers Benjamin Delpy and Vincent Le Toux announced a new attack technique targeting Active Directory at the BlueHat IL security conference. Named DCShadow, this attack allows an attacker with certain rights to create a rogue domain controller and replicate malicious objects in an AD infrastructure.

Here, we'll explain the technical foundations of the attack, its impact on Active Directory security, and how blue teams can detect it.

WHAT IS THE DCSHADOW ATTACK, AND WHY IS IT DIFFERENT?

The ability to obtain user and computer credentials while avoiding detection countermeasures is the holy grail for red teams or attackers wanting to compromise Active Directory. For this purpose, several attack techniques have been developed: LSASS injection, abusing Shadow Copy, NTFS volume parsing, ESE NT operations, sensitive attribute manipulation, etc. More info is available in the impressive synthesis from ADSecurity.org.

Hidden among these noisy attacks is one connected to DCShadow. Introduced in 2015, the DCSync attack relies on the ability of domain admins or domain controllers to ask a domain controller (DC) for data replication. (The GetChangesAll permission, granted by default to administrative accounts and DCs, was necessary to achieve this task). As described in the MS-DRSR specification for domain controller replication, these groups can request that the domain controller replicate AD objects (including user credentials) through the GetNCChanges RPC. Further technical details on the attacks are available on ADSecurity.org.

One of the main limitations of the DCSync attack is that it is impossible for an attacker to inject new objects into the targeted AD domain. Of course, this attacker could take ownership of an administrative account using the good old pass-the-hash technique and inject objects afterwards, but it requires more effort and more steps, meaning a greater probability of being busted by blue teams. The DCShadow attack removes this limitation by reversing the DCSync attack paradigm.

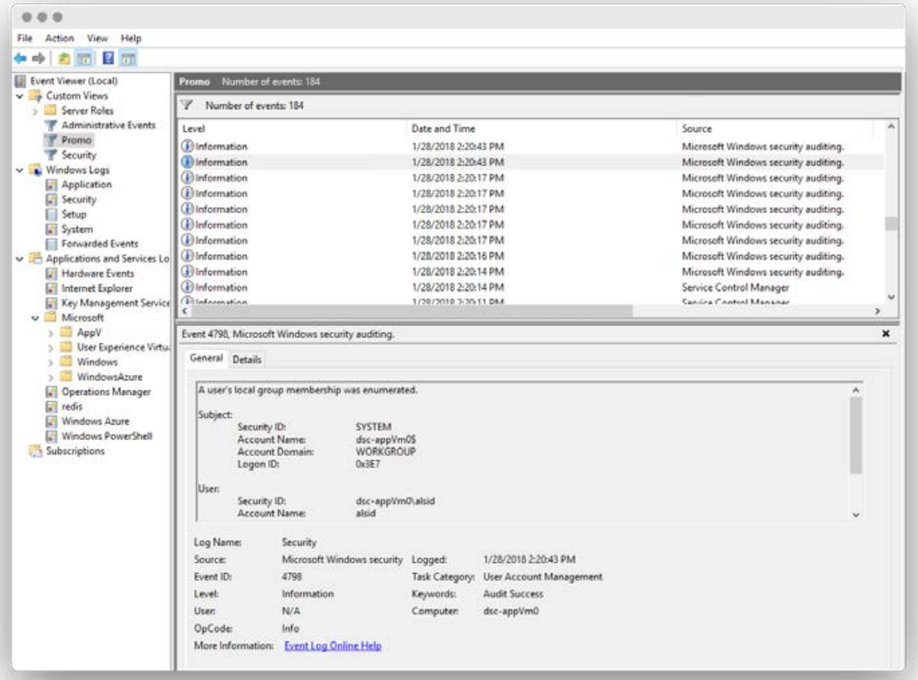
DCShadow attackers no longer try to replicate data but will register new domain controllers in the targeted infrastructure to inject Active Directory objects or alter existing ones (by replacing the attributes' content). The idea of a rogue domain controller is not new and has been mentioned in previous security publications, but it required invasive techniques (like installing a virtual machine with Windows Server) and logging on to a regular domain controller to promote the VM into a DC for the targeted domain. Not very discrete.

DCSync attack with mimikatz tool

```
mimikatz # lsadump::dcsync /user:Administrator
[DC] 'esaf.alsid.corp' will be the domain
[DC] 'WIN-28TBTC00O8E.esaf.alsid.corp' will be the DC server
[DC] 'Administrator' will be the user account

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   : 1/1/1601 1:00:00 AM
Password last change : 12/9/2017 6:37:07 PM
Object Security ID   : S-1-5-21-2991865491-4214911763-577543855-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: 520126a03f5d5a8d836f1c4f34ede7ce
    ntlm- 0: 520126a03f5d5a8d836f1c4f34ede7ce
    ntlm- 1: 0f05fa36dc8659611b411796c9b0bfbf
    ntlm- 2: 520126a03f5d5a8d836f1c4f34ede7ce
    lm  - 0: e4eb0c7067f571d8f32b61a865ff05c3
    lm  - 1: e40cac85b0ca5315a327ac9e19f0744d

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : fb15e65266d1c295f723bb22081c36b8

* Primary:Kerberos-Newer-Keys *
    Default Salt : ESAF.ALSID.CORPAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 58c4700189429cf52e25fdef586a26c194b4b21c1a545468bf53fb08b1260bee
      aes128_hmac       (4096) : bf6c097af3ac60e74e7dddf50952ade3
      des_cbc_md5       (4096) : 5d9167ea73f8838a
    OldCredentials
      aes256_hmac       (4096) : d7ee0c49bfa5a164bb3e584a5f6bf1745b44aa315df54d9641112e96bcc35fef
      aes128_hmac       (4096) : ff6e242b6d2b54f9e7b7625b1a77edce
      des_cbc_md5       (4096) : 52ba3b1979fd04b5
```

- Event log generated during a regular DC promotion

To understand the genius behind DCShadow, it is important to grasp what a domain controller is and how it is registered in the Active Directory infrastructure



Understanding what a domain controller is

As described in MS-ADTS (Active Directory Technical Specification), Active Directory is a multi-master architecture relying on dedicated services. The DC is the service (or the server hosting this service depending on your point of view) which hosts the data store for AD objects and operates with other DCs to ensure that a local change to an object replicates correctly across all DCs.

When a DC is operating as RW DC, the DC contains full naming context (NC) replicas of the configuration, the schema, and one of the domain naming contexts of its forest. In this way, every RW DC holds all objects of a domain, including credentials and all manner of secrets (like private or session keys). As such, there is no need to remind that DCs are the one and only element blue teams should focus protecting (administrative accounts or permissions are just two of the many ways to access a DC).

Services provided by a domain controller

Describing in detail the technical ways and means of a DC is complex and will not adequately explain the purpose of the DCShadow attack. To be concise, a server can be called a domain controller if it offers the following 4 key components:

- a database engine able to replicate its information (meaning it must be accessible through LDAP protocols and implement several RPCs to follow MS-DRSR and MS-ADTS specifications)
- an authentication provider accessible through Kerberos, NTLM, Netlogon, or WDigest protocols
- a configuration management system called GPO, relying on SMB and LDAP protocols
- an (optional) DNS provider used by clients to locate resources and support authentication

Synthesis of services provided by a DC

A DATA ENGINE

NTDS through LDAP and RPC

Host the **domain information** and configuration using abstract objects. It is accessible throught the LDAP and RPC protocols

AN AUTHENTICATION SERVICE

MS Kerberos

Implement a single-sign-on **authentication protocol** using ticket papadigm. Kerberos is used each time a user **authenticates to a service** (website,mailbox, file share, etc.)

A POLICY SERVICE

Group Poliicy engine (SMB + LDAP)

Manage resources (users, computers, services) of a domain. **Security policies** are deployed using GPO

A POLICY SERVICE

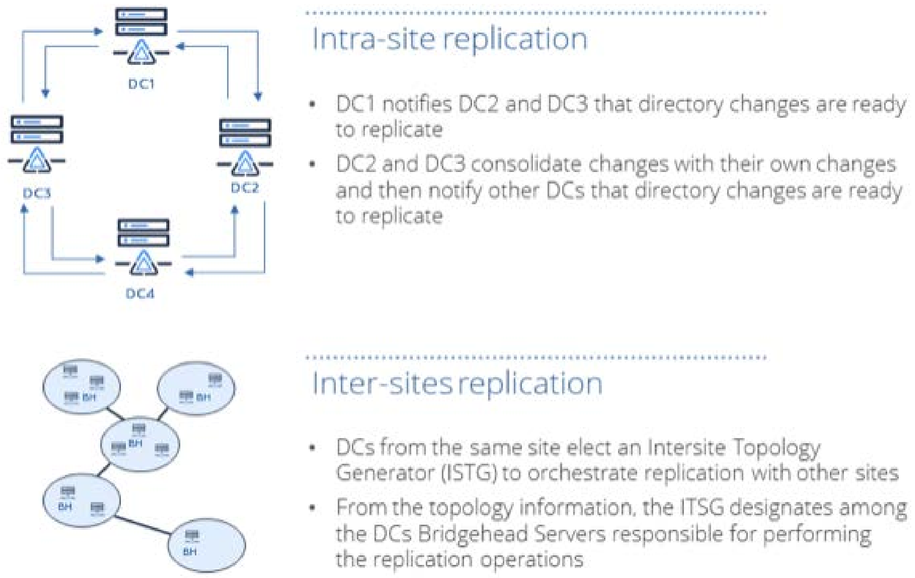
Group Poliicy engine (SMB + LDAP)

Locate the resources in the corporate network and compute the AD topology (computer.domain.corp,DOMAIN\user, user@domain.corp)

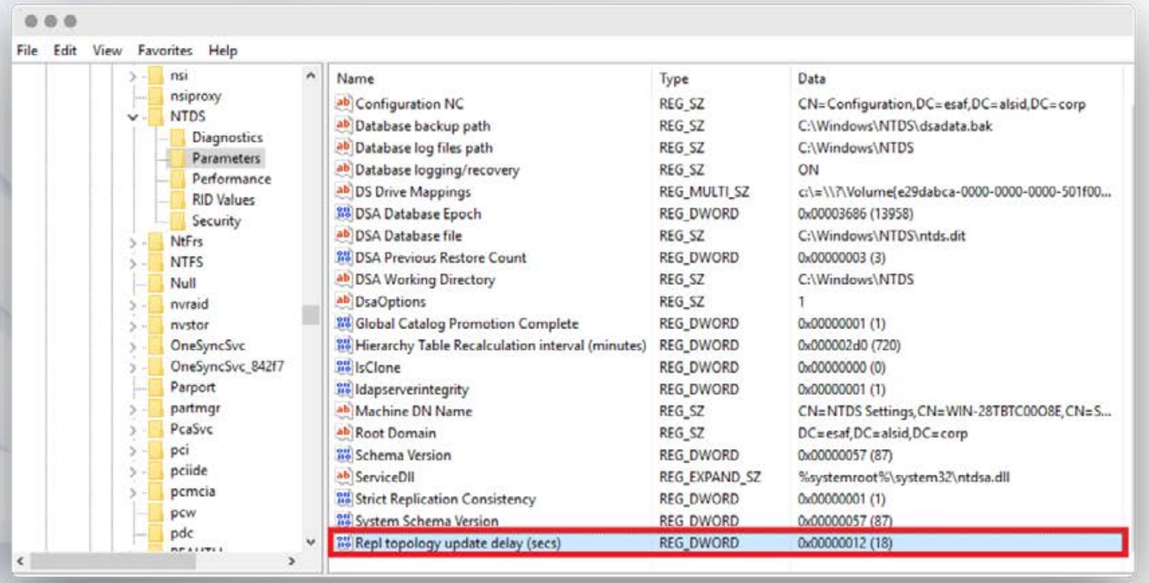
FOCUS ON ACTIVE DIRECTORY REPLICATION

In addition to hosting these services, a domain controller in the making should be registered in the directory infrastructure to be accepted by another DC as a replication source provider. The data replication is orchestrated by a built-in process (running on the NTDS service) called the Knowledge Consistency Checker (KCC).

The major function of the KCC is to generate and maintain the replication topology for replication within and between sites. In other words, the KCC process elects which DC will communicate with others to create an efficient replication process. Within a site, each KCC generates its own connections. For replication between sites, a single KCC per site generates all connections. The following schema illustrates the two kinds of replication.



- The two types of replication process



By default, the KCC initiates AD replication topology every 15 minutes to ensure consistent and regular propagation. Using the USN associated with every AD object, the KCC recognizes changes that occur in the environment and ensures that domain controllers are not orphaned in the replication topology. Fun fact: Active Directory replication processes could historically be executed through both RPC (like DrsAddEntry) and SMTP (for the Schema and Configuration partition only)!

- Registry key defining the replication time period

The brilliance of the researchers behind DCShadow was in identifying the minimal set of changes required to inject a new server into the replication topology and thereby inject malicious information to abuse this process while remaining stealthy.

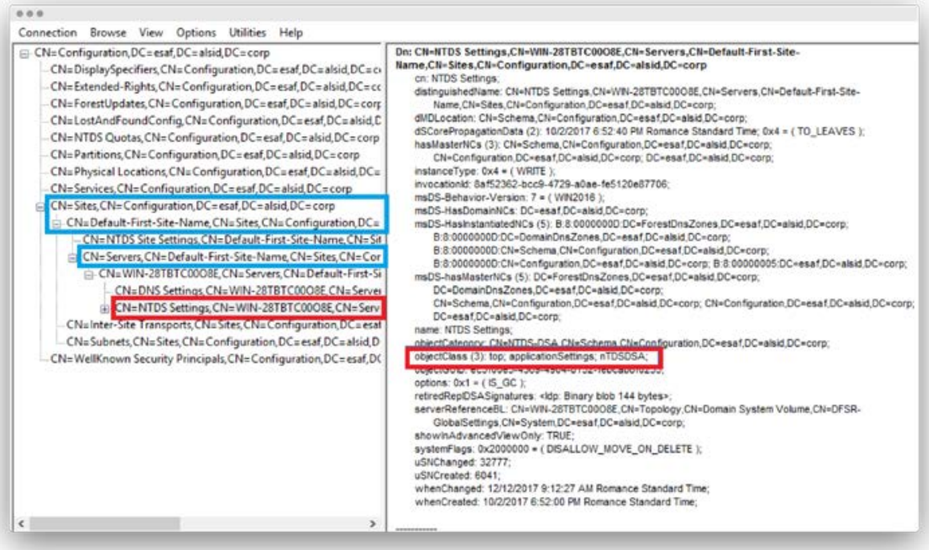
HOW DCSHADOW ACTUALLY WORKS

The DCShadow attack aims to register new domain controllers to inject malicious AD objects and so create backdoors (or any kind of illegitimate access or right). To attain this goal, the DCShadow attack must modify the targeted AD infrastructure database to authorize the rogue server to be part of the replication process.

Register a new domain controller

As mentioned in the MS-ADTS specification, a domain controller is represented in the AD database by an object of class nTDSDSA which is always located in the configuration naming context of a domain. More precisely, each DC is stored in the site's container (object class sitesContainer), as a child item of a server object.

- In blue, the containers storing the NTDS-DSA object. In red, the object itself.

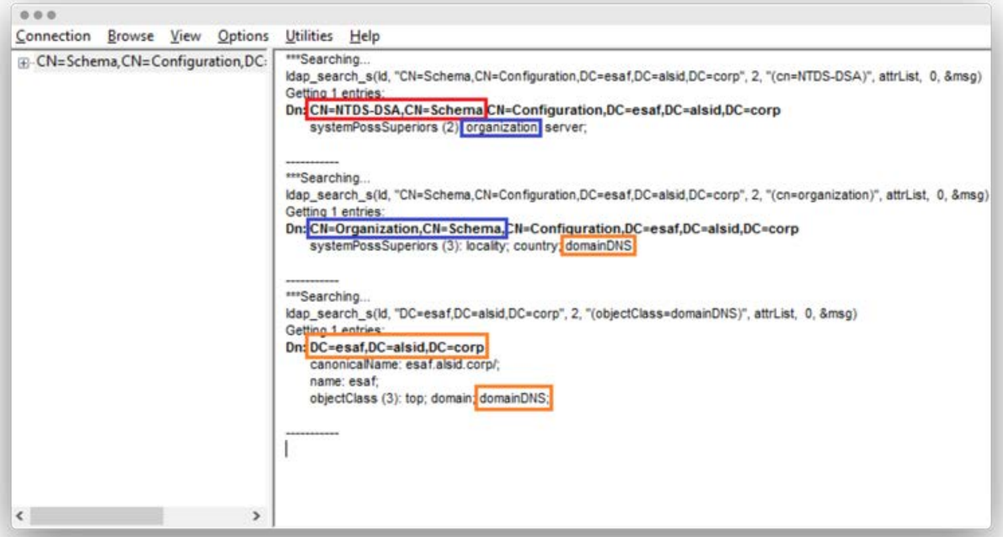


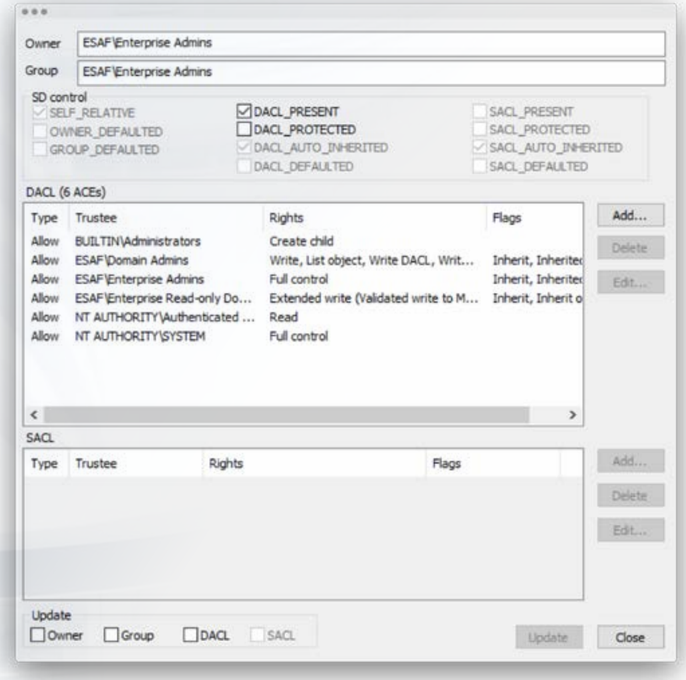
A quick look at the schema shows that NTDS-DSA objects can only be created as children of server objects, which in turn can only be part of organization or server objects:

- The server objects can only be stored in serversContainer objects which are only found in the Configuration NC
- The organization objects can only be stored in locality, country, or domainDNS objects, which can be found in the domain NC

- The schema indicates where ntds-dsa objects can be created

Thus, domain controllers (nTDSDSA objects) can only be created in the Configuration or Domain NC. In practice, it seems only the nTDSDSA objects stored in the site container (sitesContainer object) are taken into consideration. As the KCC relies on the site information to compute its replication topology, it is logical that only these objects are used. Note that creating an nTDSDSA object is not possible using the LDAP protocol.





Thus, domain controllers (nTDSDSA objects) can only be created in the Configuration or Domain NC. In practice, it seems only the nTDSDSA objects stored in the site container (sitesContainer object) are taken into consideration. As the KCC relies on the site information to compute its replication topology, it is logical that only these objects are used. Note that creating an nTDSDSA object is not possible using the LDAP protocol.

- The default access rights on the Server object

The main action of the DCShadow attack is to create a new server and nTDSDSA objects in the Configuration partition of the schema. Doing so provides the ability to generate malicious replication data and inject them into other domain controllers.

Now that we understand what the DCShadow attack does, we need to understand what kind of privileges are required to create nTDSDSA objects in the Configuration partition. A quick look at the permissions show that only `BUILTIN\Administrators`, `DOMAIN\Domain Admins`, `DOMAIN\Enterprise Admins`, and `NT AUTHORITY\SYSTEM` have control rights over the targeted containers.

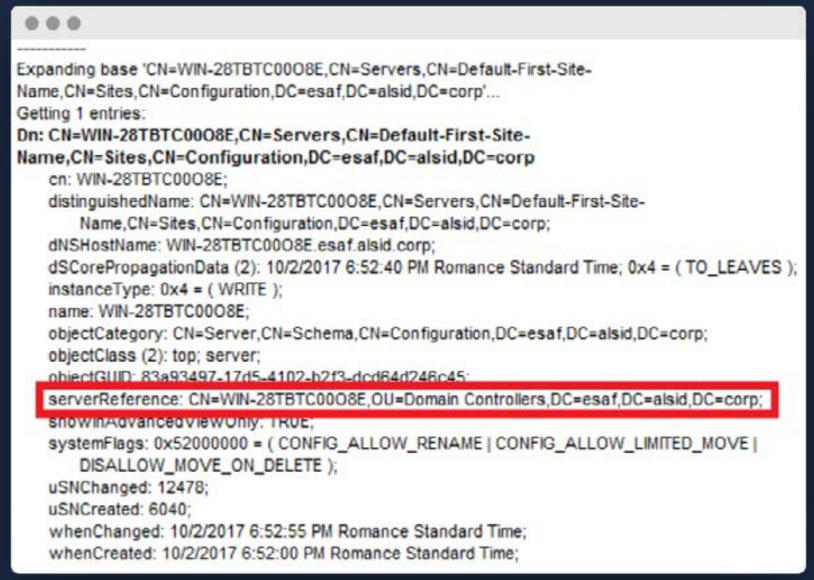
This quick analysis allows us to conclude that the DCShadow attack is not a privilege escalation vulnerability, but a mechanism for the misappropriation of Active Directory. It doesn't allow red teams to gain privileges. Instead, it provides them another solution to become persistent or take illegitimate actions in a directory infrastructure. It should thus be chalked up as another sneaky AD persistence trick and not as a vulnerability to fix.

TRUST THE NEW DOMAIN CONTROLLER

As described in the previous paragraph, the DCShadow attack relies on the addition of a new nTDSDSA object in the Configuration partition to register itself as a new member of the replication process. However, adding this sole object is not enough to allow our rogue server to initiate replication. To be part of the replication process we need to take care of two requirements:

- be trusted by other servers, meaning that we need to have valid authentication credentials
- provide authentication support to let other DCs connect to our rogue server when we need to replicate data

By using a valid computer account, a rogue server can be treated as a trustworthy AD server. The Kerberos SPN attributes will provide authentication support for other DCs. Therefore, every nTDSDSA object is linked to the computer object through the serverReference attribute.



- The serverReference attribute acts as the link between a nTDSDSA object and its related computer object

Despite the theoretical possibility of achieving this with a user account, it seems much easier and stealthier to use a computer account. In fact, it will be automatically registered in the DNS infrastructure (which will allow other DCs to locate our resource). Moreover, it will natively have the required attributes set and will have its authentication secret automatically managed.

The DCShadow attack will use a legitimate computer account to authenticate to other DCs. Although the computer object and the nTDSDSA object will bring the ability to authenticate to other DCs, the DCShadow attack still needs to let other DCs connect to the rogue server to replicate illegitimate information from it.

This last requirement is fulfilled using the Kerberos Service Principal Name (SPN). As explained extensively in several publications, SPNs are used by the Kerberos service (KDC) to encrypt the Kerberos ticket with the account associated with the SPN. The DCShadow attack will add SPNs to the regular computer object used to authenticate.

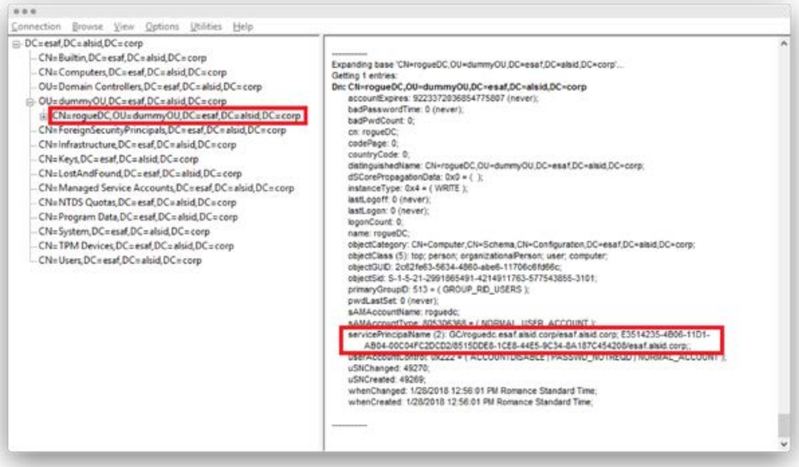
Researchers Benjamin Delpy and Vincent Le Toux succeeded in isolating the minimum set of SPNs required for the replication process to execute. Their results show that two SPNs are required to let another DC connect to the rogue server:

- The DRS service class (which has the well-known `GUID E3514235–4B06–11D1-AB04–00C04FC2DCD2`)
- The Global Catalog service class (which has the string `“GC”`)

For example, the two SPNs required by our rogue server (`named “roguedc”` with the `DSA GUID 8515DDE8–1CE8–44E5–9C34–8A187C454208` in the `alsid.corp domain`) are as follows:

```
E3514235–4B06–11D1-AB04–00C04FC2DCD2/8515DDE8–1CE8–44E5–9C34–8A187C454208/alsid.corp
GC/roguedc.alsid.corp/alsid.corp
```

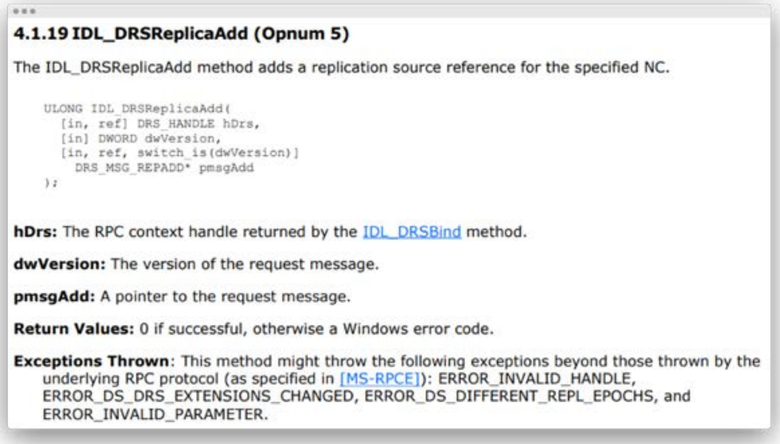
When triggering its attack, DCShadow will set those two SPNs to its targeted computer account. More precisely, the SPNs will be set using the DRSAddEntry RPC function as described in the CreateNtdsDsa function documentation (more details about MS-DRSR RPC are covered in the next section).



- A rogue computer account having the SPN of a DC

When triggering its attack, DCShadow will set those two SPNs to its targeted computer account. More precisely, the SPNs will be set using the DRSAddEntry RPC function as described in the CreateNtdsDsa function documentation (more details about MS-DRSR RPC are covered in the next section).

- Extract from the MS-DRSR specification describing the DRSReplicaAdd IDL



4.1.19 IDL_DRSReplicaAdd (Opnum 5)

The IDL_DRSReplicaAdd method adds a replication source reference for the specified NC.

```
ULONG IDL_DRSReplicaAdd(
  [in, ref] DRS_HANDLE hDrs,
  [in] DWORD dwVersion,
  [in, ref, switch_is(dwVersion)]
    DRS_MSG_REPADD* pmsgAdd
);
```

hDrs: The RPC context handle returned by the IDL_DRSBind method.

dwVersion: The version of the request message.

pmsgAdd: A pointer to the request message.

Return Values: 0 if successful, otherwise a Windows error code.

Exceptions Thrown: This method might throw the following exceptions beyond those thrown by the underlying RPC protocol (as specified in [MS-RPCE]): ERROR_INVALID_HANDLE, ERROR_DS_DRS_EXTENSIONS_CHANGED, ERROR_DS_DIFFERENT_REPL_EPOCHS, and ERROR_INVALID_PARAMETER.

For now, we can register our rogue domain controller with the replication process and be authenticated by another DC. The remaining step is forcing the DC to initiate the replication process with our malicious data.

Injecting illegitimate objects

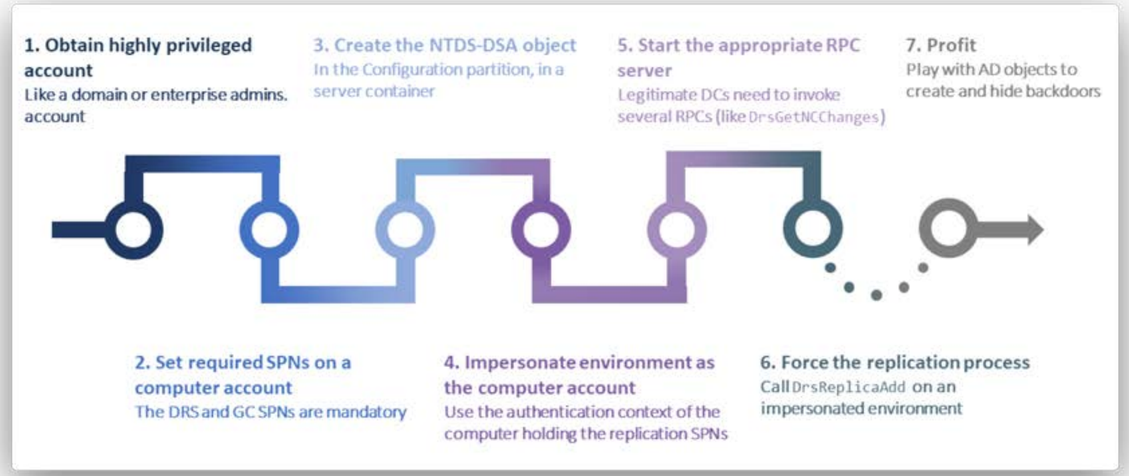
In this final section, we study how the DCShadow attack injects its illegitimate information into the DNS infrastructure.

To serve illegitimate data, the rogue domain controller must implement the minimal set of RPC functions required by the MS-DRSR specifications: `IDL_DRSBind`, `IDL_DRSUnbind`, `IDL_DRSGetNCChanges`, `IDL_DRSUpdateRefs`. The IDL of this function is provided by Microsoft in its open specifications and is now implemented in Benjamin Delpy's Mimikatz tool.

The final step of the DCShadow attack is triggering the replication process. To do so, two strategies can be conducted:

- Wait for the KCC process of another DC to initiate the replication process (requires a 15-minute delay)
- Force the replication by invoking the DRSReplicaAdd RPC function. It will change the content of the repsTo attribute that will start an immediate data replication.

Forcing the replication with the IDL_DRSReplicaAdd RPC is the last step of a DCShadow attack. It allows injecting arbitrary data into a targeted AD infrastructure. In so doing, it becomes trivial to add any backdoor in the domain (by adding new members on an administrative group or by setting SID history on a controlled user account, for example).



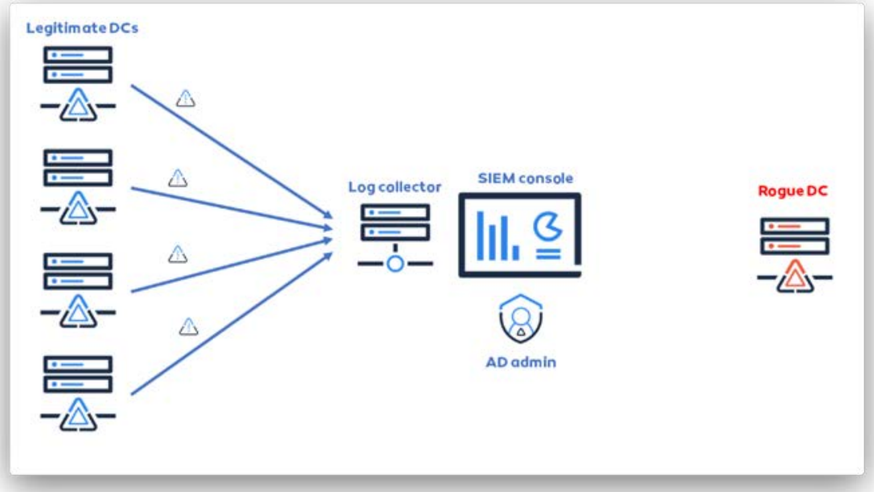
Process summary
The following chart summarizes the different operations achieved during a DCShadow attack.

- DCShadow attack process, synthesized

THE CONSEQUENCES OF DCSHADOW FOR BLUE TEAM STRATEGIES

As explained in this research paper, blue teams in charge of AD security monitoring usually rely on event log collection. Computers that are members of a domain are configured to push their logs to a central SIEM to be analyzed.

- A simplified SIEM architecture pushing event log through WinRM Event Forwarding protocol



The first problem with this approach is that only legitimate computers send their logs to the log collector. During DCShadow, the event logs related to the injection of new data are only created on the attacker's machine, which obviously will not signal itself by sending events to the SIEM. This way, the DCShadow attack can be stealthy, as only a few event logs will be generated by legitimate computers.

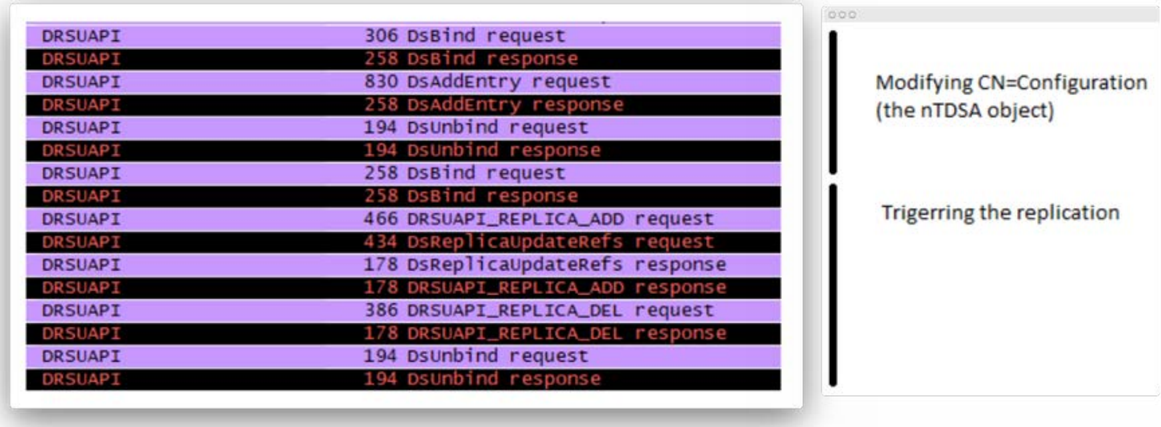
Several prerequisite actions should be taken before injecting the rogue data information into the targeted AD. Unfortunately, the AD modifications involved in setting up a rogue DC are rarely included in logging policies. For example, Configuration NC changes are almost never considered. While it is possible to be alerted of such changes, determining if such an event was caused by malicious activity or regular AD operations is time-consuming and impractical.

Blue teams need a complete redesign of their strategy to shift their focus from log analysis to AD configuration analysis. The naïve approach would be to monitor replications (DrsGetNCChanges RPC changes). By default, a SACL entry set on the root object of the domain logs the use of extended rights except for domain controllers. Thus, replication with a user account or non-DC machine is easily identified. We do not believe, however, that this method is the most efficient. Three strategies should be implemented to detect DCShadow attacks:

1. The Configuration partition of the schema should be scrutinized. nTDSDSA objects in the site's container should be matched with regular domain controllers in the domain controllers' organizational unit (or better, a list of known DC manually maintained by the administration team). Any object showing up in the first but not the second should be investigated. Please note that the rogue nTDSDSA object is removed right after the publication of the illegitimate object. For efficiency, detection measures should be able to catch object creation.
2. As demonstrated, DCs need an authentication provider. To push changes, a rogue DC must have one accessible through Kerberos with a specific service. In practical terms, this means having a Service Principal Name (`SPN`) beginning with the "`GC/`" string. The well-known RPC interface GUID "`E3514235-4B06-11D1-AB04-00C04FC2DCD2`" can also be used. Computers having this service but not present in the DC OU should also be carefully investigated.
3. Using DCShadow requires an attacker to have elevated privileges. Analyzing and monitoring the permissions present in the Configuration partition will allow blue teams to ensure no one is able to alter it except legitimate administrators. Any DACL granting access to a non-privileged entity can also be a sign of a possible backdoor.

UNCOVER DCSHADOW

Despite the lack of event logs, there are several ways to detect DCShadow. As a first step, we can consider network detection. As DCShadow relies on replication, it generates characteristic network patterns, as illustrated in the following figures.



- It requires monitoring every domain controller, even if you have dozens of them. Miss one, and you are blind.
- There are several sneaky ways to inject illegitimate data without calling DRSReplicaAdd

- You must tap/duplicate the whole traffic of a very sensitive infrastructure

A more elaborate approach would be to monitor the replication of Active Directory objects to identify suspicious patterns.

DCShadow requires creating several objects in a directory infrastructure, and Active Directory offers several ways to be informed when such an event occurs (without requiring any administrative rights).

The basic idea is to detect the creation of the nTDSDSA object and the set of the `SPN E3514235-4B06-11D1-AB04-00C04FC2DCD2` on an illegitimate machine using the replication process or notification.

To illustrate this approach, Alsid has released a series of proof-of-concepts named UncoverDCShadow to help blue teams detect DCShadow attempts. Developed in PowerShell, they can be easily connected to a SIEM infrastructure to help it detect such an attack.

UncoverDCShadow uses the ability to make asynchronous calls to the AD database using LDAP. With the well-known (or not-so-well-known) LDAP server control `LDAP_SERVER_NOTIFICATION_OID (1.2.840.113556.1.4.528)`, any user can receive information about any created, modified, or deleted object of the entire Active Directory database. Detecing the use DCShadow becomes simple.

More detail about how UncoverDCShadow works are available here.

FINAL THOUGHTS

The most important takeaway from this analysis is that DCShadow is not a vulnerability, rather an innovative way to inject illegitimate data into an AD infrastructure.

No unprivileged attacker will ever be able to use it to escalate their privileges and gain administrative access to your AD using DCShadow. Bottom-line: if your AD is properly configured and secured, you do not need to take any urgent actions.

DCShadow does not require any immediate patching campaign or special configuration to be applied. This has nothing to do with WannaCry/NotPetya incident response.

Not being a vulnerability, DCShadow will not be patched by a Microsoft update. Trying to counter it would mean changing the way AD works, and hence break the system. The authors of the research previously published the DCSync attack, and Microsoft did not issue any patch as it only uses legitimate APIs. “Fixing” it would mean forbidding DC replication. If it ain’t broke, don’t fix it. AD is not broken.

However, the fact that a new attack method is publicly available for anyone to use needs to be considered. It offers an extremely stealthy way for privileged attackers to perform actions, so detection strategies should be updated to reflect this new threat. Traditional event log analysis methods will probably fail to detect DCShadow usage. Efficiently detecting this attack technique, requires continuously monitoring the AD database to isolate illegitimate changes. This is the Alsid solution, and we are proud to already protect our customers against this attack. For more information on how we tackle this challenge, head to www.alsid.com.