ERNW
providing security.

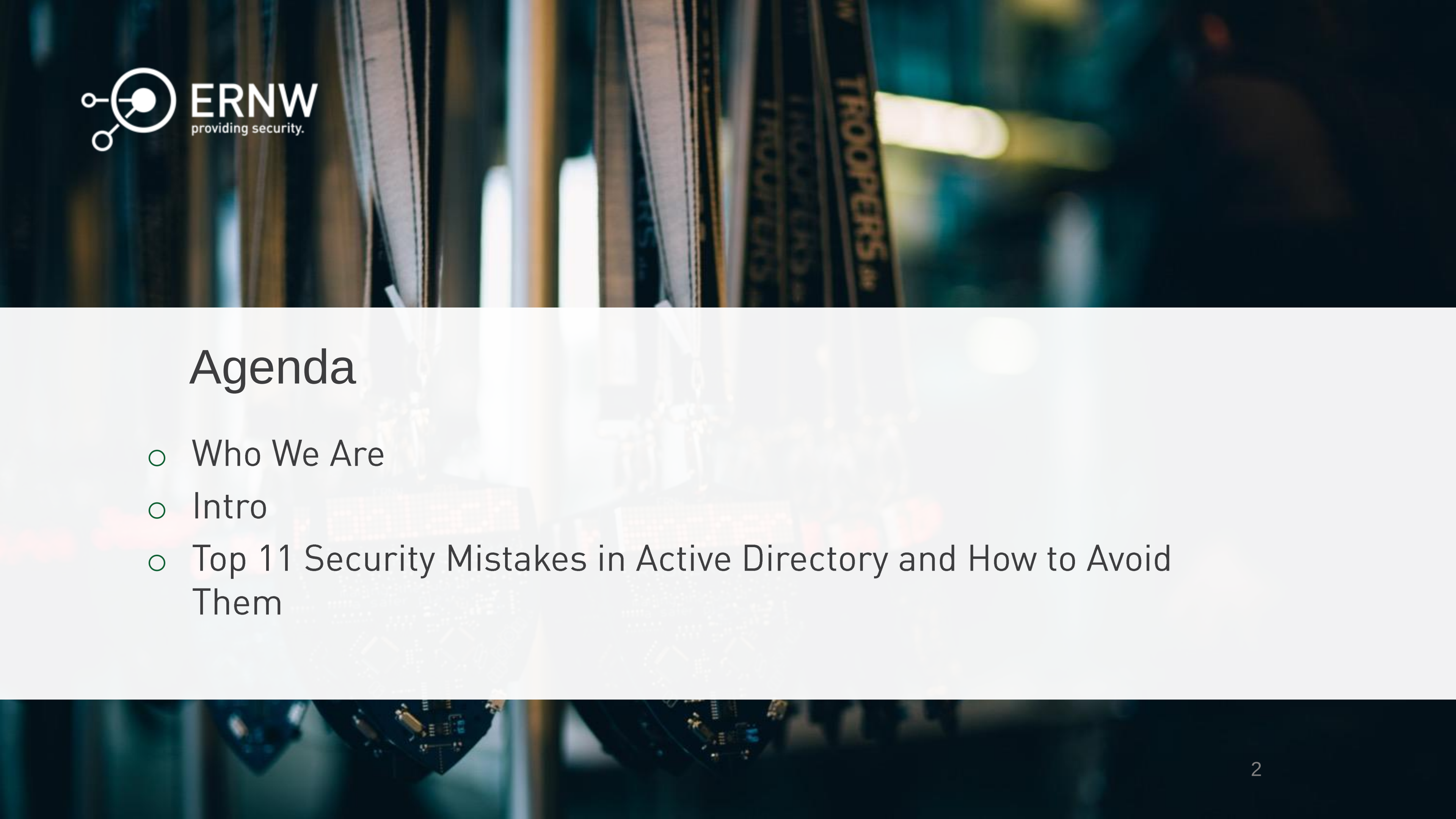
Active Directory Security Best Practices

“Top 11 Security Mistakes in Active Directory and How to Avoid Them”

Friedwart Kuhn & Heinrich Wiederkehr

Friedwart Kuhn
Digital unterschrieben von Friedwart Kuhn
Datum: 2017.04.12 20:51:31 +02'00'
Adobe Acrobat Reader-Version:
2017.009.20044

TROOPERS
MAKE THE WORLD A SAFER PLACE



Agenda

- Who We Are
- Intro
- Top 11 Security Mistakes in Active Directory and How to Avoid Them

Who We Are

- Friedwart Kuhn
  - Head of Microsoft Security Team @ERNW
  - 15+ years experience in security assessments, administration, publications and trainings
  - IT security professional with a focus on Windows Security and Active Directory Security

- Heinrich Wiederkehr
  - Member of Microsoft Security Team @ERNW
  - 4+ years in security assessments and trainings
  - IT security professional with a focus on Windows Security and Active Directory Security

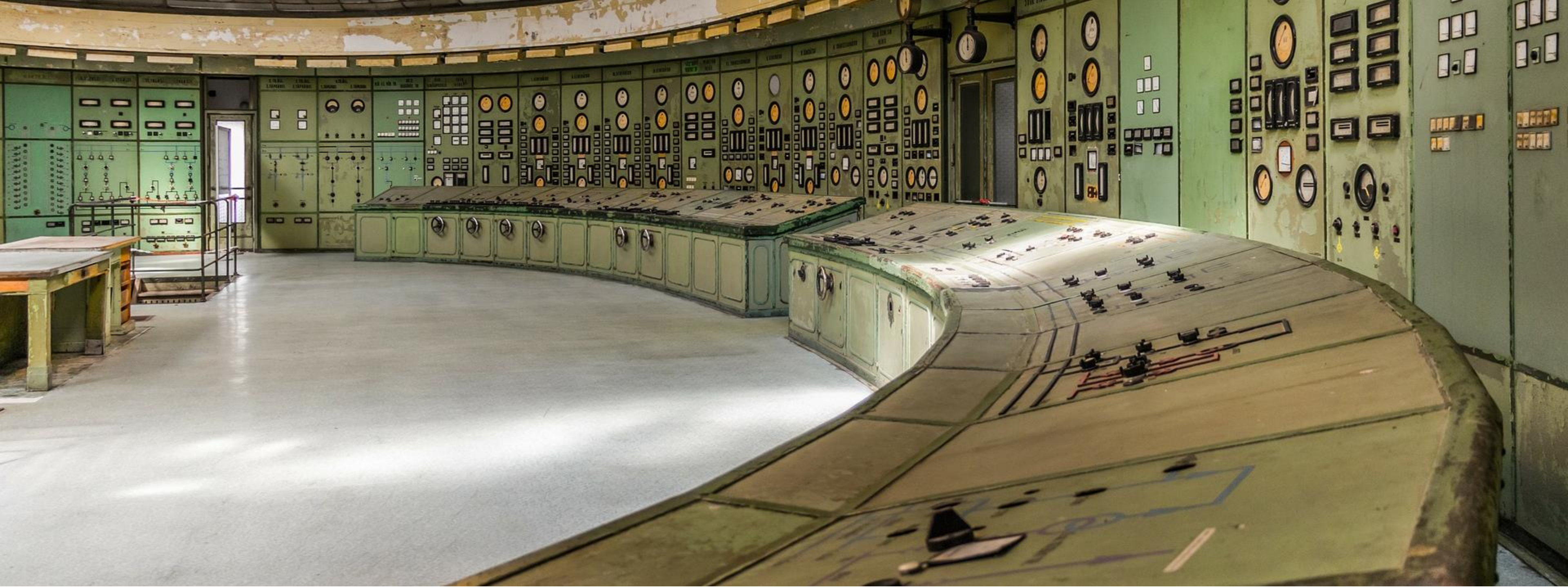
Intro

- Active Directory (AD) holds as main authentication backend in nearly *every* organization the keys to the crown jewels

- AD is heavily targeted by attackers that are using powerful, publically available tool sets

- Defense of AD environments overlooks often some typical design, implementation, configuration and operational mistakes.

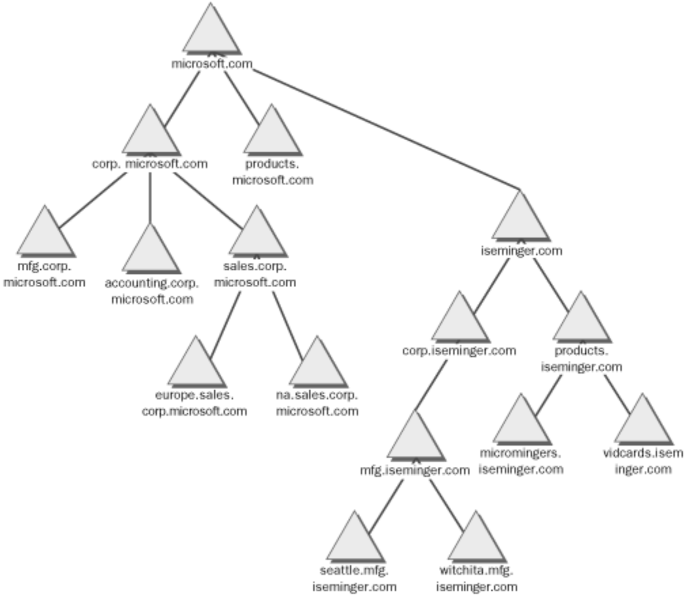
- We focus on 11 typical 'mistake areas' and we describe how to avoid or fix them



Mistake No. 1: **Lack of AD Governance**

The Problem: Lack of AD Governance

- Large enterprise ADs are
  - Historically grown
  - Distributed over different regions, companies, cultures
  - Built up and administered in different ways
- This is generally even true for a big AD of one company in one region…
- Enterprises claim to have governance, but they usually do not have AD governance


microsoft.com
corp. microsoft.com
products. microsoft.com
mfg.corp. microsoft.com
accounting.corp. microsoft.com
sales.corp. microsoft.com
europe.sales. corp.microsoft.com
na.sales.corp. microsoft.com
iseminger.com
corp.iseminger.com
products. iseminger.com
mfg.iseminger.com
micromingers. iseminger.com
vidcards.isem inger.com
seattle.mfg. iseminger.com
witchita.mfg. iseminger.com

https://www.microsoft.com/mspress/books/sampchap/3173.aspx

The Solution: Dedicated AD Governance

- Create an AD Governance Board that *governs* high level principles and policies for *all* ADs of your Organization

- Since AD is the central authentication infrastructure and holds in most Organizations keys to the "Crown Jewels", a dedicated AD Governance Board is not only justified but even necessary.

- The AD Governance Board must have at least experienced AD architects, AD security specialists and AD administrators/operators as members. The CISO should be a member.

Good AD Governance

A Program Promoting Rules of Law for Organization's AD

The Solution: Dedicated AD Governance

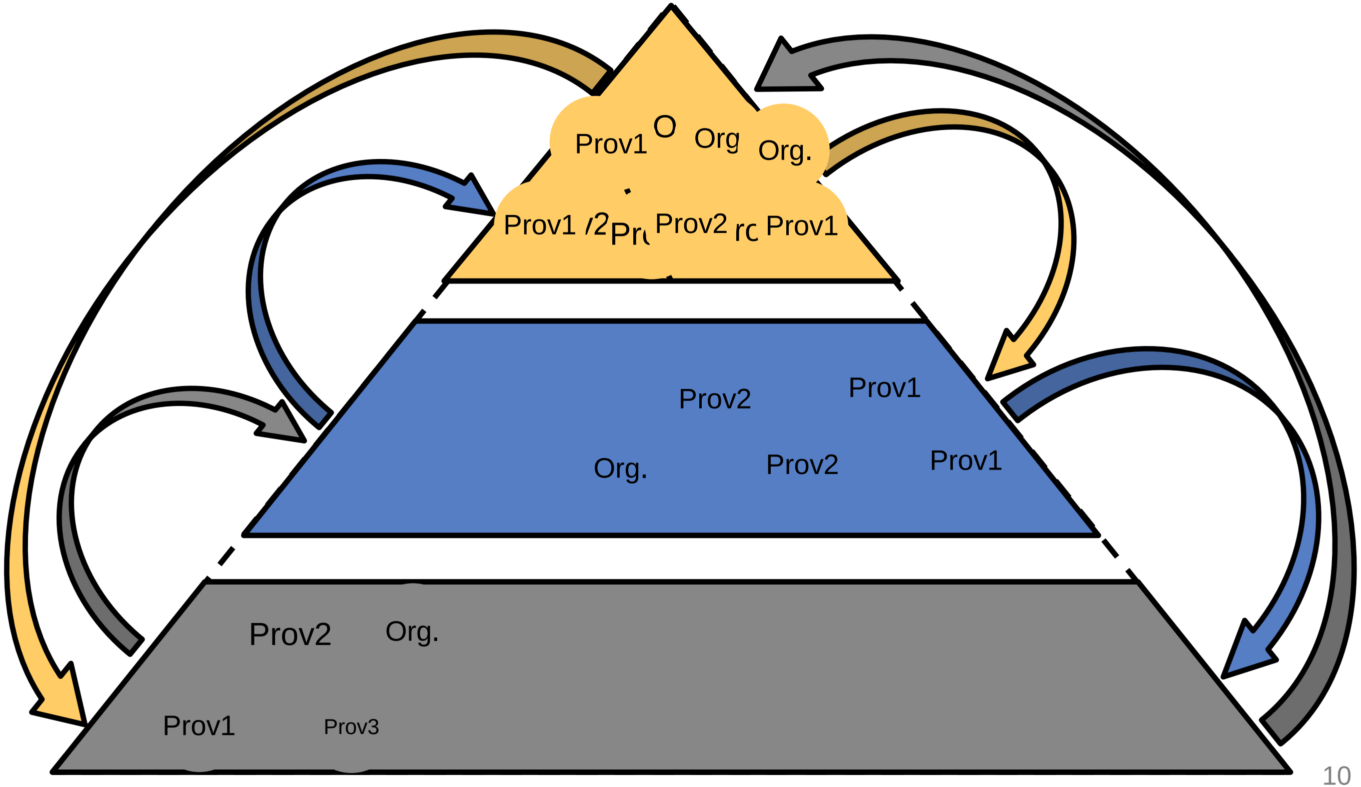
- Tasks of the AD Governance Board
  - Govern high-level security & design controls
  - Have an idea of an overall Target AD Design
  - **Provide** organizational and technical guidance such as:
    - How to implement Admin Tiers
    - How to implement PAWs
    - Hardening Guidelines for DCs, Servers, Clients, non-Windows members
    - Etc.

Good AD Governance

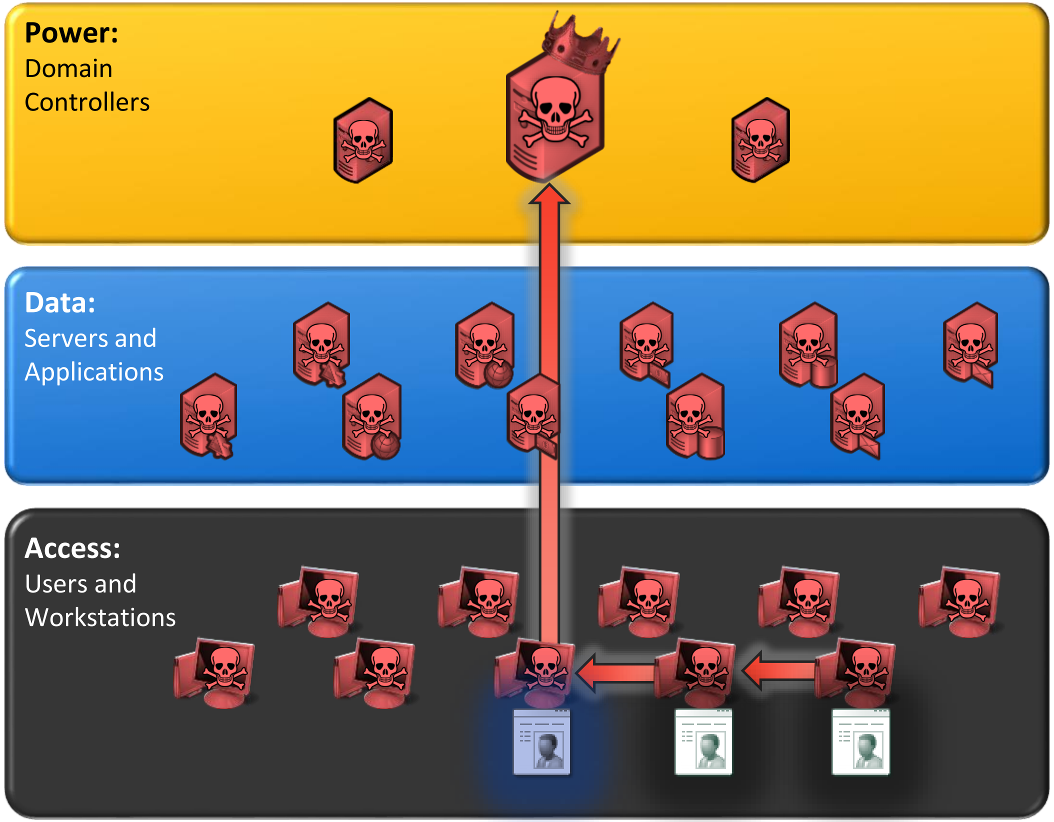
A Program Promoting Rules of Law for Organization’s AD

Mistake No. 2: **Admins (and Service Accounts) Logging on Everywhere**

The Problem: Admins Logging on Everywhere…



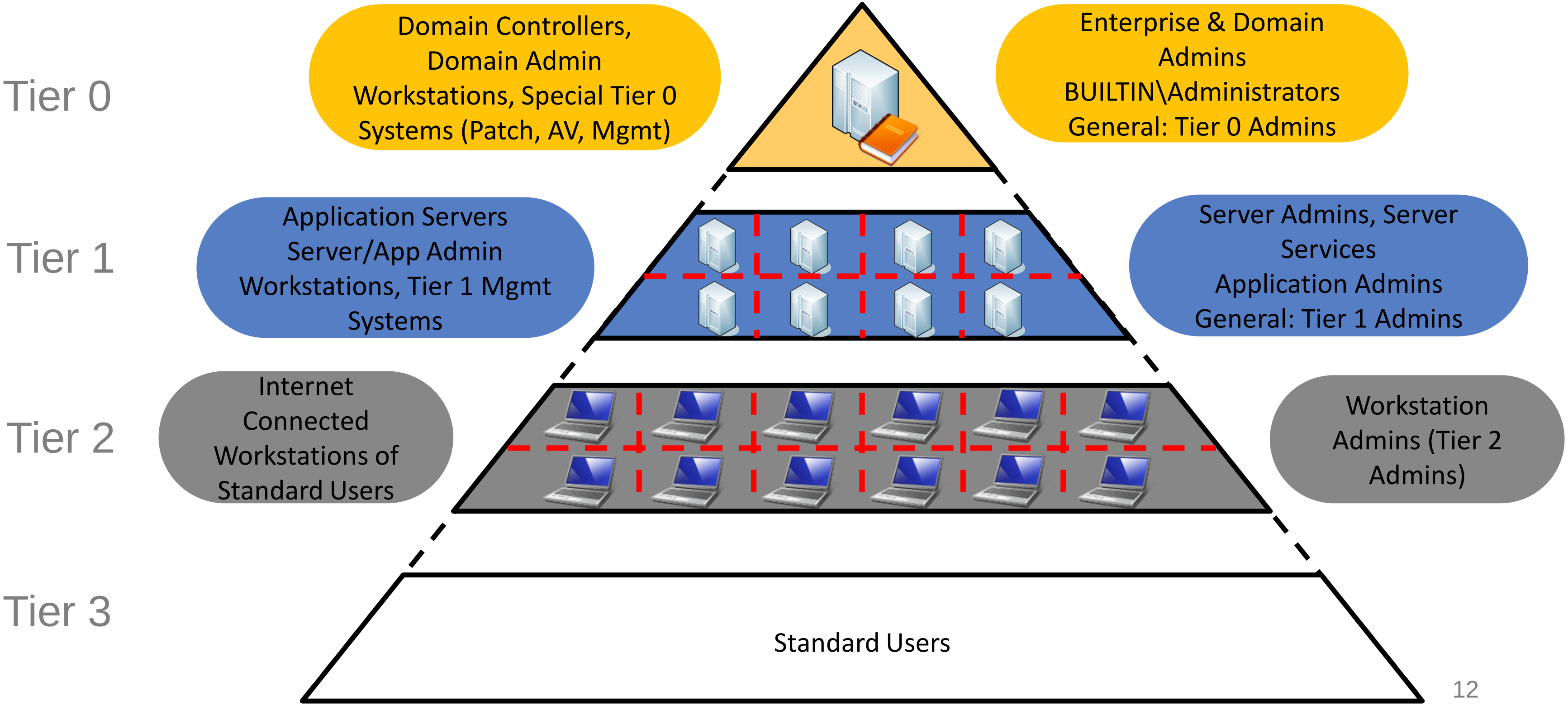
Result of Mistake 2…



1. Bad guy targets workstations en masse
2. User running as local admin compromised, bad guy harvests credentials.
3. Bad guy starts "credentials crabwalk"
4. Bad guy finds host with domain privileged credentials, steals, and elevates privileges
5. Bad guy owns network, can harvest what he wants.

This slide is from: Mark Simos, Nicholas DiCola; "TWC: Pass-the-Hash and Credential Theft Mitigation Architectures"

The Solution: Implement Administrative Tiers



Tier 0

Domain Controllers, Domain Admin Workstations, Special Tier 0 Systems (Patch, AV, Mgmt)

Enterprise & Domain Admins BUILTIN\Administrators General: Tier 0 Admins

Tier 1

Application Servers Server/App Admin Workstations, Tier 1 Mgmt Systems

Server Admins, Server Services Application Admins General: Tier 1 Admins

Tier 2

Internet Connected Workstations of Standard Users

Workstation Admins (Tier 2 Admins)

Tier 3

Standard Users

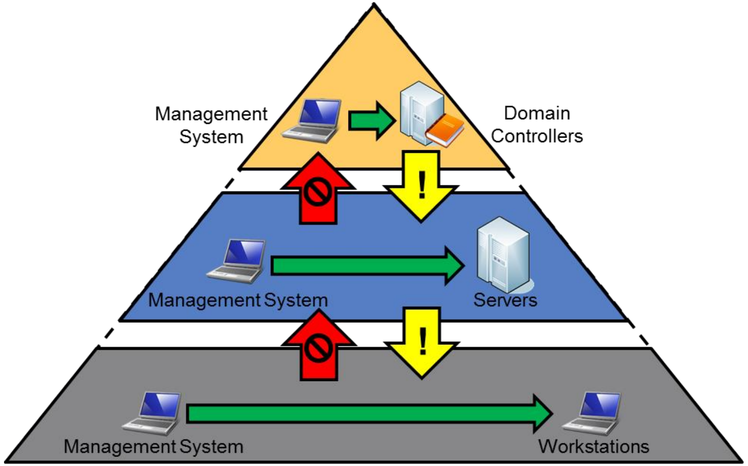
Tier Model Principles

Classify: *Every single* security principal, system, or application **has to be classified as belonging *only* to one tier**

Restrict Logons: Security principals of a higher tier ***must never log on to* a resource on a lower tier** (→ Implement logon restrictions)

Restrict Control: Security principals of a lower tier ***must never control* resources of a higher tier** (→ Implement control restrictions)

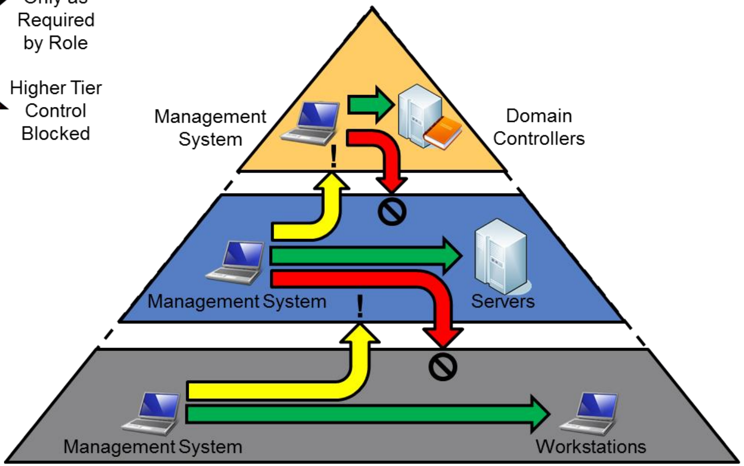
Control Restrictions vs. Logon Restrictions



🟩 Same Tier Control

! Lower Tier Control Only as Required by Role

🟥 Higher Tier Control Blocked



🟩 Same Tier Logon

🟨 Higher Tier Logon

🟥 Lower Tier Logon

⊘ Blocked

! Only as Required by Role

Implementation Guidelines

- Begin with Tier 0
  - Followed by Tier 1 and then Tier 2
- Use compartments in Tier 1
- Do not let service accounts undermine the Administrative Tier model
- Provide admins with detailed technical guidelines (about the consequences of logon & control restrictions)
- Count with a long-term project

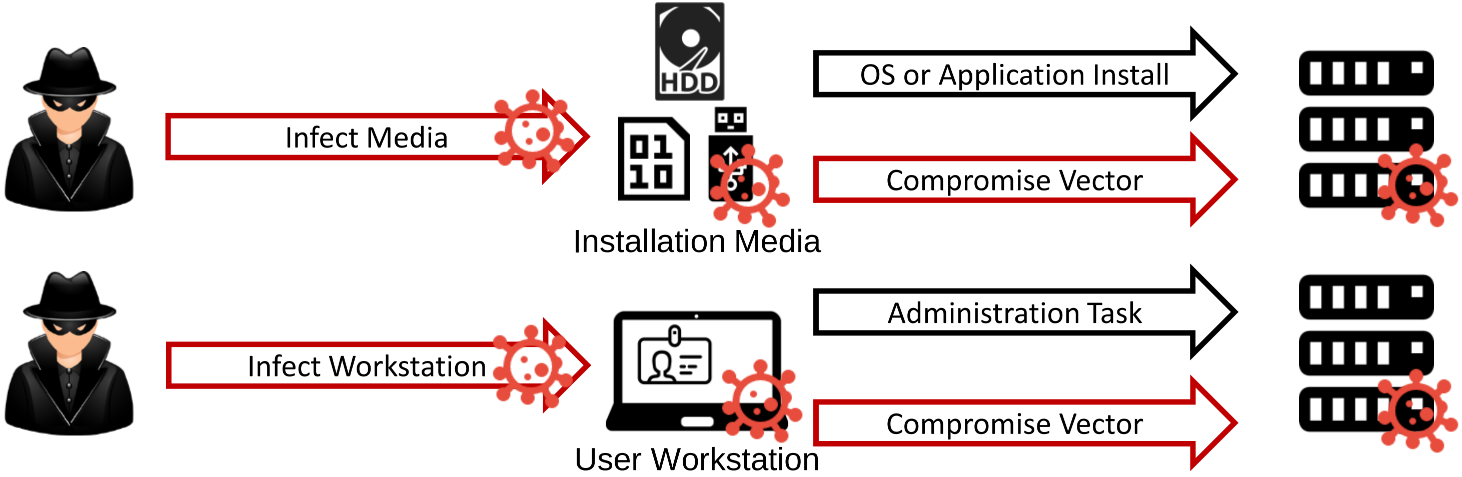
Summary

- The most important and comprehensive Active Directory-specific security control with respect to credential theft & reuse
- Basis for many other technical controls
- Future (Windows) administration model
- Requires modification in admin mindset
- Admins will have more accounts and hence higher operational effort
- Alternatives
  - None

Mistake No. 3: **Using “Dirty Sources”**

The Problem: Security Dependencies

- Security dependencies are not always as trustworthy as the object being secured. For example:

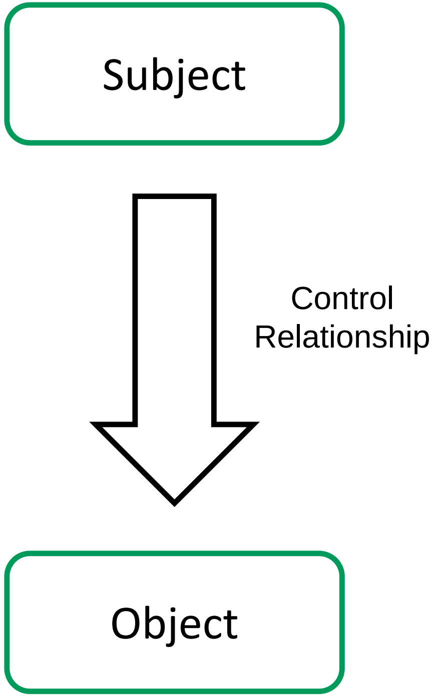


The Solution: Clean Source Principle

- Any subject in control of an object is a security dependency of that object
  - The assurances for all security dependencies **must be at or above** the **desired security level of the object** itself

⚠ **Control is transitive!** (For example if A controls B and B controls C, then A also indirectly controls C.)

- Most common areas of control are:
  - the hardware where systems are installed,
  - the installation media for the systems,
  - the architecture and configuration of the system,
  - and daily operations.



ERNW providing security.

TROOPERS MAKE THE WORLD A SAFER PLACE

Clean Source Principle: Installation

- Ensure that the installation media has not been tampered with
  - Requires validating the **software integrity** throughout the lifecycle including during **acquisition, storage**, and **transfer** until usage

- **The source of the software must be validated**
  - Physical media directly from the vendor
  - File hash validation
  - Revocation checks for digital signatures

- **The software must be stored in a location that is protected from modification**
  - especially by internet-connected hosts
  - or personnel trusted at a lower level than the systems where the software system will be installed

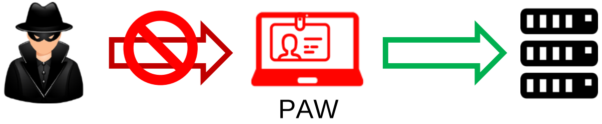
Clean Source Principle: Administration

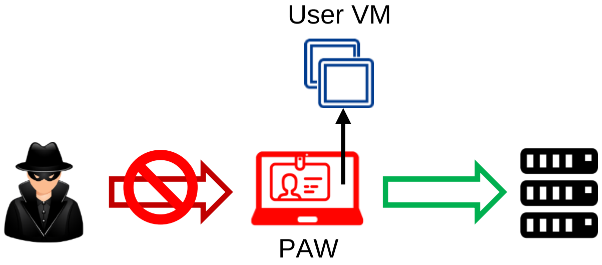
- Provide a dedicated secure administration environment for sensitive tasks that is protected from Internet attacks and sophisticated threat vectors
  - On an operating system level: Implement **Privileged Account Workstations** (PAW)
  - On an Active Directory level: Implement **Enhanced Security Administration Environment** (ESAE) and/or **PRIV** Forest(s)

- Extension of the recommended practice to use separate admin and user accounts for administrative personnel
  - Separates sensitive tasks and accounts from the daily use workstations and devices
  - Strong protection from phishing attacks, application and OS vulnerabilities, and various impersonation attacks (e.g. Pass-the-Hash or Pass-the-Ticket)

Clean Source Principle: PAWs

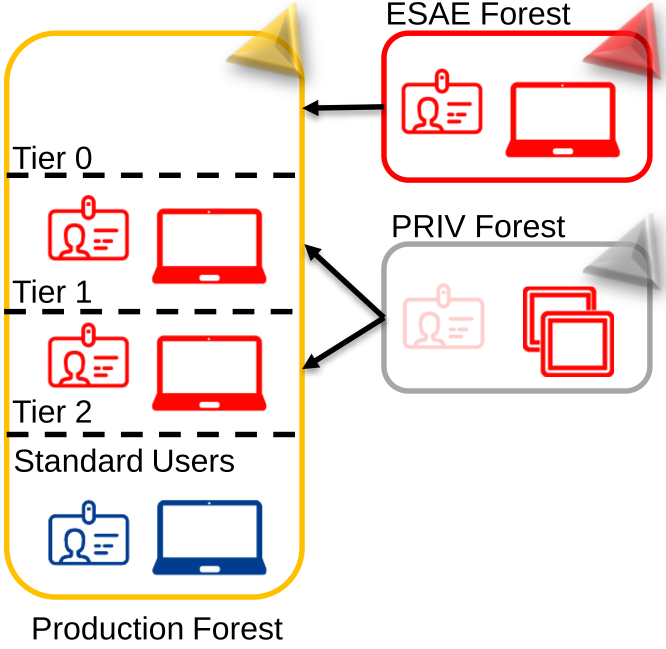
- PAW hardware profiles can be:
  - **Dedicated hardware**
    - Separate dedicated devices for user tasks vs. administrative tasks
  - **Simultaneous use**
    - Single device that can run user tasks and administrative tasks concurrently by taking advantage of OS or presentation virtualization. For example:
      - Adding a local user VM
      - Adding RemoteApp, RDP, or a VDI





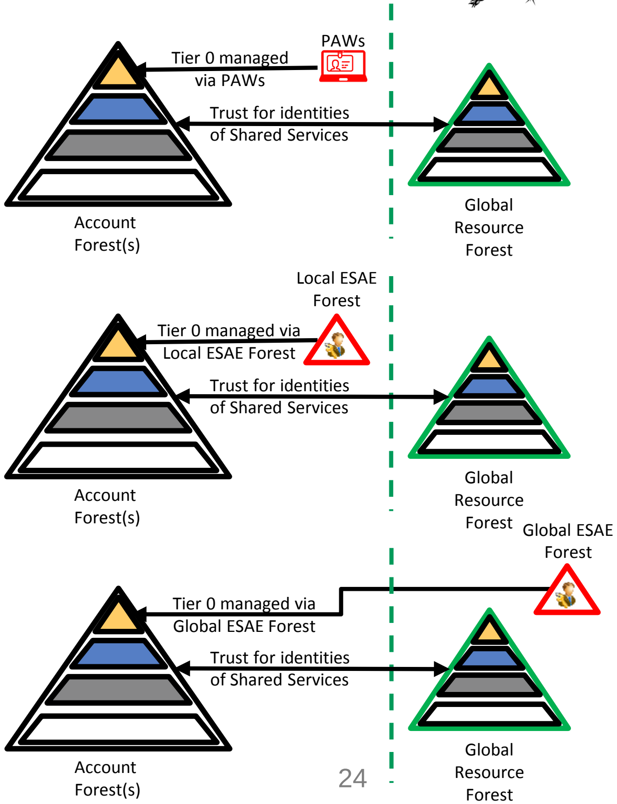
Clean Source Principle: ESAE/PRIV Forest

- Dedicated administrative forest
  - Hosts administrative accounts, workstations, and groups
  - Environment has stronger security controls than the production environment
- **ESAE forest** moves all sensitive objects for Tier 0 administration to a separate forest (except the krbtgt account)
- **PRIV forest** moves administrative identities for Tier 1 & 2 administration to a separate forest and combines this with a PAM solution (e.g. MIM 2016)

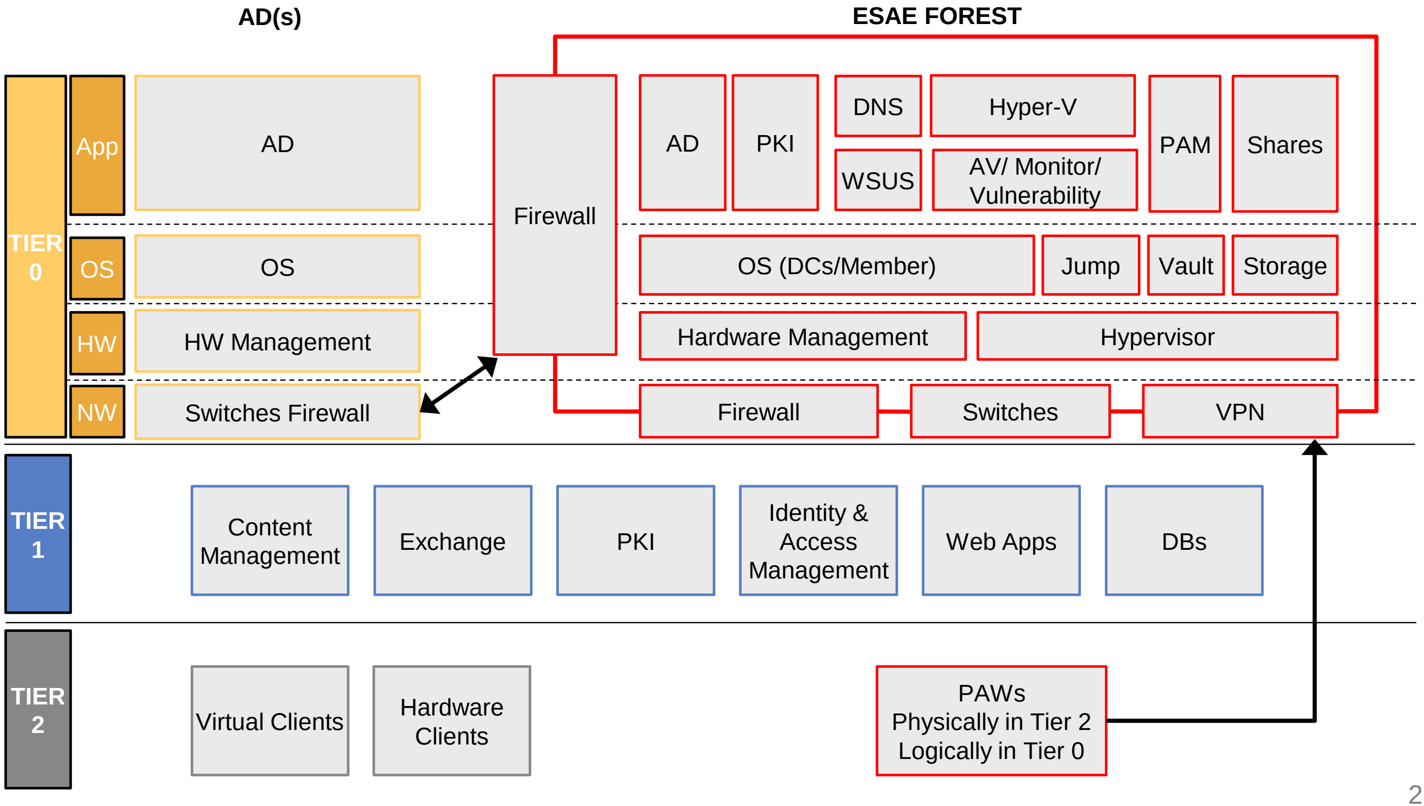


Exemplary Secure Administration Environment Models

- Option 1:
  - Tier 0 managed exclusively via PAWs
- Option 2:
  - Tier 0 managed by a Local ESAE Forest
- Option 3:
  - Tier 0 managed by a Global ESAE Forest
- All options can be combined with a PRIV Forest


PAWs
Tier 0 managed via PAWs
Trust for identities of Shared Services
Account Forest(s)
Global Resource Forest
Local ESAE Forest
Tier 0 managed via Local ESAE Forest
Trust for identities of Shared Services
Account Forest(s)
Global Resource Forest
Global ESAE Forest
Tier 0 managed via Global ESAE Forest
Trust for identities of Shared Services
Account Forest(s)
Global Resource Forest

Exemplary ESAE Forest Implementation

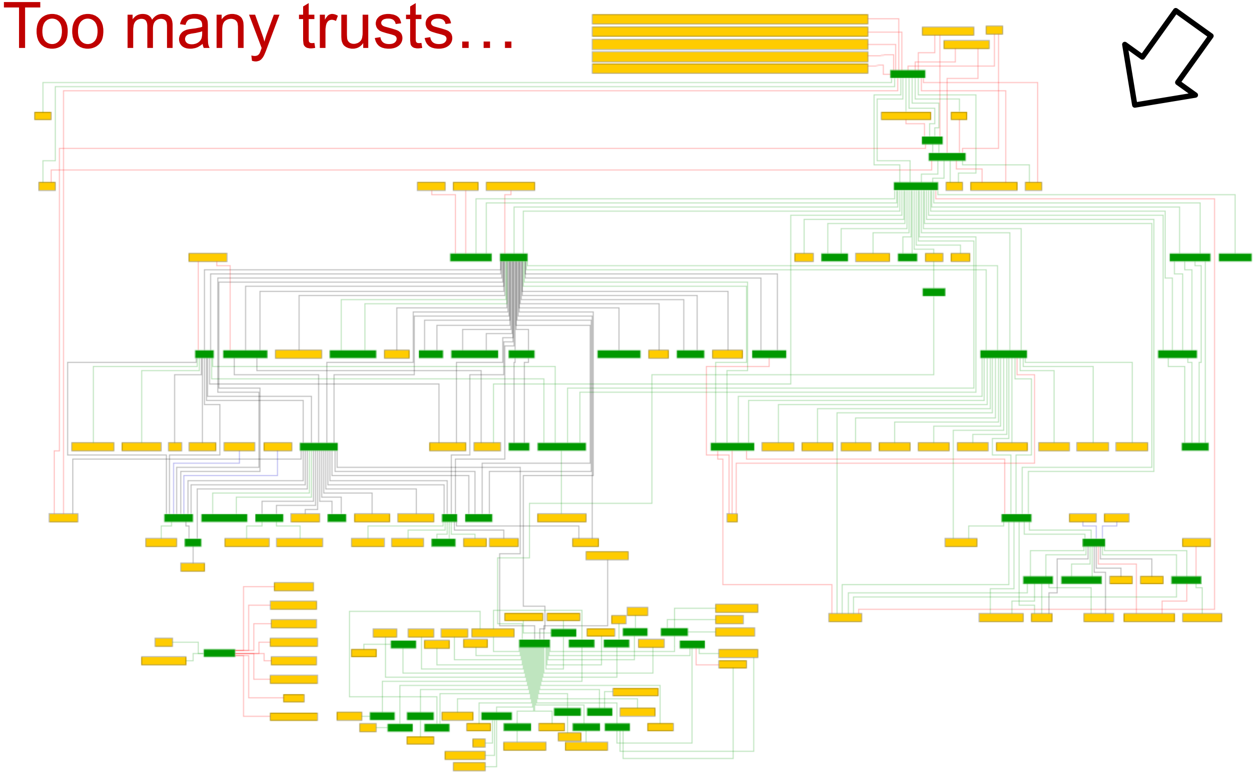


Mistake No. 4: **(AD) Borders Not Under Control**

The Problem: AD Borders Neither Well-defined Nor Controlled: Trusts

- Trusts are established without a (security) assessment of the trusted party
- Often too many trusts
- Trusts are “too open”
- Established trusts persist over many years
- Configuration errors: Privileged accounts of the trusted forest have a privileged group membership in the trusting forest

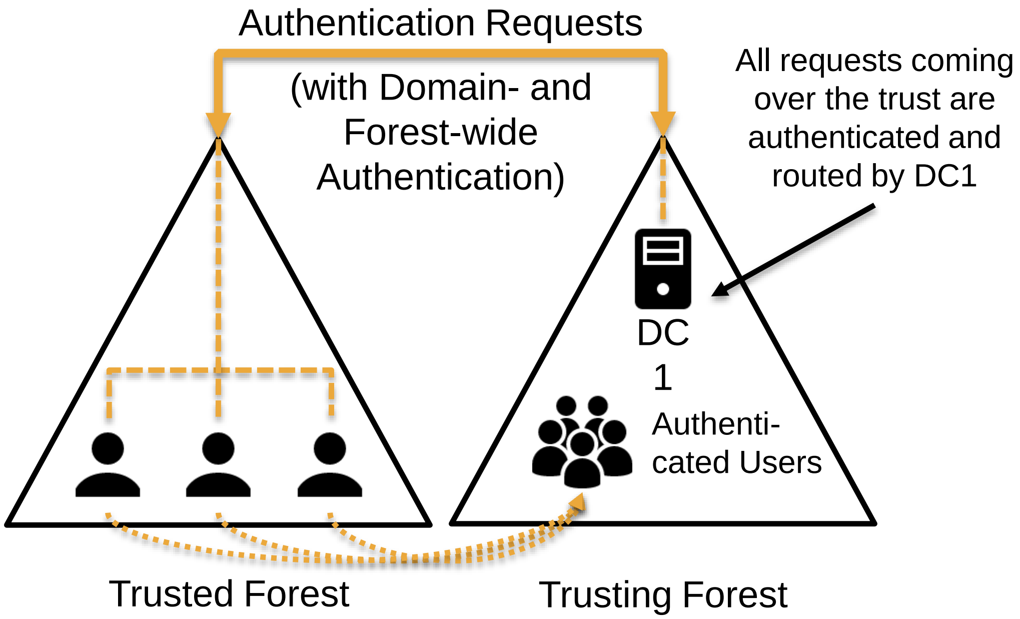
Too many trusts…



not a joke!

Trusts are too open…

- …in case of the Forest-wide authentication being enabled
- Users of the trusted Forest are in this case members of the *Authenticated Users* in the trusting Forest
- *Authenticated Users* are granted many of the default rights


Authentication Requests
(with Domain- and Forest-wide Authentication)
All requests coming over the trust are authenticated and routed by DC1
DC 1
Authenti-cated Users
Trusted Forest
Trusting Forest

The Problem: AD Borders Neither Well-defined Nor Controlled: DMZ

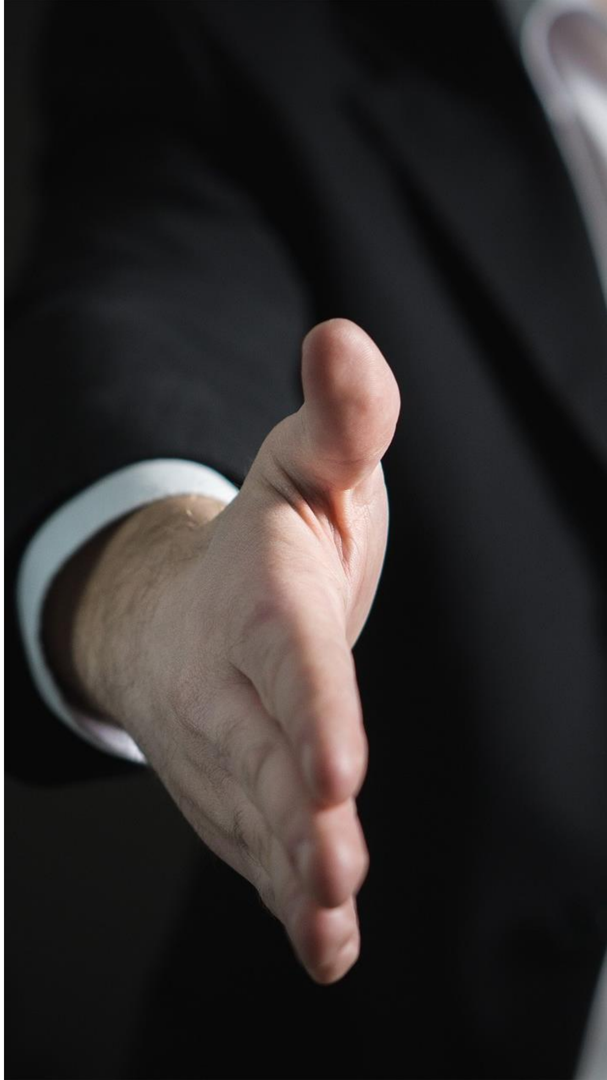
- Members of internal / productive AD are placed in the DMZ
  - High privileged admin or service accounts of internal AD log on to member systems in the DMZ
- Insecurely configured Trusts to DMZ AD
  - Bidirectional Trust
  - Forest- or Domain-wide authentication enabled

The Solution: AD Border & Trust Management

http://www.domainer.com.au/a-question-of-trust-2/

Trusts

- Be reluctant to and sparse with AD Trusts
- Perform a security assessment of the trusted AD *before* establishing the Trust…
  - …and know you own vulnerabilities ;-)
- Configure Trusts preferably:
  - Uni-directional
  - With selective Authentication
- Ensure that high privileged accounts span *only* their home AD Domain
- Review Trusts at least every six months



DMZ AD

- Strictly separate internal AD from DMZ AD
- Do not place even RODC (of internal AD) in the DMZ

Mistake No. 5: **Best Practices Lost in Time**

ERNW
providing security.

The Problem: Basics Are Overlooked

- Many AD security best practices exist for many years, but seem to be forgotten
  - Affects technical, as well as operational controls

- Most often seen in assessments:
  - Missing or outdated documentation
  - Insufficient network separation
  - Misconfiguration of the AdminSDHolder object
  - Orphaned AD objects
  - Delegation of permission underrepresented

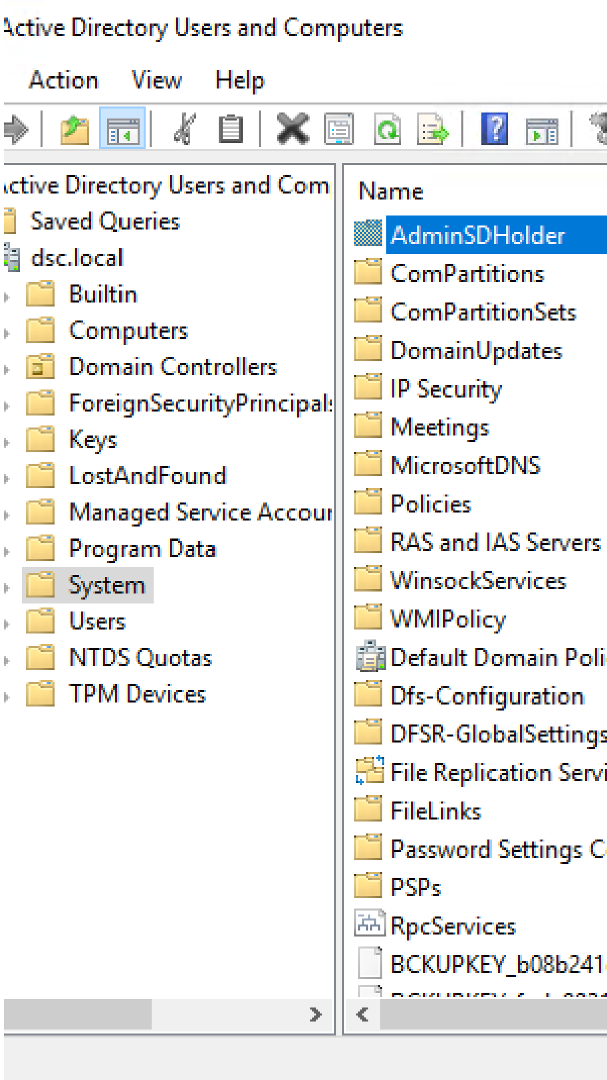
The Solution: Do the Basics

- Complete Documentation
  - Accordance with real configuration
  - Ensures protection and accurate view on the current state of the environment
  - Allows new personnel to become familiar with the environment in case of personnel shortages (e.g. illness)

- Network Isolation
  - Network infrastructure (physical) should reflect AD infrastructure (logical)
    - Avoid flat network structures
  - Network boundaries can be Forests or Administrative Tiers

AdminSDHolder Object

- Container object in the domain directory partition
- Security descriptor of this object is used as a template for all protected groups and users (e.g. Domain Admins)
  - If descriptors differ they are overwritten with those of the AdminSDHolder object
- The descriptor on this object should only be changed if absolutely necessary
  - Otherwise a new vector for a complete AD compromise is added



The Solution: Do the Basics

- AD Clean-up Process
  - Implement a process that takes care of:
    - Orphaned user accounts (from personnel which left the company)
    - Orphaned computer objects (from decommissioned systems)
    - Obsolescent group memberships

- AD Delegated Permissions
  - Allows delegating permissions without adding users to privileged groups
  - Grants users or groups only the permissions they need
  - Available via the Microsoft Management Console (MMC)

Mistake No. 6: **Too Many and Too Privileged Service Accounts**

ERNW
providing security.

The Problem: Overabundance of Service Accounts

- Not all service accounts are “**real**” service accounts
  - Sometimes **misused** as **personal accounts**

- Most of the time **passwords never expire**
  - Often in combination with **weak passwords**

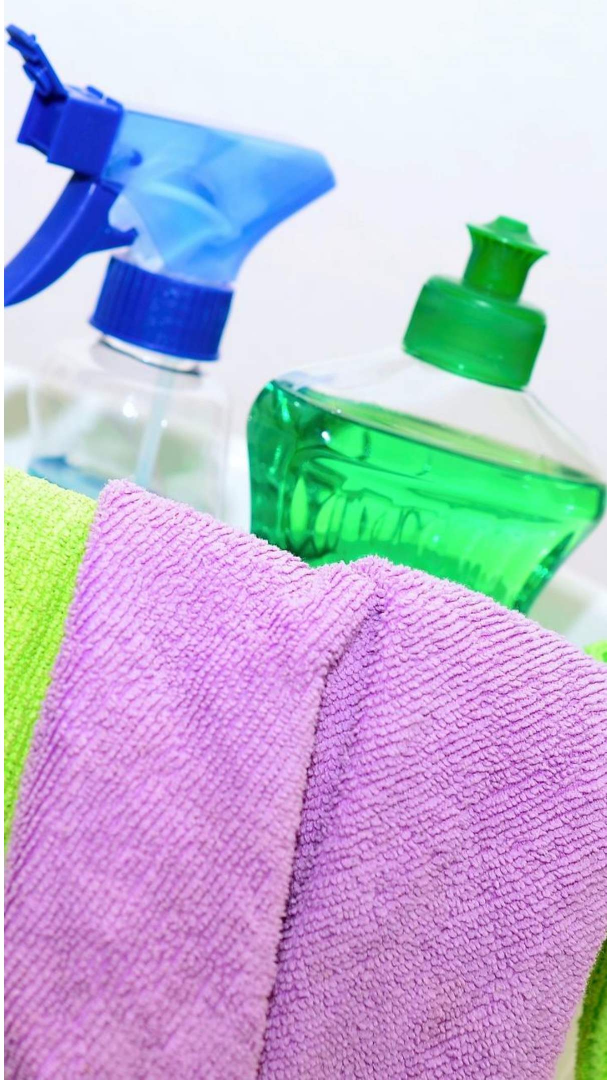
- Service accounts often **over-privileged**
  - Typical example: service accounts member of **Domain Admins** group

⚠ Usually one of the **first** targets of an attacker

ERNW
providing security.

The Solution: Service Account House Keeping

- **Regularly** check service accounts for validity
  - **Remove** all **unneeded** and **pseudo** service accounts
- **Remove** the "**Password never expires**" flag on as many service accounts as possible
- Make more service accounts **(Group) Managed Service Accounts**
- **Remove** unnecessary **privileges** from service accounts
  - Utilize **Active Directory Delegated Permissions**
  - Utilize **Temporary Group Membership** feature of Server 2016



Mistake No. 7: **Too Many Admins**

The Problem: Over-privileged Accounts

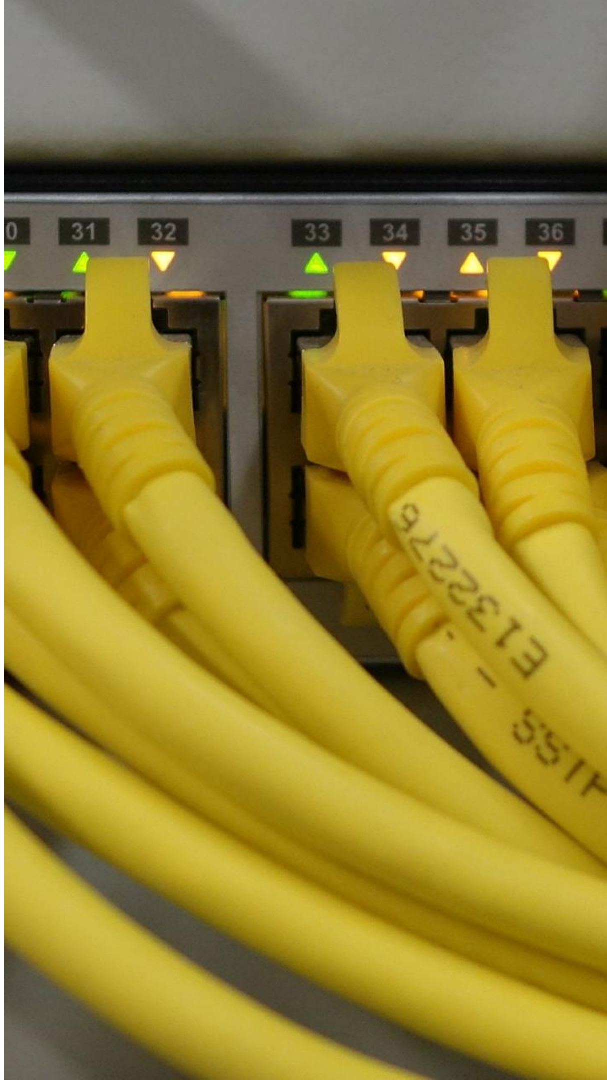
- Users often receive admin rights too easily
  - Locally, as well as in AD
  - Combined with missing role separation
- Service accounts also affected (see mistake no. 6)
- Active Directory Delegated Permissions rarely used
  - Instead focus on built-in groups
  - Prevents granular modification of rights
- Existing privileges not regularly checked

- Some numbers from various assessments:
- Example Domain I:
  - Enabled Users: 270
  - High-Priv Users: 49
  - Ratio: 18,15%
- Example Domain II:
  - Enabled Users: 1223
  - High-Priv Users: 150
  - Ratio: 12,26%

The Solution: Remove Privileges

- Make more users standard users
  - Grant permissions as granularly as possible, so you do not end up with hundreds of Domain Admins ;)
  - (Regularly) validate necessity for admin privileges
  - Local administrative privileges should only be granted in exceptional cases, as they are harder to manage

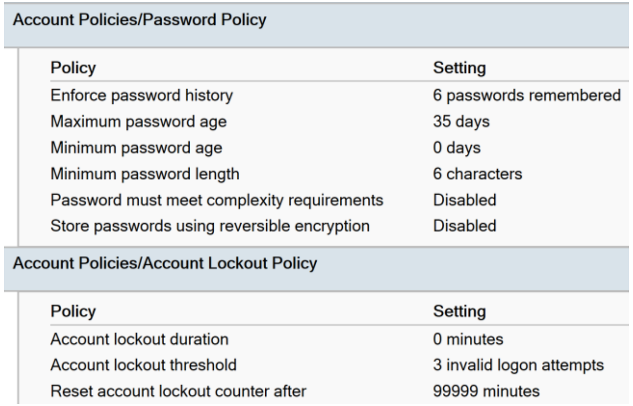
- Fix busted applications
  - Legacy software often falsely requires admin privileges
  - Can often be easily fixed (e.g. with Microsoft Application Compatibility Toolkit)



Mistake No. 8: **Using Bad Passwords**

The Problem: Bad Policies & User Awareness

- Password **policies in enterprises** are often **outdated**
  - Do not reflect **current threats** and **technological advances**
- Often only user accounts in focus, but not service accounts (e.g. passwords never expire)
- Users often have a **wrong idea** of **secure passwords**
  - Hard to remember for humans but easy to guess for computers
- May seem obsolete in the age of Pass-the-Credential attacks
  - Still relevant for an attacker aiming for a privilege escalation

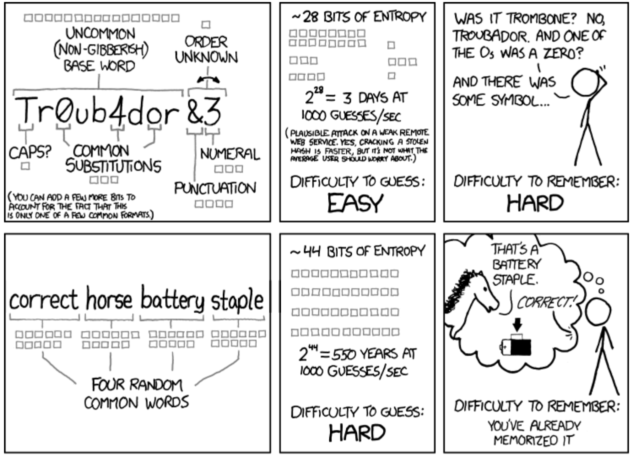


Account Policies/Password Policy

Policy	Setting
Enforce password history	6 passwords remembered
Maximum password age	35 days
Minimum password age	0 days
Minimum password length	6 characters
Password must meet complexity requirements	Disabled
Store passwords using reversible encryption	Disabled

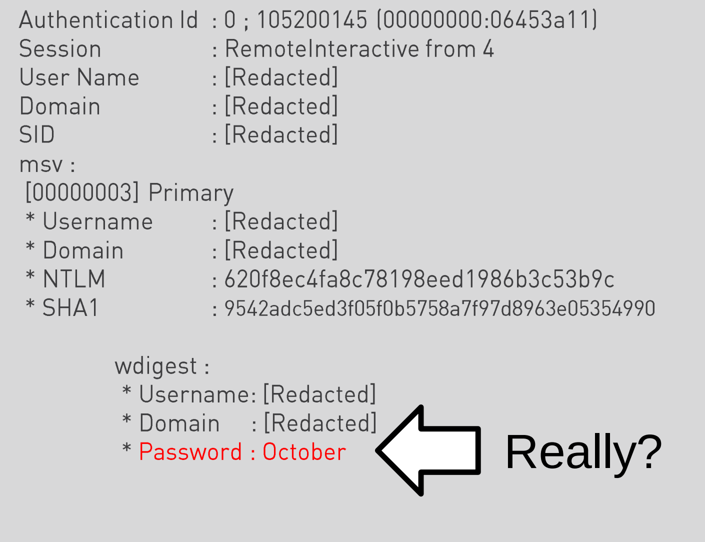
Account Policies/Account Lockout Policy

Policy	Setting
Account lockout duration	0 minutes
Account lockout threshold	3 invalid logon attempts
Reset account lockout counter after	99999 minutes


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor &3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^{28} = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^{44} = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Example I



```
Authentication Id  : 0 ; 105200145 (00000000:06453a11)
Session            : RemoteInteractive from 4
User Name          : [Redacted]
Domain             : [Redacted]
SID                : [Redacted]
msv :
 [00000003] Primary
 * Username        : [Redacted]
 * Domain          : [Redacted]
 * NTLM            : 620f8ec4fa8c78198eed1986b3c53b9c
 * SHA1            : 9542adc5ed3f05f0b5758a7f97d8963e05354990

         wdigest :
          * Username: [Redacted]
          * Domain   : [Redacted]
          * Password : October
```

Really?

- If users can they will choose a password, which fulfils the bare minimum

- If they have to change their password too often they try to work around it
  - You most probably can guess the other passwords of this user ;)

Example II

```
Authentication Id  : 0 ; 219681182 (00000000:0d18119e)
Session            : RemoteInteractive from 20
User Name          : [Redacted]
Domain             : [Redacted]
SID                : [Redacted]
msv :
 [00000003] Primary
 * Username        : [Redacted]
 * Domain          : [Redacted]
 * NTLM            : 3d8695acdd1747fa3f42e1fe4659a8f0
 * SHA1            : 50ab0e0504673f043e9b1fcdb7e0eb1af9cd0d5e

          wdigest :
           * Username: [Redacted]
           * Domain    : [Redacted]
           * Password : #Au20G08
```

Better?

- Might seem to be a better password at a first glance
- But:
  - Hard to remember
  - Only 8 characters
  - NTLM hash can be cracked in a few minutes with rainbow tables

The Solution: Update Password Policies

- Length > Complexity
  - Easier to remember
  - Can have a longer lifetime
  - Lockout thresholds can be higher

- All of this **increases** the **acceptance** and **reduces** operational **overhead**

- For **standard users**:
  - Use the Default Domain Policy
- For **high-privileged & admin accounts**:
  - Use Fine-Grained Password Policies
- For **Service Accounts**:
  - Use Fine-Grained Password Policies
  - Utilize (g)MSAs or implement a manual password reset mechanism
- For **local (admin) accounts**:
  - Utilize a management solution such as LAPS
  - ⚠ Do not use GPPs!

Recommended Password Requirements

Type	Min Age	Max Age	Min Length	History	Complexity Requirements	Lockout Threshold
Standard Users	1 day	180 days	12 characters	5 passwords	Yes	15 logon attempts
Admin Accounts	1 day	90 days	18 characters	10 passwords	Yes	10 logon attempts
Service Accounts	1 day	180 days	32 characters	20 passwords	Yes	20 logon attempts
Local Admin Accounts	1 day	30 days	18 characters	20 passwords	Yes	20 logon attempts
KRBTGT	Regular password resetting procedure every three months					

Mistake No. 9: **Running Outdated Operating Systems**

The Problem: Outdated Operating Systems

- A no-brainer for an attacker - attacking EoL OS
  - (Security) patches no longer released by the vendor
  - Exploits are some times even publically available

- Not a no-brainer, but a problem: outdated but still vendor-supported operating system versions
  - Legacy protocols
  - Insecure authentication mechanisms
  - Lack of modern, state-of-the-art security features

The Solution: Use Modern Operating System Versions

- Upgrade to new operating system versions,
- Substitute outdated systems,
- Decommission End-of-Life systems
  - If not possible: Isolation for example in an EoL Forest
    - Overall security-level should **not** be lowered
    - Creation of a separated environment for outdated systems

- Be aware of: Installation of new operating systems not enough
  - New operating system security features must also be actively used

The Solution: Use Modern Operating System Features

- Modern OS provide a lot of credential theft/reuse specific technologies

- Windows 8.1 / Server 2012 R2-specific security features
  - Authentication Policies & Silos
  - LSA Protection
  - Restricted Admin Mode for RDP

- Windows 10 / Server 2016-specific security features
  - Measured Boot and Remote Attestation
  - Virtualization-based Security
    - Device Guard
    - Credential Guard
  - Microsoft Passport

Mistake No. 10: **Vulnerable Systems and Applications Everywhere**

The Problem: Insufficient Patch Management

- Both operating system and third-party components often not up-to-date
- Regular patches and out-of-band patches both affected
  - Especially critical for OOB patches
- Usually insufficient or even no patch management at all

The Solution: Patch and Vulnerability Management

- Implementation of a proper patch and vulnerability management process for maintaining the overall security of a system
- Implement controlled patching of operating system components **and** third-party software
  - Ensure an appropriate patching time frame
- Define update procedures for security-critical (OOB) out-of-band patches guarantee roll-out in a timely manner

ERNW
providing security.

The Solution: Patch and Vulnerability Management

- Operating system patches are released monthly
  - Easy planning
  - Should be rolled-out within a week

- Application patches are released irregularly
  - A lot harder to plan for
  - Utilize security advisories and bulletins
  - Should be rolled-out within three weeks

- OOB patches must be seen as emergency changes
  - Should be rolled-out within 48 hours

Mistake No. 11: **No Active Directory-Specific Security Logging & Monitoring**

The Problem: No AD-Specific Security Logging & Monitoring

- AD-specific logging & monitoring is often restricted to AD service functionality (e. g. replication)
- Windows security monitoring often deferred to AV functionality (e. g. „AV will detect a compromise…“)
- Even if configured, security logs are not analyzed or are only analyzed in case of emergency
- Credential theft & reuse are often very difficult to detect.

The Solution: AD-Specific Security Logging & Monitoring

- Do the basics
  1. (1) Centralized logging & monitoring
  2. (2) Define **three Windows audit policies**:
     - A **baseline policy** for all Windows servers
     - A **high security policy** for high secure systems (Tier 0 & some Tier 1 systems (e. g. SAP), VIPs)
     - A very thorough **audit policy in case of assumed compromise** and for investigation cases
  3. (3) Acquire or ‚hire' AD monitoring know-how and allocate resources and personnel

The Solution: AD-Specific Security Logging & Monitoring

- Implement Admin Tiers with logon & control restrictions and monitor violations
  - Begin with Tier 0
  - Then Tier 1 & VIPs

- Monitor at least:
  - Tier 0 logons (and logon failures)
  - High privileged group membership changes
  - Violations of allowed logon types (e. g. interactive logon of service accounts)
  - Changes of attributes for sensitive AD objects (e. g. AdminSDHolder object)
  - Violations of allowed Kerberos encryption algorithms
  - Large amounts of enumeration errors
  - Some specific kerberos events IDs on DCs (e. g. ID 7 and ID 4769)
  - Yara rules for mimikatz & wce specific usage & strings

ERNW
providing security.

TROOPERS
MAKE THE WORLD A SAFER PLACE

Thank you for your time!

✉ fkuhn@ernw.de
hwiederkehr@ernw.de

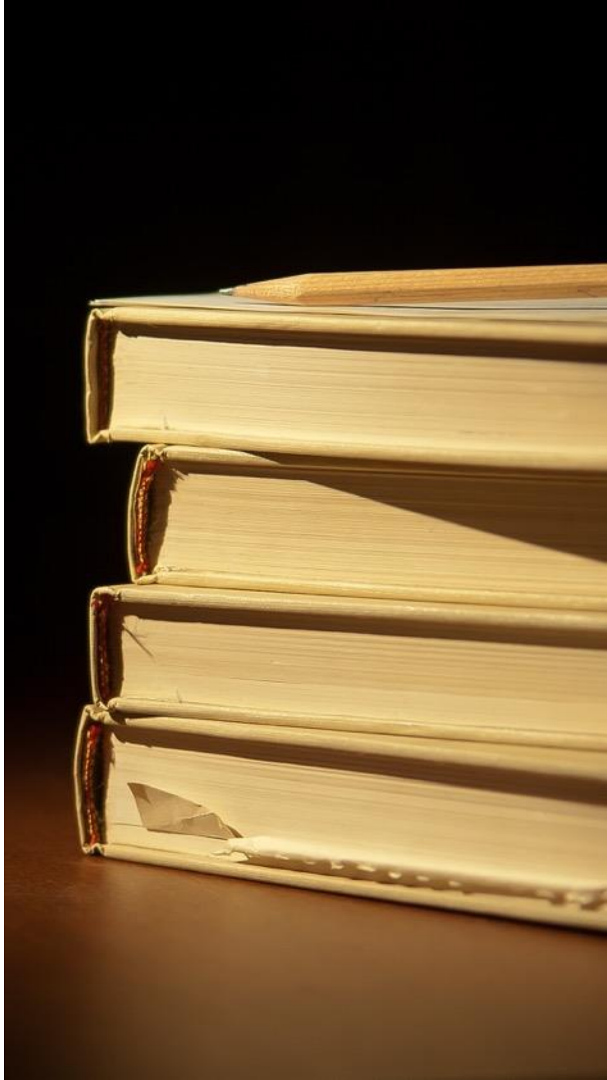
www.ernw.de

www.insinuator.net

Sources

- Link1
  - Ross Anderson, Security Engineering

- Icons
  - https://icons8.com/

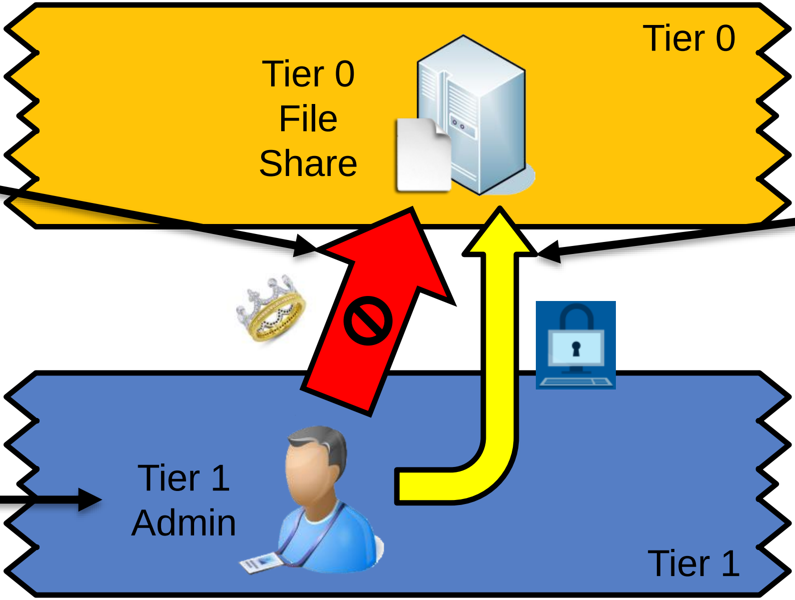


Additional Material & Information

Control/Logon Restrictions Example 1 for Admin Tiers

As the user is a tier 1 admin, he cannot control the file share system /the resource (he can only access a share with limited NTFS permissions)

Tier 1 admin must access a Tier 0 file share to store certain files

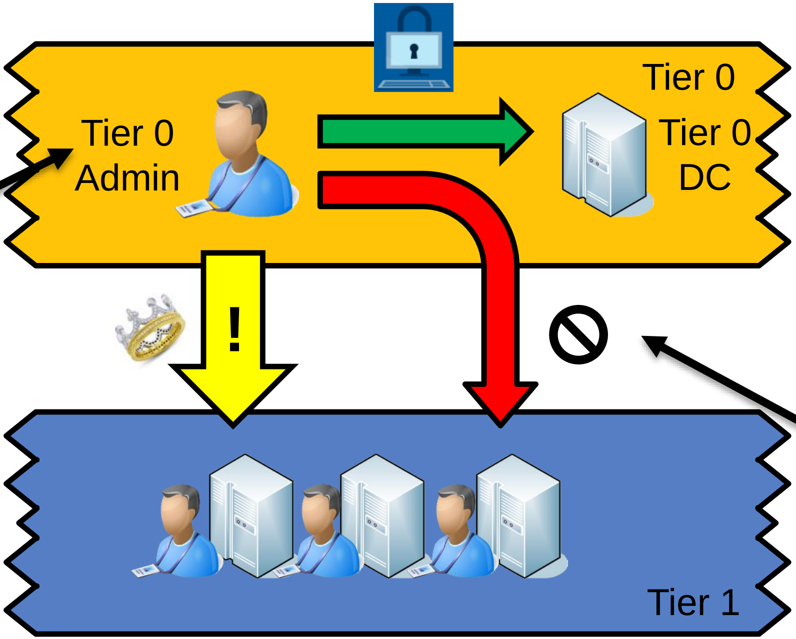


As required by his role, the Tier 1 admin can logon to a higher-tier resource to access a share and store files (well-defined and strictly monitored)

Note: A similar scenario is the access to the Netlogon share

Control/Logon Restrictions Example 2 for Admin Tiers

Tier 0 admin manages the identity store (Active Directory database). He can define group membership of Tier 0, Tier 1 (and Tier 2) accounts and he can define security settings for Tier 0 und Tier 1 servers (and even Tier 2 computers) in GPOs.



Therefore, the Tier 0 admin must access *dsa.msc* and *gpmc.msc* on a DC (where he logs on).

Thus, as required by his role, the Tier 0 admin can control lower-tier resources, but he never logs on to a lower-tier system.

The Problem: AD Borders Neither Well-defined Nor Controlled: AD Extension Into the Cloud

- Many different scenarios possible
  - Application services in Azure (WebApp, SharePoint, SQL, SAP…)
  - Domain Controller(s) in Azure (for Backup-up or authentication reasons)

- Some scenarios require synchronization of credentials to Azure

ERNW
providing security.

Azure (Cloud)

- Extension of internal AD via DirSync/ADConnect or member systems in Azure should require a strategic decision

- A connection via ADFS between on-prem AD and Azure is able to restrict on-prem credentials to on-prem AD