Internal Penetration Testing Basics

28.12.2022, Winterchaos 2022 / Chaos im LABOR mit LuXeria, Emanuel Duss

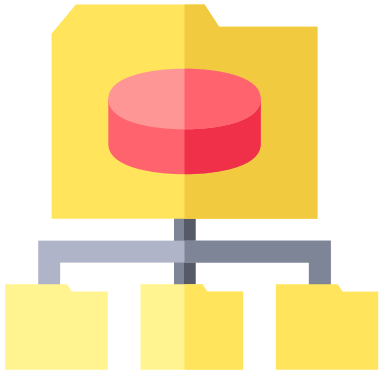
Agenda

- Welcome 👋
- Internal Penetration Testing
- Active Directory
- Local Privilege Escalation
- Credential Dumping
- Lateral Movement
- Conclusions

Internal Penetration Testing

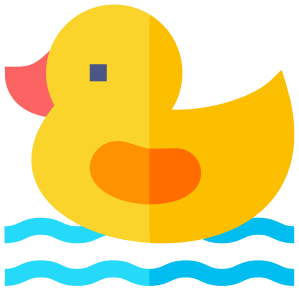
- Many methods for initial compromise
- Assume compromise: It's not **if** but **when** an attack will happen
- What can attackers do in an internal network?
  - Which services can be accessed?
  - Which privileges can be obtained?
  - Which systems can be compromised?
  - Which data can be accessed?
- Internal Pentest: Finding vulnerabilities in the internal network
- Initial Situation: access to the internal network, domain user & workstation
- The most interesting part is often the Active Directory infrastructure

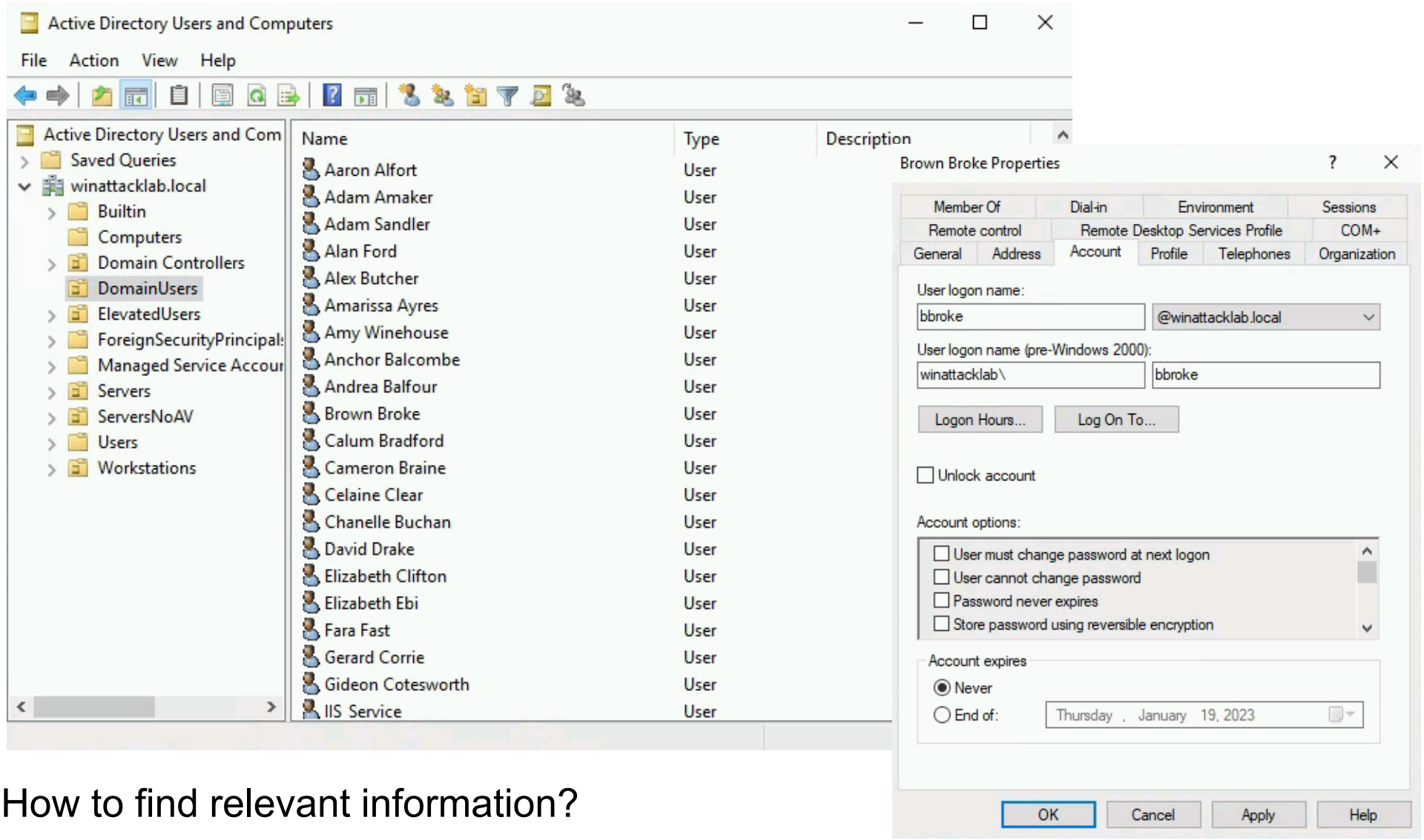
Active Directory



- Active Directory (AD) is a directory service (database) developed by Microsoft
- Used for centralized management of the IT infrastructure
- Structured in objects
  - Resources (e.g. file shares, printers)
  - Accounts / Security Principals (e.g. users, groups, computers/servers)
- A collection of objects is called a domain, stored on the Domain Controller (DC)
- Domains identified by DNS name (e.g. example.net, foobar.local)
- AD provides authentication and authorization mechanisms using NTLM or Kerberos
- Settings for the OS, applications or users can be deployed via GPOs
- Often: If the AD is compromised, everything is compromised.

AD Information Gathering





- How to find relevant information?

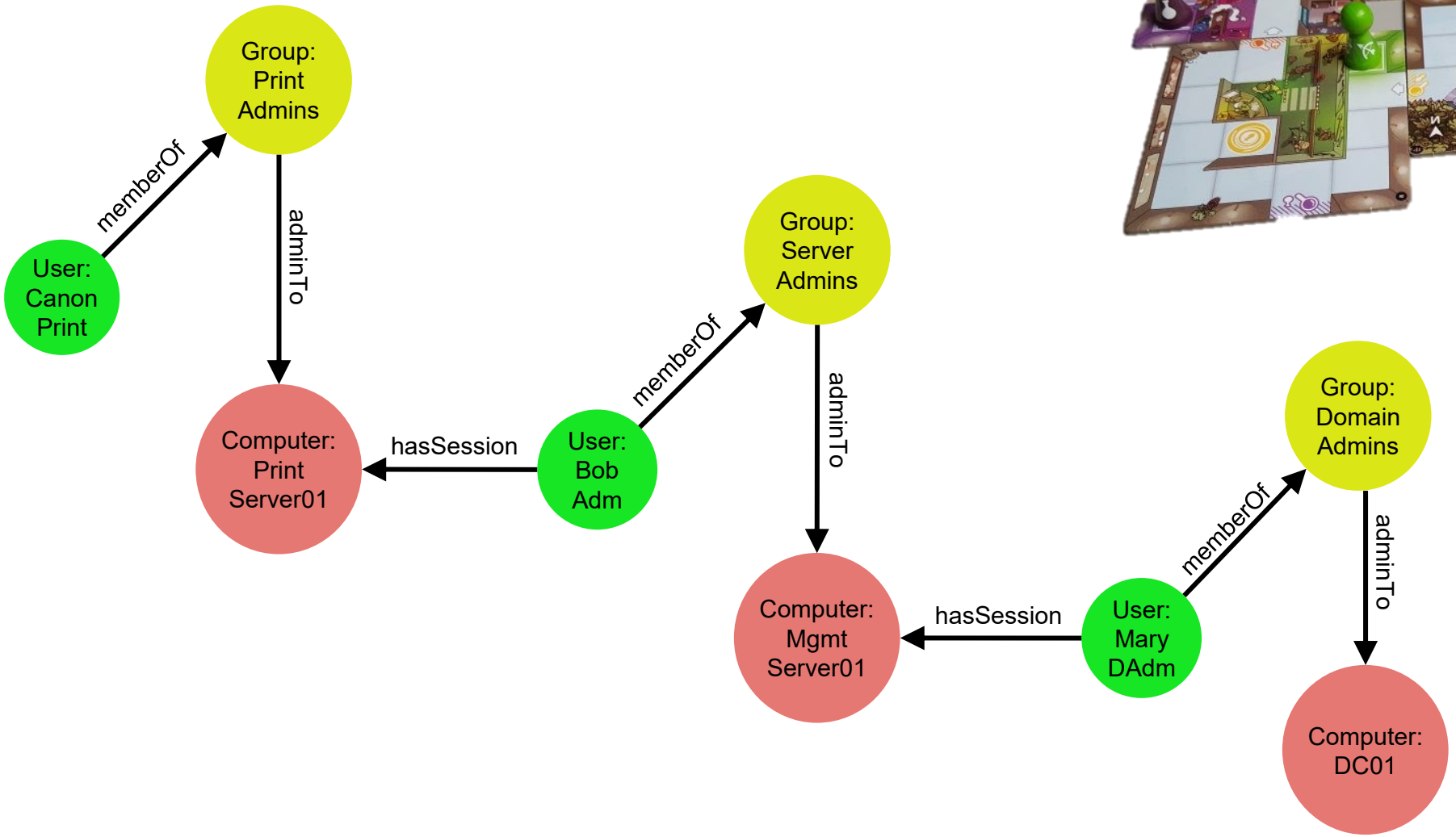
PingCastle

- «Ping Castle is a tool designed to **assess quickly the Active Directory security** level with a methodology based on risk assessment and a maturity framework. It does not aim at a perfect evaluation but rather as an **efficiency compromise**.»
- Requires access to the domain as a low-privileged user via DNS, LDAP and SMB
- Open source and free to use for non-commercial purposes.
- Web: https://www.pingcastle.com, https://github.com/vletoux/pingcastle

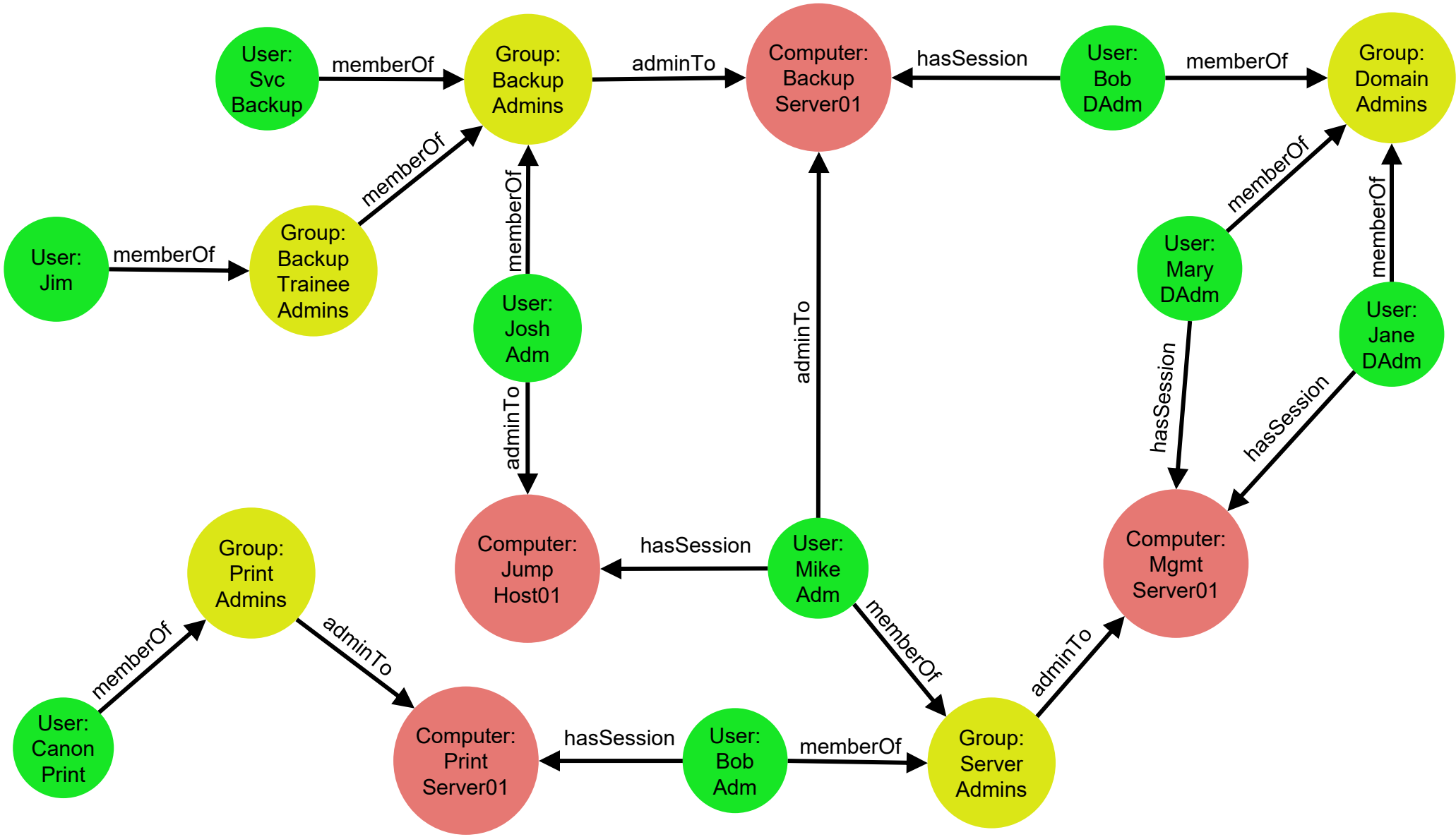
BloodHound

- BloodHound uses graph theory to reveal the hidden and often unintended relationships within an Active Directory environment.
- Relationship (edge) between multiple AD objects (nodes)
- Components
  - Collect data using SharpHound (C# Ingestor)
  - Feed data into Neo4j database (graph database)
  - Analyze data using BloodHound (Electron app)
- Open source
- Web: https://github.com/BloodHoundAD/BloodHound

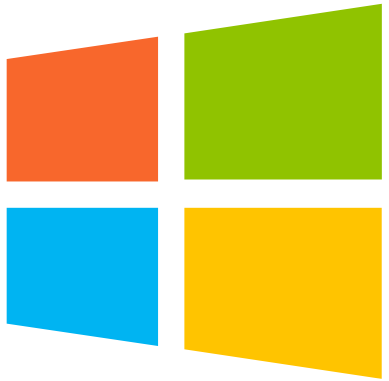
From Printer to Domain Admins



From Domain Admins to Users



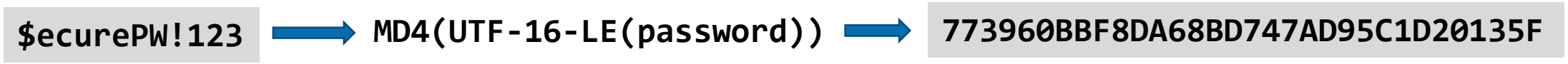
Local Privilege Escalation



- Gain elevated access on a system from low-privileged user to admin
- Common Methods
  - Vulnerable software through missing updates (OS, 3rd party services, …)
  - Stored cleartext credentials (documents, configuration files, scripts, …)
  - Local users / auto logon users distributed via GPOs
  - Write-permissions on login scripts, autostart entries, service binaries, application files
  - Modifiable services / tasks
  - Users being able to install own print drivers
  - Users being able to write to folders which are included in the PATH variable
- Many other specific misconfigurations…
- Tools like PrivescCheck can help to identify common issues

NTLM Authentication

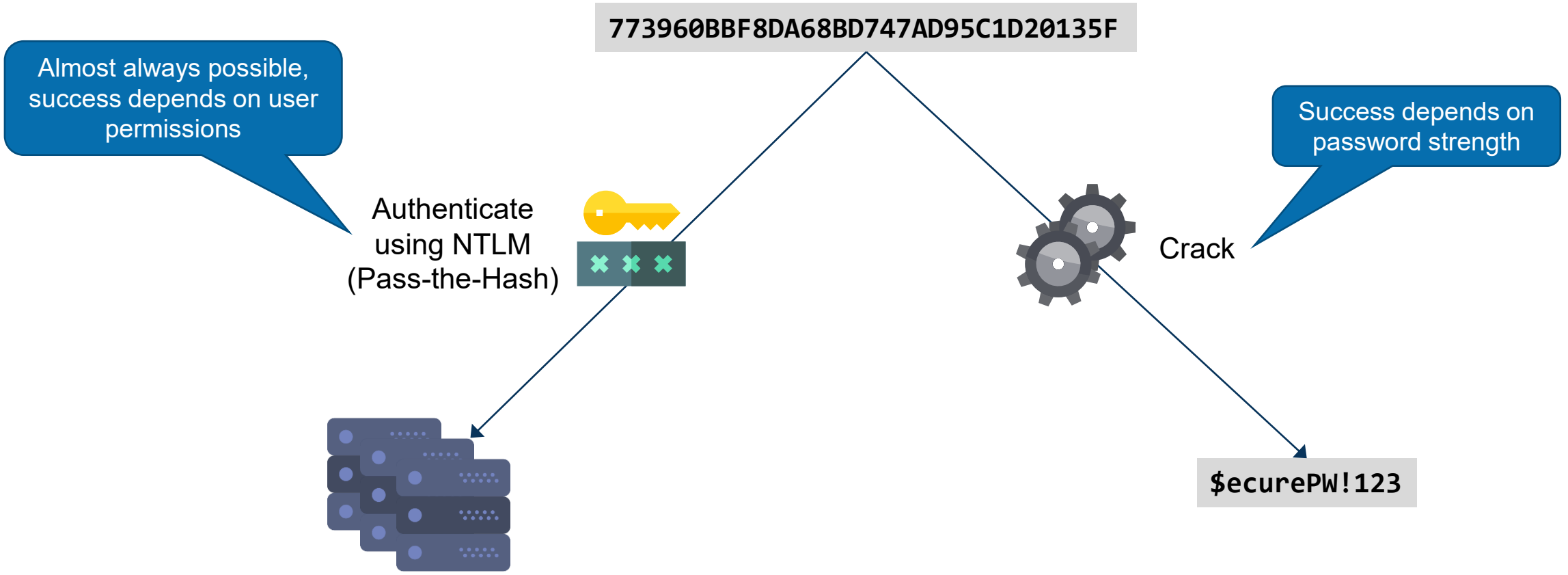
- One authentication method in Windows networks is NTLM
- Based on the user's password, but hashes are used internally
- These hashes are called **NTLM hashes** (technically NT hashes)



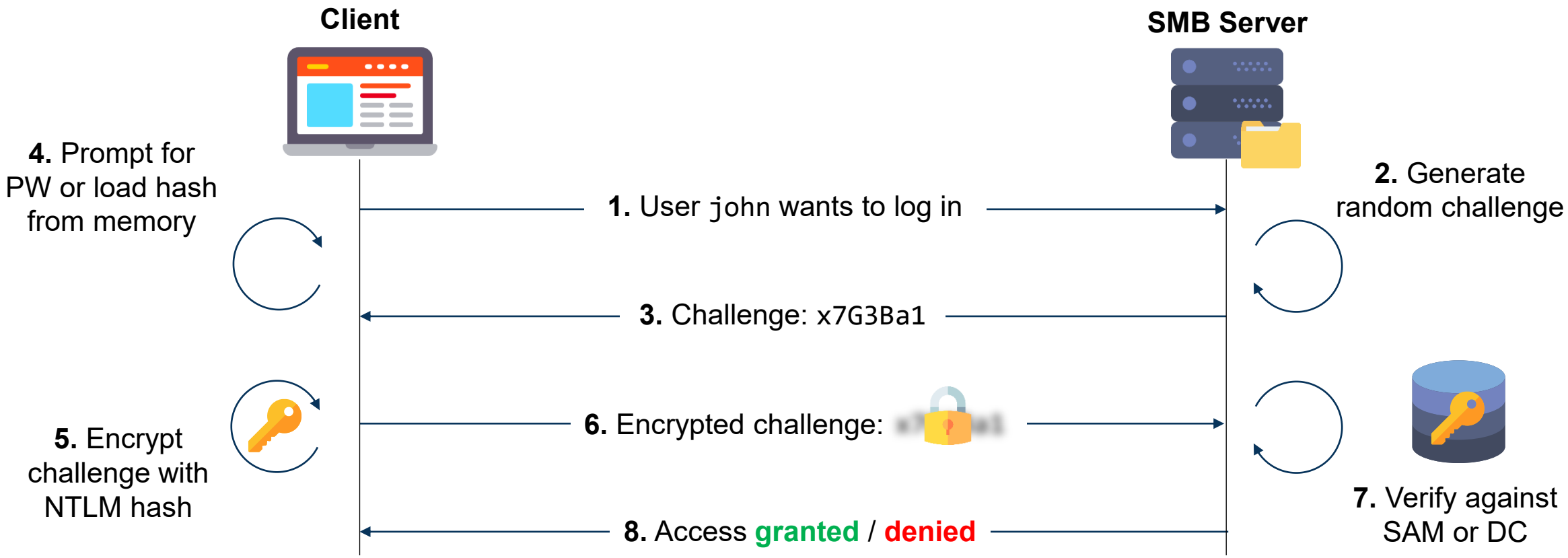
- Stored on different locations
  - **SAM** file: contains the hashes of local users
  - **LSASS.EXE** process: caches credentials of non-local users in memory
  - **NTDS.dit** file on a DC: contains the hashes of domain users
- Administrative privileges are required to access these hashes
- If an attacker acquires such privileges, they can "**dump**" credentials tools like Mimikatz

Using Hashes

- If an attacker gains access to a user's NTLM hash, they can perform two different attacks:



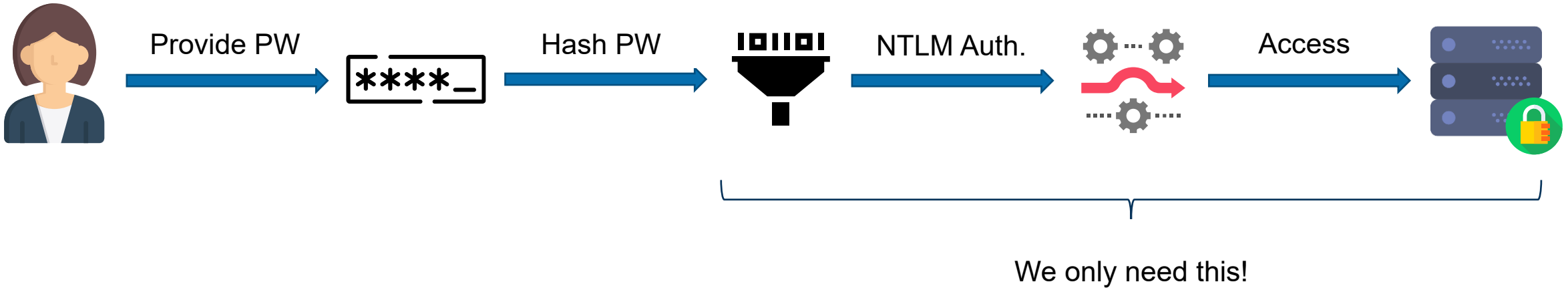
(Simplified) NTLM Authentication



→ NTLM authentication works without the plaintext password!

Lateral Movement via Pass-the-Hash

- The so-called **Pass-the-Hash** attack (published in 1997) uses a hash to authenticate against a (remote) system as the affected user
- This is not a vulnerability of the used protocols but rather a design implementation
- Not possible using regular Windows tools, because they always start with the user's PW



- There are various tools which support pass-the-hash

Conclusion

- AD Information Gathering
- Attack path analysis using BloodHound
- Local Privilege Escalation using weak WSUS configuration
- Credential Dumping using mimikatz
- Pass-the-hash

Countermeasures

- Local Privilege Escalation
  - System hardening / look for misconfigurations
  - WSUS Updates via HTTPS
- Credential Dumping / Pass-the-Hash
  - NTLM should be disabled, but this may break things
  - Don't re-use passwords for local admin accounts
  - Make use of the Protected Users Group in Windows AD
  - Implement logon restrictions for your privileged accounts to limit exposure
  - Don’t use domain accounts which are local admin on multiple systems
  - Use credential guard to protect the LSASS process
  - Use LAPS or a PAM solution

There is more…

- This was only a short introduction about basic AD attack techniques
- The attack path was very short with only one step
- There is much much more!
  - Password spraying
  - NTLM Relaying
  - Kerberos Attacks (Kerberoasting, Delegation)
  - Kerberos Relaying
  - DACL Abuse
  - GPO Abuse
  - MS SQL Server Misconfigurations
  - Active Directory Certificate Services
  - SCCM Abuse
  - Domain Trusts
- And more regarding non-AD attacks

@emanuelduss@infosec.exchange

@emanuelduss