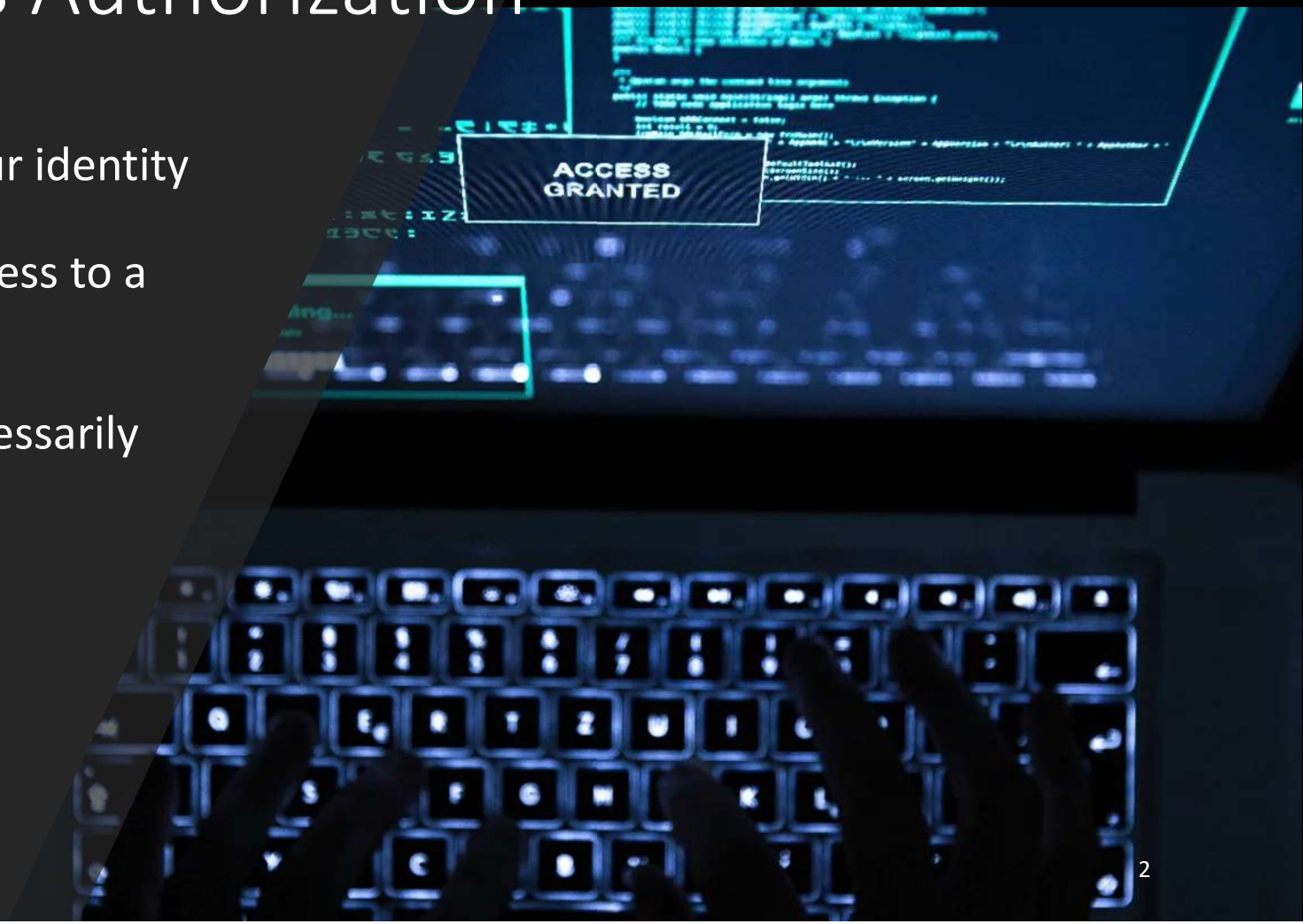
A Network Defender's Guide to Credential Attacks

Authentication vs Authorization

- Authentication - Proving your identity
- Authorization - Granting access to a resource
- These functions are not necessarily done by the same system



Authentication

- Prove your identity to an authentication authority
  - User name and password
  - MFA
  - Whatever
- Receive digital proof of identify
  - Token
  - Ticket
  - Cookie
  - Whatever

Authorization

Present your digital proof of identity to a system

System consults the access control list

Grant access to whatever your account is authorized to access

Real World Parallel

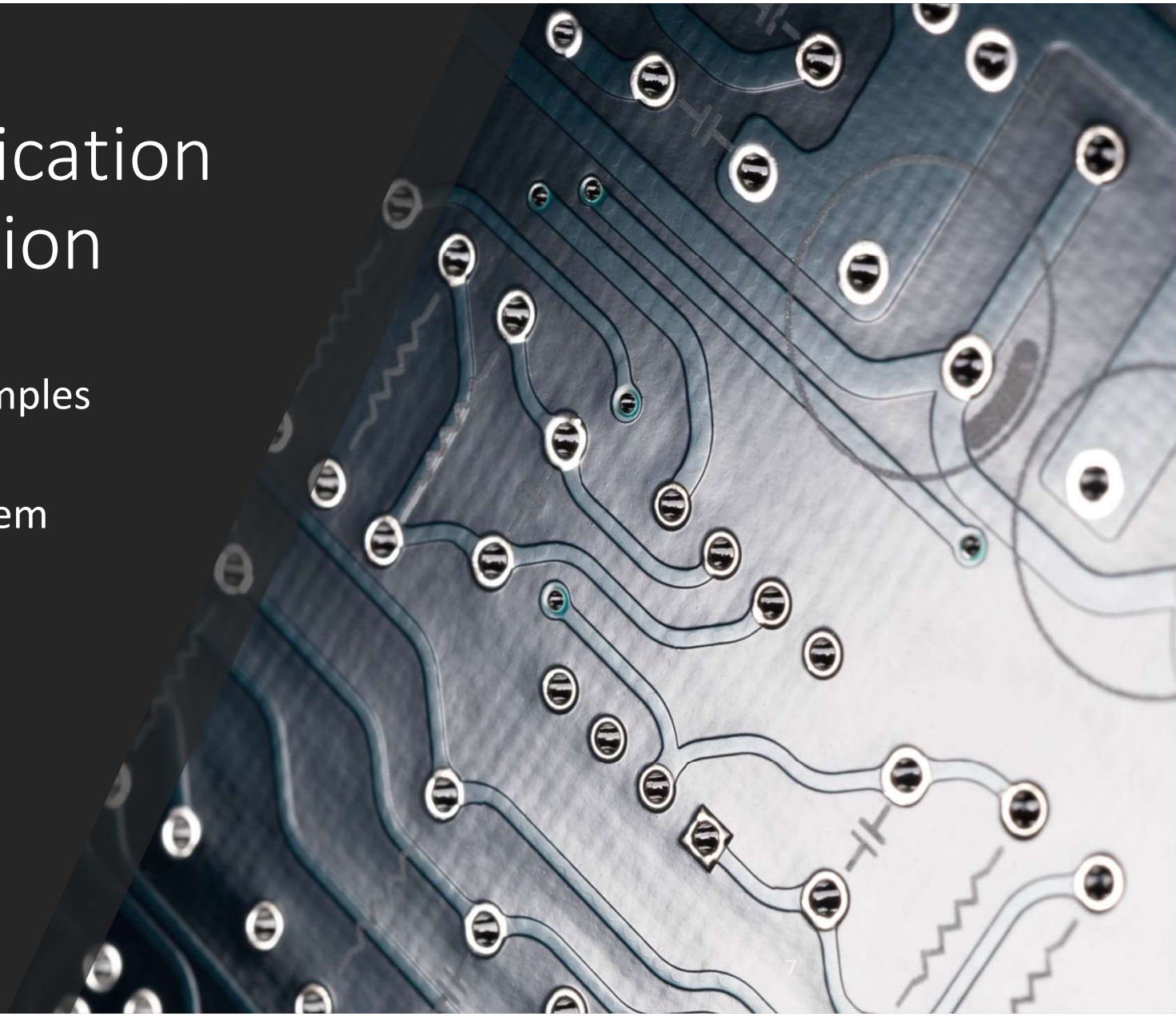
- Prove your identity to a government, get a driver's license (Authentication)
- Show your driver's and your identity is trusted (Authorization)

Real vs Cyber Difference

- Real world
  - Your driver's license has a photo
  - You might need to match your written signature
  - You are often asked for multiple forms of ID
- Cyber
  - Anyone who presents your digital proof of identity…is you
  - So…credential theft attacks are easy to do

Cyber Authentication and Authorization

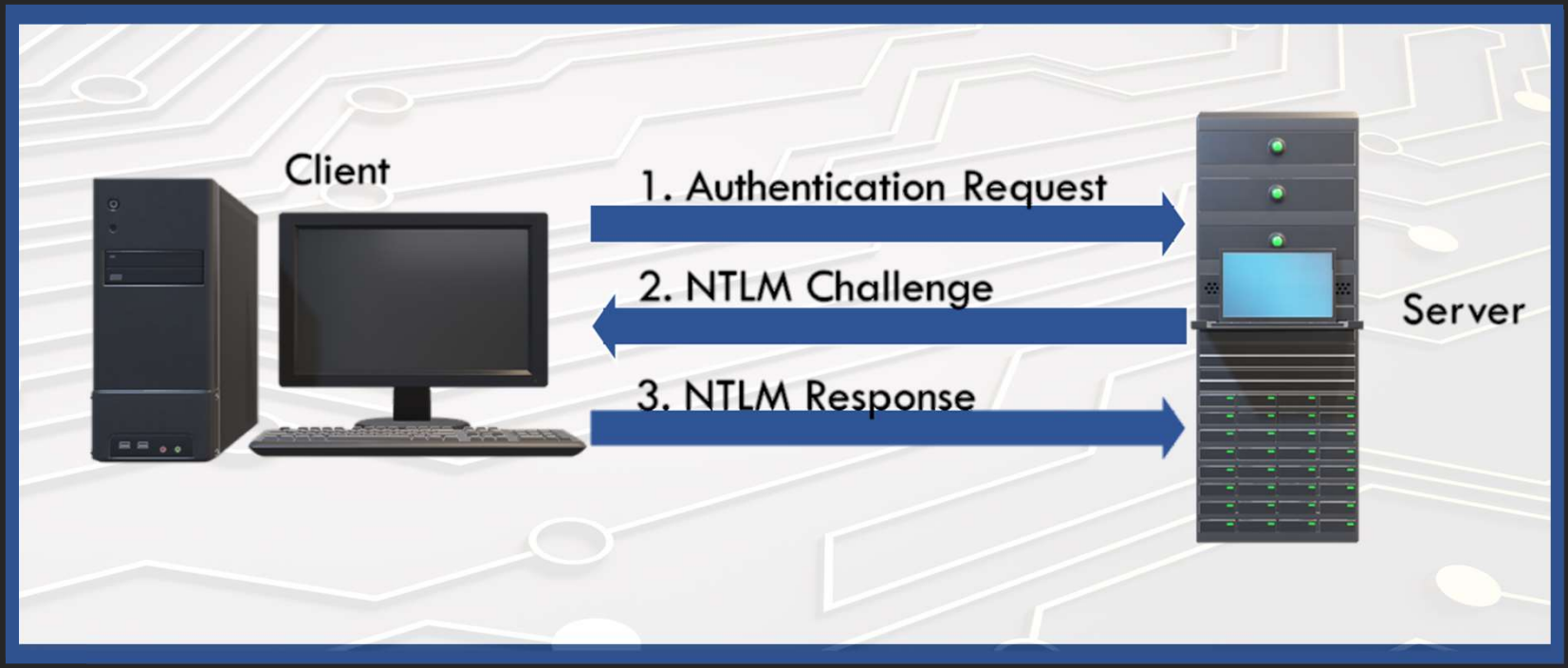
- Let’s look at specific examples
- Starting with most on-prem environments



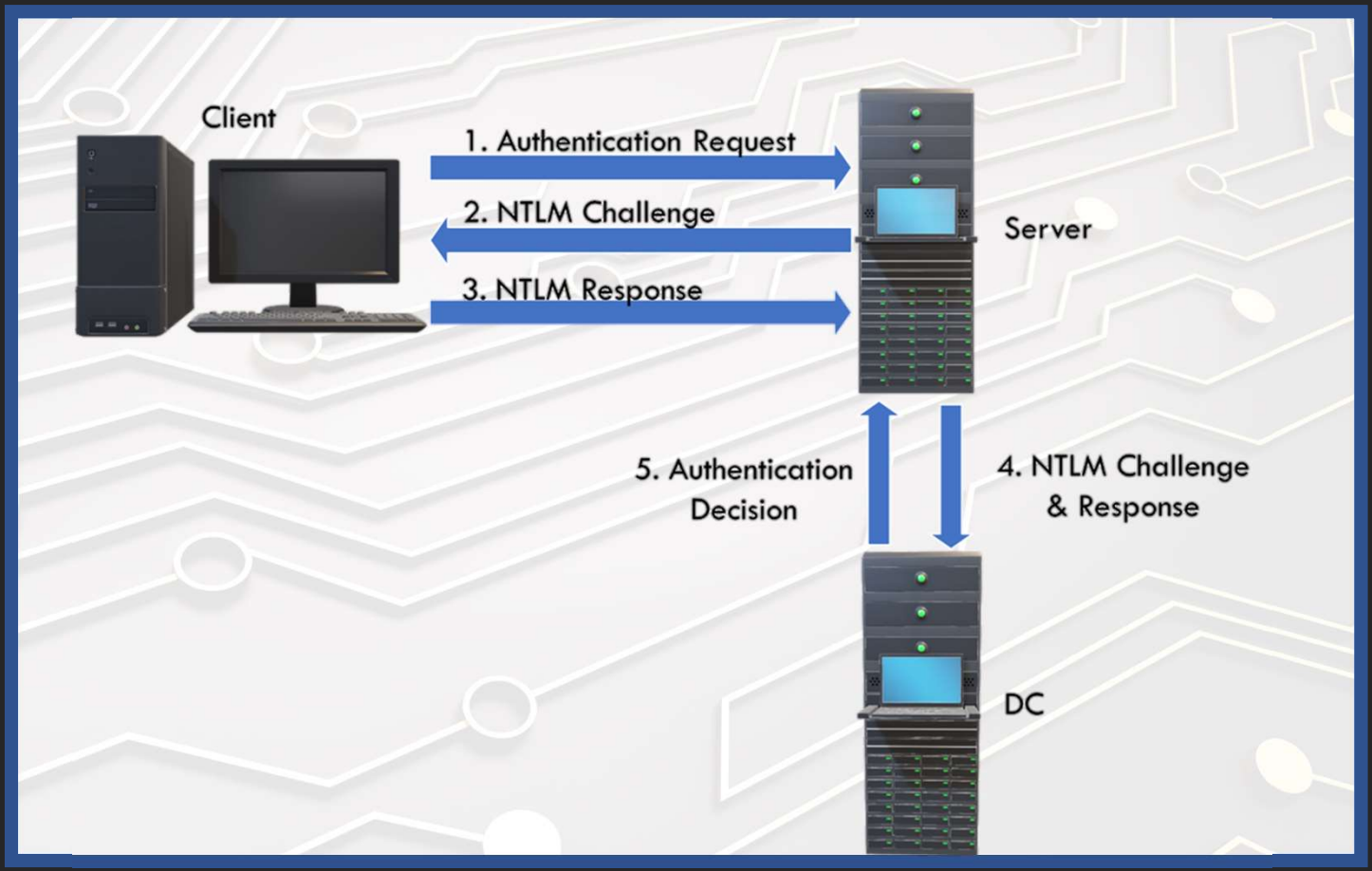
Password-based Authentication

- Username and password/passphrase are used to prove identity
- Passwords are not stored in plain text, but as a hashed representation of the password
- Local accounts stored in Security and Accounts Manager registry hive
- Domain accounts stored in NTDS.dit on Domain Controller
- Stored in Local Security Authority Subsystem Service (LSASS) process memory during interactive logon

NTLMv2 Challenge Response Protocol (Local account)

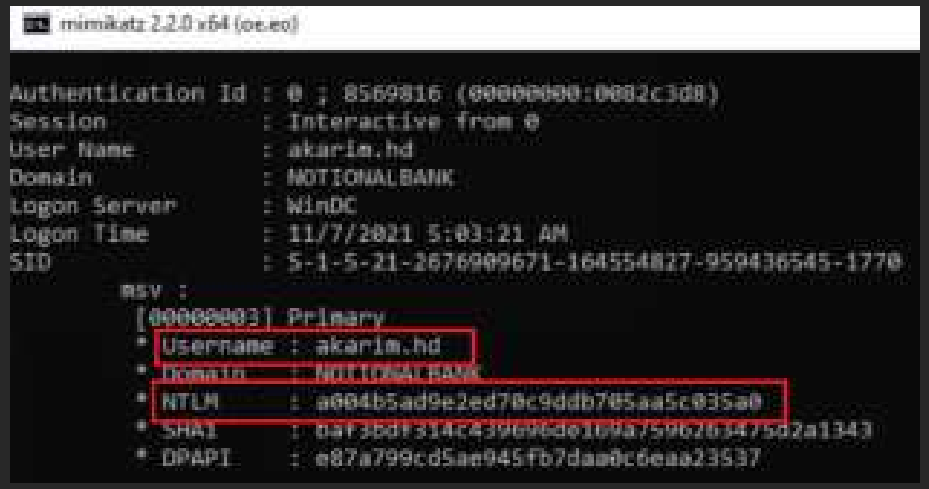


NTLMv2 Challenge Response Protocol (domain account)



Mimikatz

- With local admin permissions, attackers can steal hashes from RAM
- Other tools (pwdump, gsecdump, etc.) steal the hashes stored on disk in a similar manner



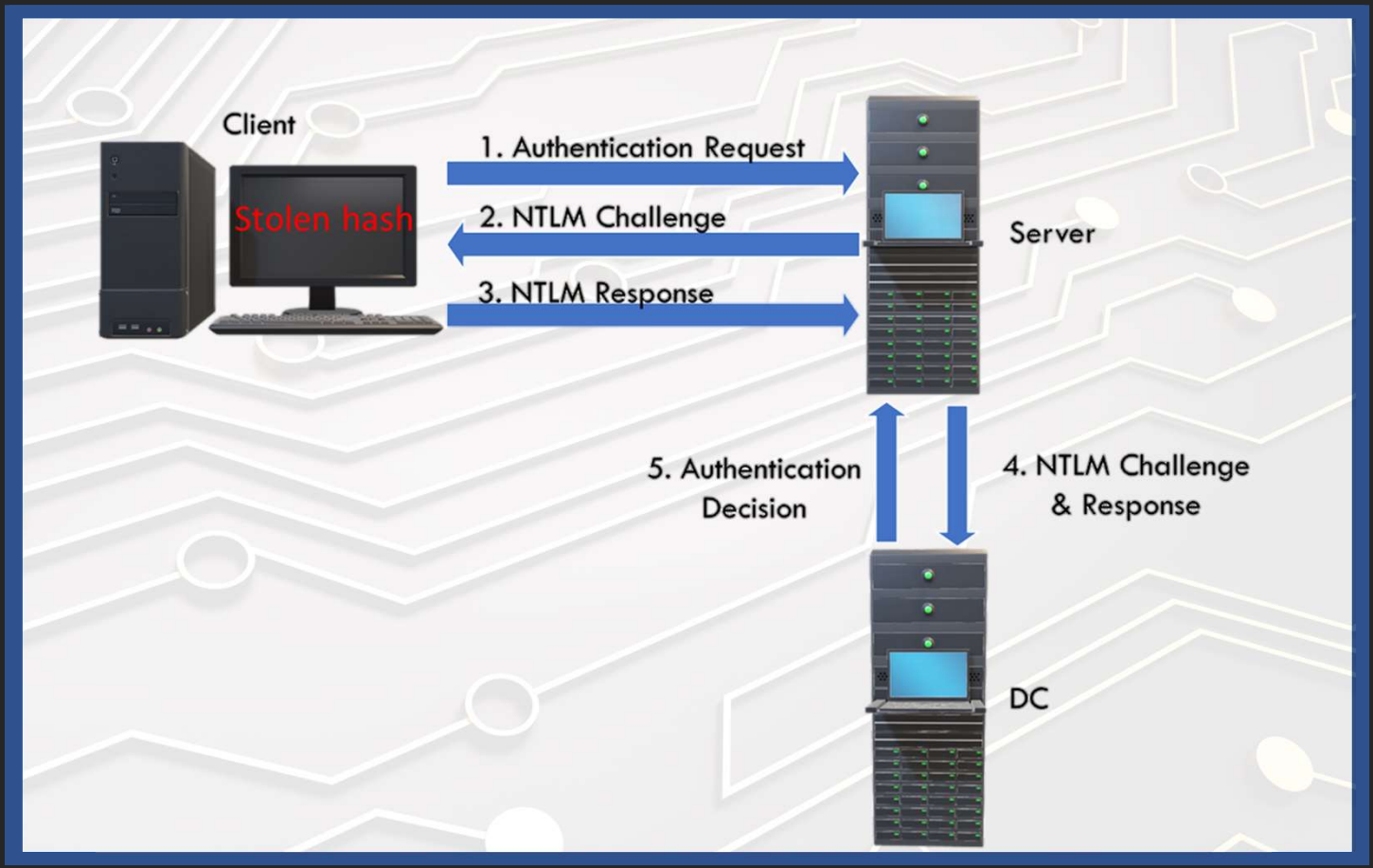
Mimikatz

- Once the attacker has the hash, a pass-the-hash (PTH) attack is performed with Mimikatz

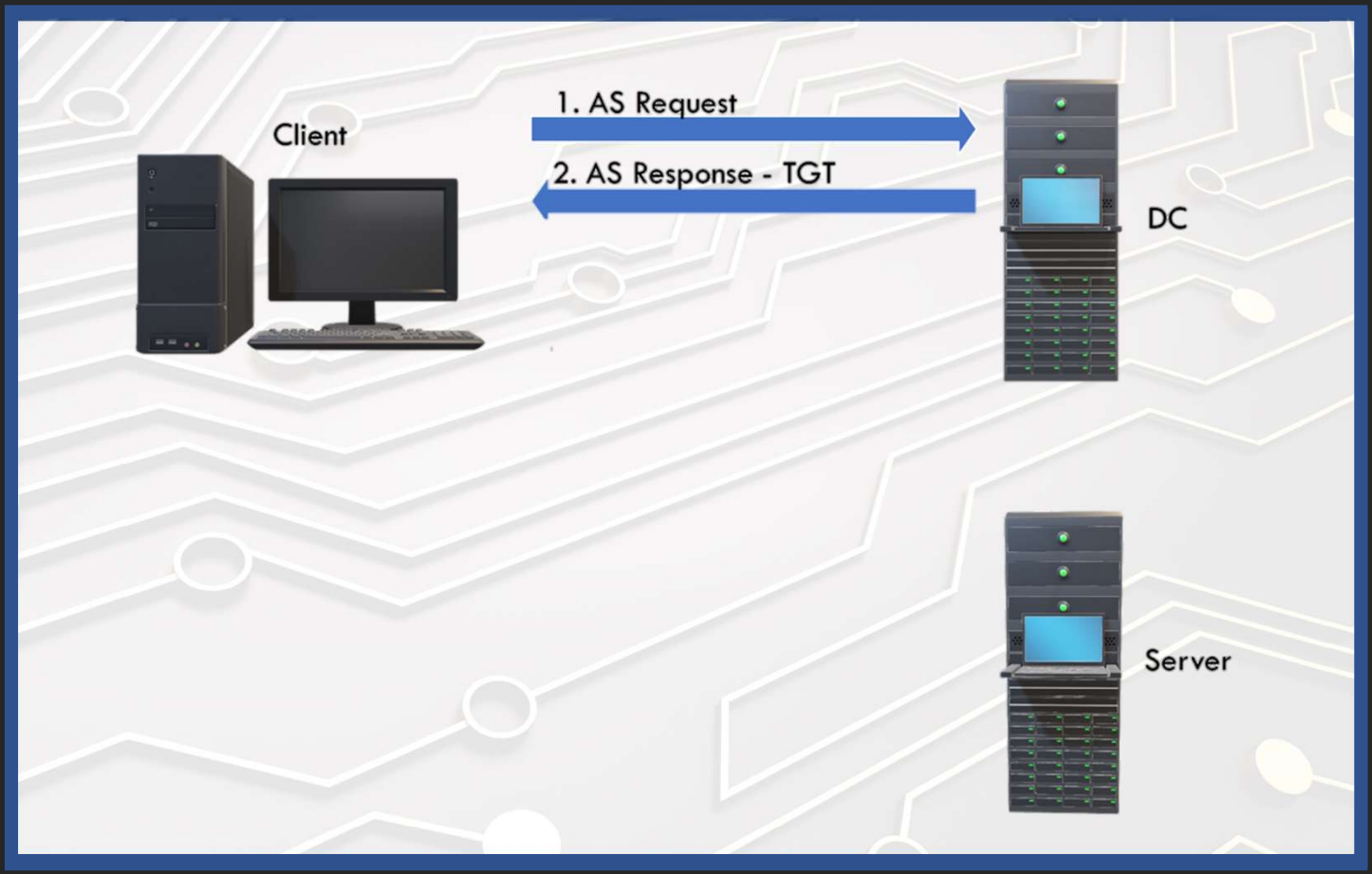
```
mimikatz 2.2.0 x64 (oe.eo)

mimikatz # sekurlsa::pth /user:akarim.hd /domain:notionalbank.com /ntlm:a004b5ad9e2ed70c9ddb705aa5c035a0
user    : akarim.hd
domain  : notionalbank.com
program : cmd.exe
impers. : no
NTLM    : a004b5ad9e2ed70c9ddb705aa5c035a0
  |  PID  1664
  |  TID  8616
  |  LSA Process was already R/W
  |  LUID 0 ; 8940369 (00000000:00886b51)
  \_ msv1_0   - data copy @ 000001788D446780 : OK !
  \_ kerberos - data copy @ 000001788D54A808
   \_ des_cbc_md4       -> null
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ des_cbc_md4       OK
   \_ *Password replace @ 000001788D6047A8 (32) -> null
```

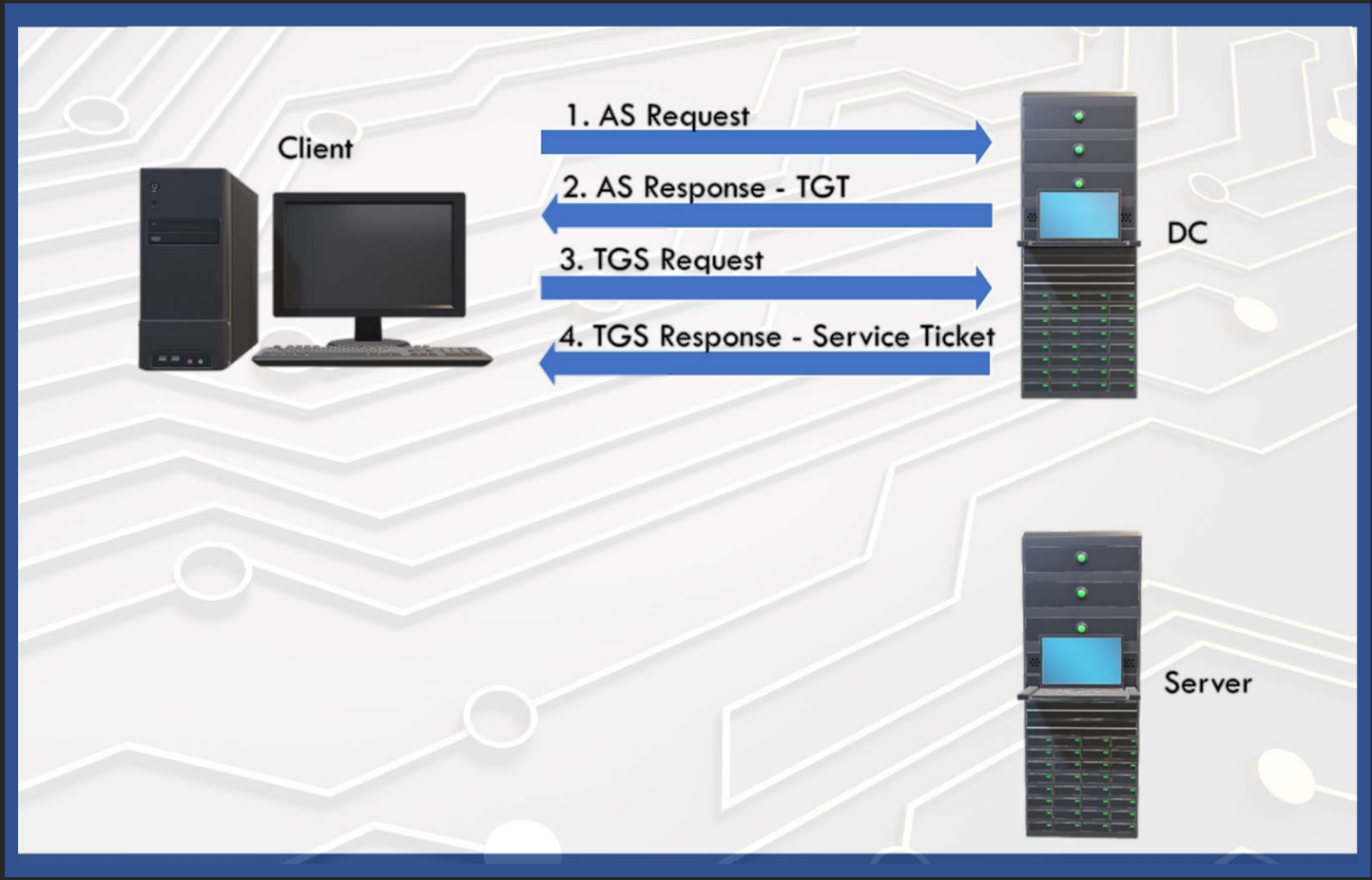
Pass-the-Hash



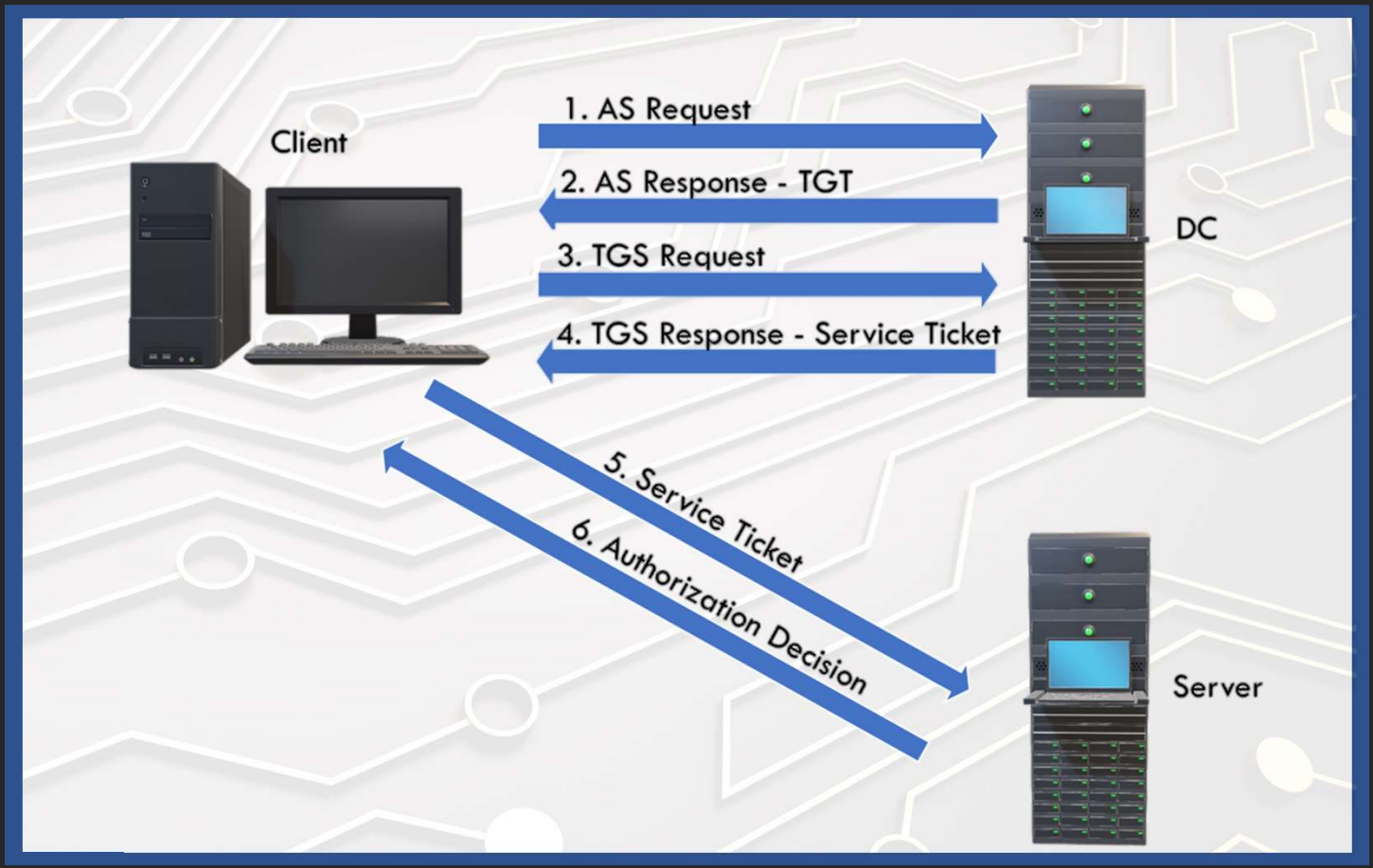
Kerberos: Authenticate to the DC



Kerberos: Request Service Ticket



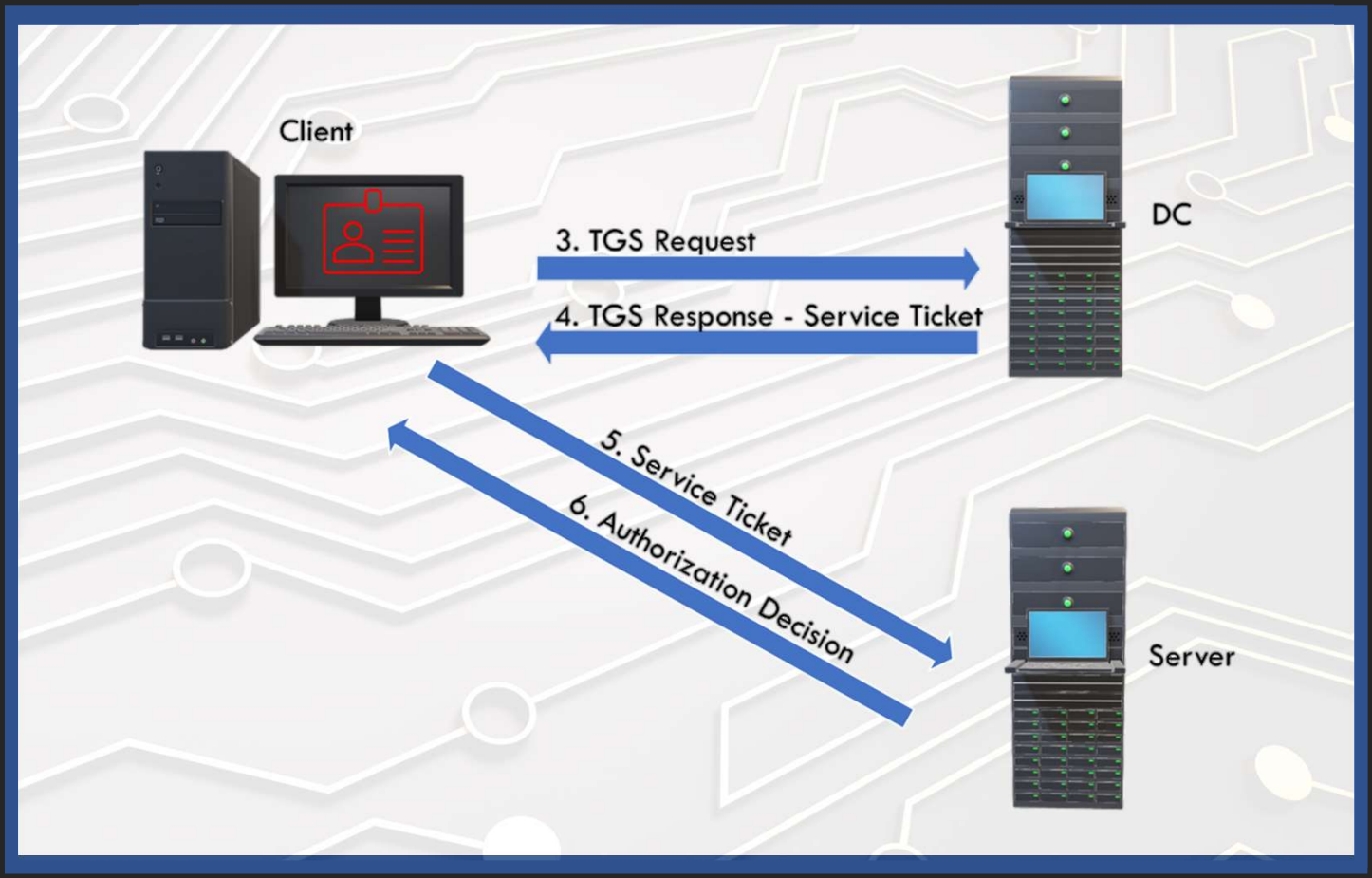
Kerberos: Authorization to Resource



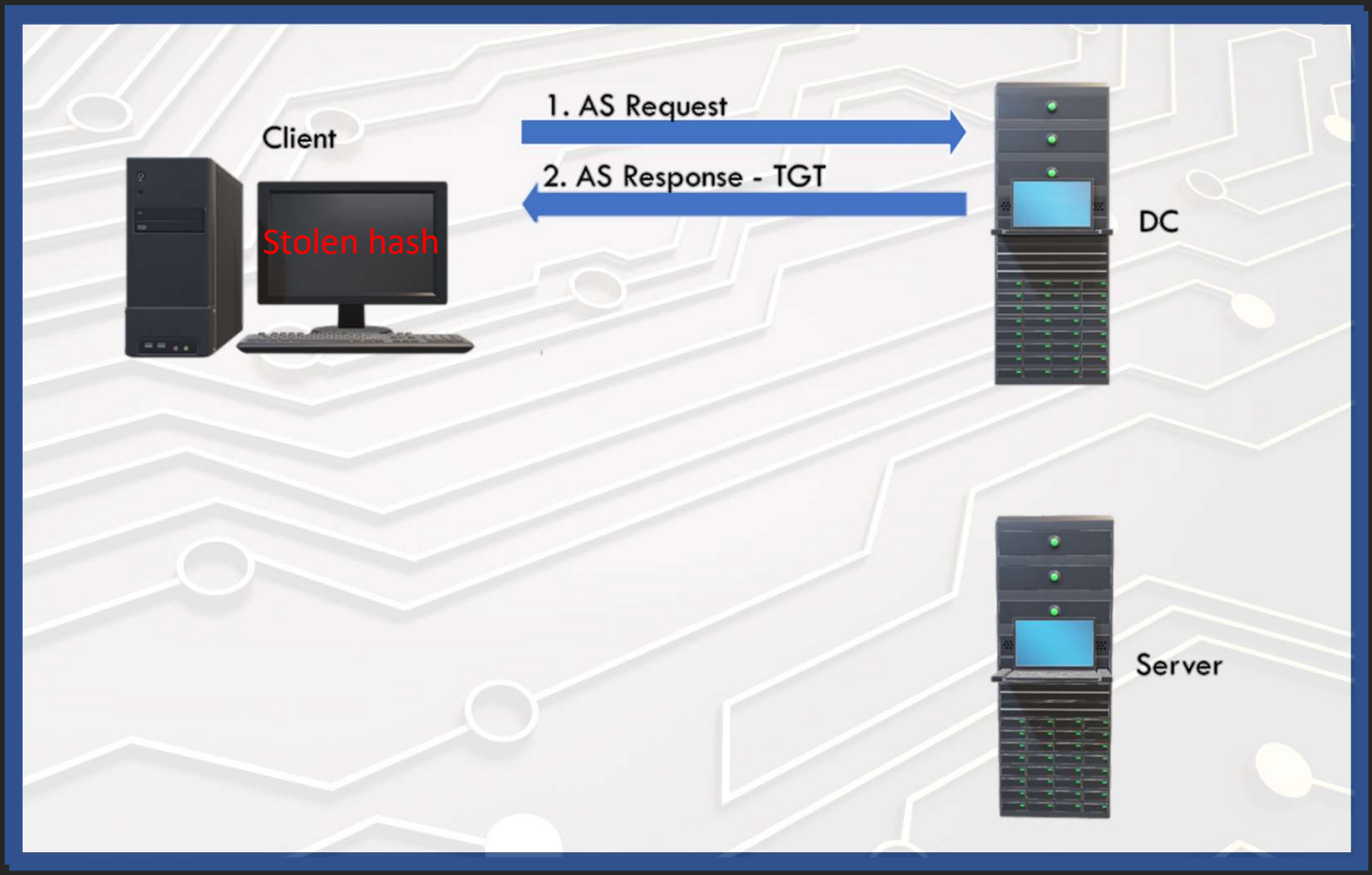
Stealing Ticket Granting Tickets (TGT)

- Just as hashes are stored in memory during an interactive logon, so are Kerberos tickets
- Mimikatz can steal the TGT from RAM
- Then do a pass-the-ticket (PTT) attack

Pass-the-Ticket



Overpass-the-hash



But wait, there's more!

- Kerberoasting
- Golden Tickets
- Silver Tickets
- Process Access Tokens
- Cached Domain Credentials
- Skeleton Keys
- DCSync Attacks

Cloud to the rescue

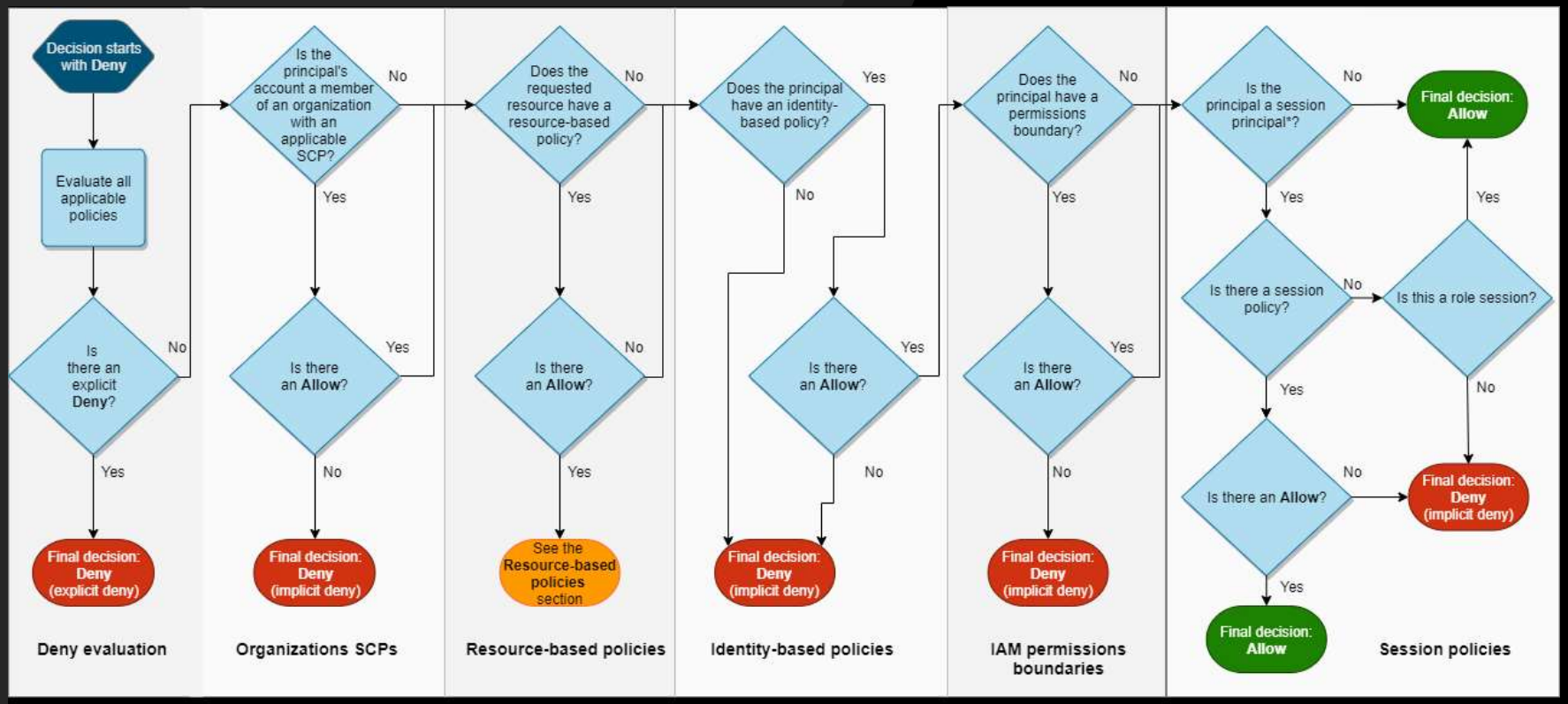
Or not…

Cloud Permissions are Complex

- Permissions can be set on accounts, roles, or resources
- Roles can be assumed by accounts or resources
- Sessions can have separate restrictions
- Conditional access can be used
- Policies need to be structured carefully and ensure least privilege
- Determining the end result can be a challenge

Cloud Permissions are Complex

AWS Policy Evaluation Flow Chart



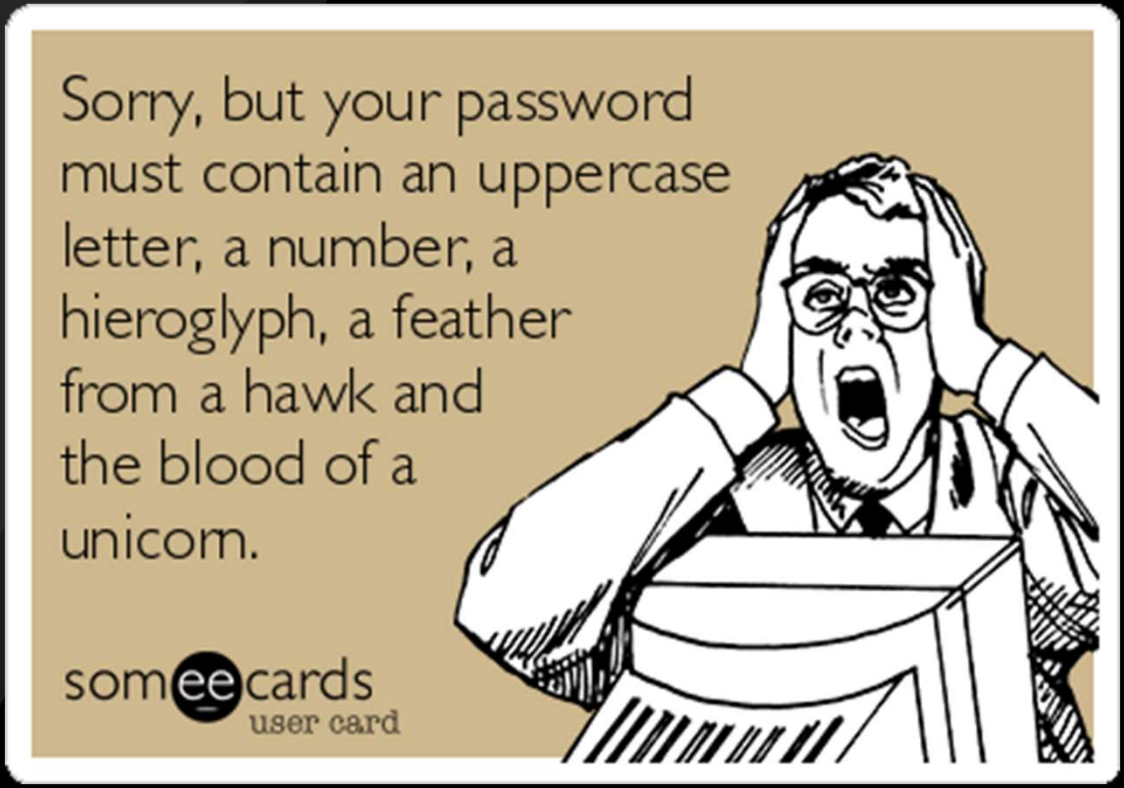
https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html

Access Keys

- Remember…your cloud services are accessed through web APIs
- Even if you use the CLI
- So you need a digital proof of identity that can be sent with each request
- This is accomplished through access keys

Access Keys

- You know how always tell people never to write down their username and password?



Cloud Account Access Keys

- Access keys used for programmatic access or Command-line interface
- These get stored on disk in plain text:
  - ~/.azure
  - ~/.aws
  - ~/.config
- Attacker can compromise on-prem system and use to pivot to cloud

```
[Profile_Name]
aws_access_key_id = AKIAI44QH8DHBEXAMPLE
aws_secret_access_key = je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

Cloud Computing Instance Credentials

- Cloud virtual machines need to interact with other services and resources
- Cloud VMs can be dynamically created as needed to meet load
- Cloud VMs are often assigned their own credential keys to access those resources

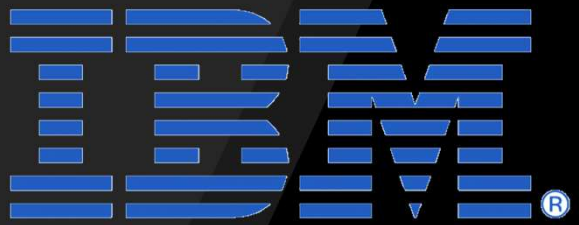
Cloud Instance MetaData Service

- Apps running on the virtual machine can access its credential over a local HTTP server on IP address 169.254.169.254
- But the access keys can also be retrieved by an attacker with access to the system
- Many types of compromise can expose these credentials (SSRF, Command injection, remote code execution, etc.)
- Once the credentials are stolen, they can be used over CLI or API access to other resources.

Identity as a Service (IDaaS)

- Identity and Access Management (IAM) centralizes your identity management
- Single Sign On (SSO) can even be realized
- The identity provider verifies each user’s identity
- Access can then be granted by any resource that trusts the identity provider

Example identity providers



It’s still the same basic concept

- Prove Identity – Can include many factors, so that’s an improvement
- Receive digital proof of identity
  - Primary Refresh Token
  - SAML Token
  - OAuth 2.0 Bearer Token
  - OpenID JSON Web Token
- Present digital proof of identity to access resources

But the fundamental problem remains

- Prove Identity
- Receive digital proof of identity
- Present digital proof of identity to access resources

But the fundamental problem remains

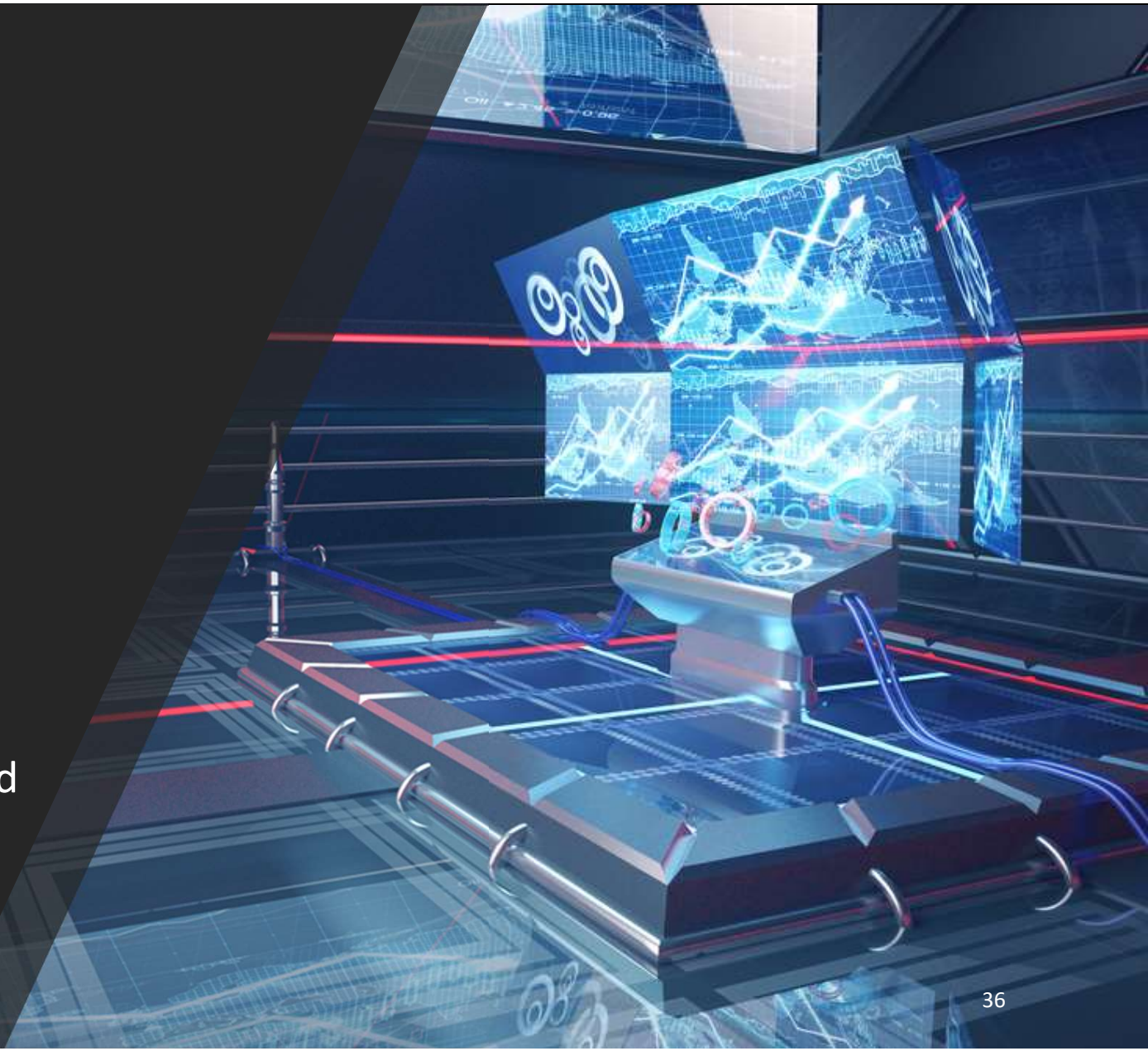
- Prove Identity
- Receive digital proof of identity
- Steal victim's proof of identity
- Present victim's digital proof of identity to access resources

Defending your credentials

- SAW/PAW
- Segmentation
- Application Control
- Windows Credential Guard
- Windows Defender Application Guard

Detecting Abuse

- UEBA
- Threat Hunting
- Alert on privileged logon to new system
- Alert on any use of instance credentials outside of expected behavior



Conclusion

- Newer doesn't always mean better
- The traditional perimeter is gone
- More attack surface
- Active cyber defense is needed