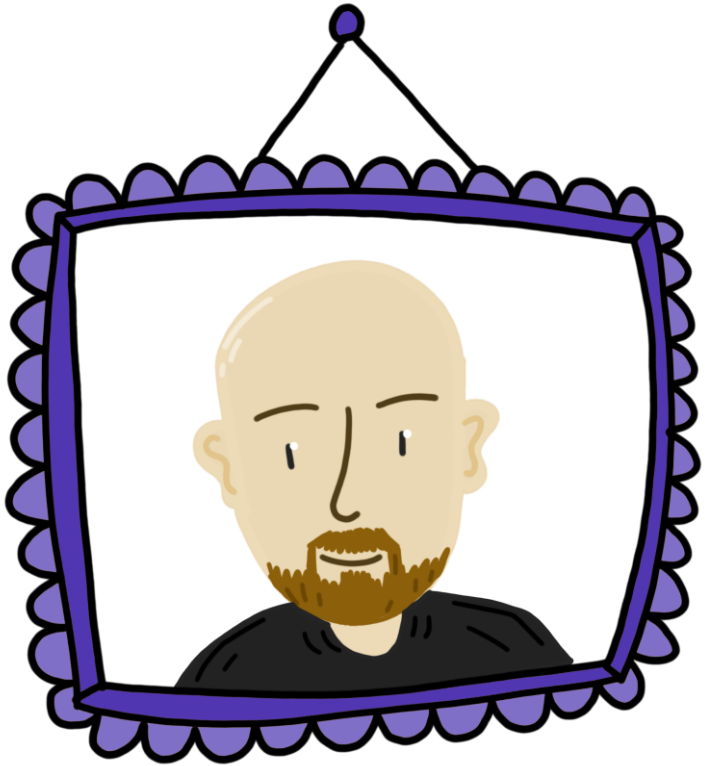
I've Got a Forged Twinkle In My Eye

Charlie Clark & Andrew Schwartz

TODAY'S TOPICS

- INTRODUCTIONS
- KERBEROS **101**
- FORGED TICKET **101**
- DETECTION METHODS
- DEMOS & **IOAS**



CHARLIE CLARK

- @EXPLOITPH
- SECURITY RESEARCHER @ SEMPERIS
- ATTACKER OF KERBEROS
- STREAMING MEDIA AFICIONADO

ANDREW SCHWARTZ

- @4NDR3W6S
- PRACTICE LEAD @ TRUSTEDSEC
- ATTACK/DETECT ALL THE THINGS
- TOTTENHAM SUPER FAN (COYS!)

A FEW DISCLAIMERS

- AD IS NOT "STATIC"
- WE WILL BE EXPLICIT ON WHAT WE CAN DETECT
- ANY DETECTION CAN BY BYPASSED
- WE DO NOT "STOP" FORGED TICKETS
- ~~WE ARE NOT FOCUSING ON SILVER TICKETS~~
- FOCUS SHOULD BE ON THE TECHNIQUE

Welcome to Kerberos 101!

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5

LSA/LSASS

2

3

6

4

7

8

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

CIFS

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5

LSA/LSASS

2 CREATE HASH FROM CREDENTIALS:
RC4 - USER'S PASSWORD
AES- PASSWORD + SALT

3

6

4

7

8

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5

LSA/LSASS

2 CREATE HASH FROM CREDENTIALS:
RC4 – USER'S PASSWORD
AES- PASSWORD + SALT

3 ENCRYPT PART OF REQUEST WITH HASH

6

4

7

8

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5

LSA/LSASS

2 CREATE HASH FROM CREDENTIALS:
RC4 – USER'S PASSWORD
AES- PASSWORD + SALT

3 ENCRYPT PART OF REQUEST WITH HASH

6

4 TGT RETRIEVAL

7

8

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5 ACCESS ATTEMPT/ACTION

dir \\fs.chocolatefactory.local\C$

LSA/LSASS

2 CREATE HASH FROM CREDENTIALS:
RC4 - USER'S PASSWORD
AES - PASSWORD + SALT

3 ENCRYPT PART OF REQUEST WITH HASH

6

4 TGT RETRIEVAL

7

8

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5 ACCESS ATTEMPT/ACTION

dir\\fs.chocolatefactory.local\C$

LSA/LSASS

2 CREATE HASH FROM CREDENTIALS:
RC4 - USER'S PASSWORD
AES- PASSWORD + SALT

3 ENCRYPT PART OF REQUEST WITH HASH

6 USE TGT TO REQUEST SERVICE TICKET

4 TGT RETRIEVAL

7

8

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

CIFS

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5 ACCESS ATTEMPT/ACTION

dir \\fs.chocolatefactory.local\C$

LSA/LSASS

2 CREATE HASH FROM CREDENTIALS:
RC4 – USER'S PASSWORD
AES- PASSWORD + SALT

3 ENCRYPT PART OF REQUEST WITH HASH

6 USE TGT TO REQUEST SERVICE TICKET

4 TGT RETRIEVAL

7 SERVICE TICKET RETRIEVAL

8

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5 ACCESS ATTEMPT/ACTION

dir \\fs.chocolatefactory.local\C$

LSA/LSASS

2 CREATE HASH FROM CREDENTIALS:
RC4 - USER'S PASSWORD
AES- PASSWORD + SALT

3 ENCRYPT PART OF REQUEST WITH HASH

6 USE TGT TO REQUEST SERVICE TICKET

4 TGT RETRIEVAL

7 SERVICE TICKET RETRIEVAL

8 SERVICE ACCESS

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 USERNAME: CBUCKET
PASSWORD: GOLD3NTICK3T

LOGIN

5 ACCESS ATTEMPT/ACTION

dir \\fs.chocolatefactory.local\C$

LSA/LSASS

2 CREATE HASH FROM CREDENTIALS:
RC4 - USER'S PASSWORD
AES - PASSWORD + SALT

3 ENCRYPT PART OF REQUEST WITH HASH

6 USE TGT TO REQUEST SERVICE TICKET

4 TGT RETRIEVAL
(AS-REQ/AS-REP)

7 SERVICE TICKET RETRIEVAL
(TGS-REQ/TGS-REP)

8 SERVICE ACCESS
(AP-REQ/AP-REP)

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

Lets talk Golden Tickets!

Lets talk Golden Tickets!

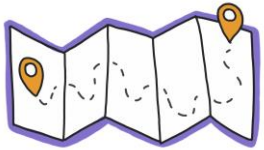
What is a Golden Ticket?

A Golden Ticket is any TGT that is NOT ISSUED by a DC

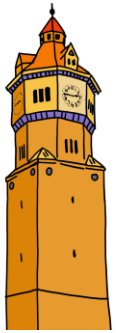
Attacker Motives:

Golden Ticket 101

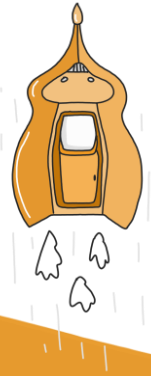
Impersonation



Unfettered Access

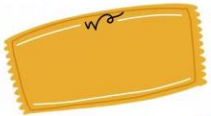


Persistence



Privilege escalation across a trust

1 CREATION OF GOLDEN TICKET (GT)



2

3

LSA/LSASS

4

5

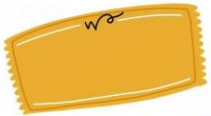
6

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF GOLDEN TICKET (GT)



2 INJECTION OF GT

LSA/LSASS

4

3

6

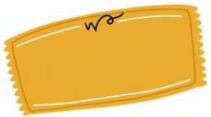
5

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF GOLDEN TICKET (GT)



2 INJECTION OF GT

LSA/LSASS

4

3 ACCESS ATTEMPT/ACTION

dir\\fs.chocolatefactory.local\c$

6

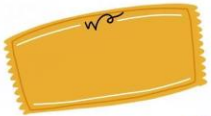
5

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF GOLDEN TICKET (GT)



2 INJECTION OF GT

3 ACCESS ATTEMPT/ACTION

dir\\fs.chocolatefactory.local\c$

LSA/LSASS

4 USE GOLDEN TICKET TO

REQUEST SERVICE TICKET

5

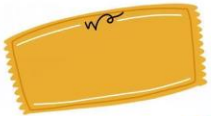
6

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF GOLDEN TICKET (GT)



2 INJECTION OF GT

LSA/LSASS

4 USE GOLDEN TICKET TO

REQUEST SERVICE TICKET

3 ACCESS ATTEMPT/ACTION

dir\\fs.chocolatefactory.local\c$

6

5

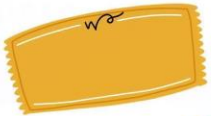
SERVICE TICKET RETRIEVAL
(TGS-REQ/TGS-REP)

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF GOLDEN TICKET (GT)



2 INJECTION OF GT

LSA/LSASS

4 USE GOLDEN TICKET TO

REQUEST SERVICE TICKET

3 ACCESS ATTEMPT/ACTION

dir\\fs.chocolatefactory.local\c$

5 SERVICE TICKET RETRIEVAL
(TGS-REQ/TGS-REP)

6 SERVICE ACCESS
(AP-REQ/AP-REP)

dc.chocolatefactory.local
(DOMAIN CONTROLLER)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

What about Silver Tickets!?

What about Silver Tickets!?

What is a Silver Ticket?

A Silver Ticket is any ST that is NOT ISSUED by a DC

Attacker Motives:

Impersonation

Avoiding Communication with the DC

1 CREATION OF SILVER TICKET (ST)

2

3

LSA/LSASS

4

5

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF SILVER TICKET (ST)

2 INJECTION OF ST

LSA/LSASS

4

3

5

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF SILVER TICKET (ST)

2 INJECTION OF ST

3 ACCESS ATTEMPT/ACTION

dir\\fs.chocolatefactory.local\c$

LSA/LSASS

4

5

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF SILVER TICKET (ST)

2 INJECTION OF ST

3 ACCESS ATTEMPT/ACTION

dir\\fs.chocolatefactory.local\c$

LSA/LSASS

4 USE SILVER TICKET TO ACCESS

SERVICE

5

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF SILVER TICKET (ST)

2 INJECTION OF ST

3 ACCESS ATTEMPT/ACTION

dir\\fs.chocolatefactory.local\c$

LSA/LSASS

4 USE SILVER TICKET TO ACCESS

SERVICE

5 SERVICE ACCESS
(AP-REQ/AP-REP)

fs.chocolatefactory.local
(SERVER)

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF SILVER TICKET (ST)

2 INJECTION OF ST

3 ACCESS ATTEMPT/ACTION

dir \\fs.chocolatefactory.local\c$

LSA/LSASS

4 USE SILVER TICKET TO ACCESS

SERVICE

5 SERVICE ACCESS (AP-REQ/AP-REP)

fs.chocolatefactory.local (SERVER)

NOTHING IS SENT TO THE DC

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

1 CREATION OF SILVER TICKET (ST)

2 INJECTION OF ST

3 ACCESS ATTEMPT/ACTION

dir \\fs.chocolatefactory.local\c$

LSA/LSASS

4 USE SILVER TICKET TO ACCESS

SERVICE

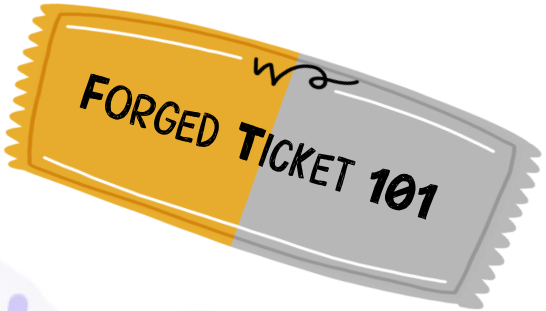
5 SERVICE ACCESS (AP-REQ/AP-REP)

fs.chocolatefactory.local (SERVER)

- NOTHING IS SENT TO THE DC
- UNLESS SERVICE ON DC IS BEING ACCESSED

ORIGINAL CONCEPT & INSPIRATION FROM BENJAMIN DELPY & WILL SCHROEDER

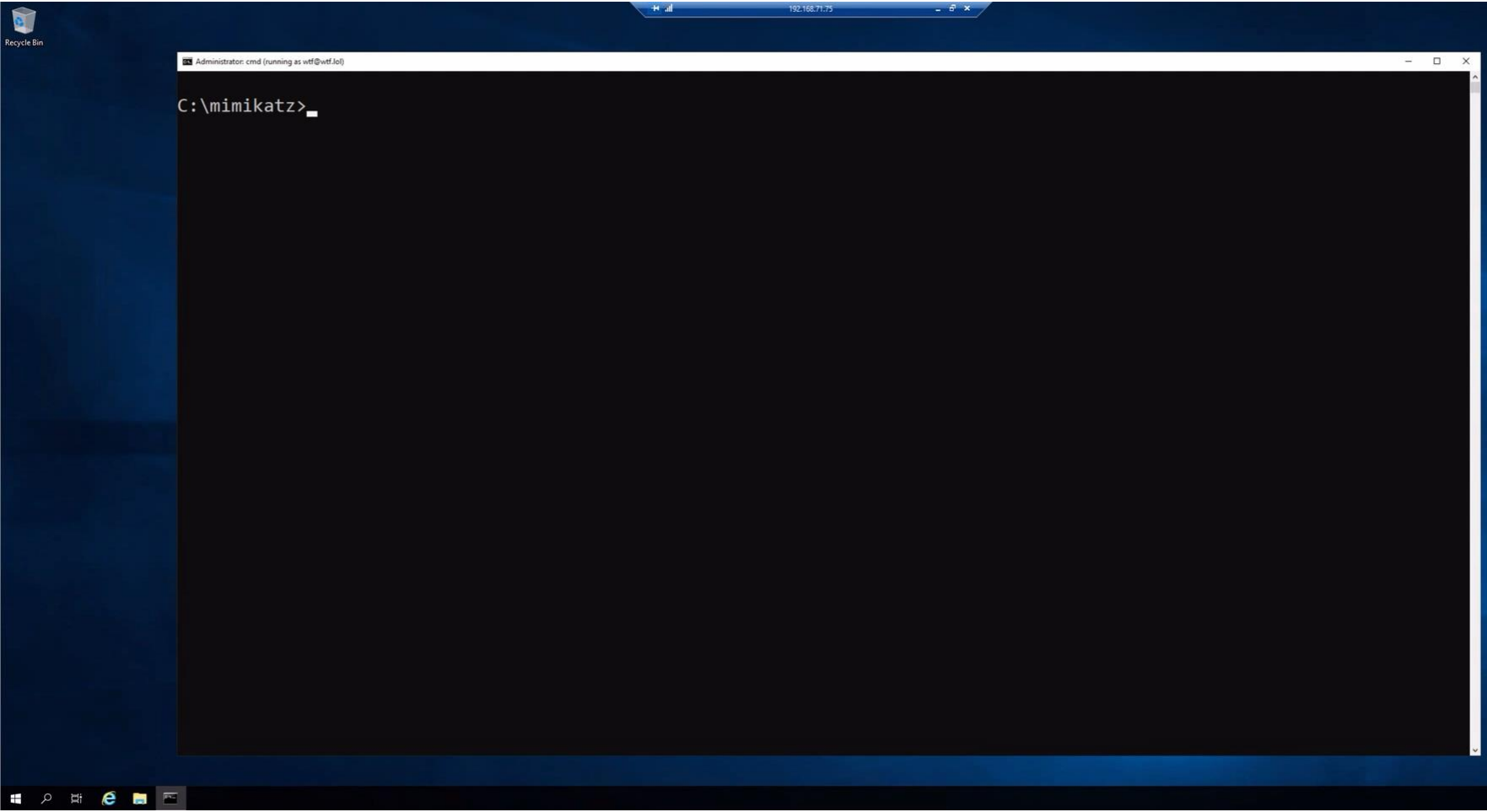
FORGED TICKET CREATION FLOW



NO REQUIREMENT OF NETWORK TRAFFIC BEING SENT DURING THIS PROCESS

DEMO

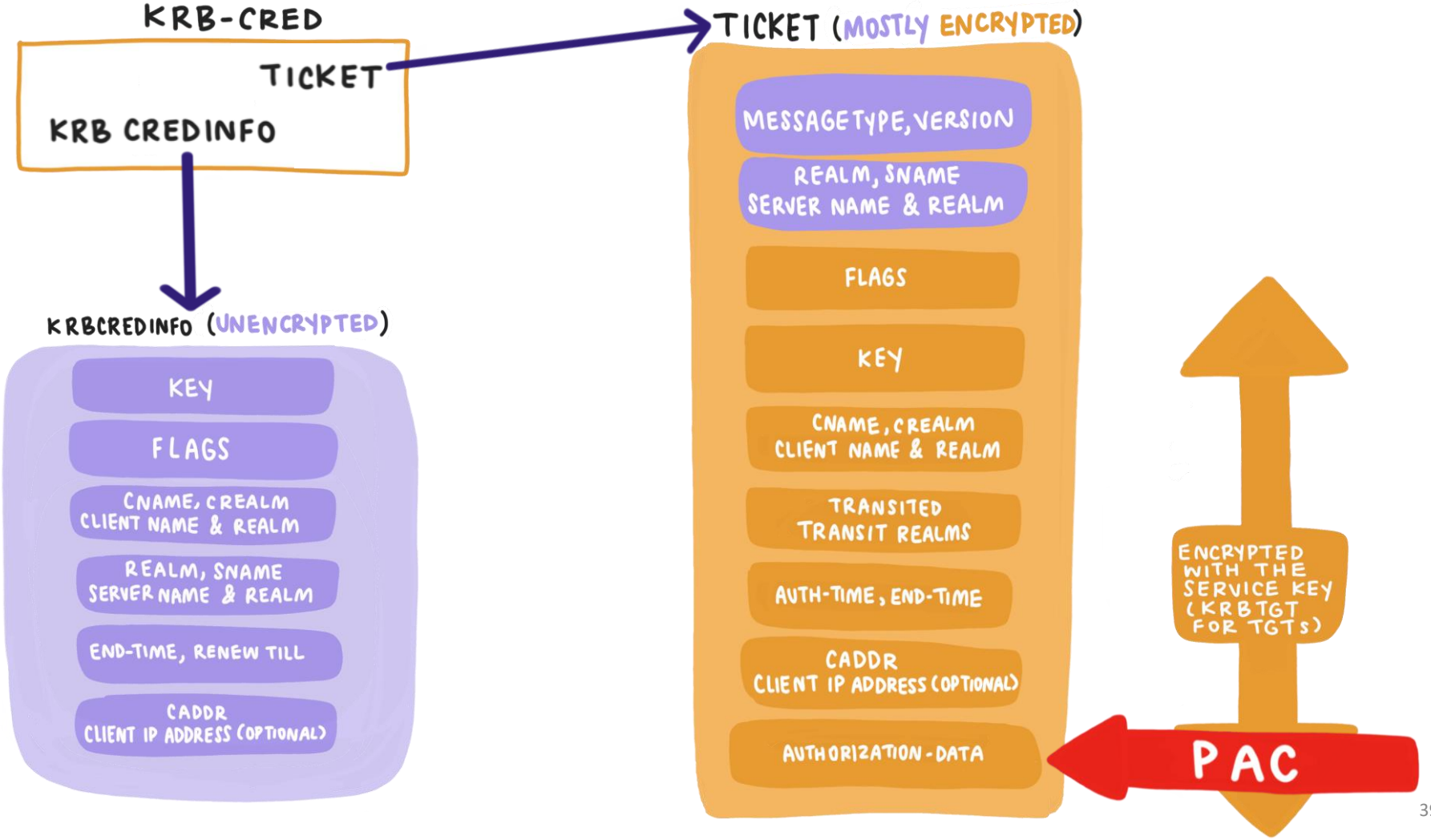
DEMO 1: GOLDEN TICKET CREATION & USAGE


192.168.71.75
Administrator: cmd (running as wtf@wtf.lol)
C:\mimikatz>
Recycle Bin
23:01
05/11/2022

CURRENT DETECTION METHODS & DRAWBACKS

EVTX **4768** EVTX **4769**

- EVENT IDS ARE INCREDIBLY "NOISY"
- OVER RELIANCE WITHOUT EXTRA PRODUCTS
- LACK OF ACTIONABLE INFORMATION
  - ENCRYPTION TYPE (NOT SESSION KEY TYPE)
  - TICKET TIMES
  - FLAGS (NOT KDC OPTIONS)
- NON-EXISTENT FOR SILVER TICKETS

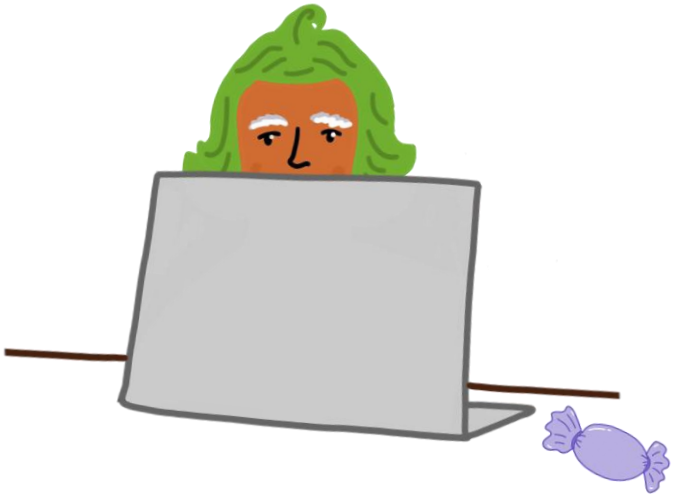

KRB-CRED
TICKET
KRB CREDINFO
KRBCREDINFO (UNENCRYPTED)
KEY
FLAGS
CNAME, CREALM
CLIENT NAME & REALM
REALM, SNAME
SERVER NAME & REALM
END-TIME, RENEW TILL
CADDR
CLIENT IP ADDRESS (OPTIONAL)
TICKET (MOSTLY ENCRYPTED)
MESSAGETYPE, VERSION
REALM, SNAME
SERVER NAME & REALM
FLAGS
KEY
CNAME, CREALM
CLIENT NAME & REALM
TRANSITED
TRANSIT REALMS
AUTH-TIME, END-TIME
CADDR
CLIENT IP ADDRESS (OPTIONAL)
AUTHORIZATION-DATA
ENCRYPTED WITH THE SERVICE KEY (KRBTGT FOR TGTS)
PAC

Levels of access to ticket telemetry

- 1st level: ticket only
- 2nd level: ticket + the service key ← required to view the **PAC**
- 3rd level: ticket + the service key & the **krbtgt** key (ST)

More keys = more insight into a potential forged ticket

TICKET DECRYPTION



REQUIREMENT

- "ON-THE-FLY" DECRYPTION REQUIRES KEYS TO BE DCSYNCED

BENEFITS

- ENABLES FULL TICKET ANALYSIS
- EXPOSES ADDITIONAL INFO TO BUILD IOAS (EX: SIGNATURES, USER ATTRIBUTES)

Practical example 1 - LSA:

- Dump sessions/tickets from **LSA**
- Encrypt dumps
- **DCS**ync keys for decryption and signature verification
- **Q**uery **DC** for proper information to perform analysis
- **W**rite output to the event log

WonkaVision: A Tool to Detect Forged Tickets

Initial Approach

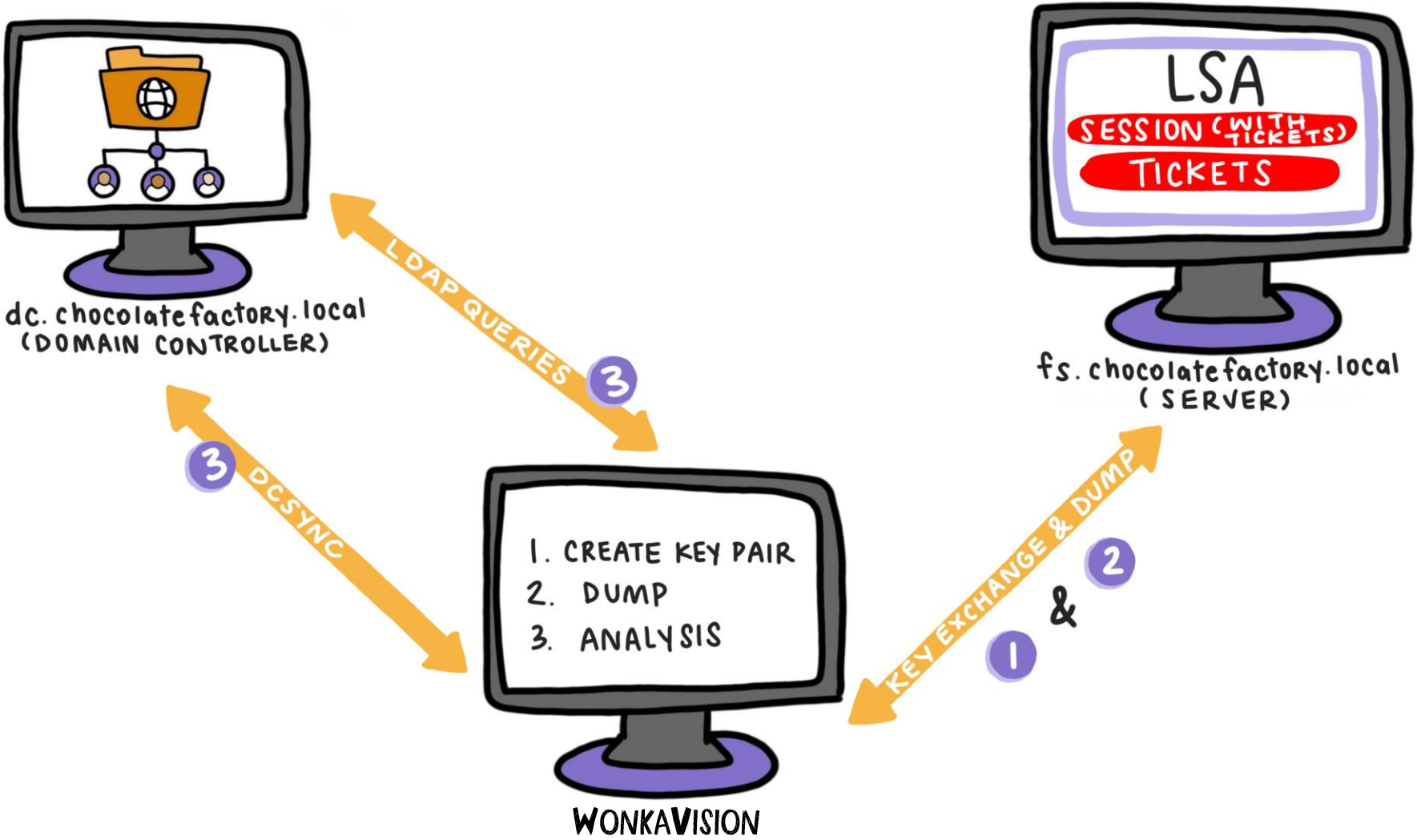
- Utilized "capability abstraction" with known **OST**s

Execution

- Attack simulation and compare against "known good"

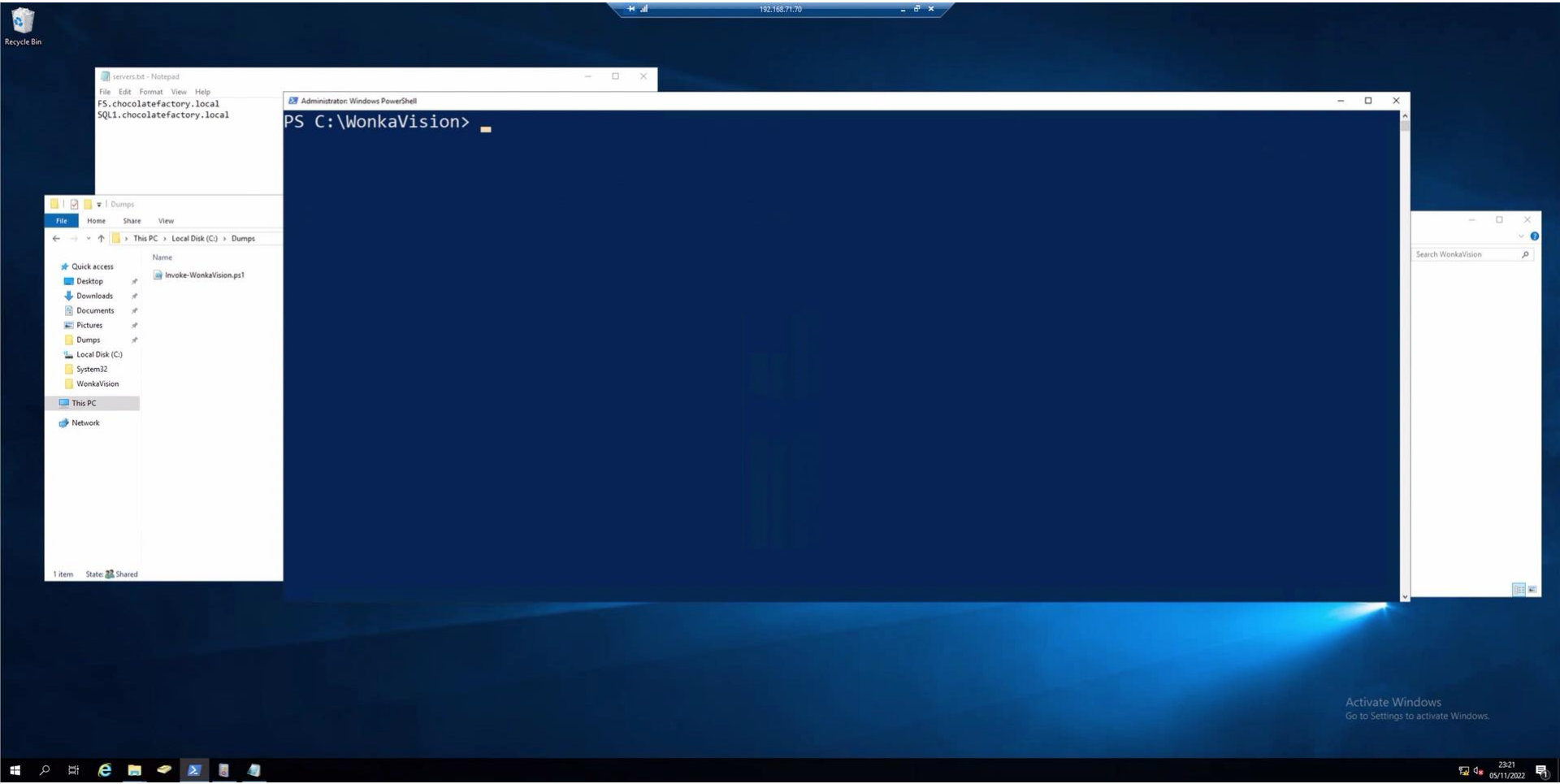
Outcome

- Accumulative scoring system with **OST** fingerprinting resulting in improved detection


dc.chocolatefactory.local
(DOMAIN CONTROLLER)
LDAP QUERIES
3
LSA
SESSION (WITH TICKETS)
TICKETS
fs.chocolatefactory.local
(SERVER)
3
DCSYNC
1. CREATE KEY PAIR
2. DUMP
3. ANALYSIS
KEY EXCHANGE & DUMP
2
&
1
WONKAVISION

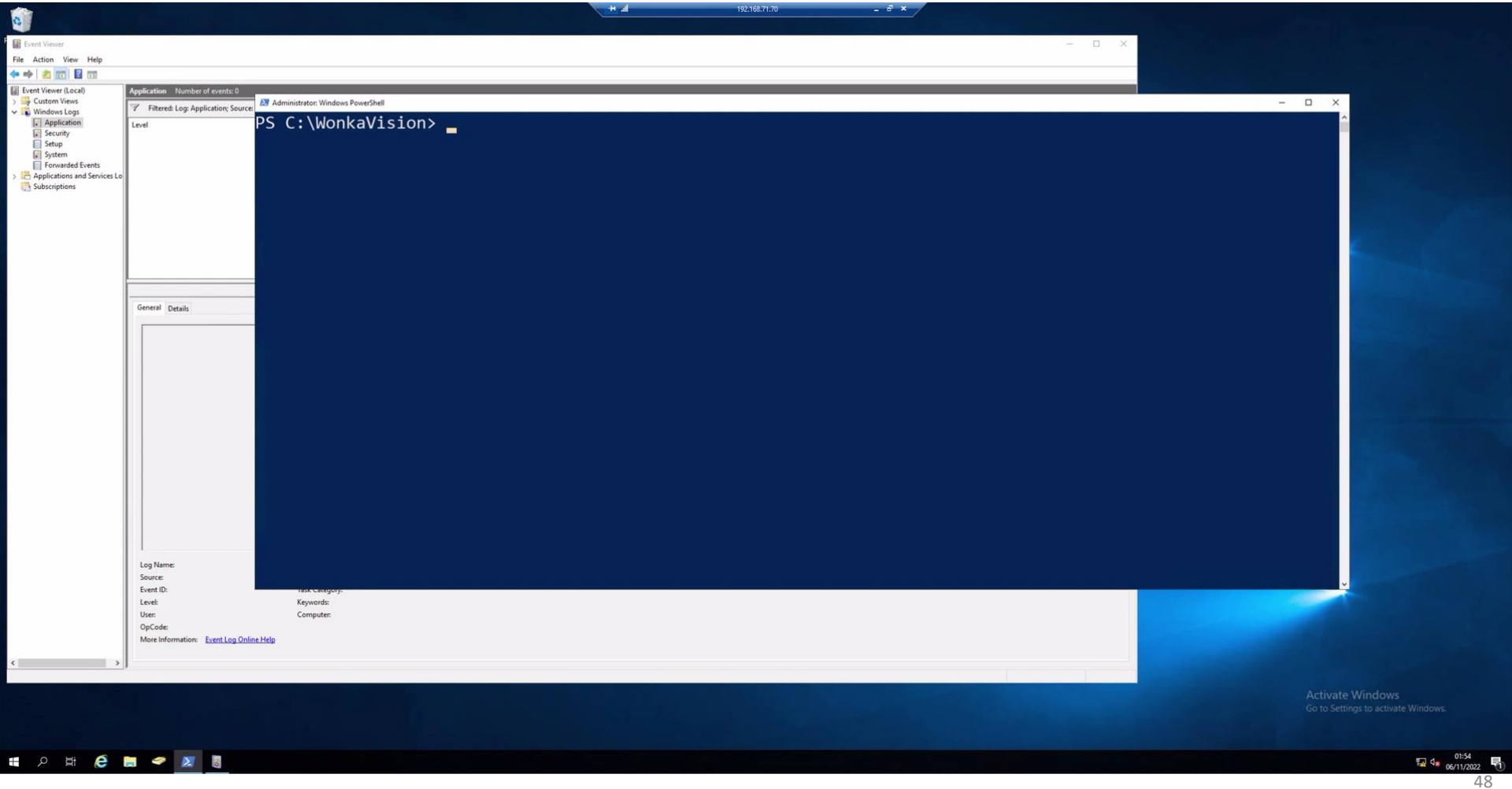
DEMO

DEMO 2: WV KEY CREATION & SESSION/DUMPING COMMANDS

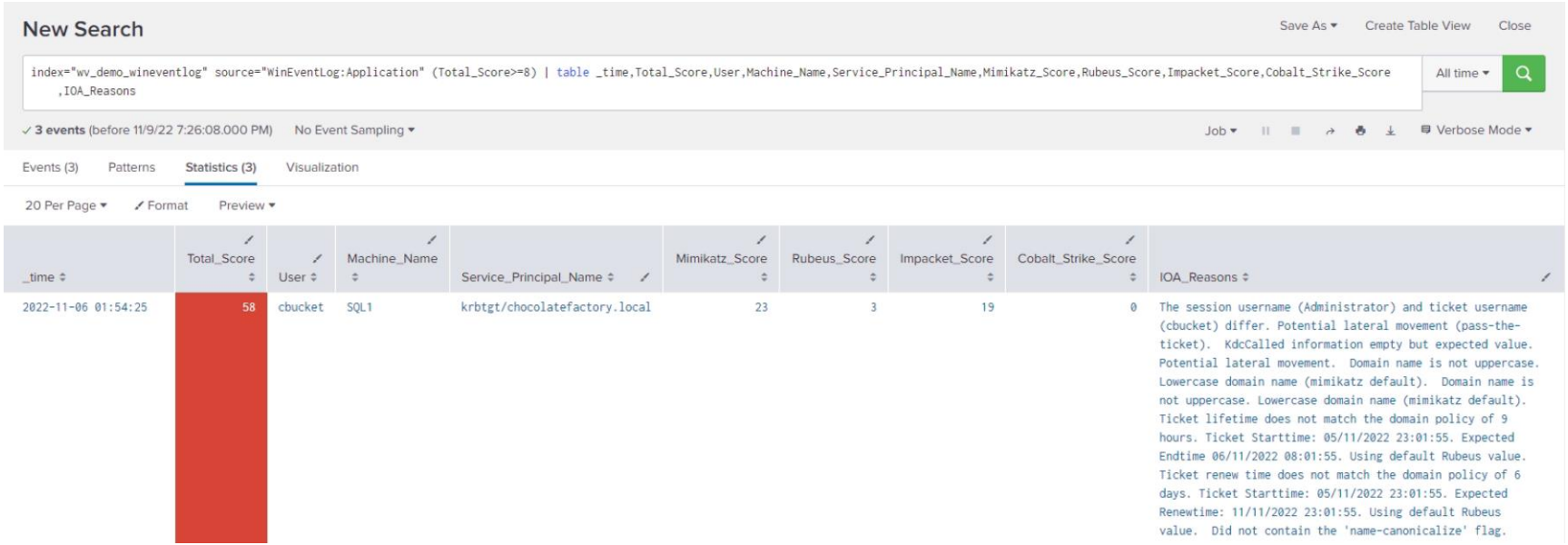

Recycle Bin
192.168.71.70
servers.txt - Notepad
File Edit Format View Help
FS.chocolatefactory.local
SQL1.chocolatefactory.local
Administrator: Windows PowerShell
PS C:\WonkaVision>
Dumps
File Home Share View
This PC › Local Disk (C:) › Dumps
Quick access
Desktop
Downloads
Documents
Pictures
Dumps
Local Disk (C:)
System32
WonkaVision
This PC
Network
Name
Invoke-WonkaVision.ps1
1 item State: Shared
Search WonkaVision
Activate Windows
Go to Settings to activate Windows.
23:21
05/11/2022

DEMO

DEMO 3: ANALYSIS COMMAND & OUTPUT IN EVTX


192.168.71.70
Event Viewer
File Action View Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services Lo
Subscriptions
Application Number of events: 0
Filtered: Log: Application; Source:
Level
General Details
Log Name:
Source:
Event ID:
Level:
User:
OpCode:
More Information: Event Log Online Help
Keywords:
Computer:
Administrator: Windows PowerShell
PS C:\WonkaVision>
Activate Windows
Go to Settings to activate Windows.
01:54
06/11/2022

SIEM LOG FORWARDING - SPLUNK



SIEM LOG FORWARDING - SENTINEL

TimeGenerated [UTC]	ParsedEventData	Source	EventLog	Computer	EventLevel	EventLevelNam

ParsedEventData	{"DataItem":{"@type":"System.XmlData","@time":"2022-11-05T13:39:39.5008352Z","@sourceHealthServiceId":"b9237f0b-b5e5-ba00-0951-44654bfd7af6","EventData":{"@xmlns":"http://schemas.micr
TotalScore	17
Session	0x156af5
MachineName	Asgard-Wrkstn
User	thanos
ServicePrincipalName	krbtgt/marvel.local
IOAs	
IOA_SessionUser	thor
IOA_KDCCalled	TicketFlags: pre_authent, initial, renewable, forwardable UpnDNSBuffer: Not Extended RequestorBuffer: None AttributesBuffer: None Tool Scores:
TScore_MimikatzScore	4
TScore_ImpacketScore	4
TScore_RubeusScore	2
TScore_CobaltStrikeScore	0
IOA_Reasons	The session username (thor) and ticket username (thanos) differ. Potential lateral movement (pass-the-ticket). KdcCalled information empty but expected value. Potential lateral movement. Did not c

THANKS TO JONATHAN JOHNSON (@JSECURITY101) FOR PROVIDING THIS SCREENSHOT

Interesting IOAs - Checksums

(New(ish)) checksums

```
TicketChecksum          :
  Signature Type        : KERB_CHECKSUM_HMAC_SHA1_96_AES256
  Signature             : 5D08D7468337D30FA10AB71D (UNVALIDATED)
FullPacChecksum         :
  Signature Type        : KERB_CHECKSUM_HMAC_SHA1_96_AES256
  Signature             : A2D7C909B00696FF52B89F1C (UNVALIDATED)
```

- Only present in non-krbtgt service tickets
- Currently only supported in Rubeus
- Rubeus incorrectly includes them in referrals

INTERESTING IOAS - CHECKSUMS - CONTINUED

```
C:\Rubeus>Rubeus.exe silver /user:wonka /service:CIFS/FS.chocolatefactory.local /aes256:2a02ac51416470cc96186b951716f1f9
1ad55e5284cc19eb07f814ff3aedf923 /ldap /nowrap

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.1

[*] Action: Build TGS

[*] Trying to query LDAP using LDAPS for user information on domain controller DC1.chocolatefactory.local
[*] Searching path 'DC=chocolatefactory,DC=local' for '(samaccountname=wonka)'
[*] Retrieving group and domain policy information over LDAP from domain controller DC1.chocolatefactory.local
[*] Searching path 'DC=chocolatefactory,DC=local' for '(|(distinguishedname=CN=Protected Users,CN=Users,DC=chocolatefact
ory,DC=local)(distinguishedname=CN=Domain Admins,CN=Users,DC=chocolatefactory,DC=local)(objectsid=S-1-5-21-652637156-682
571310-46925969-513)(name={31B2F340-016D-11D2-945F-00C04FB984F9}))'
[*] Attempting to mount: \\dc1.chocolatefactory.local\SYSVOL
[*] \\dc1.chocolatefactory.local\SYSVOL successfully mounted
[*] Attempting to unmount: \\dc1.chocolatefactory.local\SYSVOL
[*] \\dc1.chocolatefactory.local\SYSVOL successfully unmounted
[*] Retrieving netbios name information over LDAP from domain controller DC1.chocolatefactory.local
[*] Searching path 'CN=Configuration,DC=chocolatefactory,DC=local' for '(&(netbiosname=*)(dnsroot=chocolatefactory.local
))'
[*] Building PAC
```

```
C:\Rubeus>Rubeus.exe describe /servicekey:2a02ac51416470cc96186b951716f1f91ad55e5284cc19eb07f814ff3aedf923 /krbkey:2a02a
c51416470cc96186b951716f1f91ad55e5284cc19eb07f814ff3aedf923 /ticket:doIFwzCCBb+gAwIBBaEDAgEWooIEkzCCBI9hggSLMIIEh6ADAgEF
oRgbFkNIT0NPTEFURUZBQ1RPUlkuTE9DQUyiLDAqoAMCAQKhIzAhGwRDSUZTGxlGUy5jaG9jb2xhdGVmYWN0b3J5LmxvY2Fso4IENjCCBDKgAwIBEqEDAgED
ooIEJASCBCA90KV1DL4m2YIKgoEWjxmQOhG/9niPhnCr1XN66t4U21D+xvBQ0Zh6XNhsaZJb0BG4SLJp+qwc6/t0Izf6xWLRFu70sc+CjjKKH767JACLsHun
GhUxvB55VoQ1Su+gi1hqD8DIVgZauTCBN/+0BSJoIXwY6S2IZLmKnMp/kai+tuhUjMBeD8iABqcBztnENAWGqvpr1aOIhbkn7AyZzSEp7H9EqR50KcU97pCG
ZOiNHB1e5gq22WBP+XqKHWZjWeDrr9hGZOCwnb3Z9695FkgvIaQ8pApixrO9JC2WBaxbLJTLTPBV0NDetyQnPKyJEndEOvVBS9EDIB7IWTx/oMuZW1oxoLSG
zH02r9G6zznABJyICt7+g2H7wBmiKXK1mfYBnNVxiEerwHQn3QZ5wSwz36ajqP4ZXBocLdjAT+1V6FWIo6NKBm25wJh9B2m2cPYlTsbUvtAJl+qn2S/dnp6M
TRYAQ29wo45FInxxx4oUV26rabNyEZ3JFbsYsZn6F1eVUvJ+pWIrSDWmYjvDTbZ8qLTraX1Oz8sGM7NGTusmQE8g29eEDb6GutKAmy6VQYazwOHf02JiAmZe
```

```
    ServerChecksum       :
      Signature Type     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
      Signature          : DAB1F78E19402F285DC5B128 (VALID)
    KDCChecksum          :
      Signature Type     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
      Signature          : 597B075A5AAB12AE04D2FCD9 (VALID)
    TicketChecksum       :
      Signature Type     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
      Signature          : 1BD4A1556811D2DC113E8FB3 (VALID)
    FullPacChecksum      :
      Signature Type     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
      Signature          : 091588D32039ADD75E3B3D70 (VALID)
```

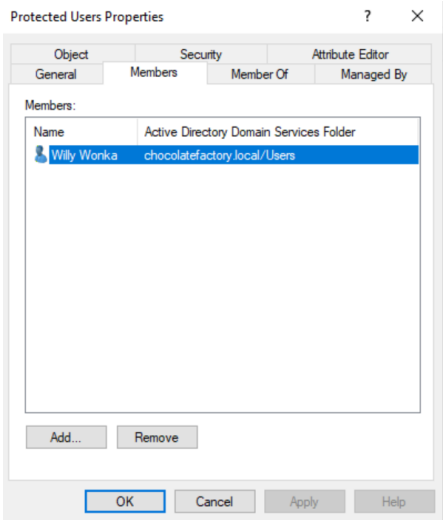
ANY KRBTGT SIGNED CHECKSUM SIGNED WITH THE SERVICE KEY IS A **HUGE** RED FLAG

Interesting IOAs - Ticket Times

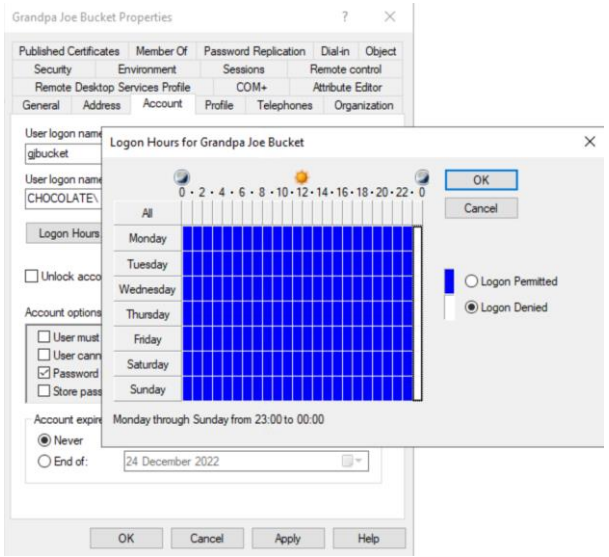
Domain Policy

```
PS C:\> (Get-DomainPolicy -Policy Domain).KerberosPolicy

MaxTicketAge           : 9 hours
MaxRenewAge            : 6 days
MaxServiceAge          : 530 minutes
MaxClockSkew           : 5
TicketValidateClient : 1
```

Protected Users Group



Logon Hours



Which ever is earliest takes priority

Interesting IOAs - Ticket Times - Continued

Genuine TGT

```
ServiceName              :  krbtgt/CHOCOLATEFACTORY.LOCAL
ServiceRealm             :  CHOCOLATEFACTORY.LOCAL
UserName                 :  wonka
UserRealm                :  CHOCOLATEFACTORY.LOCAL
StartTime                :  24/11/2022 22:35:20
EndTime                  :  25/11/2022 02:35:20
RenewTill                :  25/11/2022 02:35:20
Flags                    :  name_canonicalize, pre_authent, initial, renewable
KeyType                  :  aes256_cts_hmac_sha1
Base64(key)              :  anLa1lT0tUM4bJsKV/fv32S/pt5TTArDvCvep5kFXFA=
ASREP (key)              :  9448219BB7DDABAEF2B2282C39294D128665197E92EDC1627688C87880114FA8
```

Sapphire ticket

```
ServiceName              :  krbtgt/CHOCOLATEFACTORY.LOCAL
ServiceRealm             :  CHOCOLATEFACTORY.LOCAL
UserName                 :  wonka
UserRealm                :  CHOCOLATEFACTORY.LOCAL
StartTime                :  24/11/2022 22:22:59
EndTime                  :  25/11/2022 07:12:59
RenewTill                :  25/11/2022 22:22:59
```

- Both EndTime and RenewTill are 4 hours for members of Protected Users

Interesting IOAs - Ticket Times - Continued

Genuine TGT

```
ServiceName              :  krbtgt/CHOCOLATEFACTORY.LOCAL
ServiceRealm             :  CHOCOLATEFACTORY.LOCAL
UserName                 :  wonka
UserRealm                :  CHOCOLATEFACTORY.LOCAL
StartTime                :  24/11/2022 23:16:16
EndTime                  :  25/11/2022 00:00:00
RenewTill                :  25/11/2022 00:00:00
Flags                    :  name_canonicalize, pre_authent, initial, renewable
KeyType                  :  aes256_cts_hmac_sha1
Base64(key)              :  GiSxJzijK6uzhs4GEO8a470pTdAgy3V1I/086gL5plY=
ASREP (key)              :  9448219BB7DDABAEF2B2282C39294D128665197E92EDC1627688C87880114FA8
```

Rubeus Golden Ticket

```
ServiceName              :  krbtgt/chocolatefactory.local
ServiceRealm             :  CHOCOLATEFACTORY.LOCAL
UserName                 :  wonka
UserRealm                :  CHOCOLATEFACTORY.LOCAL
StartTime                :  24/11/2022 23:17:58
EndTime                  :  25/11/2022 09:17:58
RenewTill                :  01/12/2022 23:17:58
Flags                    :  pre_authent, initial, renewable, forwardable
KeyType                  :  aes256_cts_hmac_sha1
Base64(key)              :  L/D2wVEir+0GEtiZWIWH+lVnSOQYYaaxeMeyPzT6im8=
```

- Unless the logoff time (logon hours) is earlier, then that takes priority

Interesting IOAs - Ticket Flags

- Genuine tickets contain **name_canonicalize** (only Impacket does)
- Impacket always includes **proxiable**
- Genuine **ST**s for services configured for **UD** contain **ok_as_delegate**
- Accounts protected from delegation do **NOT** contain **forwardable**

Interesting IOAs - KDC Called

- Not related to forged tickets but indicates lateral movement (PTT)
- Returned from LSA when requesting KerbQueryTicketCacheEx3Message
- Not supported on 2008-

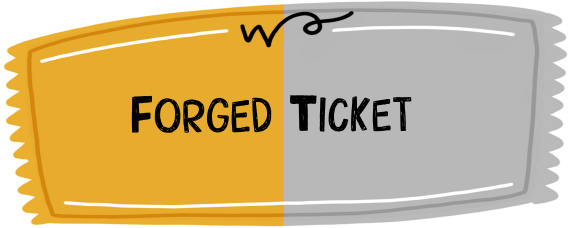
```
C:\>klist

Current LogonId is 0:0xc415a

Cached Tickets: (1)

#0>     Client: cbucket @ CHOCOLATEFACTORY.LOCAL
        Server: krbtgt/CHOCOLATEFACTORY.LOCAL @ CHOCOLATEFACTORY.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 11/25/2022 0:23:49 (local)
        End Time:   11/25/2022 9:23:49 (local)
        Renew Time: 12/1/2022 0:23:49 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

Empty due to PTT

WONKAVISION: PERSONAS & USE CASES

SOC ANALYST

NOTABLE ALERTING



PROACTIVE HUNTING

TARGETED SAMPLES
RANDOM SAMPLES

IR & REACTIVE HUNTING

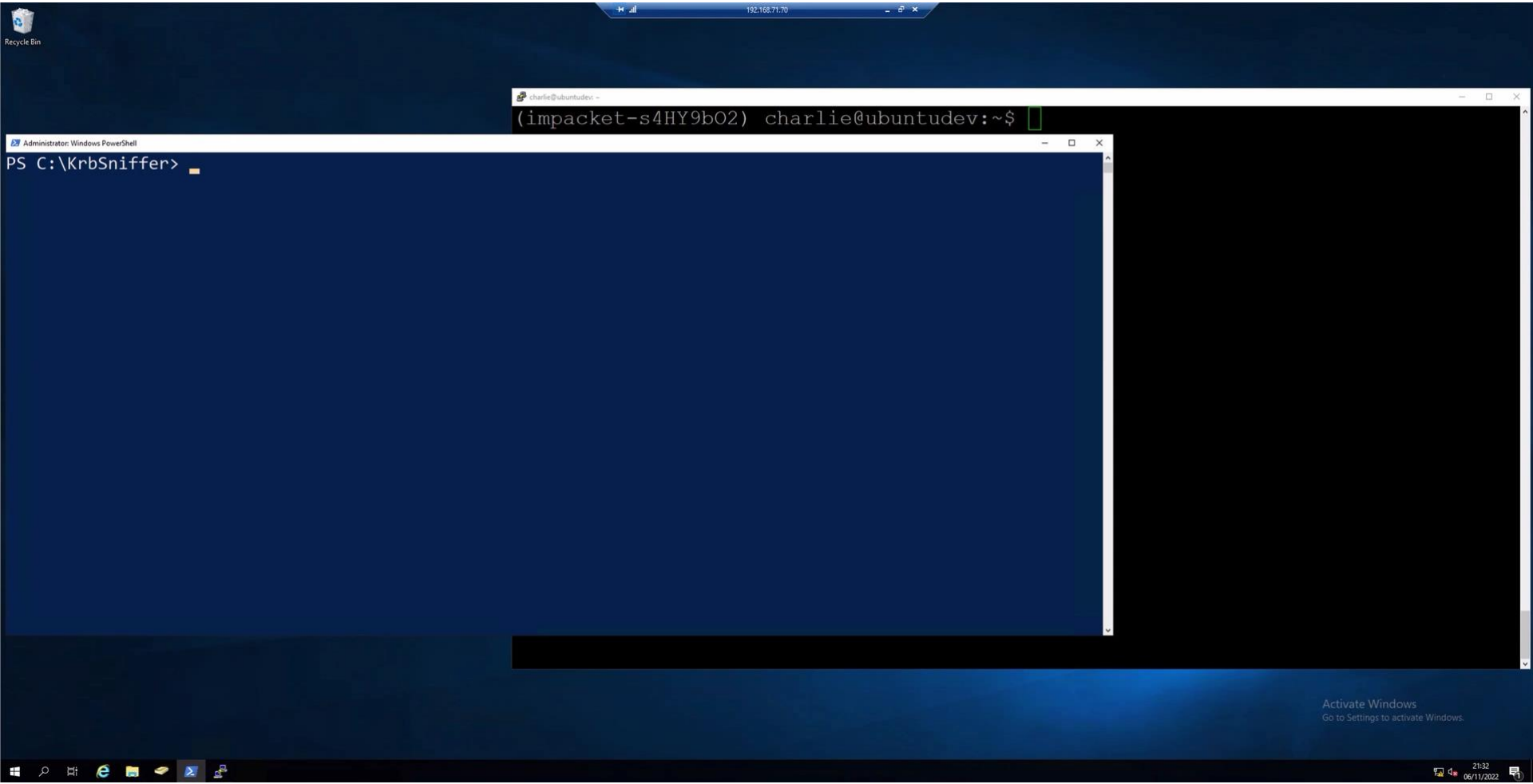
EVIDENCE OF DCSYNC

Practical example 2 - network:

- Captures traffic sent to DCs
- Fully decodes **TGS-REQ**s off the network
- Performs basic analysis of unencrypted data
- Could easily decrypt on-the-fly & do full analysis
- Could also write to the event log

DEMO

DEMO **4**: KERBEROS SNIFFER ANALYZING POSSIBLE **GT** USAGE


Recycle Bin
192.168.71.70
charlie@ubuntudev: ~
(impacket-s4HY9bO2) charlie@ubuntudev:~$
Administrator: Windows PowerShell
PS C:\KrbSniffer>
Activate Windows
Go to Settings to activate Windows.
21:32
06/11/2022

Network indicators - AS-REQ

Indicator	Genuine	Rubeus (asktgt)	Kekeo (tgt::ask)	Impacket
without pre-auth	✓	✗	✗	✓
kdc-options	40810010	40800010	40800010	50800000
etypes	-135, 3, 17, 18, 23, 24	18	17, 18, 23	18
rtime	✓	✗	✗	✓
addresses	✓	✗	✗	✗
till	2037-09-13 02:48:05 (UTC)	2037-09-13 02:48:05 (UTC)	2037-09-13 02:48:05 (UTC)	+ 1 day

Network indicators - TGS-REQ

Indicator	Genuine	Rubeus (asktgs)	Kekeo (tgs::ask)	Impacket
PA-PAC-OPTIONS	✓	✗	✗	✗
KDC-OPTIONS	40810000	40800010	40800010	40810010
ETYPES	-135, 17, 18, 23, 24	17, 18, 23, 24	17, 18, 23	3, 16, 18, 23
CNAME	✗	✓	✓	✗
ENC-AUTHORIZATION-DATA	✓	✗	✗	✗
UNCONSTRAINED **TGS-REQ**	✓	✗	✗	✗
TILL	2037-09-13 02:48:05 (UTC)	2037-09-13 02:48:05 (UTC)	2037-09-13 02:48:05 (UTC)	+ 1 DAY
AUTHENTICATOR CKSUM	✓	✗	✗	✗
AUTHENTICATOR CUSEC	✓	ALWAYS 0	ALWAYS 0	✓
AUTHENTICATOR SEQ-NUMBER	✓	✗	✗	✗

Network indicators - S4U2self

Indicator	Genuine	Rubeus	Kekeo	Impacket
PA-S4U-X509-USER	✓	✗	✗	✗
KDC-OPTIONS	40810000	40800018	40800018	40810000
ETYPES	-135, 17, 18, 23, 24	17, 18, 23, 24	17, 18, 23	18, 23
CNAME	✗	✓	✓	✗
TILL	+ 15 minutes	2037-09-13 02:48:05 (UTC)	2037-09-13 02:48:05 (UTC)	+ 1 day
PA USER NAME-TYPE	Enterprise Principal (10)	Enterprise Principal (10)	NT Principal (1)	NT Principal (1)
SNAME NAME-TYPE	NT Principal (1)	NT Principal (1)	NT Principal (1)	Unknown (0)
AUTHENTICATOR CKSUM	✓	✗	✗	✗
AUTHENTICATOR CUSEC	✓	Always 0	Always 0	✓
AUTHENTICATOR SEQ-NUMBER	✓	✗	✗	✗

Network indicators - S4U2proxy

Indicator	Genuine	Rubeus	Kekeo	Impacket
PA-PAC-OPTIONS	✓	✓	✗	✓
KDC-OPTIONS	40830000	40820010	40820010	40830000
ETYPES	-135, 17, 18, 23, 24	17, 18, 23	17, 18, 23	3, 16, 18, 23
CNAME	✗	✓	✗	✗
TILL	+ 15 minutes	2037-09-13 02:48:05 (UTC)	2037-09-13 02:48:05 (UTC)	+ 1 day
ENC-AUTHORIZATION-DATA	✓	✗	✗	✗
AUTHENTICATOR CKSUM	✓	✗	✗	✗
AUTHENTICATOR CUSEC	✓	Always 0	Always 0	✓
AUTHENTICATOR SEQ-NUMBER	✓	✗	✗	✗

Network Indicators - S4U2Proxy - Continued

- Rubeus implementation accidently orders etypes **incorrectly**
- Results in **AES128** being used for **S4U2Proxy** session key

```
128
129         // supported encryption types
130         s4u2proxyReq.req_body.etypes.Add(Interop.KERB_ETYPE.aes128_cts_hmac_sha1);
131         s4u2proxyReq.req_body.etypes.Add(Interop.KERB_ETYPE.aes256_cts_hmac_sha1);
132         s4u2proxyReq.req_body.etypes.Add(Interop.KERB_ETYPE.rc4_hmac);
133
```

AES128 is top (favored)

```
[+] Ticket successfully imported!

C:\Rubeus>klist

Current LogonId is 0:0xc415a

Cached Tickets: (1)

#0>     Client: oloompa @ CHOCOLATEFACTORY.LOCAL
        Server: LDAP/DC1.chocolatefactory.local @ CHOCOLATEFACTORY.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
        Start Time: 11/25/2022 1:18:38 (local)
        End Time:   11/25/2022 10:08:38 (local)
        Renew Time: 12/1/2022 1:18:38 (local)
        Session Key Type: AES-128-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:
```

Resulting session key is **AES128**

Network indicators - TGS-REQ, S4U2self & S4U2proxy

- All 3 contain a **TGT** within the request
- Most forged **TGT**s have an enc_part that is much smaller than expected
- Below **920** bytes seems to work well
- Doesn't require ticket decryption

LSA vs. Network

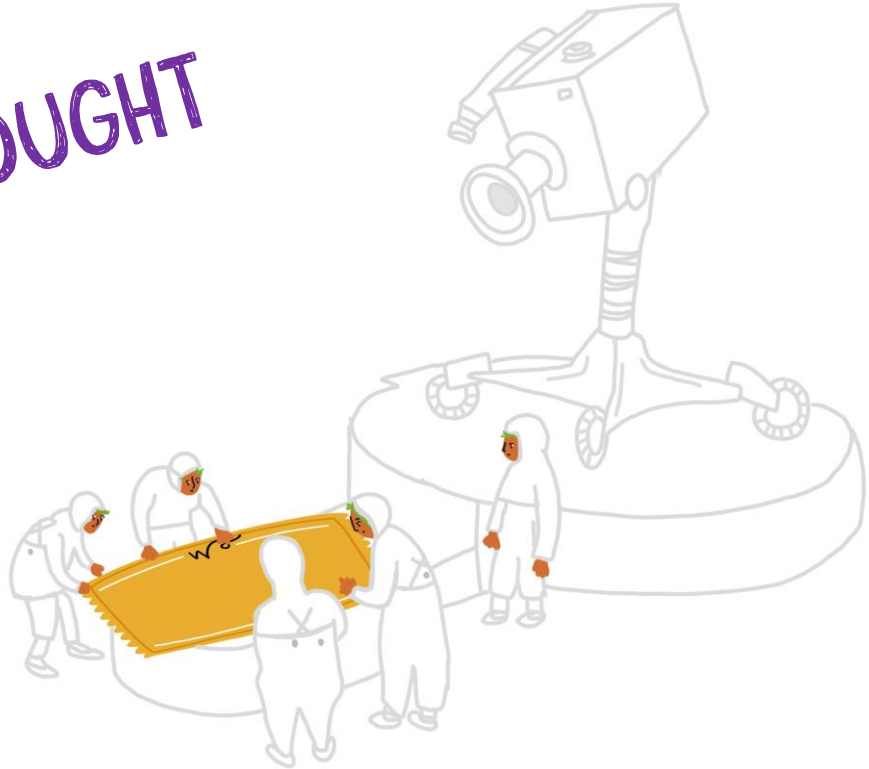
LSA

- Returns more unencrypted info (e.g. ticket flags & ticket times)
- Shows session info
- Sees service tickets & TGTs w/o passing many different protocols

Network

- Can use info about the TGS-REQ vs. just the ticket
- TGS-REQs can all be viewed looking at only DC network traffic
- Will see TGS-REQs when LSASS isn't being used (Impacket, over SOCKS proxy)

Final Thought



There's no guarantee you will detect a forged ticket, but decrypting on-the-fly increases your chance of success

QUESTIONS?