Watching the Watchdog Protecting Kerberos

Tal Be'ery, Sr. Security Research Mgr.
Michael Cherny, Sr. Security Researcher

Microsoft

Speaker Info – Tal Be'ery

- Senior Security Research Manager @Microsoft
- Former VP for Research @Aorato (Acquired by Microsoft)
- 15 years of security research
- Author of the TIME attack on SSL
- Regular speaker in Industry's top conventions
- Named a "Facebook Whitehat"
- Twitter: @TalBeerySec

Speaker Info – Michael Cherny

Agenda

- Intro
  - Kerberos authentication
  - Kerberos attacks: from stealing to forgery
  - Network monitoring
- Advanced forgery attacks against Kerberos
  - Forged Key: The Skeleton Key malware attack and detection
  - Forged Ticket: The Golden Ticket attack and detection
  - Forged PAC: MS14-068 attack and detection
  - Forged PAC: Diamond PAC attack and detection
- Conclusions

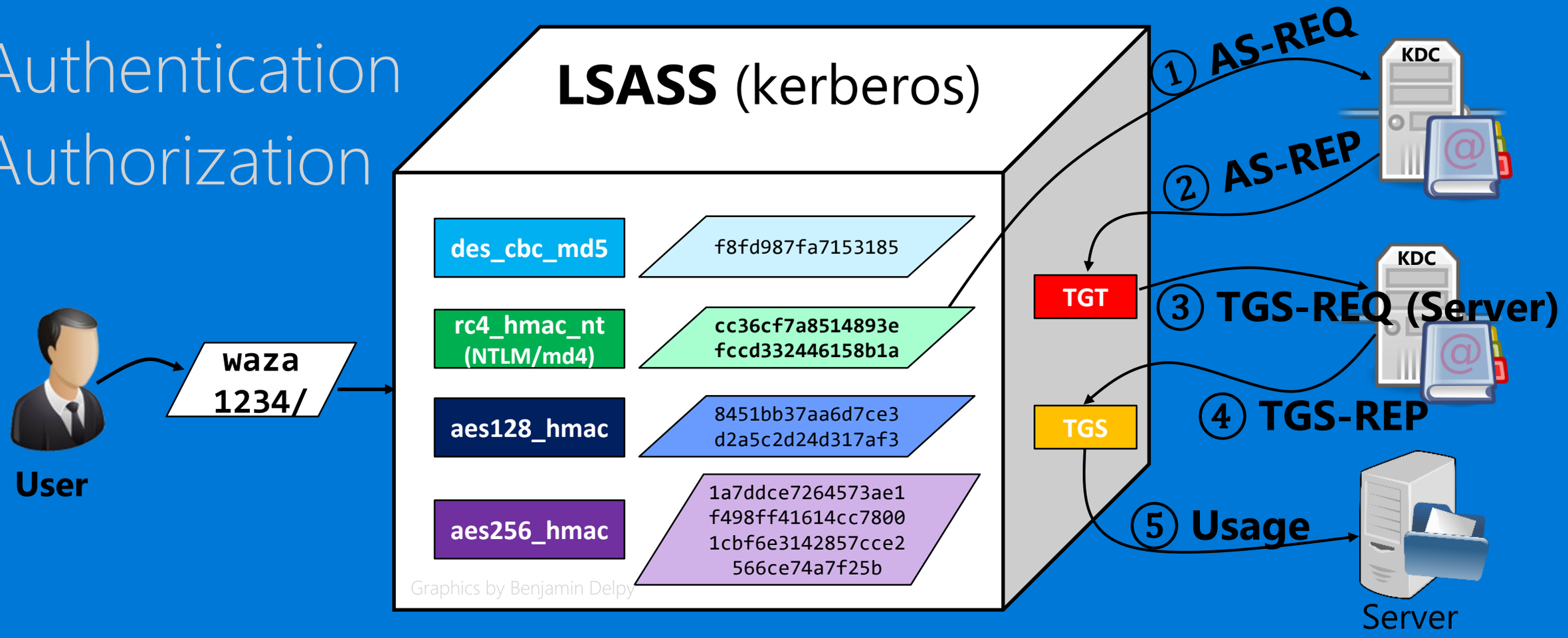
Intro

Attackers need access!

- Attackers need to move from their initial infection point to the their target = the data
- Moving requires
  - Authentication – user's identity
  - Authorization – user's privileges
- Kerberos is the default authentication and authorization protocol for Windows (and other OSs)
- Therefore, attackers must abuse the Kerberos protocol!

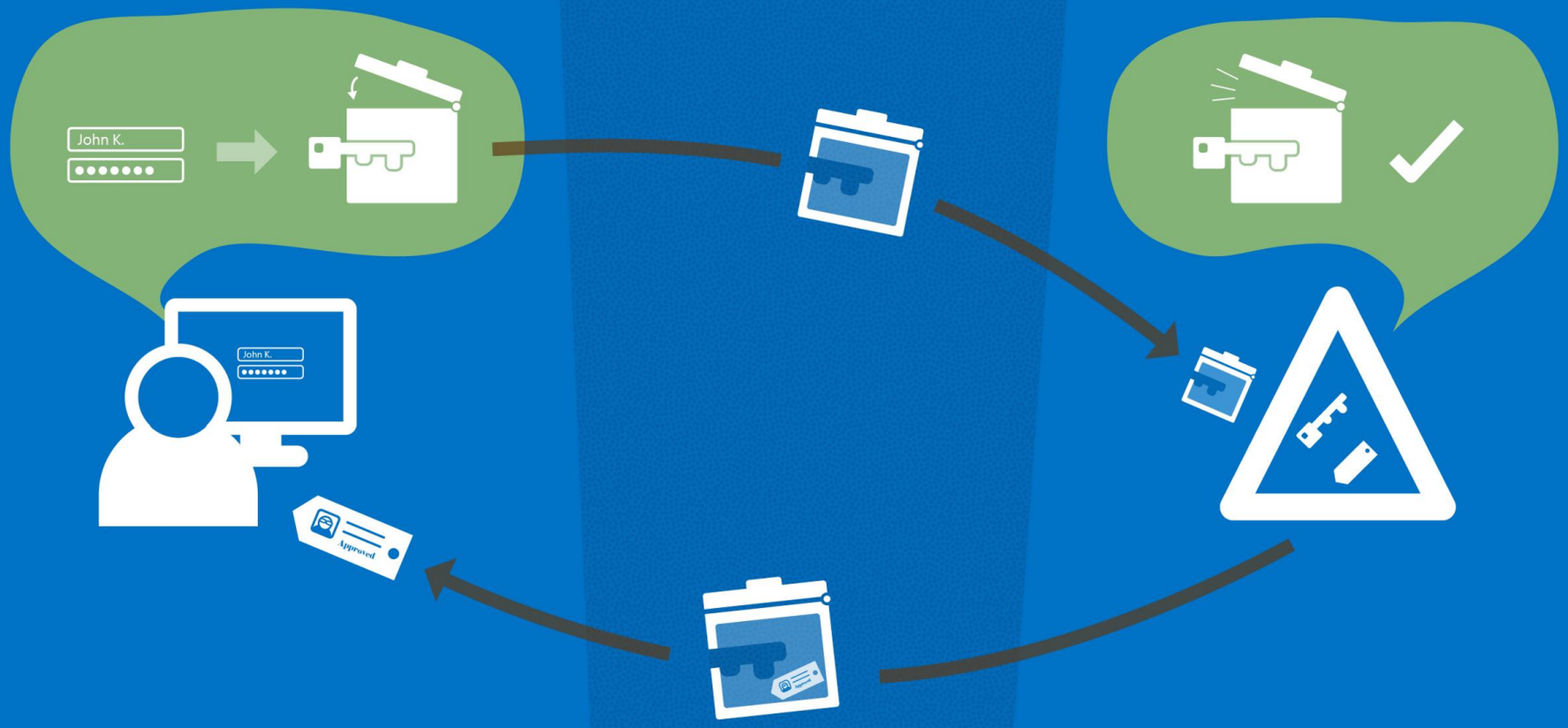
Kerberos

- Authentication
- Authorization

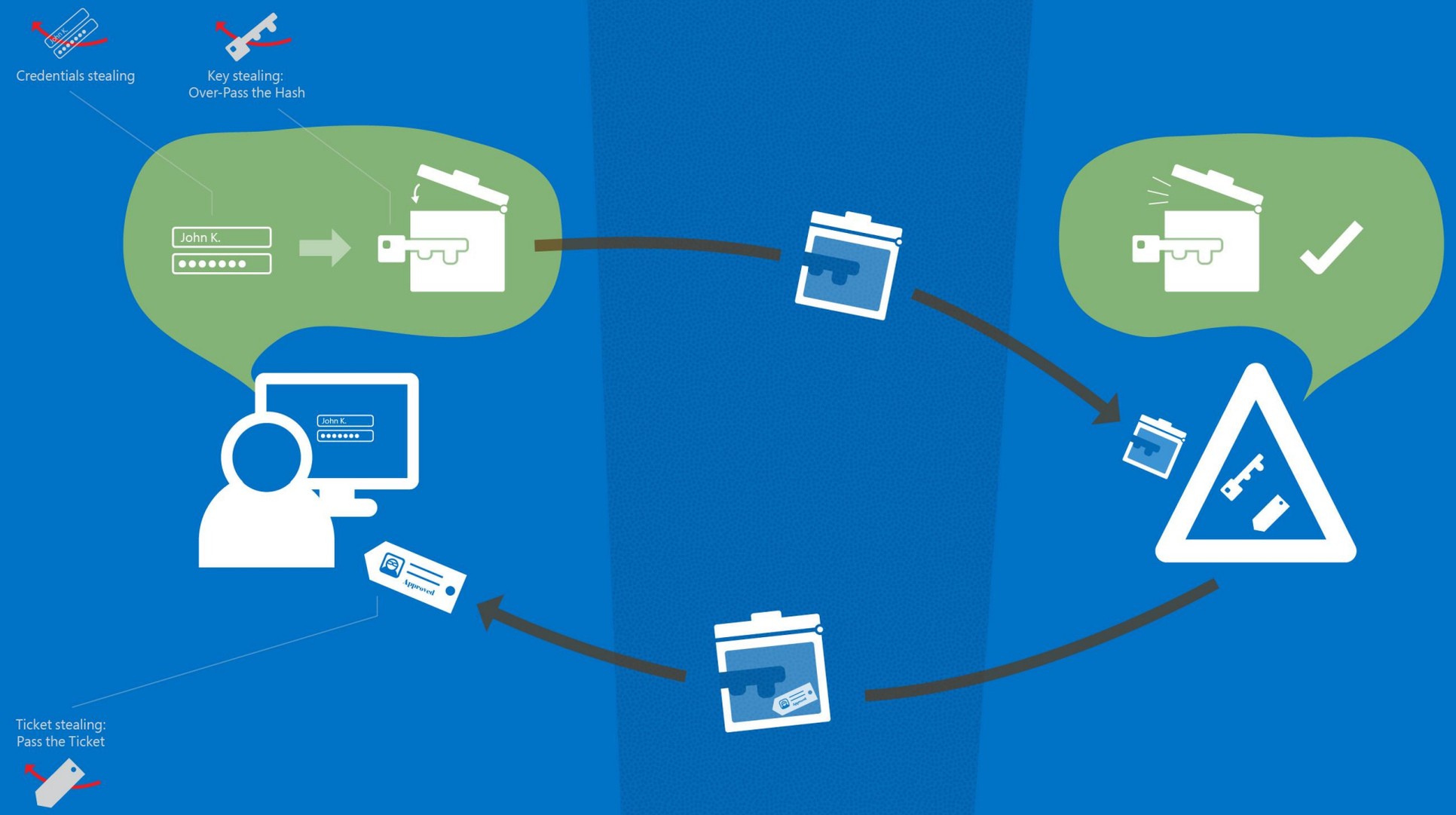


- Standard (RFC4120)

Kerberos: From Creds to Ticket



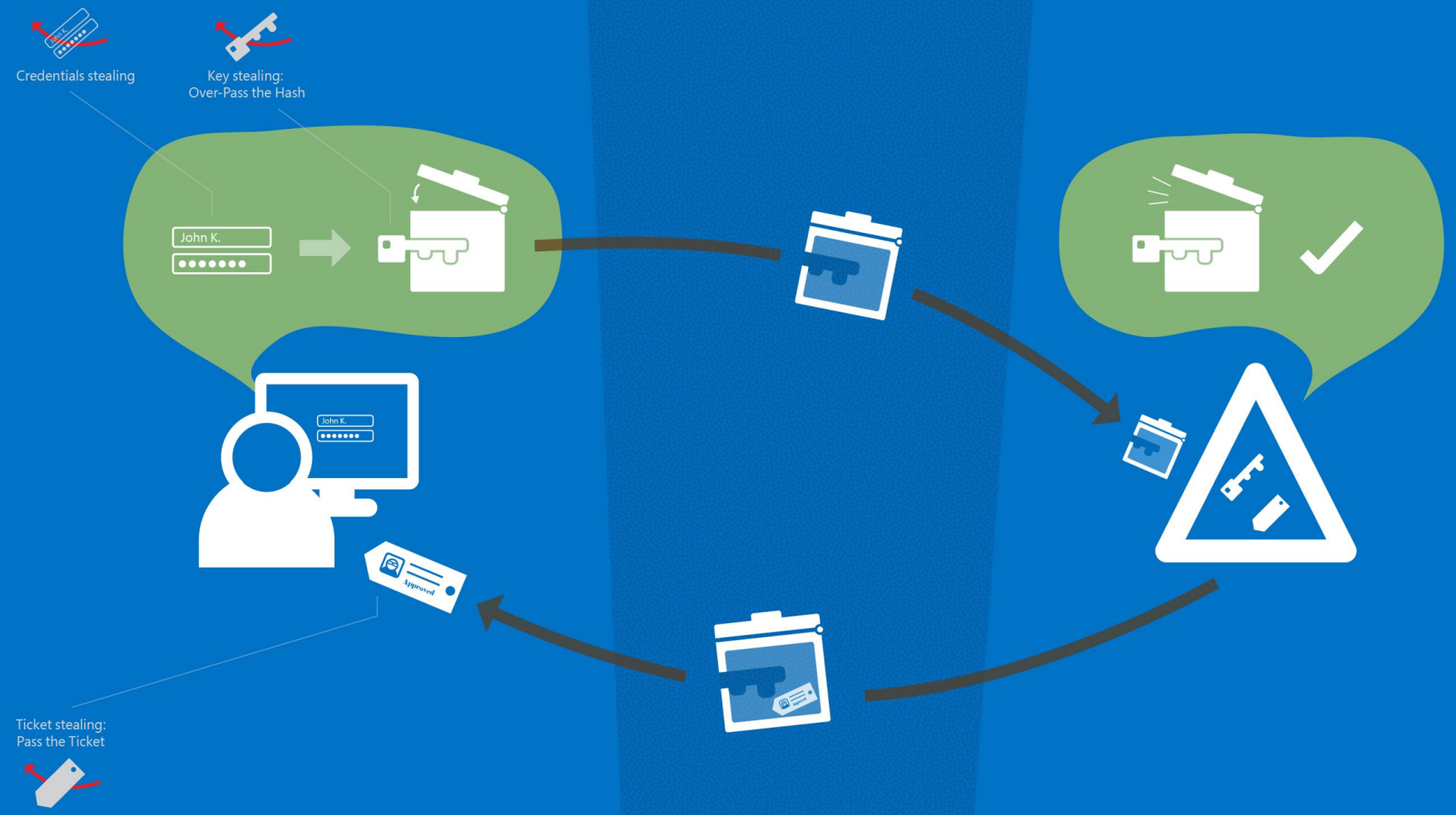
Kerberos: Stealing



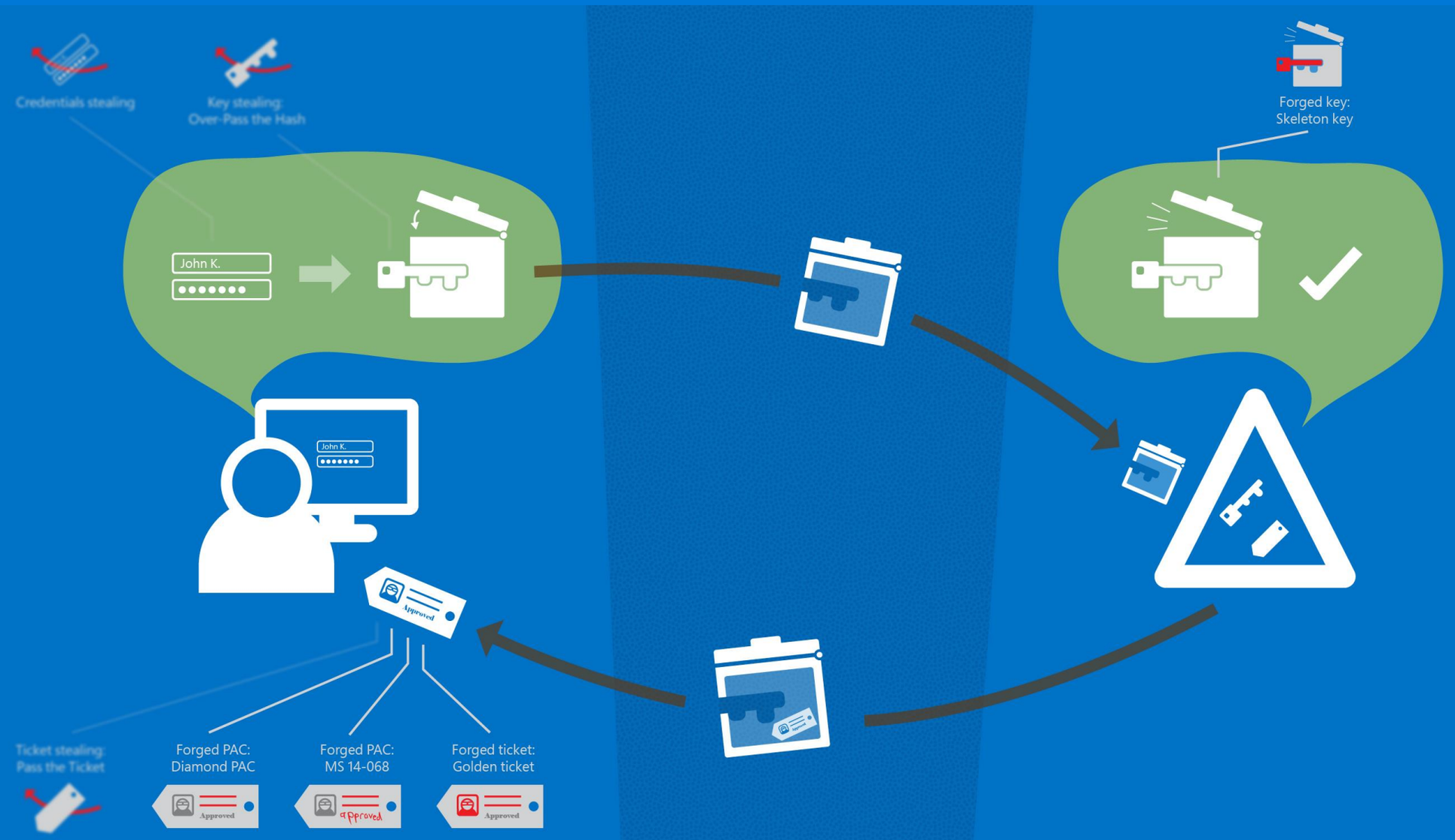
But Stealing will Only Take Attacker so Far

- Attackers can only steal what is there…
- So they have to wait for the “right” user
- Waiting is boring…
- With forgery attacker can get an on demand privileges and access

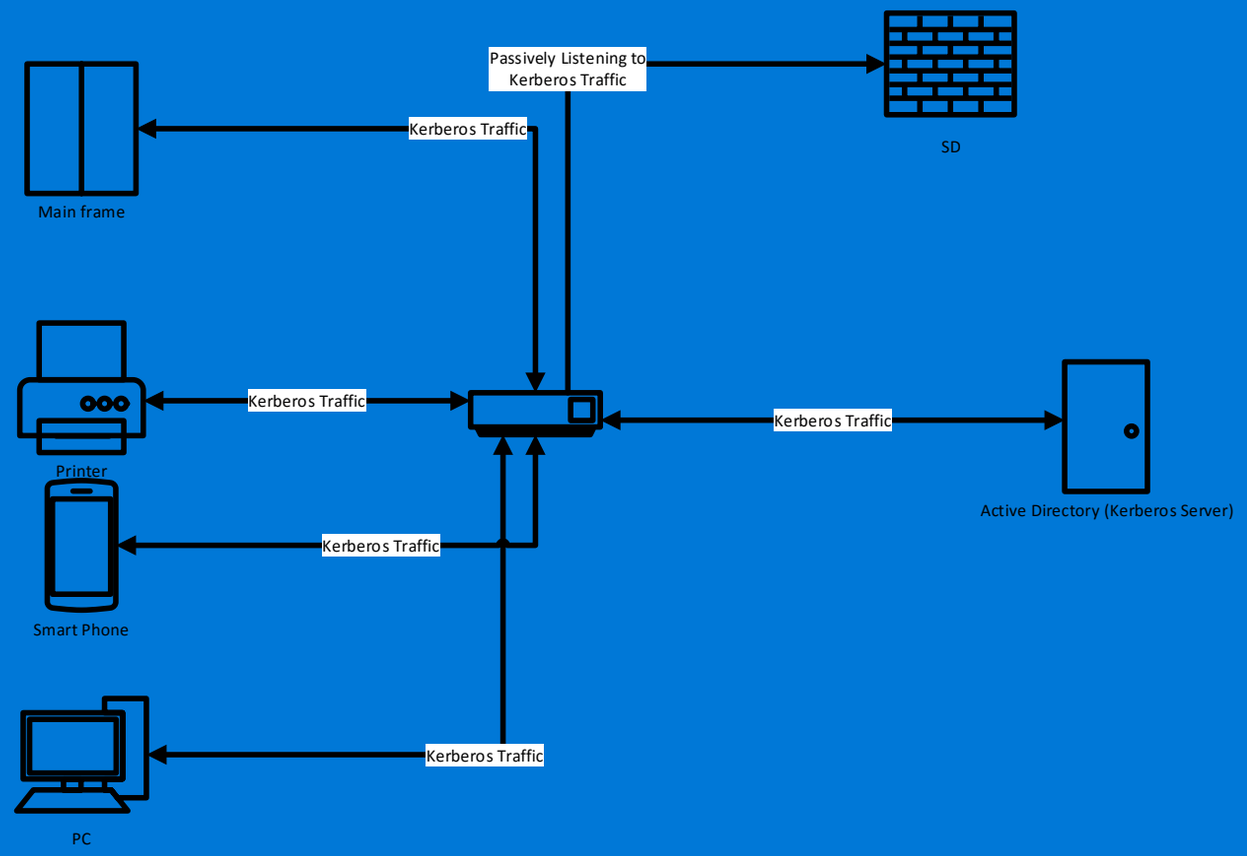
Kerberos: Stealing



Kerberos: Forgery



Network Monitoring of Kerberos



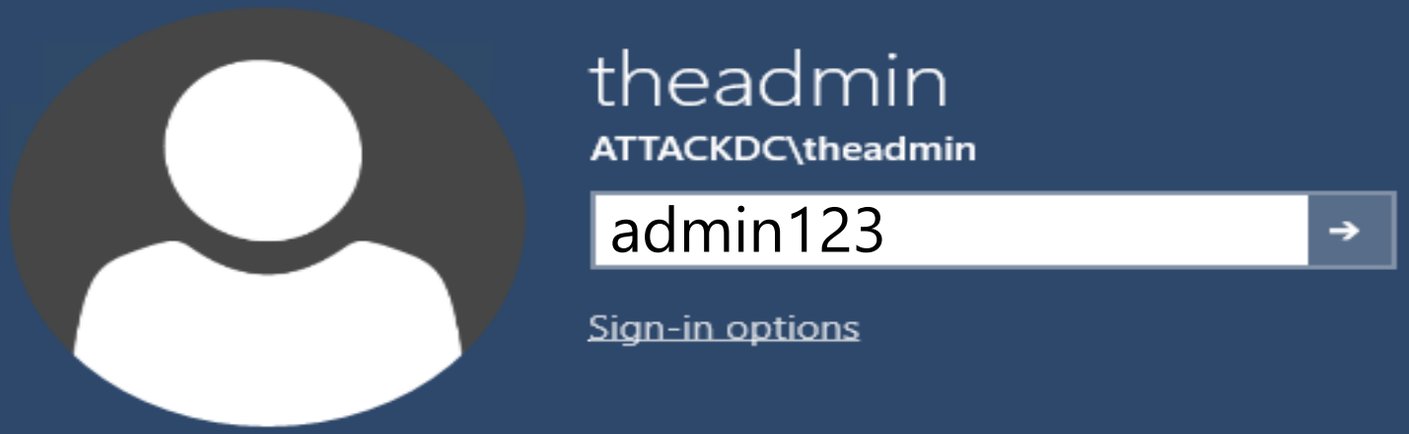
The Skeleton-Key

The attack campaign

- Attackers installed a malware on DC to authenticate as any user, by using a secret password
- Result:
  - Attacker can logon as any user, using the secret password
  - Original users' experience remains the same

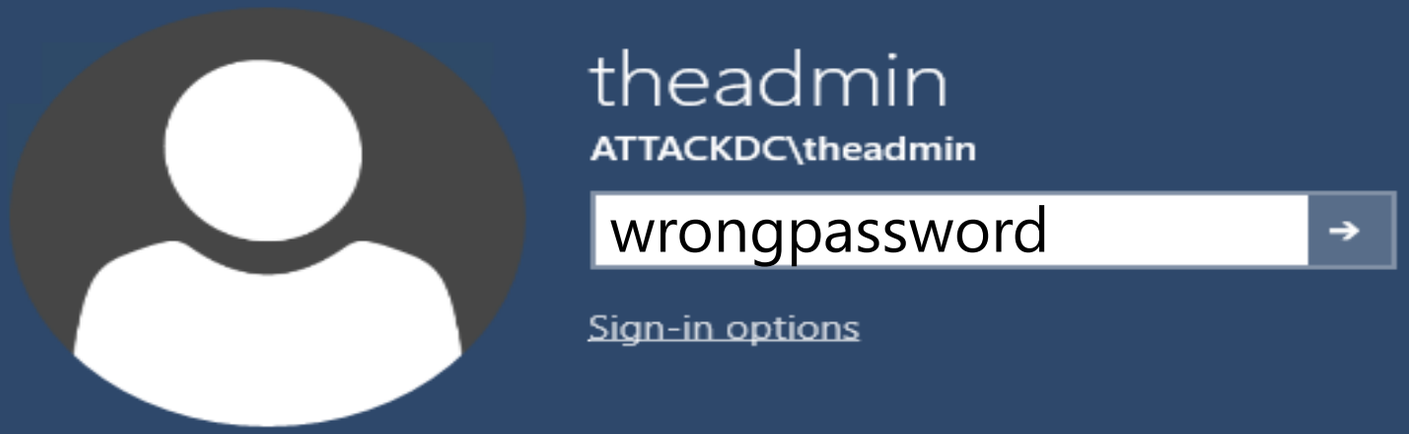
Skeleton Key Malware Effects Demo

- admin123 = real admin password
- P@$$w0rd1 = Attacker skeleton key password


theadmin
ATTACKDC\theadmin
admin123
Sign-in options

Recycle Bin
theadmin
File Explorer
Documents
Settings
Most used
Windows PowerShell
Snipping Tool
Sticky Notes
Paint
Remote Desktop Connection
Notepad
File Explorer
Command Prompt
Power
All apps
Life at a glance
Calendar
Mail
Project Spartan
Photos
Search
People
Refugees stream into Europe as crisis continues
News
OneNote
Weather
Play and explore
Music
Video
Xbox
An investor's field guide to bottom fishing
Money
Winners and Losers from Week 1 of the 2015 College Foo
Sports
Get started
Insider Hub
Windows Feedb
Search the web and Windows
Windows 10 Enterprise Insider Preview
Evaluation copy. Build 10074
7:57 AM
9/6/2015

Malware not installed on DC



theadmin

ATTACKDC\theadmin

The password is incorrect. Try again.

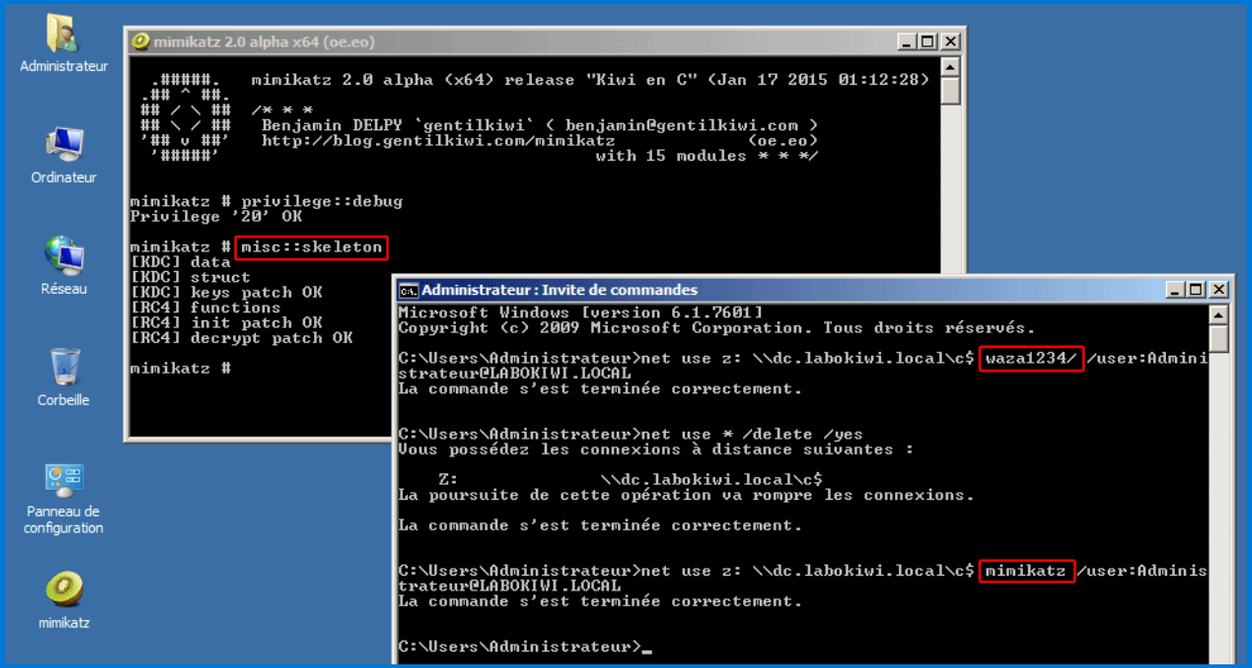
OK

Malware installed on DC

Recycle Bin
theadmin
File Explorer
Documents
Settings
Most used
Windows PowerShell
Snipping Tool
Sticky Notes
Paint
Remote Desktop Connection
Notepad
File Explorer
Command Prompt
Power
All apps
Life at a glance
Calendar
Mail
Project Spartan
Photos
Search
People
Refugees stream into Europe as crisis continues
News
OneNote
Weather
Play and explore
Music
Video
Xbox
An investor's field guide to bottom fishing
Money
Winners and Losers from Week 1 of the 2015 College Foo
Sports
Get started
Insider Hub
Windows Feedb
Search the web and Windows
Windows 10 Enterprise Insider Preview
Evaluation copy. Build 10074
7:57 AM
9/6/2015

Oh No, Skeleton Key for All!

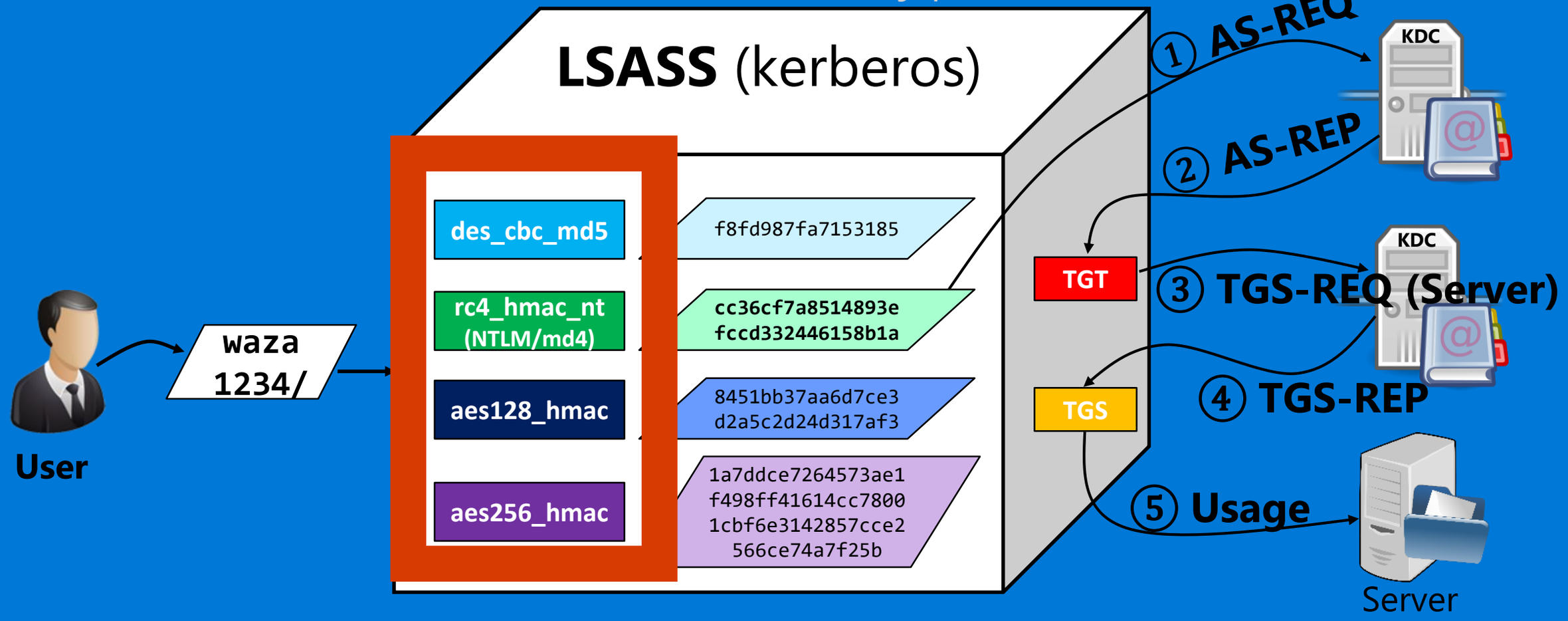
- Thanks to Mimikatz



https://twitter.com/gentilkiwi/status/556246876505509888

Kerberos

- Let's focus on the different encryptions



AES vs. RC4: Key Derivation

- Salting
  - Goal: Same passwords, different users = different keys
  - Create-Key(pw+salt)
  - AES uses the username for salt
  - RC4-HMAC don't have any!
- "Key Stretching"
  - Goal: increase CPU load per password
  - AES uses PBKDF2= Thousands of SHA rounds
  - RC4-HMAC don't have any!

https://commons.wikimedia.org/wiki/File:Jodsalz_mit_Fluor_und_Folsaeure.jpg

Kerberos Authentication In Depth

User1

waza
1234/

LSASS (kerberos)

des_cbc_md5	f8fd987fa7153185
rc4_hmac_nt (NTLM/md4)	cc36cf7a8514893e fccd332446158b1a
aes128_hmac	8451bb37aa6d7ce3 d2a5c2d24d317af3
aes256_hmac	1a7ddce7264573ae1 f498ff41614cc7800 1cbf6e3142857cce2 566ce74a7f25b

TGT

Graphics by Benjamin Delpy

① AS-REQ
Name: user1
Etype: DES, RC4, AES128, AES256

② KERB-ERR
Pre-auth-REQ
Etype: RC4,AES
Salt:user1

③ AS-REQ
PA-ENC-TS
Etype:AES

④ AS-REP
TGT+Enc
Etype: AES

user	rc4_hmac _nt	aes256_ hmac
Joe	21321...	543..
user1	**cc36cf7a ...**	1a7ddc ...
Doe		

Kerberos Authentication: Over the Wire

```
⊟ as-req
    pvno: 5
    msg-type: krb-as-req (10)
  ⊟ padata: 1 item
    ⊞ PA-DATA PA-PAC-REQUEST
  ⊟ req-body
      Padding: 0
    ⊞ kdc-options: 40810010 (forwardable, renewable, canonicalize, renewable-ok)
    ⊞ cname
      realm: aorato.research
    ⊞ sname
      till: 2037-09-13 02:48:05 (UTC)
      rtime: 2037-09-13 02:48:05 (UTC)
      nonce: 160211996
    ⊟ etype: 6 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-MD5-56 (24)
        ENCTYPE: eTYPE-ARCFOUR-HMAC-OLD-EXP (-135)
        ENCTYPE: eTYPE-DES-CBC-MD5 (3)
```

```
⊟ krb-error
    pvno: 5
    msg-type: krb-error (30)
    stime: 2014-03-10 20:05:07 (UTC)
    susec: 165032
    error-code: eRR-PREAUTH-REQUIRED (25)
    realm: aorato.research
  ⊞ sname
  ⊟ e-data: 30543031a103020113a22a04283026301da003020112a116..
    ⊟ PA-DATA PA-ENCTYPE-INFO2
      ⊟ padata-type: kRB5-PADATA-ETYPE-INFO2 (19)
        ⊟ padata-value: 3026301da003020112a1161b14414f5241544f
          ⊟ ETYPE-INFO2-ENTRY
              etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
              salt: AORATO.RESEARCHbugsb
          ⊟ ETYPE-INFO2-ENTRY
              etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

```
⊟ 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: 3041a003020112a23a0438c871bc029b90195c7d2981b0cd...
            etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
            cipher: c871bc029b90195c7d2981b0cd8e4c98fa5fa747689f86e1...
```

The Skeleton Key Malware on DC

- Downgrades Kerberos encryption to RC4
  - Hooks SamIRetrieveMultiplePrimaryCredentials()
  - Practically disable “Kerberos-Newer-Keys” (=AES)
- Patches the Decrypt() function in CDLocateCSystem structure
  - Calls the original Decrypt() (normal log-in would still work)
  - If it fails, it replaces the hash retrieved from Active Directory with the skeleton key hash and calls Decrypt() again

Why only RC4?

- Due to salting AES keys of different users create different AES keys
- Attacker must either:
  - Compute AES keys in real time – lots of CPU
  - Compute in offline for all users – lots of memory
- Malicious patching becomes harder as attacker must intervene in more places to extract user context
- Attacker's Solution: Downgrade to RC4

The Skeleton Key Malware: Kerberos

User1

Skeleton

LSASS (kerberos)

des_cbc_md5	✗
rc4_hmac_nt (NTLM/md4)	ff687678...
aes128_hmac	✗
aes256_hmac	✗

TGT

Graphics by Benjamin Delpy

① AS-REQ
Name: user1
Etype: DES, RC4, AES128, AES256

② KERB-ERR
Pre-auth-REQ
Etype: RC4, ~~AES~~
Salt: ~~user1~~

③ AS-REQ
PA-ENC-TS
Etype: RC4

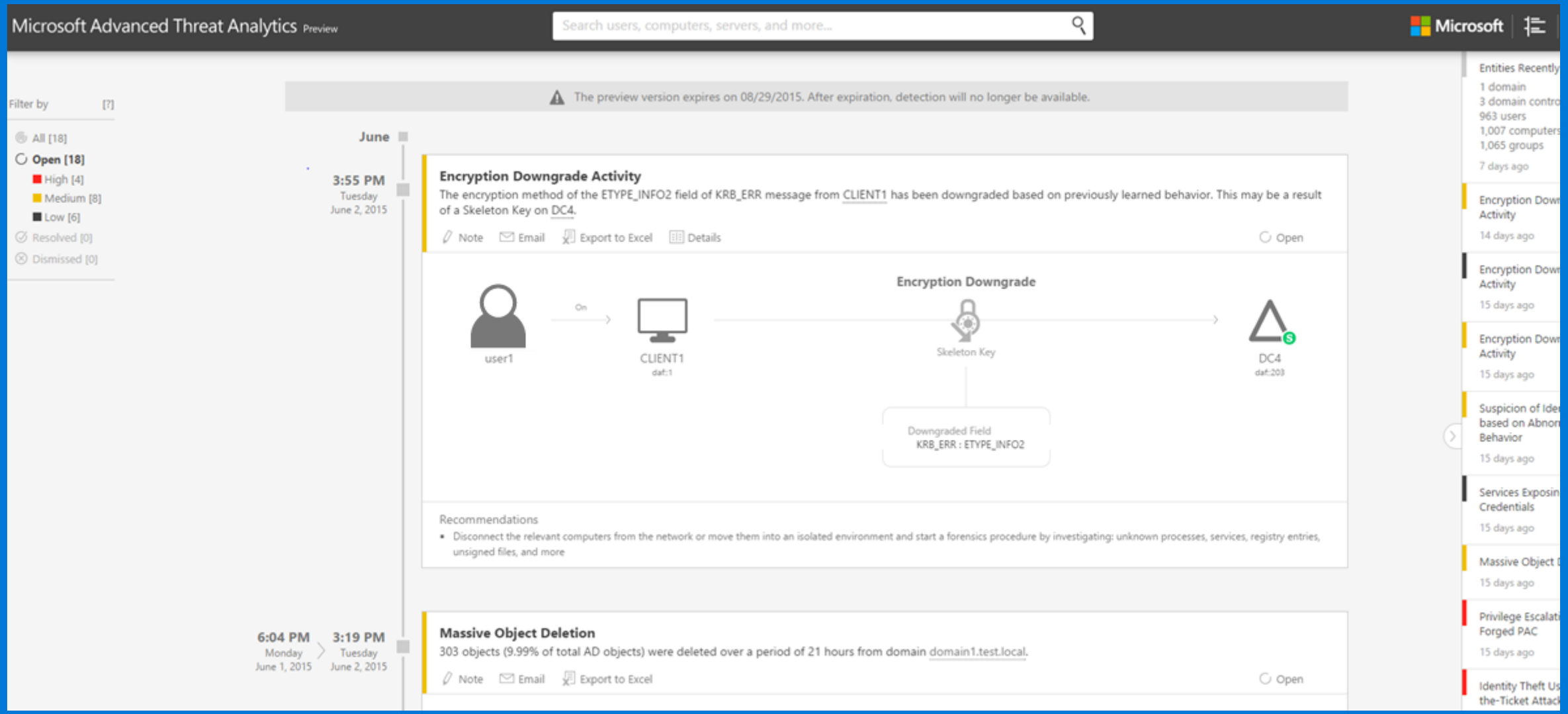
④ AS-REP
TGT+Enc
Etype: RC4

user	rc4_hmac_nt	aes256_hmac
Joe	21321...	543..
user1	cc36cf7a ...	1a7ddc ...
	ff687678...	✗

Detecting a Skeleton Key Malware

- Skeleton Key malware downgrades users' encryption to RC4
- Let's detect it!
- We know the user should be offered AES by DC
  - DC offered AES in the past
  - Judging by DFL and user's capabilities
- Why only RC4 now?

Skeleton Key Malware Detection



Detecting with a script

- The script:
  - Verifies whether the DFL is relevant (>=2008)
  - Finds an AES supporting account (msds-supportedencryptiontypes>=8)
  - Sends an AS-REQ to all DCs with only AES E-type supported
  - If it fails, then there's a good chance the DC is infected
- Publicly available for download
  - https://gallery.technet.microsoft.com/Aorato-Skeleton-Key-24e46b73

Script Detects the Skeleton Key Malware

DC-TALRES on [redacted] - Virtual Machine Connection

File Action Media View Help

mimikatz 2.0 alpha x64 (oe.eo)

```
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Jan 17 2015 01:24:17)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 15 modules * * */


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # misc::skeleton
[KDC] data
[KDC] struct
[KDC] keys patch OK
[RC4] functions
[RC4] init patch OK
[RC4] decrypt patch OK

mimikatz # _
```

CLI1-TALRES on [redacted] - Virtual Machine Connection

File Action Media View Help

Recycle Bin

Wireshark

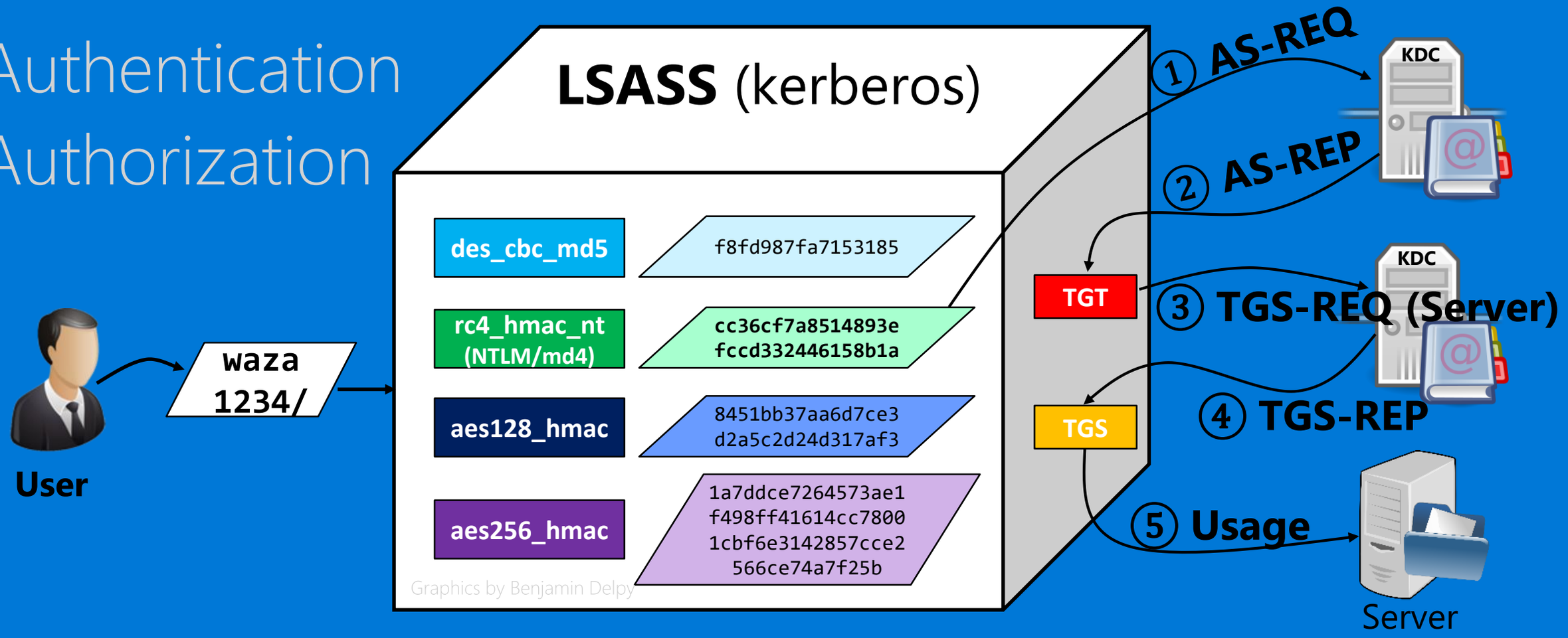
Windows PowerShell

```
PS C:\[redacted]> .\SkeletonScan.ps1
Domain Functional Level (DFL) must be at least 2008R2 to test, current DFL of domain res.talres.com is Windows2012R2Dom
ain so the check is valid
  [+] Building AS-REQ for WIN-21DGC9LKJ86.res.talres.com... Done!
  [+] Sending AS-REQ to WIN-21DGC9LKJ86.res.talres.com... Done!
  [+] Receiving KrbError from WIN-21DGC9LKJ86.res.talres.com... Done!
  [+] Parsing KrbError from WIN-21DGC9LKJ86.res.talres.com... Error Code is 14!
WIN-21DGC9LKJ86.res.talres.com DC is supposed to support AES but is not. It may be infected with the Skeleton Key malwa
re
PS C:[redacted]> _
```

Forged Ticket: Golden Ticket

Kerberos

- Authentication
- Authorization



- Standard (RFC4120)

PAC (Privilege Attribute Certificate)

- MS Specific [MS-PAC]
- Authorization Data

https://commons.wikimedia.org/wiki/File:Identification_card_JAPAN.jpg

PAC (in TGT)

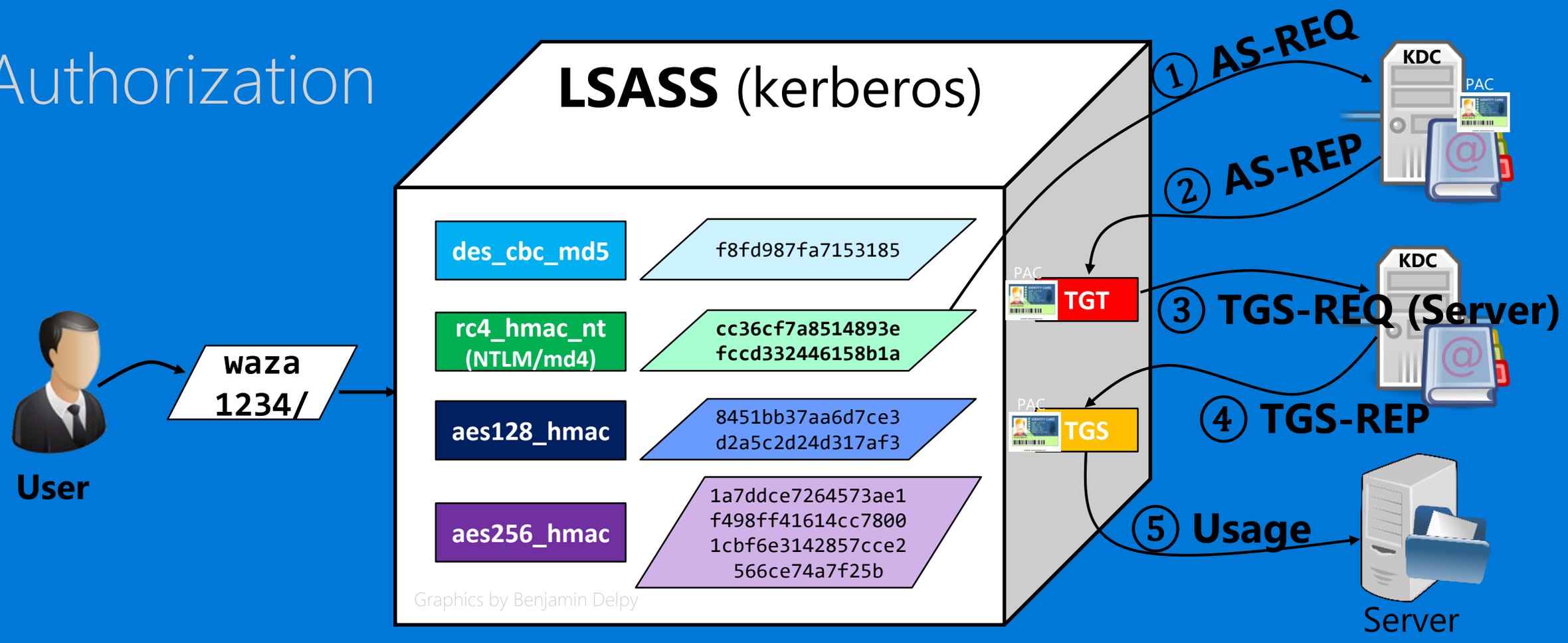
Username : **Administrator**
Domain SID
S-1-5-21-4014832156-2573456389-2040062157
User ID
500 Administrator
Groups ID
512 ***Domain Admins***
519 ***Enterprise Admins***
518 ***Schema Admins***
…

CHECKSUM_SRV – HMAC_SHA1 - **krbtgt**
3f..

CHECKSUM_KDC – HMAC_MD5 - **krbtgt**
B6..

Kerberos and PAC

- Authorization



- PAC is embedded within the encrypted ticket

TGT Integrity

- Ticket integrity is guaranteed:
  - The ticket is encrypted
  - The Key is of Kerberos internal account KRBTGT

    →Attacker cannot edit the TGT
- PAC integrity is guaranteed:
  - Embedded within the encrypted ticket
  - PAC is signed by KRBTGT

    →Attacker cannot edit the PAC
- But what if attackers obtain the KRBTGT key?

PAC (in TGT)

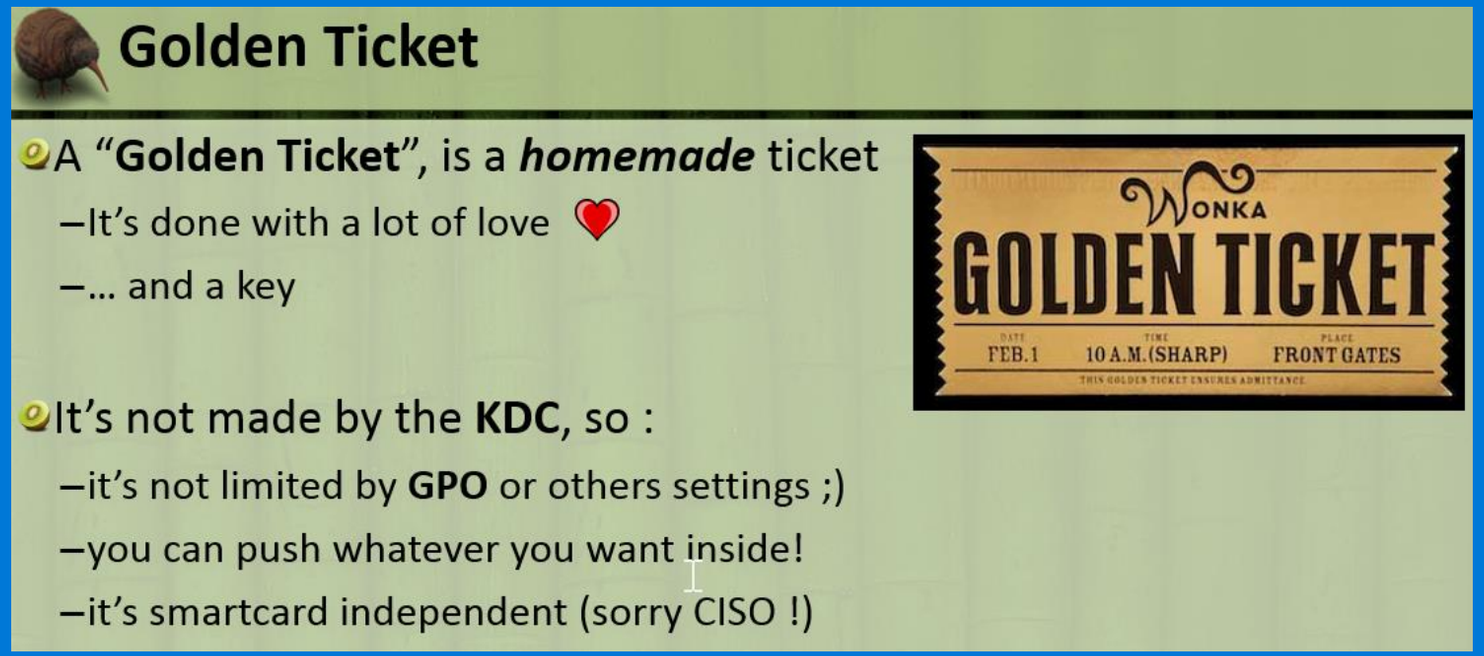
Username : **Administrator**
Domain SID
S-1-5-21-4014832156-2573456389-2040062157
User ID
500 Administrator
Groups ID
512 ***Domain Admins***
519 ***Enterprise Admins***
518 ***Schema Admins***
…

CHECKSUM_SRV – HMAC_SHA1 - **krbtgt**
3f..

CHECKSUM_KDC – HMAC_MD5 - **krbtgt**
B6..

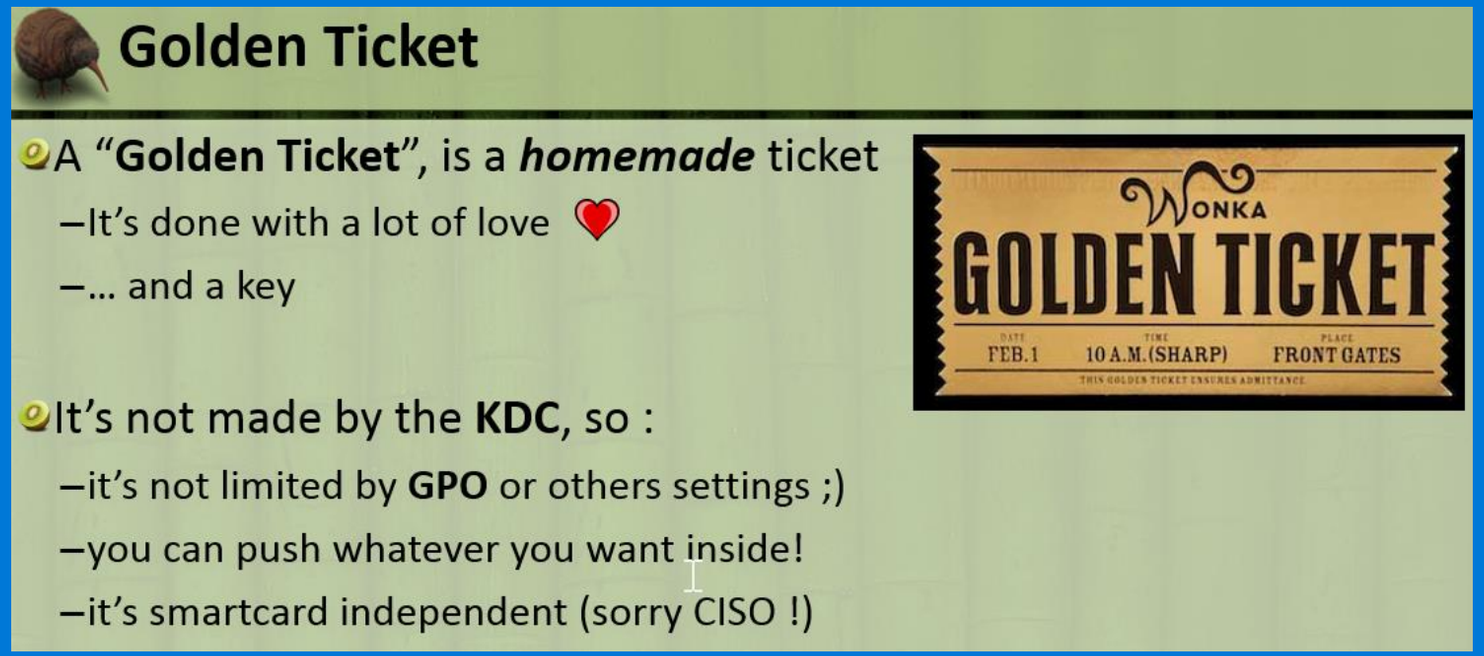
The Golden Ticket Attack

- Stealing KRBTGT key from AD allows the attackers to create arbitrary tickets
- In the words of its creator Benjamin Delpy (AKA gentilkiwi / Mimikatz):

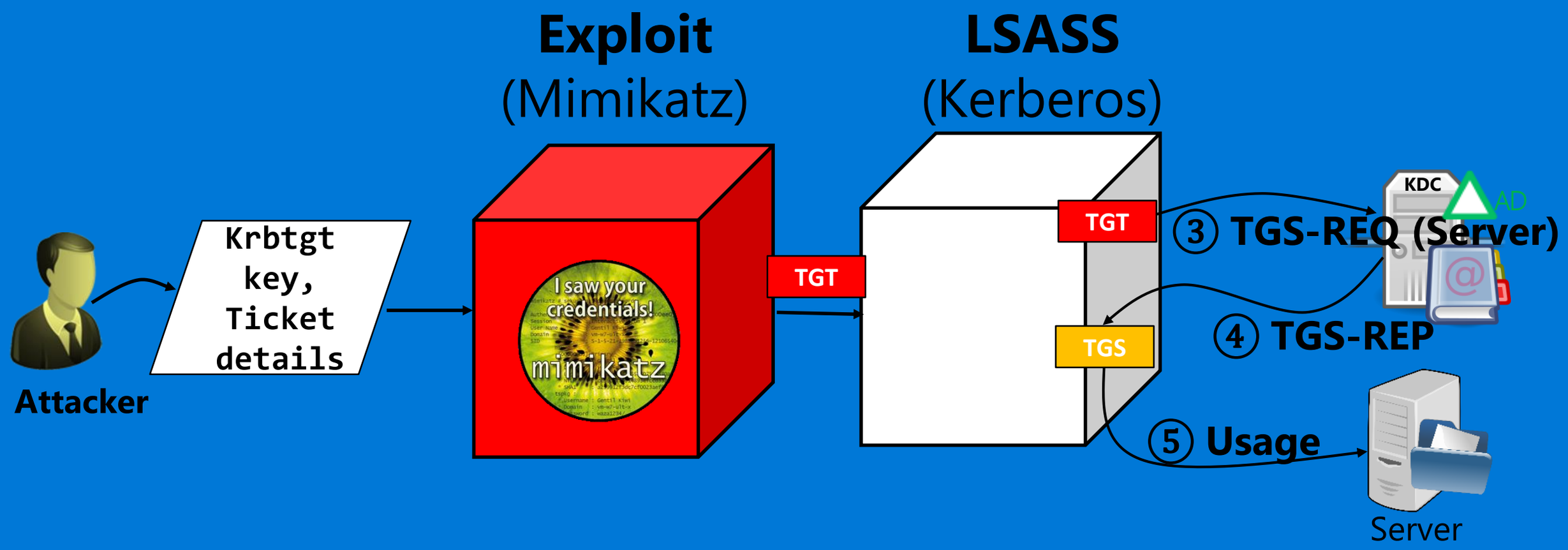


The Golden Ticket Attack

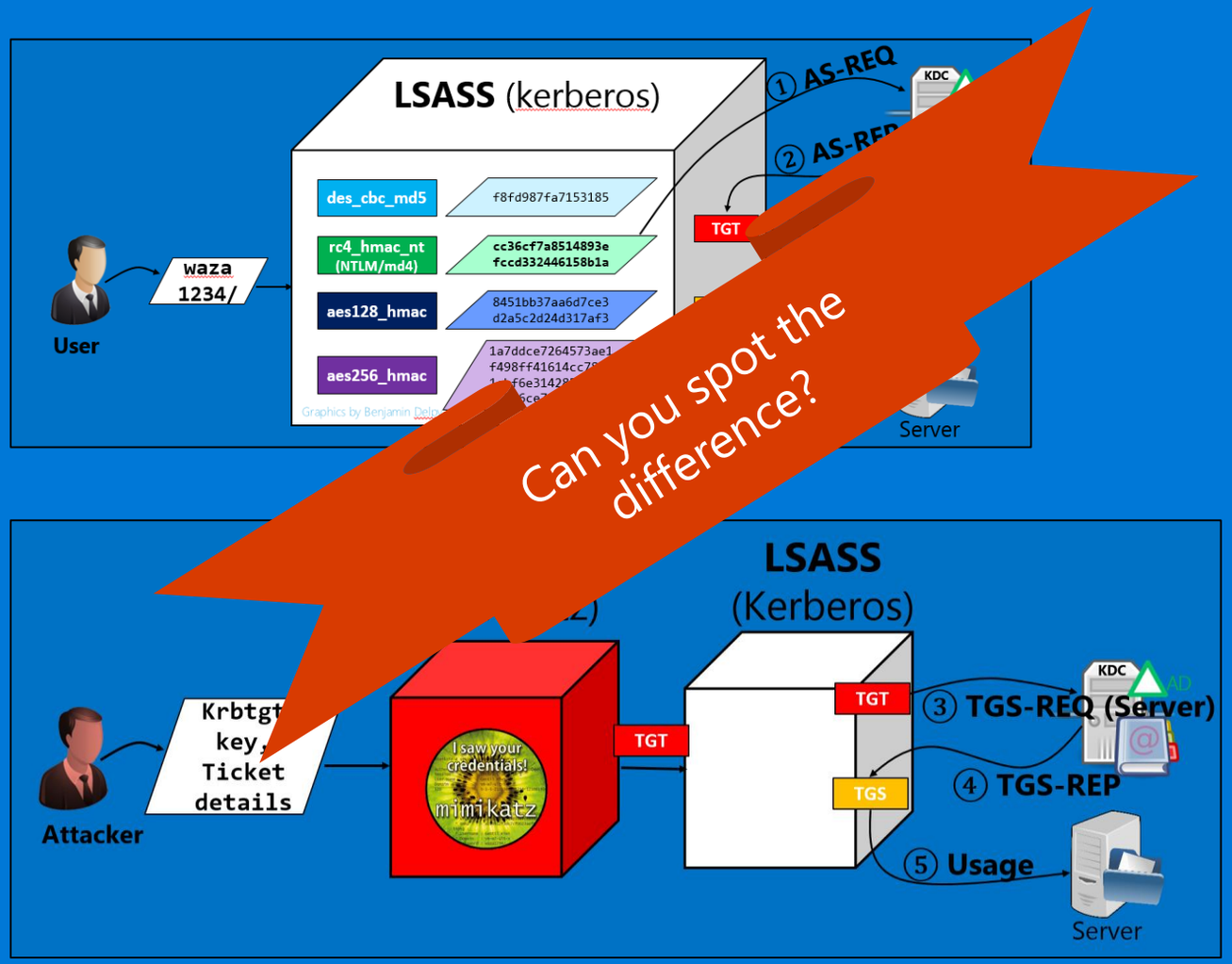
- Stealing KRBTGT key from AD allows the attackers to create arbitrary tickets
- In the words of its creator Benjamin Delpy (AKA gentilkiwi / Mimikatz):



Golden Ticket Kerberos "Auth"



Let's Play Spot the Difference



Over the network

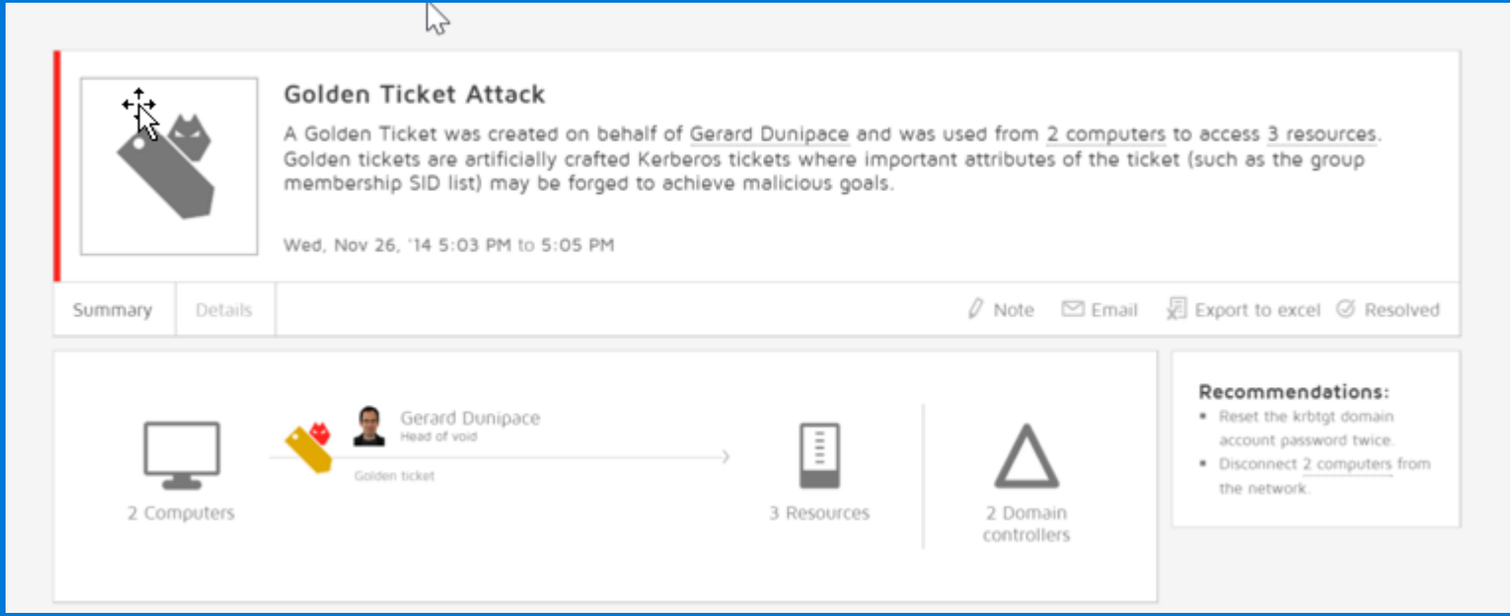
```
⊟ Kerberos AS-REP
  ⊟ Record Mark: 1633 bytes
      0... .... .... .... .... .... .... .... = Reserved: Not set
      .000 0000 0000 0000 0000 0110 0110 0001 = Record Length: 1633
    Pvno: 5
    MSG Type: AS-REP (11)
  ⊟ padata: PA-ENCTYPE-INFO2
    ⊟ Type: PA-ENCTYPE-INFO2 (19)
      ⊟ Value: 303b3039a003020112a1321b30414f5241544f2e52455345... aes256-cts-hmac-sha1-96
          Encryption type: aes256-cts-hmac-sha1-96 (18)
          Salt: 414f5241544f2e524553454152434868 6f7374616f726174...
    Client Realm: AORATO.RESEARCH
  ⊞ Client Name (Principal): AORATORESSRV8$
  ⊟ Ticket
      Tkt-vno: 5
      Realm: AORATO.RESEARCH
    ⊞ Server Name (Service and Instance): krbtgt/AORATO.RESEARCH
    ⊟ enc-part aes256-cts-hmac-sha1-96
        Encryption type: aes256-cts-hmac-sha1-96 (18)
        Kvno: 2
        enc-part: b8c8937b1f615eb6e96ff1952ddf6c80f2728fb8ca62fbff...
  ⊞ enc-part aes256-cts-hmac-sha1-96
```

```
⊟ Kerberos TGS-REQ
  ⊟ Record Mark: 1691 bytes
      0... .... .... .... .... .... .... .... = Reserved: Not set
      .000 0000 0000 0000 0000 0110 1001 1011 = Record Length: 1691
    Pvno: 5
    MSG Type: TGS-REQ (12)
  ⊟ padata: PA-TGS-REQ Unknown:167
    ⊟ Type: PA-TGS-REQ (1)
      ⊟ Value: 6e82052830820524a003020105a10302010ea20703050000... AP-REQ
          Pvno: 5
          MSG Type: AP-REQ (14)
          Padding: 0
        ⊞ APOptions: 00000000
        ⊟ Ticket
            Tkt-vno: 5
            Realm: AORATO.RESEARCH
          ⊞ Server Name (Service and Instance): krbtgt/AORATO.RESEARCH
          ⊟ enc-part aes256-cts-hmac-sha1-96
              Encryption type: aes256-cts-hmac-sha1-96 (18)
              Kvno: 2
              enc-part: b8c8937b1f615eb6e96ff1952ddf6c80f2728fb8ca62fbff...
        ⊞ Authenticator aes256-cts-hmac-sha1-96
    ⊟ Type: Unknown (167)
        Value: 3009a00703050040000000
  ⊞ KDC_REQ_BODY aes256-cts-hmac-sha1-96
```

Source	Destination	Protocol	Info	enc-part
10.0.0.4	10.0.0.5	KRB5	AS-REP	b8c8937b1f615eb6e96ff1952ddf6c80f2728fb8ca62fbff...
10.0.0.5	10.0.0.4	KRB5	TGS-REQ	b8c8937b1f615eb6e96ff1952ddf6c80f2728fb8ca62fbff...
10.0.0.5	10.0.0.4	KRB5	TGS-REQ	b8c8937b1f615eb6e96ff1952ddf6c80f2728fb8ca62fbff...
10.0.0.5	10.0.0.4	KRB5	TGS-REQ	b8c8937b1f615eb6e96ff1952ddf6c80f2728fb8ca62fbff...
10.0.0.5	10.0.0.4	KRB5	TGS-REQ	b8c8937b1f615eb6e96ff1952ddf6c80f2728fb8ca62fbff...
10.0.0.5	10.0.0.4	KRB5	TGS-REQ	b8c8937b1f615eb6e96ff1952ddf6c80f2728fb8ca62fbff...

Network-Based Detection

- Recouple TGT in TGS-REQ with its parent TGT in AS-REP
- Alert on “orphaned” TGT tickets



Forged PAC: MS14-068

Gavin Millard @gmillard · 11h
MS14-068 in the real world.
"Welcome Captain. Would you like a
#infosec
UNITED FLIGHT
FLIGHT D3048
Coach
FIRST NAME
BOARDS AT
BOARDING ZONE
2
DEPARTS
3:40PM
UNITED BOARDING PASS
UNITED
> SC US
SC UK
Andromeda botnet used to distribute new point-of-sale malware
Ohio inmate caught with pris administrative login credentials
NEWS
PRODUCTS
BLOGS
RESOURCES
VIDEOS
SC MAGAZINE
FOR IT SECURITY PROFESSIONALS
SC Magazine > News > Duqu2.0 knocks Kaspersky and security peers
Tom Reeve, senior report, SC Magazine UK
June 11, 2015
Duqu2.0 knocks Kaspersky and security peers
31
12

PAC (Privilege Attribute Certificate)

- MS Specific [MS-PAC]
- Authorization Data
- Signed by the KRBTGT key

https://commons.wikimedia.org/wiki/File:Identification_card_JAPAN.jpg

PAC (in TGT)

Username : **Administrator**
Domain SID
S-1-5-21-4014832156-2573456389-2040062157
User ID
500 Administrator
Groups ID
512 ***Domain Admins***
519 ***Enterprise Admins***
518 ***Schema Admins***
…

CHECKSUM_SRV – HMAC_SHA1 - **krbtgt**
3f..

CHECKSUM_KDC – HMAC_MD5 - **krbtgt**
B6..

Digital signature

- "A **digital signature** is a mathematical scheme for demonstrating the authenticity of a digital message"

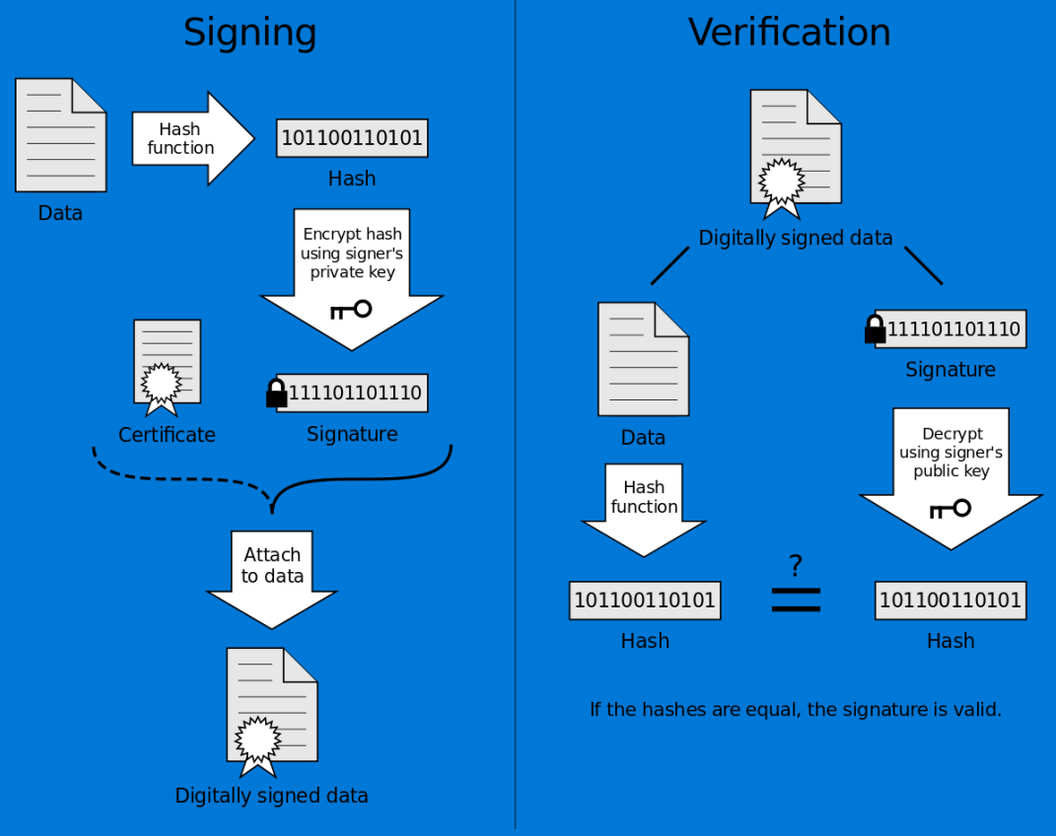


Image by ACDX https://commons.wikimedia.org/wiki/File%3ADigital_Signature_diagram.svg

TGT to Service Ticket

- The KDC :
  - Verifies (only) the inner signature (CHECKSUM_SRV)
  - Copies PAC's Content
  - Re-signs PAC with the Server Key
  - Adjusts the KDC signature

PAC (in TGT)

Username : **Administrator**
Domain SID
S-1-5-21-4014832156-2573456389-2040062157
User ID
500 Administrator
Groups ID
512 ***Domain Admins***
519 ***Enterprise Admins***
518 ***Schema Admins***
…

CHECKSUM_SRV – HMAC_SHA1 - **krbtgt**
3f..

CHECKSUM_KDC – HMAC_MD5 - **krbtgt**
B6..

PAC (in Service Ticket)

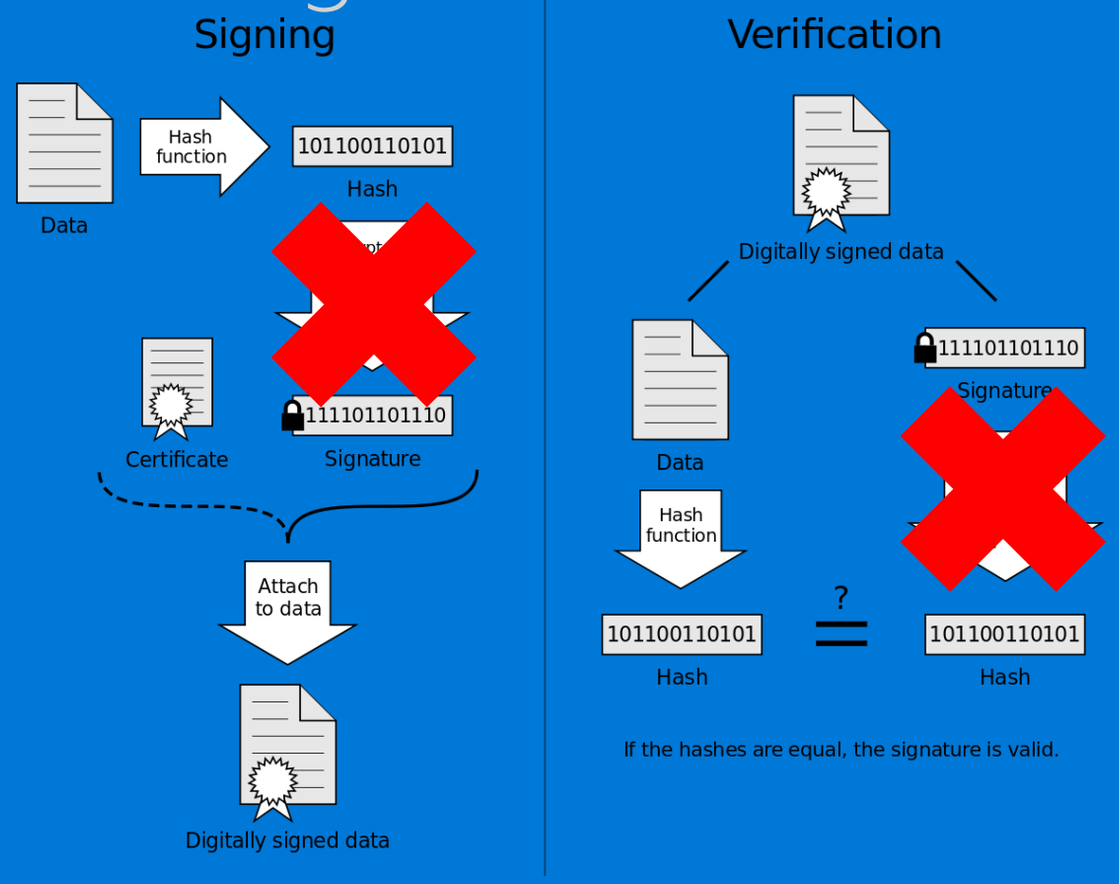
Username : **Administrator**
Domain SID
S-1-5-21-4014832156-2573456389-2040062157
User ID
500 Administrator
Groups ID
512 ***Domain Admins***
519 ***Enterprise Admins***
518 ***Schema Admins***
…

CHECKSUM_SRV – HMAC_SHA1 – **CIFS/Server**
2a..

CHECKSUM_KDC – HMAC_MD5 - **krbtgt**
56..

Vuln: Key-less “Signatures” are Allowed(!!!)

- KDC accepts key-less hashes as signatures: MD5,..
- Result: anyone can sign



http://en.wikipedia.org/wiki/Digital_signature

But is this Vuln Exploitable?

- The exploit of the vuln is not trivial
- Attackers cannot “just replace” PAC in Ticket as the ticket is encrypted
- If Attackers have the key to decrypt the ticket then they can sign the PAC too, so the vuln is not really relevant.
- Or is it? ☺

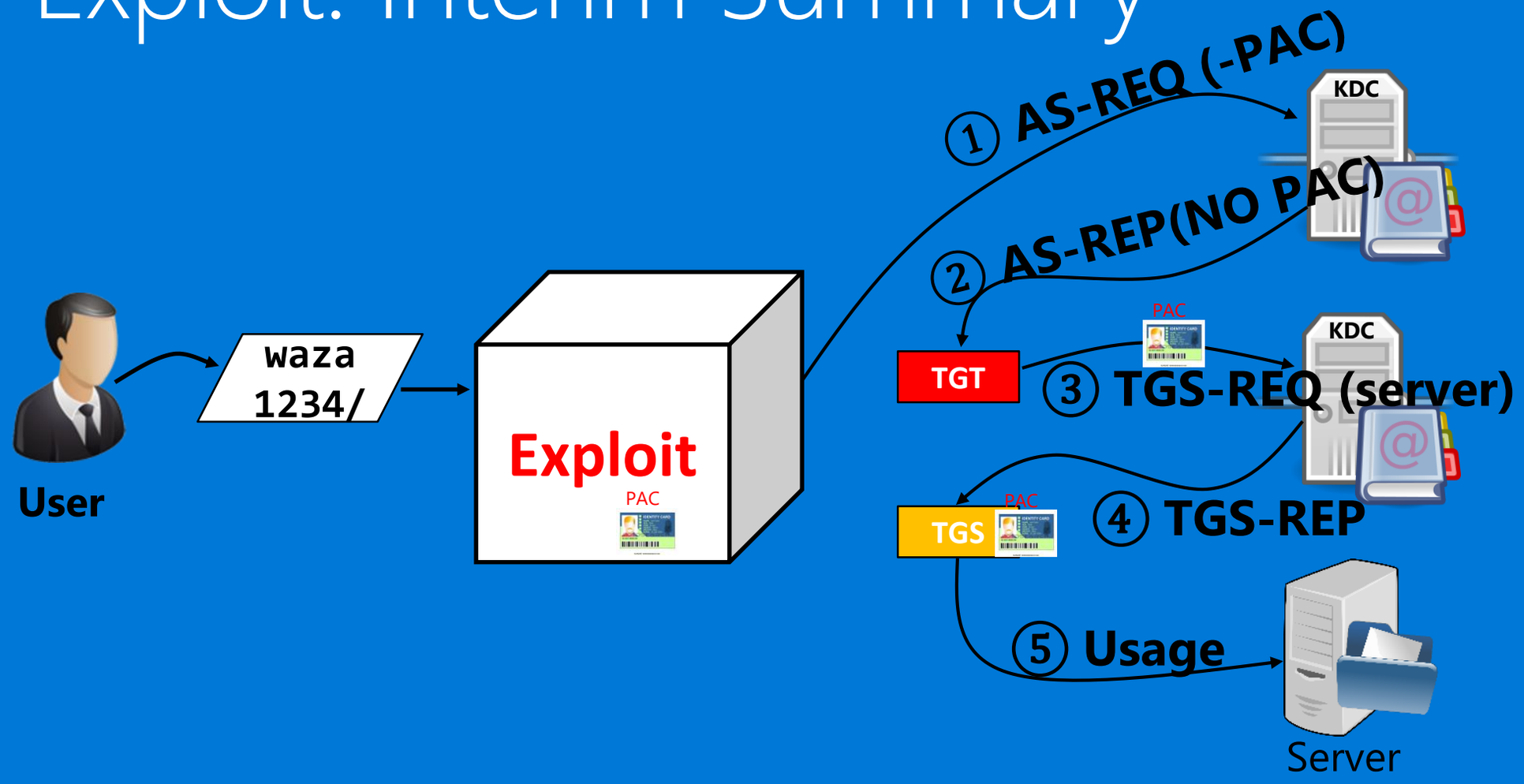
The Exploit

- Request (AS-REQ) a PAC-less TGT
- Request (TGS-REQ) a Service Ticket with Fake PAC:
  - Pac-less TGT
  - But.. With Enc-AuthorizationData, that can contain… a PAC!
  - The PAC will be a fake PAC “signed” by MD5
  - The KDC returns ticket which includes the fake PAC (or whatever else we put in)

```
⊟ Kerberos
  ⊞ Record Mark: 279 bytes
  ⊟ as-req
      pvno: 5
      msg-type: krb-as-req (10)
    ⊟ padata: 2 items
      ⊞ PA-DATA PA-ENC-TIMESTAMP
      ⊟ PA-DATA PA-PAC-REQUEST
        ⊟ padata-type: kRB5-PADATA-PA-PAC-REQUEST (128)
          ⊟ padata-value: 3005a003010100
              include-pac: False
    ⊞ req-body
```

```
⊟ Kerberos
  ⊞ Record Mark: 1386 bytes
  ⊟ tgs-req
      pvno: 5
      msg-type: krb-tgs-req (12)
    ⊞ padata: 2 items
    ⊟ req-body
        Padding: 0
      ⊞ kdc-options: 50800000 (forwardable, proxiable, renewable)
        realm: AORATO.RESEARCH
      ⊞ sname
        from: 1970-01-01 00:00:00 (UTC)
        till: 1970-01-01 00:00:00 (UTC)
        rtime: 1970-01-01 00:00:00 (UTC)
        nonce: 819883661
      ⊞ etype: 1 item
      ⊟ enc-authorization-data
          etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
          cipher: b7939bf9d9ca3334f7cfa4208d33e44044e1e336cf5d6de6..
```

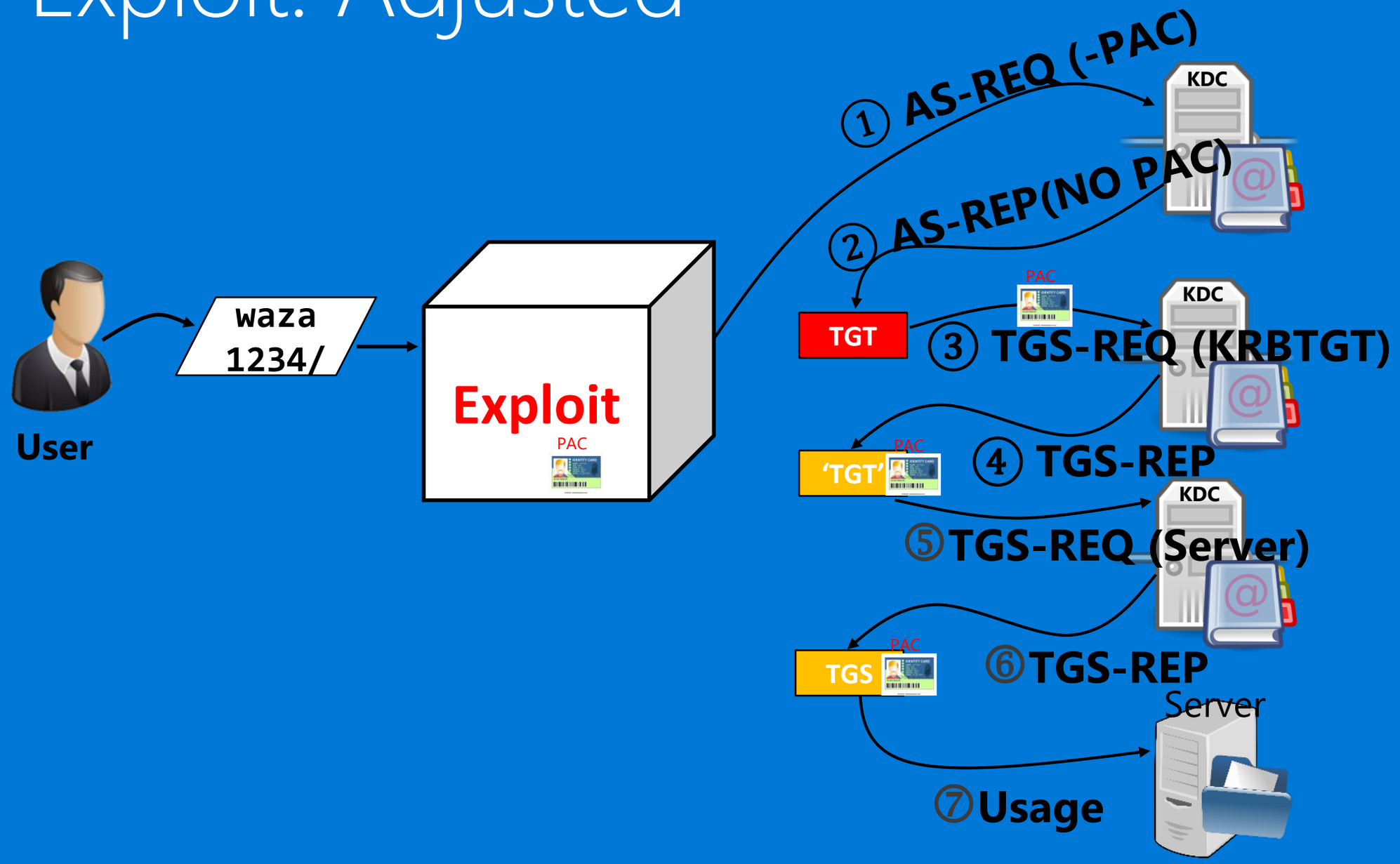
The Exploit: Interim Summary



Problem (for the attacker)

- MS11-013 fixed key-less hash signatures issue
  - For clients and servers
  - Not for KDC!

The Exploit: Adjusted



The Adjusted Exploit Explained

- Attacker's first target service is KRBTGT itself
  - PAC gets embedded "as is" within the new "TGT"
- Attacker second target is the desired service
  - The Service Ticket gets re-signed properly

PAC (in "TGT")

Username : **Administrator**
Domain SID
S-1-5-21-4014832156-2573456389-2040062157
User ID
500 Administrator
Groups ID
512 *Domain Admins*
519 *Enterprise Admins*
518 *Schema Admins*
…

CHECKSUM_SRV – MD5 – **no key**
3f..

CHECKSUM_KDC – MD5 – **no key**
B6..

PAC (in Service Ticket)

Username : **Administrator**
Domain SID
S-1-5-21-4014832156-2573456389-2040062157
User ID
500 Administrator
Groups ID
512 *Domain Admins*
519 *Enterprise Admins*
518 *Schema Admins*
…

CHECKSUM_SRV – HMAC_SHA1 –**CIFS/Server**
2a..

CHECKSUM_KDC – HMAC_MD5 - **krbtgt**
56..

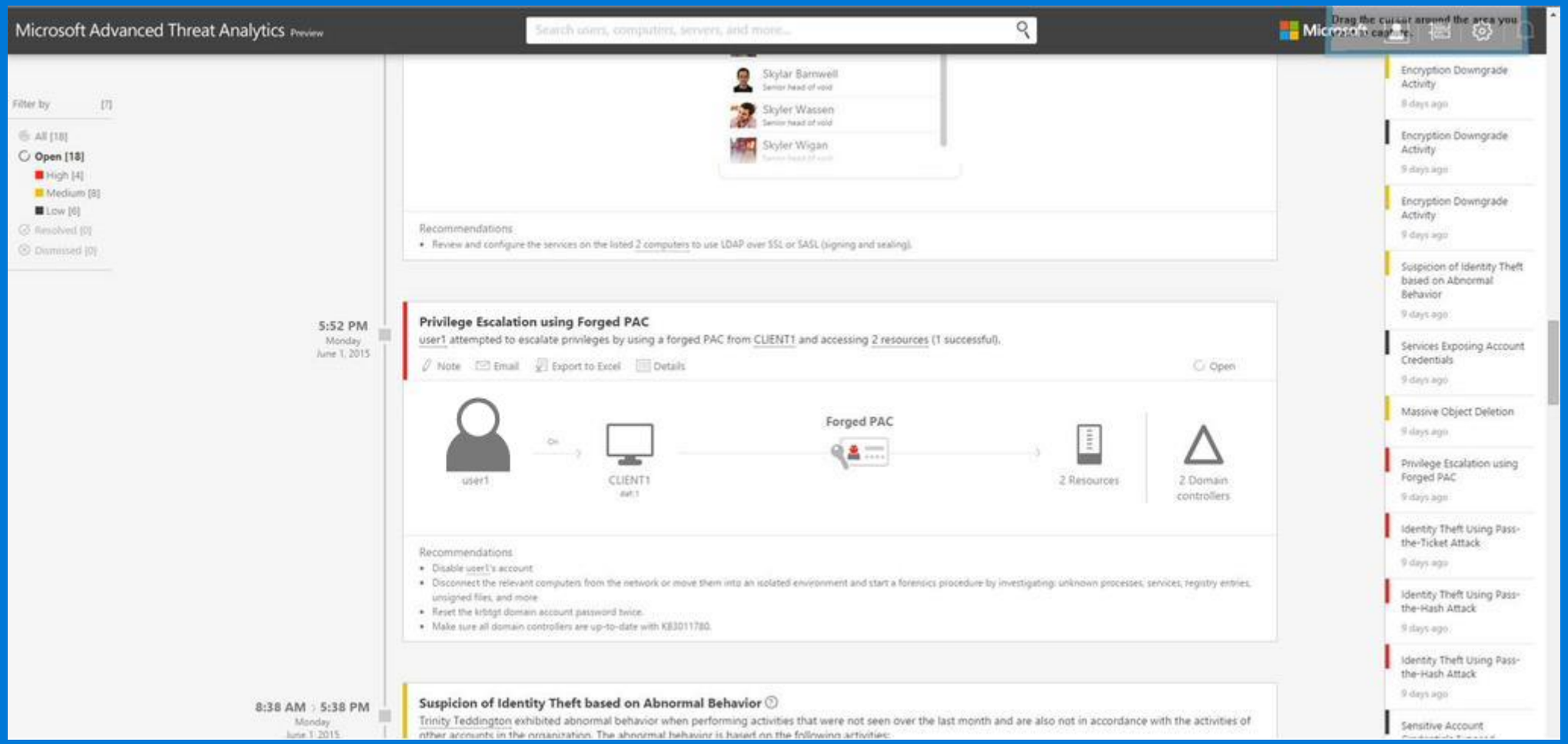
Gavin Millard @gmillard · 11h
MS14-068 in the real world.
"Welcome Captain. Would you like a
#infosec
UNITED FLIGHT
FLIGHT D3048
Coach
FIRST NAME
BOARDS AT
BOARDING ZONE
2
DEPARTS
3:40PM
UNITED BOARDING PASS
UNITED
> SC US
SC UK
Andromeda botnet used to distribute new point-of-sale malware
Ohio inmate caught with pris administrative login credentials
NEWS
PRODUCTS
BLOGS
RESOURCES
VIDEOS
SC MAGAZINE
FOR IT SECURITY PROFESSIONALS
SC Magazine > News > Duqu2.0 knocks Kaspersky and security peers
Tom Reeve, senior report, SC Magazine UK
June 11, 2015
Duqu2.0 knocks Kaspersky and security peers
31
12

Network–based Detection

- Unpatched servers:
  - PA-PAC-REQUEST == FALSE
  - ENC-AuthorizationData != NULL
  - Target Server needs PAC (NA bit == FALSE)
- Patched servers:
  - KDC returns error, as the signature type is not supported

```
⊟ Kerberos
  ⊞ Record Mark: 400 bytes
  ⊟ krb-error
      pvno: 5
      msg-type: krb-error (30)
      stime: 2014-12-05 12:36:30 (UTC)
      susec: 312573
      error-code: eRR-SUMTYPE-NOSUPP (15)
      realm: AORATO.RESEARCH
```

Network–based Detection



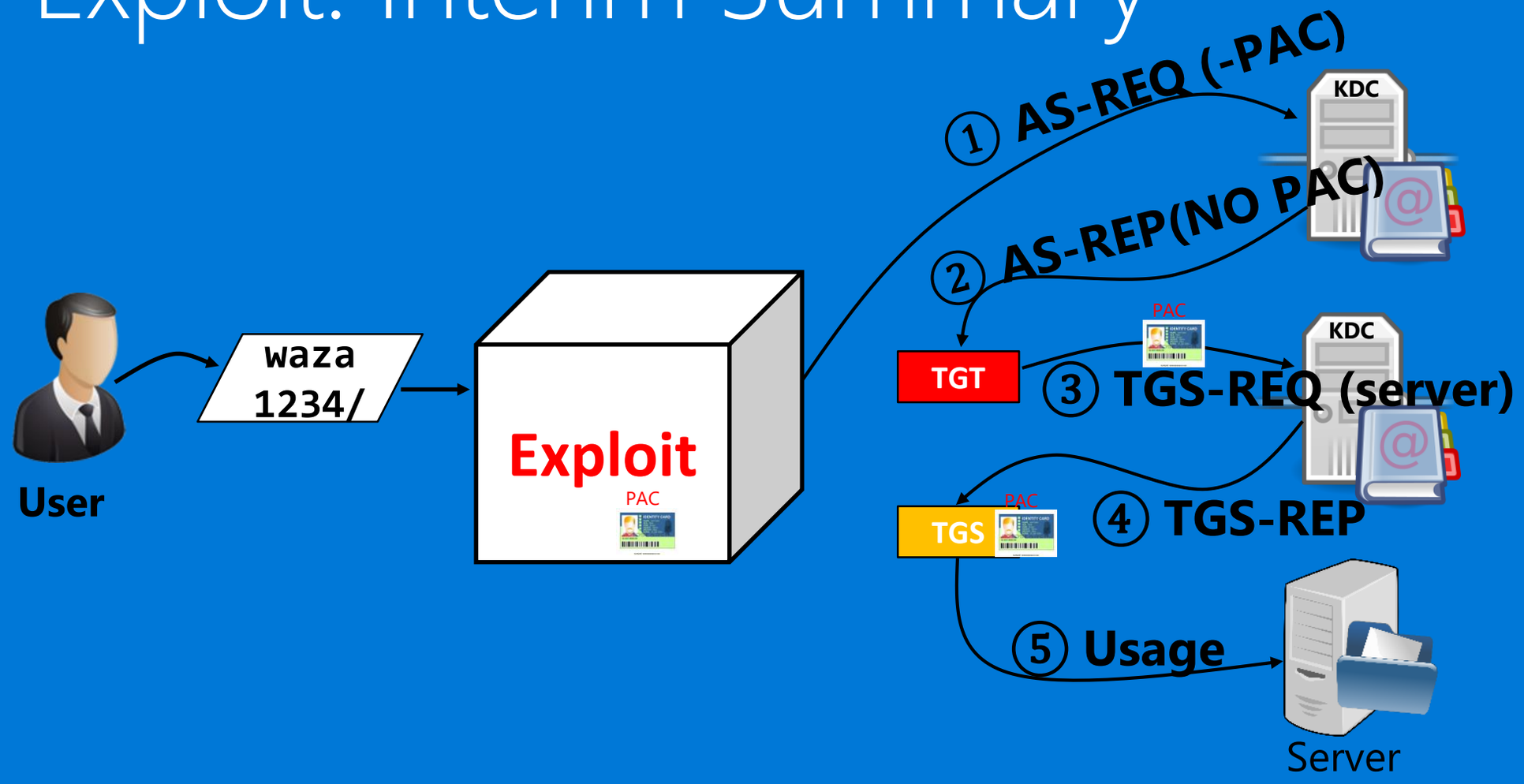
Forged PAC: Diamond PAC

What's Diamond PAC?

- A novel attack variant!
- An offspring of two attacks:
- Golden Ticket - The attacker obtains KRBTGT key
- Fake PAC – Rogue PAC is embedded within a legitimate ticket

Diamond Photograph courtesy of the U.S. Geological Survey

The Exploit: Interim Summary



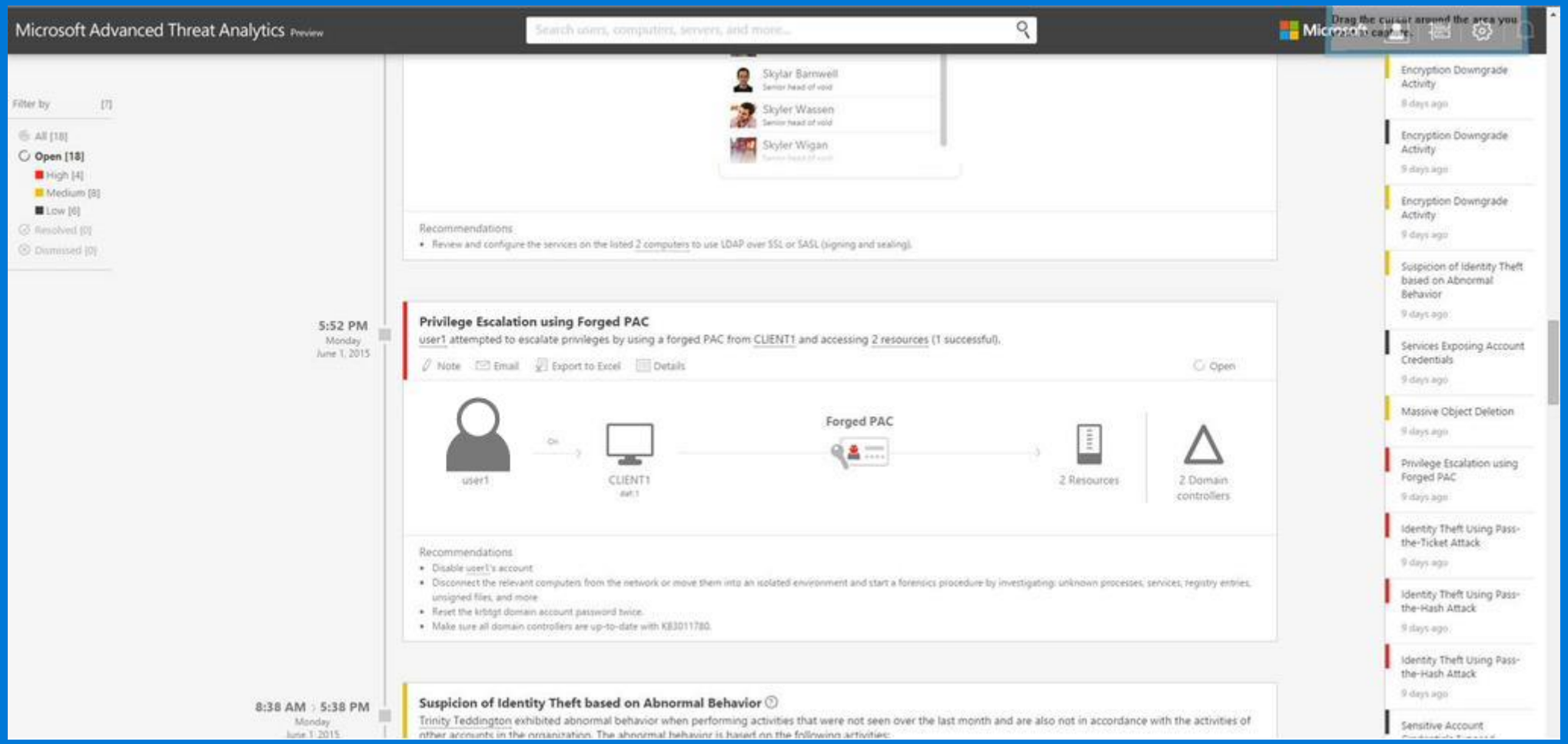
Attacker motivation

- Why use the son (Diamond PAC) if you can use its parents (Golden Ticket & Fake PAC)?
- Fake PAC is patched by MS14-068
- Golden Ticket is detectable with basic network monitoring
  - Recoupling AS-REP with TGS-REQ
- With Diamond PAC
  - Unbounded privileges, like Golden Ticket
  - But with a legitimate “family tree”

Network-Based Detection

- PA-PAC-REQUEST == FALSE
- ENC-AuthorizationData != NULL
- Target Server needs PAC (NA bit == FALSE)
- Very much like the Fake PAC detection

Network-based Detection



Parting Thoughts

Kerberos Leash

- We need to put a network monitoring leash on that Kerberos dog!
- Free Downloads
  - Skeleton Key Scanner (including source code)
    - https://gallery.technet.microsoft.com/Aorato-Skeleton-Key-24e46b73
  - Microsoft Advanced Threat Analytics (ATA) evaluation version
    - https://www.microsoft.com/en-us/evalcenter/evaluate-microsoft-advanced-threat-analytics

Conclusions

- Advanced attackers do not stop at stealing existing tokens
- They are subverting the inner workings of the Kerberos system
- Using network monitoring we can protect our networks from both attacks paradigms

Questions?