How Attackers Are Compromising Your Networks and What You Can Do About It

ROOTCON18

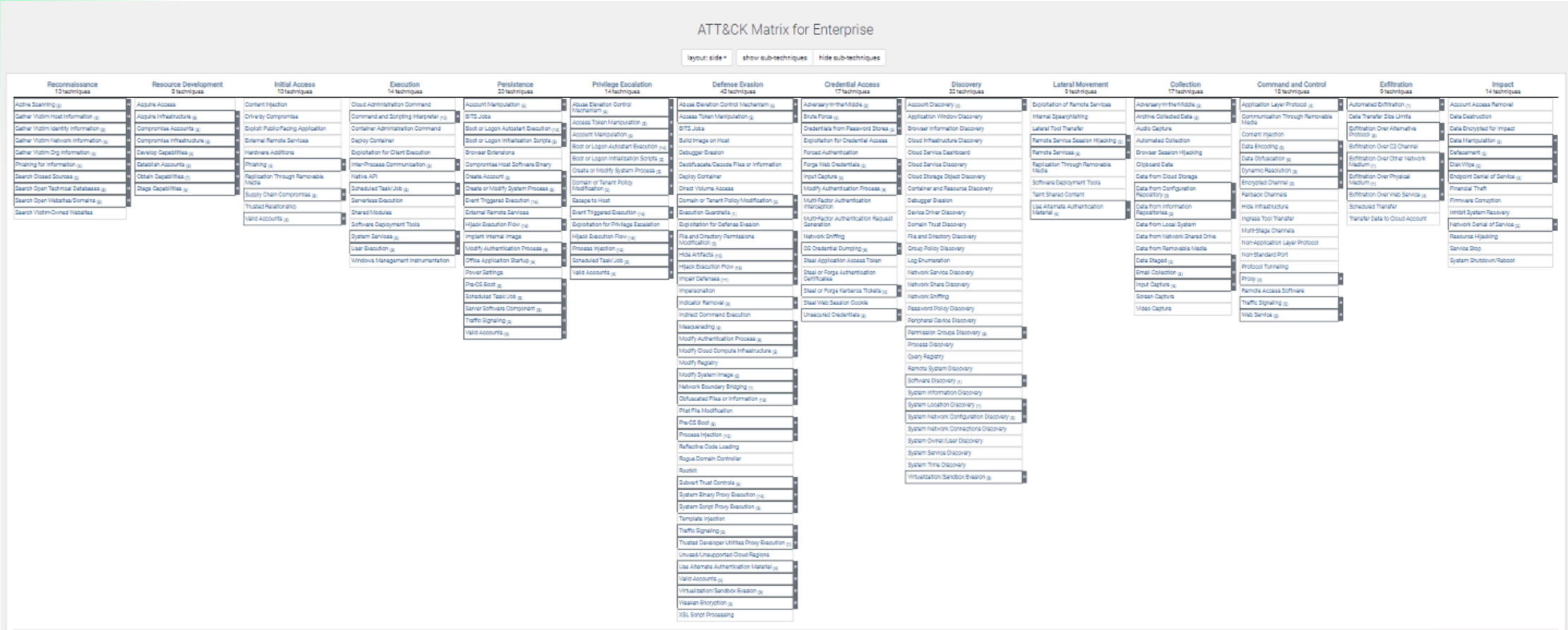
About Me

- Offsec Manager @ Red Rock IT Security Inc.
- SANS Instructor (SEC560)
- MSISE Student @ SANS Technology Institute
- Master in Information Security @ DLSU Manila
- GSP, GX-PT, GX-IH, GMOB, GPEN, GCIH, CISSP, OSEP, OSCP etc.

OUTLINE

- What are attackers doing today?
- Windows and Active Directory Primer
- Active Directory Attacks
- Active Directory Defenses


1
RECONNAISSANCE
Harvesting email addresses, conference information, etc.
2
WEAPONIZATION
Coupling exploit with backdoor into deliverable payload
3
DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.
4
EXPLOITATION
Exploiting a vulnerability to execute code on victim's system
5
INSTALL
INSTALLATION
Installing malware on the asset
6
COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim
7
ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals


ATT&CK Matrix for Enterprise
layout: side
show sub-techniques
hide sub-techniques
Reconnaissance
10 techniques
Active Scanning (3)
Gather Victim Host Information (4)
Gather Victim Identity Information (3)
Gather Victim Network Information (6)
Gather Victim Org Information (4)
Phishing for Information (3)
Search Closed Sources (2)
Search Open Technical Databases (5)
Search Open Websites/Domains (2)
Search Victim-Owned Websites
Resource Development
8 techniques
Acquire Access
Acquire Infrastructure (8)
Compromise Accounts (3)
Compromise Infrastructure (8)
Develop Capabilities (4)
Establish Accounts (3)
Obtain Capabilities (7)
Stage Capabilities (6)
Initial Access
10 techniques
Content Injection
Drive-by Compromise
Exploit Public-Facing Application
External Remote Services
Hardware Additions
Phishing (4)
Replication Through Removable Media
Supply Chain Compromise (3)
Trusted Relationship
Valid Accounts (4)
Execution
14 techniques
Cloud Administration Command
Command and Scripting Interpreter (12)
Container Administration Command
Deploy Container
Exploitation for Client Execution
Inter-Process Communication (3)
Native API
Scheduled Task/Job (5)
Serverless Execution
Shared Modules
Software Deployment Tools
System Services (2)
User Execution (3)
Windows Management Instrumentation
Persistence
20 techniques
Account Manipulation (5)
BITS Jobs
Boot or Logon Autostart Execution (14)
Boot or Logon Initialization Scripts (5)
Browser Extensions
Compromise Host Software Binary
Create Account (3)
Create or Modify System Process (4)
Event Triggered Execution (16)
External Remote Services
Hijack Execution Flow (12)
Implant Internal Image
Modify Authentication Process (8)
Office Application Startup (6)
Power Settings
Pre-OS Boot (5)
Scheduled Task/Job (5)
Server Software Component (5)
Traffic Signaling (2)
Valid Accounts (4)
Privilege Escalation
14 techniques
Abuse Elevation Control Mechanism (6)
Access Token Manipulation (5)
Account Manipulation (5)
Boot or Logon Autostart Execution (14)
Boot or Logon Initialization Scripts (5)
Create or Modify System Process (4)
Domain or Tenant Policy Modification (2)
Escape to Host
Event Triggered Execution (16)
Exploitation for Privilege Escalation
Hijack Execution Flow (12)
Process Injection (12)
Scheduled Task/Job (5)
Valid Accounts (4)
Defense Evasion
42 techniques
Abuse Elevation Control Mechanism (6)
Access Token Manipulation (5)
BITS Jobs
Build Image on Host
Debugger Evasion
Deobfuscate/Decode Files or Information
Deploy Container
Direct Volume Access
Domain or Tenant Policy Modification (2)
Execution Guardrails (1)
Exploitation for Defense Evasion
File and Directory Permissions Modification (2)
Hide Artifacts (10)
Hijack Execution Flow (12)
Impair Defenses (10)
Impersonation
Indicator Removal (9)
Indirect Command Execution
Masquerading (8)
Modify Authentication Process (8)
Modify Cloud Compute Infrastructure (6)
Modify Registry
Modify System Image (2)
Network Boundary Bridging (1)
Obfuscated Files or Information (12)
Pre-OS Boot (5)
Process Injection (12)
Reflective Code Loading
Rogue Domain Controller
Rootkit
Subvert Trust Controls (6)
System Binary Proxy Execution (14)
System Script Proxy Execution (4)
Template Injection
Traffic Signaling (2)
Trusted Developer Utilities Proxy Execution (1)
Unused/Unsupported Cloud Regions
Use Alternate Authentication Material (4)
Valid Accounts (4)
Virtualization/Sandbox Evasion (3)
Weaken Encryption (2)
XSL Script Processing
Credential Access
17 techniques
Adversary-in-the-Middle (3)
Brute Force (4)
Credentials from Password Stores (6)
Exploitation for Credential Access
Forced Authentication
Forge Web Credentials (2)
Input Capture (4)
Modify Authentication Process (9)
Multi-Factor Authentication Interception
Multi-Factor Authentication Request Generation
Network Sniffing
OS Credential Dumping (8)
Steal Application Access Token
Steal or Forge Authentication Certificates
Steal or Forge Kerberos Tickets (4)
Steal Web Session Cookie
Unsecured Credentials (8)
Discovery
32 techniques
Account Discovery (4)
Application Window Discovery
Browser Information Discovery
Cloud Infrastructure Discovery
Cloud Service Dashboard
Cloud Service Discovery
Cloud Storage Object Discovery
Container and Resource Discovery
Debugger Evasion
Device Driver Discovery
Domain Trust Discovery
File and Directory Discovery
Group Policy Discovery
Log Enumeration
Network Service Discovery
Network Share Discovery
Network Sniffing
Password Policy Discovery
Peripheral Device Discovery
Permission Groups Discovery (3)
Process Discovery
Query Registry
Remote System Discovery
Software Discovery (1)
System Information Discovery
System Location Discovery (1)
System Network Configuration Discovery (1)
System Network Connections Discovery
System Owner/User Discovery
System Service Discovery
System Time Discovery
Virtualization/Sandbox Evasion (3)
Lateral Movement
9 techniques
Exploitation of Remote Services
Internal Spearphishing
Lateral Tool Transfer
Remote Service Session Hijacking (2)
Remote Services (6)
Replication Through Removable Media
Software Deployment Tools
Taint Shared Content
Use Alternate Authentication Material (4)
Collection
17 techniques
Adversary-in-the-Middle (3)
Archive Collected Data (3)
Audio Capture
Automated Collection
Browser Session Hijacking
Clipboard Data
Data from Cloud Storage
Data from Configuration Repository (2)
Data from Information Repositories (3)
Data from Local System
Data from Network Shared Drive
Data from Removable Media
Data Staged (2)
Email Collection (3)
Input Capture (4)
Screen Capture
Video Capture
Command and Control
16 techniques
Application Layer Protocol (4)
Communication Through Removable Media
Content Injection
Data Encoding (2)
Data Obfuscation (3)
Dynamic Resolution (3)
Encrypted Channel (2)
Fallback Channels
Hide Infrastructure
Ingress Tool Transfer
Multi-Stage Channels
Non-Application Layer Protocol
Non-Standard Port
Protocol Tunneling
Proxy (4)
Remote Access Software
Traffic Signaling (2)
Web Service (3)
Exfiltration
9 techniques
Automated Exfiltration (1)
Data Transfer Size Limits
Exfiltration Over Alternative Protocol (3)
Exfiltration Over C2 Channel
Exfiltration Over Other Network Medium (1)
Exfiltration Over Physical Medium (1)
Exfiltration Over Web Service (4)
Scheduled Transfer
Transfer Data to Cloud Account
Impact
14 techniques
Account Access Removal
Data Destruction
Data Encrypted for Impact
Data Manipulation (3)
Defacement (2)
Disk Wipe (2)
Endpoint Denial of Service (4)
Financial Theft
Firmware Corruption
Inhibit System Recovery
Network Denial of Service (2)
Resource Hijacking
Service Stop
System Shutdown/Reboot

Attacker Phases

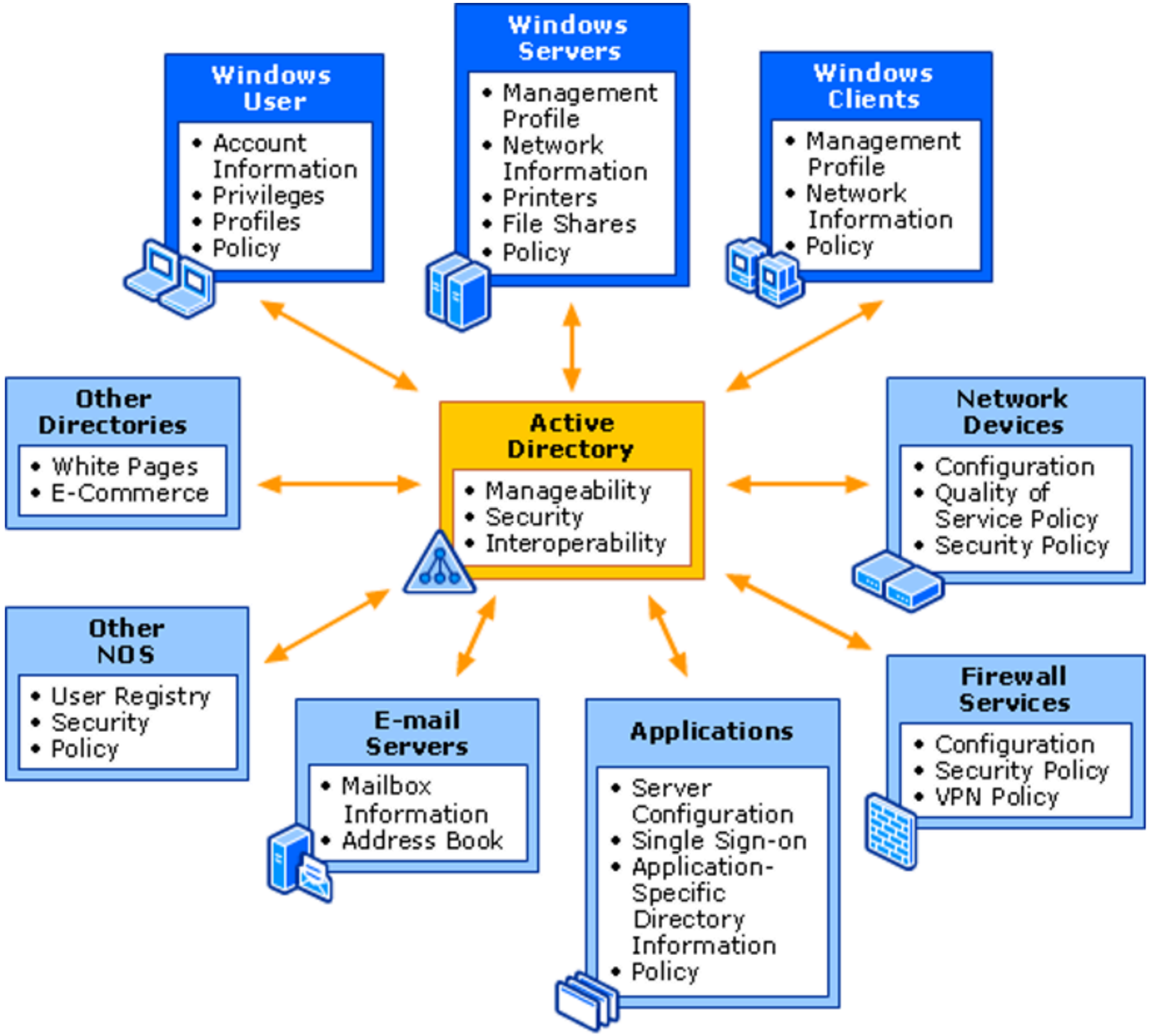
1. **Initial Access**
2. **Escalate Privileges**
3. **Actions on Objectives**

How Do Attackers Gain Access?

1. **Phishing**
2. **Login from a publicly exposed service**
3. **Exploitable publicly facing services**
4. **Internal rouge devices**
5. **Supply Chain Attacks**

Active Directory Directory Services

- **Microsoft's proprietary directory service to manage Windows networks**
- **A database that stores information of users, groups, computers, group policies, etc. stored in a domain controller**
  - A server that helps manage the whole AD database.
    - Usernames and passwords are stored here.
- **Changes to the AD database are replicated to other domain controllers inside the domain**



https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc780036(v=ws.10)

Active Directory Attacks: The Past 10 Years

1997: Pass-the-hash originally published by Paul Ashton

2001: Sir Dystic of Cult of the Dead Cow (cDc) releases SMBRelay and SMBRelay2

2007: NBNSpoof tool created by Robert Wesley McGrew (LLMNR/NBT-NS)

2008: Hernan Ochoa publishes the Pass-the-Hash toolkit (later became WCE)

2010: Windows Credentials Editor (WCE) by Hernan Ochoa

* Credits to Sean Metcalf's talk "A Decade of Active Directory Attacks: What We've Learned & What's Next" at TROOPERS24

Active Directory Attacks: The Past 10 Years

2011: First version of Mimikatz tool released by Benjamin Delpy

2012: Exploiting Windows Group Policy Preferences (GPP), Responder v1 tool

2013: Invoke-Mimikatz module released by Joe Bialek

2014:

- Abusing Microsoft Kerberos: Sorry You Guys Don't Get It" by Benjamin Delpy (Golden Tickets, Overpass-the-hash, Pass-the-ticket)
- PAC Validation, The 20 Minute Rule and Exceptions blog post about Silver Tickets by Skip Duckwell
- September: Kerberoast released by Tim Medin
- December: PowerView tool released by Will Schroeder

Windows Authentication Protocols

- **There are two authentication protocols present in the Windows world:**
  - NTLM
  - Kerberos
- **Kerberos only works in a domain environment**
  - It won’t work in workgroups
- **NTLM is support beginning Windows NT and later**
  - Uses challenge and response
  - This is different from the NT hash which is also called “NTLM”
- **Two versions of NTLM:**
  - NTLMv1 - insecure
  - NTLMv2 - less vulnerable, default since Windows 2000

Windows Password Representations and Storage

- **Windows can store passwords in two forms**
  - LANMAN
  - NT Hash
- **Local account passwords are stored in the SAM database by default**
- **LANMAN hashes are inherently insecure and should not be used**
- **In Active Directory, LANMAN and NT hashes are stored in the NTDS.dit file found on domain controllers**

LANMAN Hash Algorithm

1. **The password is padded with NULL bytes to exactly 14 characters.**
    a. If the password is longer than 14 characters, it is replaced with 14 NULL bytes for the remaining operations.
2. **The password is converted to all uppercase**
3. **The password is split into two 7-byte (56-bit) keys.**
4. **Each key is used to encrypt a fixed string.**
5. **The two results from step 4 are concatenated and stored as the LM hash.**

LANMAN Challenge/Response

1. **The client initiates a connection**
2. **The server sends a challenge**
3. **The client formulates a response from the challenge:**
    a. Padding LANMAN hash to 21 bytes
    b. Splitting LANMAN hash into three 7-byte pieces
    c. Using each piece as a DES key to encrypt challenge

NOTE:

NTLMv1 does the same thing, but it uses the NT hash as a starting point (not the LANMAN hash)

NT Hash Algorithm

- **The full password is hashed using MD4**
  - Case is preserved
  - Passwords up to 127 characters can be accepted
- **Hash is 128 bits (16 bytes)**
- **Password1234! -> 18C548F970E77575B0717EE09CA50BDC**
- **Not salted**
- **Salting is a process that combines the password with a random numeric value (the salt) before computing the one-way function.**

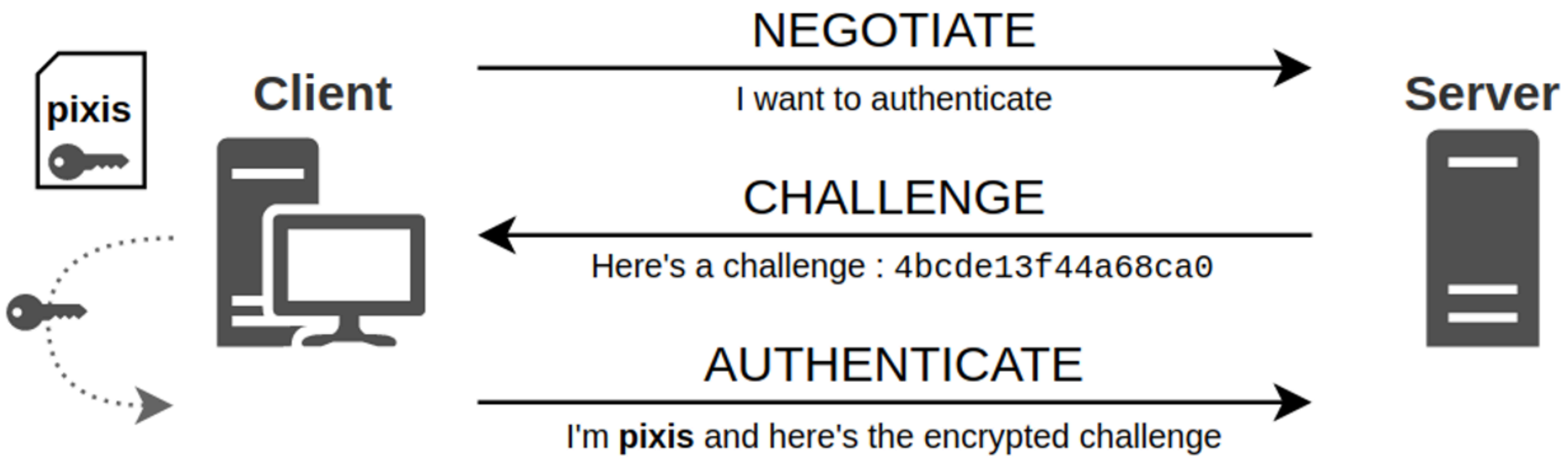
NTLM Authentication

- **NTLM authentication is a family of authentication protocols that are encompassed in the Windows Msv1_0.dll**
- **The NTLM authentication protocols include**
  - LAN Manager version 1 and 2, and NTLM version 1 and 2.
- **The NTLM authentication protocols authenticate users and computers based on a challenge/response mechanism**
- **NTLM =! NT Hash**
- **NT Hash is the password “at rest”**

NTLMv2 Challenge/Response

- **An improved version of authenticating over the network**
- **Client initiates the authentication**
- **The server sends server challenge**
- **Client formulates the response from challenge by:**
  - Creating the HMAC-MD5 of username and domain name with NT hash as the key
  - The result is called the NTLMv2 One-Way Function (OWF)
  - Then the response is created from the HMAC-MD5 of the server challenge, timestamp, client challenge, and other items, using the NTLMv2 OWF as the key

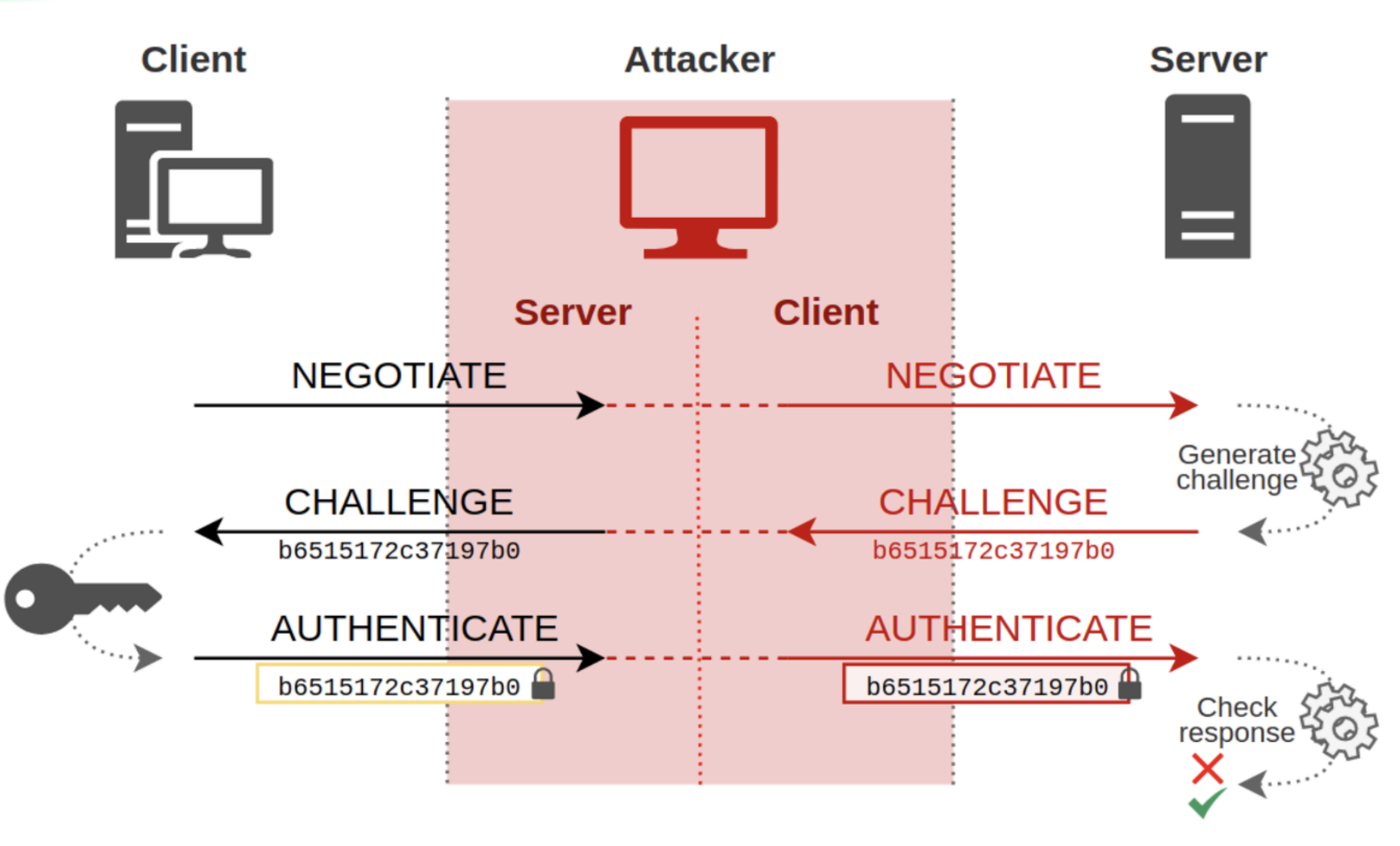
NTLMv2 in Action



https://en.hackndo.com/ntlm-relay/

Attacks on NTLMv2

- **Several attacks in AD utilize the capability in Windows to relay authentication material**
  - Sniffing of an NTLMv2 Challenge-Response
  - Relaying an SMB connection using NTLMv2
  - Instigating a connection from the victim to the attacker
    - Can be sniffed or relayed
- **Passively listen or coerce connections -> Relay to target -> Profit (dump credentials, command execution, elevate permissions)**
- **Cannot be used for “pass-the-hash”**
- **Responder, mitm6, PCredz, Hashcat**



https://en.hackndo.com/ntlm-relay/

Password Attacks

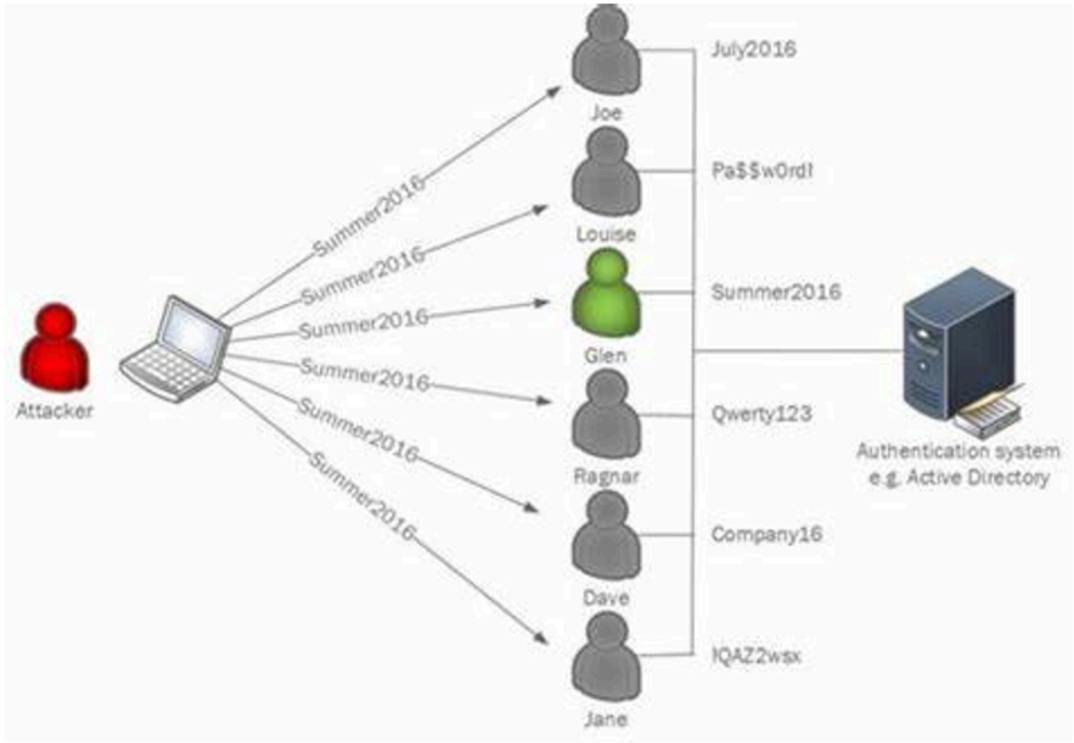
- **Password attacks can be categorized into two:**
  - Online
  - Offline
- **An online password attack is usually performed over the network with the attempt of logging in to the system**
- **A offline password attack attempts to recover the plaintext password from a gathered hash**

Password Reuse

- **Passwords are usually reused by end users and system administrators**
  - This means that a compromise of a user can be used as well to access other systems
- **Remember that a users in general would attempt to barely meet the requirements of a password policy**
  - This can lead to common passwords already found in dictionary lists
  - Passwords such as Summer2022! or Fall2022! satisfies Windows' default password policy

Password Spraying

- Password spraying is an online password attack where one password is used for many usernames
- Lesser likelihood for locking users out
- Attackers just need to get this correct once to gain a foothold



https://www.microsoft.com/en-us/security/blog/2020/04/23/protecting-organization-password-spray-attacks/

What About Unique Passwords?

- **Windows Local Administrator Password Solution (Windows LAPS) is a Windows feature that automatically manages and backs up the password of a local administrator account**
- **The implementation of LAPS adds two new attributes:**
  - ms-mcs-AdmPwd
  - ms-mcs-AdmPwdExpirationTime
- **Native LAPS PowerShell cmdlets, LAPSToolkit, PowerView, several tools exist to help you view the password**

Credential Dumping

“Adversaries may attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password. Credentials can be obtained from OS caches, memory, or structures.

Credentials can then be used to perform Lateral Movement and access restricted information.”

OS Credential Dumping

Sub-techniques (8)

ID	Name
T1003.001	LSASS Memory
T1003.002	Security Account Manager
T1003.003	NTDS
T1003.004	LSA Secrets
T1003.005	Cached Domain Credentials
T1003.006	DCSync
T1003.007	Proc Filesystem
T1003.008	/etc/passwd and /etc/shadow

https://attack.mitre.org/techniques/T1003/

Mimikatz

Mimikatz is a famous tool used to extract credentials in Windows environments

Plaintext passwords, hashes, PIN code, Kerberos tickets

Can be used to perform pass-the-hash and pass-the-ticket, or forge Golden Tickets

```
  .#####.   mimikatz 2.0 alpha (x86) release "Kiwi en C" (Apr  6 2014 22:02:03)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with  13 modules * * */


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 515764 (00000000:0007deb4)
Session           : Interactive from 2
User Name         : Gentil Kiwi
Domain            : vm-w7-ult-x
SID               : S-1-5-21-1982681256-1210654043-1600862990-1000
        msv :
         [00000003] Primary
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * LM       : d0e9aee149655a6075e4540af1f22d3b
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
```


https://github.com/gentilkiwi/mimikatz

DCSync

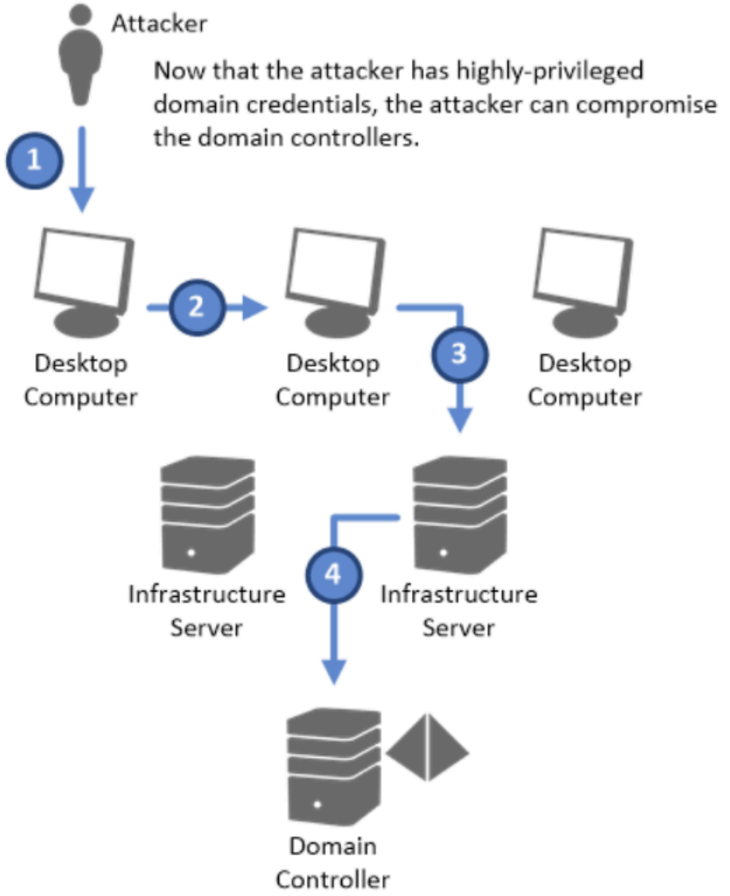
- **Uses the domain replication protocol to pretend as a DC and extract hashes**
- **Technique discovered by Benjamin Delpy, presented at Black Hat 2018**
- **Replicate information using the Directory Replication Service Remote Protocol (MS-DRSR)**
  - You cannot turn off this feature, it is necessary for the domain

```
PS C:\Users\chris.admin\Desktop> Invoke-Mimikatz -Command '"lsadump::dcsync /use
r:dev\krbtgt"'
Hostname: WINDOWS4.dev.testlab.local / S-1-5-21-4275052721-3205085442-277024194
2

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Aug 23 2015 23:05:23)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 16 modules * * */


mimikatz(powershell) # lsadump::dcsync /user:dev\krbtgt
[DC] 'dev.testlab.local' will be the domain
[DC] 'SECONDARY.dev.testlab.local' will be the DC server

[DC] 'dev\krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 4/18/2015 3:24:57 PM
Object Security ID   : S-1-5-21-4275052721-3205085442-2770241942-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 8b7c904343e530c4f81c53e8f614caf7
    ntlm- 0: 8b7c904343e530c4f81c53e8f614caf7
    lm  - 0: e05671e8363b1a1300afd2b86ddc3af6
```


https://blog.harmj0y.net/redteaming/mimikatz-and-dcsync-and-extrasids-oh-my/

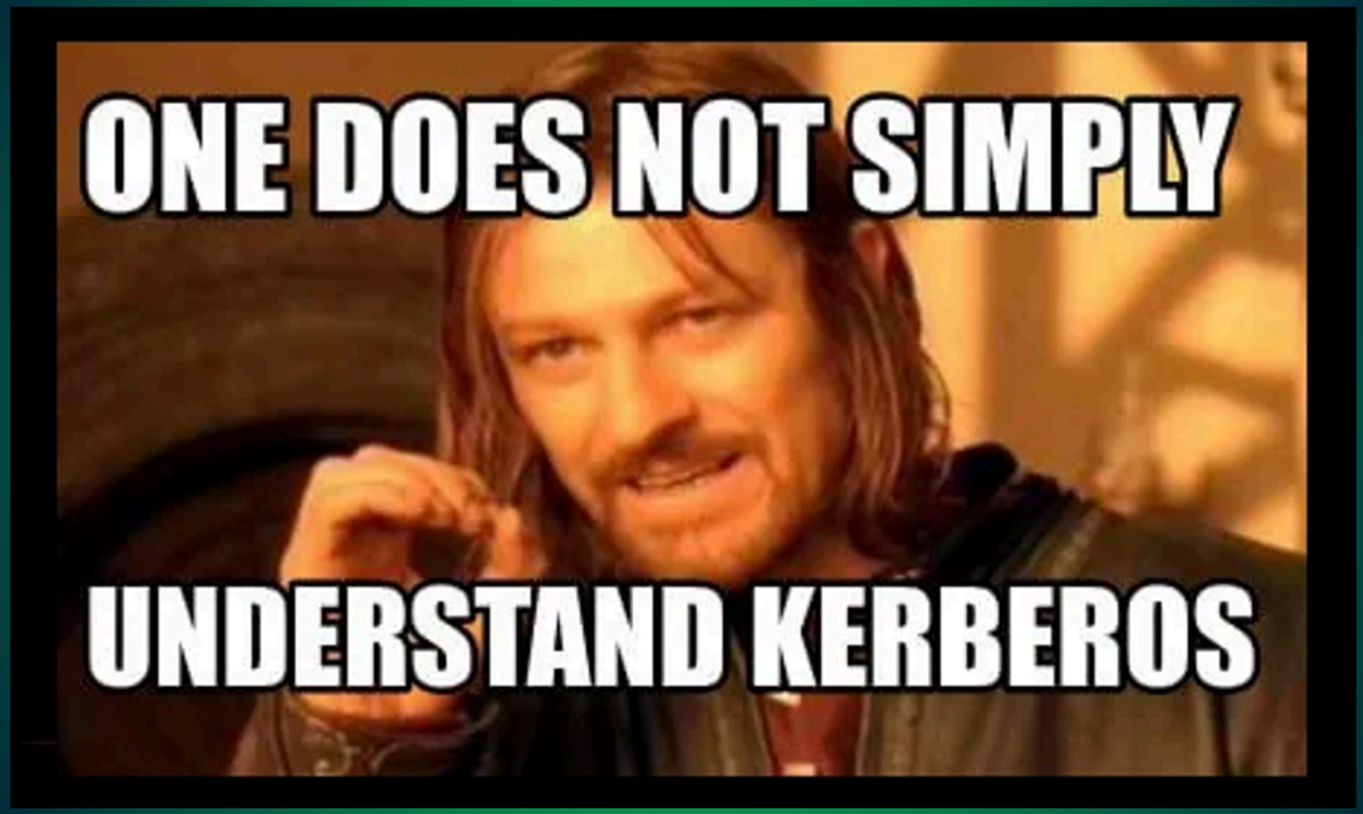
Pass-the-Hash

- **Hashes are extracted from memory, SAM, NTDS.dit**
- **Use the hash directly to authenticate to other systems**
  - Tools places the hash in the memory of LSASS
  - Windows automatically presents this material to target resources
- **No need to know the “plaintext”**
- **Basically, this is how Windows works**
- **Several benefits**
  - No account lockout
  - Cracking is not required
  - Gives access as the user whose hash is employed



https://www.microsoft.com/en-us/download/details.aspx?id=36036

Kerberos

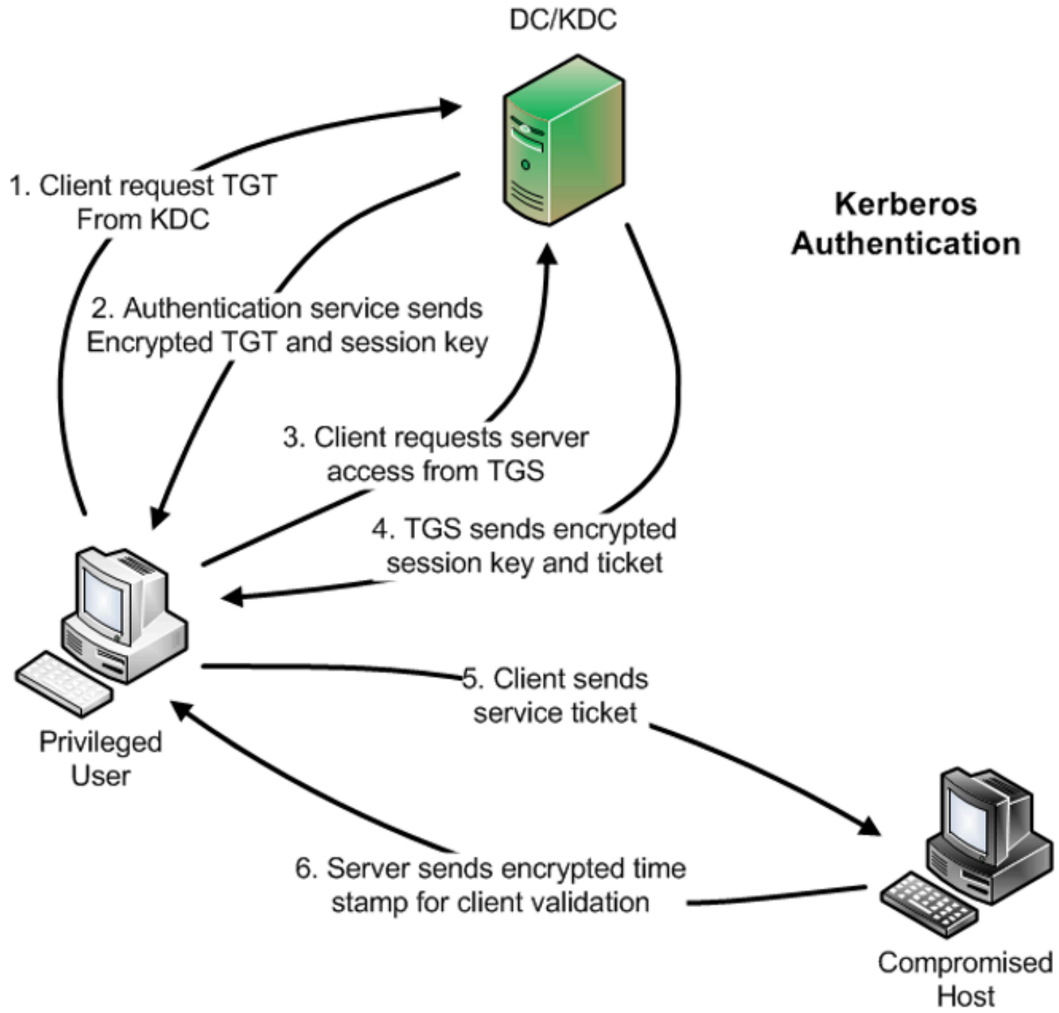


What is Kerberos?

- Kerberos is the main and preferred authentication protocol in Windows domains
- Kerberos relies on Service Principal Names (SPNs) and computer names for authentication
- Whenever we use an IP address, Kerberos can’t work – it will fall back to NTLM authentication
- NTLMv2 is rarely disabled in any environment

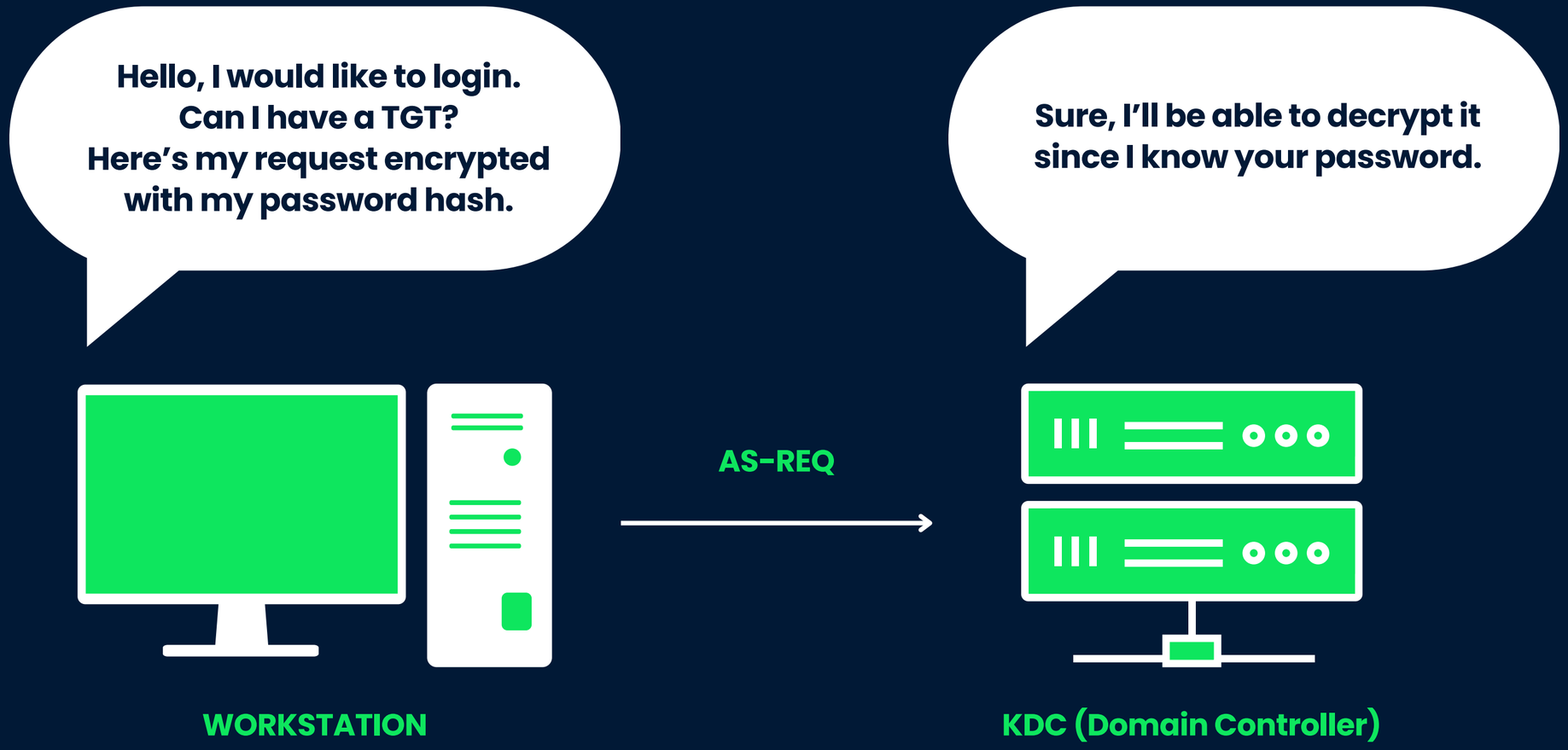
Kerberos Authentication

- **Kerberos is a network authentication protocol based on tickets**
- **Three main pieces:**
  - KDC (Key Distribution Center)
  - Client requesting access
  - Service the client is attempting to obtain access to
- **Pre-authentication leading to a Ticket Granting Ticket**
- **Request a Service Ticket**
- **Use Service Ticket**
- **The Privilege Attribute Certificate (PAC) is inside Kerberos tickets, defining if the user has access to the resource**

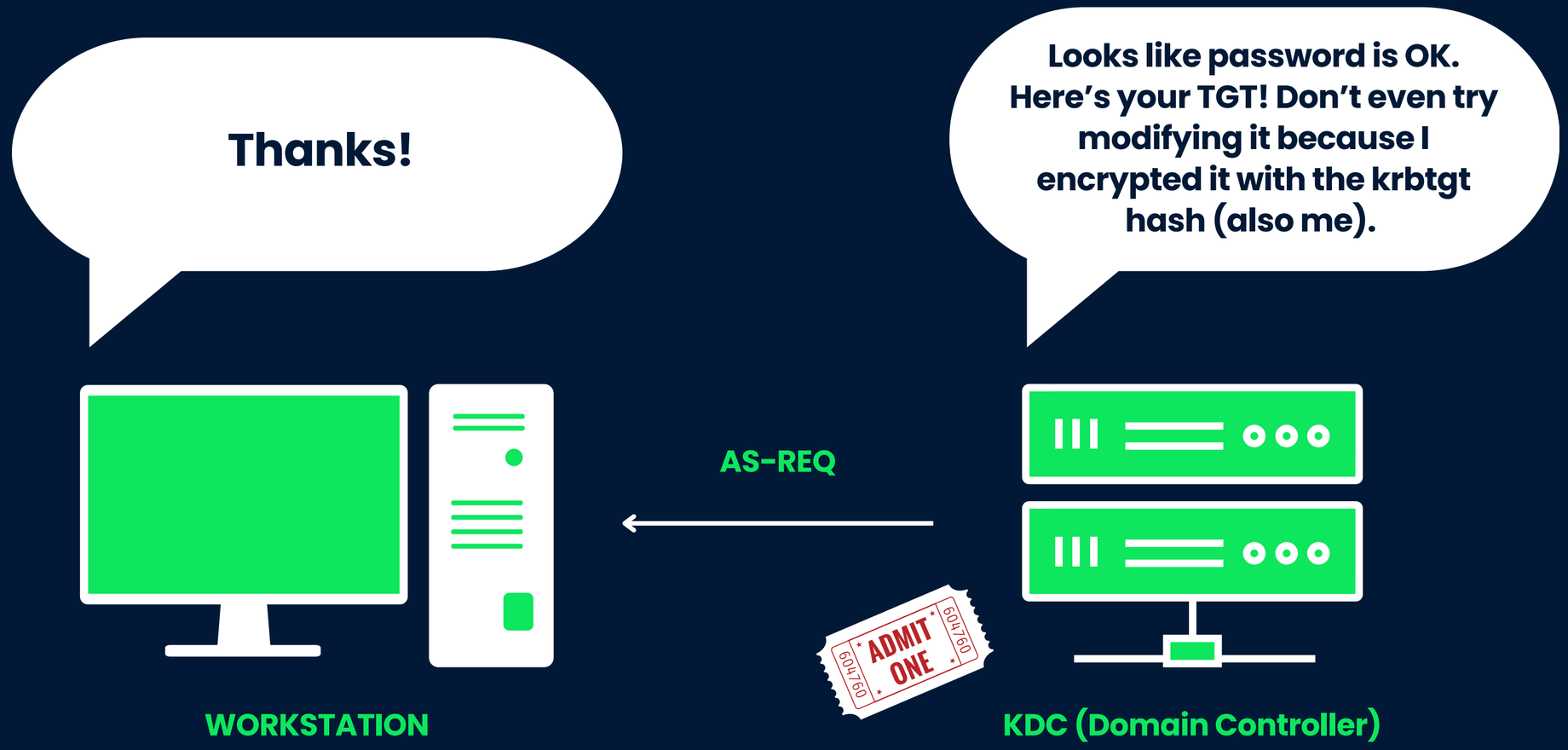


https://www.sans.org/blog/kerberos-in-the-crosshairs-golden-tickets-silver-tickets-mitm-and-more/

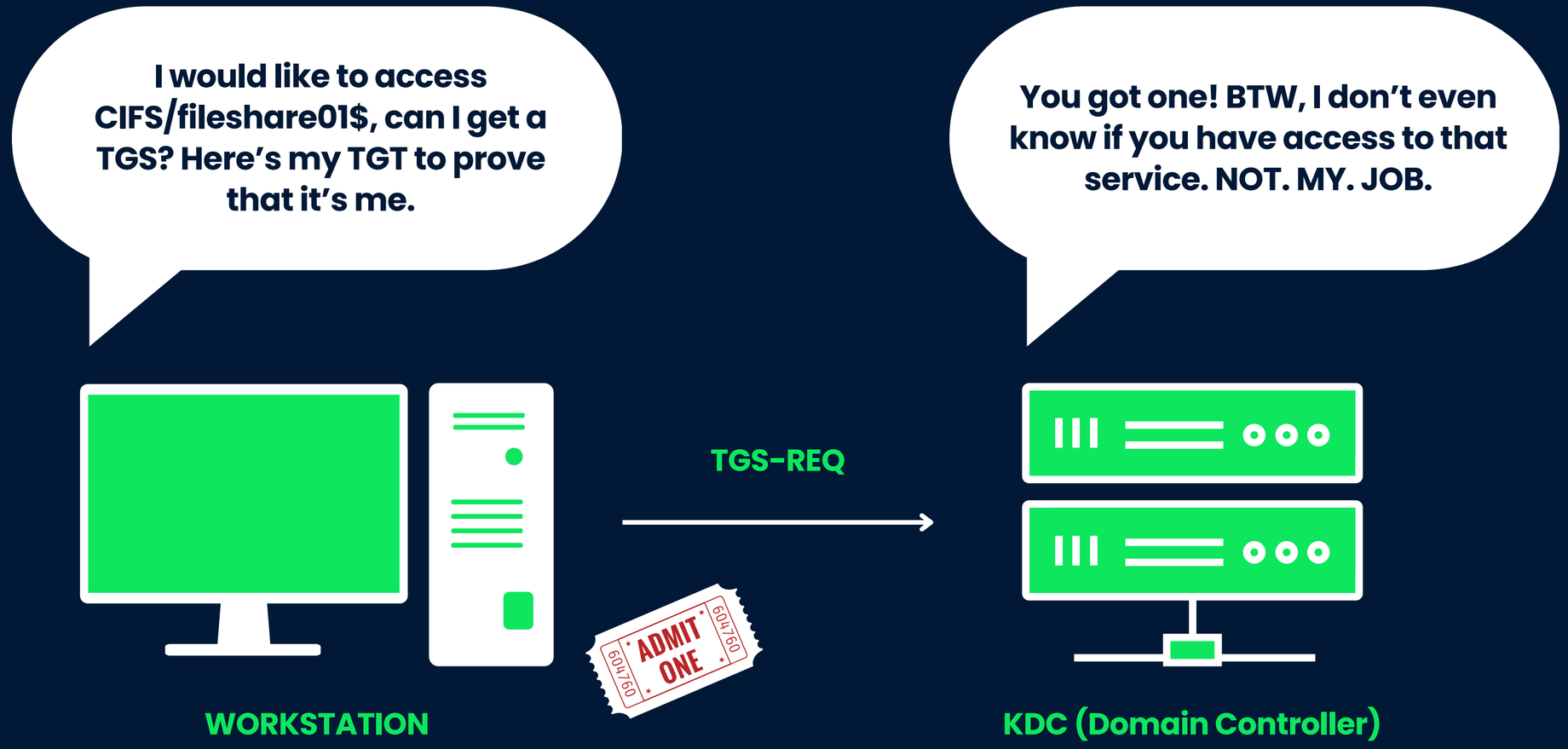
Kerberos in a nutshell



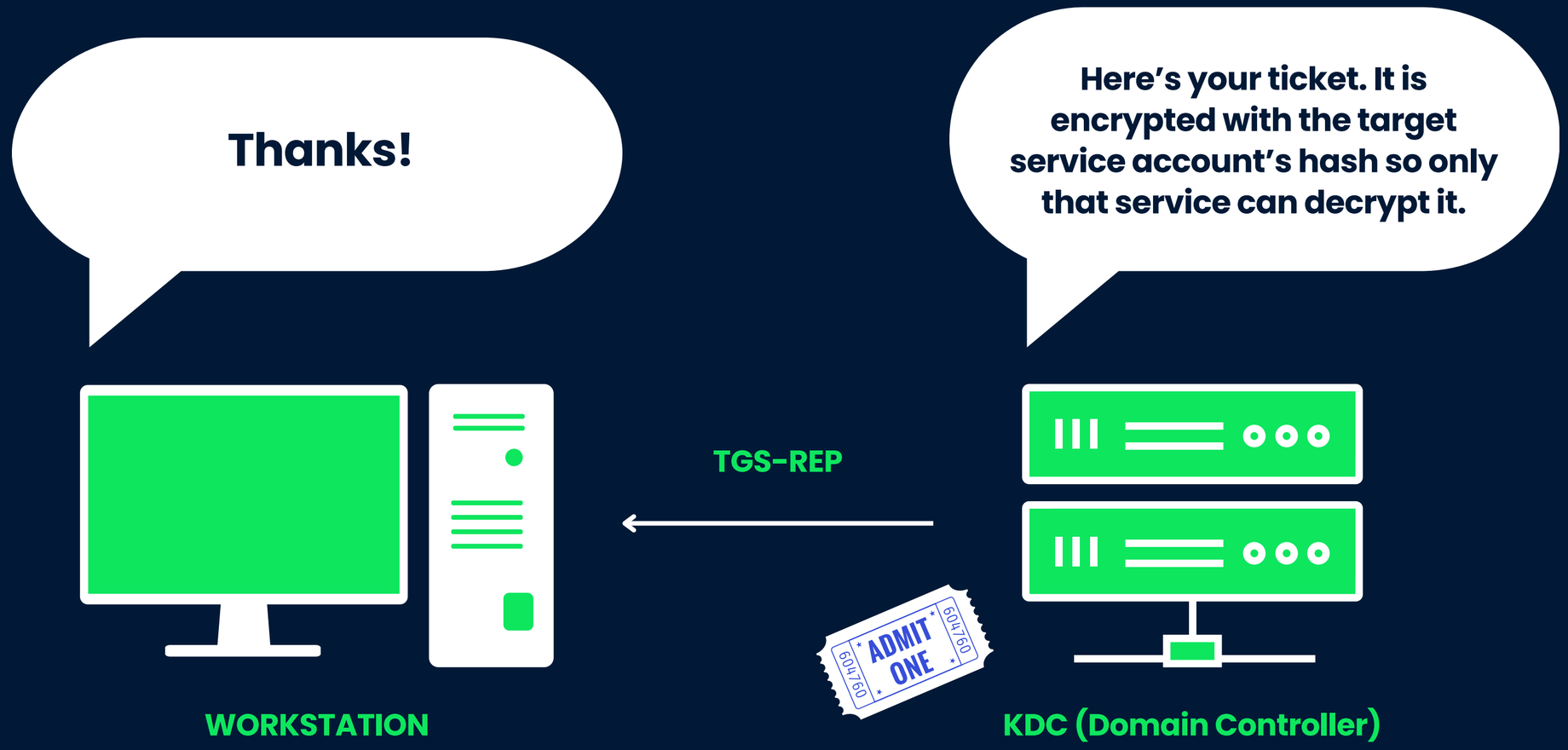
Kerberos in a nutshell



Kerberos in a nutshell



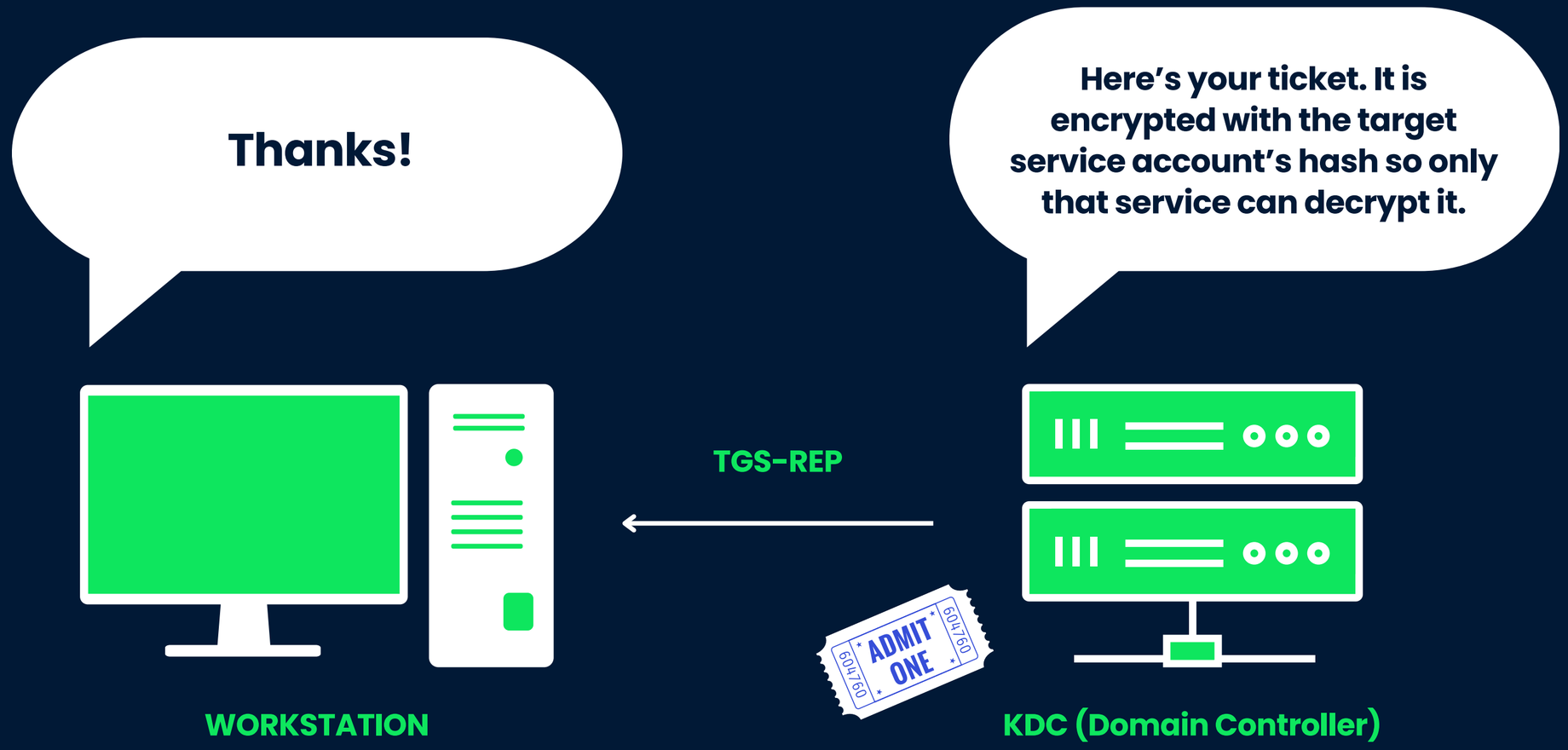
Kerberos in a nutshell



Kerberoasting

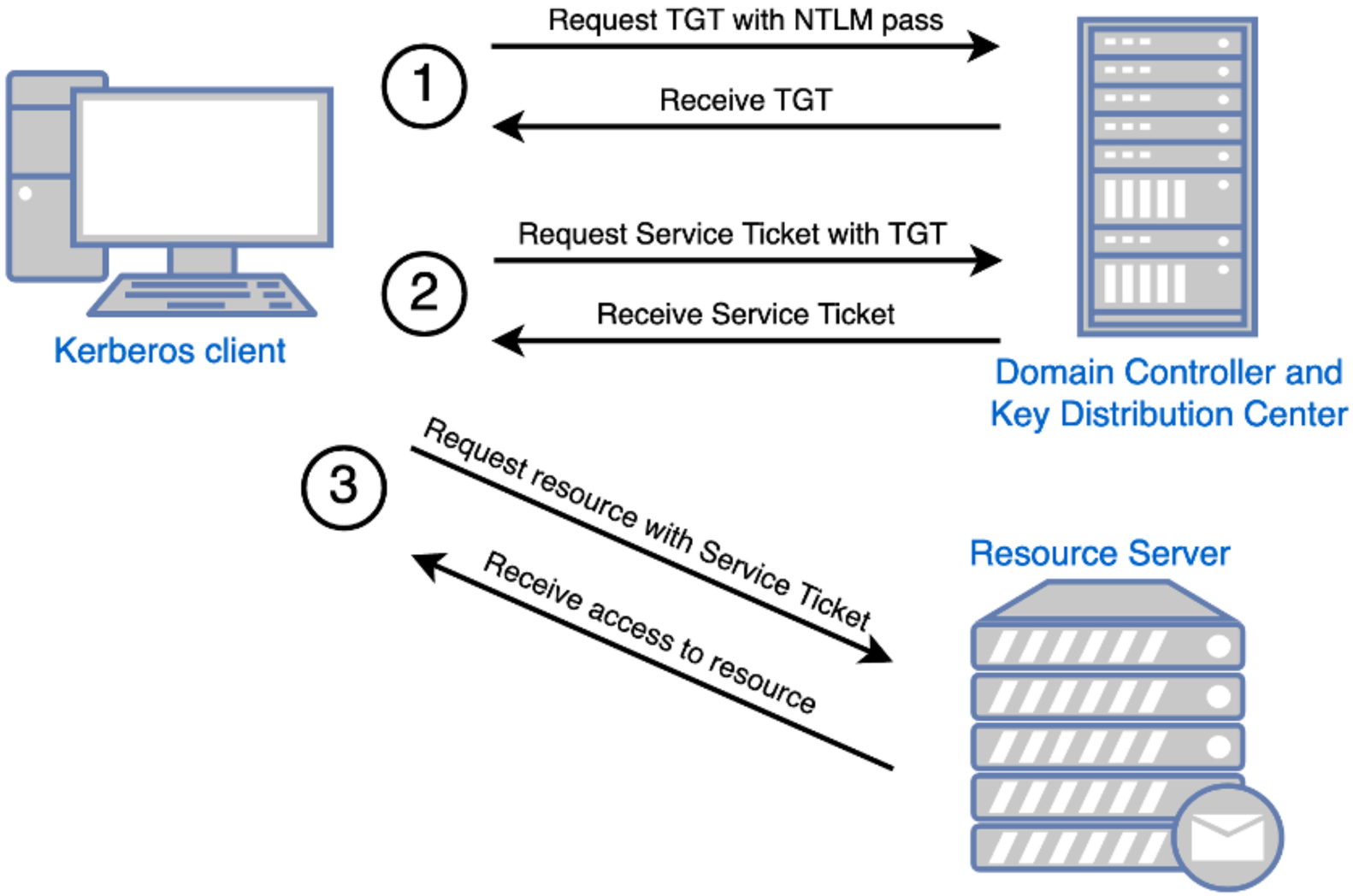
- **An attack discovered by Tim Medin last 2014**
- **A client can request a ticket for any service**
  - The client does not need permissions to the service
  - The service could be inaccessible due to a firewall
  - The service could be offline
  - The server could be recycled, as long as the SPN Exists
- **Kerberoasting is basically requesting tickets and cracking it offline**

Kerberos in a nutshell



Golden Ticket

- **A Golden Ticket is a forged TGT signed with the krbtgt hash**
  - Can be created without any interaction with the KDC - Kerberos is “stateless”
  - We do need the krbtgt hash (which means we have administrative privileges on the domain)
- **Tickets created using Mimikatz is valid for 10 years**

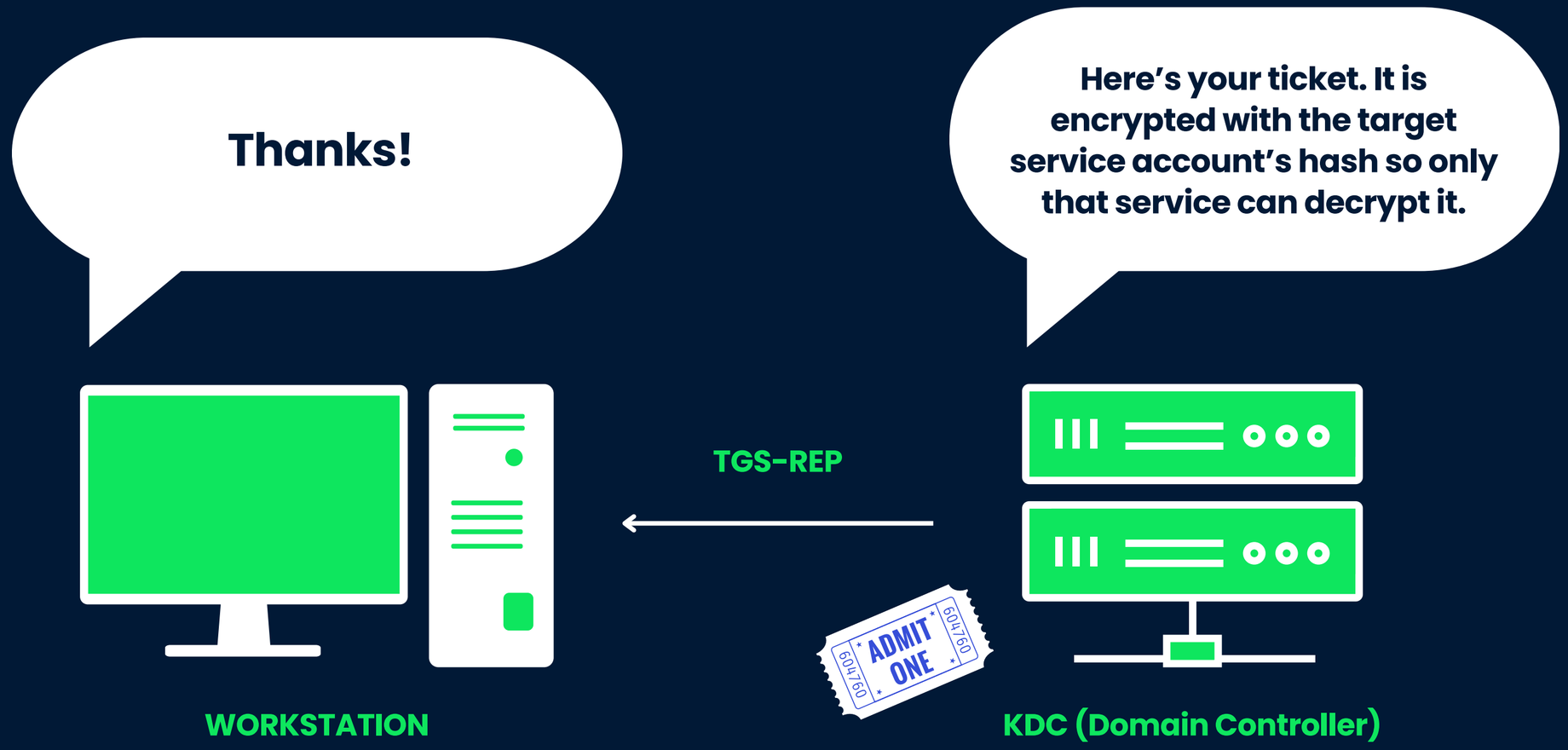


https://www.optiv.com/insights/source-zero/blog/kerberos-domains-achilles-heel

Silver Ticket

- **Remember, service ticket is signed using the service account's hash**
- **If we have the service account's hash, we can modify the PAC to escalate privileges and sign it**
- **This will work as long as the KDC does not validate the second signature**
  - HTTP and IIS will always verify due to the nature of AppPools
  - CIFS (SMB), HOST, MSSQLSvc, TERMSRV won't verify

Kerberos: TGS-REP



Active Directory Certificate Services (AD CS)

- **AD CS is a server role using Microsoft's Public Key Infrastructure (PKI)**
  - PKI is used for trust and encryption
    - Internally generated and trusted TLS certificates for HTTPS sites
    - Authentication via certificates for wireless and VPN
    - Authenticating and encrypting email
- **SpecterOps researchers identified several attacks again the most common AD CS misconfigurations**
  - 5 credential or certificate thefts methods
  - 6 account and domain persistence mechanism
  - 8 escalation techniques, including ways to become Domain Admin
- **Released at BlackHat 2021 by Will Schroeder and Lee Christensen**
  - https://specterops.io/wp-content/uploads/sites/3/2022/06/Certified_Pre-Owned.pdf

AD CS: ESC1

- **A escalation scenario where a regular domain user can become domain admin**
- **The Enterprise CA grants low-privileged users enrollment rights.**
  - The Enterprise CA's configuration must permit low-privileged users the ability to request certificates.
- **Manager approval is disabled.**
  - This setting necessitates that a user with CA "manager" permissions review and approve the requested certificate before the certificate is issued.
- **No authorized signatures are required.**
  - This setting requires any CSR to be signed by an existing authorized certificate.
- **An overly permissive certificate template security descriptor grants certificate enrollment rights to low-privileged users.**
  - Having certificate enrollment rights allows a low-privileged attacker to request and obtain a certificate based on the template.

What To Do Against Relaying Attacks

- **Relaying attacks all happen because of DNS Fallback**
- **Look for usage of LLMNR/NBNS/mDNS using Wireshark**
- **Test disabling the mentioned protocols**
- **Why did DNS fail? Figure that problem out**
- **Disable WPAD (Windows Proxy AutoDiscovery)**
- **Test SMB Signing**
  - Known to slow down SMB transfers. The amount of the performance loss depends greatly on the capabilities of the hardware involved.

What To Do Against Credential Stealing and Reuse

- **Domain Protected Users**
  - Prevents caching of credentials in memory
  - Kerberos will only use AES128 and AES256
  - When the domain functional level is 2012R2, NTLM is not allowed
- **Use “Protected Process”**
- **Windows Credential Guard**
- **Remote Credential Guard**

What To Do Against Kerberoasting

- Look for Kerberoastable accounts
- Disable the use of RC4 as encryption type
- Create a Fine-Grained Password Policy
- Apply Fine-Grained Password Policy to each Kerberoastable account

What To Do Against Password Attacks

MICROSOFT ENTRA PASSWORD PROTECTION

1. **Create a custom list of banned passwords**
2. **Install small agent on each on-prem DC**
3. **Start in Audit Mode, move to Enforced**

CONCLUSION

- **Attackers will always need initial access**
  - Can you detect it?
  - How long can you detect it?
- **Once they gain access, they will move inside your network**
  - Can you detect it?
  - How long can you detect it?
- **Reduce the ways attackers can get in**
- **Reduce the time to detect**
- **Increase their time to escalate**

Thank You

ROOTCON18