

# Making \$ With COMPUTER\$

Queen City Con 0x3

John Askew & Jake  
Hildreth

2025-11-07



# John

Hacker

Red Team Lead



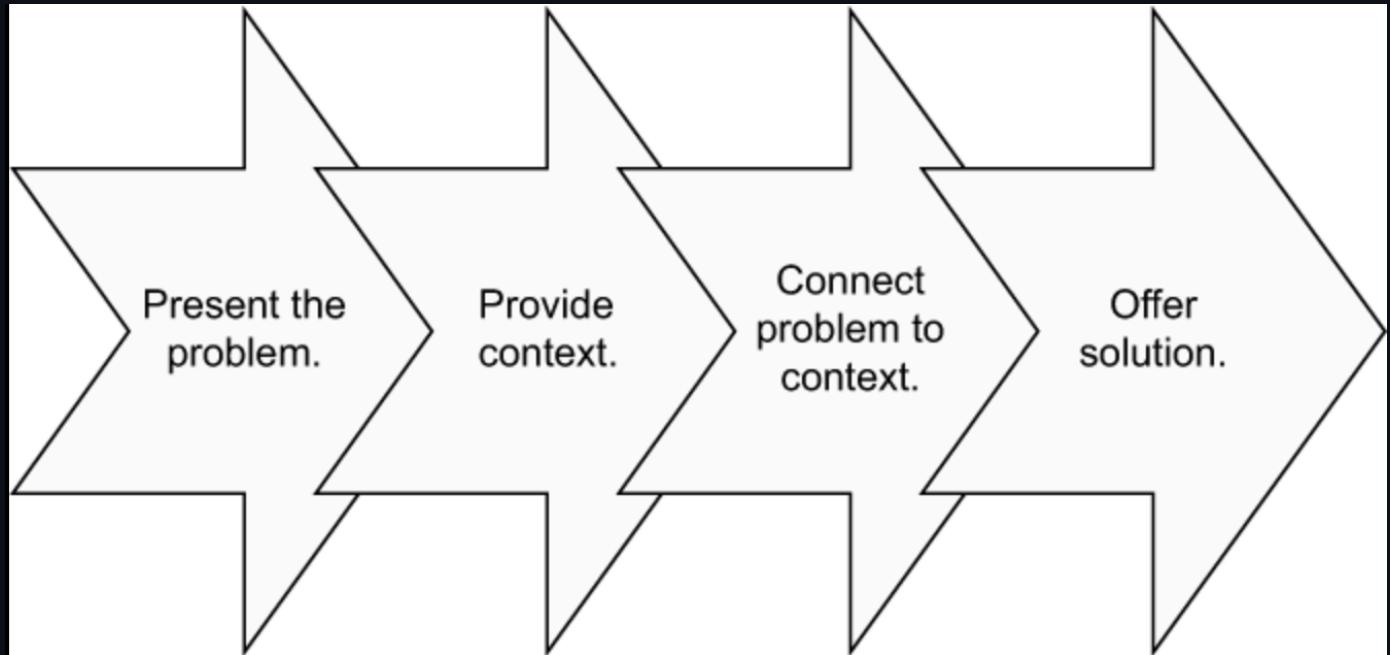
# Jake

Defender  
Security Consultant



# Agenda

- The Problem
- Some Context
- A Demo or Two
- Some Solutions



# The Problem

**If a user can join a computer to your domain  
they can own your Active Directory forest**

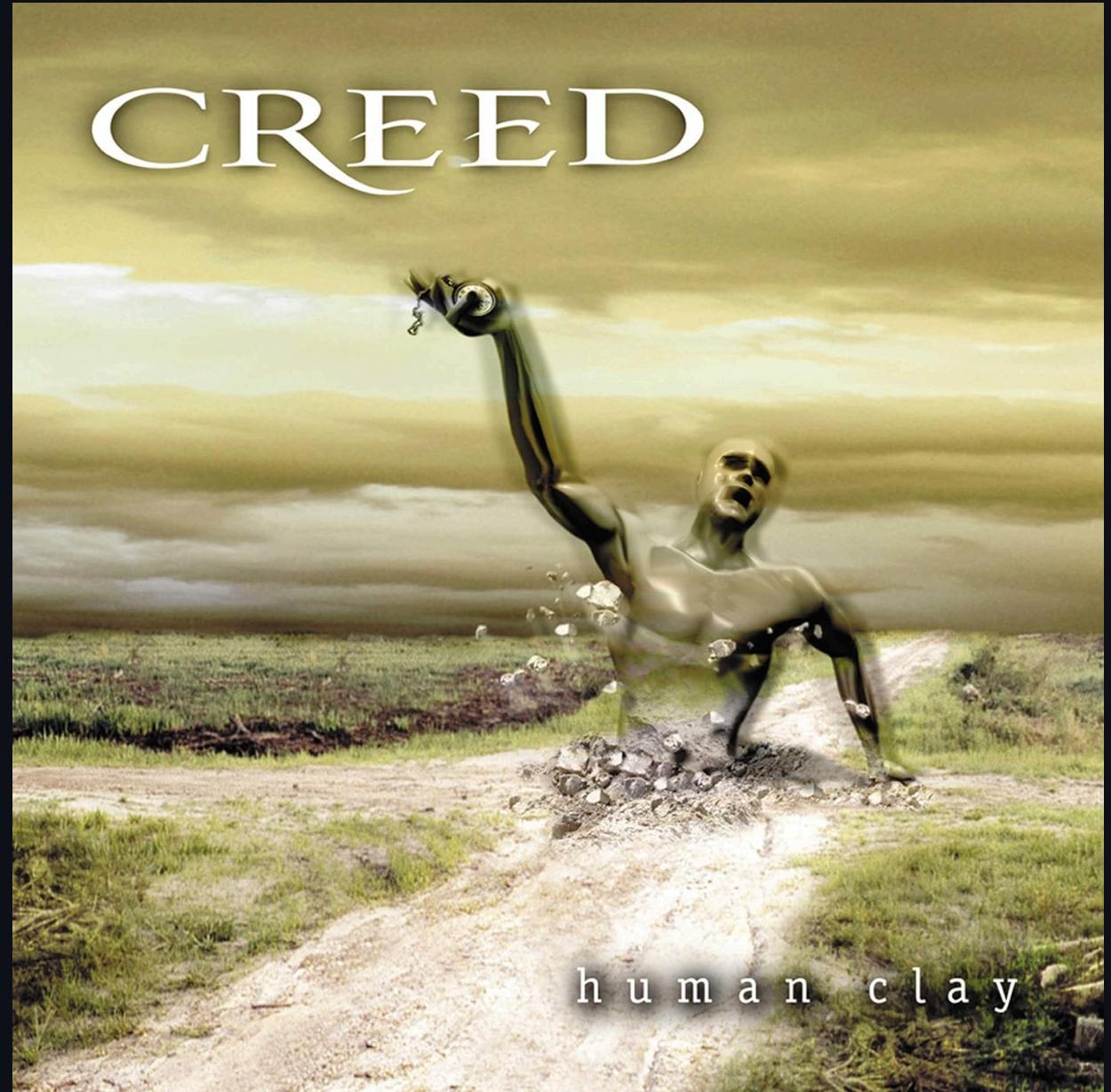
**If a user can join a computer to your domain**

**they can own your Active Directory forest in minutes**

# The Problem

- `ms-DS-MachineAccountQuota` attribute
  - Number of computers a single user can add to the domain
  - Active Directory (AD) default value: 10
- `SeMachineAccountPrivilege` User Right
  - Who is allowed to add computers to a domain
  - AD default: Authenticated Users
- Made sense 25 years ago
- But now computers are more dangerous!

This Also Made  
Sense 25 Years  
Ago:



# Why This Matters:

- MACHINE\$ accounts are valuable to attackers!
- MANY attack chains need an attacker-controlled machine account
- Defaults = easier to create one than to compromise one
- Remove defaults -> break attack chains

# Some Context

# What Makes Computer Accounts Special?

- Password differences:
  - Complex & 120 characters long
  - Changed automatically every 30 days
- Service Principal Names (SPNs)
  - Tells others what services are available
- Note: local admin password is NOT the computer account password

# The Attacker's Advantage

- Machine accounts...
  - tend to be less scrutinized (evasion, persistence)
  - often have different permissions (privilege escalation)
  - can be created without creds using relaying (initial access)
- Controlling an SPN is a powerful attack primitive!
  - (You are a legitimate Active Directory service)

# Real-World Experience

- How common is this?
  - Jake: 80% at default
  - John: never seen it set properly when first engaging a customer
- Why hasn't this been fixed?
  - Relatively unknown outside security circles
  - Conflicting hardening guidance
  - Operations > Security

A Demo or Two

A Demo or Two (It's Actually Three)

# Demo Environment

- Game of Active Directory (by Mayfly277)
  - <https://github.com/Orange-Cyberdefense/GOAD>
- Tools used:
  - <https://github.com/fortra/impacket>
  - <https://github.com/PennywOrth/NetExec>
  - <https://github.com/CravateRouge/bloodyAD>
  - <https://github.com/ly4k/Certipy>



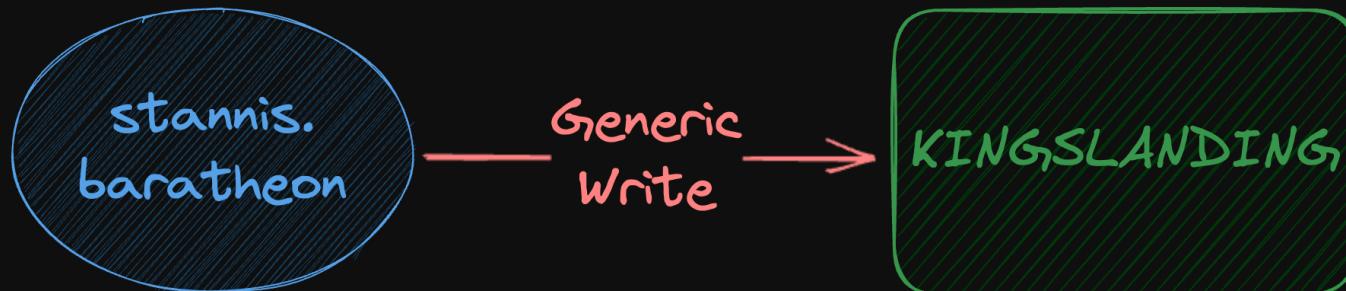
## Demo: Privilege Abuse

- Look for extra permissions granted to Domain Computers
  - Added to privileged groups?
  - Access to other AD objects via ACLs?
- Create a machine account and you can abuse those permissions

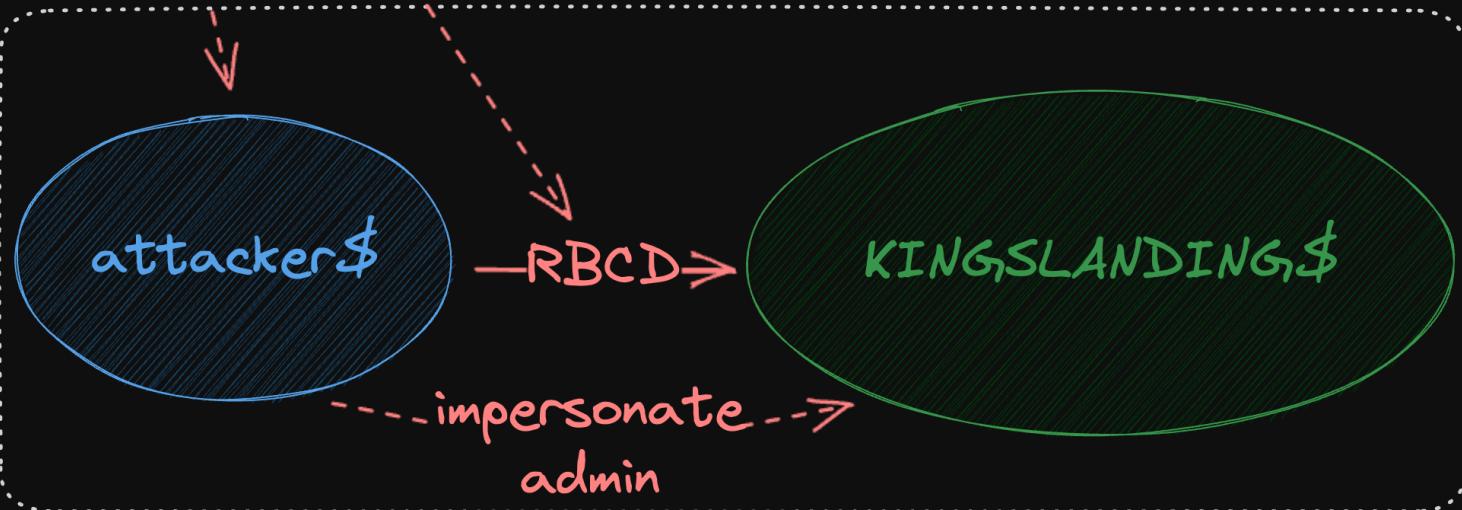
# Demo: Delegation Abuse

- Active Directory account delegation is EASY to misconfigure
- "Resource-Based Constrained Delegation" is maybe the most commonly abused
- Turn a "GenericWrite" permission on a computer object into a full compromise
- <https://eladshamir.com/2019/01/28/Wagging-the-Dog.html>

## PREREQUISITES



## RESULT



# Demo: AD CS Abuse

- Active Directory Certificate Services is EASY to misconfigure
- Domain Computers are often allowed to enroll templates (ESC1)
- Turn `altSecurityIdentities` write access into a full compromise (ESC14A)
  - (The default Domain Computers cert template meets all other attack requirements)
- References:
  - <https://posts.specterops.io/certified-pre-owned-d95910965cd2>
  - <https://posts.specterops.io/adcs-esc14-abuse-technique-333a004dc2b9>

# Many Other Attacks

- Persistence using "stale" machine accounts
- Unconstrained Delegation & Constrained Delegation attacks
- CVE-2021-42278 & 42287 ("noPAC" privilege escalation)
- "SPN-in-the-Middle"
- "DumpGuard" technique
  - <https://specterops.io/blog/2025/10/23/catching-credential-guard-off-guard/>
- Whatever next thing comes out

# Some Solutions

# Prevention:

## Set **ms-DS-MachineAccountQuota** to 0

```
#requires -Modules ActiveDirectory

# Set variables
$MAQ = 'ms-DS-MachineAccountQuota'
$Domain = Get-ADDomain -Identity example.com

# Set Correct Value
Set-ADDomain -Identity $Domain -Replace @{$MAQ=0}
```

Now only Administrators can add computers to domain without first precreating a computer account.

# Prevention:

## Restrict SeMachineAccountPrivilege

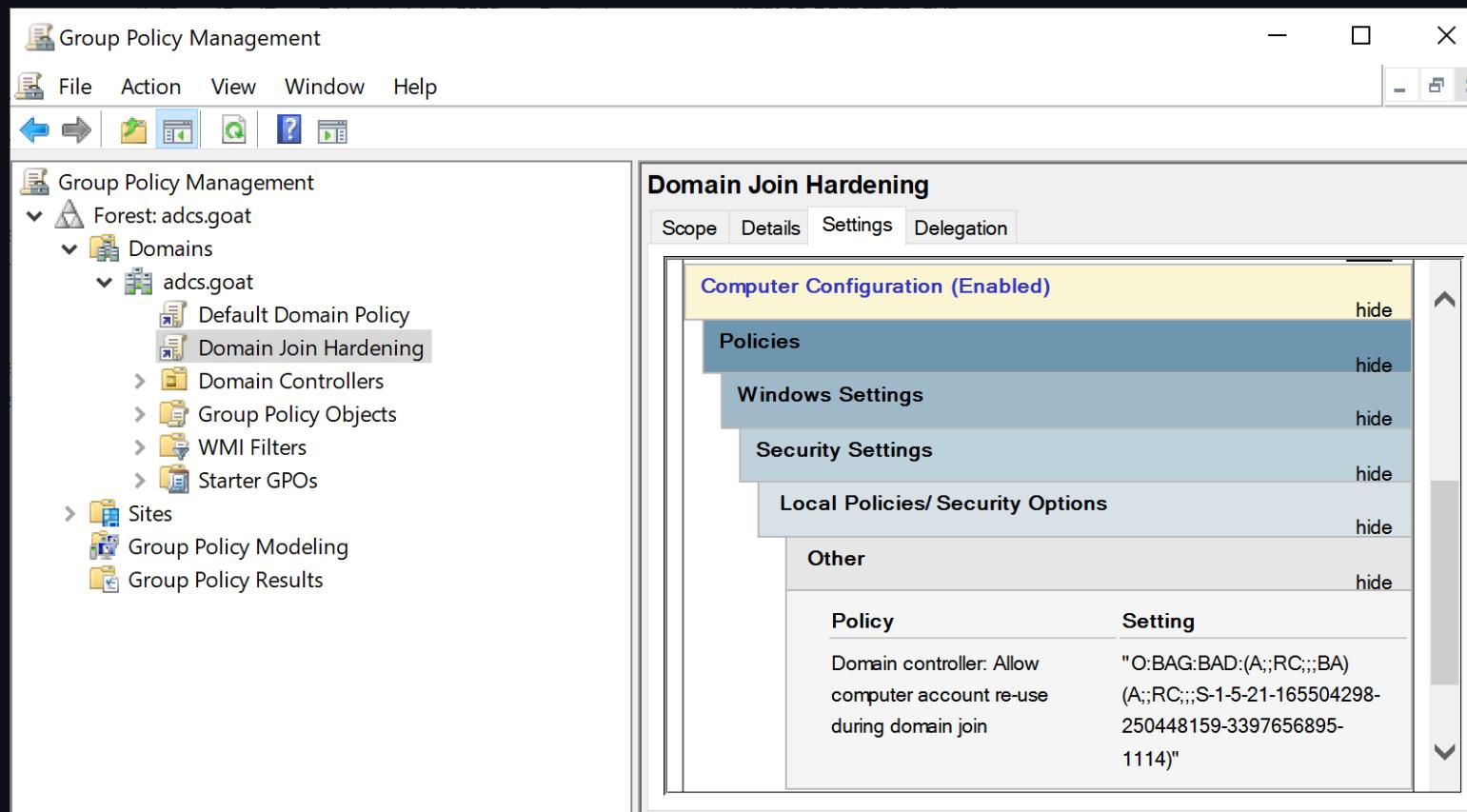
The screenshot shows the Windows Group Policy Management console. The left pane displays a tree structure of group policies under 'Forest: adcs.goat'. In the 'Domains' section, 'adcs.goat' is expanded, showing 'Default Domain Policy', 'Domain Controllers', 'Group Policy Objects', 'WMI Filters', and 'Starter GPOs'. Under 'Domain Controllers', 'Restrict SeMachineAccountPrivilege' is selected. The right pane shows the 'Restrict SeMachineAccountPrivilege' dialog box. The 'Settings' tab is selected, displaying the 'Computer Configuration (Enabled)' section. The 'Policies' section contains the 'Restrict SeMachineAccountPrivilege' policy, which is highlighted. Below it are sections for 'Windows Settings' and 'Security Settings', both of which have a 'hide' button next to them. The 'Local Policies/ User Rights Assignment' section also has a 'hide' button. At the bottom, a table lists a single policy assignment:

Policy	Setting
Add workstations to domain	adcs\ Computer Joiners

# Prevention:

## Follow New Domain Join Guidance: Additional Prep

Configure Trusted Computer Account Owners:



The screenshot shows the Group Policy Management console. The left navigation pane shows a tree structure under 'Forest: adcs.goat' with 'Domains' expanded, showing 'adcs.goat' which contains 'Default Domain Policy', 'Domain Join Hardening' (which is selected), 'Domain Controllers', 'Group Policy Objects', 'WMI Filters', and 'Starter GPOs'. Below 'adcs.goat' are 'Sites', 'Group Policy Modeling', and 'Group Policy Results'. The right pane is titled 'Domain Join Hardening' and contains tabs for 'Scope', 'Details', 'Settings', and 'Delegation'. The 'Settings' tab is active, showing the 'Computer Configuration (Enabled)' section. This section includes categories for 'Policies', 'Windows Settings', 'Security Settings', 'Local Policies/ Security Options', and 'Other'. Under 'Other', there is a table with two rows:

Policy	Setting
Domain controller: Allow computer account re-use during domain join	"O:BAG:BAD:(A;;RC;;;BA)(A;;RC;;;S-1-5-21-165504298-250448159-3397656895-1114)"

## Prevention:

### Follow New Domain Join Guidance: Perform

1. Admin01, a *Trusted Computer Account Owner*, pre-creates the Computer object in a target OU/container
2. Admin02, a *Computer Joiner*, performs domain join w/minimal privileges required

More details, including required permissions for each admin:

<https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/active-directory-domain-join-permissions>

# Prevention:

## Offline Domain Join (more secure!)

1. Using a *Trusted Computer Account Owner* with appropriate permissions, provision a new computer object:

```
djoin /provision /domain contoso.com /machine NewComputer /savefile offlinedomainjoin.txt
```

2. On physical computer, complete the join:

```
djoin /request0DJ /loadfile offlinedomainjoin.txt /windowspath %SystemRoot% /localos
```

**No additional permissions are required!**

More details:

<https://learn.microsoft.com/en-us/windows-server/remote/remote-access/directaccess/directaccess-offline-domain-join>

# Detection

## Monitor for New Machine Accounts

- Event ID 4741 is logged on Domain Controllers
- (you ARE collecting logs from ALL your domain controllers... right!?)

General Details

A computer account was created.

Subject:

Security ID:	SEVENKINGDOMS\stannis.baratheon
Account Name:	stannis.baratheon
Account Domain:	SEVENKINGDOMS
Logon ID:	0x7A24050B

New Computer Account:

Security ID:	SEVENKINGDOMS\test\$
Account Name:	test\$
Account Domain:	SEVENKINGDOMS

Attributes:

SAM Account Name:	test\$
Display Name:	<value not set>
User Principal Name:	-
Home Directory:	<value not set>
Home Drive:	<value not set>
Script Path:	<value not set>
Profile Path:	<value not set>
User Workstations:	<value not set>
Password Last Set:	<never>
Account Expires:	<never>
Primary Group ID:	515
AllowedToDelegateTo:	-
Old UAC Value:	0x0
New UAC Value:	0x84
User Account Control:	'Password Not Required' - Enabled 'Workstation Trust Account' - Enabled
User Parameters:	<value changed, but not displayed>
SID History:	-
Logon Hours:	<value not set>
DNS Host Name:	-
Service Principal Names:	-

Additional Information:

Privileges	SeMachineAccountPrivilege
------------	---------------------------

Log Name: Security

Source: Microsoft Windows security Logged: 11/1/2025 9:30:54 PM

Event ID: 4741 Task Category: Computer Account Management

Level: Information Keywords: Audit Success

User: N/A Computer: kingslanding.sevenkingdoms.local

OpCode: Info

More Information: [Event Log Online Help](#)

# Detection

## Investigate Machine Accounts

- The machine account creator is indicated in the `mS-DS-CreatorSID` field
- By default the user account cannot delete the machine account it created
- <https://learn.microsoft.com/en-us/windows/win32/adschema/a-ms-ds-creatorsid>

The screenshot shows the Active Directory Explorer interface. On the left, the tree view displays the structure of the default domain, specifically the `DC=sevenkingdoms,DC=local` container, including various OUs like `CN=Builtin`, `CN=Computers`, and `CN=test`. On the right, a table provides detailed information about the attributes of a specific object, likely a machine account named `test` under `CN=Computers`.

Attribute	Syntax	Count	Value(s)
<code>accountExpires</code>	<code>Integer8</code>	1	<code>0xFFFFFFFFFFFFFF</code>
<code>badPasswordTime</code>	<code>Integer8</code>	1	<code>0x0</code>
<code>badPwdCount</code>	<code>Integer</code>	1	<code>0</code>
<code>cn</code>	<code>DirectoryString</code>	1	<code>test</code>
<code>codePage</code>	<code>Integer</code>	1	<code>0</code>
<code>countryCode</code>	<code>Integer</code>	1	<code>0</code>
<code>distinguishedName</code>	<code>DN</code>	1	<code>CN=test,CN=Computers,DC=sevenkingdoms,DC=local</code>
<code>dSCorePropagationData</code>	<code>GeneralizedTime</code>	1	<code>1/1/1601 12:00:00 AM</code>
<code>instanceType</code>	<code>Integer</code>	1	<code>4</code>
<code>isCriticalSystemObject</code>	<code>Boolean</code>	1	<code>FALSE</code>
<code>lastLogoff</code>	<code>Integer8</code>	1	<code>0x0</code>
<code>lastLogon</code>	<code>Integer8</code>	1	<code>0x0</code>
<code>localPolicyFlags</code>	<code>Integer</code>	1	<code>0</code>
<code>logonCount</code>	<code>Integer</code>	1	<code>0</code>
<code>mS-DS-CreatorSID</code>	<code>Sid</code>	1	<code>S-1-5-21-3343131542-3316494596-4064008797-1120</code>
<code>name</code>	<code>DirectoryString</code>	1	<code>test</code>

## Remediation:

### Find Inactive and Suspicious Computer Objects:

Huy Kha (aka DebugPrivilege) wrote an article with an easy-to-use script that finds computer accounts that look *funky*

# Machines Gone Rogue



DebugPrivilege

Follow

6 min read · 1 day ago

<https://medium.com/@Debugger/machines-gone-rogue-a01d726f5f10>

Outro



# Key Takeaways

- Design decisions from 25 years ago aren't always so great
- Computer accounts are more dangerous than most people realize
- There are multiple attack paths from COMPUTER\$ to \$
- Defense requires technical controls AND organizational change



# Call to Action

- Check your environment TODAY
- Start the conversation with your security and identity teams
- Don't wait for an incident to fix this



# Resources

<https://github.com/jakehildreth/QueenCityCon25>



# Thanks!

	<b>John Askew</b>	<b>Jake Hildreth</b>
Email	<a href="mailto:john@terrapinlabs.io">john@terrapinlabs.io</a>	<a href="mailto:jake@jakehildreth.com">jake@jakehildreth.com</a>
Web	terrapinlabs.io	jakehildreth.com
GitHub	sk3w	jakehildreth
LinkedIn	/in/sk3w	/in/jakehildreth
BlueSky	@sk3w.bsky.social	@dotdot.horse
QR 😊		